HEXAGON

# HxGN Robotic Automation

# User Manual

# v.2026.1

# Table of Contents

**HxGN Robotic Automation** ..... **1**

**Table of Contents** ..... **2**

**Introduction** ..... **1**

**About the Manual** ..... **4**

What's new this release ..... 4

ZG Integration ..... 4

ABB Robot Integration ..... 4

A new Metrology API ..... 5

CLMS Integration ..... 5

Splitting a PRESTO XL cell into individual and simultaneously working PRESTO L cells ..... 5

Mirroring and Reversing a program ..... 5

Other Changes: ..... 5

**Getting things ready!** ..... **7**

Software and Hardware Requirements ..... 8

System Hardware Requirements ..... 8

Software Requirements ..... 9

Minimum Metrology Software Versions ..... 10

Leica Software ..... 10

Leica AS1 ..... 10

Leica T-Scan ..... 11

ZG software ..... 12

Robot Software ..... 12
Software Installation ..... 13
HxGN Robotic Automation Installation ..... 13
Services and Ports ..... 22
Installation order Erlang and RabbitMQ ..... 23
Erlang Installation ..... 24
RabbitMQ Installation ..... 28
Troubleshooting RabbitMQ ..... 31
HyperScan installation & Configuration ..... 33
Camera driver installation ..... 33
Scanning Program Installation ..... 35
Scanning Program Configuration ..... 37
Device Management ..... 39
Configuration and Licensing ..... 41
Licensing ..... 41
Nalpeiron Zentitle ..... 41
Hexagon Unified Licensing ..... 45
Online activation ..... 45
Offline activation ..... 47
CLMS Licensing ..... 49
Online License Activation ..... 49
Offline License Activation ..... 51
Query Existing License ..... 53
User Management ..... 55

User Management Portal ..... 59
JSON configuration through App Manager ..... 62
Generic Host ..... 62
Metrology ..... 68
Cell JSON ..... 70
Tracker bridge ..... 79
PLC Communication ..... 82
Configuration Troubleshooting ..... 84
Generic Host is not running before opening HxGN Robotic Automation ..... 84
Generic Host crashes or fails to restart ..... 84
Cell connection is not visible in HxGN Robotic Automation ..... 85
Database issues during program export ..... 85
Splitting a cell into 2 smaller cells ..... 86
Cell Hardware Setup ..... 90
FANUC robot ..... 91
Configuration ..... 93
Installation ..... 98
Execution ..... 101
Troubleshooting ..... 103
KUKA robot ..... 104
Configuration ..... 104
Installation ..... 105
Execution ..... 105
Troubleshooting ..... 106

ABB robot ..... 107
Configuration ..... 107
Installation ..... 107
Execution ..... 108
Troubleshooting ..... 113
PLC ..... 113
Configuration ..... 113
Execution ..... 115
Troubleshooting ..... 116
Trackers and Scanners ..... 116
Leica Scanners ..... 116
AS1 Scanning extension ..... 116
ZG Scanners ..... 117
TCP and External Axis Configuration ..... 117
How to define the TCP of the sensor on the Fanuc Robot ..... 117
How to ensure that the TCP is defined properly on the Fanuc robot ..... 119
Leica AS1 TCP ..... 119
Case 1. Fanuc Robot is fixed ..... 120
Case 2. Fanuc Robot mounted on slider or 7th axis. ..... 121
**UI Overview ..... 132**
HxGN Robotic Automation ..... 132
Application Mode Tabs ..... 134
Modelling ..... 135
Cell Layout ..... 135

Alignment ..... 135
Programming ..... 136
Execution ..... 137
Specific Application Mode Toolbar ..... 138
Special Mode Toolbar ..... 138
Regular Selection ..... 139
Polygon Selection ..... 140
Polygon Features Selection ..... 141
Pick on CAD ..... 141
Show/hide grid ..... 141
Enable/disable drag ..... 143
Enable/disable snapping ..... 144
Alignment Mode ..... 145
Measurement Mode ..... 146
Snapping Frame Generation Mode ..... 147
Attach Frame Generation Mode ..... 151
Feature Generation Mode ..... 154
Project Tree / Library / Alignment / Program Windows ..... 154
Search bar and Advanced Search Options ..... 156
Program and Subroutine filtering ..... 161
Geometries / Measurements Windows ..... 162
Active Machine / Active Sensor / Active Program line buttons ..... 167
Graphic Window ..... 167
Status / Execution Status / Connection / GoPro / Windows ..... 168

Status ..... 168
Execution Status ..... 169
Connection ..... 169
GoPro ..... 171
Properties Window ..... 173
Metrology / Collision Engine Buttons ..... 175
Special interactions ..... 175
RA Main Menu Button ..... 176
Settings ..... 178
General ..... 178
Collision ..... 179
Dragging ..... 179
Region ..... 179
Graphics ..... 179
Docking ..... 180
Tracker ..... 183
Robot Settings ..... 184
Cloud of Points settings ..... 184
Simulation Settings ..... 184
Planning ..... 184
Colors ..... 185
Sounds ..... 185
Import ..... 186
Top-right Menu ..... 186

Keyboard Shortcuts ..... 186
HxGN Execution Manager ..... 189
HxGN Robotic Automation App Manager ..... 193
194
Right Click menu and Basic Tab ..... 194
**Modelling** ..... **198**
Machine Creation ..... 198
Slider ..... 199
Slider creation wizard ..... 204
Tracker on slider ..... 208
Robot ..... 209
Robot creation wizard ..... 217
Robot on slider ..... 217
Align Robot and Slider orientation and Position. ..... 220
List of supported robot models ..... 223
Robot tools ..... 226
Rotary Table ..... 227
Rotary table creation wizard ..... 231
Exporting an object to the library ..... 232
Changing the machine properties ..... 233
Configuring machine saved positions ..... 240
Machine testing ..... 241
D-H Parameters ..... 242
Cell Modelling ..... 245

Adding items to the cell ..... 246
Importing items from the library ..... 246
Importing items as CAD models ..... 247
Analysis of CAD sparse parts ..... 249
Loading and attaching a Tool to the Robot ..... 249
Adding a Sensor to the cell ..... 251
Adding a Fixture to the cell ..... 252
Adding a Part to the cell ..... 254
Adding Targets to the cell ..... 255
Defining Collision rules among cell items ..... 256
Arranging items into the cell ..... 259
Alignment Tool ..... 259
Snapping frames ..... 264
Grouping and Merging cell items ..... 269
Items selection ..... 269
Grouping items ..... 270
Merging items ..... 272
**Alignment ..... 274**
Alignment - Leica ..... 274
Prerequisites ..... 274
Manual alignment workflow ..... 276
Rotary Table alignment using three planes ..... 276
Part alignment using features ..... 285
Alignment wizard workflow ..... 291

Cell alignment wizard ..... 293
Rotary Table used as absolute workspace reference ..... 301
Part alignment wizard ..... 303
Best fit alignment wizard ..... 304
Add reflectors wizard ..... 307
Dual tracker cell alignment workflow ..... 315
Robot on slider and tracker on slider workflow ..... 316
Robot alignment ..... 317
Prequisites: ..... 318
Alignment - ZG ..... 330
Manual Tracker And Sensor Calibration ..... 330
Calibration Precautions ..... 330
Manual Calibration Procedure ..... 331
Tracker Calibration ..... 331
HyperScan Scanner Calibration ..... 337
Tracker And Sensor Calibration using HxGN Robotic Automation ..... 340
Tracker Calibration ..... 340
Scanner Calibration ..... 342
**Programming ..... 345**
Program versus subroutine ..... 345
Clearance settings ..... 346
Language syntax ..... 349
Supported commands ..... 350
Grouping ..... 352

Dialog ..... 353
Safety ..... 354
Motion ..... 355
Robot ..... 358
Rotary Table ..... 358
Slider ..... 358
Continuous Motion group ..... 359
PLC ..... 359
Acquire/release Lock for Multi Robot Synchronization ..... 361
Metrology ..... 361
Programming tools ..... 361
Generic command menu ..... 362
Automatic Program Adjustment ..... 364
Migration of old Programs ..... 366
Added notes ..... 366
Manual Programming Mode ..... 367
Programming mode special toolbar commands ..... 367
Controlling a cell item using the teach pendant ..... 369
Robot teach pendant ..... 370
Rotary Table/slider teach pendant ..... 371
Tracker teach pendant for Leica only ..... 372
Moving a cell item dragging it in the 3D environment ..... 379
Pick on CAD ..... 382
Adding commands ..... 386

Programming - Leica ..... 391
Supported Commands ..... 391
Compensation (HxGN Robotic Automation Compensation) ..... 391
Compensation (PolyWorks Reference points) ..... 392
Sensors ..... 392
Start Measurement - AT960 ..... 393
Stop Measurement - AT960 ..... 393
Other Tracker Commands ..... 394
Change AS1 Settings ..... 396
Programming ..... 397
Compensation ..... 397
Type of reflector measurement ..... 399
Measure Nominal Reflectors workflow ..... 400
Measure Actual Reflectors workflow ..... 401
Evaluating the Quality of the Compensation ..... 403
Measure Nominal requirements ..... 404
Understanding alignment data ..... 405
Customizing RMS tolerance ..... 408
Measure Actual requirements ..... 409
Measure Reference points workflow ..... 409
Reading Sensor data ..... 412
Programmatic checks and alignments ..... 416
Sphere check ..... 417
Workflow ..... 417

Recommended robot motions for sphere measurements ..... 421
Execution ..... 422
Three plane distance check ..... 423
Workflow ..... 423
Recommended robot motions ..... 425
Execution ..... 426
Sphere Alignment ..... 427
Workflow ..... 428
Recommended robot motions for range measurements ..... 431
Execution ..... 433
Programming - ZG ..... 435
Program properties ..... 435
Scan command ..... 439
Tracker Visibility on the Scanner ..... 440
Collect markers ..... 441
Wait for the Point cloud to be fully generated ..... 443
Transfer Station ..... 444
Dynamic Transfer Station (Real Time Tracking) ..... 447
Static Transfer Station ..... 447
Example Programs ..... 448
No Transfer Station example: ..... 448
Collect Makers (required for Static or Dynamic Transfer Station) ..... 448
Example Static Transfer Station Program: ..... 449
Example Dynamic Transfer Station Program: ..... 450

Path Planning ..... 451
Automatic mode programming – Feature path planning ..... 451
Feature measurement settings ..... 458
Save fixed positions ..... 463
Path planning algorithm flowchart ..... 464
Evaluate the quality of the scan ..... 465
Automatic mode programming – Featureless path planning ..... 468
Trajectory scanning ..... 468
Patch scanning ..... 473
Other Path Planning modules ..... 477
Mirroring a program ..... 477
Creating the mirroring plane ..... 478
Mirroring a program ..... 478
Reversing a program ..... 479
Validation and Simulation ..... 481
Direct Command Validation ..... 481
Simulation Errors and Warnings ..... 483
Simulation Error List ..... 483
Program Simulation Accuracy ..... 485
Build Program Simulation ..... 485
Simulation Warnings and Errors ..... 488
Warnings ..... 488
Errors ..... 489
Simulation Error Feedback ..... 489

Simulation Player UI ..... 490
Player skipping frames ..... 492
Hiding the Point Cloud from 3D window ..... 492
Sending data to Metrology ..... 493
Notable use cases ..... 494
Single robot cell with Rotary Table and stationary tracker ..... 494
Single robot cell with stationary part holder and movable tracker ..... 495
Single robot cell with stationary part holder and stationary tracker ..... 498
Single robot cell with Rotary Table and movable tracker ..... 499
Dual robot cell with stationary part holder and dual movable trackers - Leica only ..... 501
Dual Tracker Compensation ..... 502
Parallel programming ..... 503
Tracker crossing ..... 504
Concurrent resource ..... 506
Complete programming workflow ..... 510
**Execution ..... 513**
Special case: Executing a safety motion in programming tab when connected to a real cell ..... 513
Execution from HxGN Robotic Automation ..... 514
Special case: Splitting a cell into 2 individual workstations ..... 521
Robot Program Export ..... 524
Configuring the application to show the export button. ..... 524
Exporting a program ..... 524

Execution from Execution Manager (EM) ..... 525
Part properties ..... 527
SQLite ..... 529
General User workflow ..... 529
Program page ..... 530
Editing a program ..... 530
Removing a program ..... 531
Adding a program directly to the dashboard ..... 533
Playlist page ..... 533
Creating a playlist ..... 533
Editing a playlist ..... 535
Flags page ..... 536
Adding flags ..... 537
Workspace (Dashboard) page ..... 538
Select the Workstation ..... 543
Set the Execution mode ..... 544
Executing a playlist in automatic and semi automatic mode ..... 544
Executing a program or a playlist in manual mode ..... 545
Logs ..... 546
Cell status ..... 547
**Metrology Connection ..... 549**
Adding Metrology to the Metrology drop down ..... 549
Using Workstations and Metrology results ..... 549
PC-DMIS and Inspire ..... 549

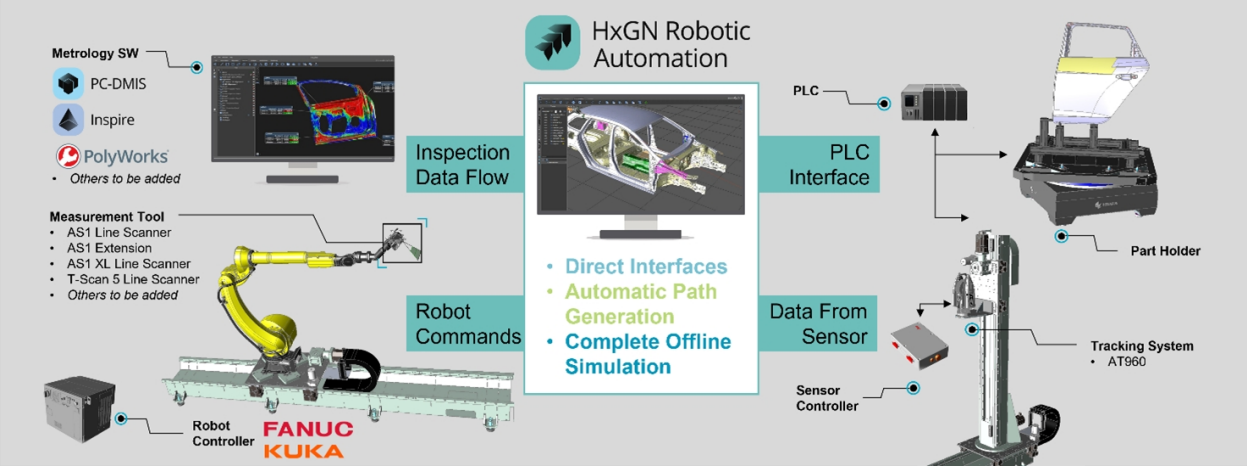
PolyWorks ..... 551
Import features by establish the connection with a metrology ..... 551
Inspire ..... 553
PC-DMIS ..... 561
PC-DMIS Static point measurement ..... 569
PolyWorks ..... 577
Programming PolyWorks Macros ..... 581
Named COP ..... 586
Scan Settings ..... 590
Setting metadata ..... 593
Reference points ..... 596
List of supported features ..... 599
Feature Generation Mode ..... 605
**Troubleshooting ..... 610**
**Frequently asked questions ..... 611**
**Known Issues ..... 616**

# Introduction

HxGN Robotic Automation is a software application, delivered as part of a robot-based inspection cell, which is an integrated system composed of multiple hardware and software components connected and coordinated to perform inspection tasks.

HxGN Robotic Automation allows the user to create an integrated solution for part inspection, combined with a metrology SW application (Inspire, PC-DMIS or PolyWorks) and connected to automation hardware devices (robots, PLCs, sensors, trackers- either static or moving on a slider, part holders- either static or movable as a rotary table).

The image below shows the different components that are part of a robot-based inspection cell, all interconnected to create a robotic solution for part inspection.



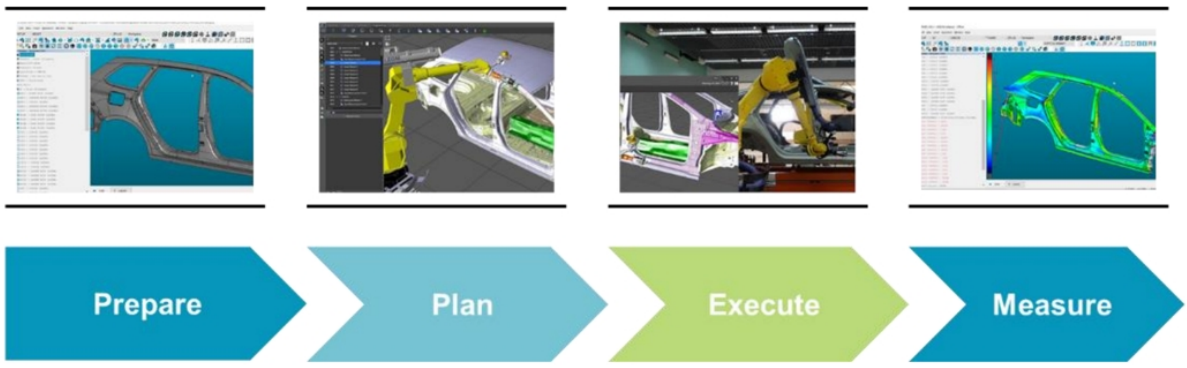

HxGN Robotic Automation supplies the toolset to program and control robotic cells designed for metrology and part inspection. It simplifies the interface with the cell components and the robot programming perfecting the data acquisition process needed to achieve smart manufacturing.

Besides its ease of use and interoperability, which makes it a universal robot programming platform, one of its key differentiators is its planning capabilities, with a path planning engine that is tailored to support the main measurements devices for part inspection and their measurement strategies needed for best data acquisition.

The manual is divided in different topics that can be accessed in the navigation index or by search.

The image below shows the complete user workflow to execute a robotic inspection task, which can divide into four main steps:

| **Prepare**: | **Plan**: | **Execute**: | **Measure**: |
|---|---|---|---|
| Preparation of the measurement task based on the inspection plan in the metrology application | Automatic generation of the robot program for part inspection in HxGN Robotic Automation | Online robot program execution for measurement data acquisition using Execution Manager | Measurement data analysis in the metrology application based on data received during the program execution |




HxGN Robotic Automation closes the gap between automated inspection data acquisition for quality assurance and process optimization, thanks to the complete integration with other Hexagon Manufacturing Intelligence portfolio.

As part of the digital transformation, all this data (metrology data) can be used to populate and enrich a QMS (Quality Management System) which makes the best

use of the quality and production data to improve the PLM (Product Lifecycle Management) process from start to end.

# About the Manual

This manual is tailored towards regular HxGN Robotic Automation users and service users looking to setup a complete cell.

Some sections will be tailored specifically towards installing HxGN Robotic Automation in a real cell and will require advanced knowledge of the physical cell and setup. These sections will be marked.

## What's new this release

### ZG Integration

- Using markers to get accurate measurements
- Static, Dynamic and No - transfer station to manage relationship between the part and the tracker
- Sensor and Tracker calibration through HxGN Robotic Automation
- See: HyperScan installation.htm, Alignment - ZG.htm, Programming - ZG.htm

### ABB Robot Integration

- See: ABB robot

## A new Metrology API

- Any Metrology can now integrate with HxGN Robotic Automation!
- See: (Documentation coming at a later date)

## CLMS Integration

- Supporting the Chinese licensing software for HxGN Robotic Automation
- See: CLMS Licensing

## Splitting a PRESTO XL cell into individual and simultaneously working PRESTO L cells

- See: Splitting a cell into 2 smaller individual cells

## Mirroring and Reversing a program

- Allow a program to be mirrored over a mirroring plane towards any combination of sensors and robots
- See: Mirroring a program, Reversing a program

## Other Changes:

- Application Manager for easier JSON handling (JSON Configuration through Application Manager)
- AS1XL scanning distance implementation (AS1XL properties example)
- Removal of SQL Express (removal of installation need) (Configuration and Licensing.htm)
- Timeout implementation of Dialog Command (Dialog command)

- Addition of subroutines (Program vs subroutine)
- Getting environment variables from RDS (Reading sensor data)

# Getting things ready!

In this section will detail the software installation and it's requirements, from a single offline user to a whole robotic cell installation

# Software and Hardware Requirements

## System Hardware Requirements

| Category | Recommended Version | Minimum Version |
|---|---|---|
| Processor | Recommended Intel Core i9 9th Generation or higher (for example, i9-9xxxHx) 8 Cores hyper-threading enabled | At min: 2 GHz or higher quad core processor |
| RAM | 64 GB | 64 GB |
| Hard Drive | 2 TB HDD with at least 5 GB free | 2 TB SDD with at least 5 GB free |
| Graphics | NVIDIA RTX series graphics card equipped with 20 GB of memory | NVIDIA RTX series graphics card equipped with 20 GB of memory |
| Connectivity | Three Ethernet ports may be needed for specific online installations. | One Gigabit Ethernet port. |

HxGN Robotic Automation3D visualization component requires fully compliant OpenGL or Direct3D runtime support.

Virtualized environments not allowing direct access to the GPU will not work.

## Software Requirements

| Software | Comments | Minimum Version |
|---|---|---|
| Operating System | Windows 11 will also work | Windows 10 64-bit |
| Browser | Compatible with Windows Edge or Chrome | |
| Precondition Installers | This software is required to be installed before installing HxGN Robotic Automation | otp_win64_23.3 and rabbitmq-server-3.8.11 |

HxGN Robotic Automationdoes not require elevated access rights, however some connected applications require so.

PC-DMIS: Adminstrative rights

HyperScan Software: Read, write and USB-write. Make certain the following folder has "Full Control" access: "<C:\Program Files\ZG\HyperScan <VersionNr> \SaveData>.

Hexagon uses the **Sophos** anti-virus tool to test HxGN Robotic Automation. The user will need to confirm the performance of any other anti-virus tool. http://sophos.com/products/enterprise/endpoint/security-and-control/

## Minimum Metrology Software Versions

| Software | Comments | Minimum Version |
|---|---|---|
| PC-DMIS | | 2025.2 or later<br>2024.1 SP2 if used with Execution Manager |
| Inspire | | 2025.1 or later |
| PolyWorks | | 2025 IR5.1 or later |

## Leica Software

### Leica AS1

| Software | Comments | Minimum Version |
|---|---|---|
| RDS | | 6.4 (1) |
| Leica AP21 | | AP21 Software 1.2.0.x.Update |
| Leica CB21 | | CB21 Software 1.1.0.x.Update |
| AS1 Scanner | | Firmware 1.4.0.X.fwx |

There is a new feature in RDS 6.5 that can cause an issue: In case the lock gets lost and the user tries send a new RDS command (before an internal RDS timeout has been reached, this error message can occur: Requirements not met.(1)

## Leica T-Scan

| Software | Comments | Minimum Version |
|---|---|---|
| RDS | | 6.4 (2) |
| T-Scan Collect | | 10.3.8.X |
| T-Scan Interface | | 10.3.8.X |
| Twin CAT | | TC31-ADS-Setup 3.1.4024.Xx |

There is a new feature in RDS 6.5 that can cause an issue: In case the lock gets lost and the user tries send a new RDS command (before an internal RDS timeout has been reached, this error message can occur: Requirements not met. (2)

### Leica Tracker

| Software | Comments | Minimum Version |
|---|---|---|
| Leica Tracker | | AT9x0 Software 2.3.1.X.Update |

## ZG software

| Software | Comments | Minimum Version |
|---|---|---|
| Camera driver: Spinnaker | | Spinnaker 2.4.0.110 |
| Scanning Program : HyperScan | | HyperScan 4.4.0.209 |

## Robot Software

| Software | Comments | Minimum Version |
|---|---|---|
| FANUC | | FANUC OS from 7.70 to 9.40 |
| KUKA | | KUKA.SystemSoftware from 8.2 to 8.4 |
| ABB | | RobotStudio 2025 |

# Software Installation

Configuration files used in HxGN Robotic Automation are saved under:

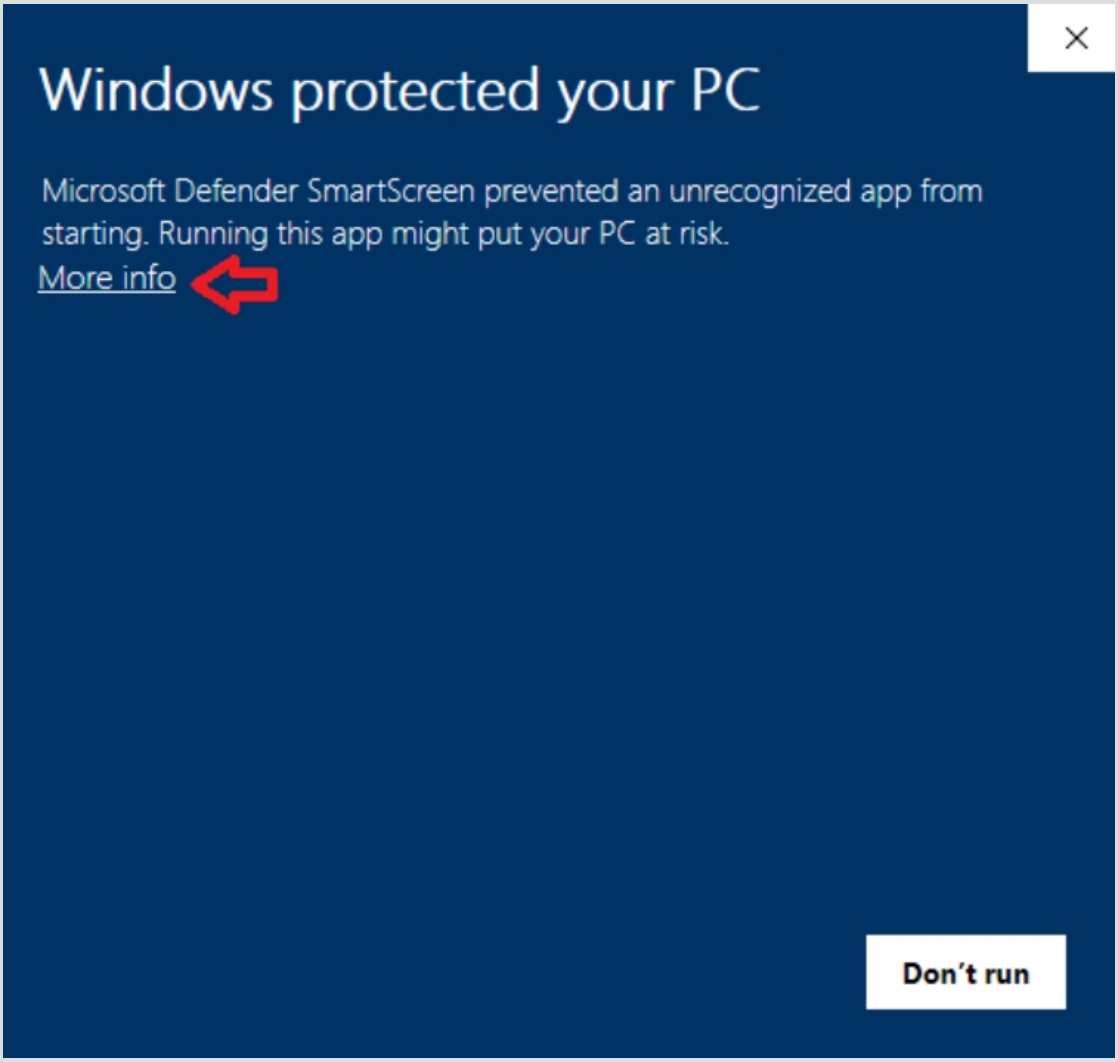
- “%appdata%\Roaming\Hexagon\HxGN.RA.GenericHost”
- “%appdata%\Roaming\Hexagon\RA”

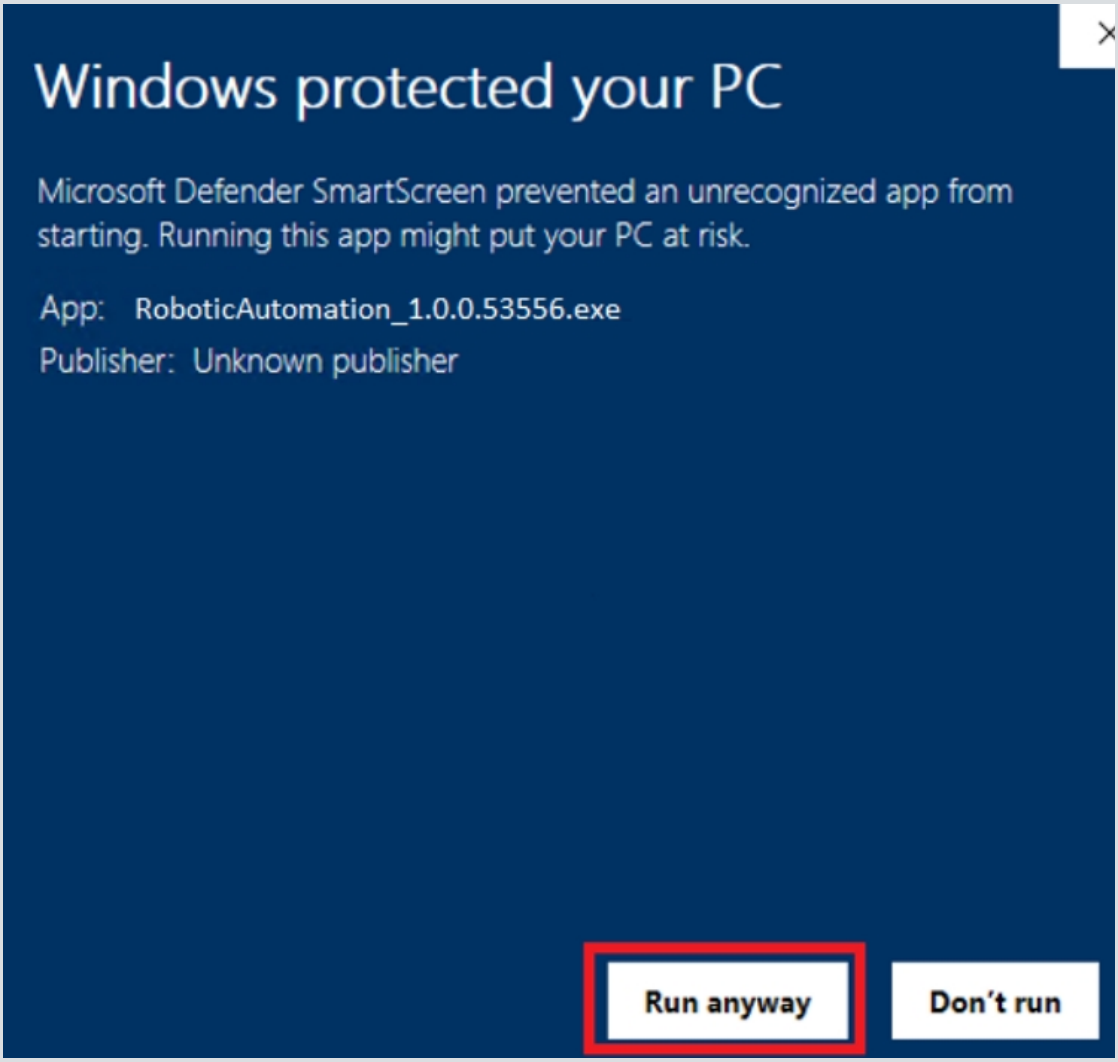
## HxGN Robotic Automation Installation

To use HxGN Robotic Automation for offline programming it is needed to install the Robotic Automation main application by following these steps.

Right click on HxGN Robotic Automation installer and click run as administrator.

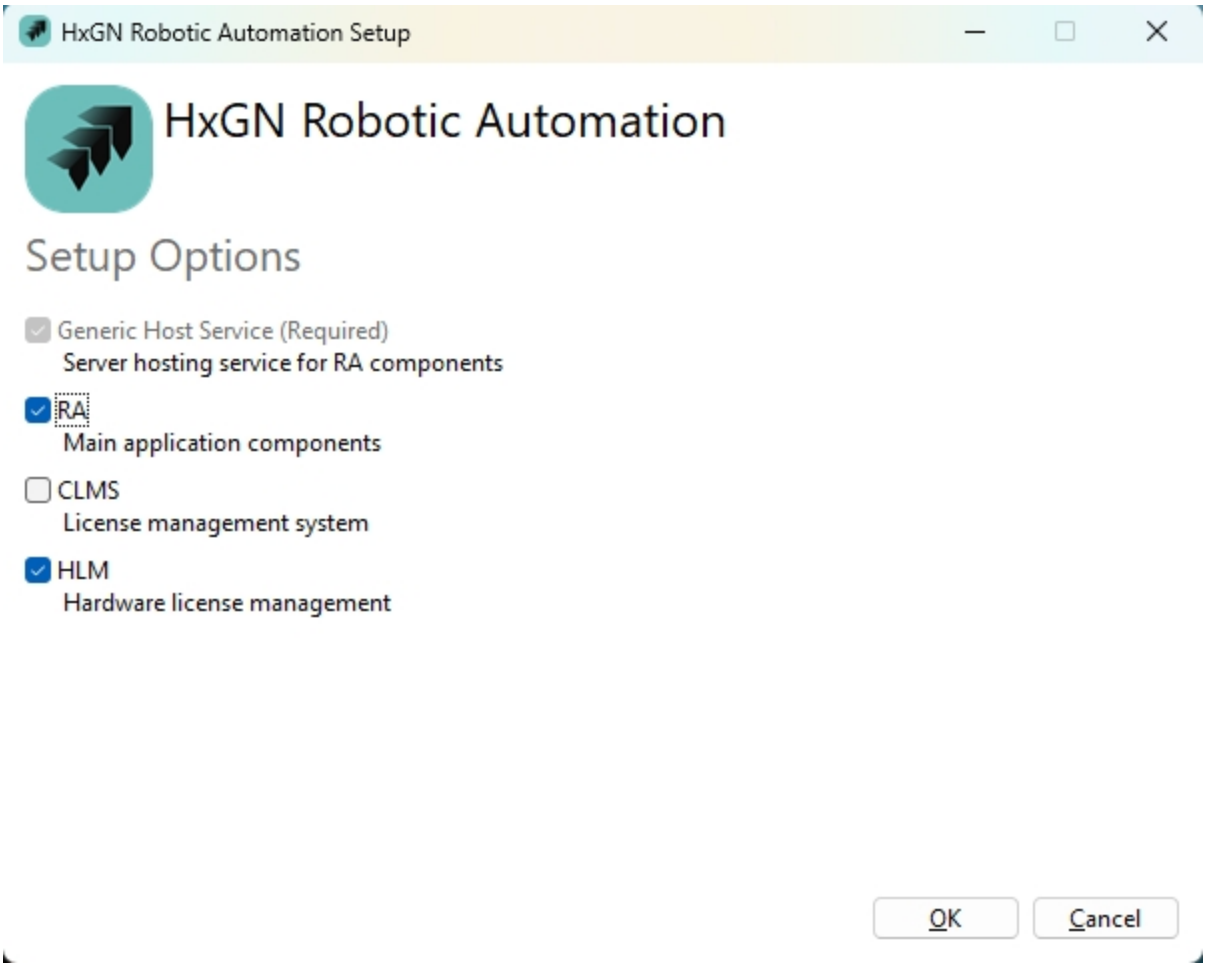
In case you get a warning message from Window as follows, please click on **More Info**, and then click **Run anyway** button.


Windows protected your PC
Microsoft Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.
More info
Don't run



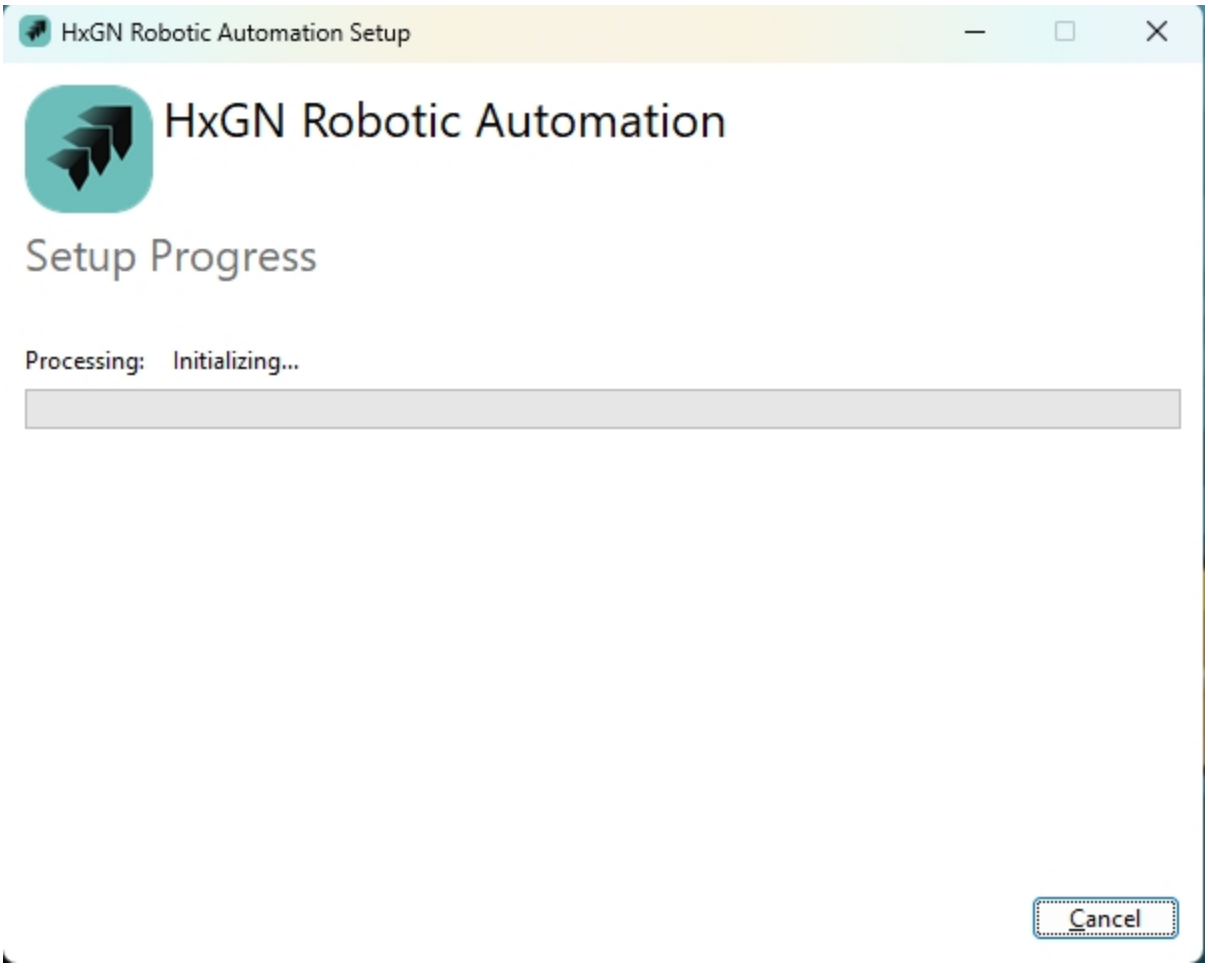

When the installer dialog box pops up, check the checkbox to agree on the license terms and condition and then click on **Options**.

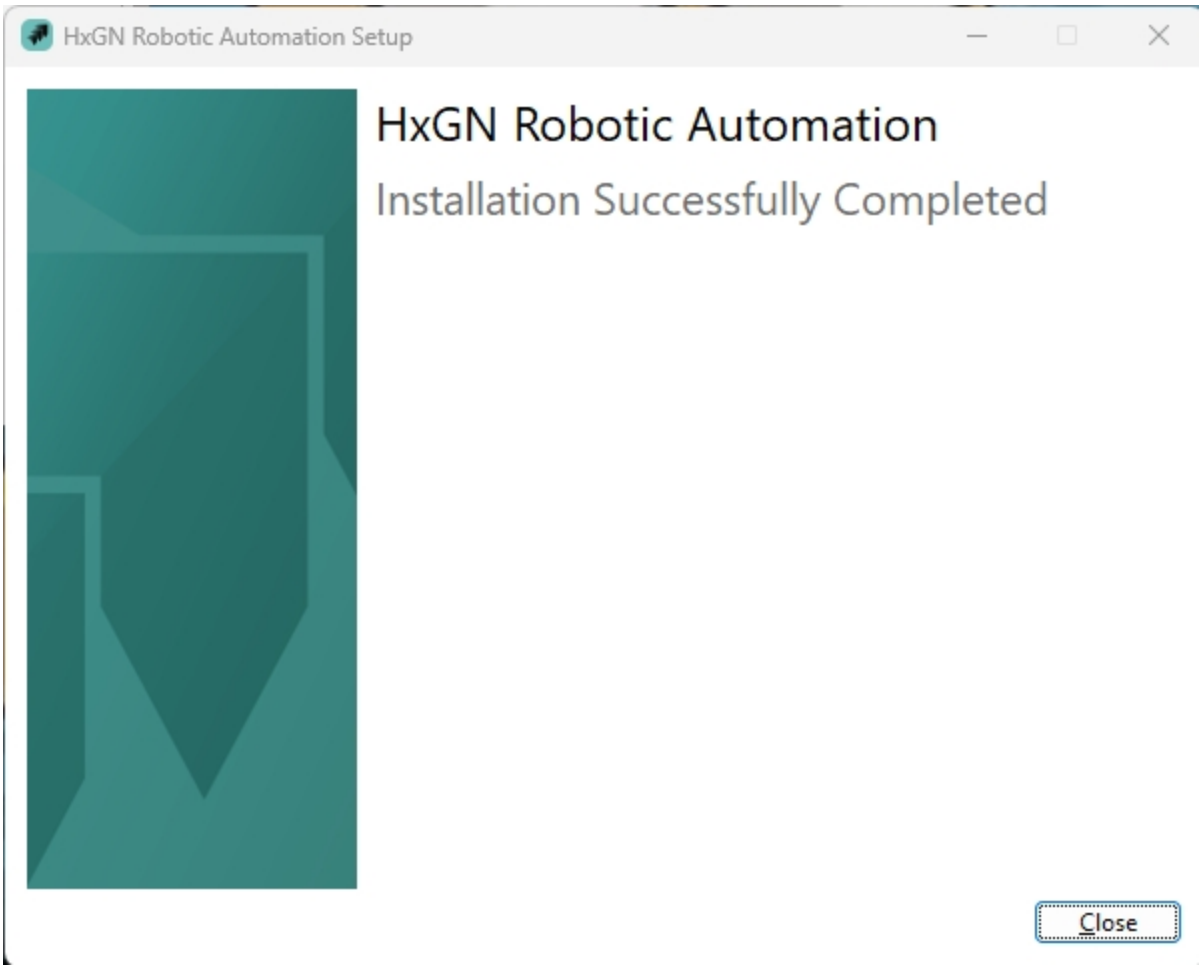
The possibility to untick the Robotic Automation checkbox is only in case of Dual robot cell when it's needed to install only Generic Host in the second computer. Alternatively you can select an alternative license system such as CLMS for the Chinese market. Keep the default settings and click on **Ok**.



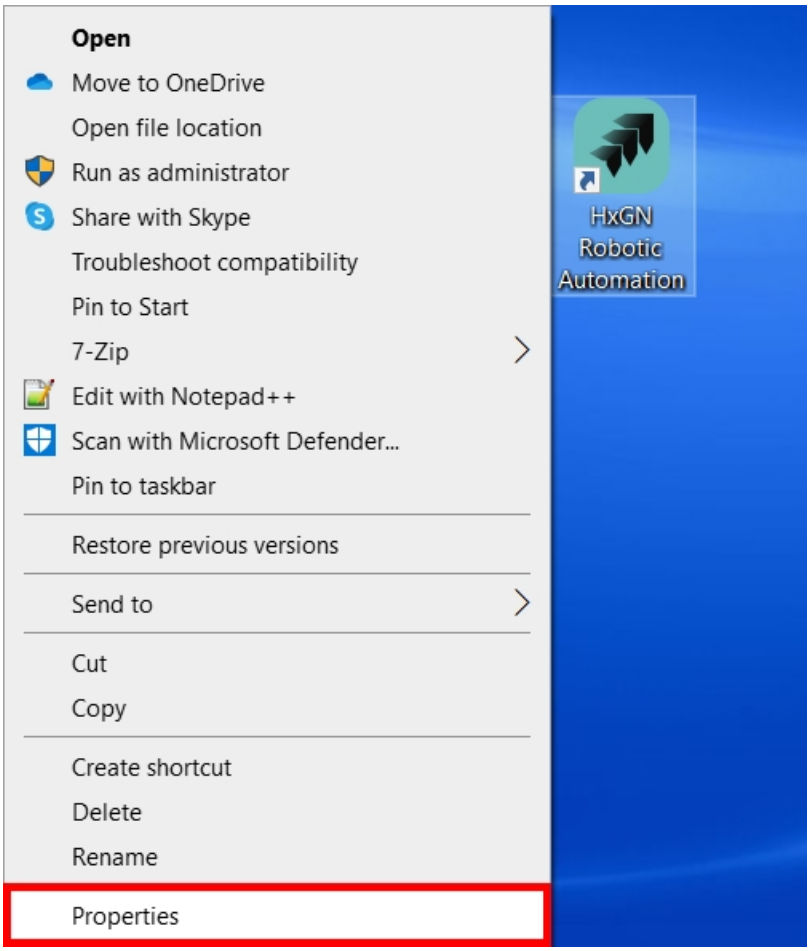

Click **Install**.

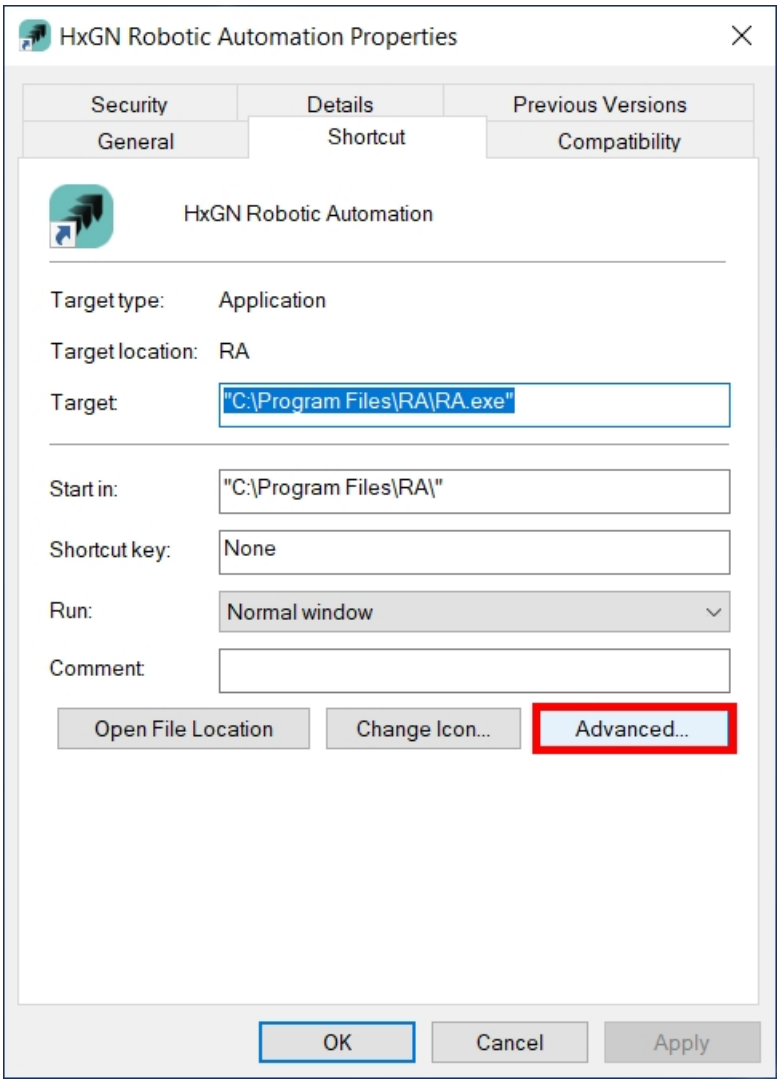
After the installation is completed the dialog box can be closed and HxGN Robotic Automation shortcut icon will be created on your Desktop.

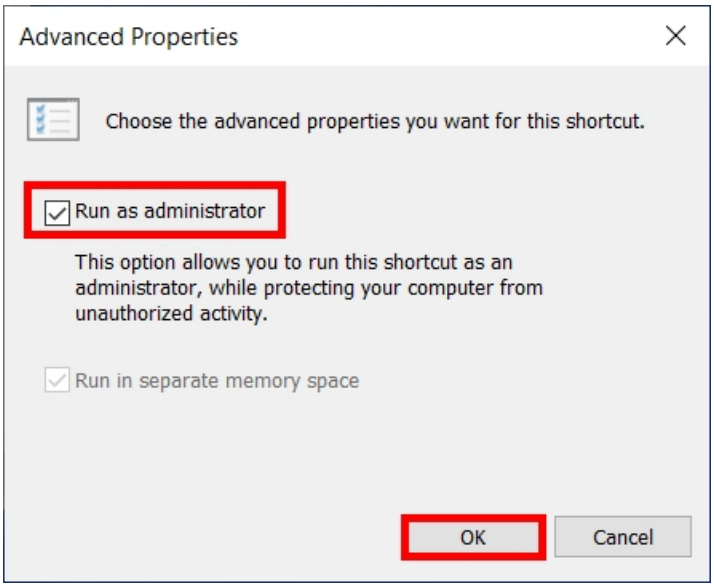

HxGN Robotic Automation Setup
HxGN Robotic Automation
Setup Progress
Processing: Initializing...
Cancel




It is always recommended to execute HxGN Robotic Automation as administrator. Right-click on the desktop shortcut and select **Properties**. Check the checkbox in the shortcut advanced properties as shown in the images below.


Open
Move to OneDrive
Open file location
Run as administrator
Share with Skype
Troubleshoot compatibility
Pin to Start
7-Zip
Edit with Notepad++
Scan with Microsoft Defender...
Pin to taskbar
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
HxGN
Robotic
Automation


HxGN Robotic Automation Properties
Security
Details
Previous Versions
General
Shortcut
Compatibility
HxGN Robotic Automation
Target type:
Application
Target location:
RA
Target:
"C:\Program Files\RA\RA.exe"
Start in:
"C:\Program Files\RA\"
Shortcut key:
None
Run:
Normal window
Comment:
Open File Location
Change Icon...
Advanced...
OK
Cancel
Apply




## Services and Ports

When HxGN Robotic Automation application is installed on a real cell and connected to real HW devices (Robot, PLC, Sensor, Tracker, etc.) for online programming and execution, the following steps must be executed.

> ⚠ **Warning:** IT department must be asked to set security policies to allow user admin privileges as Windows Firewall may block connections to RabbitMQ in the cell PC.

- Make certain these **service are not blocked**:
  - HxGN.RA.ServerGenericHost
  - RAbbitMQ
- Make certain these ports are **not occupied**:

- Generic Host and Execution manager services use by default port **80**.
- Robot Ports are by default **6008** or **6009** in case of more than one robot installed.
- Ports for Leica devices need to be configured during installation the System Integrator.
- RabbitMQ (used for communication with PC-DMIS/Inspire uses by default ports **5672** and **15672**.

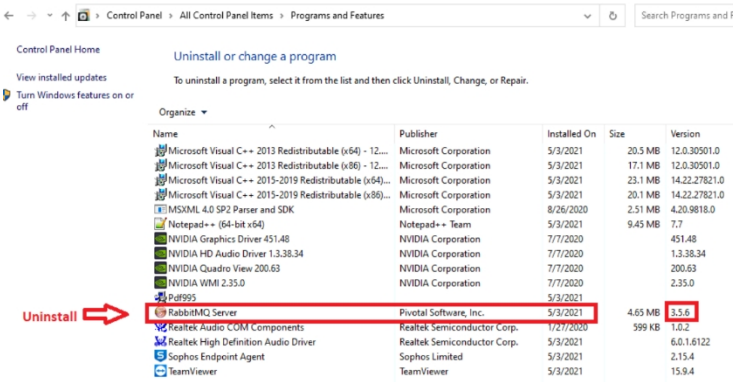
## Installation order Erlang and RabbitMQ

**Make sure to install the following versions in the order bellow:**

1. erlang: otp_win64_23.3.exe (1)
2. RabbitMQ – version rabbitmq-server-3.8.11.exe (1)

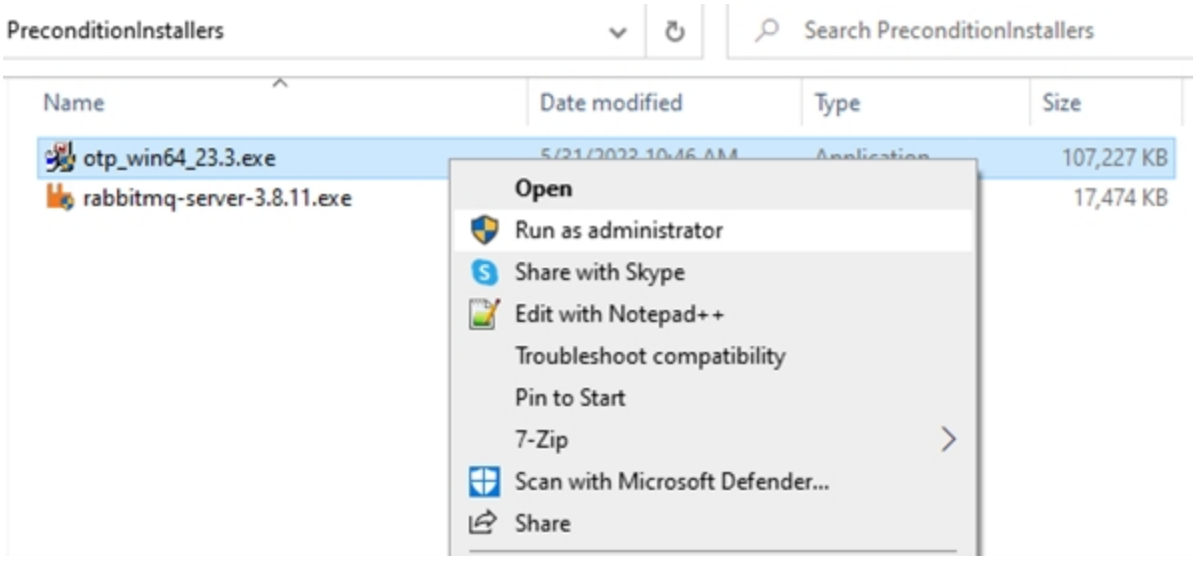
Needed only if Inspire or PC-DMIS metrology applications are used.(1)

Verify that previous versions of RabbitMQ Server and erlang are removed, you may uninstall it from **Control panel | Programs and Features:**

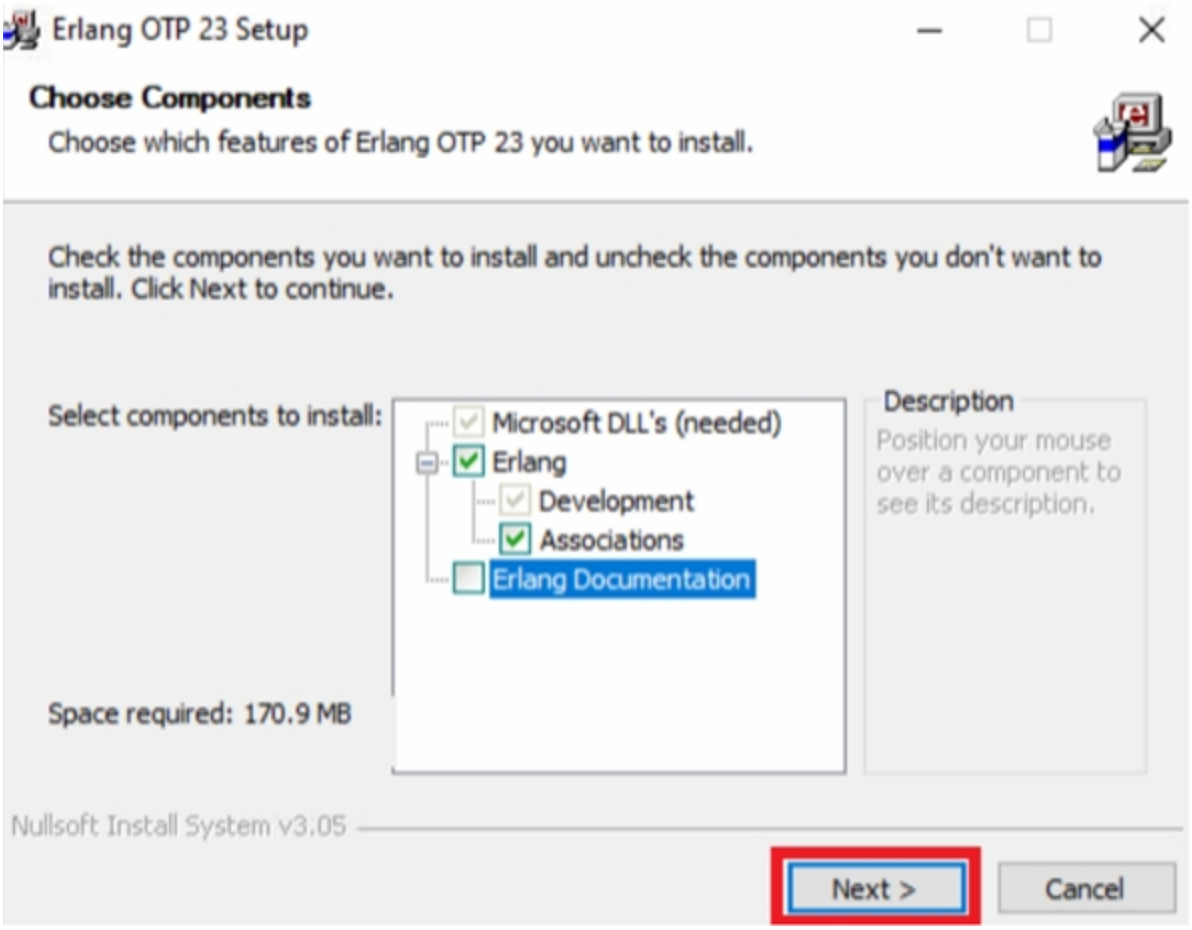



## Erlang Installation

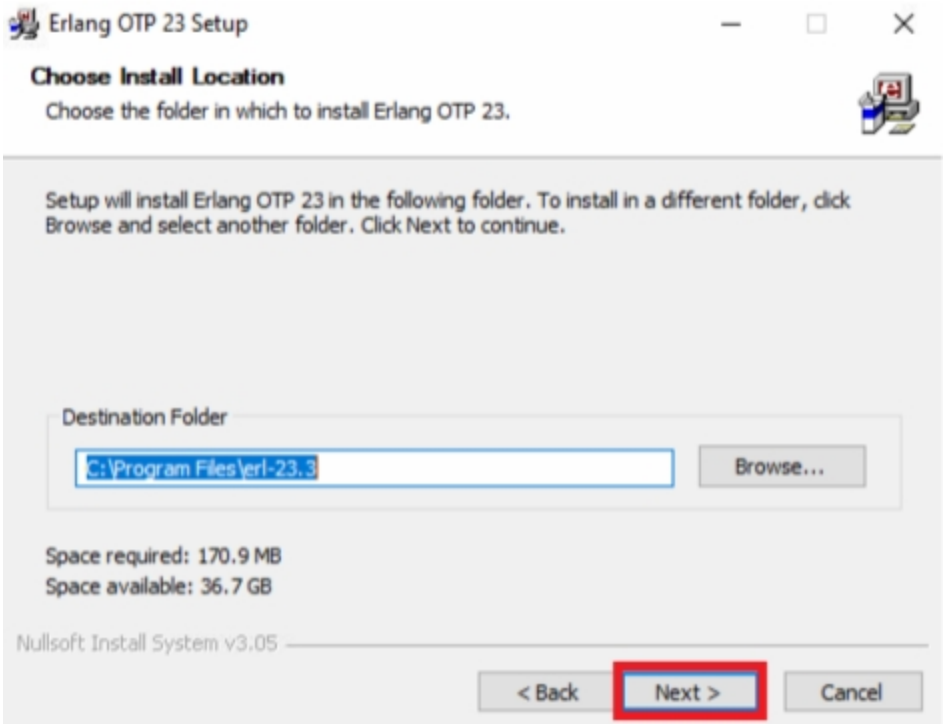
Right-click on **otp_wind64_23.3.exe** and run it as administrator.



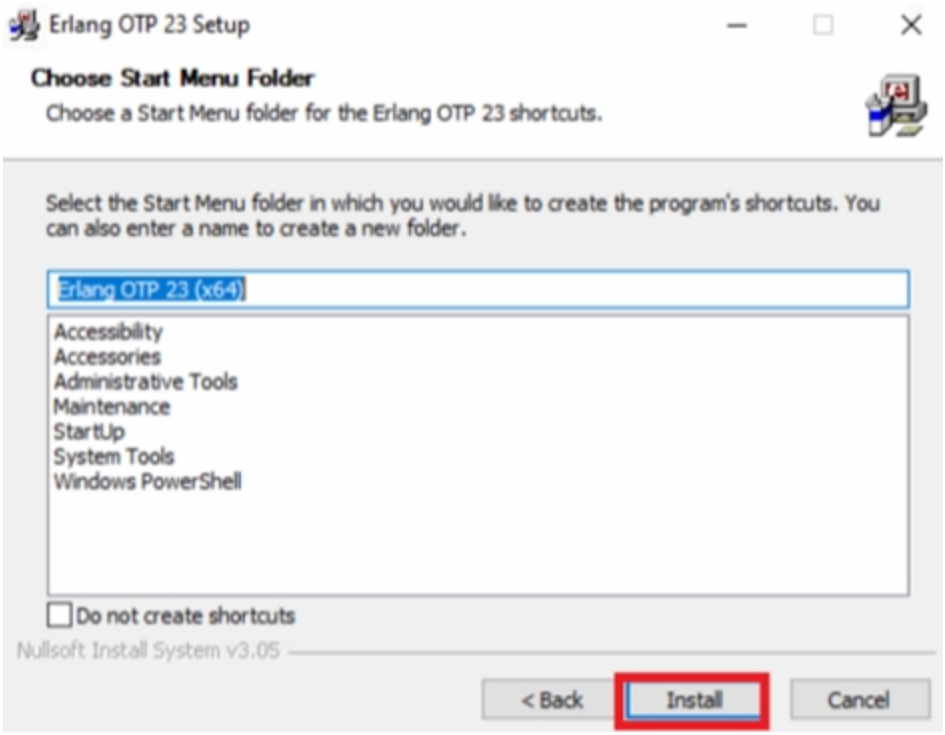

You may uncheck **Erlang Documentation** and click **Next**:



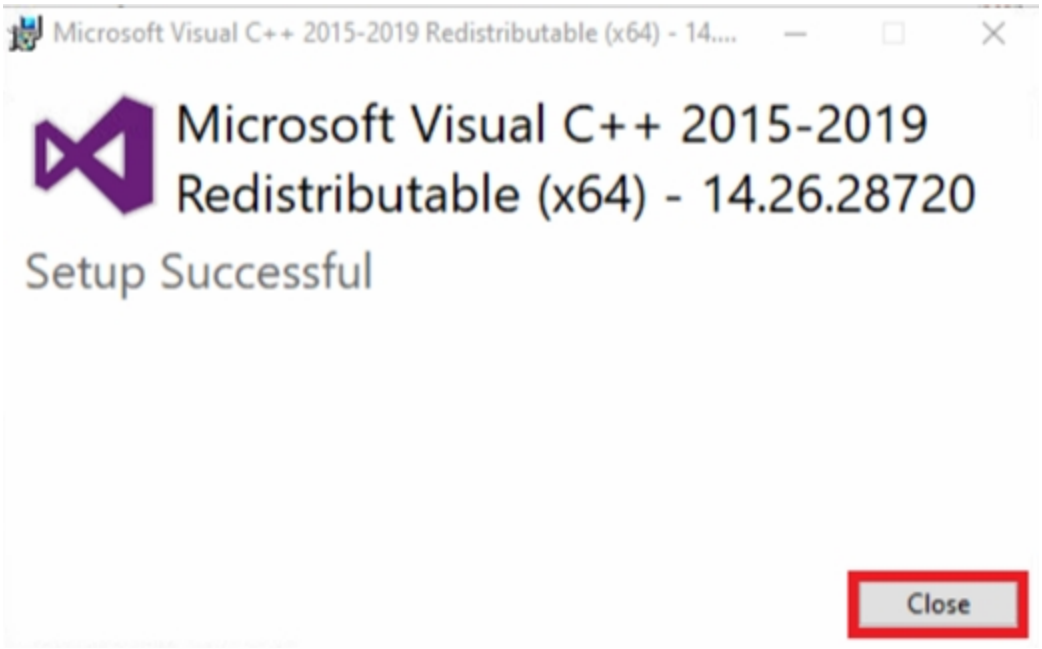

Leave install location as default **C:\Program Files\erl-23.3** and click **Next**:

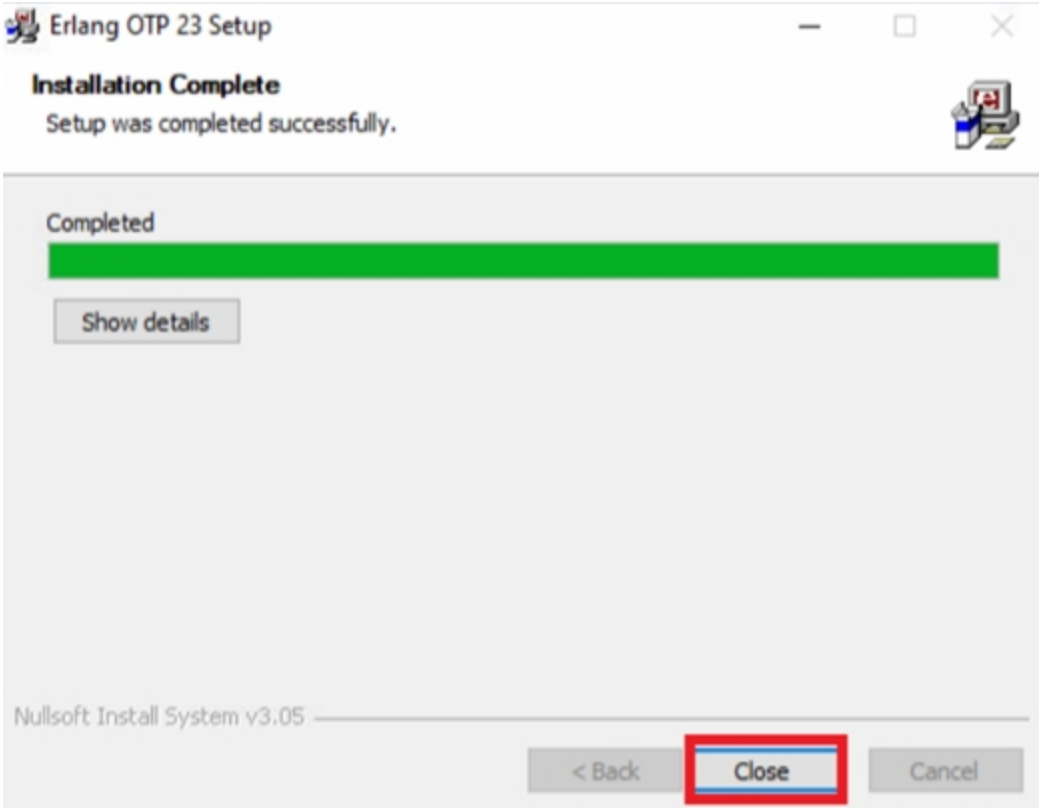



Click **Install**:




Check **I agree to the license terms and conditions** and click **Close** once setup is successful:

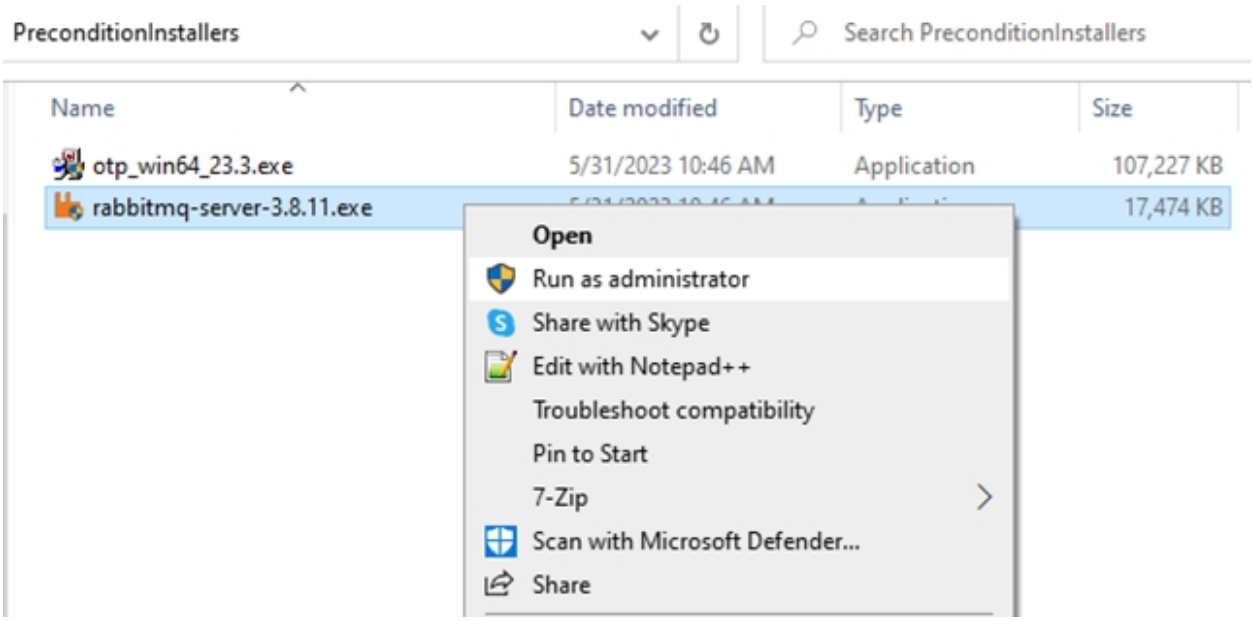




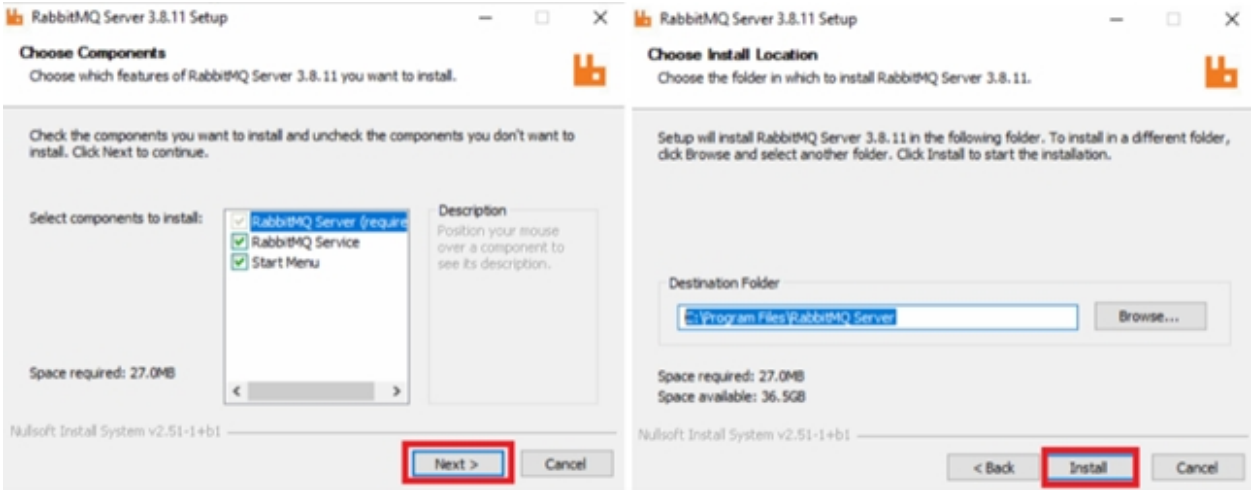

## RabbitMQ Installation

> ⚠ **Warning:** if the computer name is changed, it may be needed to uninstall and install again RabbitMQ as this may cause the COP not to be sent to metrology applications. See section Troubleshooting for errors in RabbitMQ connection.

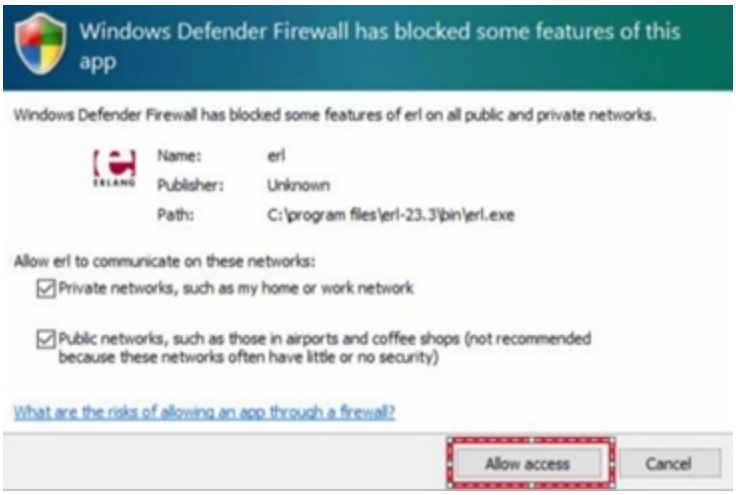
Right-click **rabbitmq-server-3.8.11.exe** and click **Run as administrator**:



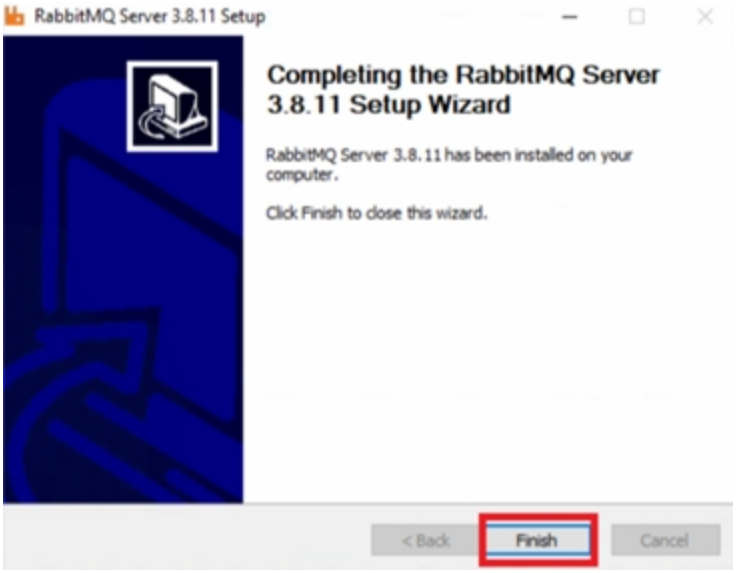

Select **Next**, and leave the destination folder as **C:\Program Files\RabbitMQ Server** and then click **Install**:



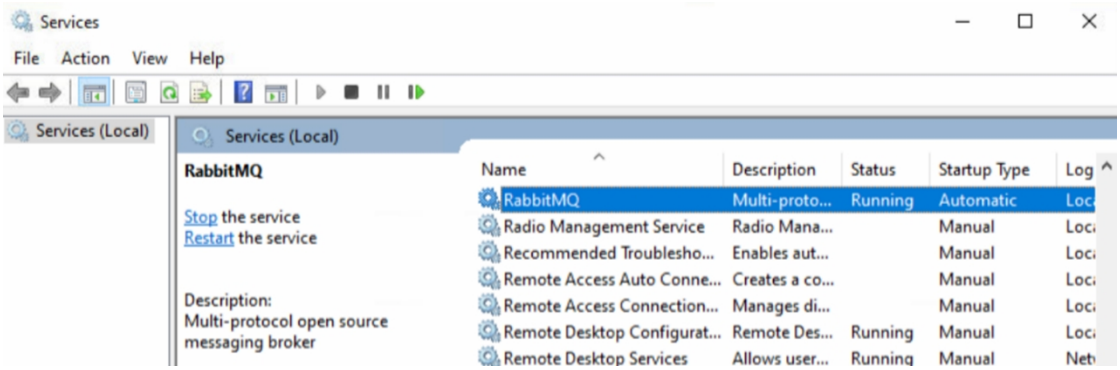

Click on **Allow access** to allow RabbitMQ to access the configured networks:




Click **Finish** to close the wizard:




Verify on Services that **RabbitMQ** is in Running status:

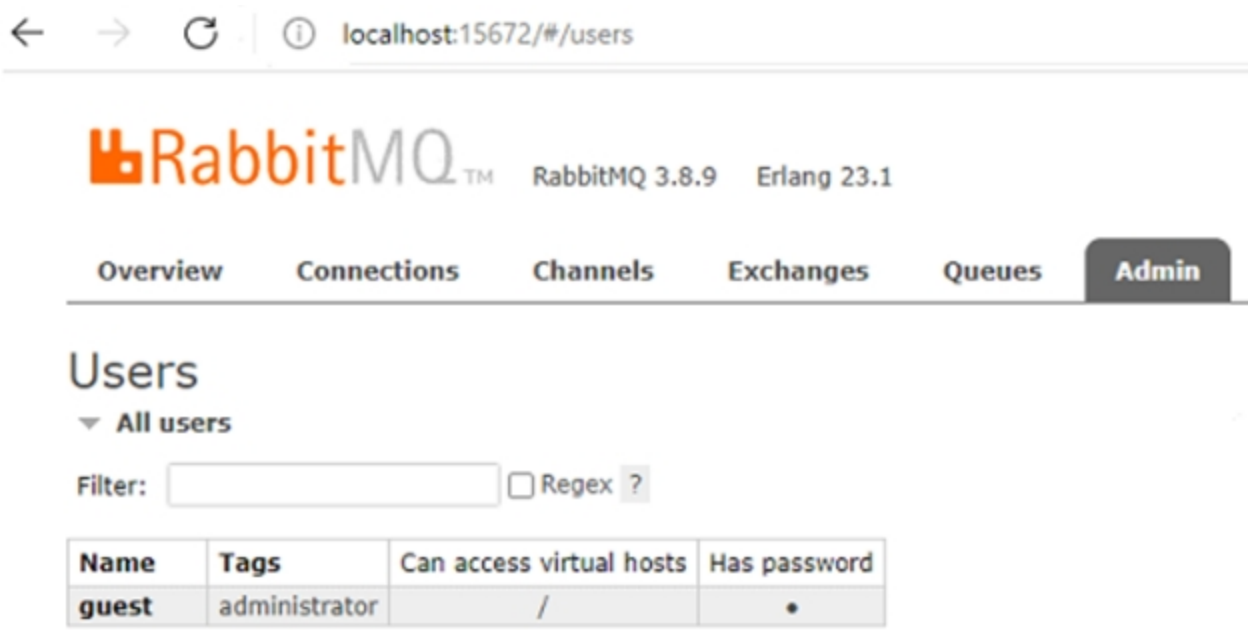



**Troubleshooting RabbitMQ**

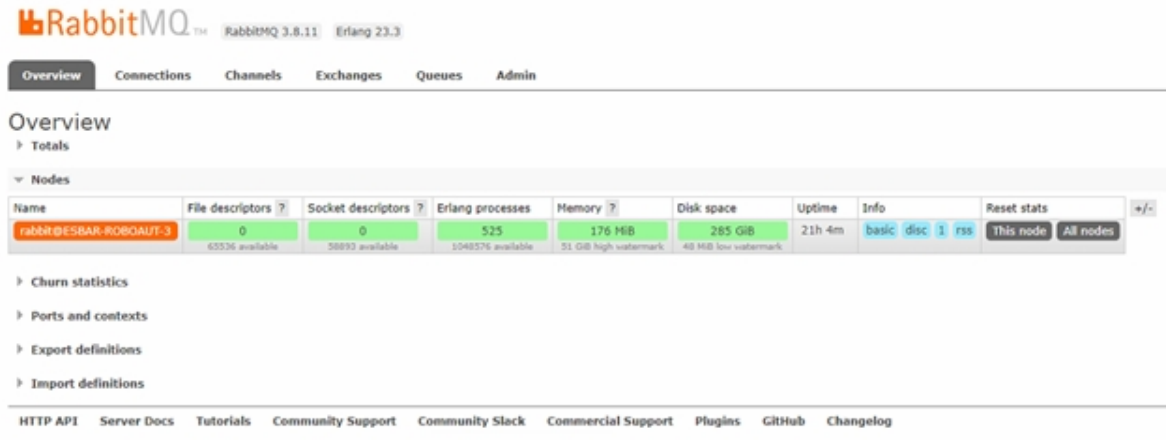
To check if it is possible to connect to RabbitMQ follow these steps:

1. enable the User Management plugin:
    1. Open RabbitMQ command prompt (Windows menu | RabbitMQ Server)
    2. Run next command “rabbitmq-plugins enable rabbitmq_management.”
    3. Restart the RabbitMQ service.
    4. Go to browser and navigate to the following URL: http://localhost:15672
    5. You have now access to the management with user “guest” and password “guest.”

       Under **Admin** | **Users** you can see all users with “administrator” tag:




2. Within **Overview** tab check if your computer name is listed and its status, i.e., RabbitMQ is receiving information.




3. Check that the Windows Firewall does not block connections to RabbitMQ. To configure the Windows Firewall for RabbitMQ, follow these steps:
    1. Open the **Control Panel**, navigate to **System and Security**, then click on **Windows Defende Firewall**.

2. Click **Advanced Settings**.
3. In the left menu, select **Inbound Rules**, then **New Rule...** in the right pane.
4. Choose **Port** as the rule type and click **Next**.
5. Select the **TCP** protocol, select **Specific local ports**, write the required RabbitMQ ports (defaults are **5672** and **15672**, include others as needed) and click **Next**.
6. Select **Allow the connection** and click **Next**.
7. Choose when this rule applies (**Domain**, **Private**, **Public**)- select all if unsure.
8. **Name** the rule (e.g., "RabbitMQ"), add an optional **Description**, and click **Finish**.
9. Repeat the same process for **Outbound Rules** if needed.

## HyperScan installation & Configuration

In case of a ZG scanner, below is the procedure to install the HyperScan software.

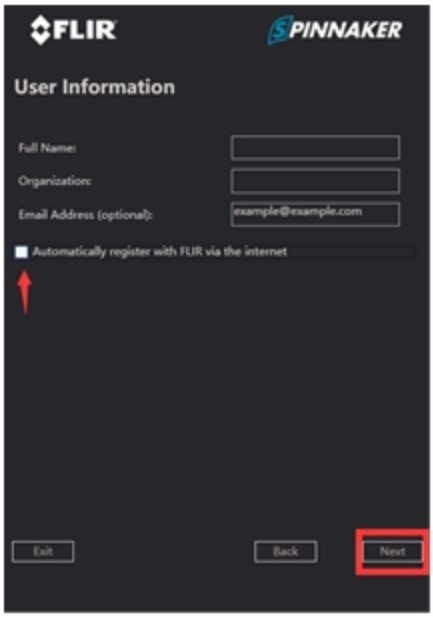
HyperScan is the dedicated acquisition software for the ZG Scanner and ZG H-Track Ultra (Super). The following files are required for installation:

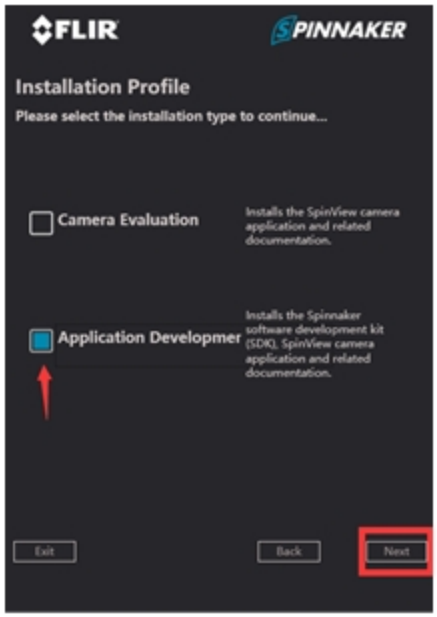
1. Camera Driver: `SpinnakerSDK_FULL.X.X.X.XXX_x64.exe`
2. Scanning Program: `HyperScanX.X.X_Setup.exe`

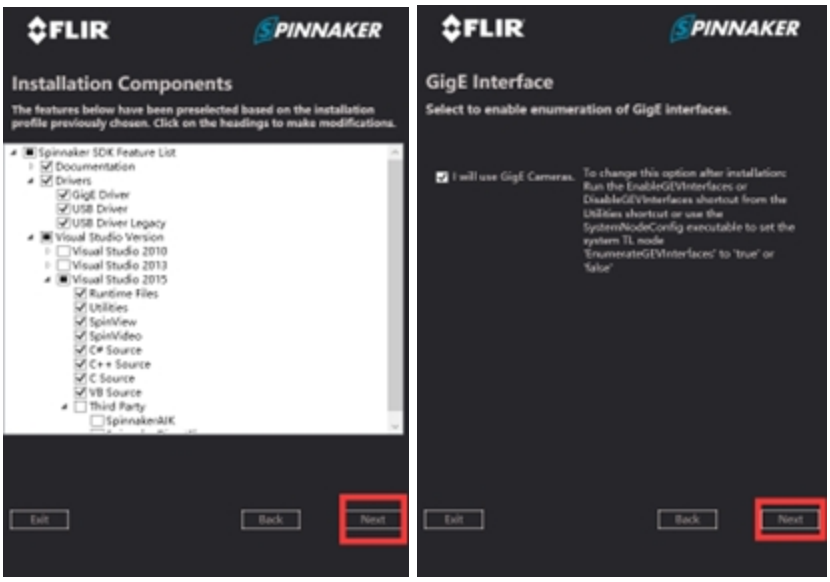
### Camera driver installation

The camera driver needs to be installed only once per computer. Subsequent updates only require updating the scanning program, and no changes to the camera driver are necessary.

Double-click the camera driver `SpinnakerSDK_FULL.X.X.X.XXX_x64.exe` and follow the steps shown in the image below to complete the installation.





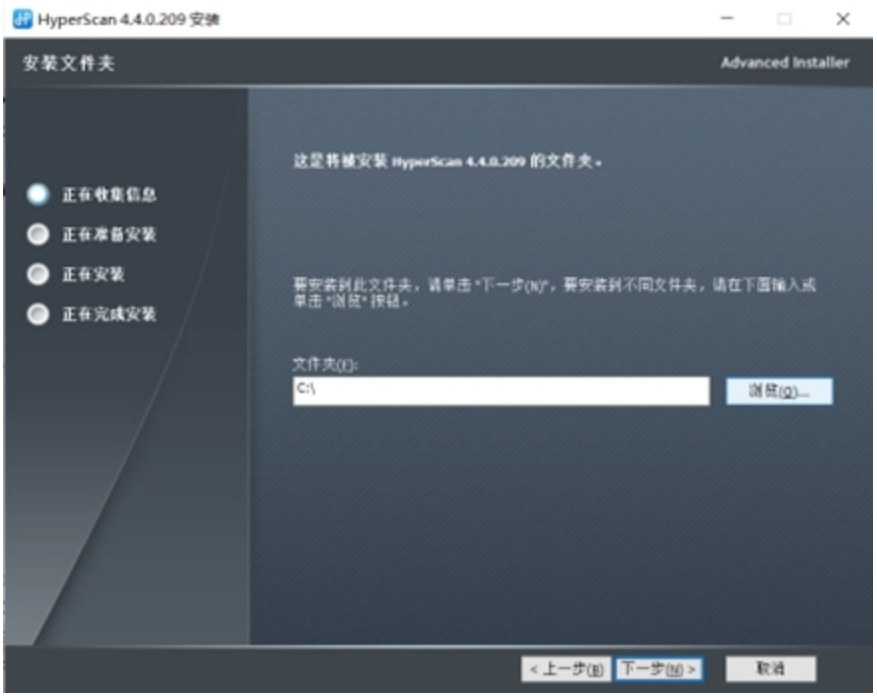

Proceed with the installation after selecting the directory.

## Scanning Program Installation

The scanning program is the operational software for the scanner. If a new version of the scanning program is released, you can directly install the new version. Both old and new versions can coexist.

1. Double-click the scanning program `HyperScanX.X.X_Setup.exe` to begin the installation.

The default installation path is the C drive, but it is recommended to select a drive other than C.

2. 

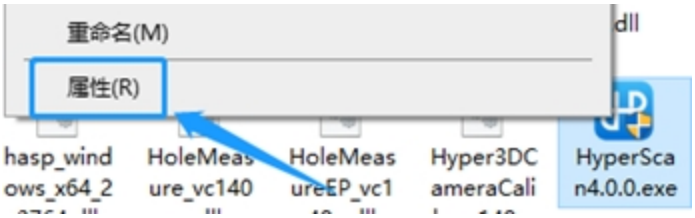
Click the "Install"(安装) button to complete the installation of the scanning program.

3.


4. A shortcut icon for the scanning program will appear on the desktop.

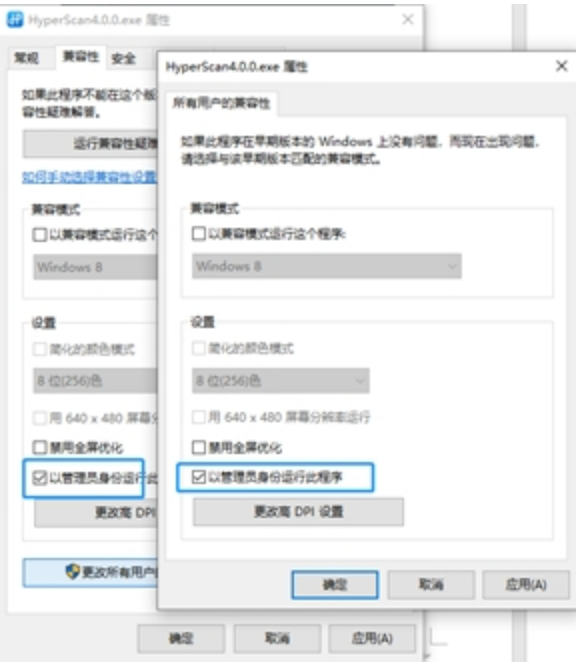
## Scanning Program Configuration

Right-click on the scanning program icon `HyperScan.exe` and follow the steps below to set administrator permissions for the scanning program. Click on Properties(属性), then select Compatibility(兼容性).



When the HyperScan scanning software is installed in folders such as `*C:/Program Files*` that require administrator permissions, it needs corresponding permissions to run.

Check the option to Run as Administrator for all users.




## Device Management

A set of scanner device files includes the following: calibration board file, scanner file, calibration rod file, tracker file, and camera file.

`*XXXXXX_L_XXXXXXXXX.haso*`: Track left camera parameter file

`*XXXXXX_R_XXXXXXXXX.haso*`: Track right camera parameter file

`*Scan_board.zghs*`: Scan calibration board file

`*Scan_config.zghs*`: Scan parameter file

`*Track_board.zghs*`: Track calibration rod file

`*Track_config.zghs*`: Track parameter file

`*HyperScan_config.zgcfg*`: Consolidated configuration file, combining the Scan calibration board file, Scan parameter file, Track calibration rod file, and Track parameter file

1. Importing the Consolidated Configuration File:

   Open the software and click on "Device Management" in the upper right corner of the interface.

   Click "Import Configuration"(导入配置) and select the consolidated file from the local `config` folder to import.

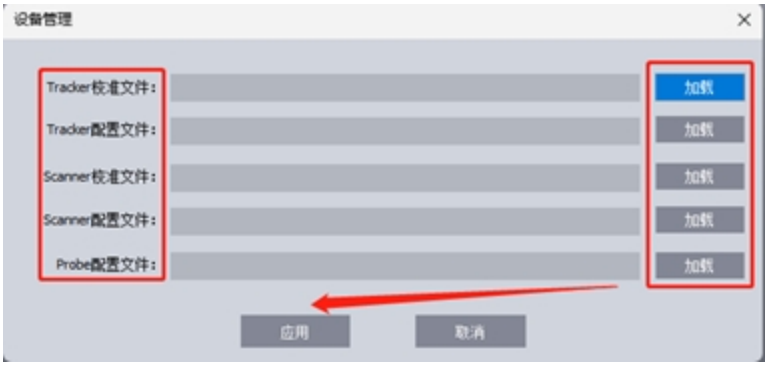
2. Replacing Individual Device Files:

   1. Open "Device Management" and click "Update Configuration"(更新配置).

      Click "Load" and select the individual device file to be replaced from the local `config` folder.

   2.

Click "Apply"(应用).

3. 


# Configuration and Licensing

## Licensing

There are 2 different licenses providers available for HxGN Robotic Automation. The legacy Nalpeiron Zentitle and the newer Hexagon Unified Licensing (HLM).

### Nalpeiron Zentitle

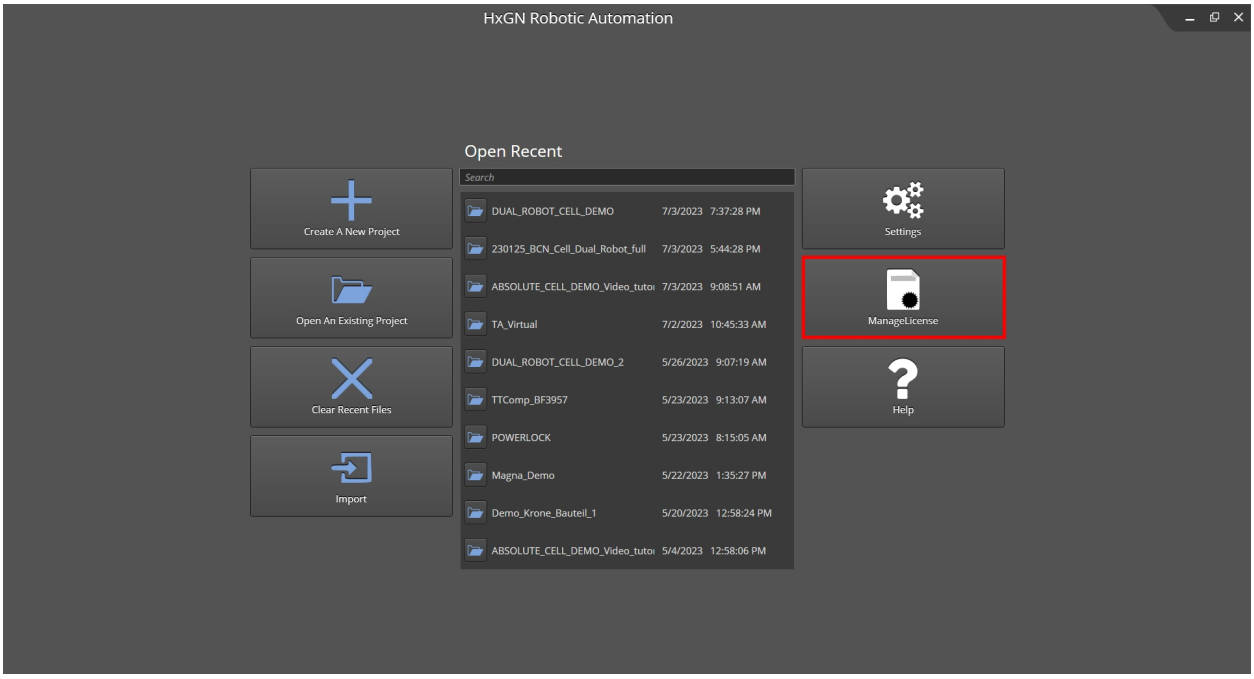
If the Napleiron Zentitle is used, inside “C:\ProgramData\Hexagon\RA\Config” folder, inside “RAconfig.xml” file, the following string must be edited.

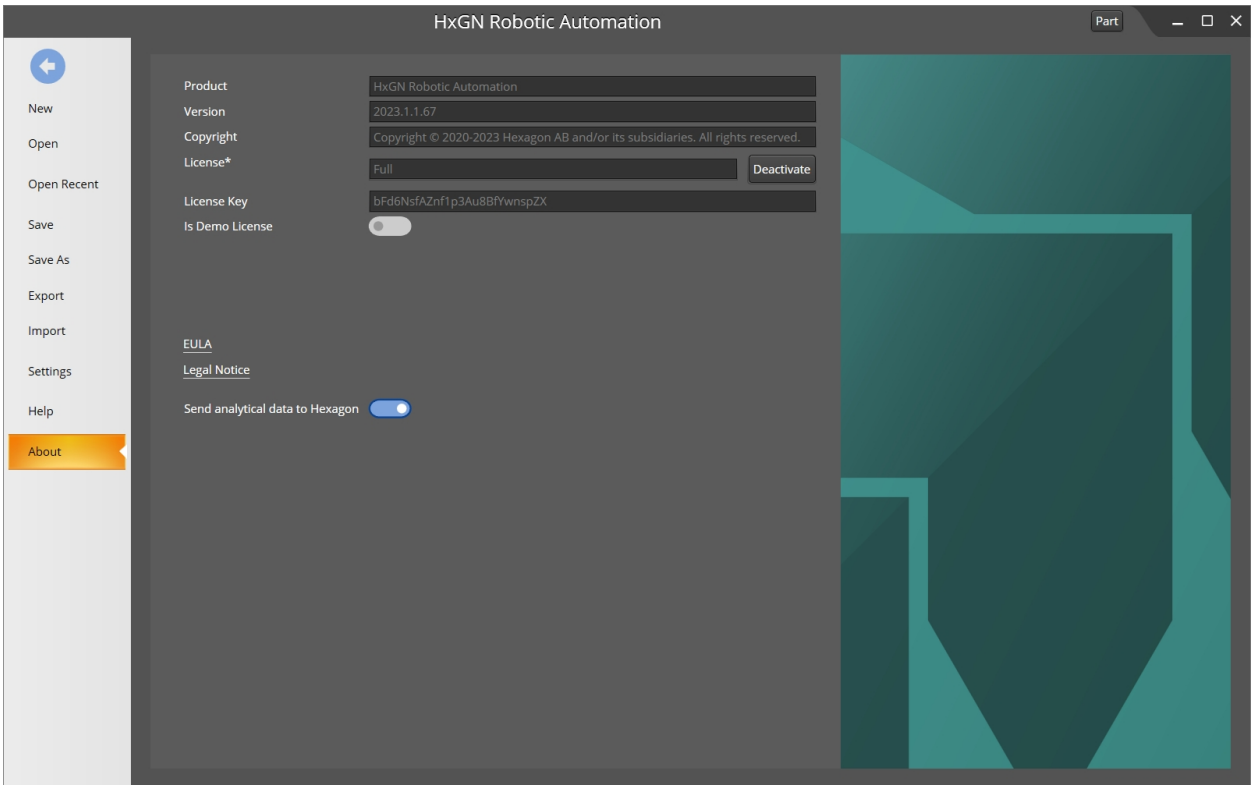
*<ConfigSetting Name="UseUnifiedLicensing" Value="False" DefaultValue-e="True" DisplayName="" Description="" TypeName="System.Boolean" />*

Value tag must be changed to “**False**”, file saved, and HxGN Robotic Automation restarted.

If this is the first installation, HxGN Robotic Automation will be launched in Viewer mode. Viewer mode will only allow you to open already created projects but will not allow you to execute programs and save projects.

To activate a license, from the welcome dashboard click on **Manage License** button and then click on **Activate** button.



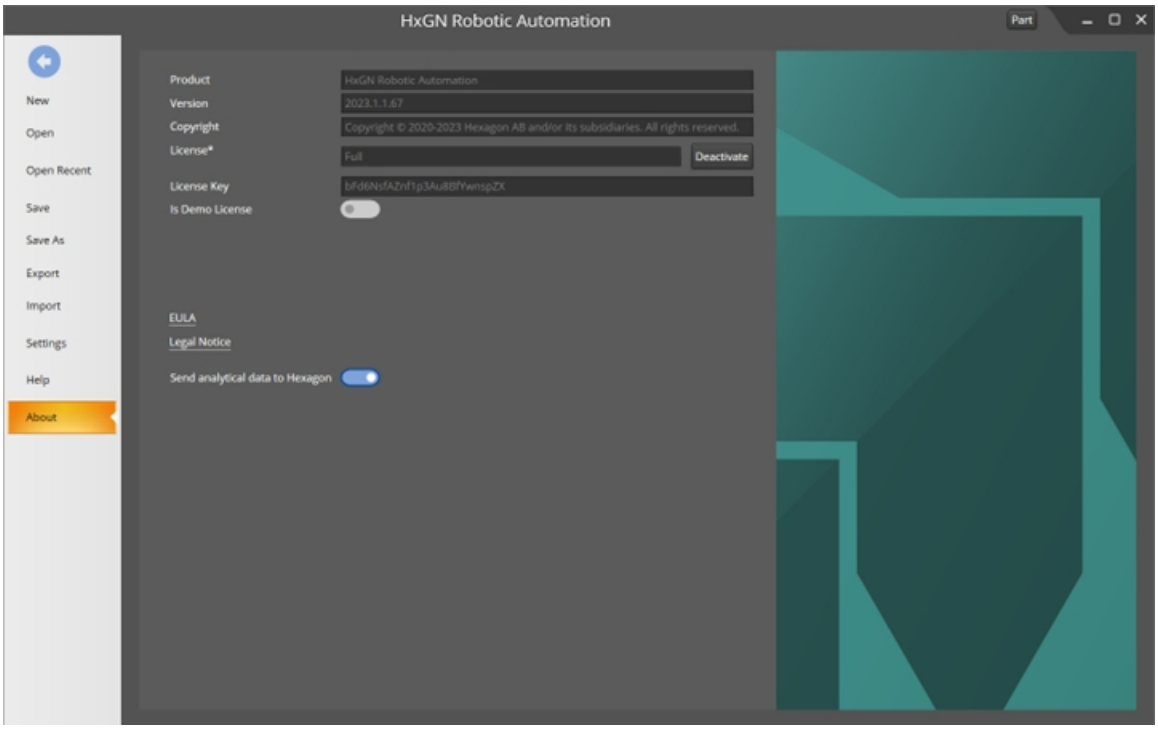



The **Activate Your License** dialog will appear. The user can enter the License key, fill the required fields, and click on **Activate** button.

> ⚠ **Warning**: to correctly activate the license, the same user that installed HxGN Robotic Automation to Windows must be logged in to activate the license.

License activation is not allowed from Remote Session. The Software will automatically try to activate over the internet.

If the license key and all the other fields are correct, the license will be shown, and **Deactivate** button will appear.



**Warning**: To transfer the license on another computer, first it must be deactivated from the computer where in use otherwise the activation on the new computer will fail (too many activations for the same license).

## Hexagon Unified Licensing

If the Hexagon Unified Licensing is used no change should be made. In case of an issue verify inside “C:\ProgramData\Hexagon\RA\Config” folder, inside “RAconfig.xml” file, the following string must be edited.

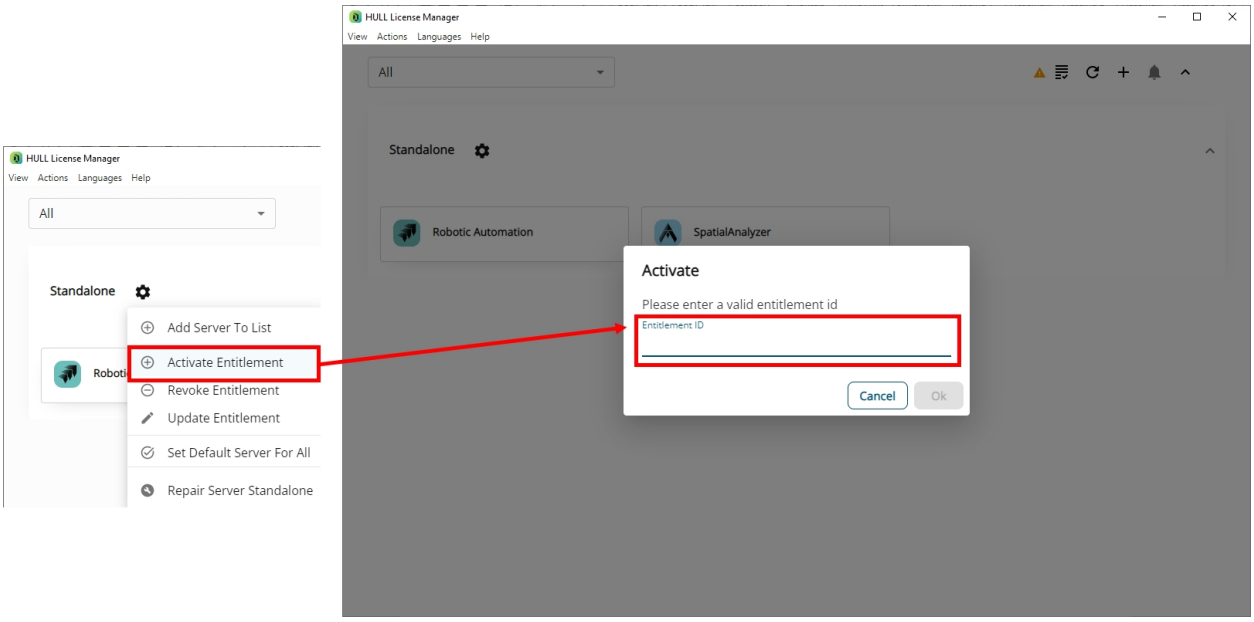
*<ConfigSetting Name="UseUnifiedLicensing" Value="False" DefaultValue-e="True" DisplayName="" Description="" TypeName="System.Boolean" />*

Value tag must be changed to "**True**", file saved, and HxGN Robotic Automation restarted.

To enable the license, open the **Hexagon License Manager application** (HLM from now on) that was installed together with HxGN Robotic Automation.

#### Online activation

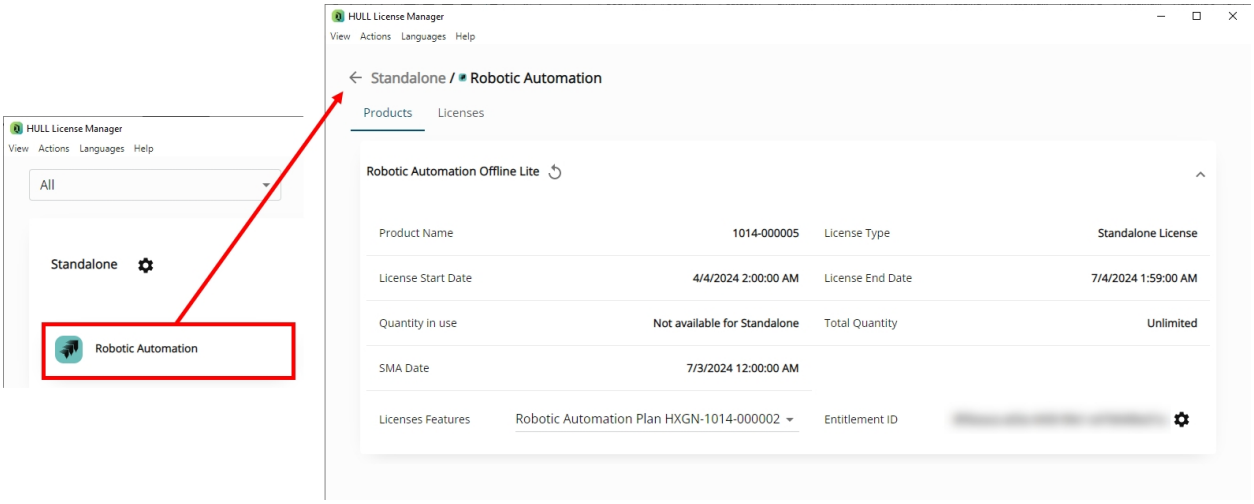
Click on the cog icon and select on Activate Entitlement.



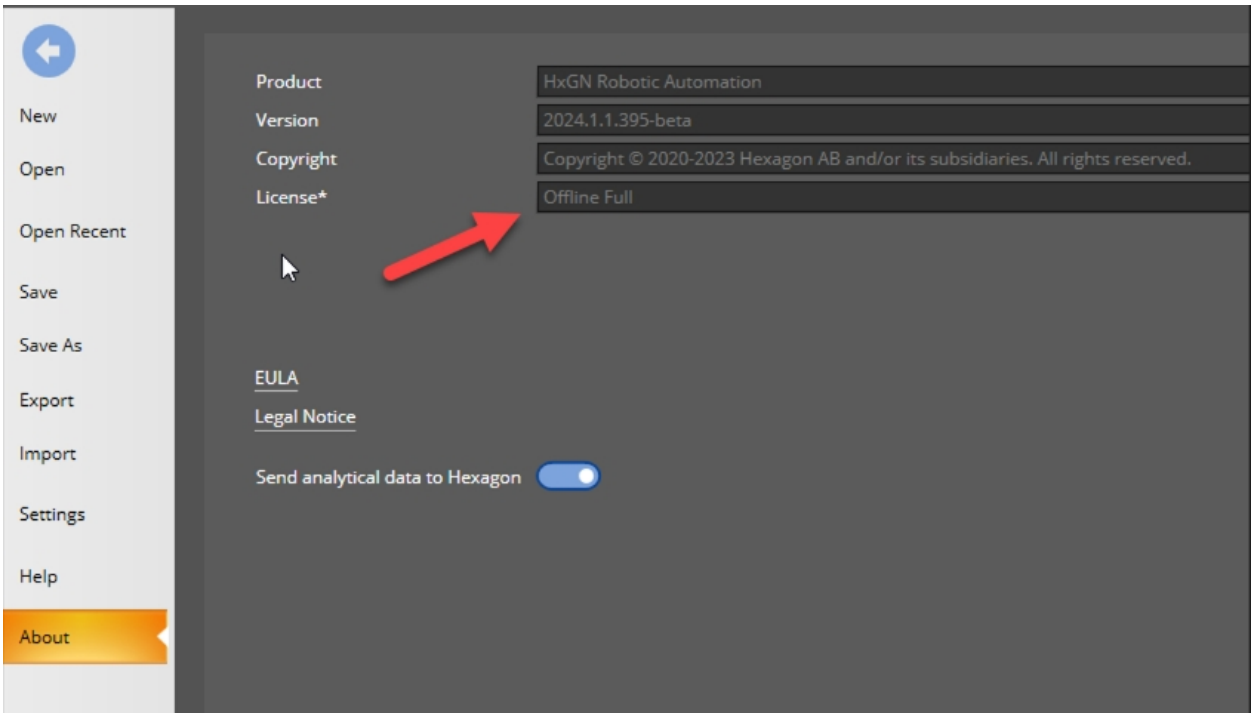

Enter the Entitlement ID and press the **Ok** button to confirm.

Once the entitlement is activated it will be shown as a tile in HLM app.

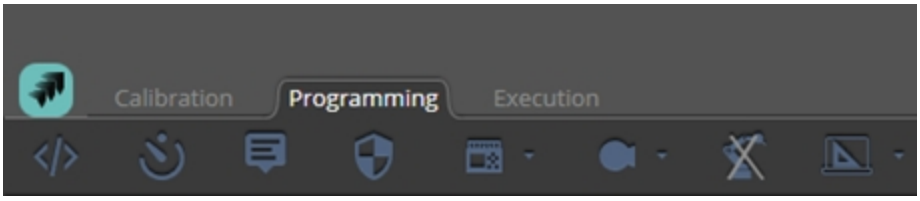
Clicking on the tile it is possible to see the details of the license activated on the system.




The license type can be visualized also in the About section of HxGN Robotic Automation.



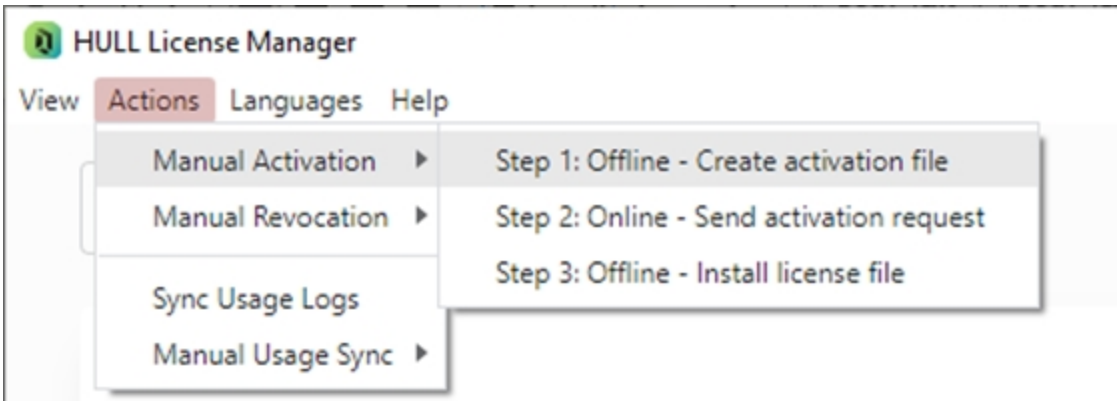

Depending on the license type in use, “Lite” or “Full”, the Modelling capabilities (and the related Modelling and Cell Layout tabs) will be disabled or enabled respectively. See e.g., image below showing how the application mode tabs will look like in case of using a “Lite” license.




**Offline activation**

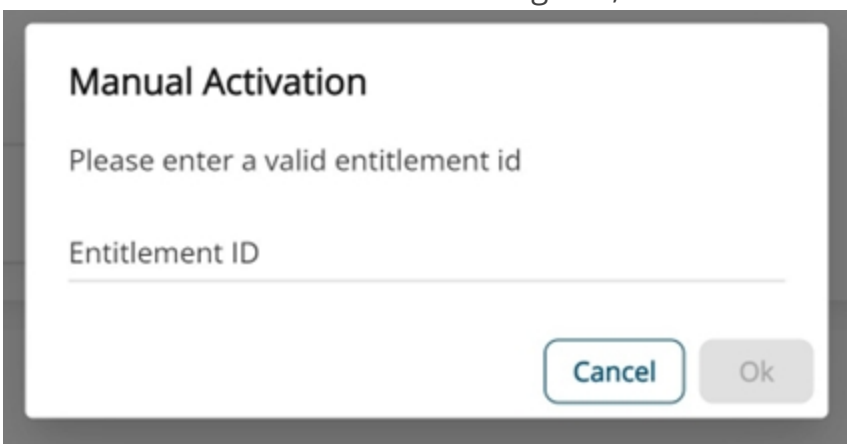
Manual activation allows a user in an offline environment to activate an entitlement. This process involves moving files to and from a computer with internet connection and the offline computer. This “handshaking” ensures that the process was completed successfully.

There are 3 steps involved in Manual Activation.




1. Offline Computer - Create activation file

   Enter the Entitlement Id in the dialog box, click **Ok**.

   a. 


   b. The Save dialog is displayed. Click **Save** to use the default or give it a meaning full name, then click **Save**. This will create a file with a HMA1 extension that is needed in the next step.

   c. Copy file from Offline Computer to transfer to the Online Computer.

2. Online Computer - Send activation request

   a. An **Open dialog** will be displayed. Select the HMA1 file from the Offline Computer and click **Open**.

b. HLM will communicate with the EMS server and then a Save dialog will be displayed. Click **Save** to use the default or give it a meaning full name, then click **Save**. This will create a file with a HMA2 extension that is needed in the next step.

c. Copy file from the Online Computer to be transferred to the Offline Computer.

3. Offline Computer - Install license file

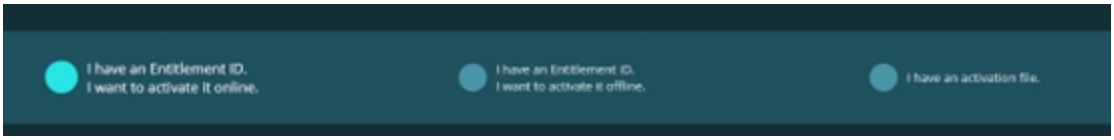
   a. An Open dialog will be displayed. Select the HMA2 file from the Online Computer and click **Open**.

   b. HLM will process the file and install the license.

At this point the license should be installed, ready for use and displayed in HLM.

## CLMS Licensing

The customer activation includes the following two types:



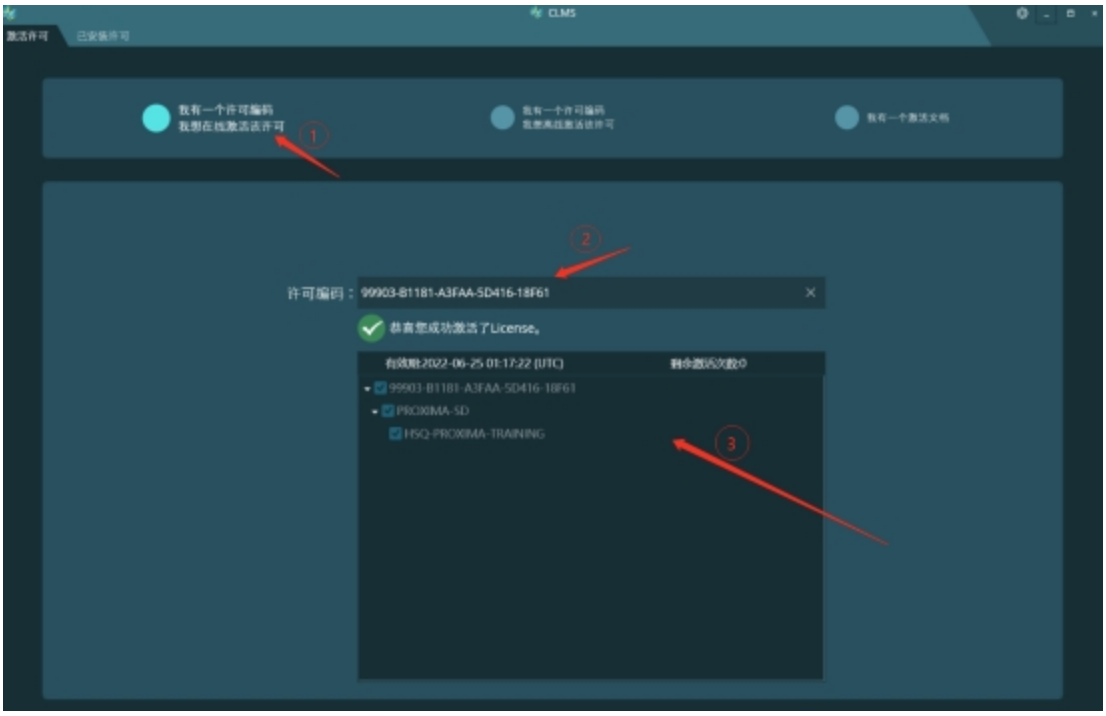

**Online License Activation**

Activate the license when there is an internet connection.

The activation method for online license activation is very simple.

1. After the service engineer or user obtains the license code.
2. Enter the "license code" on the corresponding interface of "I have a license and I want to activate it online",
3. The online license activation will be automatically completed.

4. After the license is successfully activated, there will be an information prompt indicating *activation successful*, the corresponding modules of the license, the expiration date, and the remaining number of licenses.




| | |
|---|---|
| ⚠ | If the corresponding software licenses exceed the number of licenses that can be activated with the license code, there will be a prompt as shown in the following figure: |

**Offline License Activation**

Obtain the license when there is no internet connection.

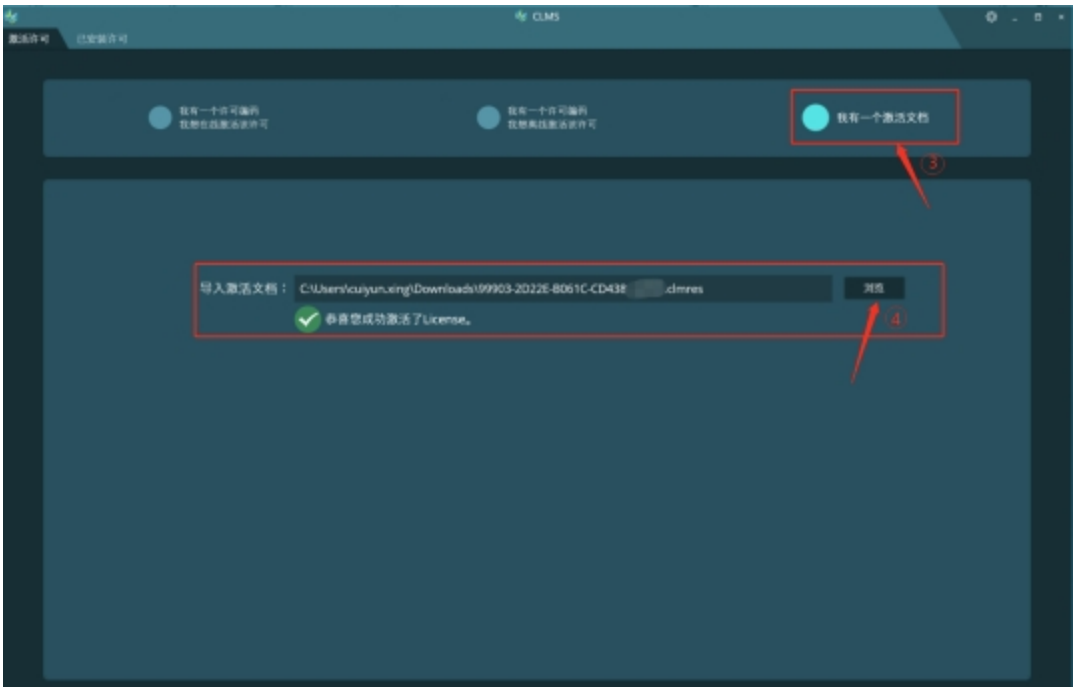
Offline activation of software licenses is used when the devices purchased by users do not support connecting to the external network.

In the absence of an internet connection, service engineers or end - users can carry out this process on the premise of internal network support or internal approval.

The specific operations are as follows:

1. Enter the obtained license code.
2. Locate the generated offline document according to the prompts.
3. Send the offline activation application document to HexCloud360@hexagon.-com.

4. The person in charge of generating software licenses will send the corresponding offline activation license document to the applicant.
5. After receiving the offline activation license document, the applicant only needs to import the activation document in the "I have an activation document" section to activate the license.

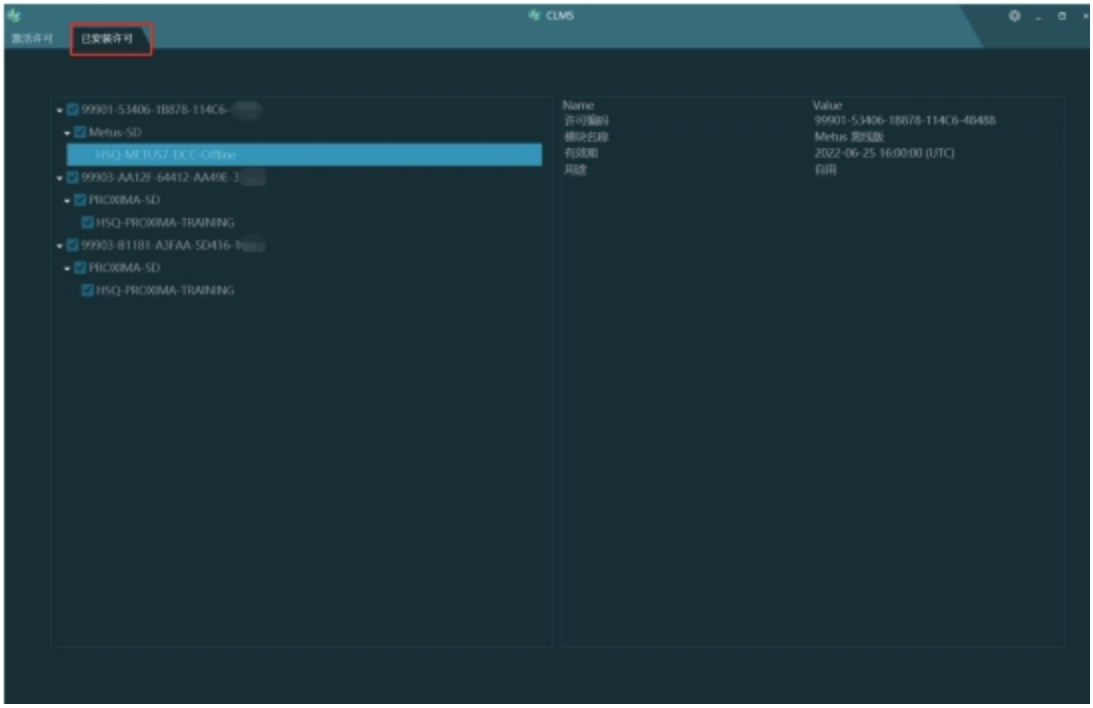



The sample of the offline activation application document generated by the client is as follows:

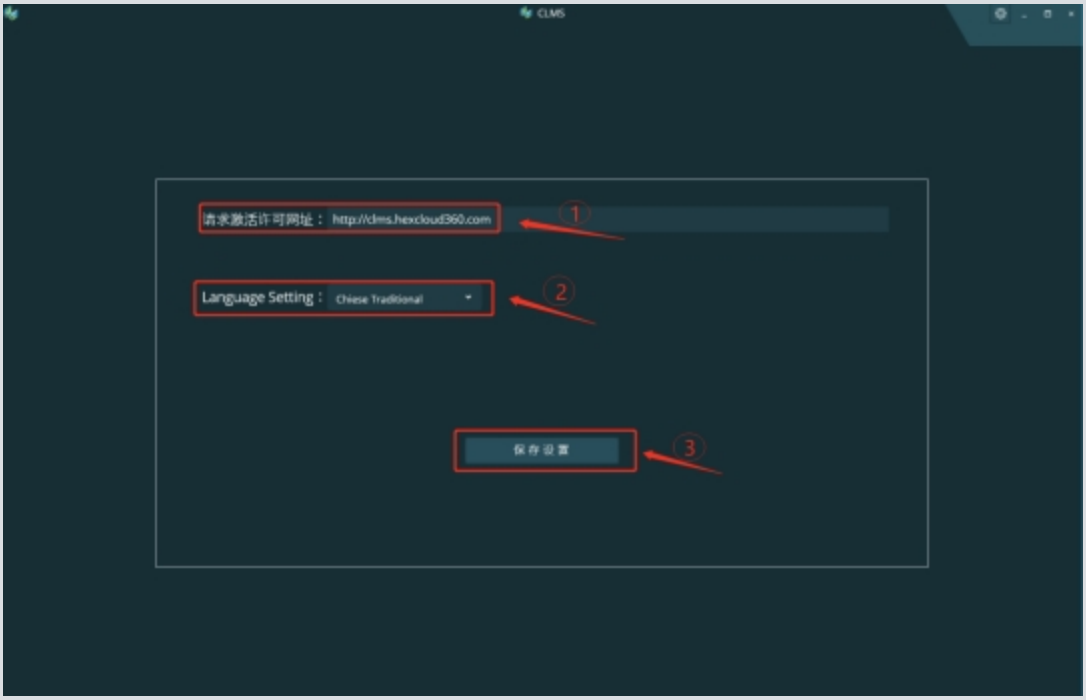
The sample of the offline activation license document provided by the server is as follows:

#### Query Existing License

Users can view the summary of the activated license information through this interface. The information on this interface can only be viewed when the *Online License Activation* and *Offline License Activation* have been **successfully activated**. The display information is as shown in the following figure:




If users do not need to make any settings, they do not need to pay attention to this page. If users need to deploy using their own network and activate the license, they can make the settings on this page:



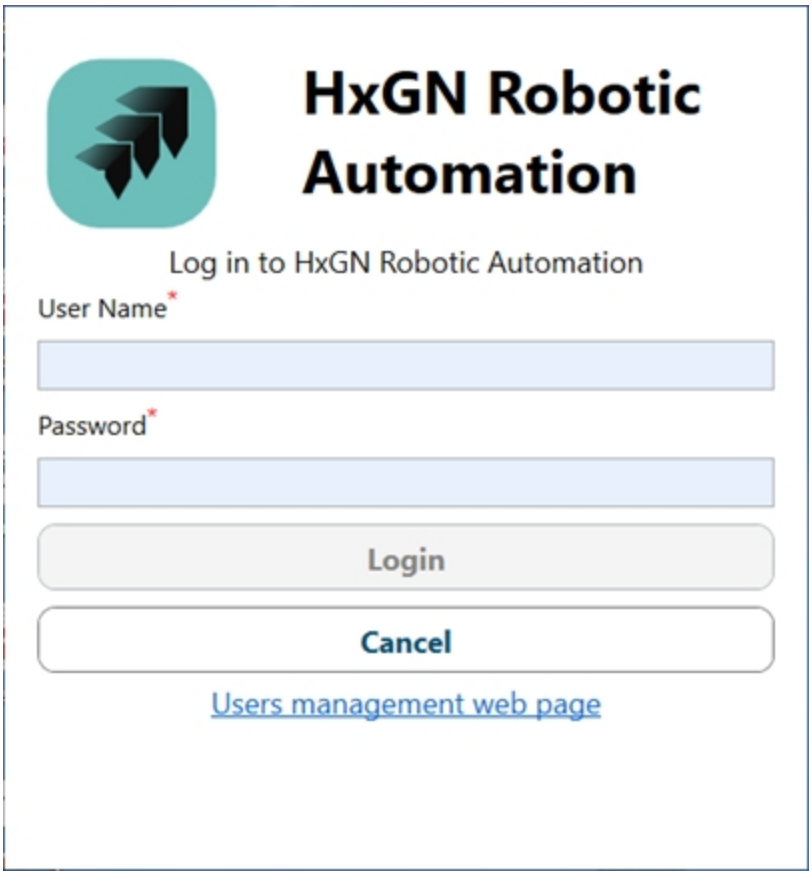

## User Management

HxGN Robotic Automation provides a user management portal where it is possible to define roles that decide the access rights to HxGN Robotic Automation and Execution Manager apps, according to the following table:

| Permission | User Manager | Programmer | Operator |
|---|---|---|---|
| Limited Execution Manager access:<br>• execute playlists/programs<br>• change language | ✅ | ✅ | ✅ |

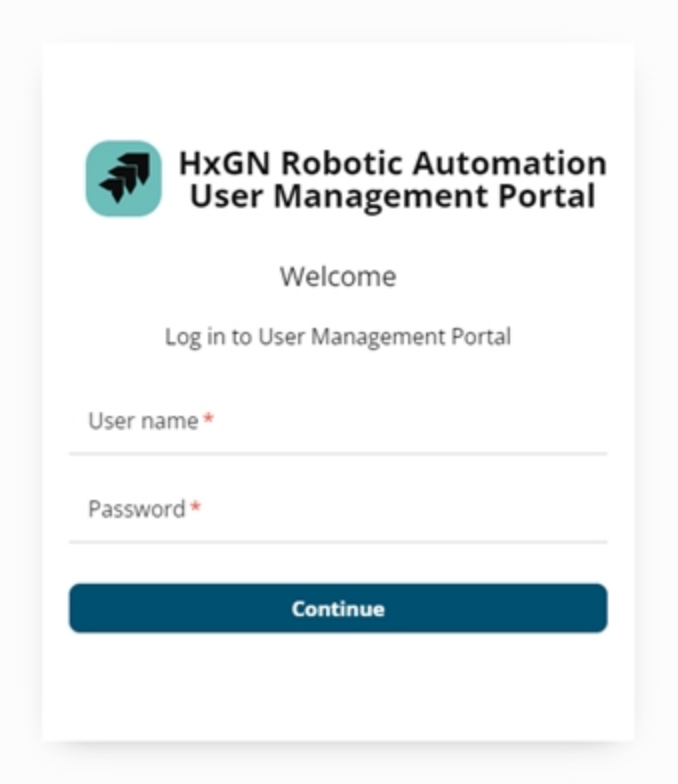
| Permission | User Manager | Programmer | Operator |
|---|---|---|---|
| • visualize execution logs<br>• visualize cell status | | | |
| Full HxGN Robotic Automation access | | | |
| Full Execution Manager access | | | |
| User Management portal access to create, edit, and remove users | | | |

When the user opens HxGN Robotic Automation, he will see a login screen. Without entering a correct username and password the user will not be allowed to login into the application.

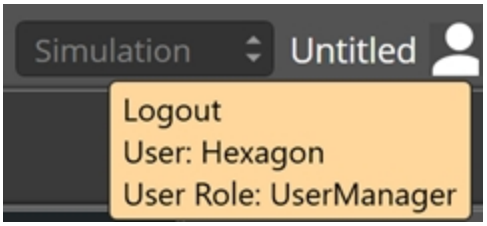



The user can login into HxGN Robotic Automation by inserting username and password, and by clicking on the **Login** button. If the user clicks on the **Cancel** button, the login screen closes, and HxGN Robotic Automation must be restarted to login.

When clicking on the link to the **Users Management web page**, the user will be redirected to the user management web app, where he can configure new users and passwords as explained this section.

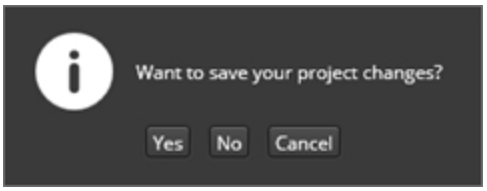



When logged in in HxGN Robotic Automation, the user can logout from the current session by clicking the logout button ( ) in the top-right part of the UI. Hovering over the logout button, the user will see the current username and user role.

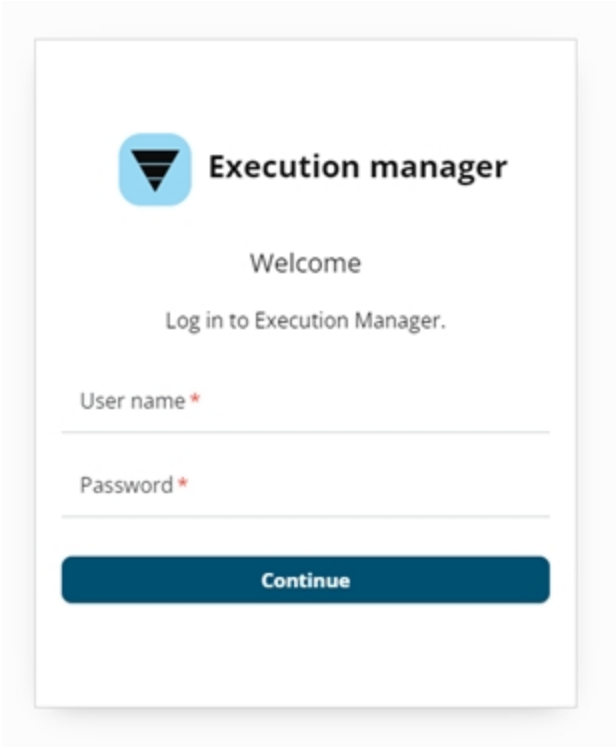



After clicking on the logout button, the user will be asked if he wants to save the project.

**Yes**, saves the project, **No** discards the changes. In case these two options are chosen, the login screen will be shown again. In case the user clicks on **Cancel**, the logout process will be canceled, and the prompt dialog box closed.




When the user opens Execution Manager, he will see a login screen. Without entering a correct username and password the user will not be allowed to login into the application. Depending on the role, the user will have different permissions when logged in to Execution Manager, as explained in detail in Execution form Execution Manager

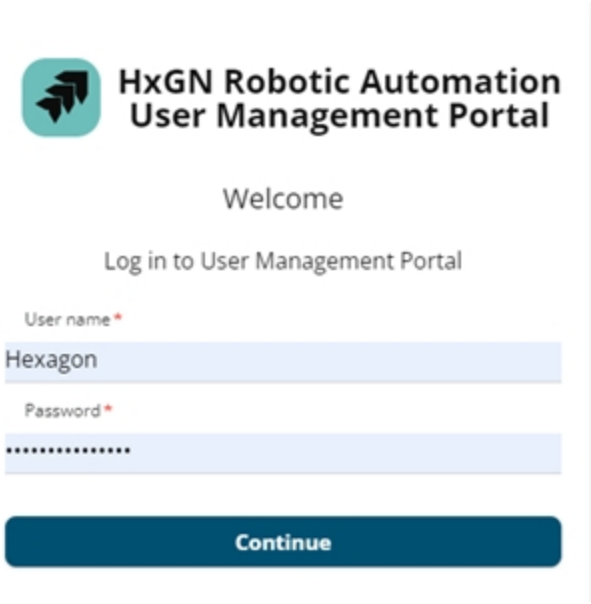



## User Management Portal

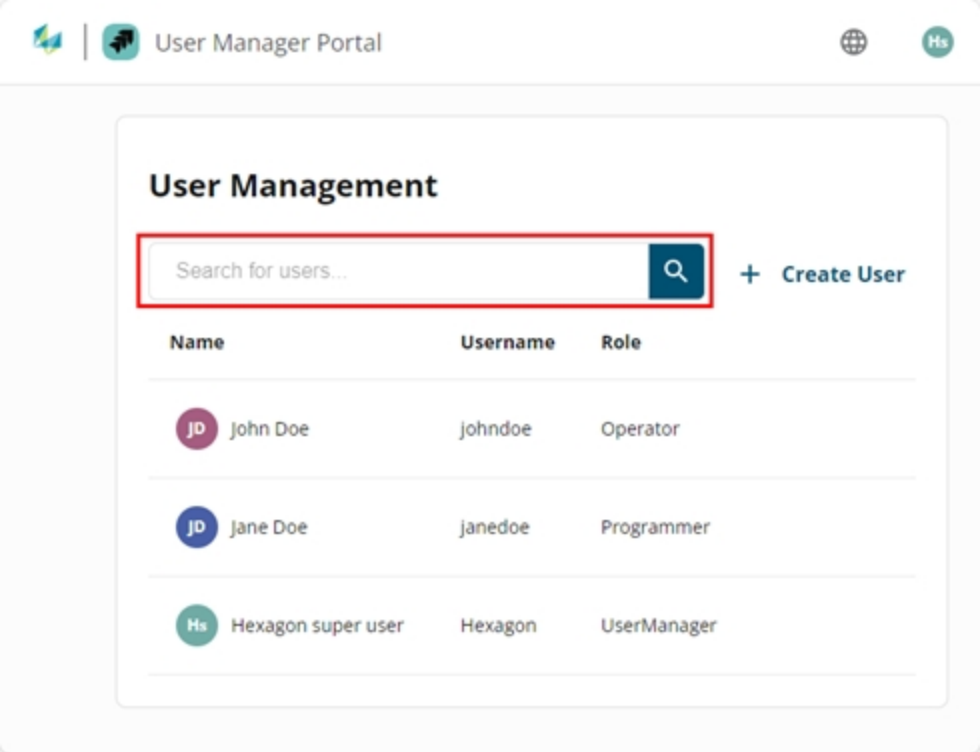
The user can login into HxGN Robotic Automation User Management Portal using his credentials and clicking on the **Continue** button. If the user does not have the necessary permissions, an error message will be shown.

By default, a User Manager is available with Username **Hexagon** and Password **Hexagon**.

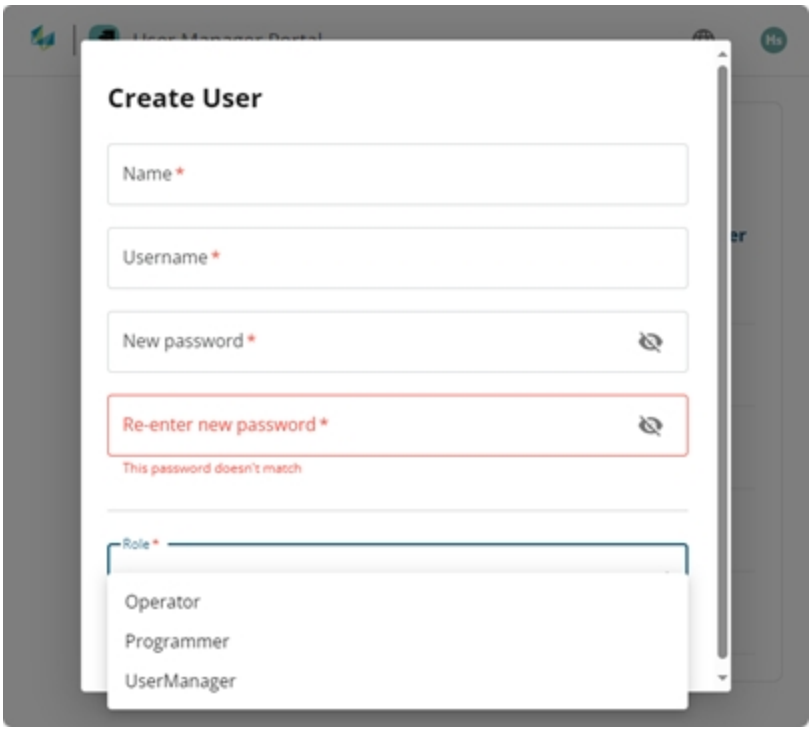
It's recommended to create a new user after first login in the user management portal and delete the default user.




The User Manager portal will look like the following image where the list of all users created by the User Manager are visible in a list with a search bar on top.




The User Manager can create users by clicking on the **+ Create User** button that will open the following pop-up where it is possible to define Name, Username, and password for the new user and assign a role among the possible ones, as explained in the previous chapter and related table.



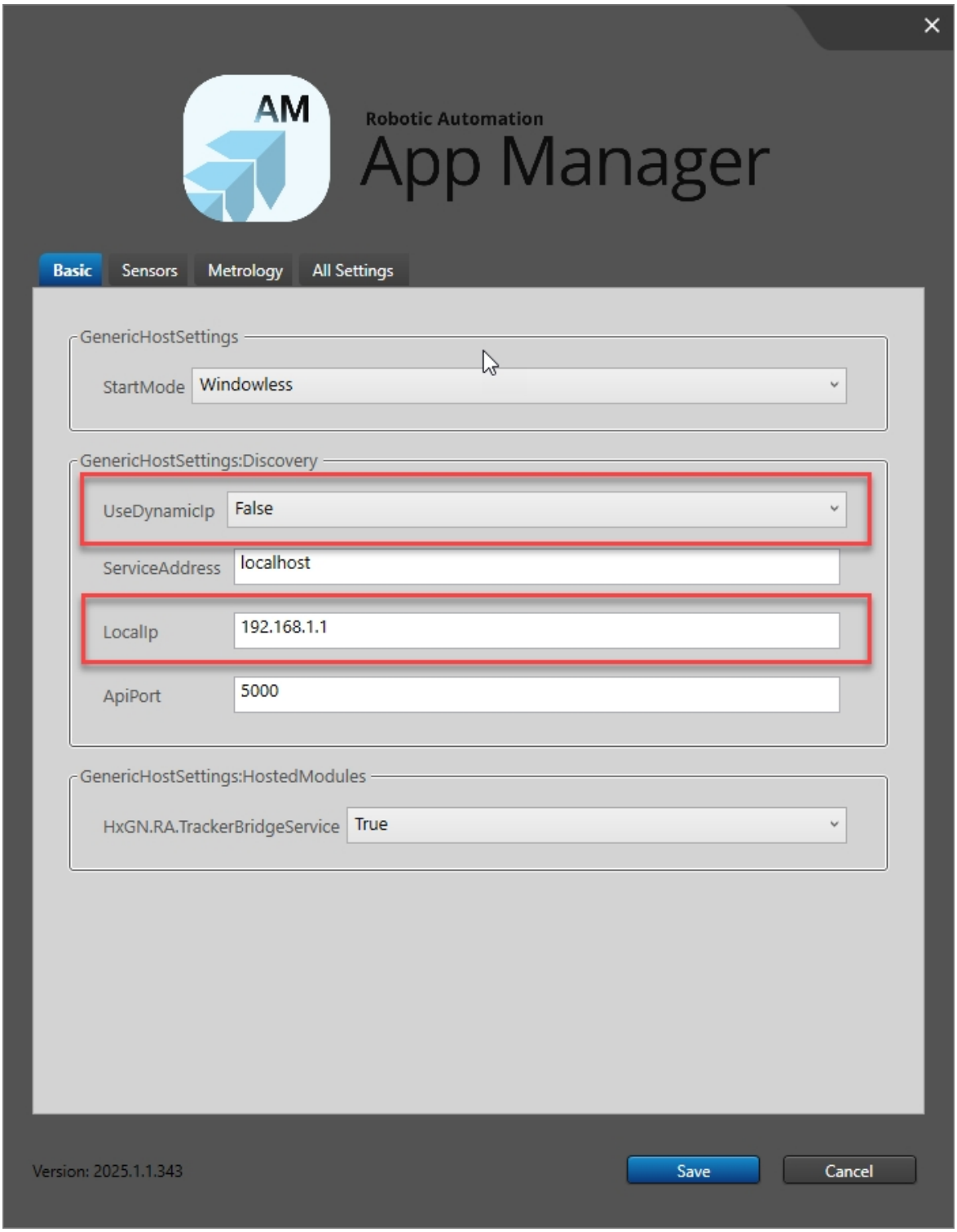

JSON configuration through App Manager

### Generic Host

Open the App Manager and on the main page you:

1. Configure the “UseDynamicIp” to **false** (all small letters)

2. Set up the IP in “LocalIp” from the network card where Leica HW devices are configured (e.g., 192.168.5.245).


AM
Robotic Automation
App Manager
Basic
Sensors
Metrology
All Settings
GenericHostSettings
StartMode
Windowless
GenericHostSettings:Discovery
UseDynamicIp
False
ServiceAddress
localhost
LocalIp
192.168.1.1
ApiPort
5000
GenericHostSettings:HostedModules
HxGN.RA.TrackerBridgeService
True
Version: 2025.1.1.343
Save
Cancel

In case of an offline PC, where there is no internet connection or local network card IP, the following parameters must be set:

"UseDynamicIp": false,

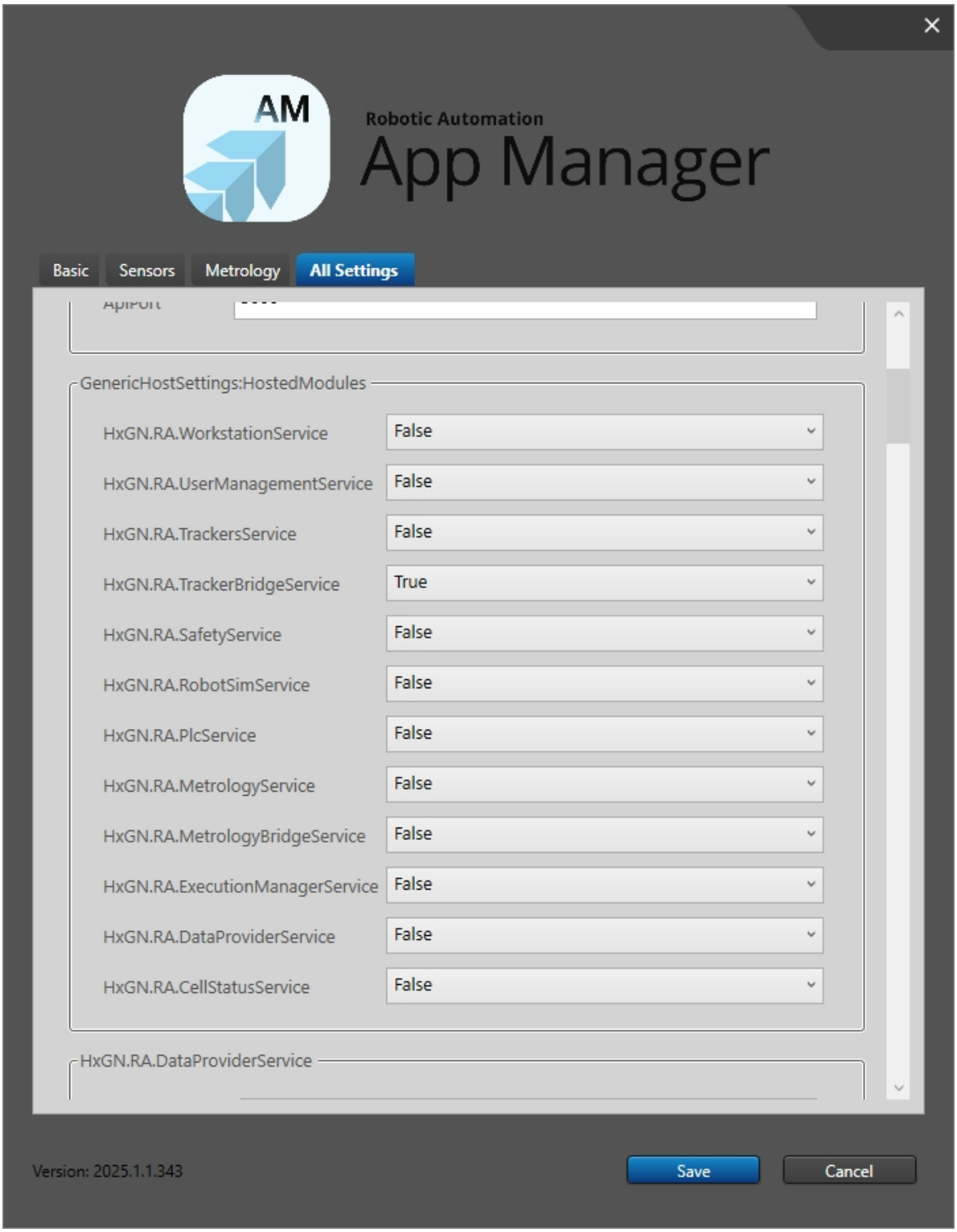
"LocalIp": "127.0.0.1", //This entry is used if "UseDynamicIp" is false or not present.

In case of a **dual robot cell**, and because the new implementation includes Tracker bridge inside the Generic Host, it is required to install only Generic Host in the second computer using the RA installer

The App Manager Settings, it should be configured in the second computer like in the image below.

1. Change the "*UseDynamicIp*" to **false**.
2. Set up the IP in "*LocalIp*" from the network card where the Leica HW is configured in the second computer (slave), e.g. **192.168.5.246** in the example below.
3. Change "*DiscoveryServiceAddress*" to the IP of the first computer (master), e.g. **192.168.5.245** in the example below.

4. Finally, in the "*HostedModules*" section, comment on all the services using "//" at the beginning except the "*HxGN.RA.TrackerBridgeService*."


AM
Robotic Automation
App Manager
Basic
Sensors
Metrology
All Settings
GenericHostSettings:HostedModules
HxGN.RA.WorkstationService
False
HxGN.RA.UserManagementService
False
HxGN.RA.TrackersService
False
HxGN.RA.TrackerBridgeService
True
HxGN.RA.SafetyService
False
HxGN.RA.RobotSimService
False
HxGN.RA.PlcService
False
HxGN.RA.MetrologyService
False
HxGN.RA.MetrologyBridgeService
False
HxGN.RA.ExecutionManagerService
False
HxGN.RA.DataProviderService
False
HxGN.RA.CellStatusService
False
HxGN.RA.DataProviderService
Version: 2025.1.1.343
Save
Cancel

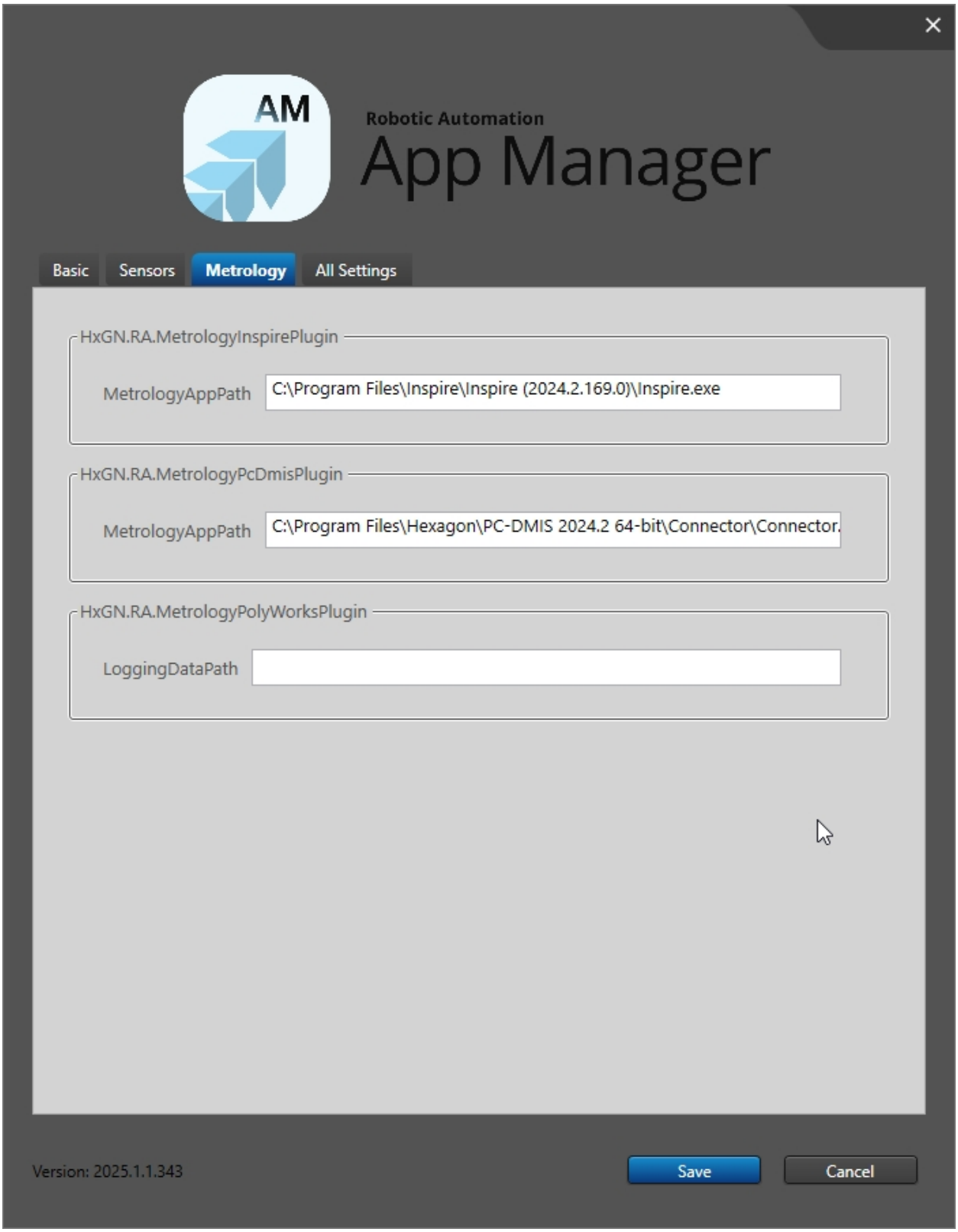
The Generic Host port is set by default to 80 and there is no need to set this port.

## Metrology

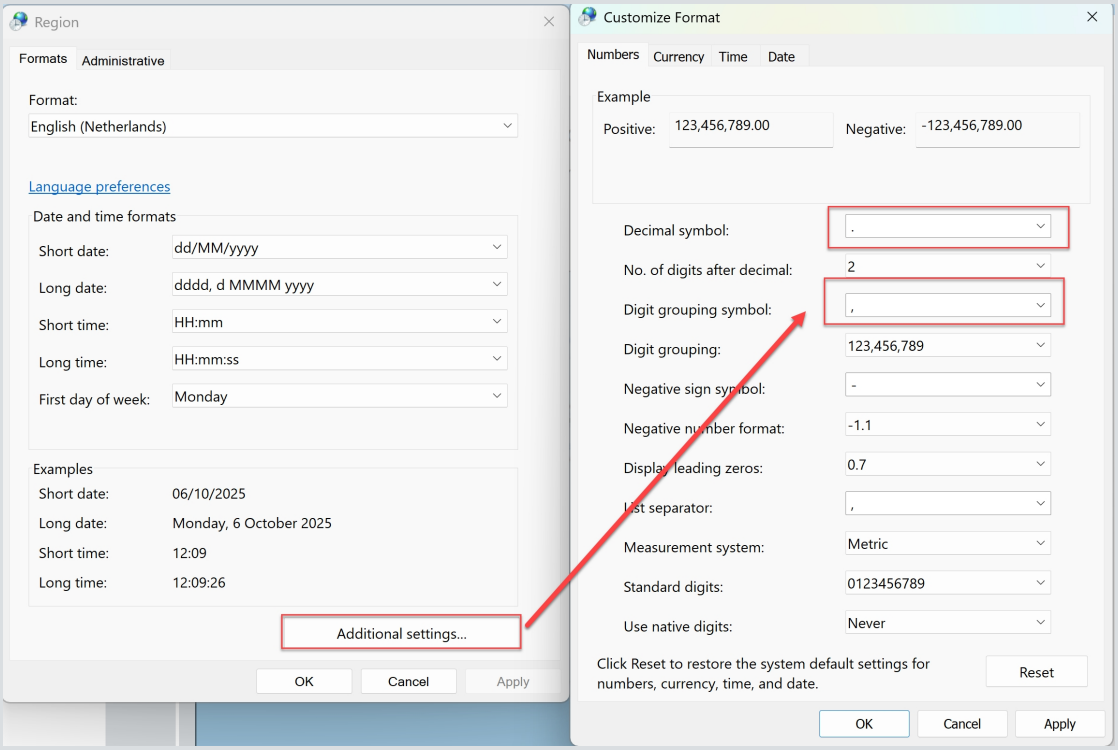
The Metrology path which applies to both HxGN Robotic Automation and Execution Manager is required to Execute validated programs exported the simulated or real data to Metrology.

| | |
|---|---|
| | HxGN Robotic Automationwill scan the Metrology Plugin folder located in: "C:\Program Files\Hexagon\ServerGenericHost\Plugins\Metrology" to scan for any new plugins. |

The available Metrology plugins will be shown in the Metrology tab of the App Manager and will require their executable to be linked.


AM
Robotic Automation
App Manager
Basic
Sensors
Metrology
All Settings
HxGN.RA.MetrologyInspirePlugin
MetrologyAppPath
C:\Program Files\Inspire\Inspire (2024.2.169.0)\Inspire.exe
HxGN.RA.MetrologyPcDmisPlugin
MetrologyAppPath
C:\Program Files\Hexagon\PC-DMIS 2024.2 64-bit\Connector\Connector.
HxGN.RA.MetrologyPolyWorksPlugin
LoggingDataPath
Version: 2025.1.1.343
Save
Cancel

For PolyWorks be sure that in Control Panel | Clock and Region clicking on Region | Additional settings..., the Decimal symbol is set to "." (dot), and the Digit grouping symbol is set to "," (comma), as shown in the picture below.




## Cell JSON

Every automation device in the cell has a unique ID to avoid any malfunction when using it.

To connect with the hardware, each robotic cell includes a file called **Cell.json** which needs to be created or modified with a special format where the devices GUID and computer names are configured.

There is a need to update the devices GUID in the HxGN Robotic Automation project to match exactly the corresponding device GUID used in the real cell model file (**Cell.json**).

Using a PRESTO cell delivered as turn-key solution by Hexagon, the following steps are not needed as the Cell.json file will be delivered together with the PRESTO cell.

Please, follow the next steps to generate and configure your Cell.json file:

1. Prepare your project template in HxGN Robotic Automation
2. Unzip the Cell.Json.Creator.2.0.1.7z to Cell.Json.Creator.2.0.1
3. Copy to that folder the project template (it must have **.bf** extension)
4. Open the Command Prompt and navigate to Cell.Json.Creator directory.
5. Type the next command:

Command format is:

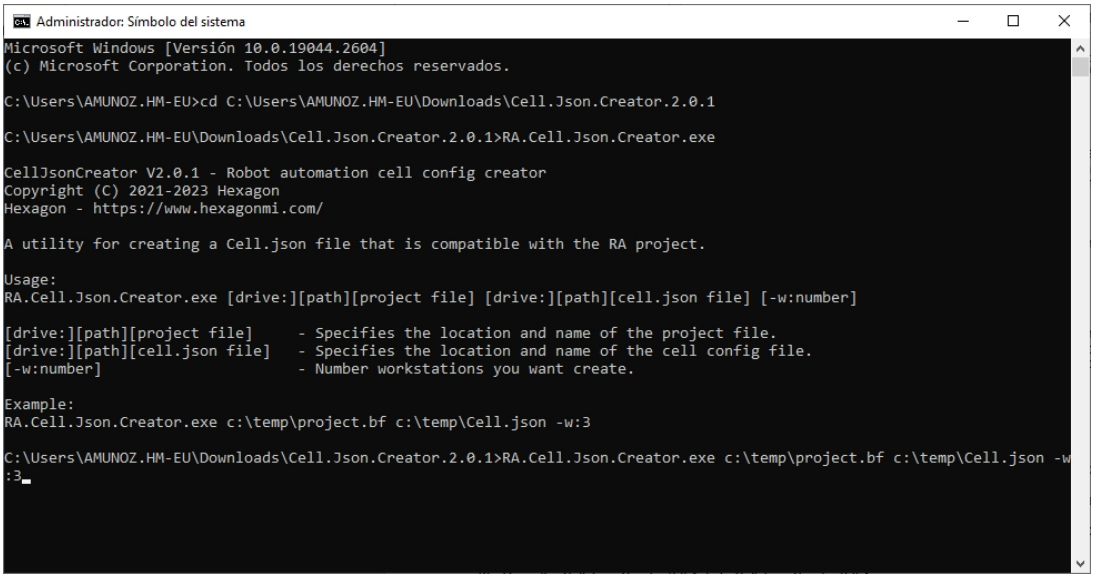
Cell Json command properties

```
RA.Cell.Json.Creator.exe [drive:][path][project file] [drive:][path]
[cell.json file] [-w:number]
```

| Command Property | Explanation |
|---|---|
| [drive:][path][project | - Specifies the location and name of the project |

| Command Property | Explanation |
|---|---|
| file] | file |
| [drive:][path][cell.json file] | - Specifies the location and name of the cell config file. |

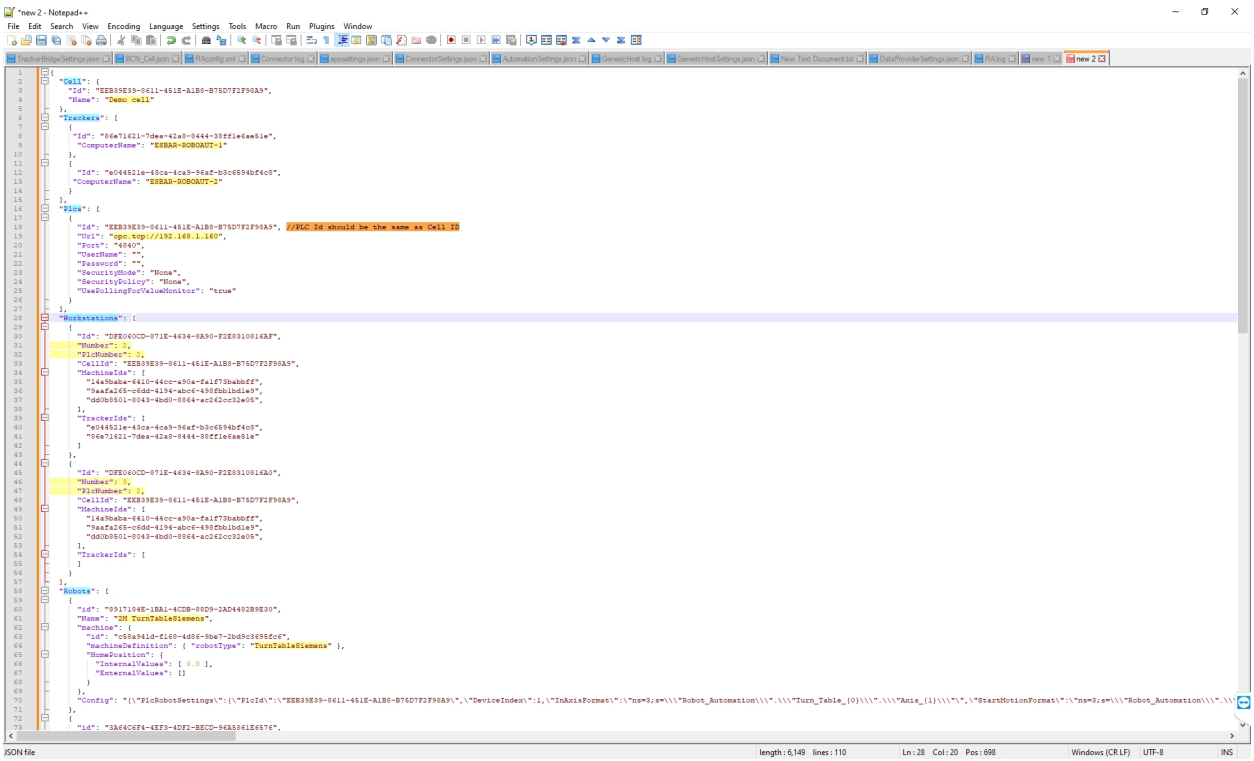
For example, type: RA.Cell.Json.Creator.exe project.bf Cell.json

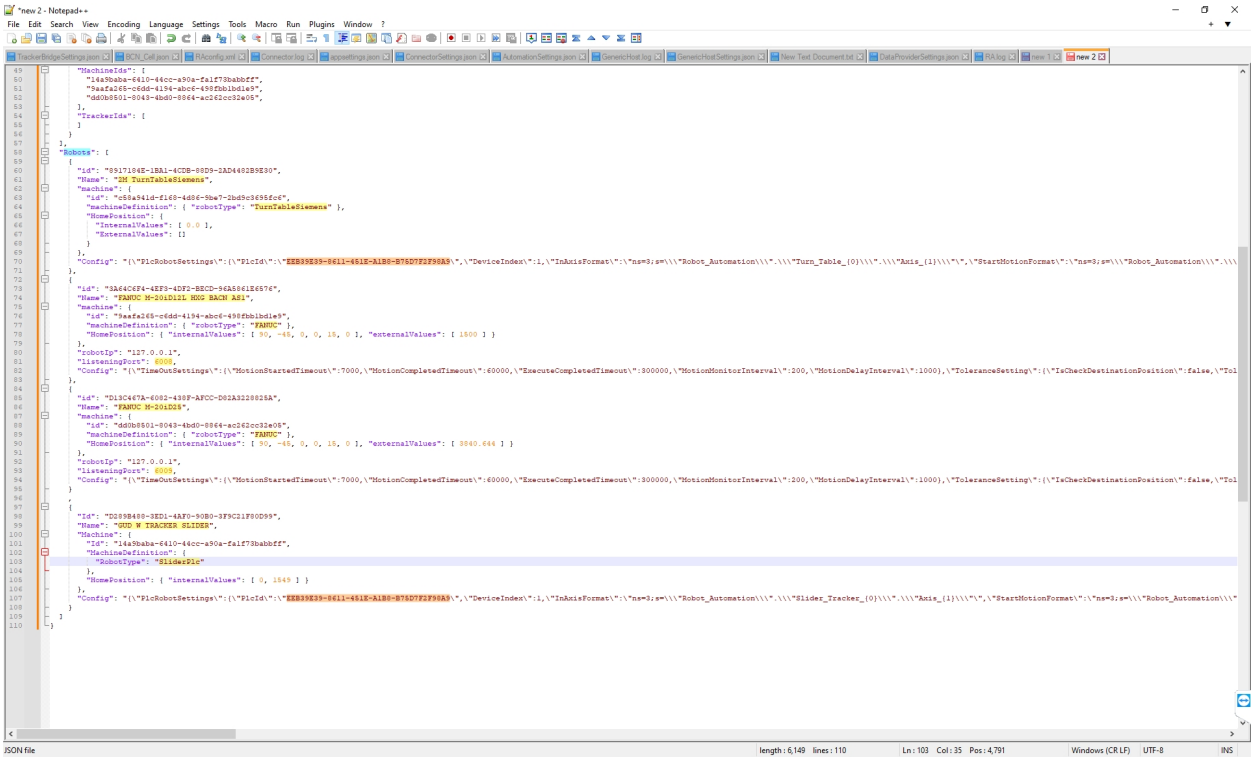



/

If you are in the project folder, then you may copy the project to the RA.Cell.Json.Creator folder and run it instead of typing the whole path.

A “demo_cell.json” file will be created as shown in the image below. It is just a template and still need to be manually edited.


*new 2 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
"Cell": {
"Id": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
"Name": "Demo cell"
},
"Trackers": [
{
"Id": "86e71621-7dea-42a8-8444-38ffle6ae5le",
"ComputerName": "ESBAR-ROBOAUT-1"
},
{
"Id": "e0445214-43ca-4ca9-96af-b3c6594bf4c8",
"ComputerName": "ESBAR-ROBOAUT-2"
}
],
"Plcs": [
{
"Id": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9", //PLC Id should be the same as Cell Id
"Url": "opc.tcp://192.168.1.160",
"Port": "4840",
"UserName": "",
"Password": "",
"SecurityMode": "None",
"SecurityPolicy": "None",
"UsePollingForValueMonitor": "true"
}
],
"Workstations": [
{
"Id": "DFE040CD-871E-4634-8A90-F2E8310816AF",
"Number": 2,
"PlcNumber": 2,
"CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
"MachineIds": [
"14a9baba-6410-44cc-a90a-fa1f73babbff",
"9aafa265-c6dd-4194-abc6-495fbb1d1e9",
"dd0b8501-8043-4bd0-8864-ac262cc32e05",
],
"TrackerIds": [
"e0445214-43ca-4ca9-96af-b3c6594bf4c8",
"86e71621-7dea-42a8-8444-38ffle6ae5le"
]
},
{
"Id": "DFE040CD-871E-4634-8A90-F2E8310816A0",
"Number": 2,
"PlcNumber": 2,
"CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
"MachineIds": [
"14a9baba-6410-44cc-a90a-fa1f73babbff",
"9aafa265-c6dd-4194-abc6-495fbb1d1e9",
"dd0b8501-8043-4bd0-8864-ac262cc32e05",
],
"TrackerIds": [
]
}
],
"Robots": [
{
"id": "8917104E-1BA1-4CDB-88D9-2AD482B9E30",
"Name": "2M TurnTableSiemens",
"machine": {
"id": "c58a941d-f168-4d86-9be7-2bd9c3695fc6",
"machineDefinition": { "robotType": "TurnTableSiemens" },
"HomePosition": {
"InternalValues": [ 0.0 ],
"ExternalValues": []
}
},
"Config": "{\"PlcRobotSettings\":{\"PlcId\":\"EEB39E39-8611-451E-A1B8-B75D7F2F98A9\",\"DeviceIndex\":1,\"InAxisFormat\":\"ns=3;s=\\\"Robot_Automation\\\".\\\"Turn_Table_{0}\\\".\\\"Axis_{1}\\\"\",\"StartMotionFormat\":\"ns=3;s=\\\"Robot_Automation\\\".\\\
},
{
"id": "3A64C6F4-4EF3-4DF2-BECD-96A5861E676",
JSON file
length : 6,149 lines : 110
Ln : 28 Col : 20 Pos : 698
Windows (CR LF)
UTF-8
INS

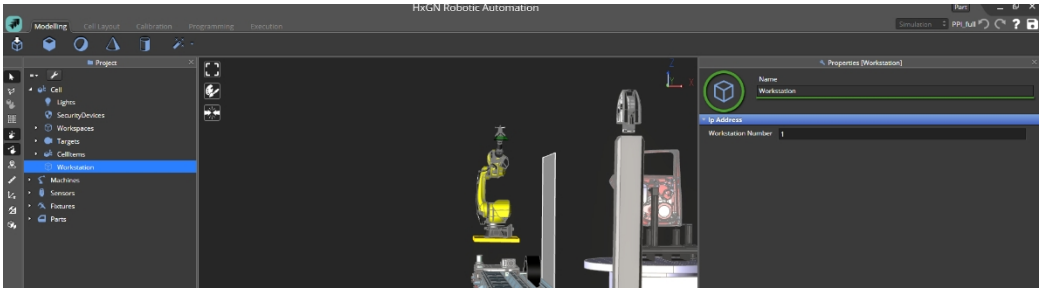

*new 2 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
"MachineIds": [
"14a9baba-6410-44cc-a90a-fa1f73babbff",
"9aafa265-c6dd-4194-abc6-495fbb1d1e9",
"dd0b8501-8043-4bd0-8864-ac262cc32e05",
],
"TrackerIds": [
]
}
],
"Robots": [
{
"id": "8917104E-1BA1-4CDB-88D9-2AD482B9E30",
"Name": "2M TurnTableSiemens",
"machine": {
"id": "c58a941d-f168-4d86-9be7-2bd9c3695fc6",
"machineDefinition": { "robotType": "TurnTableSiemens" },
"HomePosition": {
"InternalValues": [ 0.0 ],
"ExternalValues": []
}
},
"Config": "{\"PlcRobotSettings\":{\"PlcId\":\"EEB39E39-8611-451E-A1B8-B75D7F2F98A9\",\"DeviceIndex\":1,\"InAxisFormat\":\"ns=3;s=\\\"Robot_Automation\\\".\\\"Turn_Table_{0}\\\".\\\"Axis_{1}\\\"\",\"StartMotionFormat\":\"ns=3;s=\\\"Robot_Automation\\\".\\\
},
{
"id": "3A64C6F4-4EF3-4DF2-BECD-96A5861E676",
"Name": "FANUC M-20iD25L HOG BACH AB1",
"machine": {
"id": "9aafa265-c6dd-4194-abc6-495fbb1d1e9",
"machineDefinition": { "robotType": "FANUC" },
"HomePosition": { "internalValues": [ 90, -48, 0, 0, 18, 0 ], "externalValues": [ 1600 ] }
},
"robotIp": "127.0.0.1",
"listeningPort": 6000,
"Config": "{\"TimeOutSettings\":{\"MotionStartedTimeout\":7000,\"MotionCompletedTimeout\":60000,\"ExecuteCompletedTimeout\":300000,\"MotionMonitorInterval\":200,\"MotionDelayInterval\":1000},\"ToleranceSetting\":{\"IsCheckDestinationPosition\":false,\"Tol
},
{
"id": "D13C467A-6082-430F-AFCC-D02A3228826A",
"Name": "FANUC M-20iD25",
"machine": {
"id": "dd0b8501-8043-4bd0-8864-ac262cc32e05",
"machineDefinition": { "robotType": "FANUC" },
"HomePosition": { "internalValues": [ 90, -48, 0, 0, 18, 0 ], "externalValues": [ 3840.644 ] }
},
"robotIp": "127.0.0.1",
"listeningPort": 6005,
"Config": "{\"TimeOutSettings\":{\"MotionStartedTimeout\":7000,\"MotionCompletedTimeout\":60000,\"ExecuteCompletedTimeout\":300000,\"MotionMonitorInterval\":200,\"MotionDelayInterval\":1000},\"ToleranceSetting\":{\"IsCheckDestinationPosition\":false,\"Tol
}
,
{
"Id": "D289B480-3ED1-4AF0-90B0-3F9C21F80D99",
"Name": "GUD W TRACKER SLIDER",
"Machine": {
"Id": "14a9baba-6410-44cc-a90a-fa1f73babbff",
"MachineDefinition": {
"RobotType": "SliderPlc"
},
"HomePosition": { "internalValues": [ 0, 1545 ] }
},
"Config": "{\"PlcRobotSettings\":{\"PlcId\":\"EEB39E39-8611-451E-A1B8-B75D7F2F98A9\",\"DeviceIndex\":1,\"InAxisFormat\":\"ns=3;s=\\\"Robot_Automation\\\".\\\"Slider_Tracker_{0}\\\".\\\"Axis_{1}\\\"\",\"StartMotionFormat\":\"ns=3;s=\\\"Robot_Automation\\\"
}
]
}
JSON file
length : 6,149 lines : 110
Ln : 103 Col : 35 Pos : 4,791
Windows (CR LF)
UTF-8
INS

The Cell.json is structured with the following sections that can be modified as follows:

1. **Cell**": within "Name" the user can specify the cell name will be used when connecting to the cell from HxGN Robotic Automation, e.g. "Demo Cell" in the example above.
2. **Trackers**": within "ComputerName" the user must set the computer name controlling the tracker. E.g., "ESBAR-ROBOAUT-1" in the example above.
3. **Plcs**": here the user must make sure that PLC IP and Port within respectively "Url" and "Port" are correct and ensure that the "Id" is the same used in the "Id" tag of "Cell" section.
4. **Workstations**": this section define an subset of cell devices to be used for inspecting a part (e.g. only a rotary table is needed to measure a part loaded on it even if the complete cell consists of two or more rotary tables, see chapter **Single robot cell with multiple turn tables and stationary tracker**).

   To distinguish between different workstations, it is needed to define "Number" and "PlcNumber" within the "Workstations" section.

   1. "Number": it is the value defined in an HxGN Robotic Automation project to identify the workstation, e.g. "1" in the example below where the cell is made of a robot, a turn table, and a tracker.



```
"Workstations": [
  {
    "Id": "DFE060CD-871E-4634-8A90-F2E8310816AF",
    "Number": 2,
    "PlcNumber": 2,
    "CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
    "MachineIds": [
      "14a9baba-6410-44cc-a90a-fa1f73babbff",
      "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
      "dd0b8501-8043-4bd0-8864-ac262cc32e05",
    ],
    "TrackerIds": [
      "e04452le-43ca-4ca9-96af-b3c6594bf4c8",
      "86e71621-7dea-42a8-8444-38ff1e6ae51e"
    ]
  },
  {
    "Id": "DFE060CD-871E-4634-8A90-F2E8310816A0",
    "Number": 3,
    "PlcNumber": 2,
    "CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
    "MachineIds": [
      "14a9baba-6410-44cc-a90a-fa1f73babbff",
      "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
      "dd0b8501-8043-4bd0-8864-ac262cc32e05",
    ],
    "TrackerIds": [
    ]
  },
  {
    "Id": "BCD5888B-4F4D-4A5C-8674-B1F48C28F894",
    "Number": 1,
    "PlcNumber": 1,
    "CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
    "MachineIds": [
      "c58a941d-f168-4d86-9be7-2bd9c3695fc6",
      "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
    ],
    "TrackerIds": [
      "e04452le-43ca-4ca9-96af-b3c6594bf4c8",
    ]
  }
```


2. "PLC Number": it is the value defined in the PLC project to identify the entire workstation. In the PLC the workstation is composed of all the HW components but within HxGN Robotic Automation project it may be needed to use only a portion of them (e.g. for the same workstation

defined above, for robot calibration purposes HxGN Robotic Automation only needs to connect with the robot and the turn table but not the tracker, see picture below where the “TrackerIds” field is empty).

This differentiation between workstation defined both in HxGN Robotic Automation and PLC enables the user to create only one Cell.json file per cell which is configured to allow different workstation configurations in HxGN Robotic Automation, without the need to change the global workstation defined in the PLC, e.g. see the example below where four workstation configurations are defined in HxGN Robotic Automation (“Number”: 1, 2, 3, 4) using two PLC workstation (“PlcNumber”: 1, 2) defining four different (“Number”, “PlcNumber”) couples: (2, 2), (3, 2), (1, 1), (4, 1).

| | |
|---|---|
| ⚠ | **Warning:** “Number” values must be uniquely identified within the Cell.json file, while “PlcNumber” values can be common to different workstation “Number” values. |
| ⓘ | In case the cell has no PLC, the PLCNumber value has to be "null" |

```
"Workstations": [
  {
    "Id": "DFE060CD-871E-4634-8A90-F2E8310816AF",
    "Number": 2,
    "PlcNumber": 2,
    "CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
    "MachineIds": [
      "14a9baba-6410-44cc-a90a-fa1f73babbff",
      "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
      "dd0b8501-8043-4bd0-8864-ac262cc32e05",
    ],
    "TrackerIds": [
      "e04452le-43ca-4ca9-96af-b3c6594bf4c8",
      "86e71621-7dea-42a8-8444-38ffle6ae51e"
    ]
  },
  {
    "Id": "DFE060CD-871E-4634-8A90-F2E8310816A0",
    "Number": 3,
    "PlcNumber": 2,
    "CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
    "MachineIds": [
      "14a9baba-6410-44cc-a90a-fa1f73babbff",
      "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
      "dd0b8501-8043-4bd0-8864-ac262cc32e05",
    ],
    "TrackerIds": [
    ]
  },
  {
    "Id": "BCD5888B-4F4D-4A5C-8674-B1F48C28F894",
    "Number": 1,
    "PlcNumber": 1,
    "CellId": "EEB39E39-8611-451E-A1B8-B75D7F2F98A9",
    "MachineIds": [
      "c58a941d-f168-4d86-9be7-2bd9c3695fc6",
      "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
    ],
    "TrackerIds": [
      "e04452le-43ca-4ca9-96af-b3c6594bf4c8",
    ]
  }
```


5. **Robots**: in this section the user can define the different machines used in the cell. HxGN Robotic Automation treats all machines as kinematic structures, either they are industrial robots with six rotational axes (also mounted on a

linear slider) or simpler machines like rotary tables (one rotational axis) or sliders either with one or two linear axes, used to move the tracker in the cell.

The user can select the "**RobotType**" according to the robot brand used for industrial robots, or depending on the kinematic machine type, as described in the following table (currently only "FANUC" and "KUKA" types are supported as robot brands, "TURN_TABLE_SIEMENS" for rotary tables, and "SLIDER_PLC" for tracker sliders). It is possible to use both the string or the number as values (e.g. "FANUC" or "1"):

```
RobotType:
  ("VIRTUAL")] Virtual = 0,
  ("FANUC")] Fanuc = 1,
  ("KUKA")] Kuka = 2,
  ("ABB")] Abb = 3,
  ("STAUBLI")] Staubli = 4,
  ("COMAU")] Comau = 5,
  ("UNIVERSAL")] Universal = 6,
  ("TURN_TABLE_SIEMENS")] TurnTableSiemens = 7,
  ("FANUC_RG")] FanucRg = 8,
  ("OTHER")] Other = 9,
  ("SLIDER_PLC")] SliderPlc = 10,
```

If there is **more than one robot** in the cell, verify that inside "**Robots"** each one is configured to use a different port. The ports are configured, e.g. using FANUC robots, when the user installs the Karel OS in the robot controllers (see chapter **Hardware Setup**), and it must be the same as the value used in "listeningPort". E.g. in the "Cell.json" example below, one robot is using port 6008 and the other port 6009.

```
{
  "id": "3A64C6F4-4EF3-4DF2-BECD-96A5861E6576",
  "Name": "FANUC M-20iD12L HXG BACN AS1",
  "machine": {
    "id": "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
    "machineDefinition": { "robotType": "FANUC" },
    "HomePosition": { "internalValues": [ 90, -45, 0, 0, 15, 0 ], "externalValues": [ 1500 ] }
  },
  "robotIp": "127.0.0.1",
  "listeningPort": 6008,
  "Config": "{\"TimeOutSettings\":{\"MotionStartedTimeout\":7000,\"MotionCompletedTimeout\":60000,\"
},
{
  "id": "D13C467A-6082-438F-AFCC-D82A3228825A",
  "Name": "FANUC M-20iD25",
  "machine": {
    "id": "dd0b8501-8043-4bd0-8864-ac262cc32e05",
    "machineDefinition": { "robotType": "FANUC" },
    "HomePosition": { "internalValues": [ 90, -45, 0, 0, 15, 0 ], "externalValues": [ 3840.644 ] }
  },
  "robotIp": "127.0.0.1",
  "listeningPort": 6009,
  "Config": "{\"TimeOutSettings\":{\"MotionStartedTimeout\":7000,\"MotionCompletedTimeout\":60000,\"
}
```


In case of turn table or a tracker slider, the PLC variables in the “Config” settings need to match with the ones configured in the PLC and the “DeviceIndex” value need to be changed in case the cell has more than one tracker sliders, turn tables. In this case, the user needs to differentiate them by using a different number. This is not needed for robots as they are uniquely identified by the “Port” attribute.

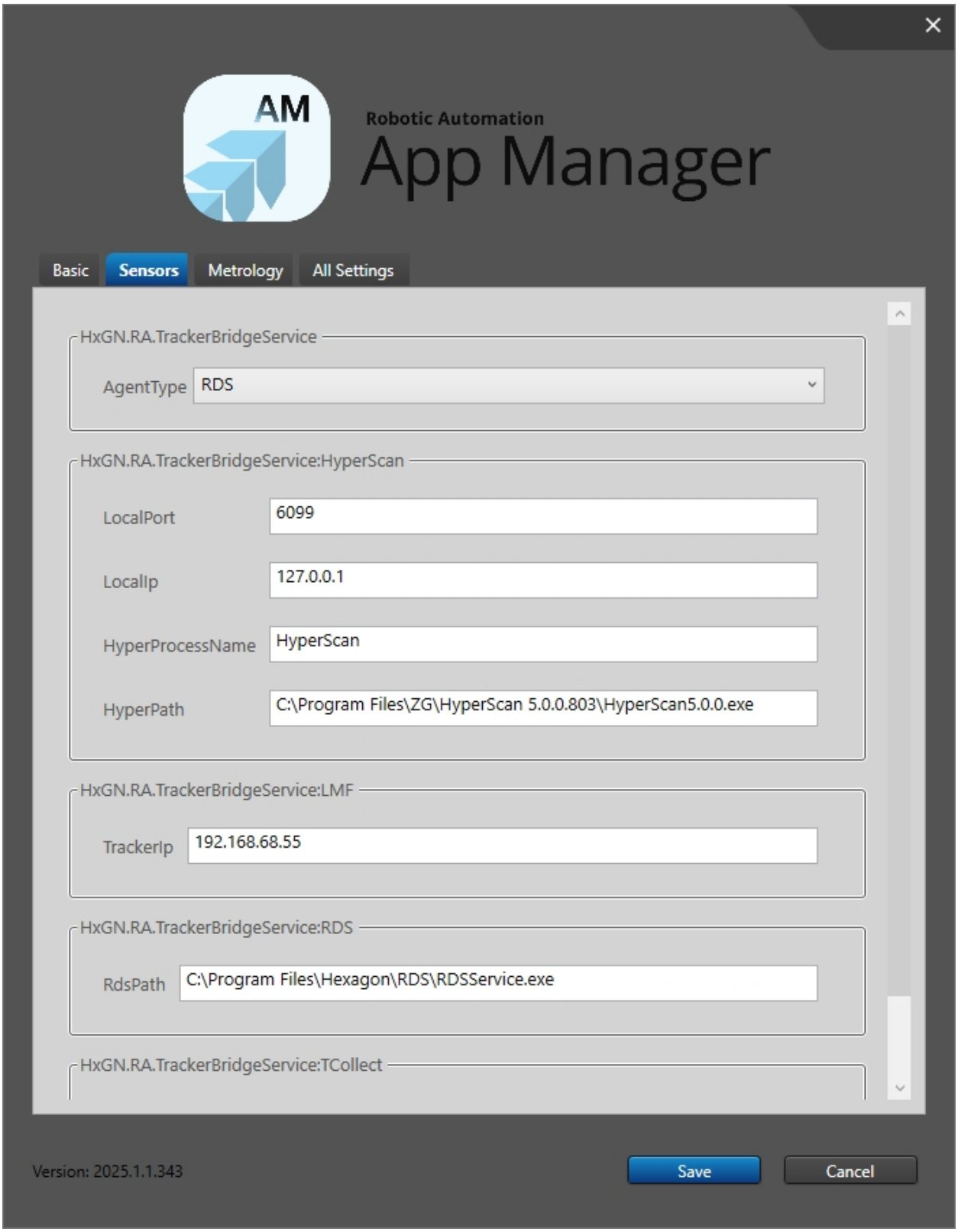

```
"Robots": [
  {
    "id": "8917184E-1BA1-4CDB-88D9-2AD4482B9E30",
    "Name": "2M TurnTableSiemens",
    "machine": {
      "id": "c58a941d-f168-4d86-9be7-2bd9c3695fc6",
      "machineDefinition": { "robotType": "TurnTableSiemens" },
      "HomePosition": {
        "InternalValues": [ 0.0 ],
        "ExternalValues": []
      }
    },
    "Config": "{\"PlcRobotSettings\":{\"PlcId\":\"EEB39E39-8611-451E-A1B8-B75D7F2F98A9\",\"DeviceIndex\":1,\"InAxisFormat\":\"ns=3;s=\\\"Robot_Automation\\\".\\\"Turn_Table_{0}
  },
  {


  {
    "Id": "D289B488-3ED1-4AF0-90B0-3F9C21F80D99",
    "Name": "GUD W TRACKER SLIDER",
    "Machine": {
      "Id": "14a9baba-6410-44cc-a90a-fa1f73babbff",
      "MachineDefinition": {
        "RobotType": "SliderPlc"
      },
      "HomePosition": { "internalValues": [ 0, 1549 ] }
    },
    "Config": "{\"PlcRobotSettings\":{\"PlcId\":\"EEB39E39-8611-451E-A1B8-B75D7F2F98A9\",\"DeviceIndex\":1,\"InAxisFormat\":\"ns=3;s=\\\"Robot_Automation\\\".\\\"Slider_Tracker_{0}\
  }
```


## Tracker bridge

Depending on the scanner that is used, please use the following settings:

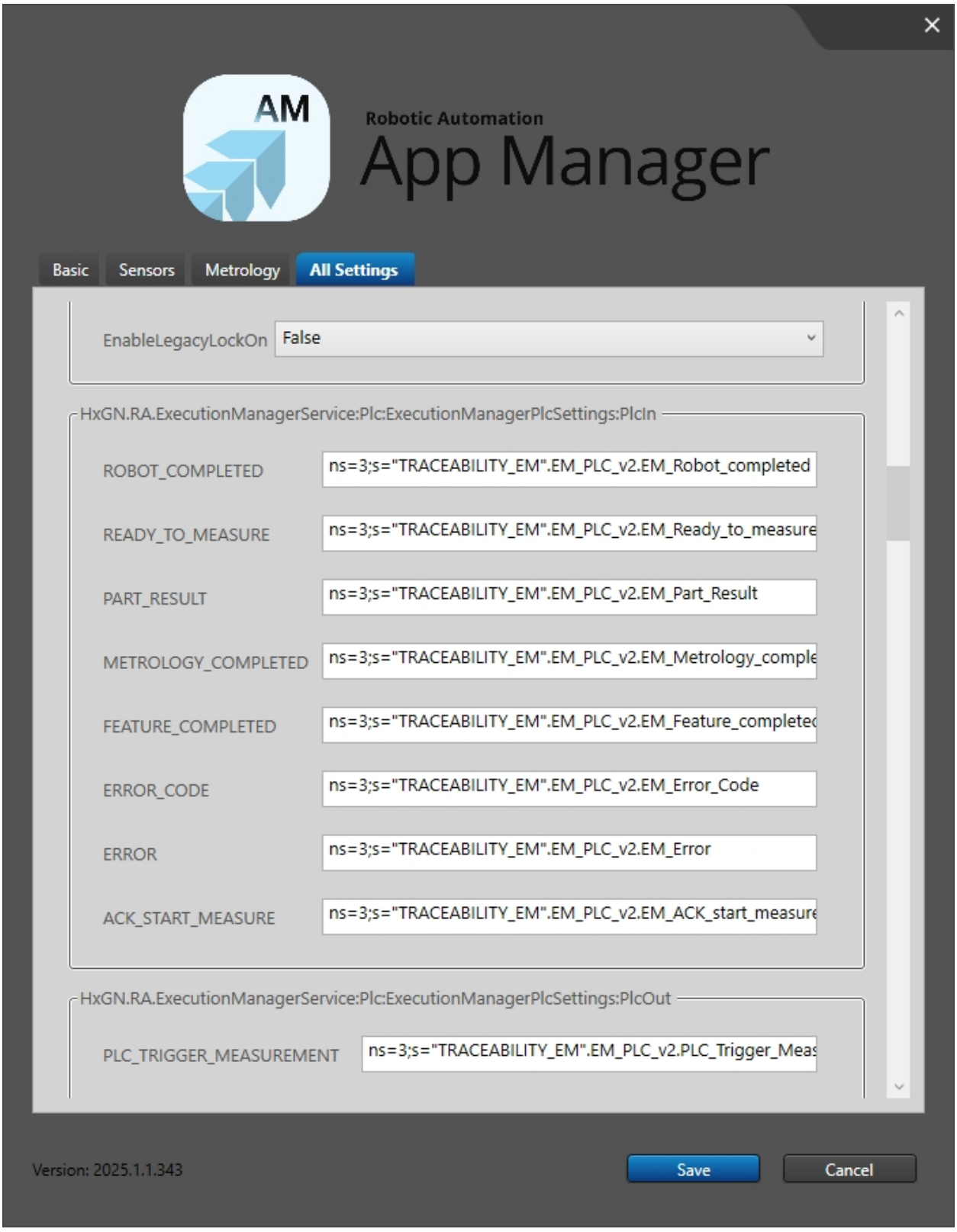
- "RDS" when AS1 or AS1 XL sensor is used.
- “TCollect" when T-Scan 5 sensor is used.
- "HyperScan" when ZG HyperScan sensor is used.
- "LMF" when ATS800 sensor is used

This change is done in the Sensor tab of the App Manager


AM
Robotic Automation
App Manager
Basic
Sensors
Metrology
All Settings
HxGN.RA.TrackerBridgeService
AgentType
RDS
HxGN.RA.TrackerBridgeService:HyperScan
LocalPort
6099
LocalIp
127.0.0.1
HyperProcessName
HyperScan
HyperPath
C:\Program Files\ZG\HyperScan 5.0.0.803\HyperScan5.0.0.exe
HxGN.RA.TrackerBridgeService:LMF
TrackerIp
192.168.68.55
HxGN.RA.TrackerBridgeService:RDS
RdsPath
C:\Program Files\Hexagon\RDS\RDSService.exe
HxGN.RA.TrackerBridgeService:TCollect
Version: 2025.1.1.343
Save
Cancel

## PLC Communication

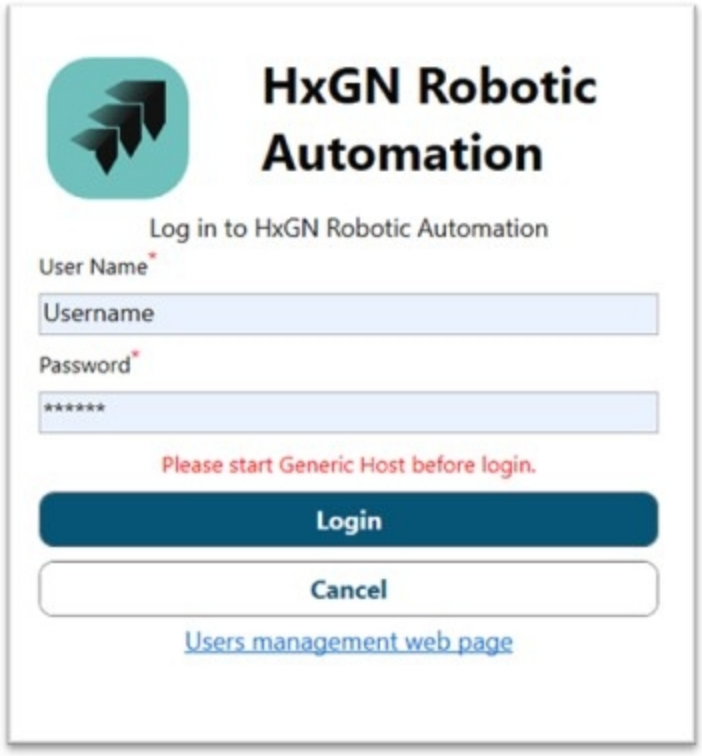
If Execution Manager is used in automatic or semiautomatic mode, the cell PLC must be configured according to the latest documentation about the PLC/EM workflow and contain a program that communicates the needed variables to the Execution Manager. Therefore the following variable definitions within the “Plc” tag need to be adjusted to comply with the defined workflow.


Robotic Automation
App Manager
Basic
Sensors
Metrology
All Settings
EnableLegacyLockOn
False
HxGN.RA.ExecutionManagerService:Plc:ExecutionManagerPlcSettings:PlcIn
ROBOT_COMPLETED
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_Robot_completed
READY_TO_MEASURE
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_Ready_to_measure
PART_RESULT
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_Part_Result
METROLOGY_COMPLETED
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_Metrology_comple
FEATURE_COMPLETED
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_Feature_completec
ERROR_CODE
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_Error_Code
ERROR
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_Error
ACK_START_MEASURE
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.EM_ACK_start_measure
HxGN.RA.ExecutionManagerService:Plc:ExecutionManagerPlcSettings:PlcOut
PLC_TRIGGER_MEASUREMENT
ns=3;s="TRACEABILITY_EM".EM_PLC_v2.PLC_Trigger_Meas
Version: 2025.1.1.343
Save
Cancel

## Configuration Troubleshooting

#### Generic Host is not running before opening HxGN Robotic Automation

If you open a project in HxGN Robotic Automation and get the following error:




It is needed to make sure that the Generic Host is running. For this, it is just needed to check Robotic Automation App Manager and check if the status is stopped, and in case start it again.

#### Generic Host crashes or fails to restart

If user wants to open Generic Host as a console application, it is needed to use the Robotic Automation App Manager to stop Generic Host running as a service, and then start it again as console application.

Another issue could be a wrong configuration in the Metrology settings of the App Manager

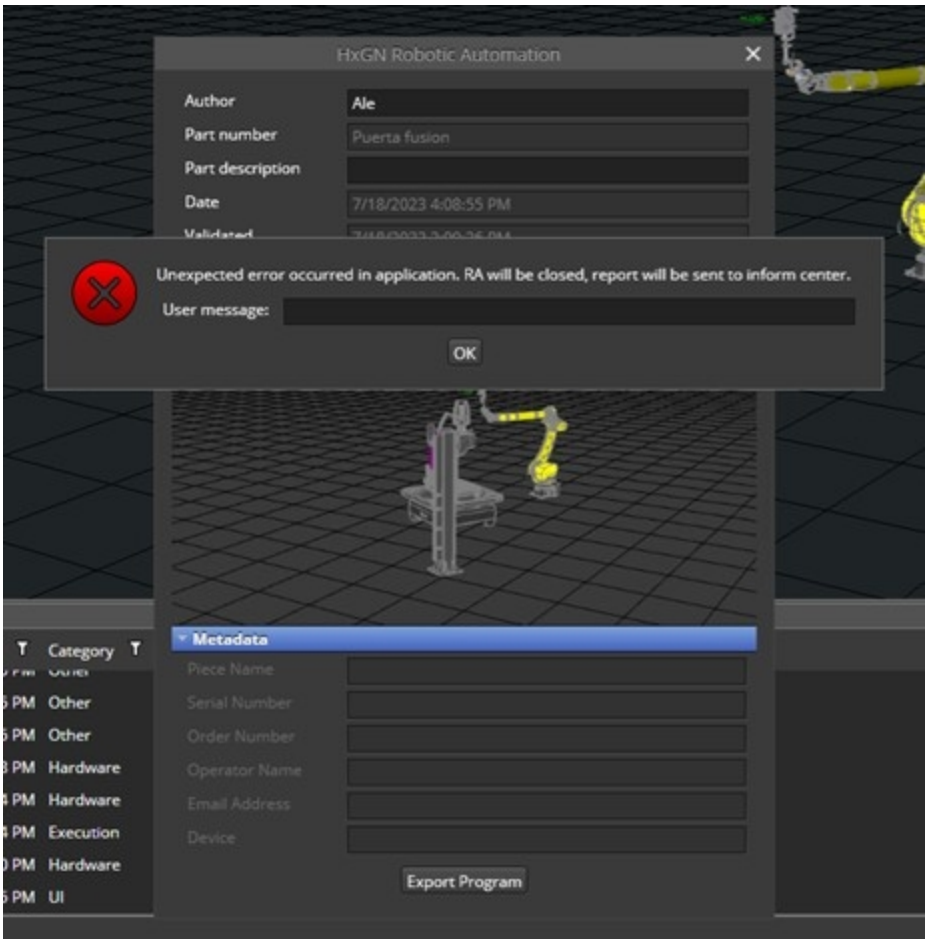
If all these steps did not help, review that all the settings in the App Manager.

#### Cell connection is not visible in HxGN Robotic Automation

1. Look at "*C:\ProgramData\Hexagon\HxGN.RA.GenericHost\CellConfigJson*" and make sure a well configured Cell.json file is saved in this path.
2. Restart Generic Host and HxGN Robotic Automation and try again.
3. Start GenericHost from the .exe application as administrator and check the logs to understand if there is any issue.

#### Database issues during program export

If user is facing issues when try to export program to database or open Execution Manager in browser. Make sure all configuration files in "C:\ProgramData\Hexagon" have been updated with the new ones in the installation folder within "C:\ProgramFiles\Hexagon." Some of these files were updated after the database implementation and using the old ones may cause connection problems. After updating the files do not forget to keep your local configuration. In the image below is shown one example of this issue:




## Splitting a cell into 2 smaller cells

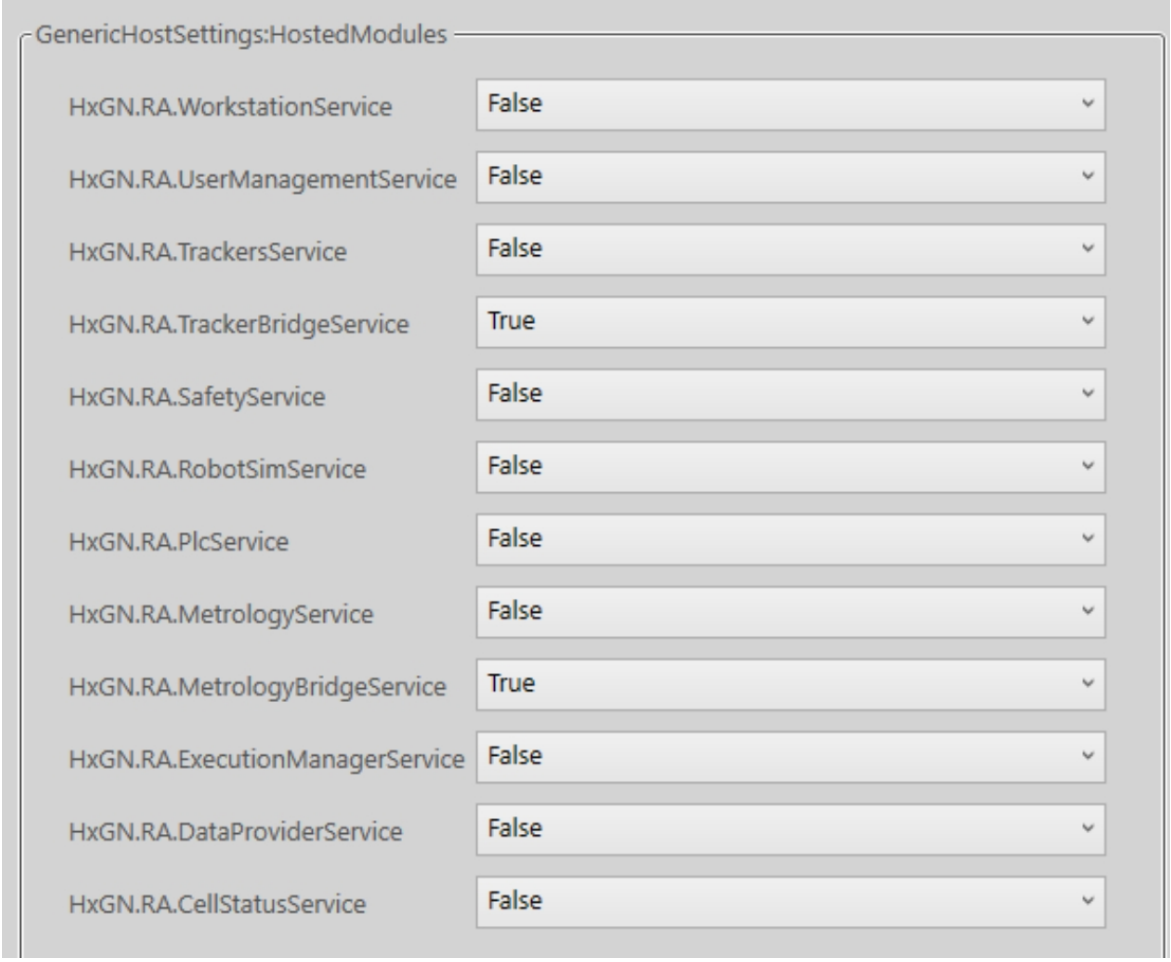
Splitting a cell into various individually working cells (aka workstations) requires a Metrology license for each individual cell. The exception is PolyWorks as it allows opening more than 1 project at the same time under 1 license. See: Work-stations and Metrology behavior

When splitting a dual robot, dual tracker and dual scanner cell into 2 cells the system will always have:

- A primary Generic Host
- A secondary Generic host

The primary GH can be setup as normal, there are no special changes. The secondary GH requires various changes:

1. The Metrology Service and the Metrology bridge Service to be enabled.



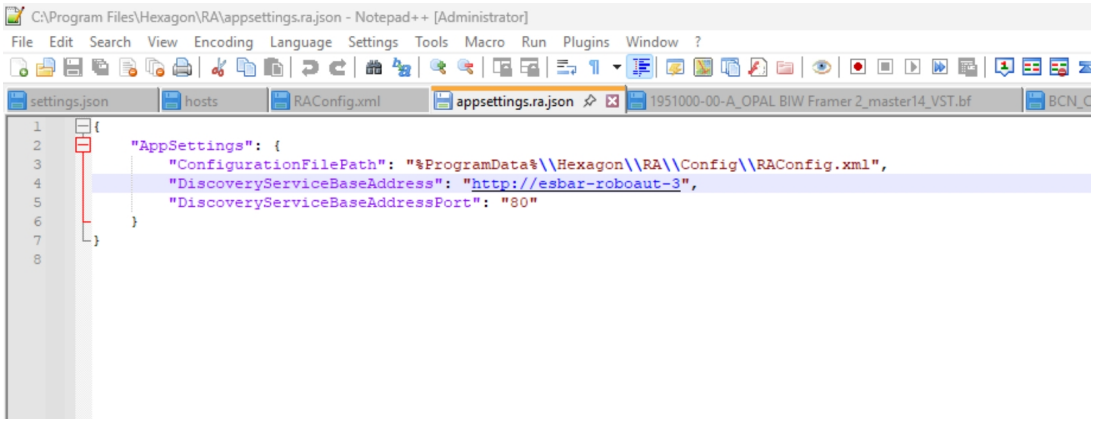
In case the Metrology bridge Service is not enabled on the secondary PC, this will cause the Metrology to be always opened on the PC of the primary GH and could cause issues in case you cannot open more than 1 Metrology on that PC.

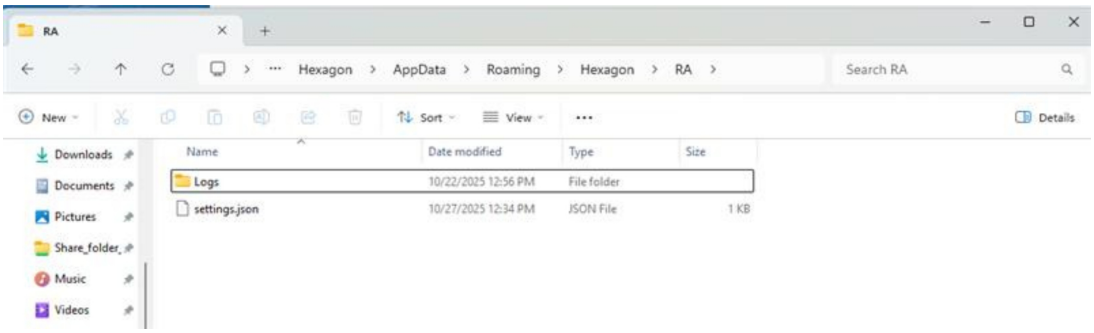
- PC-DMIS only allows **one** instance per computer
- Inspire only allows **one** instance per computer
- PolyWorks allows **two** instances per computer

2. Pointing the secondary GH to the primary.

- Update the ServiceAddress to point towards the primary GH

- In the Appsettings.ra.json update the DiscoveryServiceBaseAddress to the IP (or name) of the primary GH and paste it in the %APPDATA%/Hexagon/RA




> When opening Execution Manager in the secondary PC, rather than for instance "http://localhost/execmanager/" it will need to say:
> http://<IP or domain name of primary GH PC>/execmanager/

Setting up the cells afterwards can be done through the following steps: Splitting a cell into 2 workstations.

## Cell Hardware Setup

This section contains all the necessary information for setting up the communication with HW devices available within a robotic cell (Robots, PLCs, Tracker,

Scanners, external axes as Turn Table or Sliders).

## FANUC robot

Communication with Fanuc robot controller happens via TCP/IP socket messaging, with FANUC software versions see Robot Software Requirements

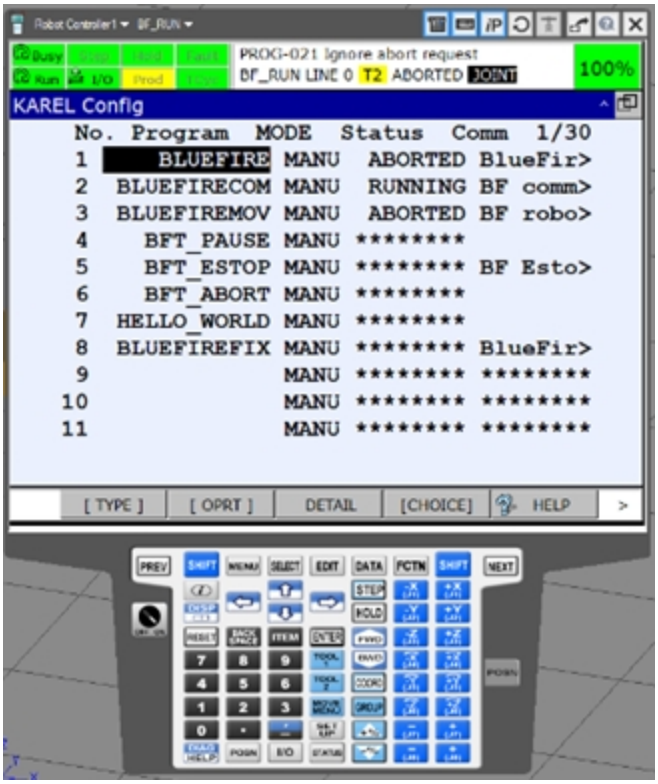
| Recommended Options | Minimim Options |
|---|---|
| R632 KAREL | R632 KAREL |
| R566 KAREL Diagnostic | R566 KAREL Diagnostic |
| R648 User Socket Msg | R648 User Socket Msg |
| H552 Handling Tool | |
| H521 English Dictionary | |
| R796 Ascii Program Loader | |
| R507 Ascii Upload | |
| R651 FRL Params | |
| J971 KAREL Use Sprt FCTN -> J742 (1) | |
| R641 PC Interface | |
| J624 Remote TCP | |

| Recommended Options | Minimim Options |
| --- | --- |
| R624 Tracking Jog | |
| FVRC Virtual Robot | |

Most of the options are available by default, some are combined, split, or renamed in different robot versions or configurations.

The final required option list should be verified with Fanuc support, depending on HW/SW versions, e.g., American, or European robot editions, etc.

This option can be used to easily troubleshoot any connection problems between the FANUC robot controller and HxGN Robotic Automation using the KAREL Config where the BLUEFIRECOM KL program can be easily stopped and executed again in case of any errors in the TCP/IP socket communication that cannot be managed otherwise.(1)

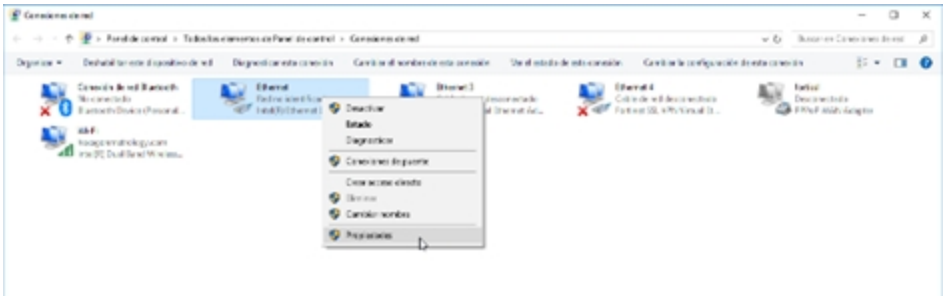


## Configuration

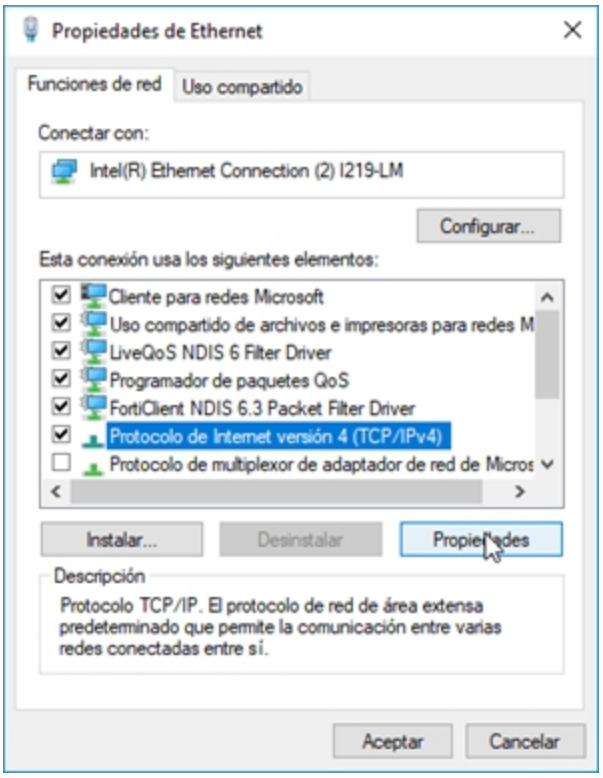
Robot and the PC must be configured in the same domain. Configure the robot TCP/IP following the next steps from the robot teach pendant:

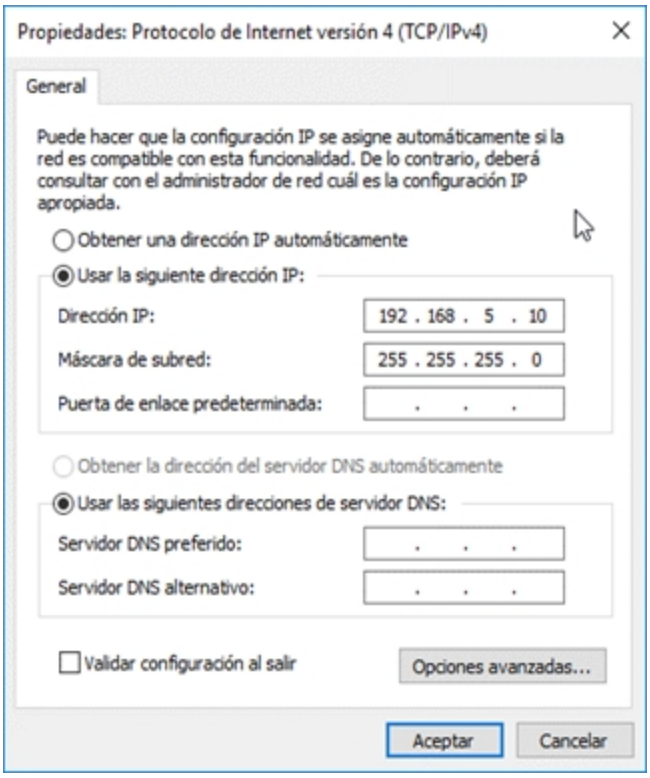
- Press Menu / SETUP (6) / NEXT (0) / Host Comm (9)
- Select TCP/IP (1) in the list and press DETAIL (F3).
- The TCP/IP address can be found on this screen.
- If you change it:
- Enter IP and Sub net mask.
- Click “Next.”
- Click “Init.”
- Restart Controller.

On the computer, open the network settings, right click on the ethernet network adaptor used to communicate with the Robot, and go to its properties:



Set the TCP/IP address according to the robot address: the computer must be on the same domain.

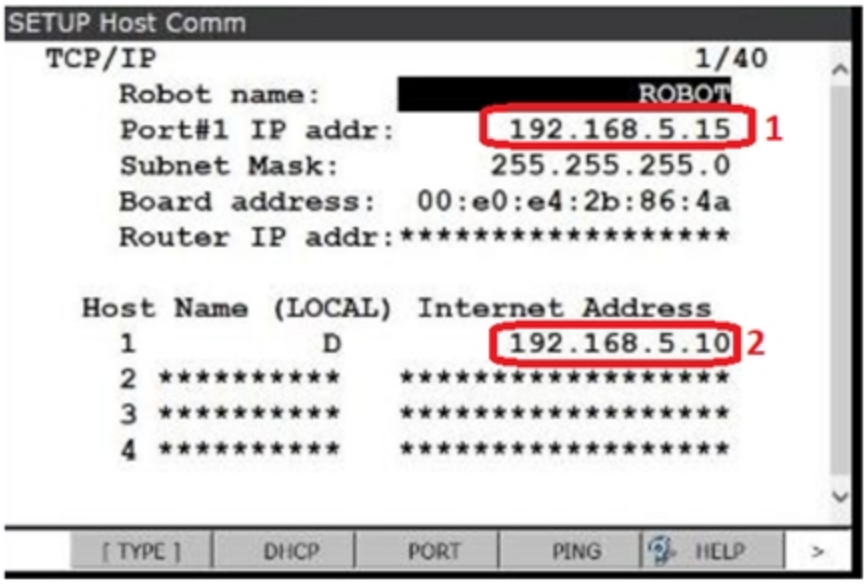

Propiedades de Ethernet
Funciones de red
Uso compartido
Conectar con:
Intel(R) Ethernet Connection (2) I219-LM
Configurar...
Esta conexión usa los siguientes elementos:
Cliente para redes Microsoft
Uso compartido de archivos e impresoras para redes M
LiveQoS NDIS 6 Filter Driver
Programador de paquetes QoS
FortiClient NDIS 6.3 Packet Filter Driver
Protocolo de Internet versión 4 (TCP/IPv4)
Protocolo de multiplexor de adaptador de red de Micros
Instalar...
Desinstalar
Propiedades
Descripción
Protocolo TCP/IP. El protocolo de red de área extensa predeterminado que permite la comunicación entre varias redes conectadas entre sí.
Aceptar
Cancelar




Follow the next steps in order check if the connection between the robot and the PC is established successfully:

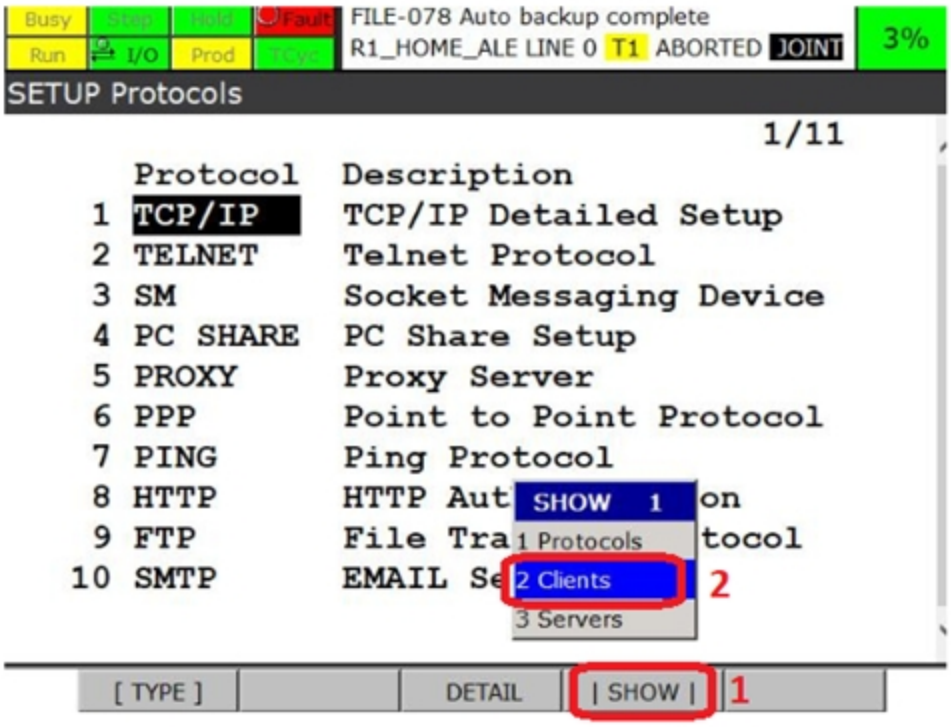
1. Ping on the Robot IP is succeeded (192.168.5.15). In case of more than one robot, then different port Ip address need to be configured for each one. For example, IP address = 192.168.5.15 for robot 1 and IP address = 192.168.5.20 for robot 2.

Ping on the PC IP is succeeded (192.168.5.10)

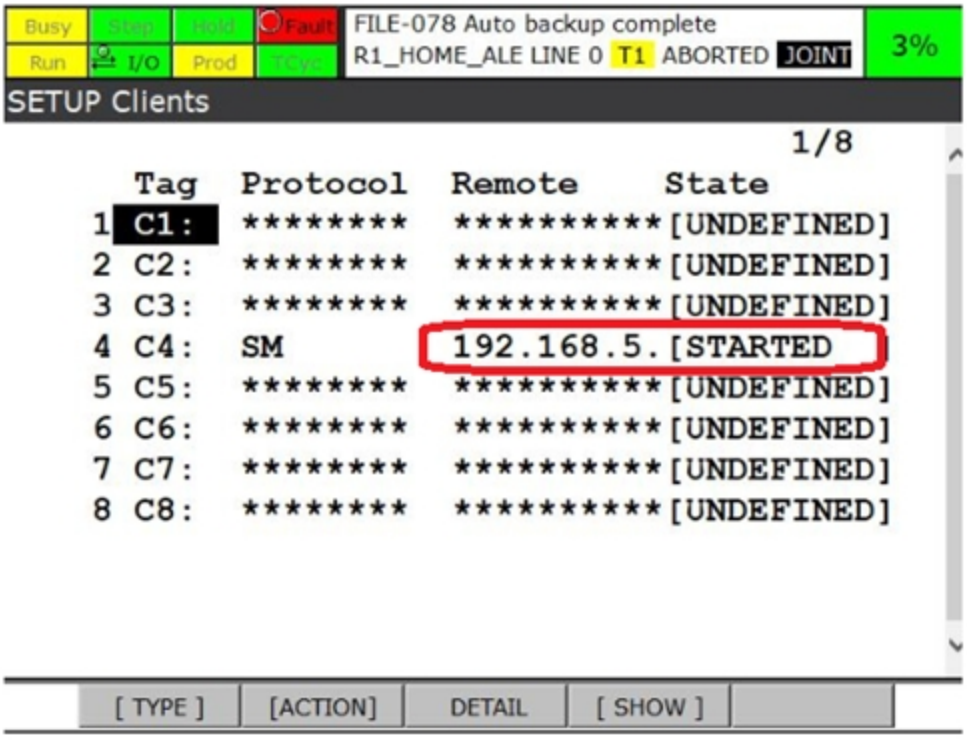
2. 


If the Ping on the PC fails, then check:

1. PC Firewalls
2. Try to change PC Connection Port
3. Check existing clients on the robot.




Make sure that a client with the IP of the computer exists and is started:




## Installation

To establish communication with Fanuc robot it is required to install the correct Karel scripts depending on the robot controller version.

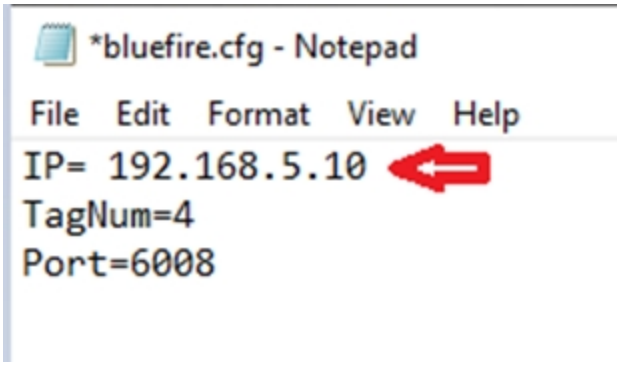
The latest scripts for every supported versions can be distributed upon request.

| | |
|---|---|
| ⚠ | Continuous Motion Group is available only from 2024_04_16 version. |

Follow the next steps:

1. Look at the robot controller version and ask HxGN Robotic Automation team for the right package.
2. Download the package and extract files to USB drive. Make sure that the files are directly copied on the USB root, not inside any folder.

Locate the **bluefire.cfg** file in the USB, **Modify and write the IP of the computer** that you are going to work with. Take also into account if cell has more than one robot (this is the case for dual robot cells), then different port needs to be configured for each one. For example, port = 6008 for robot 1 and port = 6009 for robot 2. Finally, this configuration must be the same in the cell.json configuration file (see chapter Cell Json Configuration).

3. 


4. In the “bluefire.cfg” check the default value for the following parameters and change it if it is needed:
    - UserTool=1
    - UserFrame=1
    - AutoExe=False
    - PwrNormal=BF_RUN
    - PwrSemi=BF_RUN
5. Change parameter **AutoExe=False to True** if you want to automatic run BF after restart the controller
6. Put USB to UD1
7. Install files to controller:

a. Click: Menu->7. File
b. Click: (F5) UTIL->1 Set Device->7 USB Disk (UD1:)
c. Select *.* (all files)->Enter.
d. Select BFUNINSTAL.CM->Enter.
e. Choice YES
f. Click: OK
g. Select BFINSTALL.CM->Enter.
h. Choice YES
i. You need see message “Execution is completed successfully.”
j. Click: YES
k. Installation script creates a log file in USB stick (INSTALL.LOG)

8. The bluefire.cfg file is copied automatically to the FR robot memory through the installation script.

   If for some reason, it needs to be done manually, please follow the next steps:

   a. Click: Menu->7. File
   b. Click: (F5) UTIL->1 Set Device->7 USB Disk (UD1:)
   c. cSelect *.* (all files)->Enter.
   d. Select BLUEFIRE.CFG->COPY_>CHOICE->FR:->DO_COPY
   e. Extract USB drive from UD1.

9. Log file is stored in FR:BFLOG.DG if has been previously enabled in Blue-fire.cfg.

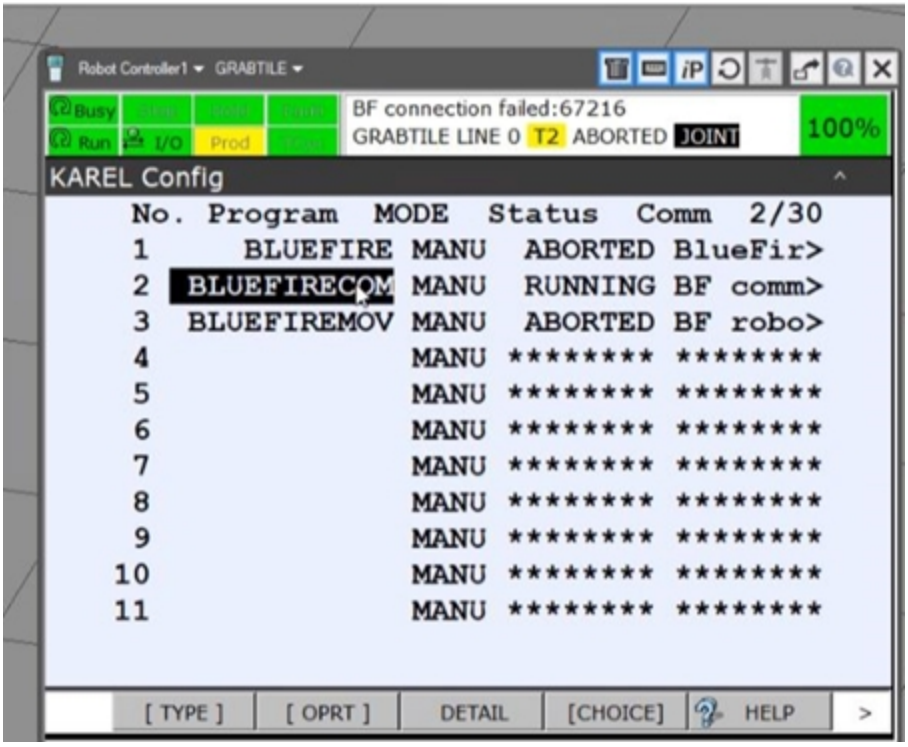
   You can show this file in browser http://robotHost/FR/BFLOG.DG

| | |
|---|---|
| ⚠ | Do not forget to extract the USB from UD1. Otherwise, it will cause delays in the robot reaction. |

### Execution

Run the Fanuc robot Karel following the next steps in Teach Pendant:

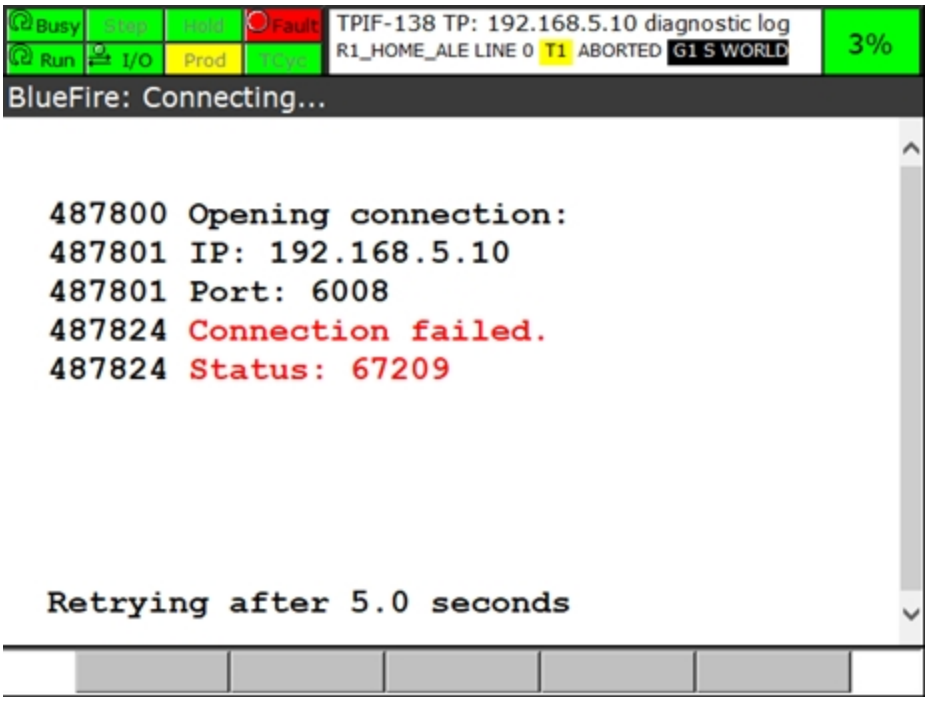
1. Click: Menu->6. Setup->Karel Config
2. Select: BLUEFIRE->F2->OPRT->RUN->YES
3. BLUEFIRECOM is running, BLUEFIRE is aborted.




After running the Blue Fire Karel in robot teach pendant user can consult the status of the Karel in:

1. Click: Menu->9. User-> Karel status and robot connection

   If for any reason robot is not connected properly to the computer, or the Generic Host is not running, the following message will appear on the robot: Connection Failed.

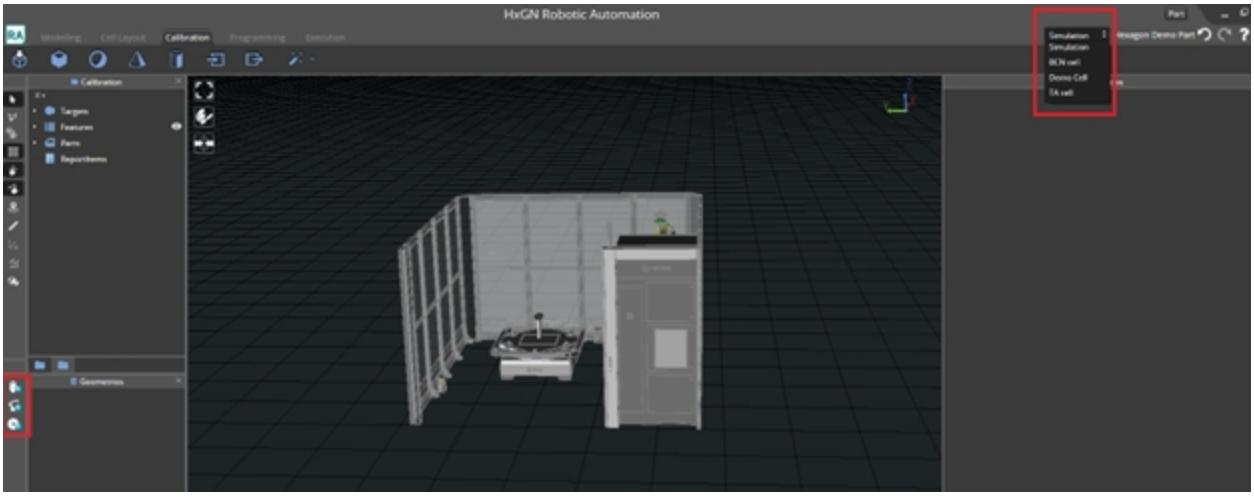
   a. 


   b. Otherwise, the connection will be successfully established as shown in the picture below.

Follow the next steps to establish the connection between Robot and HxGN Robotic Automation application:

1. Open template project or create a new empty project.
2. In case the robot is not included in the template, go to modelling mode, and import the right robot from your library.
3. Go to Calibration/Programming/Execution tab where the connection drop box is enabled and change from simulation to the configured cell.

4. Once you select the right cell if configuration is right then HxGN Robotic Automation is automatically connected with real Hardware.
5. The blue colour on the machines will turn to green after machines are connected and user will be able to move the robot from Execution tab.




## Troubleshooting

In case of failure, please provide the Fanuc controller backup.

Follow the next steps if it is not possible to connect:

1. Try to ping to the robot and make sure that all the cables are connected.
2. Try to restart the Karel programs.
3. Try to restart the robot.
4. Make sure that the computer firewalls are not affecting the connection.

In case more information is needed, there are two types of logs which can help to debug the problems:

- BlueFire.log
- BFLOG.DG

To enable robot logging to edit the bluefire.cfg. For that you will need to edit the file on the USB and perform uninstall and install again, the set up the next parameter as is shown below:

“IsLogEnabled=True”

Remove the "IsLogEnabled="True" parameter afterwards as it will delay the robot cycle time!

## KUKA robot

KUKA Robot requirement for new installation:

- KR C4 robot controller with TCP/IP connection to PC.
- KUKA.SystemSoftware operating system (see Robot Software)
- KUKA.EthernetKRL for data exchange via TCP/IP data communication (see Robot Software)

### Configuration

Obtain the latest Kuka software package

Continuous Motion Group is available only from 2024_04_16 version.

In the BlueFire.xml file go to line 5 to adjust the IP and to line 6 to adjust the Port to the address of your remote PC.

The address must be in the same subnet as the robot itself.
The default port is 49152

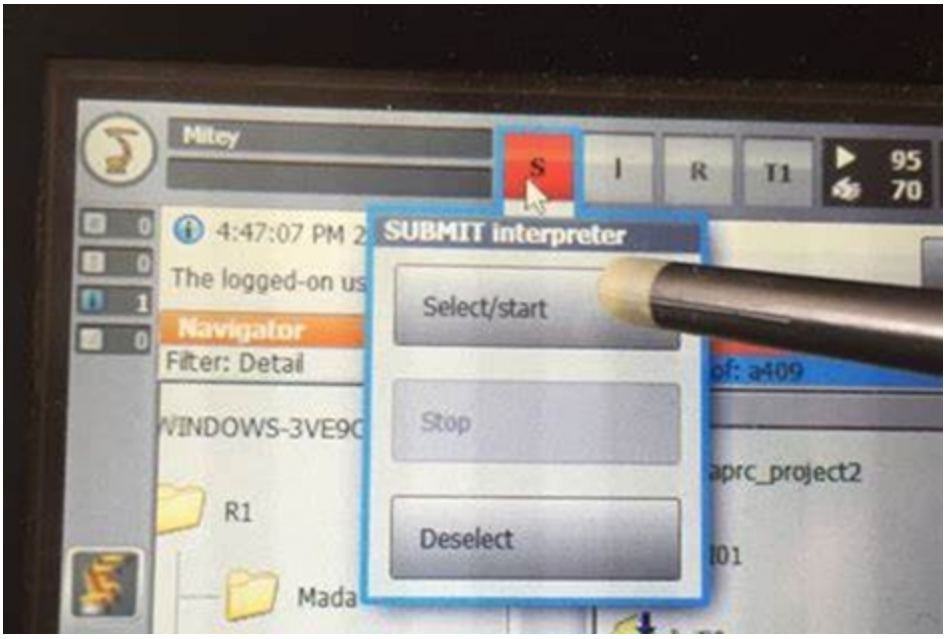
## Installation

1. Login as expert on the KUKA controller teach pendant.
2. Reduce the HMI and open the Windows Explorer
3. Copy the BlueFire.xml file into the folder C:\KRC\Roboter\Config\User\Common\EthernetKRL.
4. Copy the other files into a folder of your choice on the controller (KRC\R1\Program\BlueFire)

## Execution

1. Login as expert on the KUKA controller teach pendant.

   Stop and cancel the submit interpreter (Capital “S” on top)

2. 

3. Select and start bluefirecom.sub from the folder navigation menu on the left.
4. Select bluefire.src from the same folder.
5. Activate automation mode and start the motors.

6. Run the bluefire.src program.
7. Connect to the cell where the KUKA robot is installed in HxGN Robotic Automation to start the connection. For any issues, please connect with Kuka Diagnostic.

After replacing RJ45 cables, the controller might need a restart. KUKA controllers are sensitive to cable switches and might slow down the connection without a restart. If it is not connecting check if a firewall is active on the PC where HxGN Robotic Automation is running. This sometimes blocks the connection to the robot. If it is not connecting check if other software is active on the robot (e.g., Metrolog i-Robot), that might interfere with HxGN Robotic Automation.

### Troubleshooting

In case of failure, please provide the Kuka controller backup.

Follow the next steps if it is not possible to connect:

1. Try to ping to the robot and make sure that all the cables are connected.
2. Try to restart the Karel programs.
3. Try to restart the robot.
4. Make sure that the computer firewalls are not affecting the connection.

In case more information is needed, there are two types of logs which can help to debug the problems:

- BlueFire.log
- BFLOG.DG

To enable robot logging to edit the bluefire.cfg. For that you will need to edit the file on the USB and perform uninstall and install again, the set up the next parameter as is shown below:

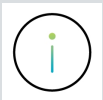
“IsLogEnabled=True”

Remove the "IsLogEnabled="True" parameter afterwards as it will delay the robot cycle time!

## ABB robot

### Configuration

- Get the latest RobotStudio 2025 installation package RobotStudio - Software and digital | Software and digital | ABB.
- Locate the Ethernet port at the bottom left corner on the front of the ABB robot controller and connect it to your computer using an Ethernet cable.
- Open Control Panel, go to Network and Internet, select Network and Sharing Center, click on Change adapter settings, and set the IP address of the Ethernet adapter connected to the ABB robot to be on the same subnet as the robot's IP.



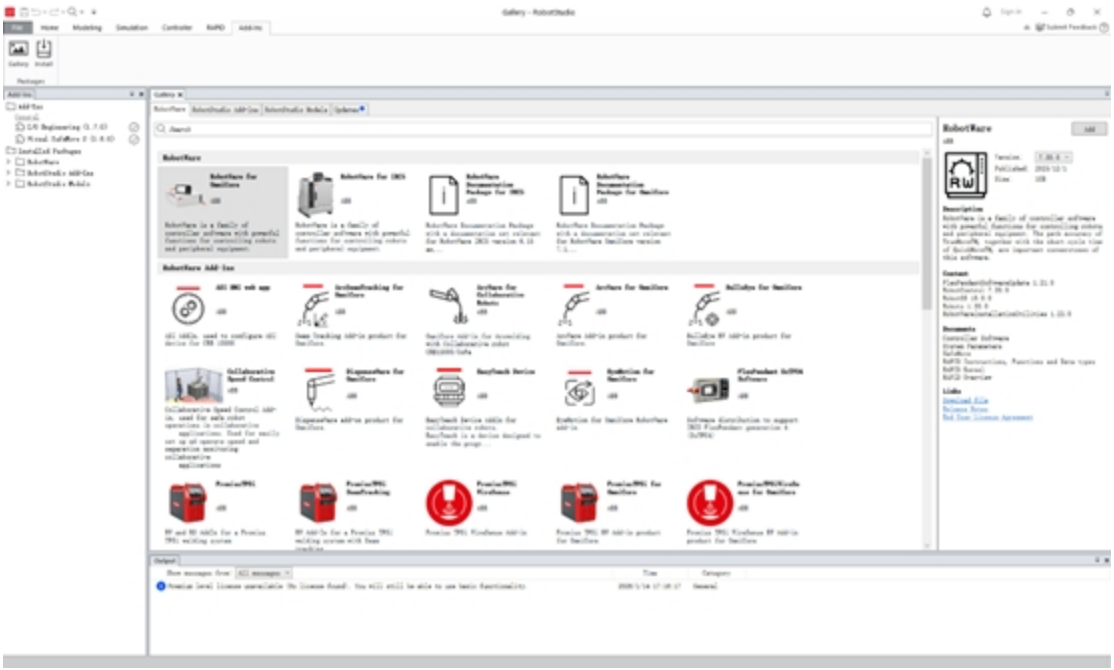
For example, if the robot's IP is 192.168.125.1, set the computer's Ethernet IP to 192.168.125.10.

### Installation

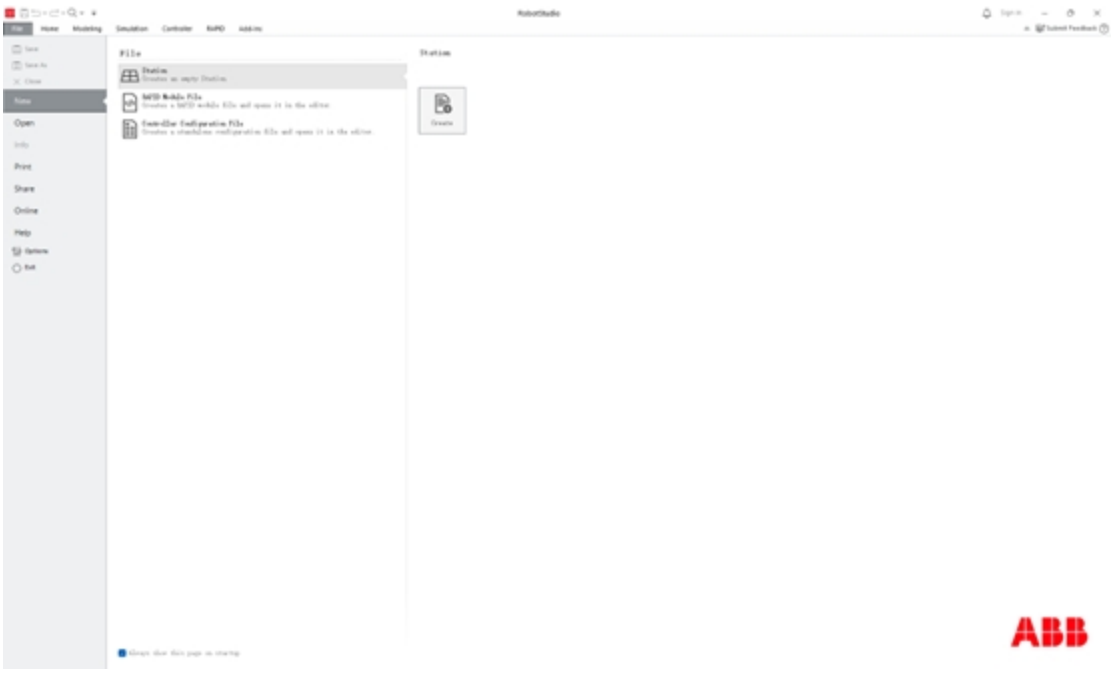
Open RobotStudio and create a new station.

## Execution

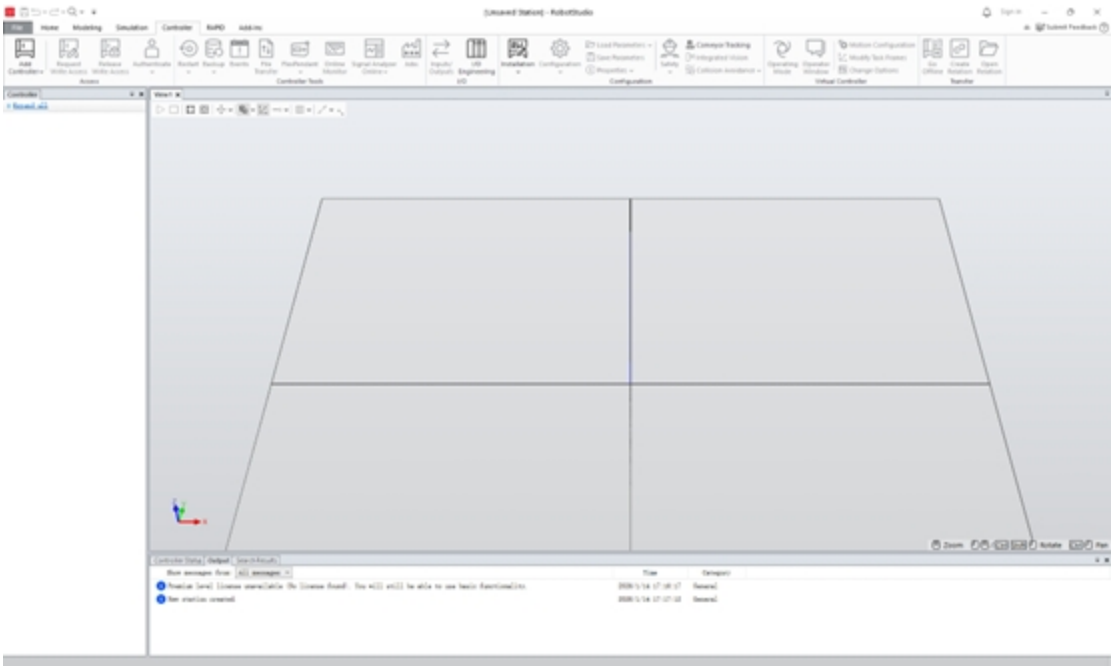
Click on the **Add-Ins** bar and download the required RobotWare 6.16 for the IRC5 controller, which is a file with the extension `.rspak`. Then, select **Install Package** from the menu bar, locate the `.rspak` file in your user directory, click **Open**, and the add-in will be installed.

1. 

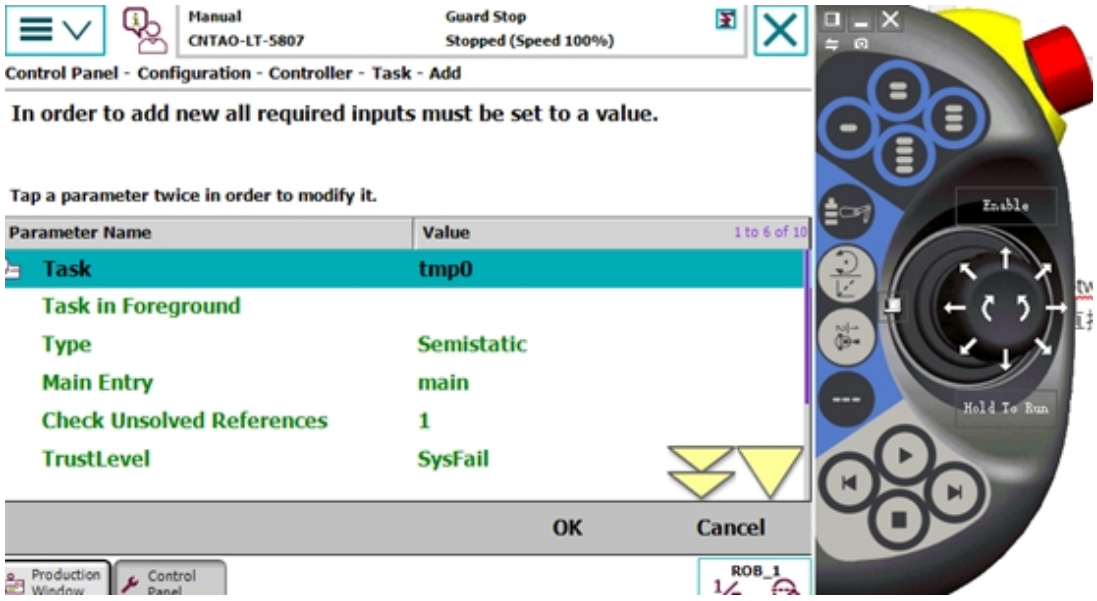
2. Create a new station.



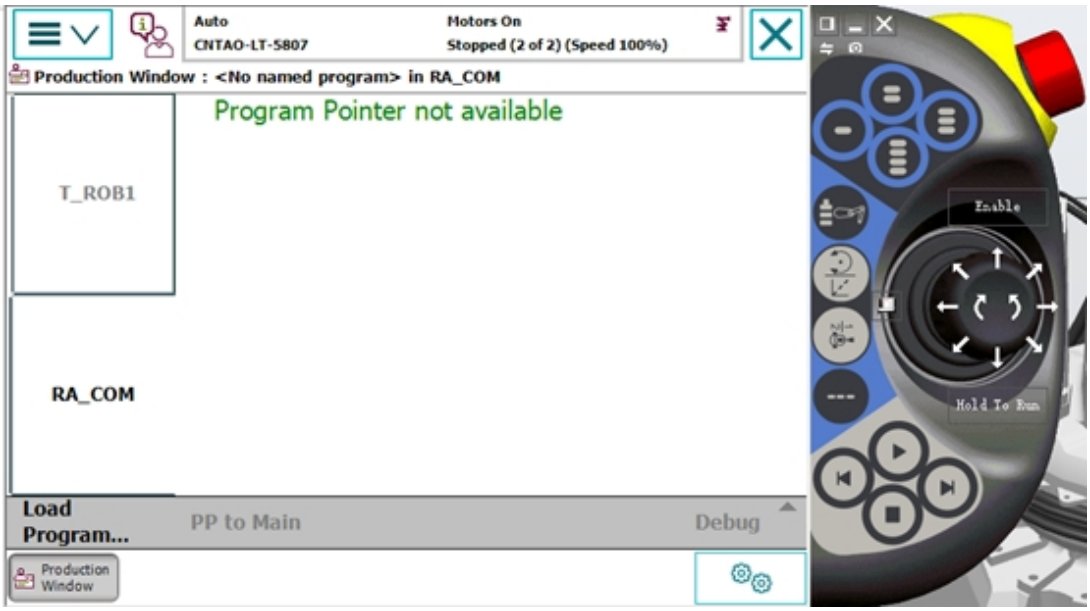
3. After entering the station, click on the controller, add a new controller, and then click <Connect to Controller>.



4. Enter the IP address of the controller to connect to.
    a. then click OK to establish the connection.
    b. Before connecting, locate the Ethernet port at the bottom left corner on the front of the ABB robot controller and connect it to your computer using an Ethernet cable.
    c. Then, open Control Panel, go to Change adapter settings, and set the IP address of that connection to be on the same subnet as the robot.
5. After connecting, turn the switch to power on the robot, and use the control cabinet knob to set it to manual mode.
    a. Then, click the button in the upper left corner of the teach pendant screen, and sequentially click: Control Panel -> Configuration -> Topics -> Controller -> Task.
    b. Click Add to enter the following interface.



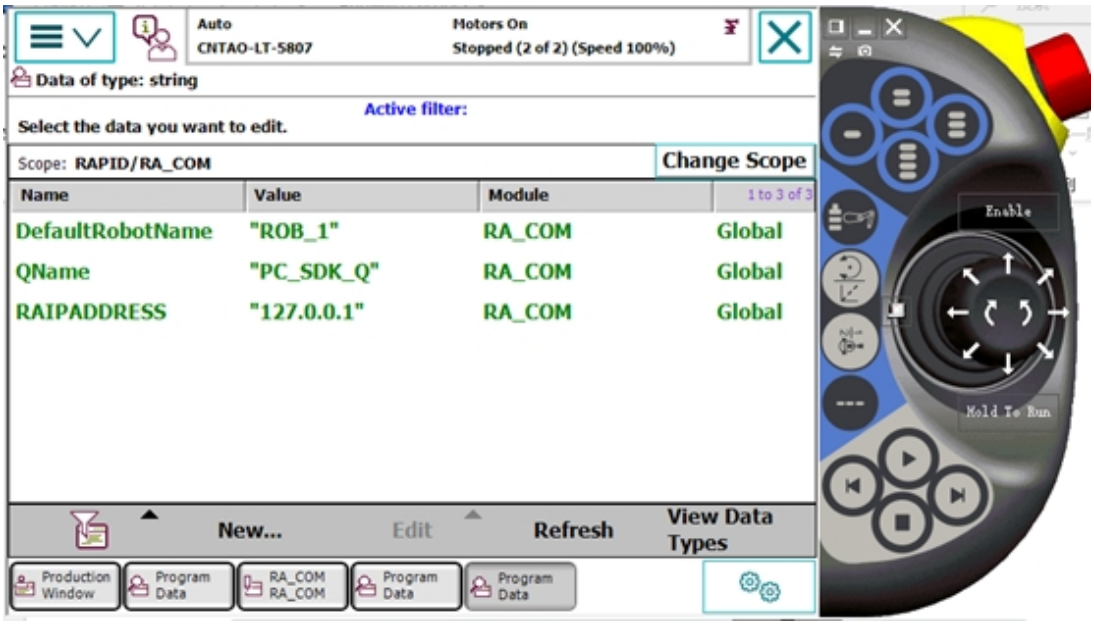
6. Change the name of the Task to RA_COM (must be in all uppercase English letters), set the type to normal, change the RMQ type to remote.
    a. Click OK to restart.
    b. After restarting, the interface will appear as shown in the following image:




7. Next, in the RobotStudio software, click on the Controller section.
    a. In the tree structure on the left, right-click on **T_ROB1** and **RA_COM** respectively.
    b. Select **Load Program**, and load the corresponding program files provided by the RA team.
    c. Restart the controller.
8. After restarting, you can see that the controller displays the program content of **T_ROB1** and **RA_COM**.

a. Click the button in the upper left corner, then click in order: **Program data** -> **string** ->**change scope**, and select **RA_COM**.

b. The interface will then appear as follows:




9. Select `rapid address` and enter the IP address of the host computer.

   a. Similarly, under the `num` data type in program data, select `RAPORT` and enter the port number.

   b. Save the settings and restart.

10. After restarting, when connecting to RA, click on the production window, select both the `T_ROB1` and `RA_COM` programs.

    a. Click `pp to main`.

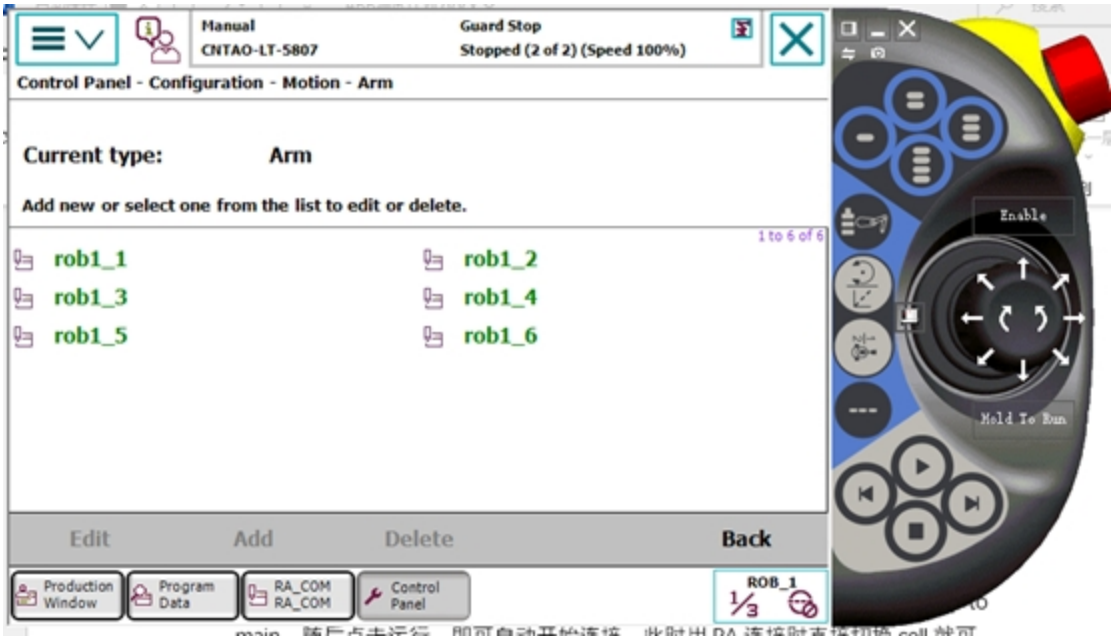
    b. Click the run button to automatically initiate the connection.

    c. In the Programming module of RA, click the simulation toggle selection box in the upper right corner to select the corresponding cell and connect to the real robot.

## Troubleshooting

Encountering a limit mismatch error in RA

If you need to adjust the robot's limits, on the teach pendant, click: **Upper left corner -> Configuration -> Topics -> Motion -> Arm**. The following interface will appear, where you can click to enter and modify the limit values for each axis.



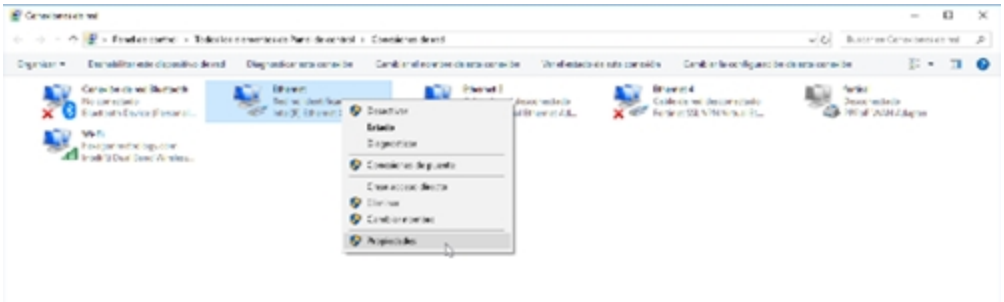

## PLC

Turn table and Tracker Slider are connected to the PLC and HxGN Robotic Automation communicates with them through the PLC. The configuration of the variables to communicate in a proper way are defined in the Cell.json file by default and user do not need to do any other operation.

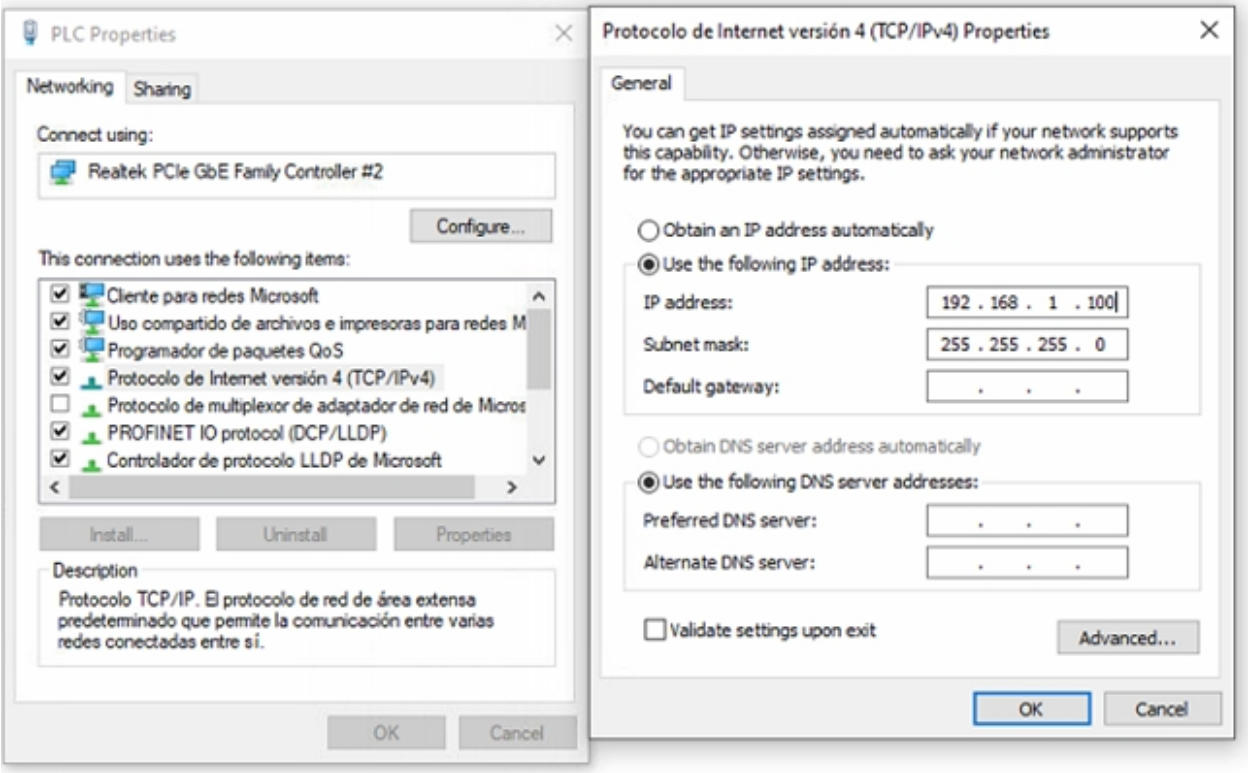
## Configuration

Enter the IP address of the computer on the same domain of the PLC.

Open the network settings, right click on the ethernet network that you will work with, and go to its properties:



Set the TCP/IP address according to the PLC address: the computer must be on the same domain.




Try to ping to the PLC typing the IP on the web browser to check if the connection between the PLC and the computer succeeds. The following picture will appear.

Make sure that you also configure the same PLC IP in the Cell.json file.

For example, if the PLC IP Address is 192.168.1.160, in cell.json file should be like below:

```
"PlcSettings": {
  "Url": "opc.tcp://192.168.1.160",
```

### Execution

Follow the next steps to establish the connection between Turntable and Tracker Slider with Robotic Automation application:

1. Open a template project or create a new empty project.
2. In case the Turntable or Tracker Slider are not included in the template, go to modelling mode, and import the right machine from your library.
3. Go to Calibration/Programming/Execution tab where the connection drop box is enabled and change from simulation to the configured cell.
4. Once you select the right cell if configuration is right then HxGN Robotic Automation is automatically connected with the real Hardware.

The blue colour on the machines will turn to green after machines are connected and users should be able to move machines from Execution tab.

### Troubleshooting

It is possible to access the Siemens diagnostic page to check for any issues in the PLC.

## Trackers and Scanners

Make certain the trackerbridge settings are configured correctly as described in Tracker Bridge

### Leica Scanners

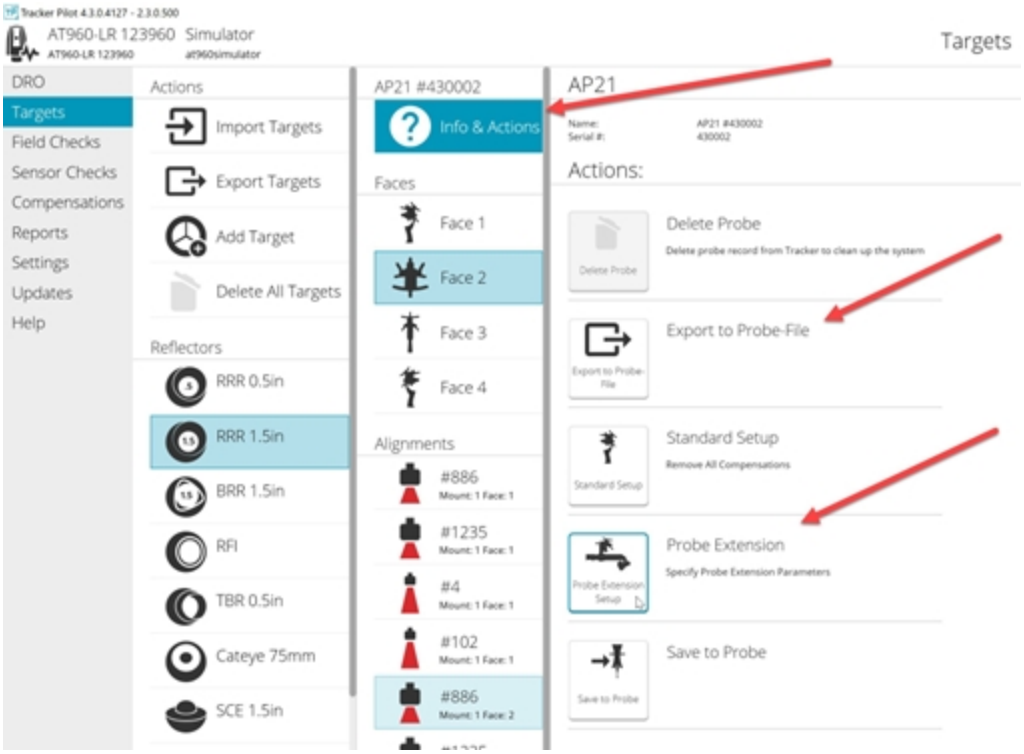
Please refer to the Leica documentation on how to set up the connection with AT960 tracker and either T-Scan 5, AS1 (also using the scanning extension), and AS1 XL scanners.

#### AS1 Scanning extension

Once the extension is added in the RA template project, you need to follow Leica documentation to mount and connect the extension on the robot and perform the following steps:

1. Open Tracker Pilot as advanced user.
2. Select the target (AS1) and navigate to the info&actions tab.
3. Export the current probe file to backup your current configuration and alignment in case you need to recover it.
4. Click on Probe Extension and select the right extension and rotation.

Now it is mandatory to re-calibrate the new configuration using the proper sphere and workflow.




### ZG Scanners

Please refer to the ZG documentation on how to set up the connection with tracker and scanners.

## TCP and External Axis Configuration

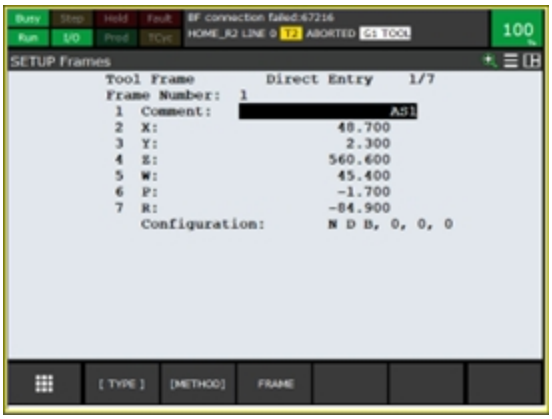
### How to define the TCP of the sensor on the Fanuc Robot

In the Robot Teach Pendant, MENU – 6 SETUP – 5 Frames, select the tool frame number 1 and click on DETAIL.

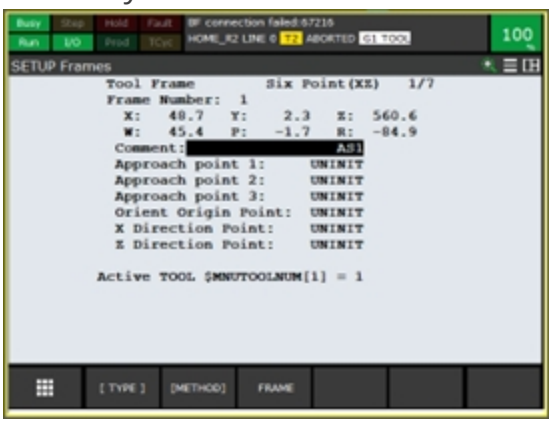
Then, in the new window click on METHOD button where all the possible methods to define the TCP are shown.

There are two different options that can be considered to correctly define the TCP for the sensor:

1. DIRECT ENTRY. In this case, it is the user who just needs to copy the corresponding values from Robotic Automation. This information will be found in the project tree, under the Robot, tool properties.




SIX POINT. Considering the sensor as a complex tool (its Z axis is not parallel to the Z axis of the face plate), user will need to follow the Six Point method to manually define the TCP. Recommended option is to use the Six Point (XZ).

2. 


For more information about how to record each one of the points check the Fanuc manual.

## How to ensure that the TCP is defined properly on the Fanuc robot

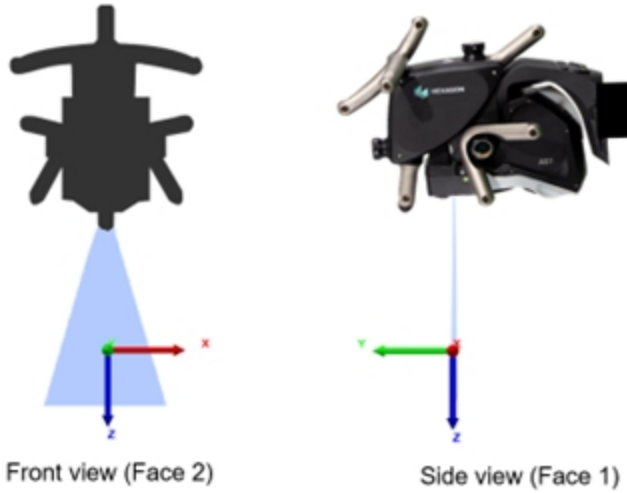
Each tool mounted on the robot must be configured to the supplier's specifications.

#### Leica AS1 TCP

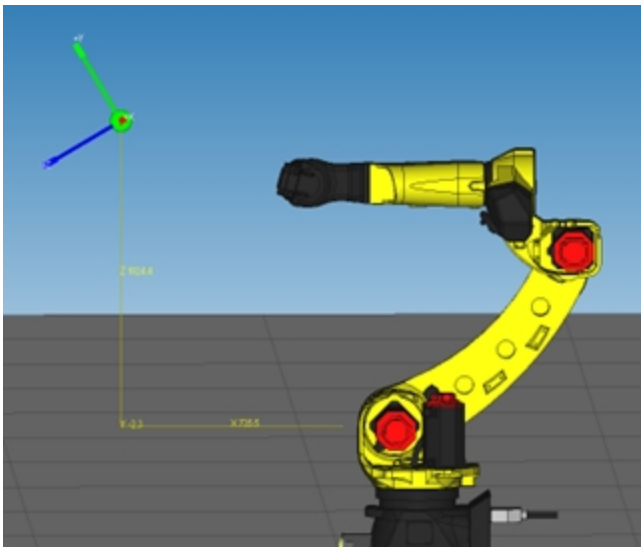
Following the robot tool definition from Leica used for the AS1, it should be considered the next two points:

1. TCP is defined in the middle of the scan line and working range.

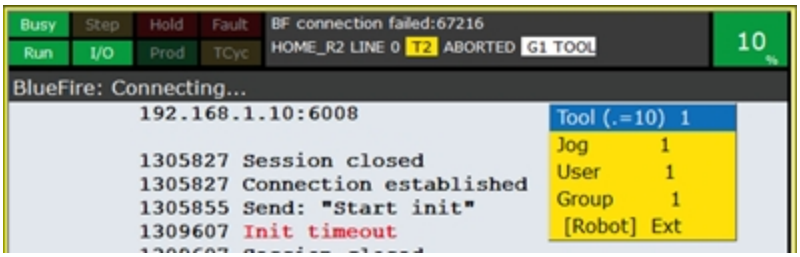
   The following axis direction:

2. 


**Case 1. Fanuc Robot is fixed**



First, it is needed to make sure that the right tool frame is selected (1 by default), and the robot is in Tool coordinates before moving.




Then move the robot in X, Y and Z axis and check if the orientations are correct. Finally, rotate X, Y and Z and TCP should be static while robot is rotating in each of the axis around the TCP.

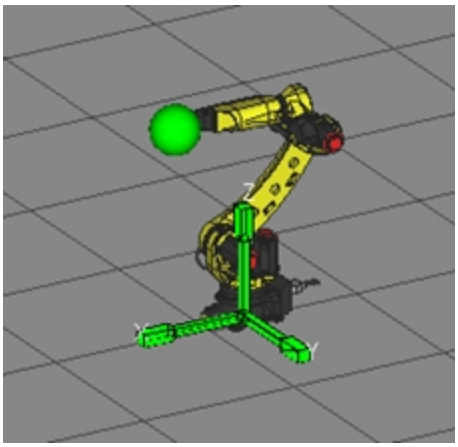
**Case 2. Fanuc Robot mounted on slider or 7th axis.**

In these cases, it is especially important to configure the 7th axis properly. If not, the TCP will have a non-desire behavior.

1. 7th Axis Configuration

   It is recommended to use Roboguide to facilitate the configuration of the 7th axis.

   a. Identify in which axis of the robot we should define the 7th axis. To do this, simply upload our backup robot to Roboguide and display the robot's coordinates by clicking on its base. If the user does not have Roboguide license, consider that the x axis is always the side where cables are plugged in the robot.



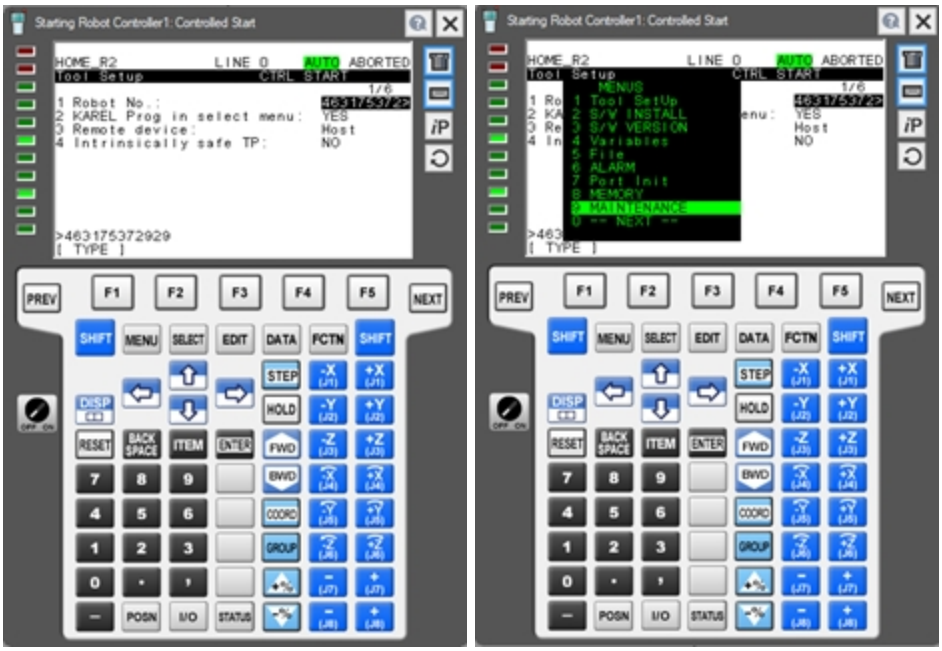
b. Move the robot to 0º in all joints.

c. Identify in which robot axis the slider is defined.

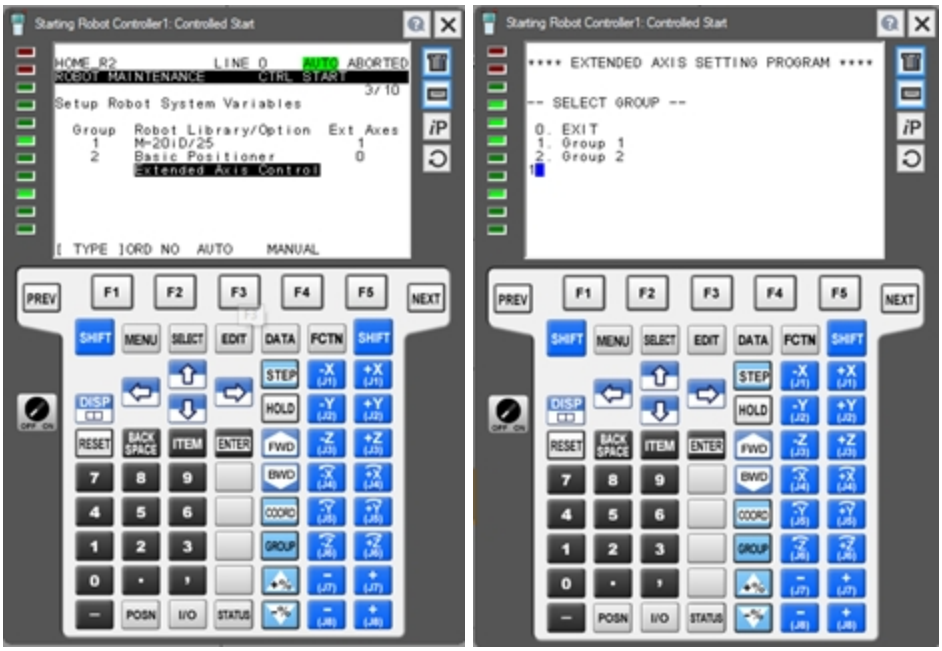
Once done, it is required to do a cycle power using CTRL starting mode.

d.

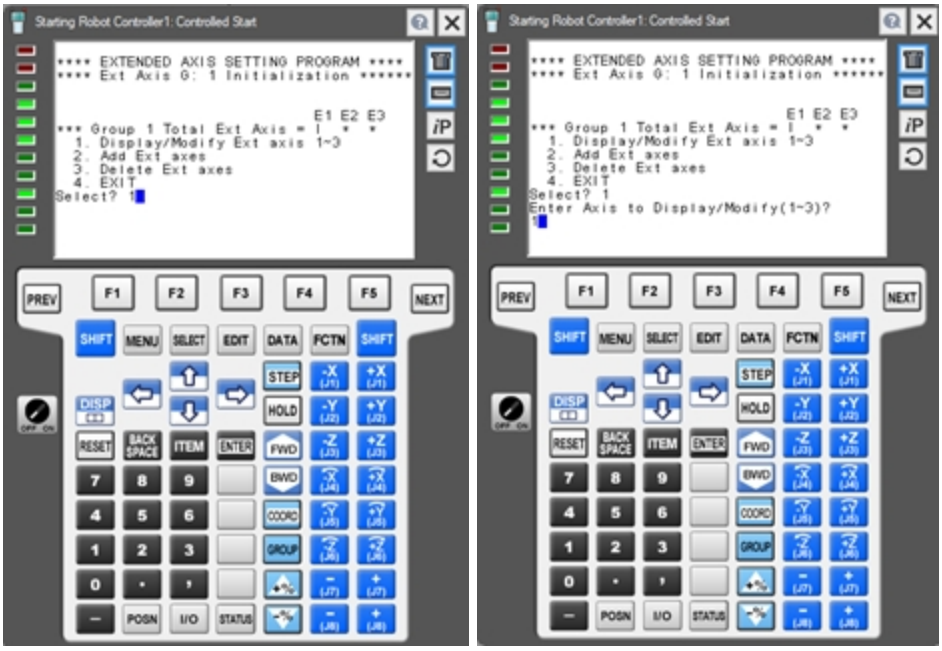

e. In the main screen, click on the MENU button and select the MAINTENANCE option.




f. Select using the cursor “Extended Axis Control” and press F4 (MANUAL). Then continue selecting Group 1.

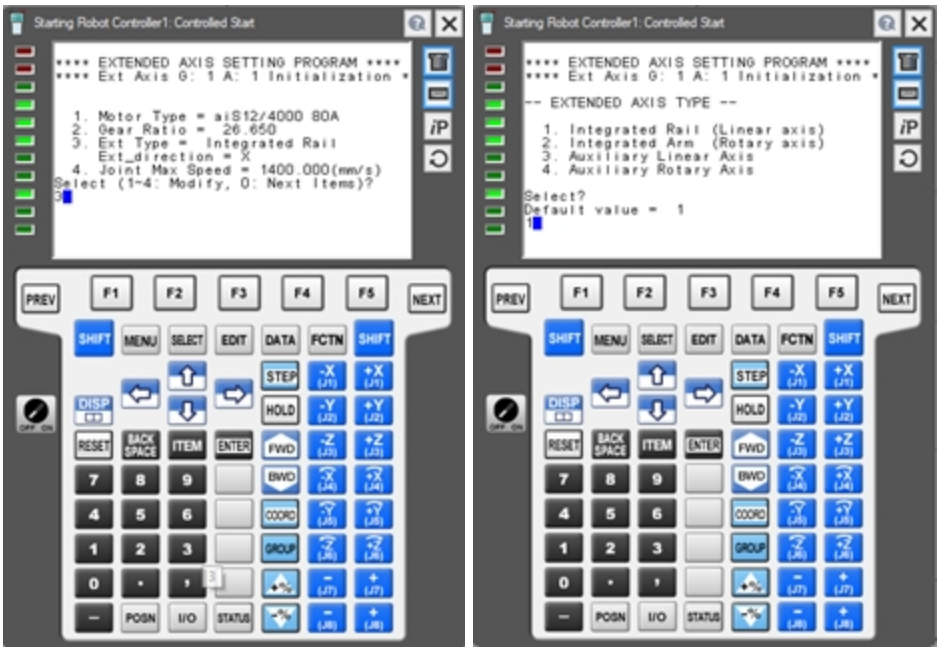


g. Select first option to Display/Modify Ext Axis and press enter to continue, then enter the axis to display/Modify (it will be 1 in this case)




h. The Gear Ratio value can be easily verified by moving the robot on the slider 1 meter and checking that the distance is correct with a tape measure.

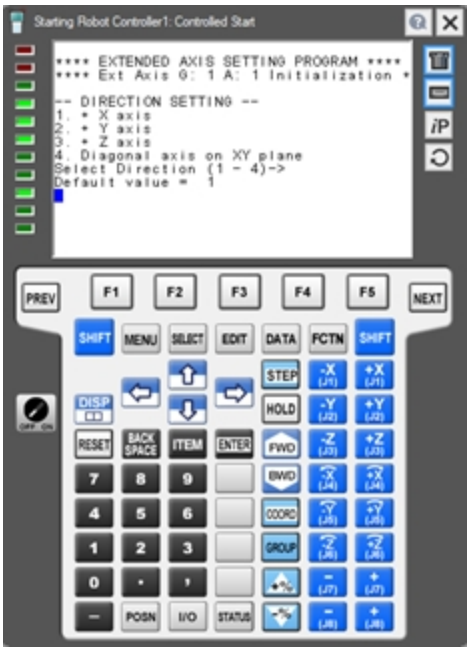
i. For Ext Type option, make sure is configured as **Integrated Rail (Linear axis)**, that is mean option “1” need to be selected.



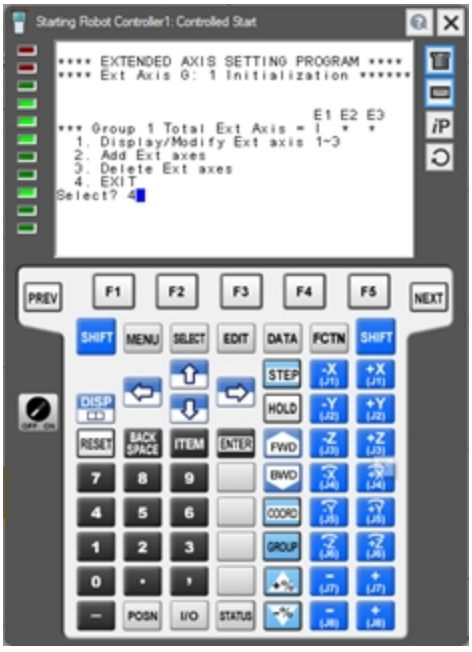

j. Next step is to define in which direction is configured the extended axis. Move the robot to 0 degrees in all joints, then considered that X axis is the one where the cables are plugged, figure out in which axis is moving the robot on the slider (X, Y or Z).

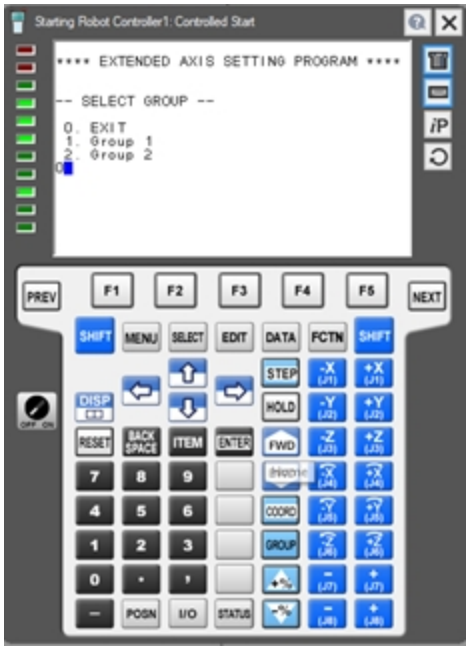
k. Select the value for the right axis and press enter.

   Depending on the placement of the gear motor the user may have to change this configuration later.




I. Select EXIT options to move forward until the main screen.



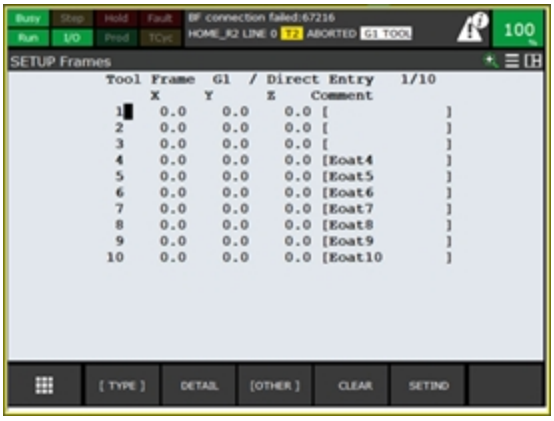
m. In the main screen, press FCTN button and select 1START (COLD). Robot will be rebooted, and all the changes applied.

n. Usually after changing the configuration, it will be required to apply DCS changes and cycle power the robot to be able to use the robot in automatic mode.

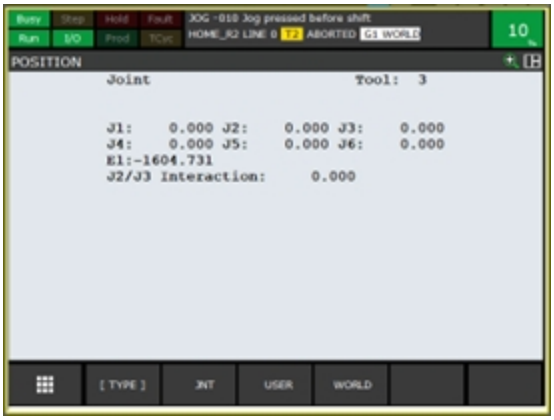
2. 7th Axis Check

   Once 7th axis is configured, it's needed to check if TCP is working as expected. For thar purpose:

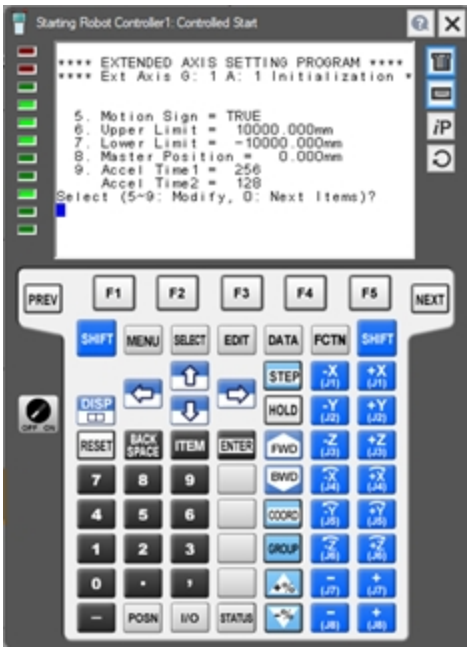
Select an empty tool frame.

a. 


Move the robot to 0 degrees in all joints and select JOINT as coordinates. Move robot in +/- J7. Make sure robot is moving in the expected direction and that E1 value on robot Teach pendant is also the expected one.

b. 


If result is the opposite of the expected there is need to configure again the 7th axis, go to the option number 5 called Motion Sign (as shown in the image below) and change value from TRUE to FALSE or vice versa.



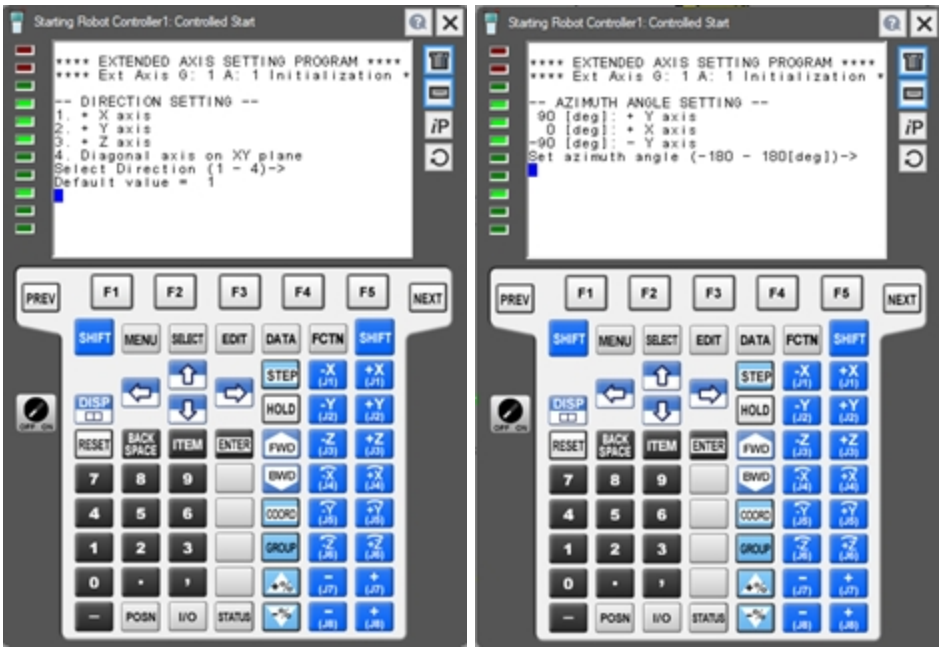

c. Move the robot to 0 degrees in all joints and select WORLD as coordinates. Move robot in +/- J7 and the expected movement is that all joints will be moved sliding on the slider to keep the TCP static.

   If this is not the case, and TCP is not static user should consider two options:

   i. Change Motion Sign as explained in the previous step.

   ii. Change Extended Axis Settings. If user noticed that instead of the expected movement the TCP seems to be moved like in a diagonal trajectory try the following configuration. Select option 4 in the extended axis setting menu. Try the different options (90, -90 degrees) until one of them meets the expected movement of the

TCP.


Starting Robot Controller1: Controlled Start
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization
-- DIRECTION SETTING --
1: + X axis
2: + Y axis
3: + Z axis
4: Diagonal axis on XY plane
Select Direction (1 - 4)->
Default value = 1
Starting Robot Controller1: Controlled Start
**** EXTENDED AXIS SETTING PROGRAM ****
**** Ext Axis G: 1 A: 1 Initialization
-- AZIMUTH ANGLE SETTING --
90 [deg]: + Y axis
0 [deg]: + X axis
-90 [deg]: - Y axis
Set azimuth angle (-180 - 180[deg])->
PREV
F1
F2
F3
F4
F5
NEXT
SHIFT
MENU
SELECT
EDIT
DATA
FCTN
STEP
HOLD
DISP
RESET
BACK SPACE
ITEM
ENTER
FWD
BWD
COORD
GROUP
POSN
I/O
STATUS

# UI Overview

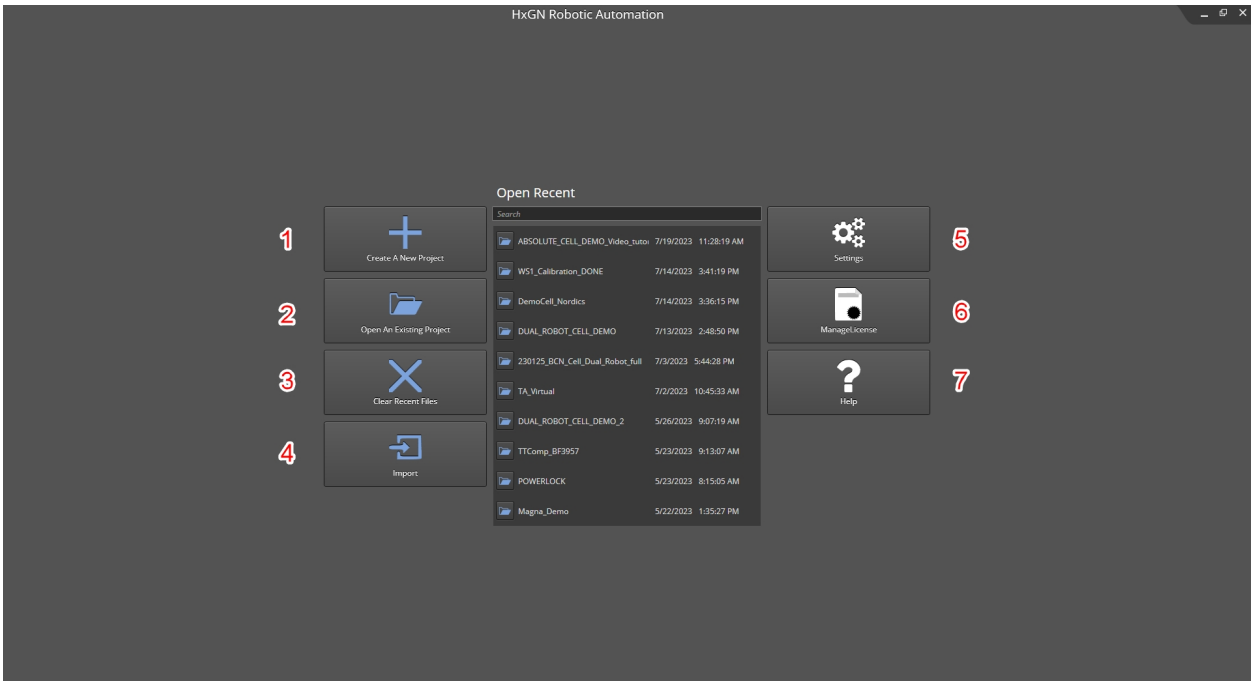
## HxGN Robotic Automation

After launching the HxGN Robotic Automation application, the user can click on:

1. **Create A New Project** to start building a new project from scratch

   > ⚠ **Warning**: the project name must not have any "." characters (e.g., "MyProject" is a valid name but "MyProject.2023" is not).

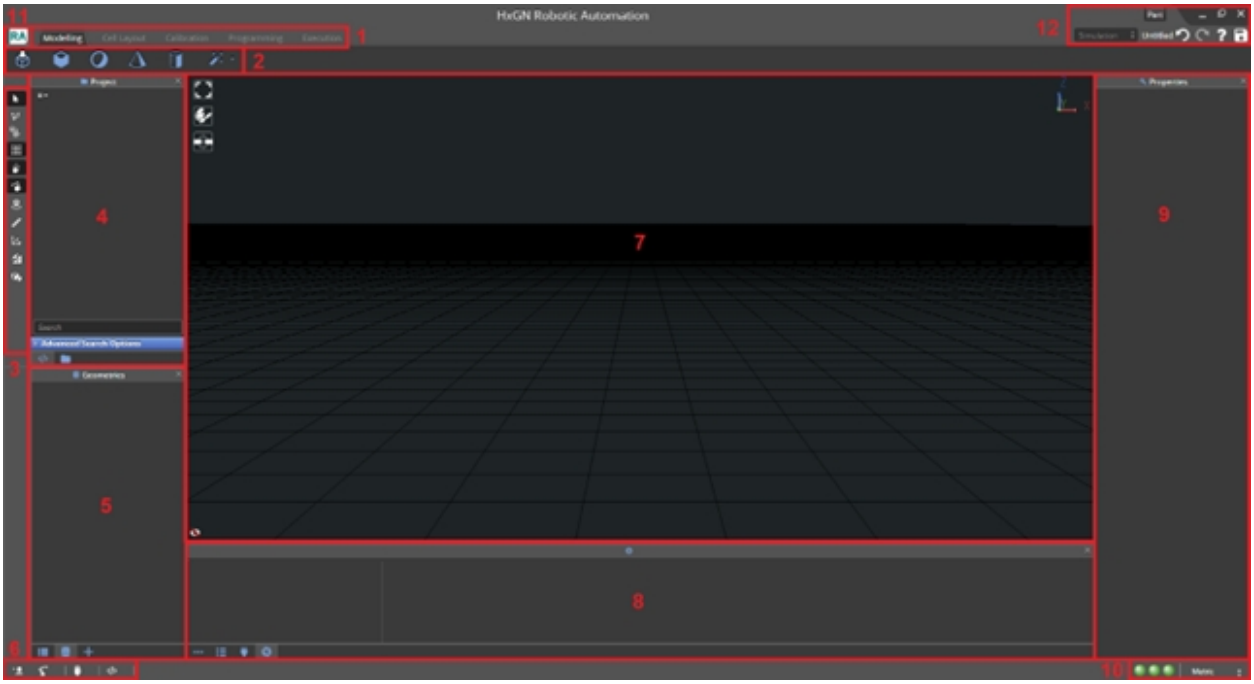
2. **Open An Existing Project**. The **Open Recent** area can also be used to choose a project that was recently used, and a Search bar helps to filter among them.
3. **Clear Recent Files**, to remove all files from Open Recent area.
4. **Import**, to extract and open a project from a zip file previously exported from HxGN Robotic Automation
5. **Settings**, to navigate to the HxGN Robotic Automation settings, described in detail in chapter Settings Menu
6. **Manage License**, to manage the activation of the license as explained in chapter Configuration and Licensing
7. **Help**, to be redirected to the web help.




HxGN Robotic Automation UI can be divided into twelve fundamental areas.

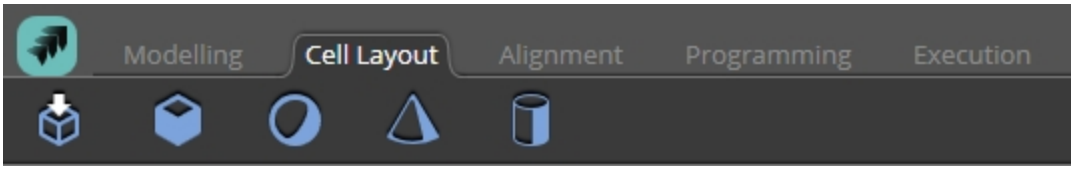
1. Application Mode Tabs
2. Specific Application Mode Toolbar
3. Special Mode Toolbar
4. Project Tree and Library / Alignment / Program Windows
5. Geometries / Measurements / Conditions Windows
6. Active Machine / Active Sensor / Active Program Line Buttons
7. 3D Graphic Window
8. Status / Execution Status / Connection / GoPro Windows
9. Properties Window
10. Metrology / Validation Engine / Collision Engine Buttons

11. RA Main Menu Button
12. Top-right Menu




## Application Mode Tabs

HxGN Robotic Automation UI is defined by five main tabs defining the Application modes, corresponding to the main user flows which are Modelling, Cell Layout, Alignment, Programming and Execution



### Modelling

This is an application mode for advanced users. The focus is to enable the user to model any kinematic or static element. After modelling, you can export them to your library (right-click) and in this is the way generate your own elements catalogue to work with.

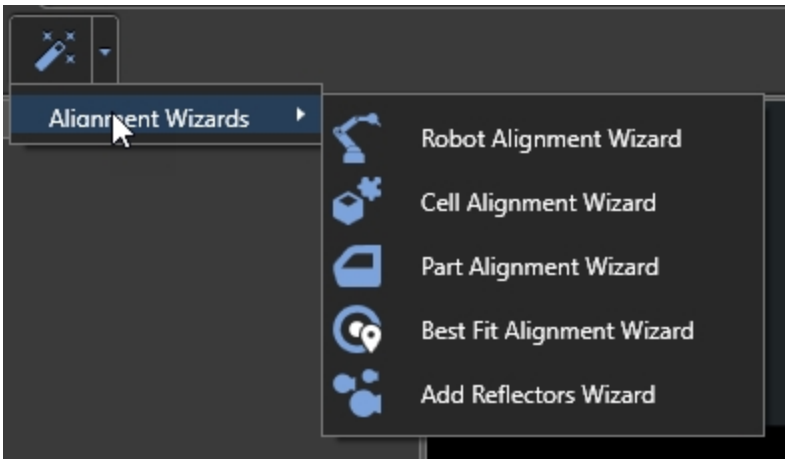
### Cell Layout

The Cell Layout mode's main purpose is to create a virtual representation of a cell with moving components. In this mode, the kinematic or static elements can be imported from the library or directly by importing CAD models and placed according to the cell layout of real or simulated cells. Existing cell layouts can be edited.

### Alignment

Once the elements are positioned and oriented in the cell, the next step is going to be Alignment. Aligning a cell means aligning the elements in the virtual environment exactly according to where the cell elements are in the real physical cell. This step is critical for the process since any difference between real and virtual cells will affect path planning, collision detection, scans that will be received from the robot, etc.

This mode has a wizard to aid you with the alignment containing the following steps:




- Robot Alignment Wizard
- Cell Alignment Wizard
- Part Alignment Wizard
- Best Fit Alignment Wizard
- Add Reflectors Wizard

The first step is to align all the elements in the cell, using the Cell Alignment. Then with the cell aligned, the robot and the tool must be aligned to have them placed as they are in the real cell in the most precise way.

Lastly you can Add Reflectors through the wizard that are either linked to the Rotary Table (and will rotate with it) or placed at fixed location within the cell. Reflectors can either be added through picking on CAD (selecting it in the cell UI) or through measuring them with the tracker.

In case a new part must be aligned in the cell, Part Alignment wizard is available to simplify the process.

### Programming

In this application mode, the user can:

1. Create Programs in Manual or Automatic Mode for Part Inspection.
2. Edit existing programs e.g., to adjust them to be perfected for the required task the robot will perform.
3. Validate the programs to be collision free and simulate the program.
4. Set the program specific properties (define the Metrology application to be used).
5. Get the features from the connected Metrology application.

Depending on the application the following could be considered:

1. If the program will be for measuring parts and executing it online (not simulation), make sure your cell is aligned, the robot and tool are calibrated, your fixture, and part are loaded, properly aligned and the features you are planning on measuring are imported (from Robotic Automation or Metrology).
2. If the program will be for measuring parts and execute it online, make sure your cell is aligned, robot and tool are calibrated, you have the Rotary Table Compensation map in case there is Rotary Table in the project, your fixture and part are loaded, properly aligned and the features you are planning on measuring are imported.

### Execution

Application mode where the user can:

- Execute Programs Offline to simulate the inspection process and see the simulated data.
- Execute Programs Online to collect actual data that is sent to Metrology.
- Control the Program Execution (start, resume, stop)
- Export programs to be executed by Execution Manager.

## Specific Application Mode Toolbar

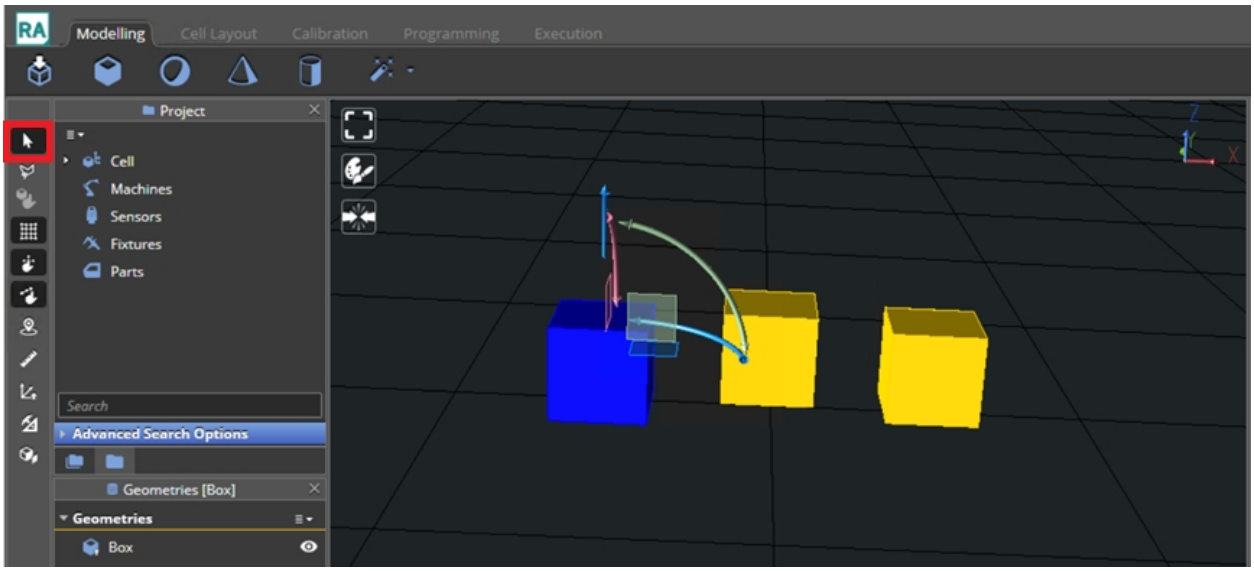
This toolbar changes depending on the Application Mode in use, and it will be explained in detail in the dedicated chapters.

## Special Mode Toolbar

Special modes toolbar is found on the left side of the window.

### Regular Selection

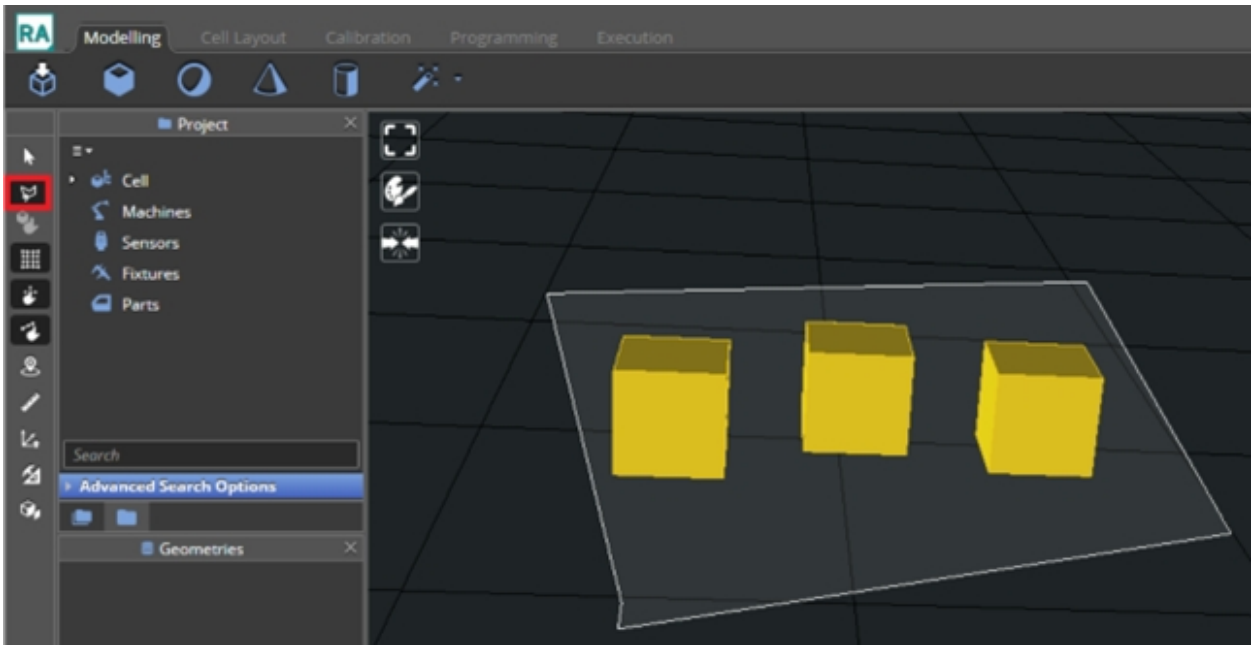
Regular choice / polygonal selection controls the way elements are selected on Robotic Automation. As you can see in the image below, regular choice makes the mouse cursor able to select a single object.




## Polygon Selection

Polygonal choice makes the mouse cursor to be able to draw a polygonal area where cell elements are present. Do the choice with the left button of the mouse and do right-click when the desired polygon is created, then all the cell elements within the polygonal selection will be selected and highlighted after pressing the [Enter] key.

| | |
|---|---|
| ⚠ | The polygonal features are linked to the camera view, zooming in and zooming out from the 3D world will change the world view, but the polygonal selection will stay the same size! |




### Polygon Features Selection

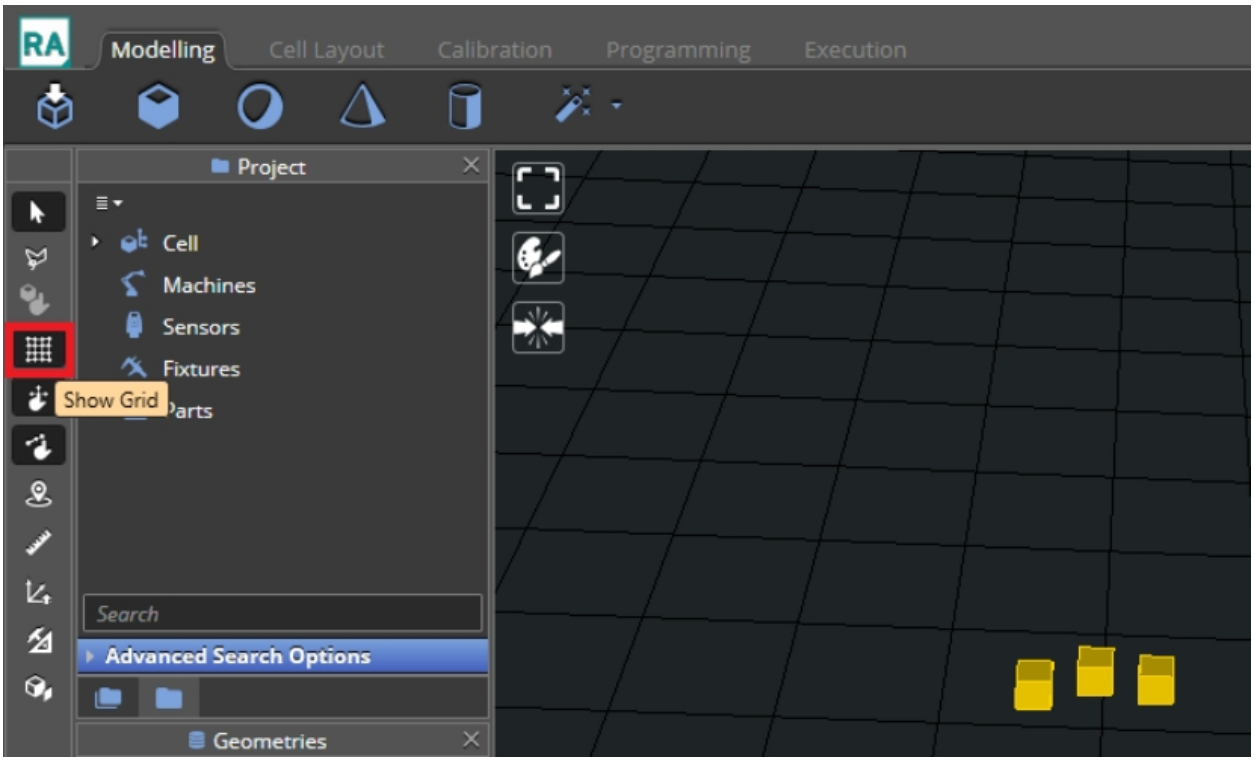
Polygonal choice makes the mouse cursor to be able to draw a polygonal area where cell elements are present. Do the choice with the left button of the mouse and do right-click when the desired polygon is created, then all the features within the polygonal selection will be selected and highlighted after pressing the [Enter] key.

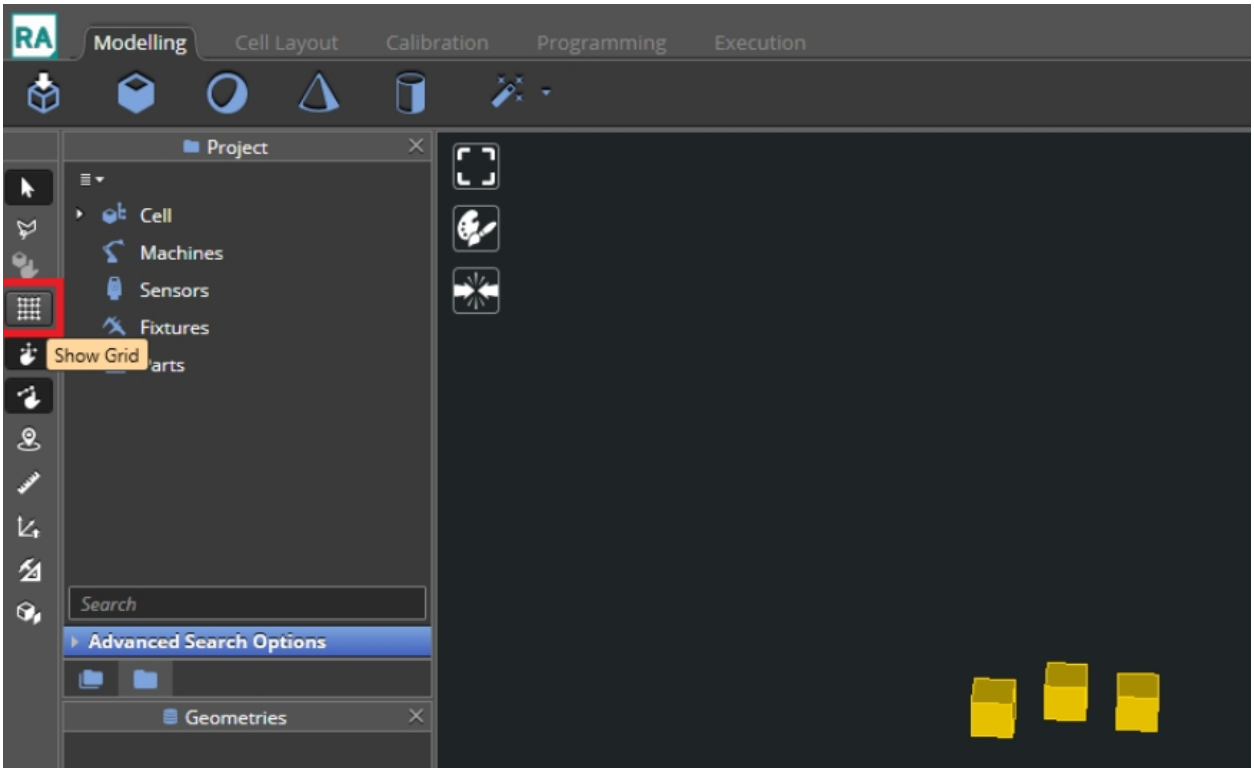
### Pick on CAD

Pick on CAD is a special mode, used to adjust robot position using the TCP (Tool Center Point) or directly create manually robot positions. It can be only activated in programming mode, see Pick on CAD

### Show/hide grid

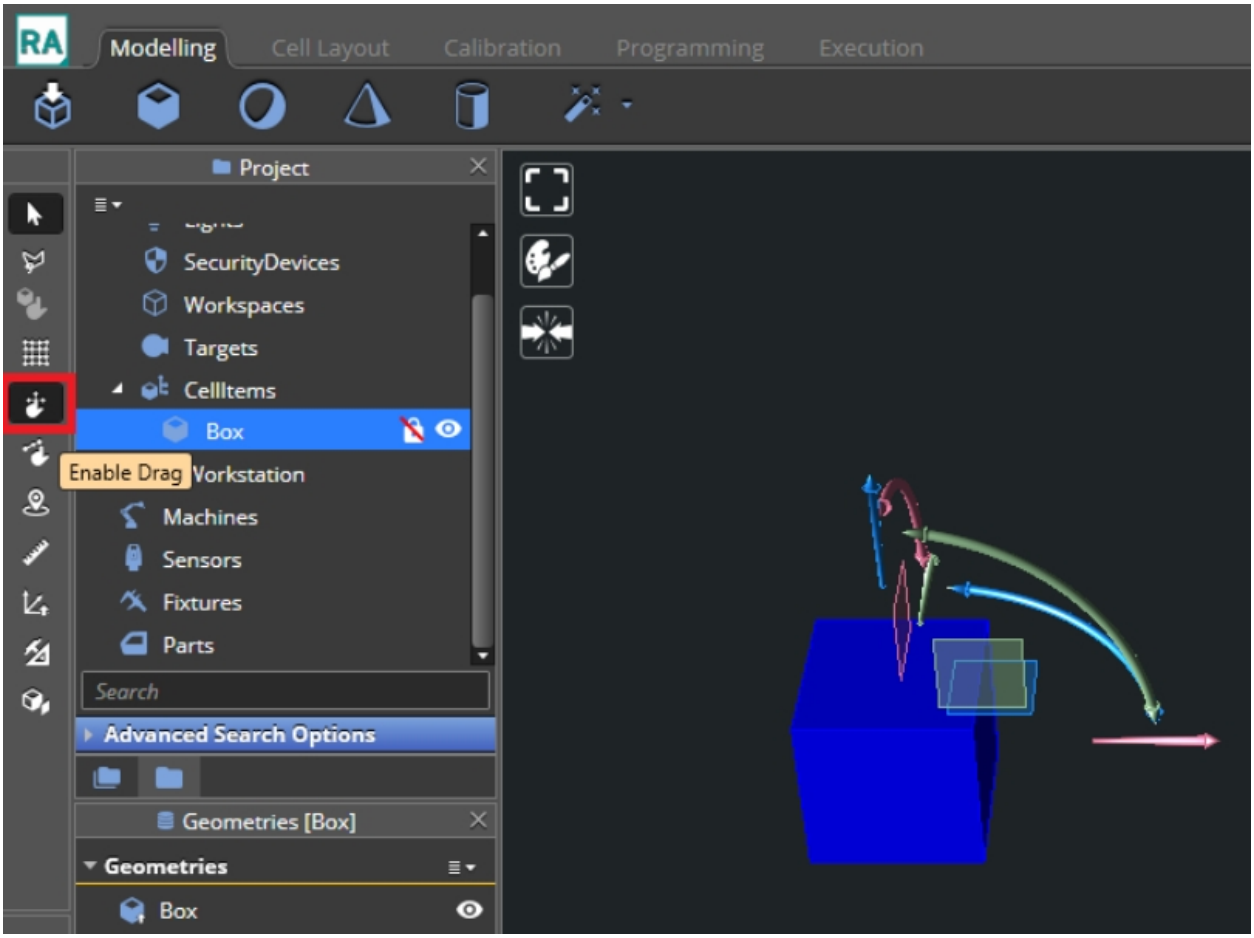
Toggle the “Show/hide grid” button to visualize or hide the grid in the graphic window.

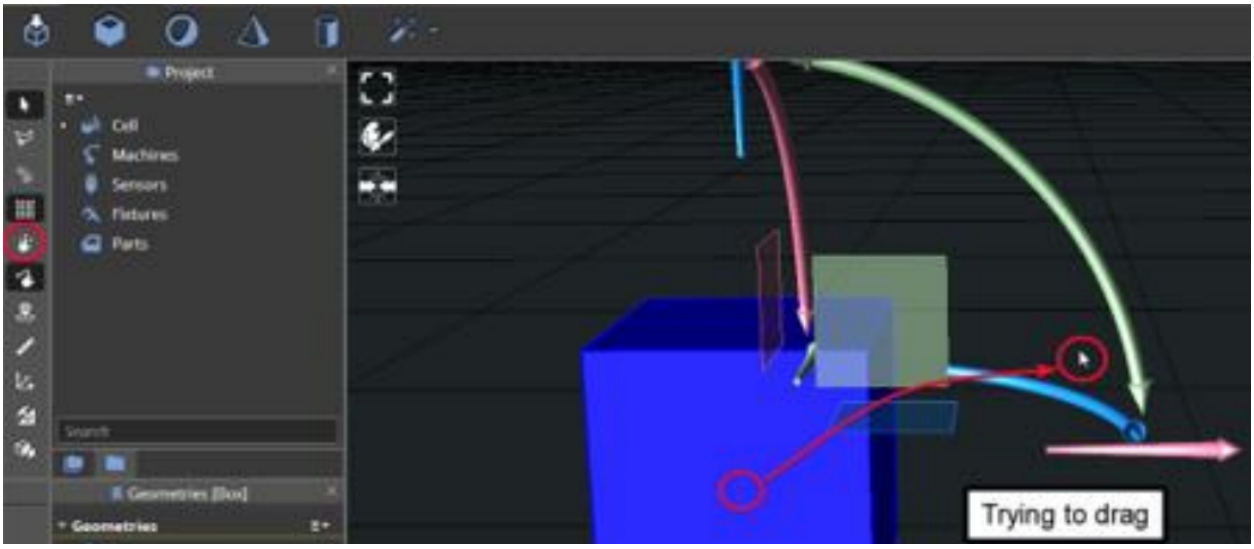

RA
Modelling
Cell Layout
Calibration
Programming
Execution
Project
Cell
Machines
Sensors
Fixtures
Show Grid
Search
Advanced Search Options
Geometries


RA
Modelling
Cell Layout
Calibration
Programming
Execution
Project
Cell
Machines
Sensors
Fixtures
Show Grid
Search
Advanced Search Options
Geometries

## Enable/disable drag

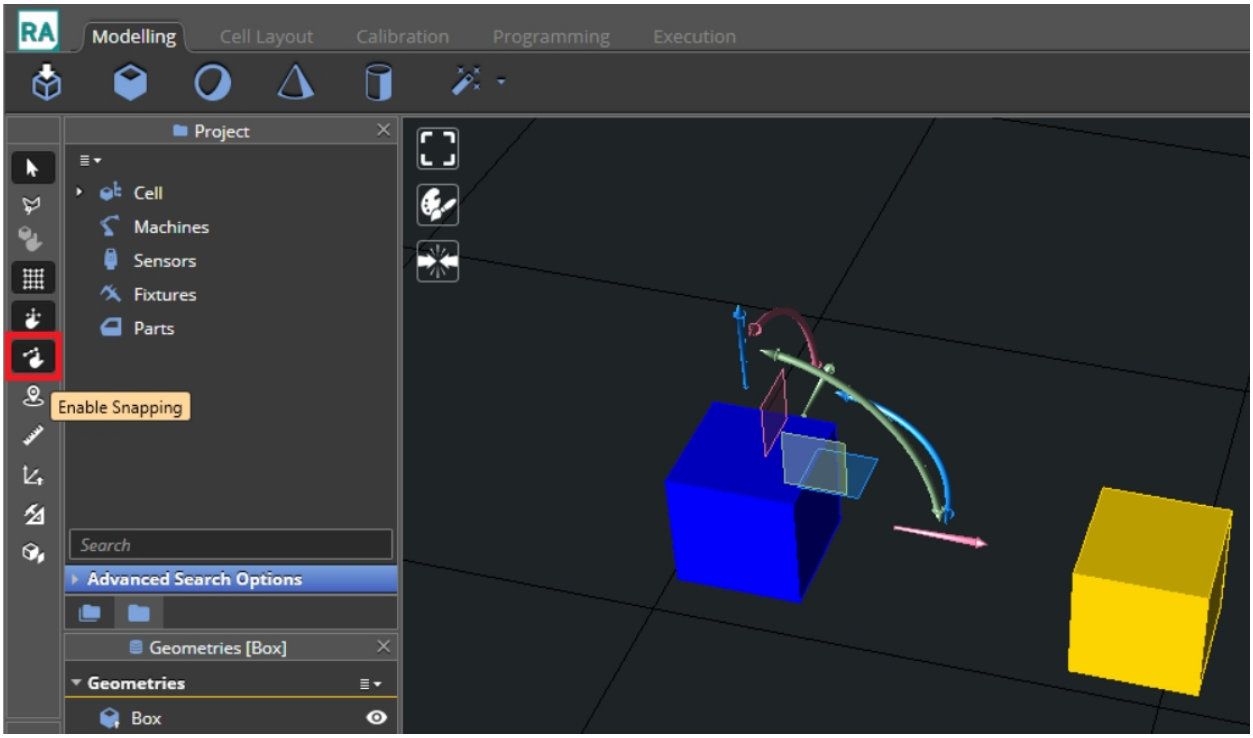
Toggle the “Enable/disable drag” button to be able to drag or to disable the dragging. If dragging is disabled, if the user tries to drag the object, it won’t move.

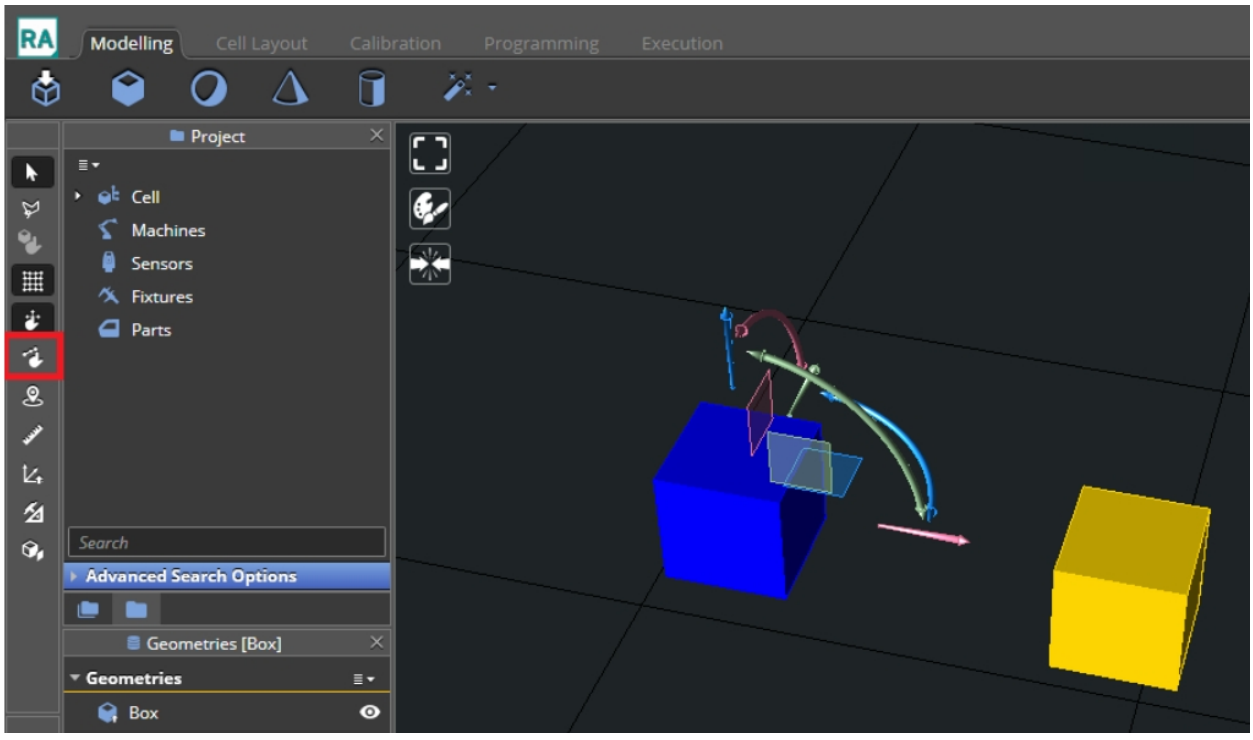


## Enable/disable snapping

Toggle the “Enable/disable snapping” button to be able to snap or to disable snapping.



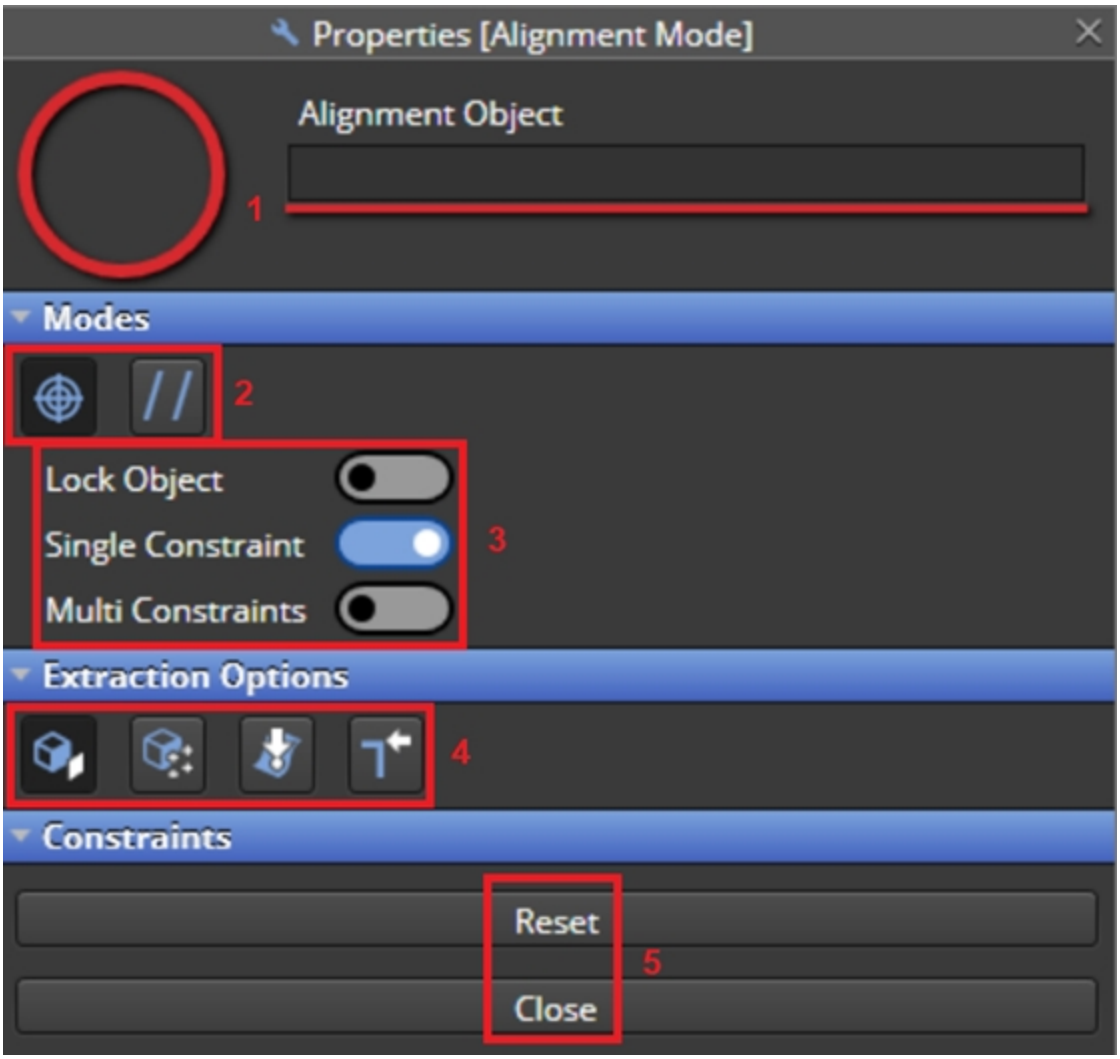



### Alignment Mode

Alignment brings the possibility to the user to set up constraints between objects or features to align them.

1. Name of the object that you are currently aligning.
2. Select the align mode: coaxial or parallel.
3. Selections switches:
    a. Lock Objects: if checked, the object will be locked after finishing the alignment.
    b. Single Constraints: if checked, the single constraint mode will be selected.
    c. Multi Constraints: if checked, the multiple constraint mode will be selected.
4. Extraction features modes:
    a. Feature: extracts feature from object.
    b. Feature from points: extracts feature generated from points.
    c. Surface point: extracts point from surface.
    d. Edge point: extracts points from edge
5. Reset or close the current alignment.

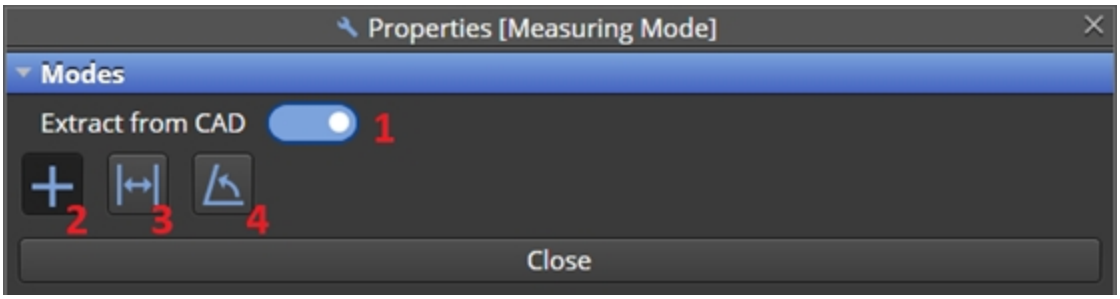
For more details on how to use the manual alignment mode, please refer to Manual Alignment Workflow

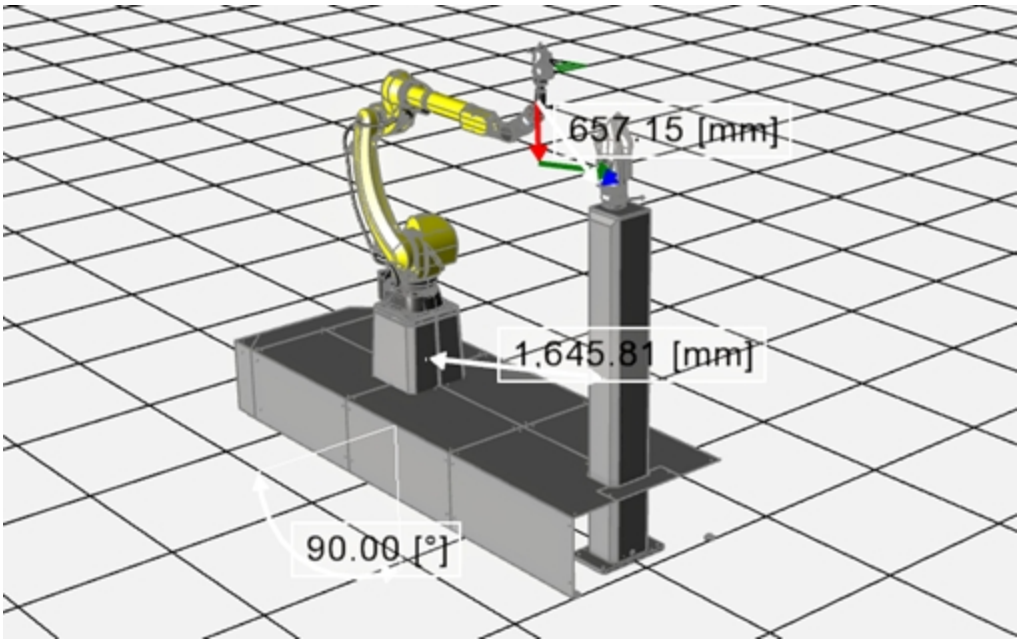



### Measurement Mode

1. **Extract from CAD**: if checked the features that you will measure will be from the CAD.
2. **Free Point**: to calculate the distance between two free points selected in the cell.
3. **Distance**: to calculate the distance between two surfaces selected in the cell.

4. **Angle**: to calculate the angle between two surfaces selected in the cell.



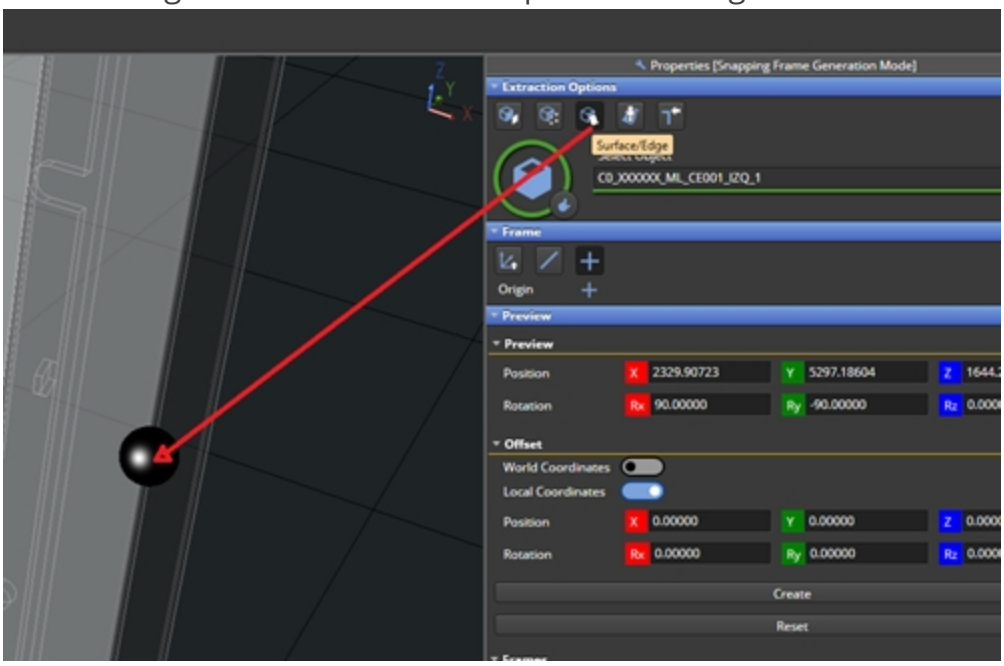




## Snapping Frame Generation Mode

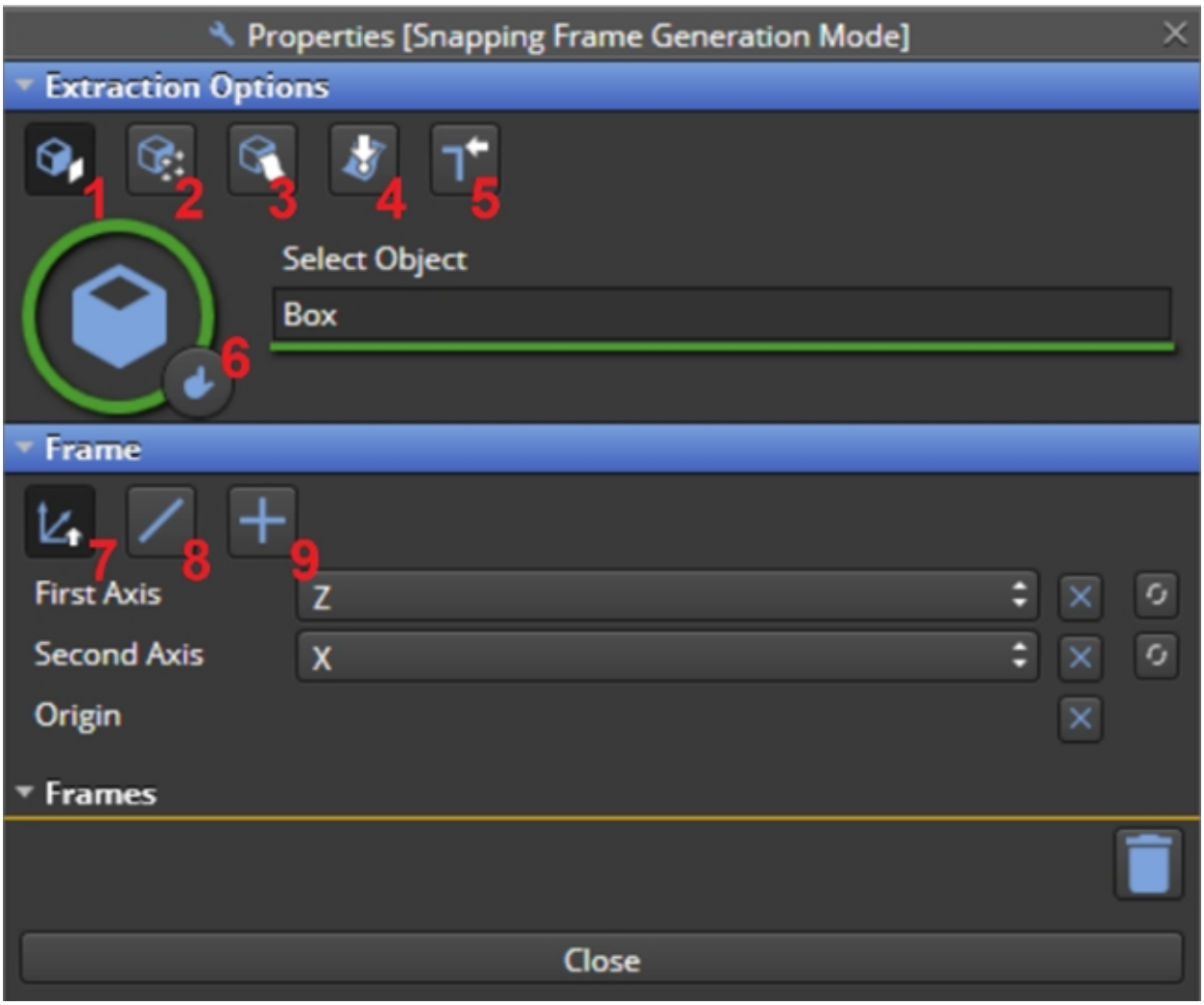
This mode is to generate frames to an object. Please refer to Modelling.htm for more information about its usage in Modelling mode. A snapping frame is a point added to a cell item, once 2 objects with snapping frames get closer, a bubble appears and if we bring them close enough and stop dragging, the items will be aligned by the snapping frame.

There are few options to extract features:

1. Feature: extracts feature from object.
2. Feature from points: extracts feature generated from points.

   Surface/Edge: This is to extract the point from Edge between two surfaces:
3. 

4. Surface point: extracts point from surface.
5. Edge point: extracts points from edge.
6. Select the object that you want to create a snapping frame on.
7. Pick first axis of the frame then pick the second axis of the frame and to finish pick the origin of the frame.
8. Pick the origin of the normal and then pick the origin of the normal again.
9. Pick the origin of the point..



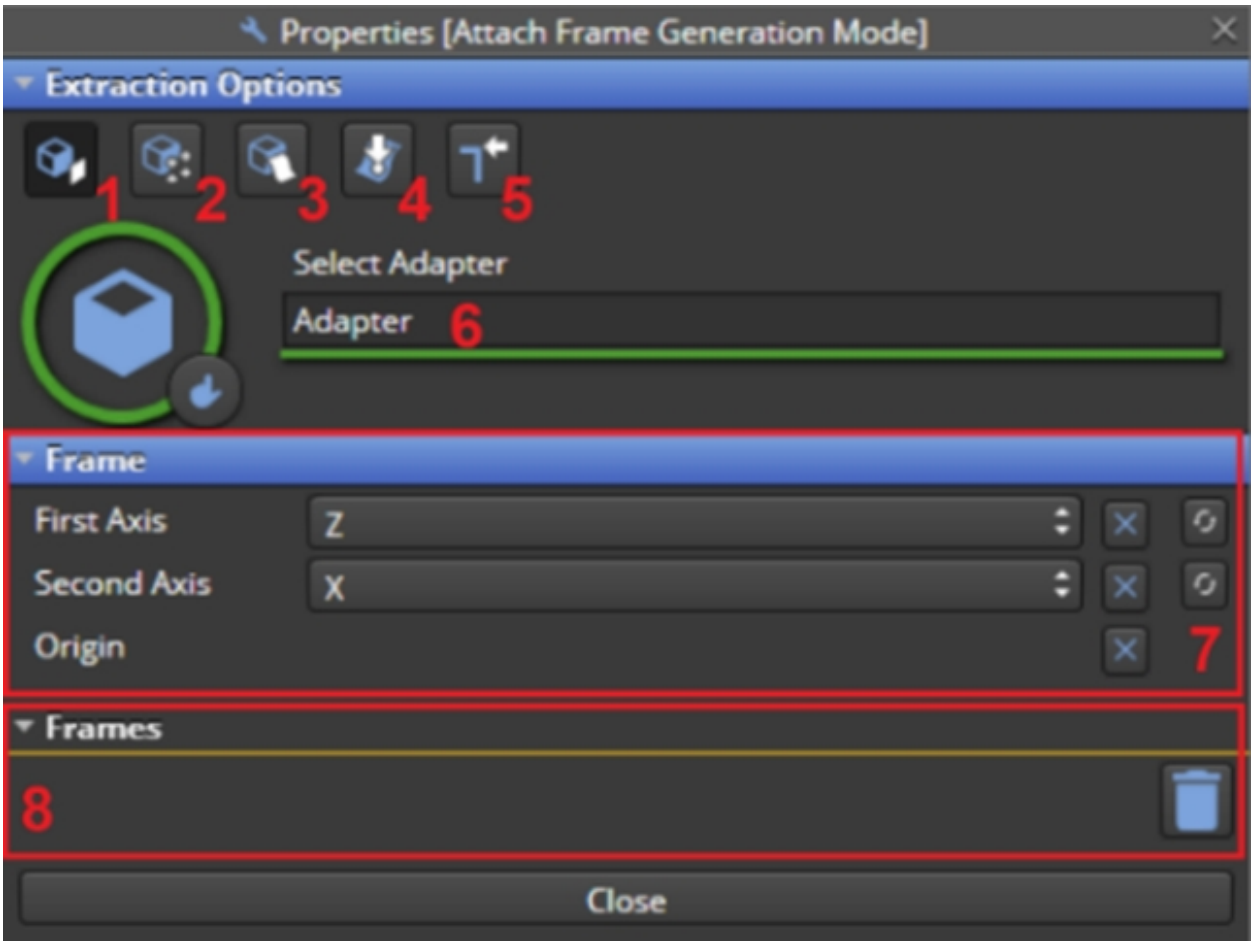

Once the object and the frames conditions are selected, we can see a preview of the position and rotation of the frame. Also, we can choose into “World Coordinates” or “Local Coordinates.” Then we can press “Create” to create it or “Reset” to reset it.

Properties [Snapping Frame Generation Mode]
Select Object
Box
Frame
First Axis
Z
Second Axis
X
Origin
Preview
Preview
Position
X -2232.99
Y -1.45
Z 877.26
Rotation
Rx 90.00
Ry 0.00
Rz 180.00
Offset
World Coordinates
Local Coordinates
Position
X 0.00
Y 0.00
Z 0.00
Rotation
Rx 0.00
Ry 0.00
Rz 0.00
Create
Reset
Frames
Close

### Attach Frame Generation Mode

This mode is to generate frames to an adapter. Please refer to Modelling.htm for more information about its usage in Modelling mode.

1. Extraction features mode.
2. Feature: extracts feature from object.
3. Feature from points: extracts feature generated from points.
4. Surface point: extracts point from surface.
5. Edge point: extracts points from edge.
6. Select the adapter that you want to create attaching frame on.
7. Select the axis and the origin of the frame.
8. List of the frames created on the adapter.




Once the adapter is selected and the axis and the origin are picked, we can see a preview of the position and rotation of the frame. Then if you want to create the frame you must click on “Create” and then the frame will be created and will appear on the list. We can reset as well if the preview is not how we expected and try again.

Properties [Attach Frame Generation Mode]
Extraction Options
Select Adapter
Adapter
Frame
First Axis
Z
Second Axis
X
Origin
Preview
Preview
Position
X 2555.50
Y 565.31
Z 686.02
Rotation
Rx 0.00
Ry -54.11
Rz 90.97
Offset
World Coordinates
Local Coordinates
Position
X 0.00
Y 0.00
Z 0.00
Rotation
Rx 0.00
Ry 0.00
Rz 0.00
Create
Reset
Frames
Close

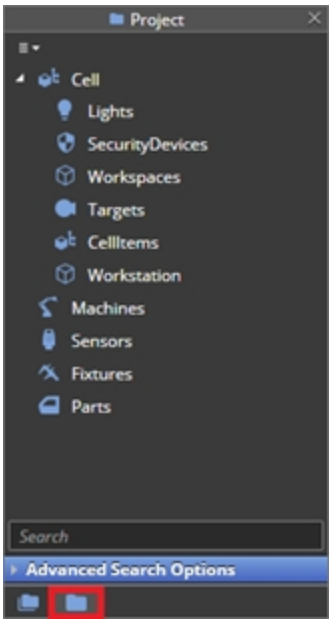
### Feature Generation Mode

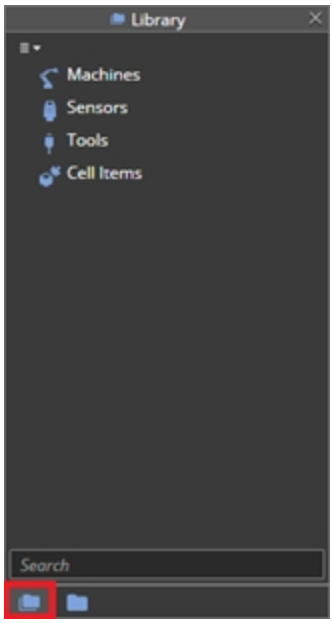
Click the button to open the Feature Generation Mode, see chapter Feature Generation Mode for more information.

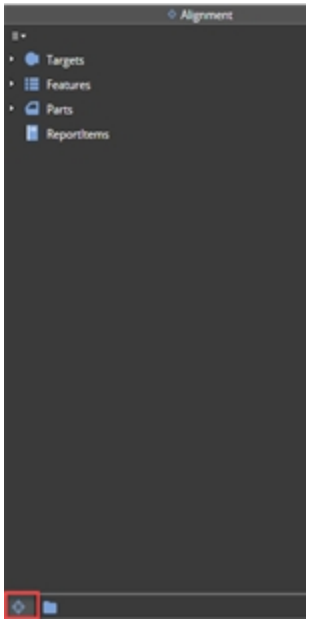
## Project Tree / Library / Alignment / Program Windows

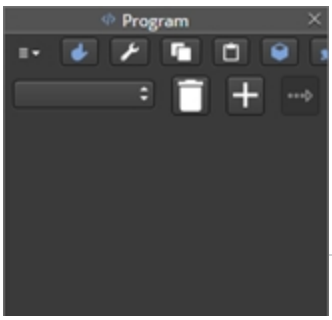
This window is divided into two tabs depending on the specific application mode used.

- If using either Modelling or Cell Layout modes, the **Library** and **Project Tree** tabs will be shown
- In case Alignment mode is used, the **Project Tree** and **Alignment** tabs
- Using either Programming or Execution modes, the **Program** and Project Tree tabs.

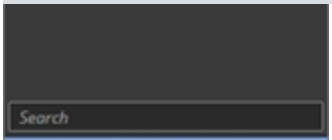




The **Project** tree is defined by five areas:

1. **Cell**: in this list, you can find:
    a. **Lights**: currently not supported
    b. **Security devices**: currently not supported
    c. **Workspaces**: Any workspace the user created will be displayed here.
    d. **Targets**: Any reflectors created either manually or through the rotary table compensation will be displayed here, as well as T-Probes.
    e. **Cell items**: All the items in the cell will be displayed here (walls, fences, and other physical objects).
    f. **Workstation**: defines the cell elements used by the project as defined in the Cell.json configuration file (see Configuration and Licensing.htm).
2. **Machines**: On this list, you will find all the machines this project has (Robots, Sliders, Turntables).
3. **Sensors**: All available sensors (for example Trackers) will be displayed under this tree.
4. **Fixtures**: All fixtures in the cell will be displayed here.
5. **Parts**: All the parts in this cell, including their features, will appear under this tree.

Similarly, the **Library** is composed by four areas with the same definitions supplied above for the project tree:

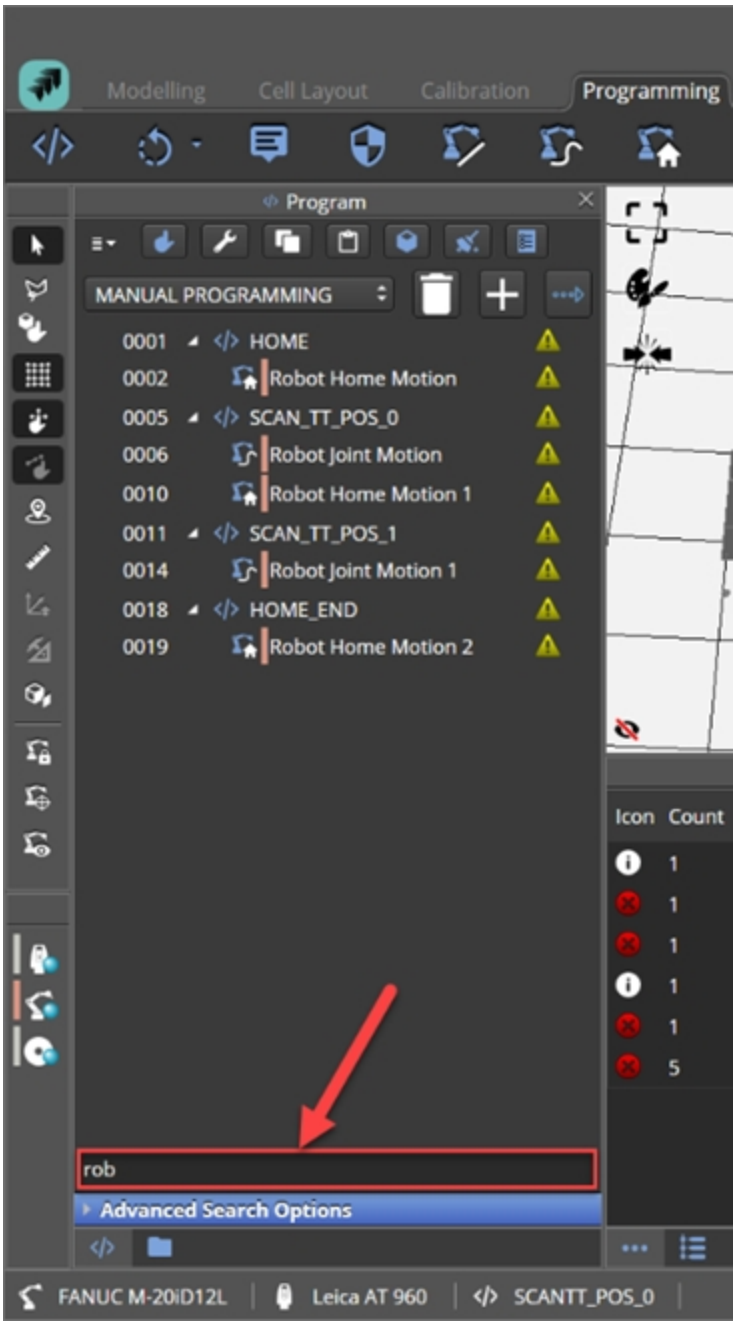
1. Machines
2. Sensors
3. Tools
4. Cell Items

For details on the Alignment window please refer to chapter Alignment - Leica.htm

**Pro Tip**: by selecting any item in the Project tree and using the [CTRL + F] keyboard shortcut, the focus will be jumping to the selected item in the 3D Window.

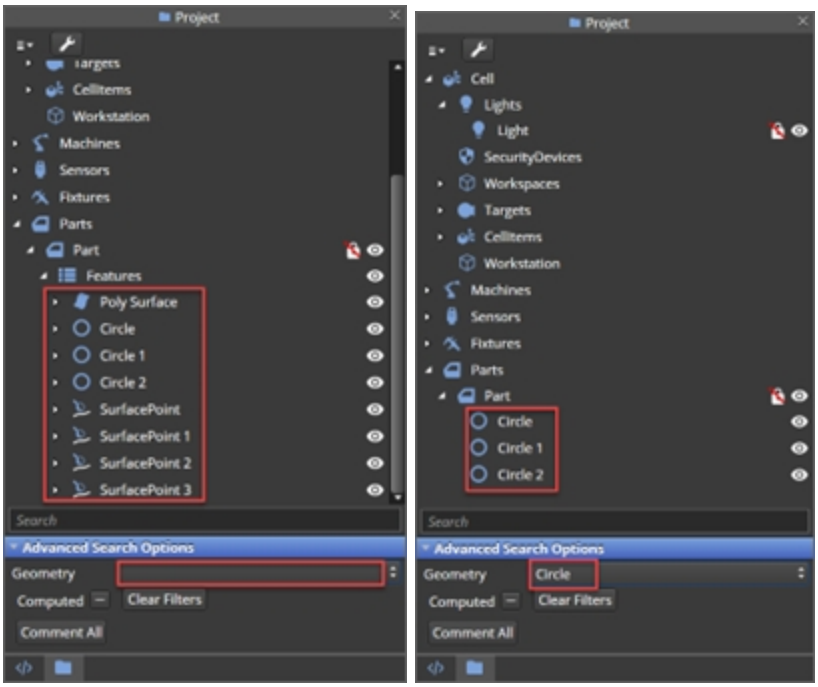
### Search bar and Advanced Search Options

Project, Program and Library windows have a search bar where the user can write the text to be search within the project tree, library items or program commands, respectively.



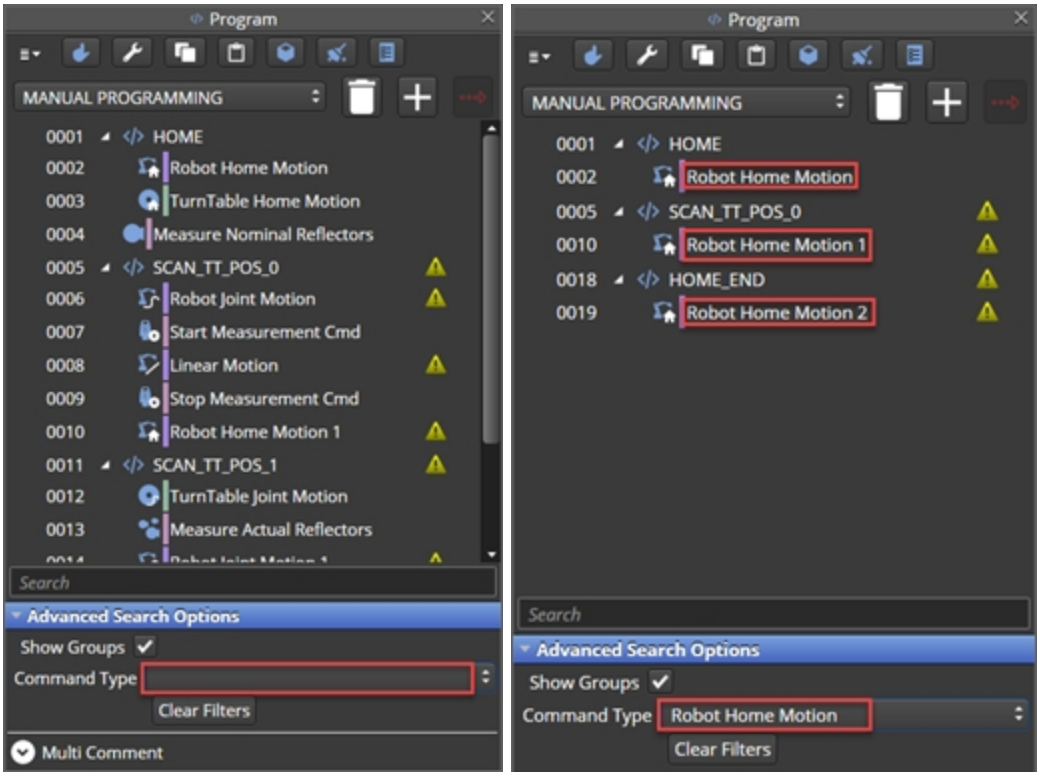

By clicking on **Advanced Search Options**, it is possible to show added capabilities.

In Project window it is possible to filter the features by the geometry types available within the project.

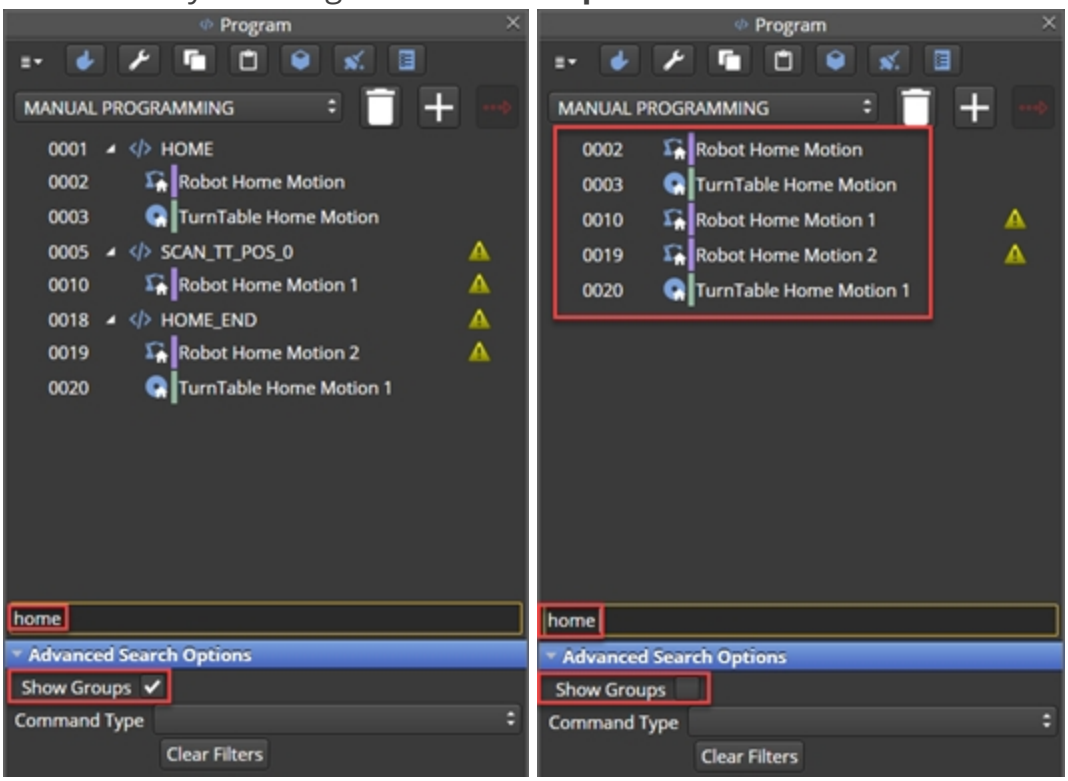



In Program window is possible to:

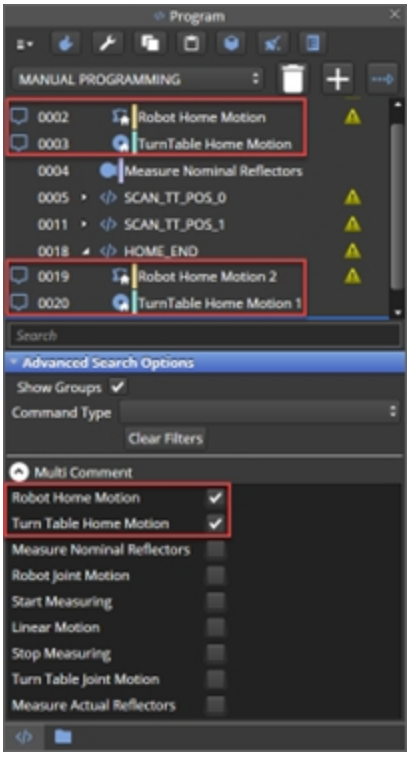
Filter command by type selecting it in the **Command Type** drop-down menu:

1. 

Show or hide the command groups after searching for specific text with the search bar by checking the **Show Groups** checkbox:

2. 

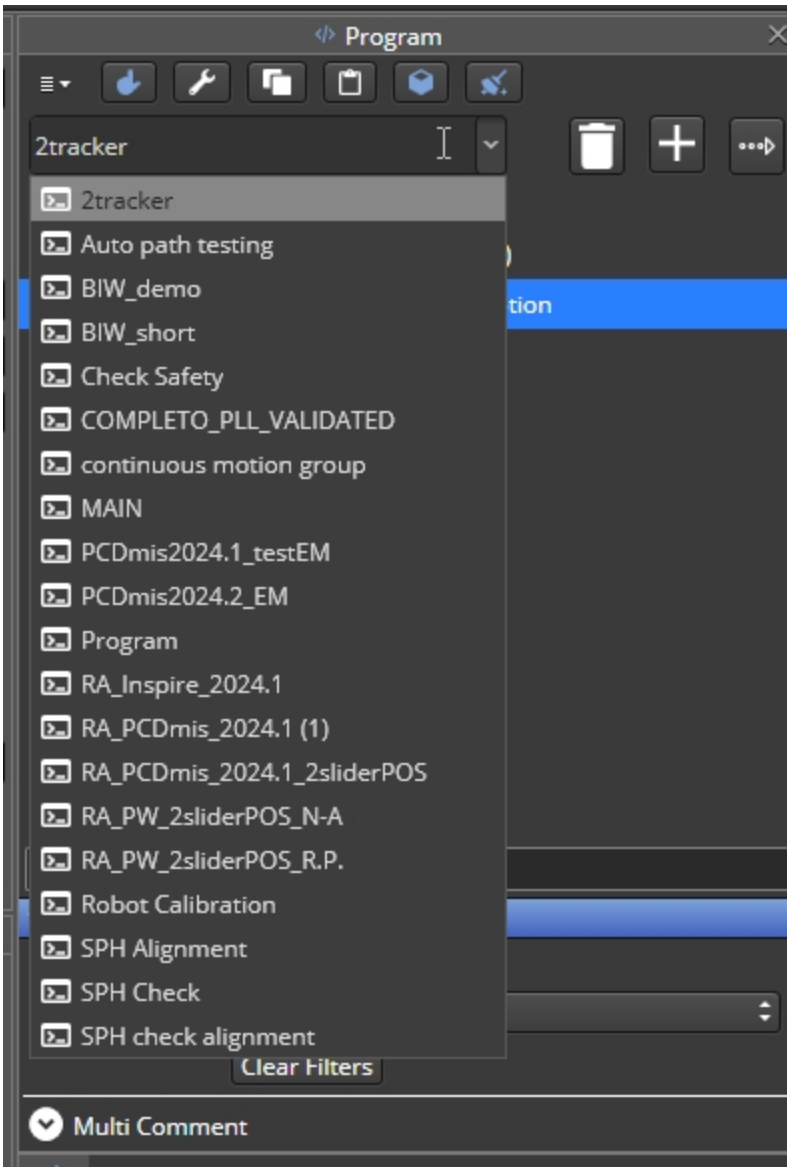
Perform a multi command comment for all the commands of the same type:

3. 


## Program and Subroutine filtering

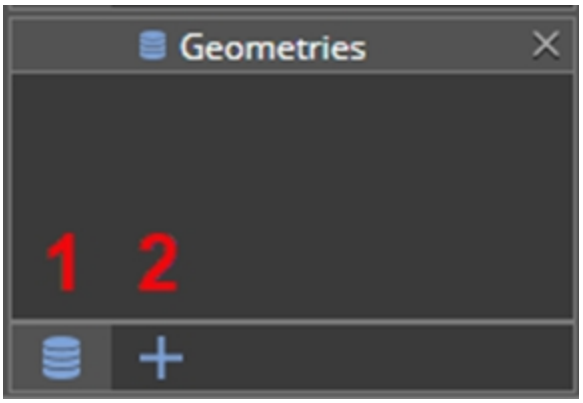
Under the program toolbar, a dropdown can be found containing all the programs and subroutines.

This dropdown has a filtered list, so typing in the search field will filter the available programs or subroutines in the list.

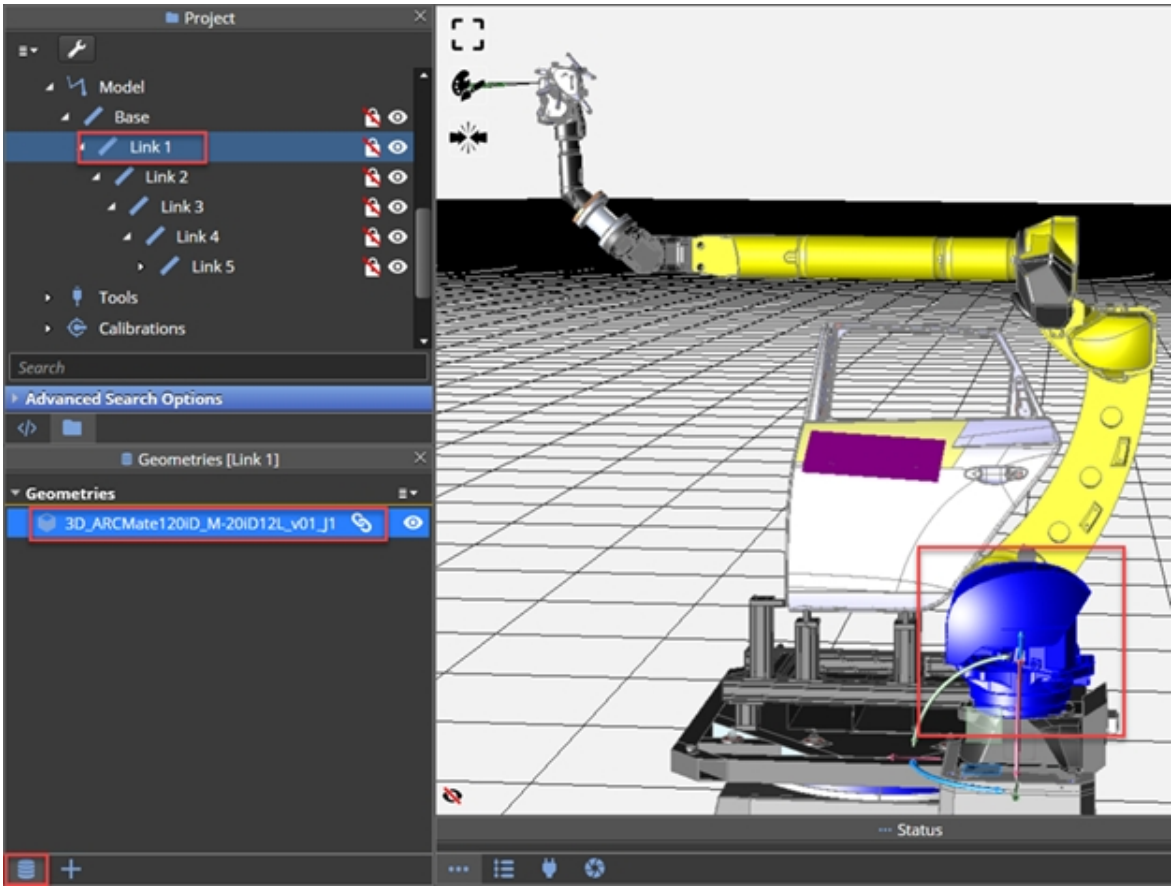



## Geometries / Measurements Windows

This window is made of two tabs:



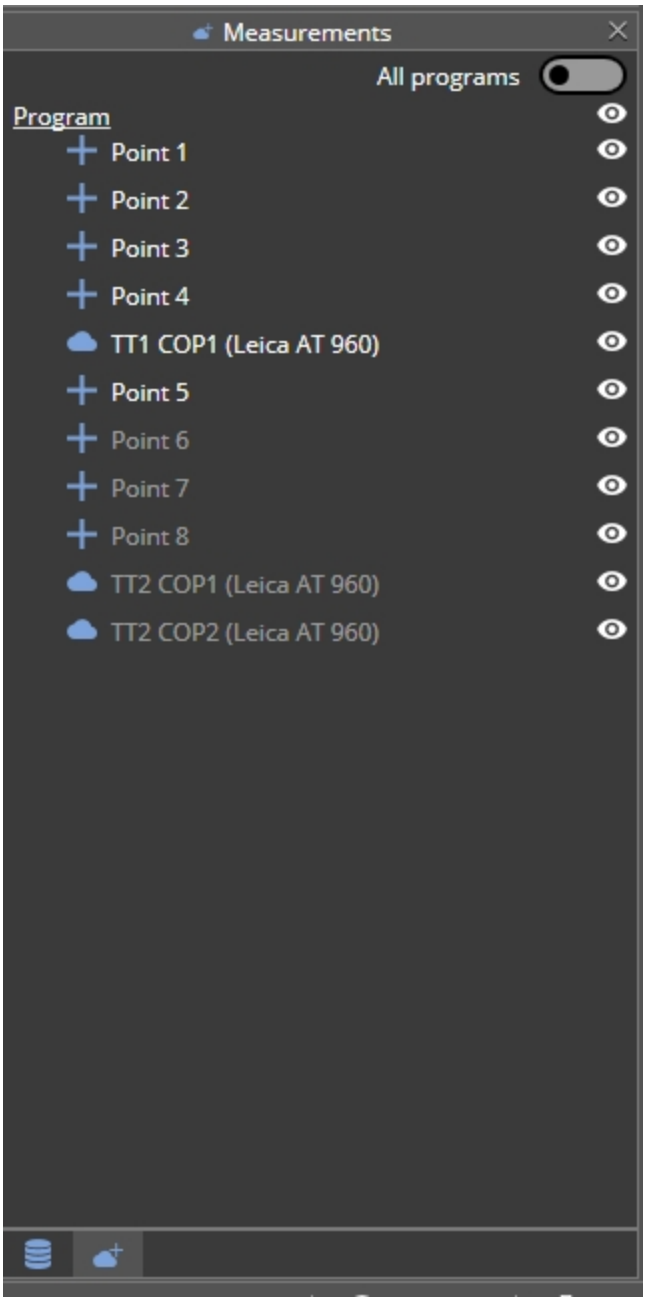

**Geometries** (always visible in all modes): this area has the geometrical objects files that are associated with any items modelled within the cell (e.g., a robot joint CAD drawing).

1. 

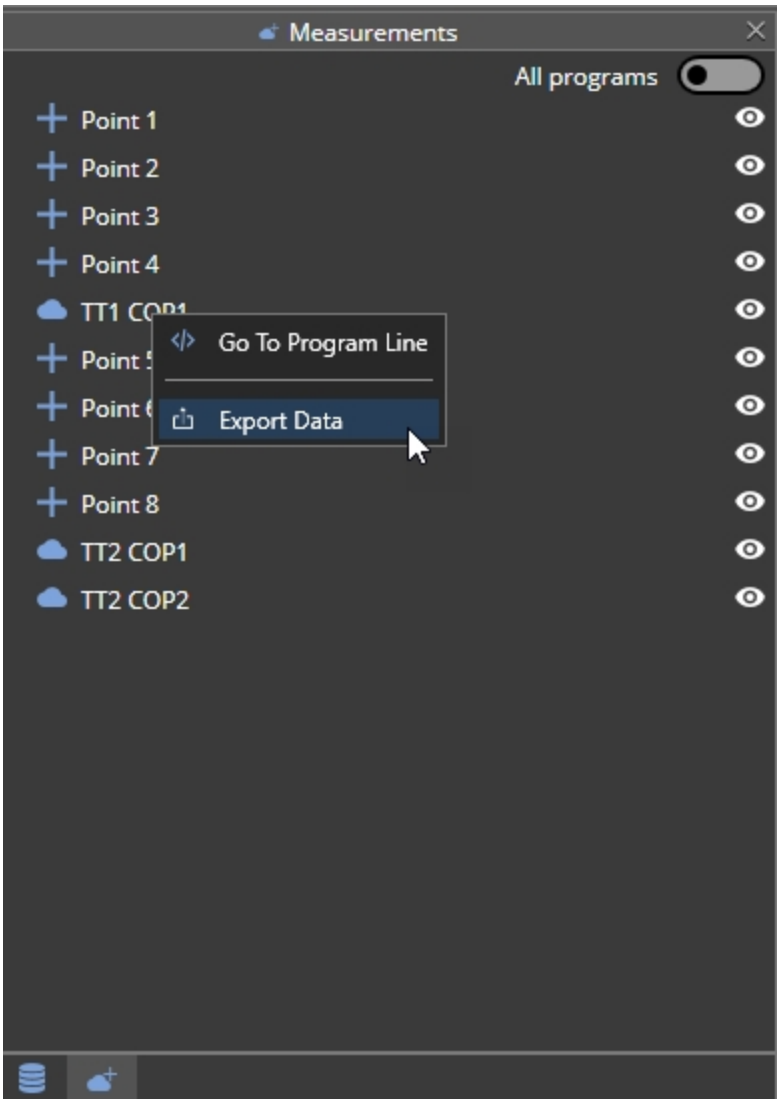
2. **Measurements** (visible only in Alignment, Programming and Execution modes): this area will hold the points measured by the tracker and the COP (Cloud of Points) acquired by the laser scanner during program execution.

   Measurements are greyed out in case the simulation player did not arrive at the time stamp they are generated yet.

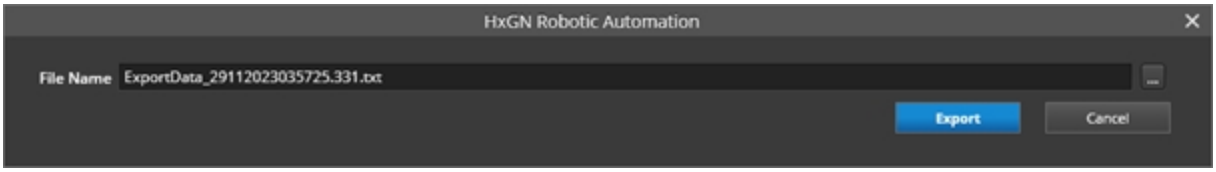
   The eye icon controls the visibility of a single measurement (point or point cloud) or the whole program




COP data can be exported by right-clicking on a Point Cloud object and selecting **Export Data**.




A pop-up will appear allowing the user to select the .txt file where the COP data will be saved as an ordered list of x, y, z coordinates and click on **Export** to save the file.



## Active Machine / Active Sensor / Active Program line buttons

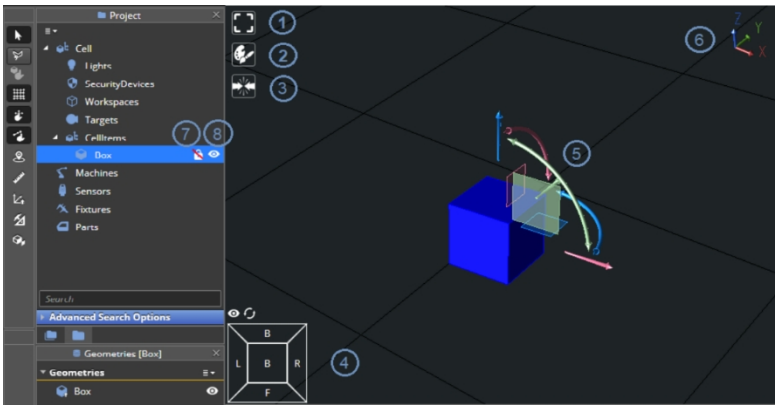
This area in the lower-left corner of the UI shows the selected elements of the projects.

1. **Active Machine**: the machine currently selected, e.g., a robot or a rotary table
2. **Active Sensor**: the sensor currently selected, e.g., the Leica AT 960 tracker
3. **Active Command**: the program name (e.g., MyProgram) or program command currently selected

## Graphic Window

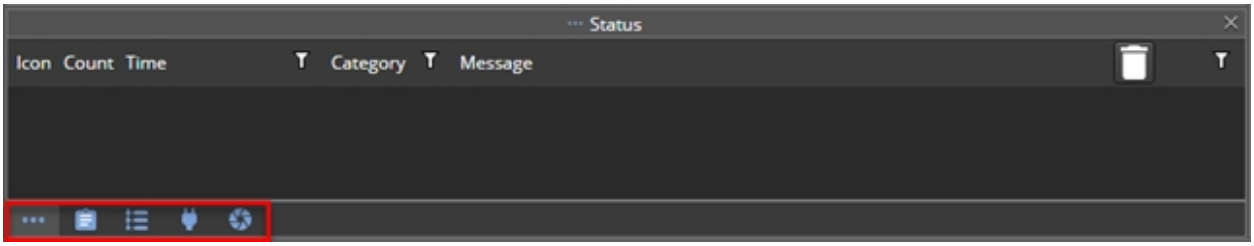
The Graphic Window can be divided into eight parts:

1. Fit all in view or Zoom in Selection
2. Graphic Menu
3. Collision Menu
4. View Modes
5. Manipulator to Drag and Rotate Selected Items
6. Cell Coordinate System
7. Object Position Lock/Unlock
8. Object Visibility Show/Hide



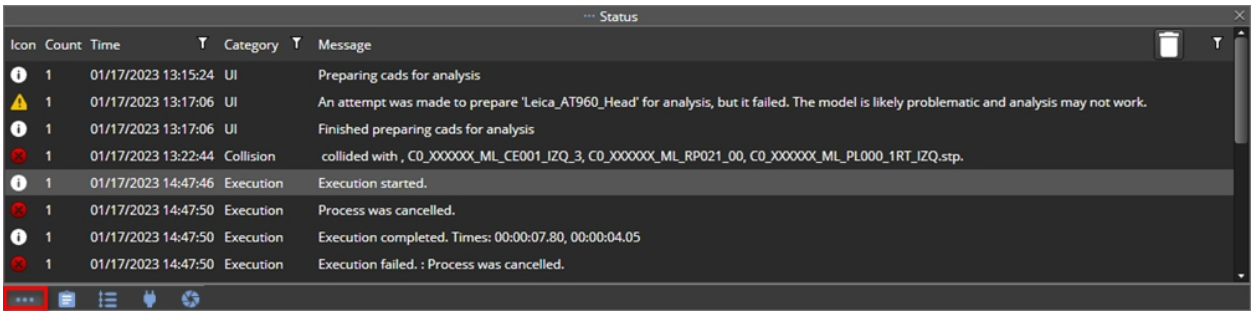

## Status / Execution Status / Connection / GoPro / Windows

This window is divided into five main tabs as described below.



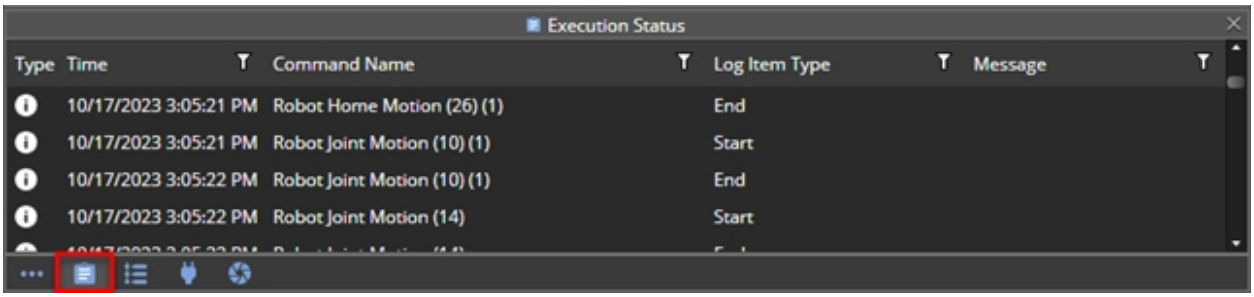

### Status

The Status window will show all the information, warning and error messages from the various user actions or software actions (errors in loading, CAD (Computer Aided Design) loaded, program started, etc.).



## Execution Status

The Execution Status window helps user to follow the flow of the program execution and see errors that only happens during the execution. It reduces the number of status messages that are seen in the status windows and make the working flow easier to understand.




## Connection

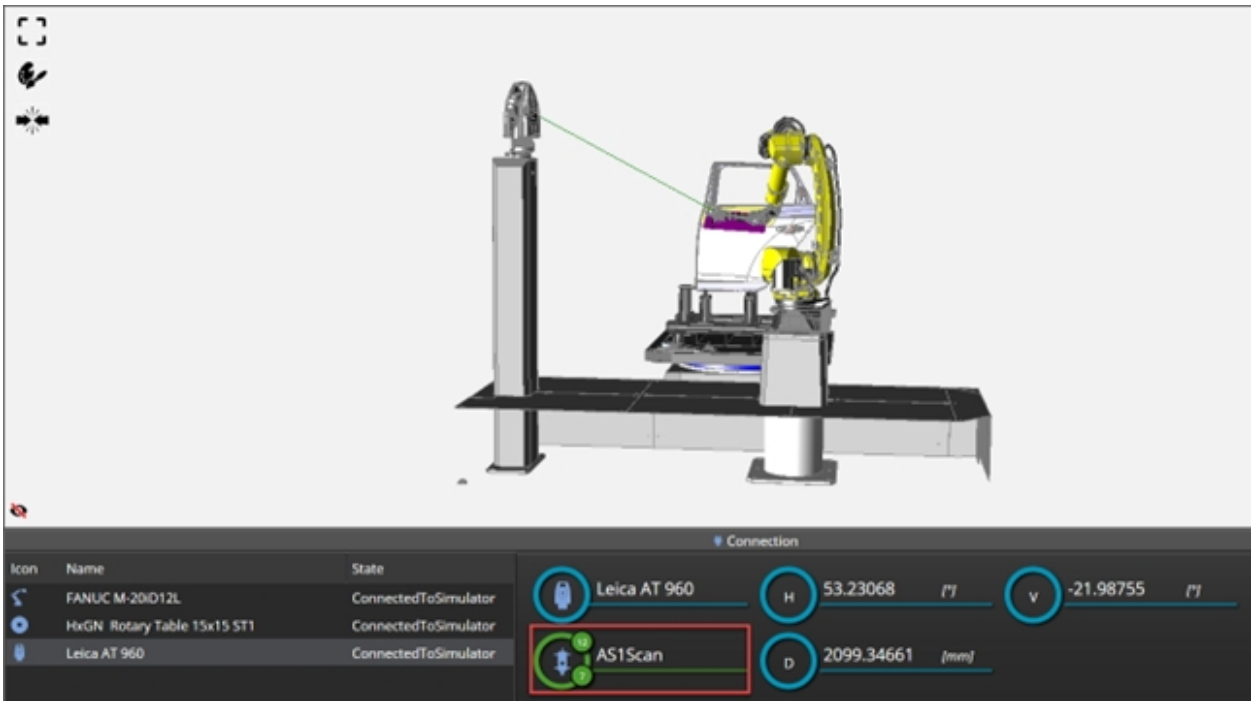
The Connection window will show:

1. On the left side, the various cell items including their icon, name, and connection state
2. On the right side, the selected item details. By default, the item coordinates are shown in joint space (e.g., for a 6 DOF robot all the 6 joint values).

To visualize the position of the robot TCP in World coordinates, it is possible to toggle the button. To restore the joint value view it is possible to toggle again the button.

The cell item details will also show more information depending on the item selected.

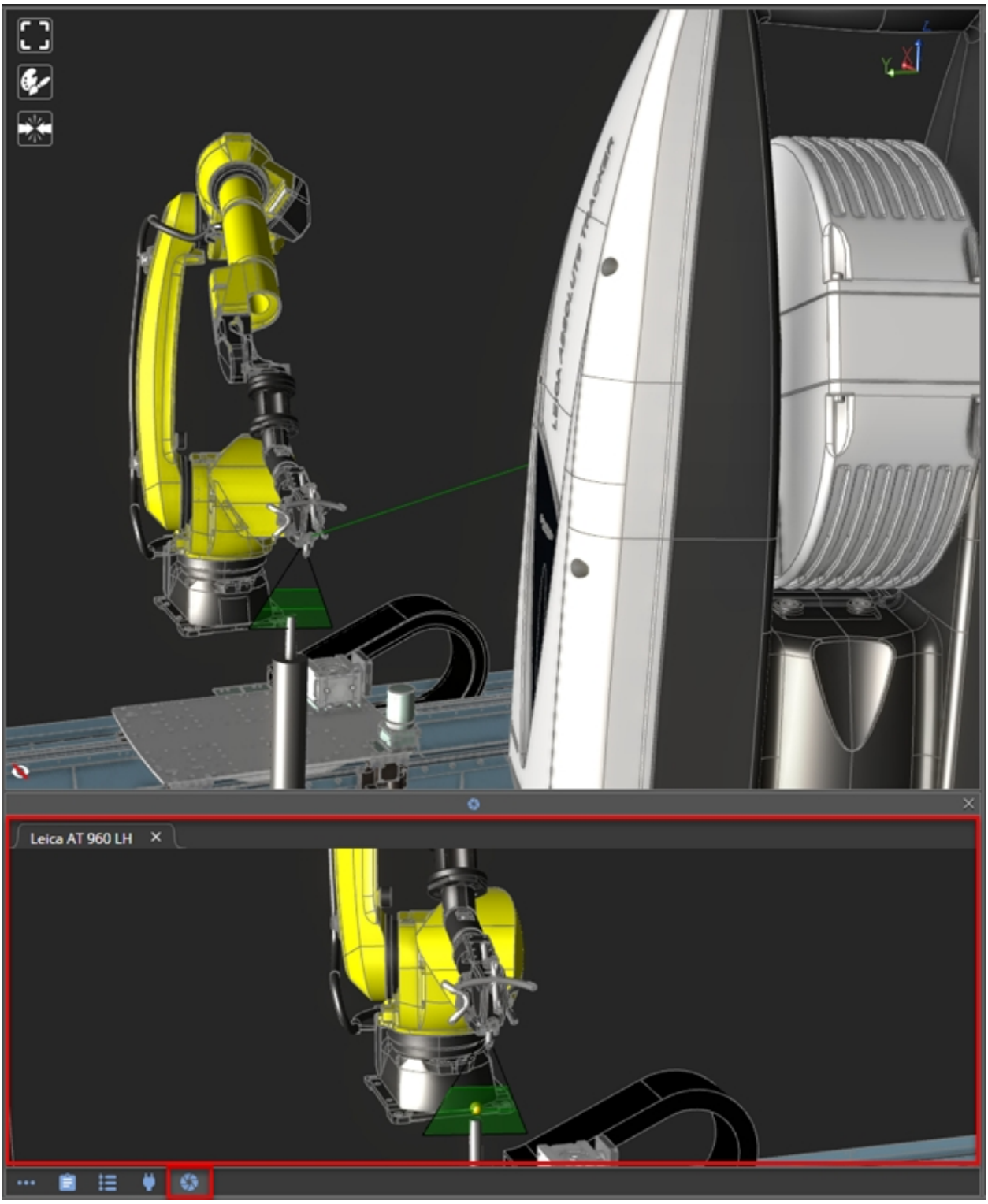
- The robot will show the tool connected and the inputs and outputs.
- The tracker will show the scanner face or the reflector to which is locked to.



Additionally, the window will have an option to access the teach pendant for the selected cell item by clicking the button.

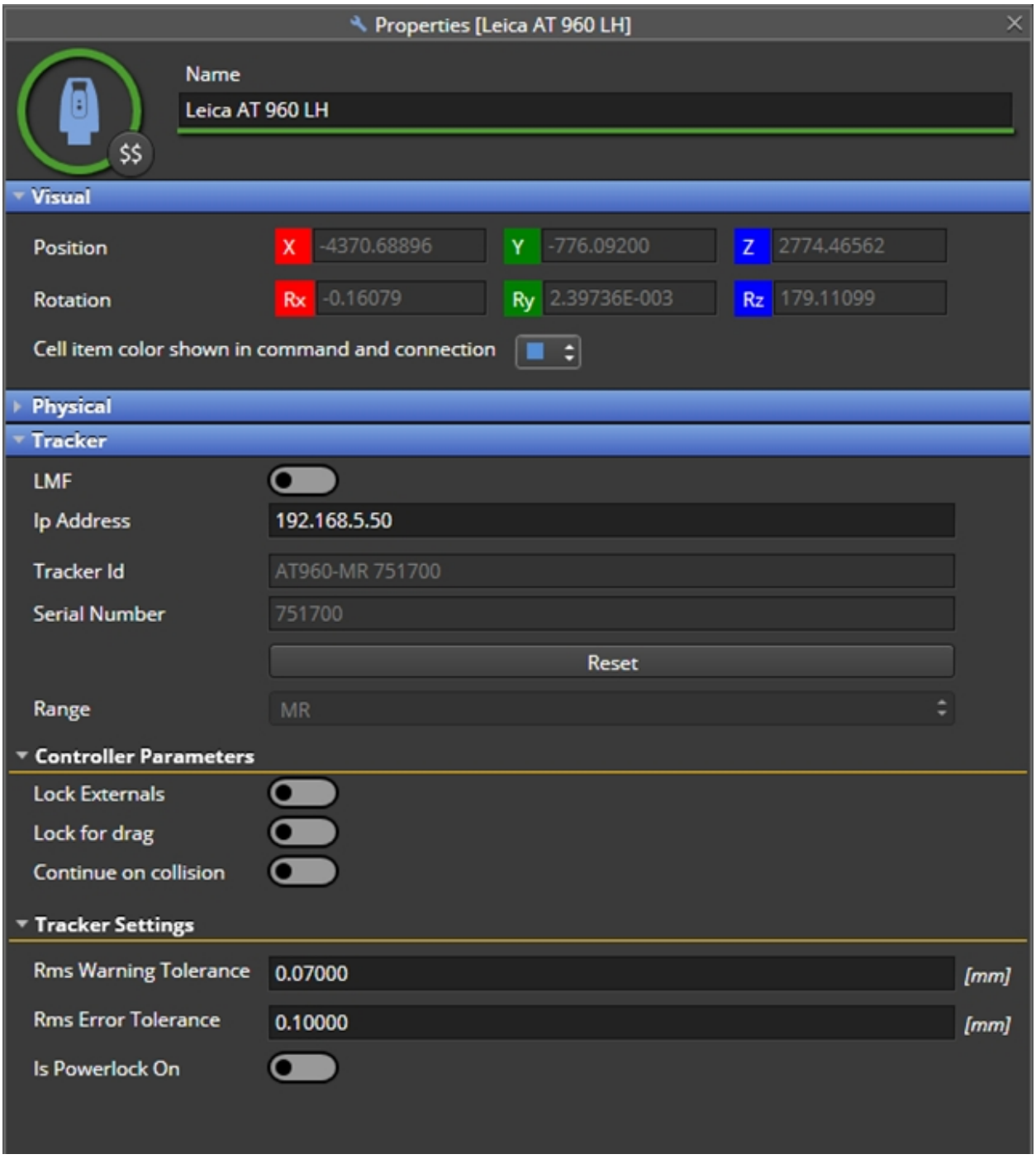
### GoPro

The GoPro window is a viewport from either the tracker or the scanner's point of view. This window can be activated through the teach pendant.


Leica AT 960 LH

## Properties Window

This window will show the properties of the selected item in the cell or program command and in general is used to allow the user to specify the relevant parameters of the item or action that is being executed.


Properties [Leica AT 960 LH]
Name
Leica AT 960 LH
$$
Visual
Position
X -4370.68896
Y -776.09200
Z 2774.46562
Rotation
Rx -0.16079
Ry 2.39736E-003
Rz 179.11099
Cell item color shown in command and connection
Physical
Tracker
LMF
Ip Address
192.168.5.50
Tracker Id
AT960-MR 751700
Serial Number
751700
Reset
Range
MR
Controller Parameters
Lock Externals
Lock for drag
Continue on collision
Tracker Settings
Rms Warning Tolerance
0.07000
[mm]
Rms Error Tolerance
0.10000
[mm]
Is Powerlock On

## Metrology / Collision Engine Buttons

This area has two buttons: starting from the left, the first button is used to connect with a **Metrology application**and the second one to interact with the **Collision engine**.

The buttons have three colors:

1. **Grey**: this means that HxGN Robotic Automation is not connected to the metrology SW or Collision engines is disabled
2. **Green**: this means that HxGN Robotic Automation is connected to the metrology SW or Collision engines is ready.
3. **Red**: this means an error happened connecting with the metrology SW or Collision engines is not ready.

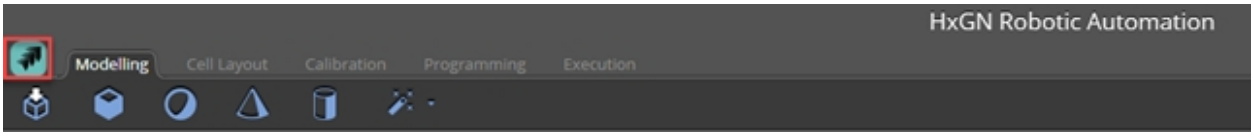
Additionally, in the right part the user can define the measurement unit system between Metric, Imperial or Custom using a drop-down menu.

## Special interactions

- Right clicking menu on the tracker will give a shortcut to some of the existing functions and an additional option to point the tracker at a certain point in the 3D space. The tracker laser will then try to reach the point and stop at any intersecting CAD. Only in case of the ATS800 will we paint the tracker line in the 3D space.

## RA Main Menu Button

The RA main menu can be opened by clicking on the RA icon ( ) on the top-left corner of the screen.




The possible actions are shown in the picture below in the left menu available on light grey background and are self-explanatory.

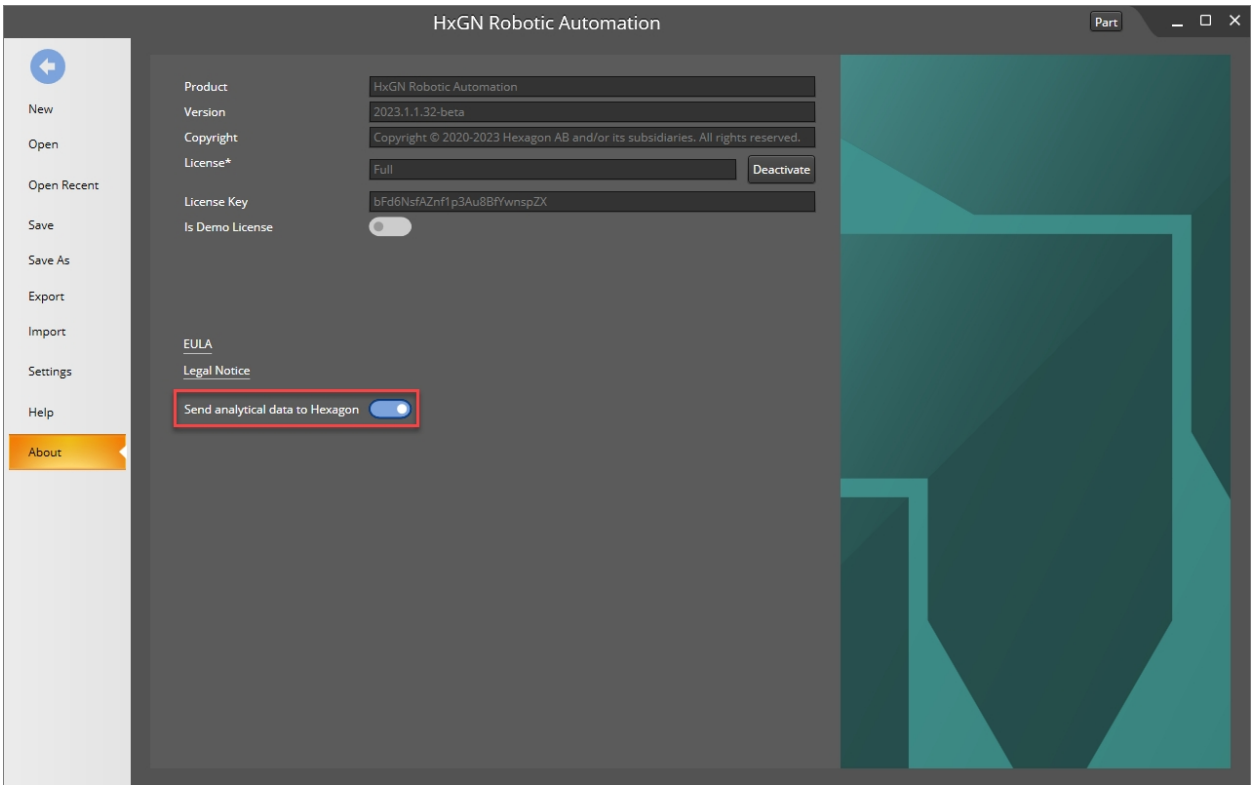
1. **New**, to create a new project
2. **Open**, to open an existing project
3. **Open Recent**, to open an existing recent project
4. **Save**, to save the project
5. **Save As**, to save the project as a new project specifying a new name
6. **Export**, to save the project in a compressed .zip file format
7. **Import**, to extract a project from a compressed .zip file format previously exported.
8. **Settings**, to navigate to the HxGN Robotic Automation settings, described in detail in chapter Settings Menu
9. **Help**, to be redirected to the web help.
10. **About** section has information about the HxGN Robotic Automation version installed and the license key in use.

    User can also consult the EULA and the Legal Notice about 3rd party software in use by clicking the EULA and Legal Notice buttons respectively.

User can opt-in for sending analytical data to Hexagon to improve HxGN Robotic Automation by enabling the relative toggle switch. If enabled, the diagnostics collects crashes and errors, reporting the following data about the event occurred:

- Device Name
- OS
- User ID
- Country
- Language
- Date and Time

When any personal data is collected, such personal data will be processed per section 9.3 of the EULA




## Settings

Within the RA main menu, it is possible to define the global settings of HxGN Robotic Automation in the different sections described below.

#### General

The general settings have folder locations where:

1. **CAD Folder**: store/load the CAD folders.
2. **Library Folder**: store/load the cell item library folders.

#### Collision

To enable collision detection using a switch button.

#### Dragging

Advanced settings for the dragging mode in the 3D Window.

1. **Snapping Angle**: default value is 5°
2. **Snapping Length**: default value is 10 mm.
3. **Snapping Proximity**: default value is 1000 mm.

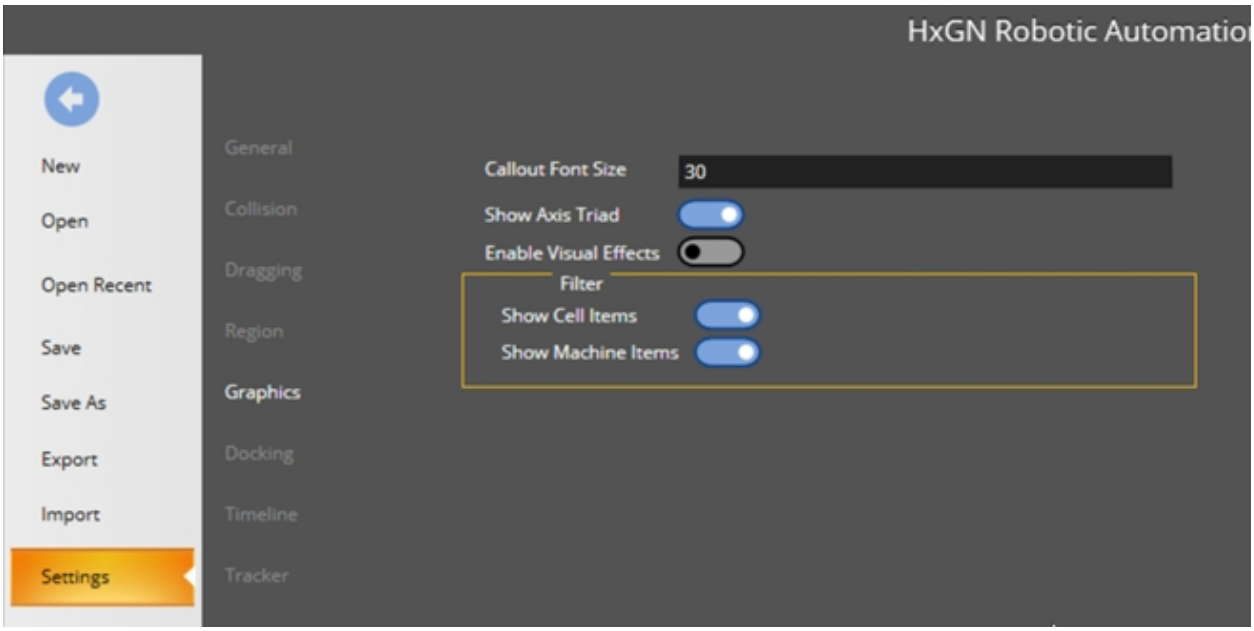
#### Region

To change the Language settings (e.g., English).

#### Graphics

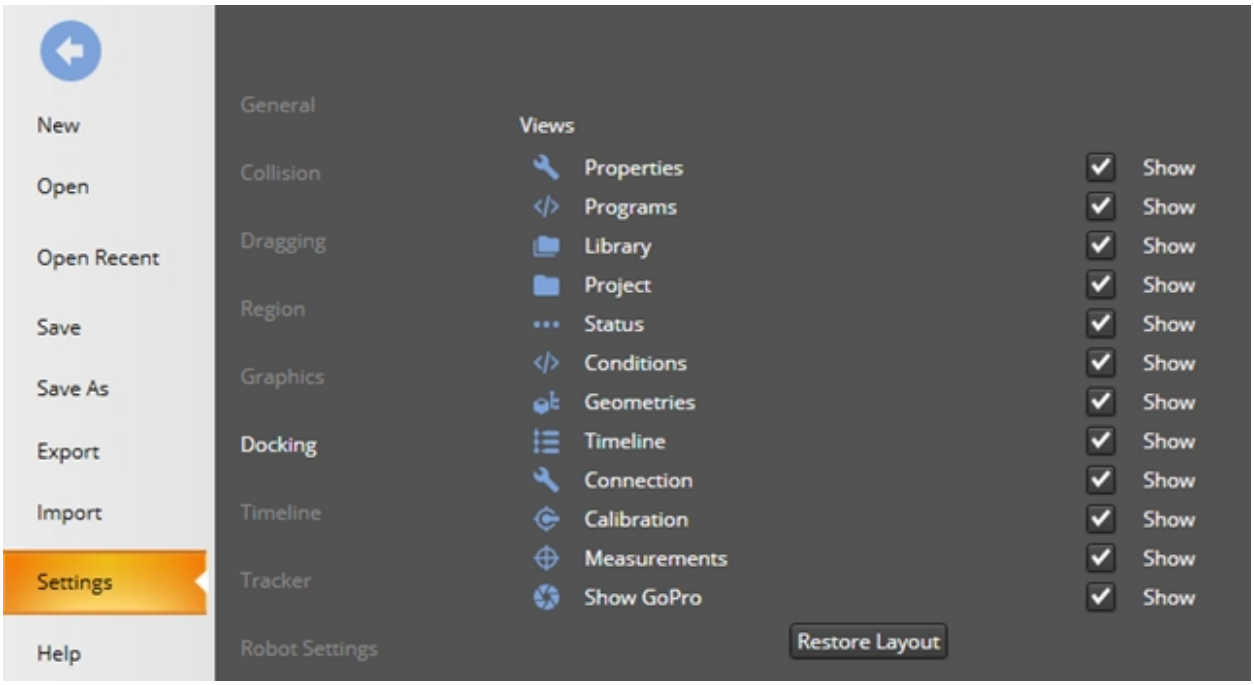
To change advanced graphics settings as depicted in the picture below.

Enable Visual Effect will improve the visual representation of the cell items but may affect the performances.



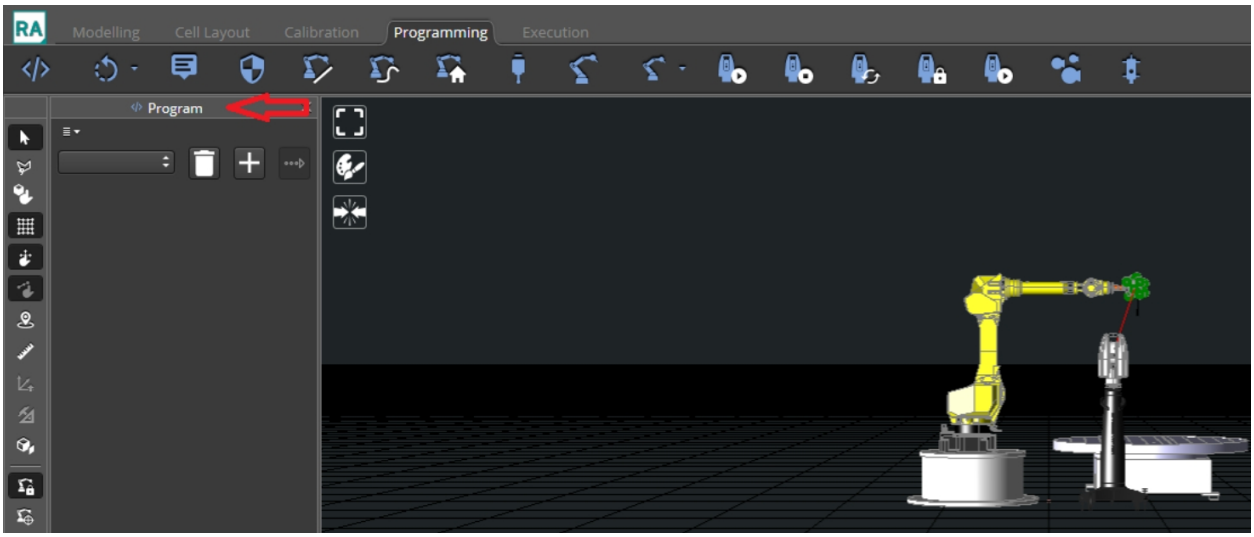

**Docking**

HxGN Robotic Automation application GUI (Graphical User Interface) has several panels showing in different Tabs (Modelling, Cell Layout, Alignment, Programming, or Execution). The user has the option to Show that panel or Restore Layout to default.

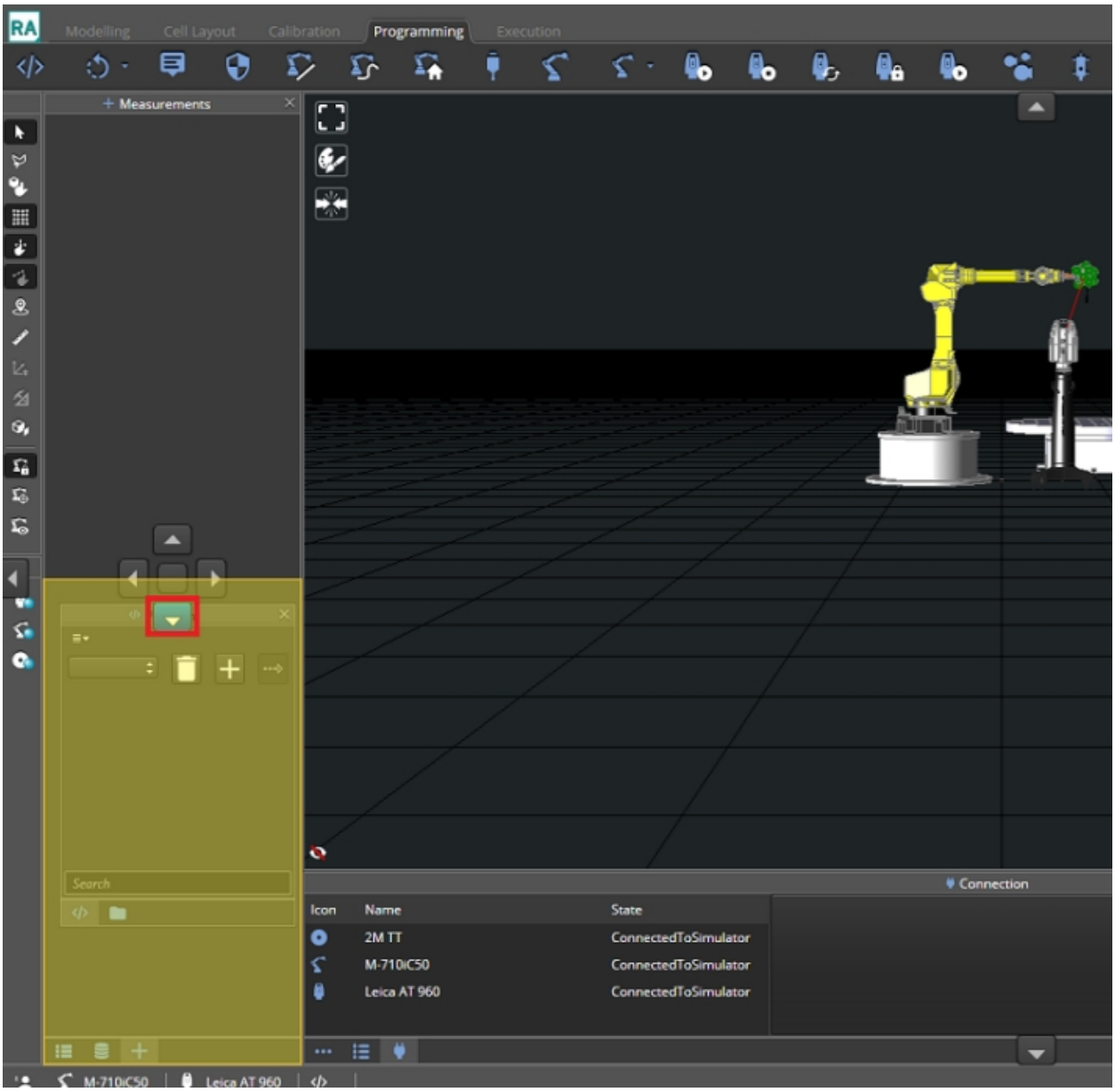



To rearrange a panel, click on the panel with the mouse, hold the left mouse button and drag the panel to any location. Releasing the left mouse button will place it in the new desired location.

For example, if you want to change the programming panel, click where the red arrow with mouse. Hold the left button and drag it; a preview of new location will be shown in yellow, if you position it above one of the arrows:

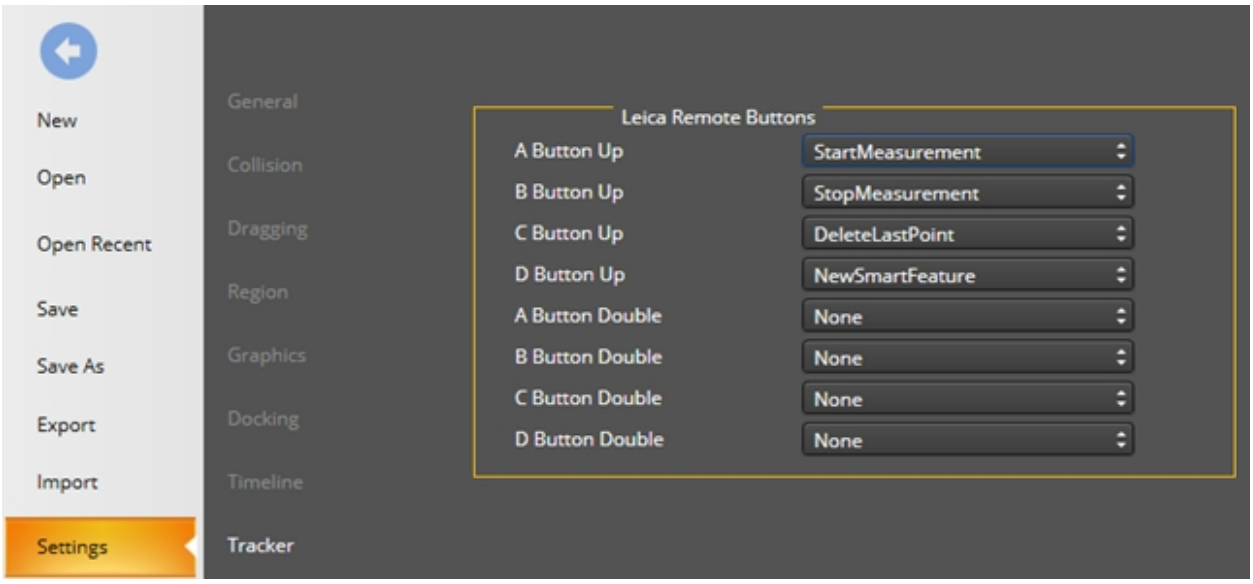



Once you release the left mouse button, it will be positioned in the desired new location. In any stage, use can go to Settings | Docking and click on Restore Layouts.




**Tracker**

This section is used to configure Leica Remote Buttons when using a Leica sensor, e.g., using a T-Probe for starting a measurement by clicking the A Button.




#### Robot Settings

This section allows to change default Joint Motion Speed (10% override) and Linear Motion Speed (400 mm/s).

#### Cloud of Points settings

This section allows to change the COP sparsity (e.g., 1 means that 100% of points will be shown, 10 means 10% of points will be shown) and visualize or hide the point normal vectors.

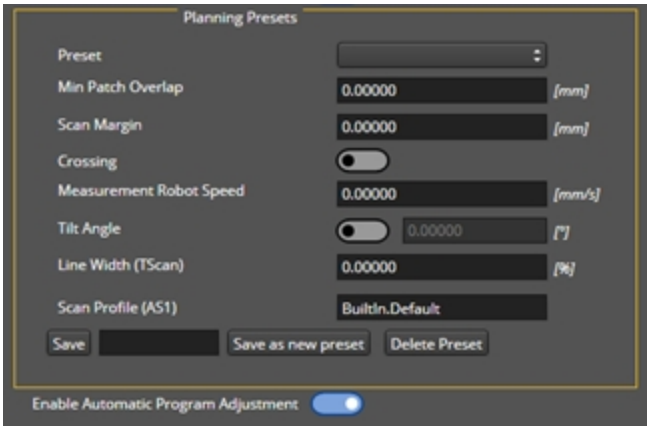
#### Simulation Settings

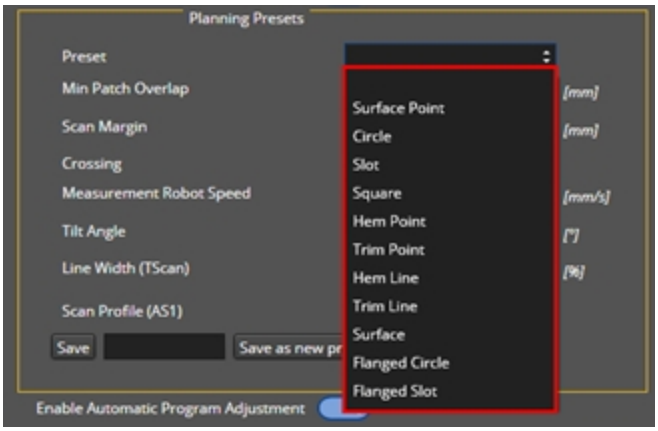
This section defines the simulation accuracy in Hertz. The higher the number, the more “snapshots” our simulation will take. This will mean a longer calculation time and a higher accuracy in collision predictions.

#### Planning

In this area it is possible to define presets for the path planning algorithm to be used to scan optimally the supported metrology features (e.g., circle or slot). It is

also possible to define a new custom preset by defining a name and clicking on Save in case the preset was previously saved or alternatively Save as new preset.




#### Colors

Define the colors used for the various elements of the application.

#### Sounds

Enable Sounds to emit the same sound emitted by the real AT960 sensor when a measurement is collected.

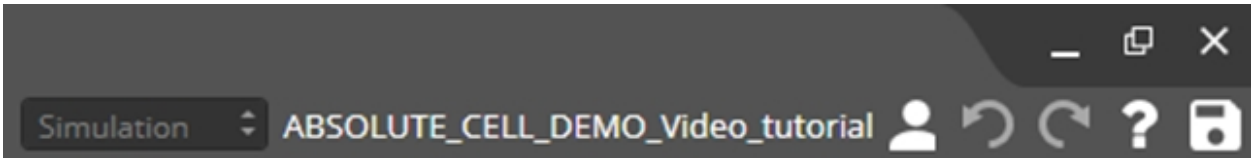
**Import**

Auto rename choice in case the imported project has the same name, following Windows convention.

## Top-right Menu

The top right area allows to connect with simulation or real cell installations using the drop-down menu, user logout , undo and redo changes made on the project currently in use, save it with the relative icon and perform the usual operations as minimize, expand, and close the HxGN Robotic Automation application.

Clicking on the icon, the user is redirected to the web help page.




## Keyboard Shortcuts

| Action | Key Shortcut | Comment |
|---|---|---|
| Copy | Ctrl + C | |
| Paste | Ctrl + V | |
| Add Program | Ctrl + P | |
| Add Linear Motion | Ctrl + L | |
| Add Joint Motion | Ctrl + J | |

| Action | Key Shortcut | Comment |
|---|---|---|
| Zoom Fit | Cltr + F | On selected item. If no item is selected, general zoom. |
| Save | Ctrl + S | |
| Save As | Ctrl + Shift + S | |
| Open Project File | Ctrl + O | |
| New Project File | Ctrl + N | |
| Undo | Ctrl + Z | |
| Redo | Ctrl + Y | |
| Cancel Action | Esc | |
| Confirm Selection | Enter | |
| Remove Selected Item | Del | |
| Rename | F2 | |
| Reset Measurement | F3 | |
| Start Measurement | F4 | |

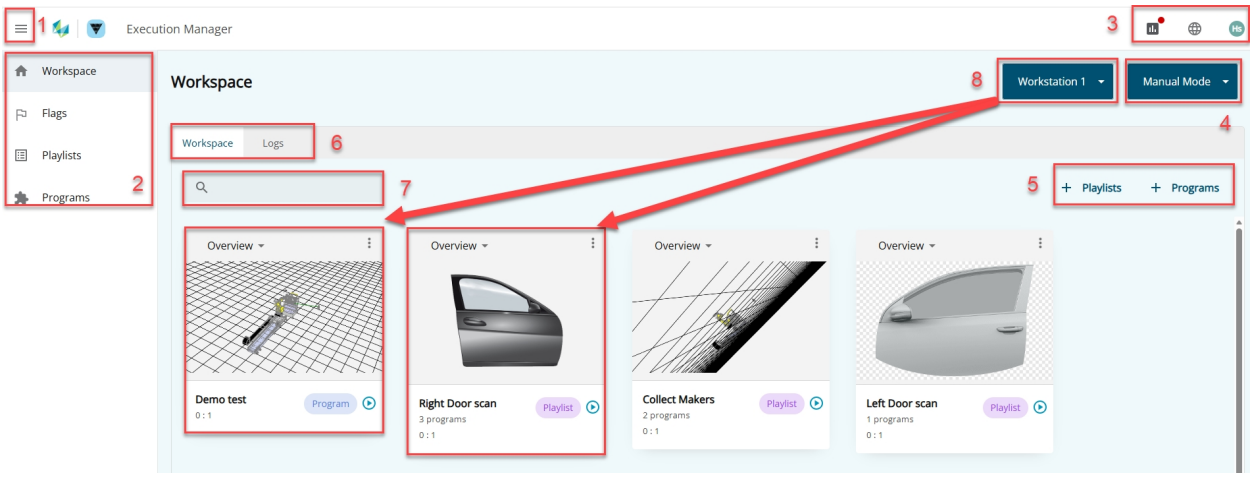
| Action | Key Shortcut | Comment |
|---|---|---|
| Execute | F5 | |
| Associate Measurement | F6 | |
| Execute Next Step | F10 | |
| Select Group | G | Hold G while on 3D window and it will select the group, not the item. |
| Execute Command | Tab + Double-Click | While on programing or execution modes |
| Clone Item | Alt + Drag | |
| Expand Tree | Ctrl + "+" | |
| Collapse Tree | Ctrl + "-" | |
| Scoll up program line | Page up / Mouse Wheel Up | |
| Scroll down program line | Page down / Mouse Wheel Down | |
| Move to the first program line | Home | |

| Action | Key Shortcut | Comment |
|---|---|---|
| Move to the last program line | End | |
| Pick on CAD Rotate | Shift + Drag | |
| Pick on CAD Flip Normal | Alt + Drag | |
| Range Selection | Shift + Click | |
| Multiple Selection | Ctrl + Click | |
| Drag/Rotate by a Fixed Step | Shift + Drag | Rotates at 5° for rotation and 10 mm for translations |

## HxGN Execution Manager

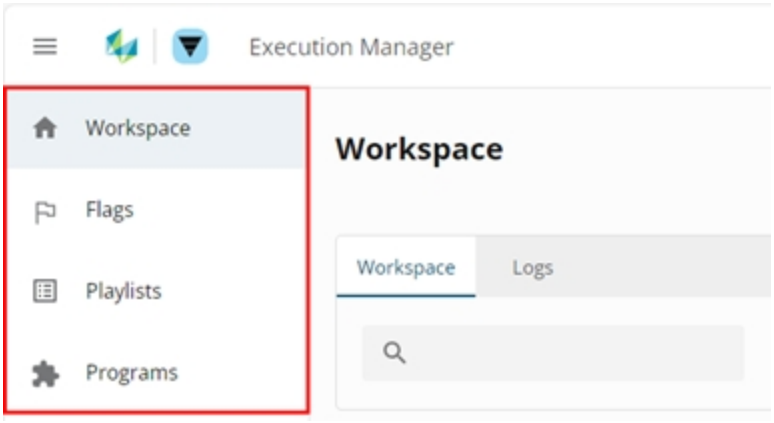
The Execution Manager UI is divided into ten principal areas as depicted in the following picture.

At the first execution the Home page is shown, and pages can be navigated from the left side bar.




1. **Toggle Side bar** expands or collapses the side bar by clicking the burger menu.
2. **Side bar** to navigate among the following pages:
    a. **Workspace page**: page that allows to select and run desired program or playlists, see the cell status and switch between automatic and manual mode.
    b. **Flags page**: allows to configure the flags and associate them to playlists. Execution Manager running in automatic mode will listen to the flags set by the PLC to execute the associated playlist.
    c. **Playlists page** allows to create playlists made of existing programs.
    d. **Programs page** shows a list of all the programs in the database and allow the user to import programs previously exported with HxGN Robotic Automation version 2.1.2 and newer. Additionally, the user can

edit the metadata of a program.

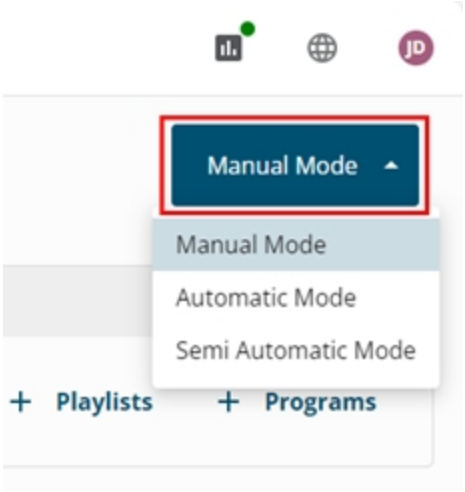



**Top-right menu**: to check the **Cell Status** (status of the cell items), change the **Language**, change the **User**.

3.


4. **Manual/Automatic/Semi Automatic Mode toggle** to switch between manual or automatic modes:
    a. In manual mode programs and playlists can be executed in the order defined by the user.
    b. In automatic mode the cell PLC will control the execution of the playlists by sending the relative Flag Code to the Execution Manager that will automatically select the corresponding playlist to be executed (unattended cell).
    c. In semi automatic mode the cell PLC will control the execution of the playlists by sending the relative Flag Code to the Execution Manager that will automatically select the corresponding playlist to be executed only

after user confirmation (attended cell using e.g., barcode reader to identify the part).

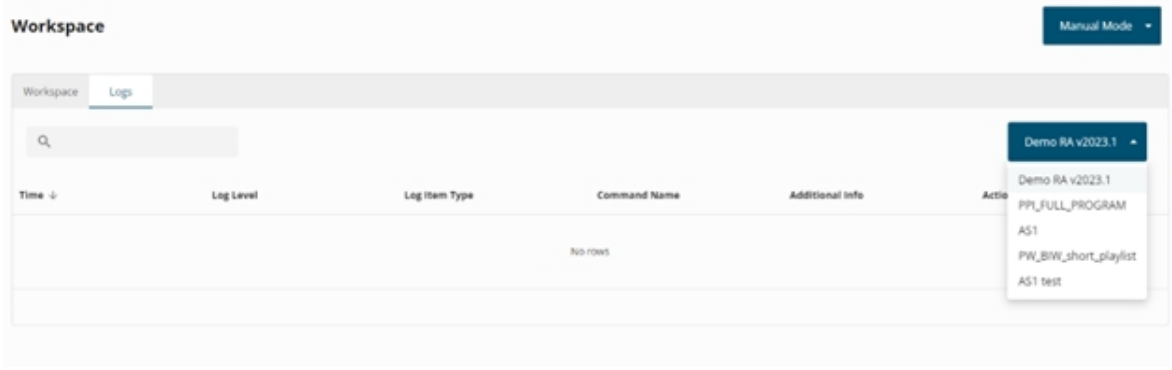



5. Add a playlist or Add a program to the home page.

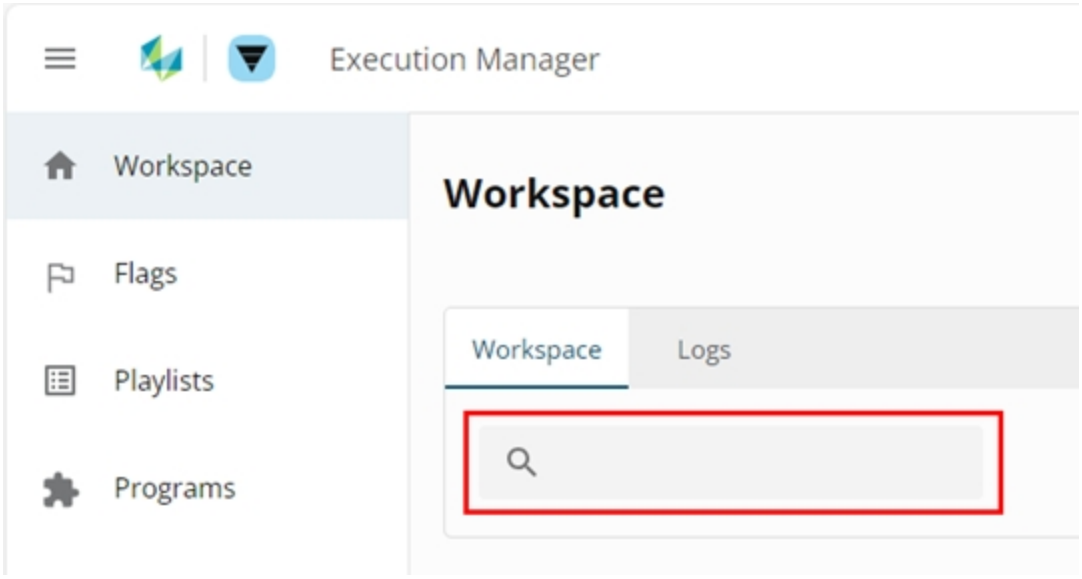
   **Workspace** vs **Logs** choice to switch between Workspace and Logs view.

   If **Workspace** is selected, programs and playlist added will be shown.

   In case **Logs** is selected, the execution logs of programs and playlists will be shown as picture below.

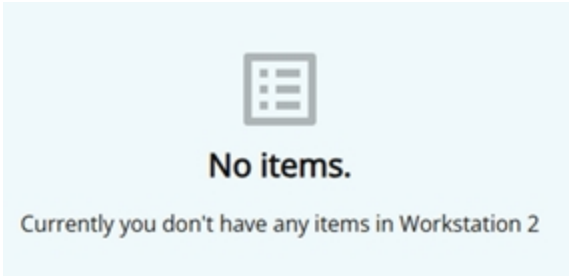
6. 

**Search bar** allows the user to search and filter within the current tab by name or part number.

7. 


**Workstation Selection Dropdown** allows the user to select the workstation to work with.

If the selected workstation doesn't contain any programs or playlists, this will be displayed in a message.

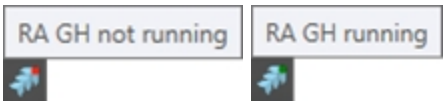
8. 


## HxGN Robotic Automation App Manager

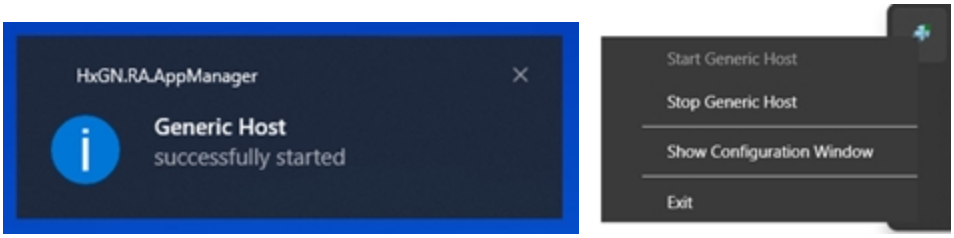
Robotic Automation App Manager is a small utility that can be used to

- Easily check the status (running / not running) of the Generic Host (the core service manager that is responsible to coordinate all the services used by HxGN Robotic Automation and Execution Manager.)
- Enable or disable running services
    - Change settings of the HxGN Robotic Automation

## Right Click menu and Basic Tab




Robotic Automation App Manager is started automatically when windows user logs in and its icon ( ) can be found the taskbar.



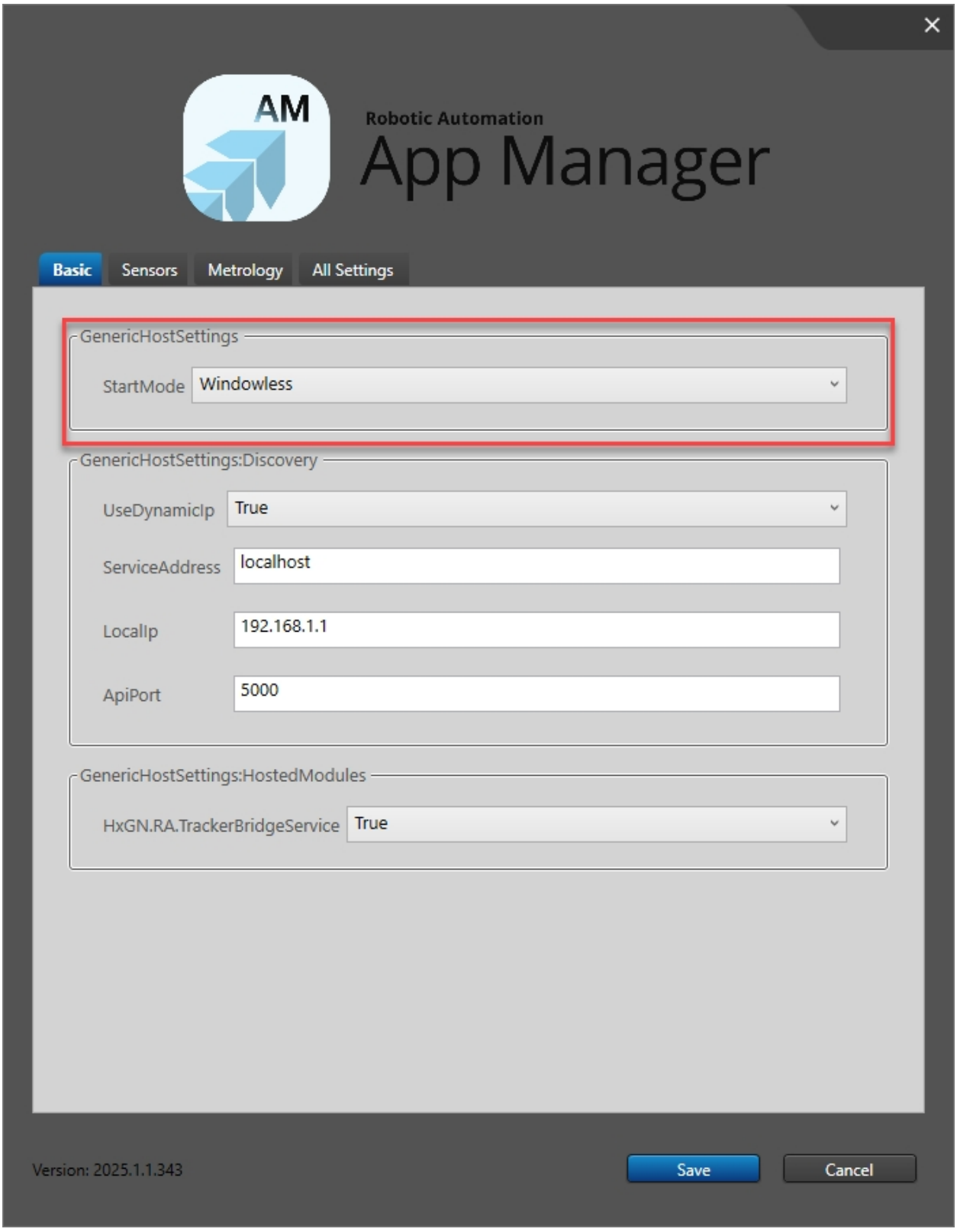

By right-clicking on the icon, the user can Stop/Start Generic Host and show the configuration window where it is possible to select on of the three modes in which Generic Host can be executed:

1. **Windowless**: Generic Host is executed as a console application without showing the console window. This is the default value on new installation, and it allows to communicate with the RDS agent but does not allow to interact with it.

This choice is the preferred one if working in online programming mode thus connected to a robotic cell and not requiring interaction with RDS agent.

It requires user login to start.

2. **Console**: Generic Host is executed as a console application as above but showing the console window. This is useful for easier debugging of issues in Generic Host and to interact with the RDS agent. This choice is the preferred one if working in online programming mode thus connected to a robotic cell and requiring interaction with RDS agent.
3. **Windows service**: Generic Host is executed as Windows service. This choice is the preferred one if working in offline programming mode and not connected to a robotic cell. Does not require user login to start and it is started at Windows startup, without need for the user to login.


AM
Robotic Automation
App Manager
Basic
Sensors
Metrology
All Settings
GenericHostSettings
StartMode
Windowless
GenericHostSettings:Discovery
UseDynamicIp
True
ServiceAddress
localhost
LocalIp
192.168.1.1
ApiPort
5000
GenericHostSettings:HostedModules
HxGN.RA.TrackerBridgeService
True
Version: 2025.1.1.343
Save
Cancel

After changing the execution mode, either Windows needs to restart, or user needs to log off to apply it.

The other settings will be explained in the configuration module. Configuration and Licensing.htm.

# Modelling

This section is aimed for **Application Engineers**. Users do not have to model robotic cells; they usually start using HxGN Robotic Automation already configured to work on the robotic cell installation where it is installed.
To be able to use the modelling capabilities, the users must have a valid Full license.A Lite license type will not allow to use the modelling capabilities of HxGN Robotic Automation

## Machine Creation

HxGN Robotic Automation application has the capability to model machines which include the following:

1. **Slider**: machine extension, which includes a fixed part and linear moving part to supply more DOFs (degree of Freedom) either to a robot or a tracker.
2. **Robot**: industrial anthropomorphic robots:
    a. Six axes robot (e.g., Kuka)
    b. Six axes coupled robot – where there is a dependency between joint J2 to joint J3. (e.g., Fanuc)
3. **Rotary Table**: part holder, which includes a fixed part and a rotating moving part.

Prior to building the model the following must be prepared:

1. **CAD file** (format such as STP, JT, IGS, etc.) that must be capable of being separated into groups for each joint.

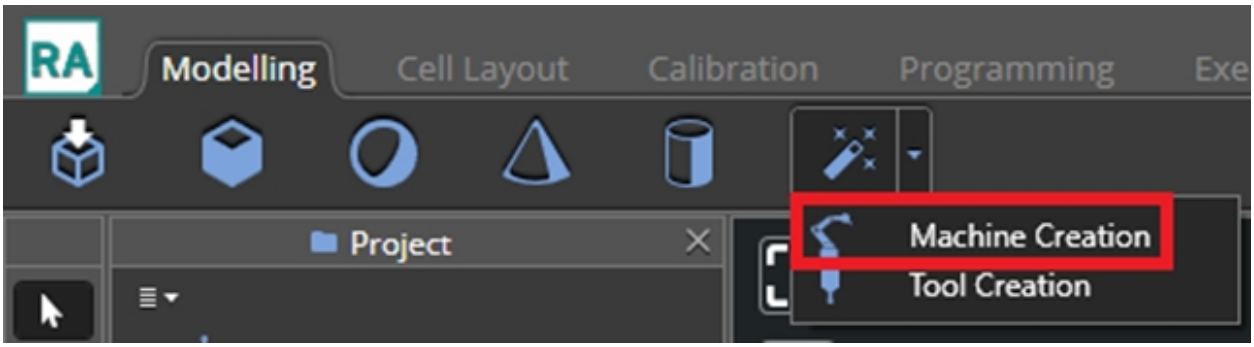
2. **Model name**, data sheet specification of the Robot or Part holder or Robot extension. This is usually a PDF file which has the specification drawing and dimensions, joint ranges, and velocities.

   When defining joints range parameters such as Min and Max, pay attention that it is not allowed to have joints limitations which are wider/higher than the limitation of the real robot.

3. **DH parameters** (for further explanation see chapter D-H_Parameters).

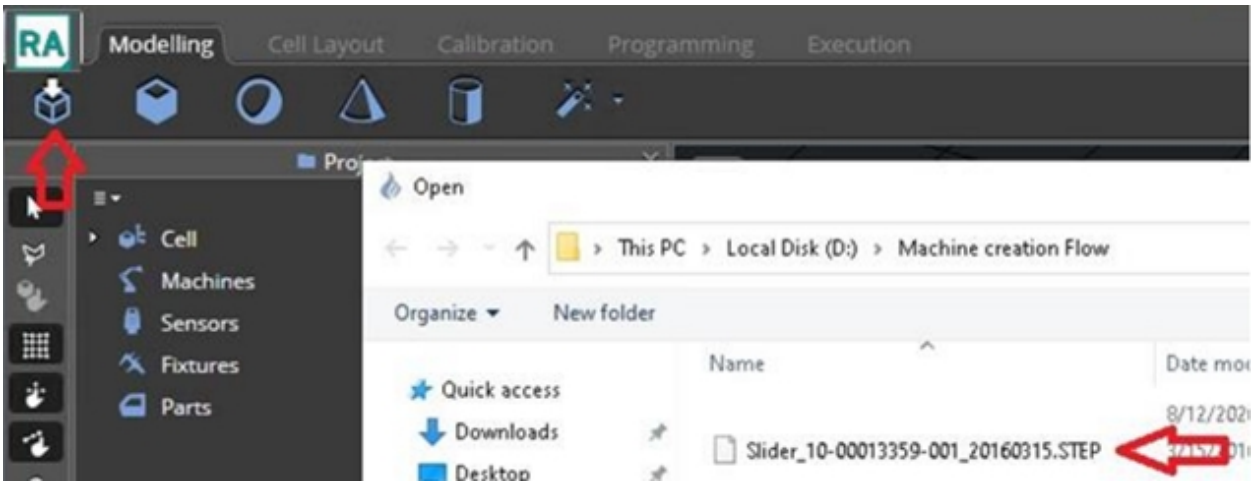
4. **Offset Transform parameters** (usually needed for slider).

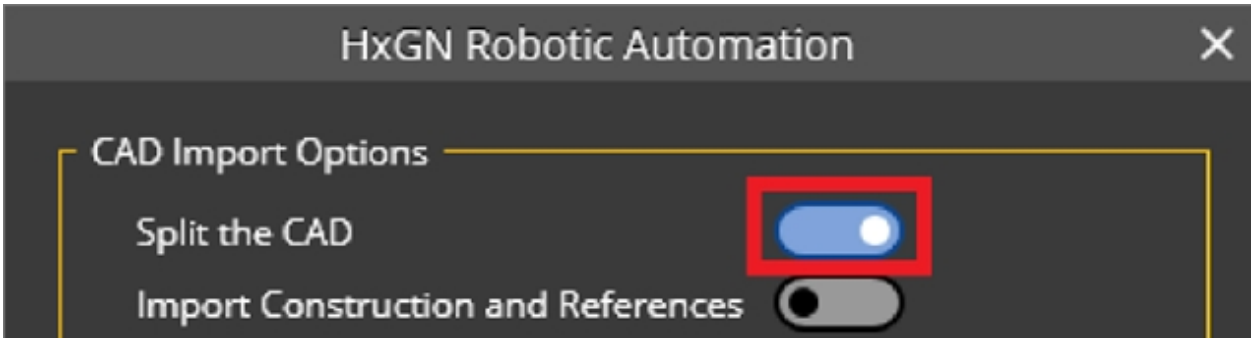
HxGN Robotic Automation also supplies a **Machine Creation** wizard to simplify the creation of machines.




## Slider

Open a new project with Robotic Automation application and under **Modelling** tab, click on the **Import CAD** icon to add the slider's CAD file and check **Split the CAD** switch (disabled by default).

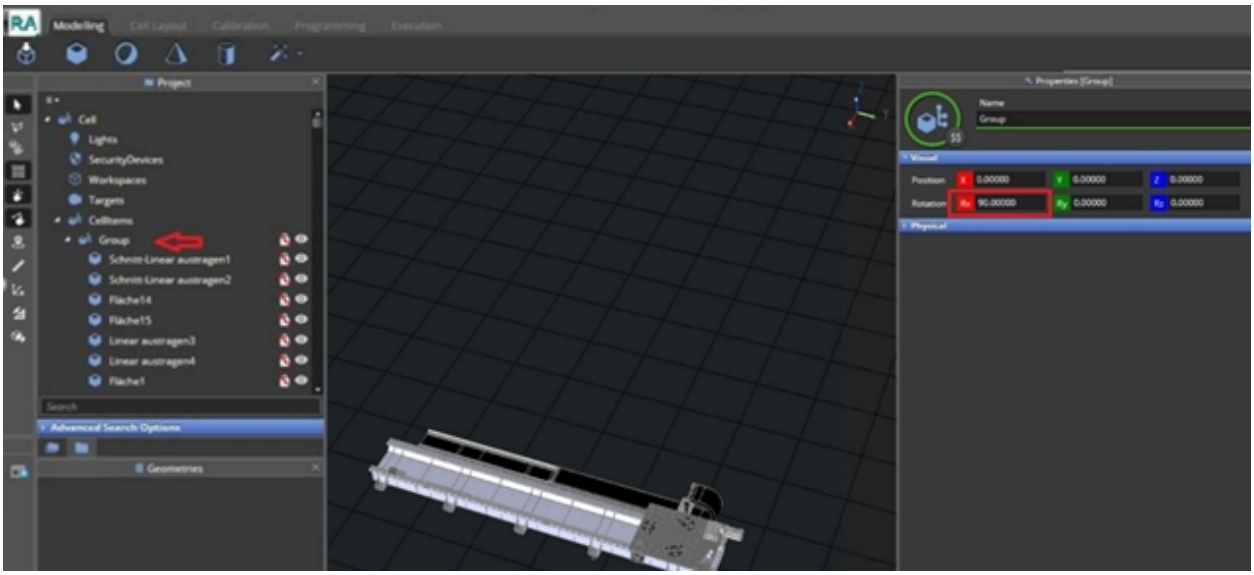



The CAD will show under **Cell items** as a **Group**. The CAD coordinate system may need to be changed. Double click on the **Group**, the properties box will open on the right, edit the **RX**, **RY** or **RZ** fields, and make sure the slider moving part is up to Z direction (see the example below, RX should be 90°).

---

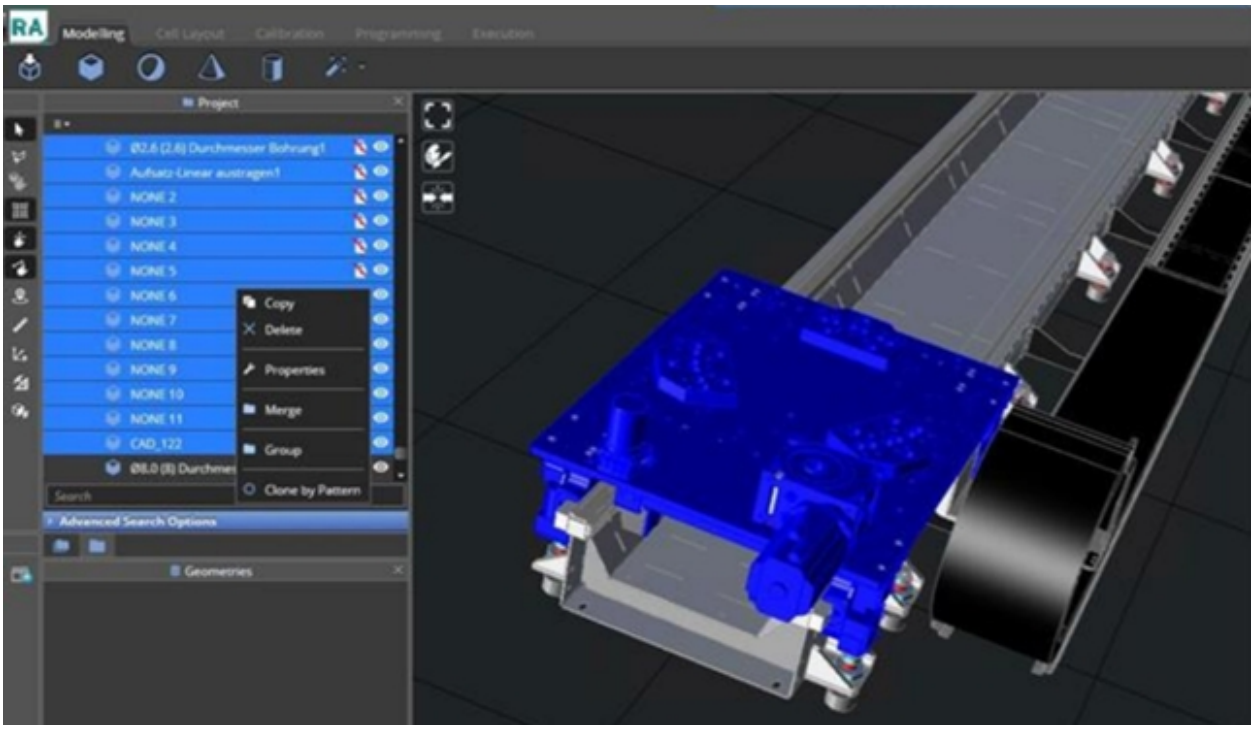
| | |
|---|---|
| Modelling | 200 |

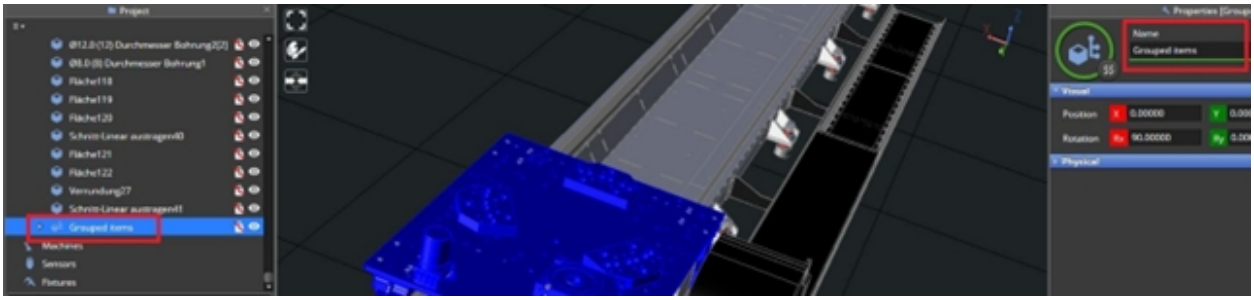



Within the group it is possible to remove unnecessary CADs and divide into two groups: fixed and moving.

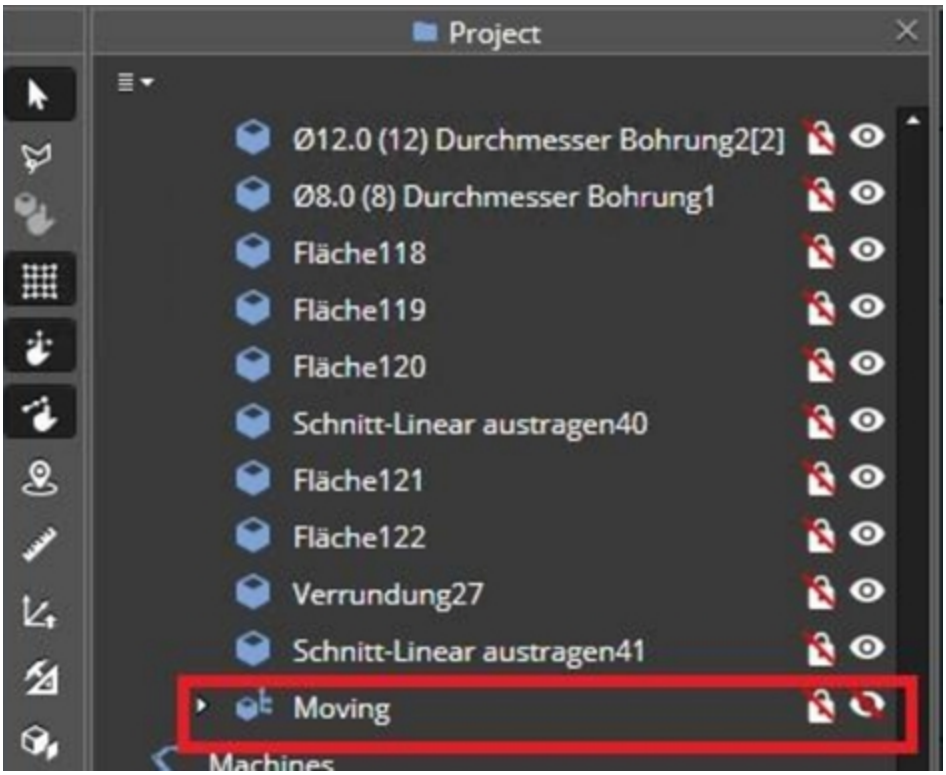
After grouping correctly, the CADs in each group must be merged by right clicking on the Group and selecting **Merge**.

User can find the CAD elements, and by selecting the [CTRL] Key, right click and Group. Grouped CADs will be moved into **Grouped Items** in the project tree and can be renamed as “Moving” by double click and rename them in the Property window.



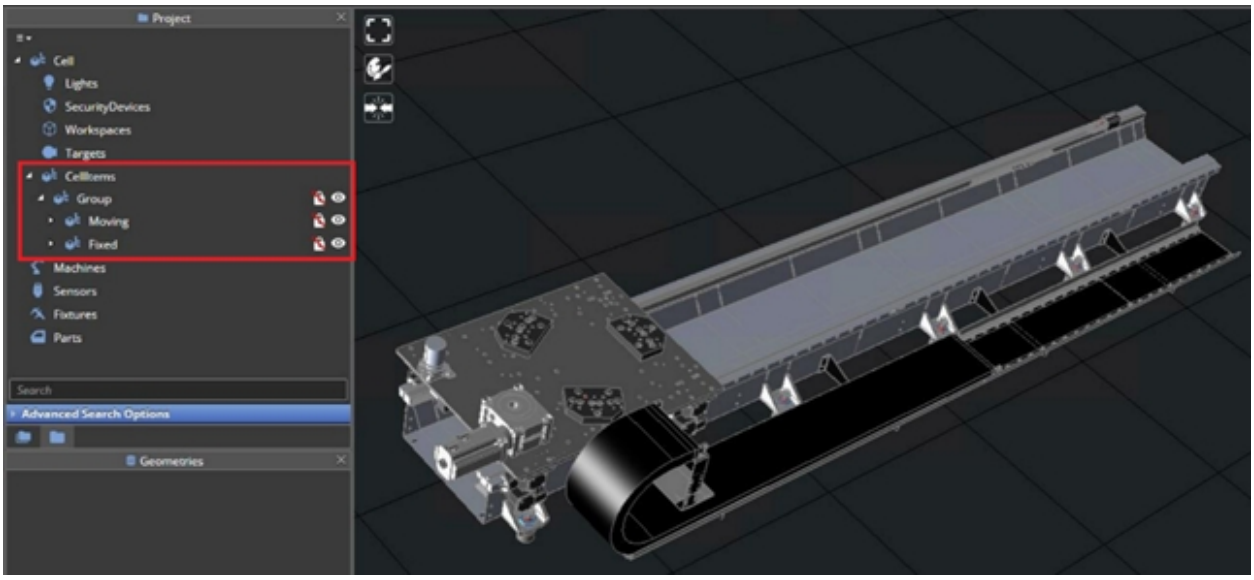




To group the fixed parts, user must hide the new “Moving” group that was just created.




After removing all remaining unnecessary CADs, the user must select the remaining CAD items, group them, and rename the new group to “Fixed” in properties as done before.

The Moving group can now be shown again, and the two groups will be visible within the project tree.



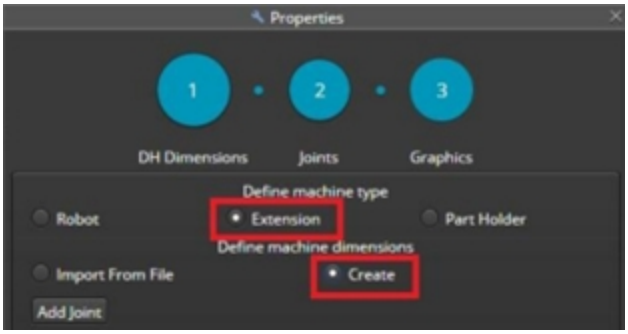

To complete the modelling, the two groups can be merged by right-clicking, selecting Merge, and renamed again as “Moving” and “Fixed” as done in the earlier step.

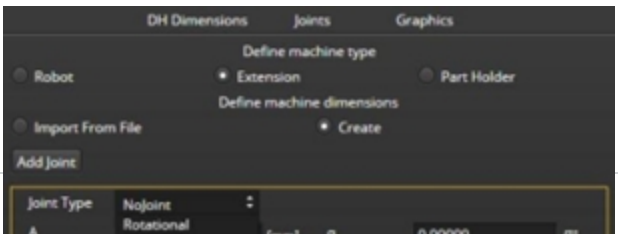
If CAD orientation has changed, it can be fixed by changing RX or RY in the Properties window.

## Slider creation wizard

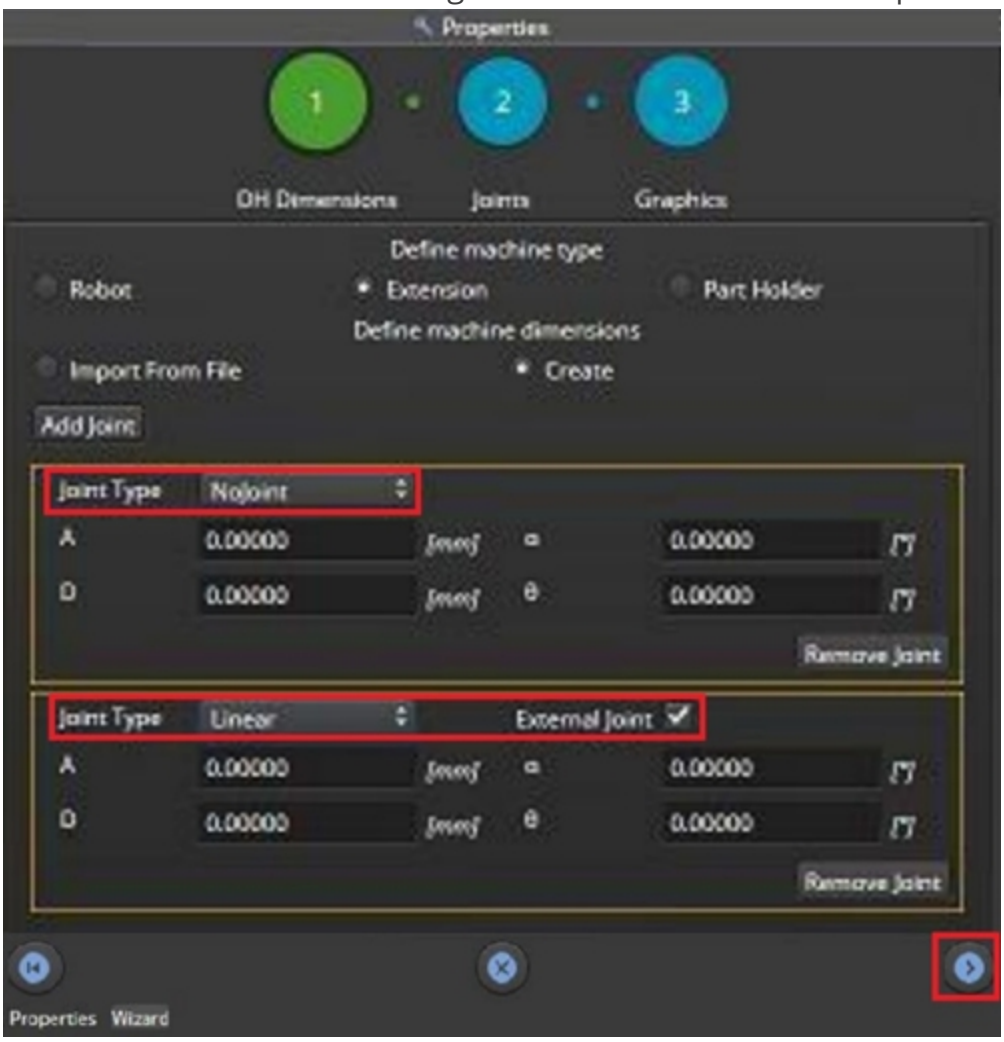
After clicking on the **Machine Creation** wizard, on the Properties window:

1. Select **Extension** as machine type, select **Create** as machine dimensions, then click **Add Joint** and select **No Joint** from the drop-down menu.



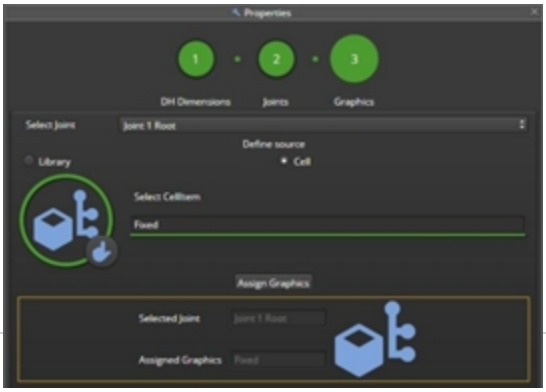
Click again on **Add Joint** to add another joint selecting Linear from the drop-down menu for the moving joint and checking **External Joint**. Once done, click on the arrow on the bottom right to continue to the next step.

2. 


3. Define joints parameters such as Min and Max velocity. This step can be skipped and change later but always make sure that it is not allowed to have joints limitation which are outside of the real machine joint limits. Click again

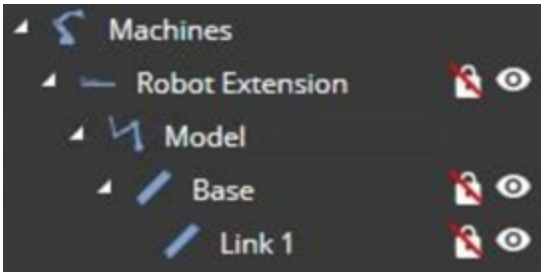
on the bottom-right corner to continue with the next step.

4. Assign graphics: from the **Select Joint** drop-down menu select Joint 1 Root first and then set **Define source** to Cell. Click on the hand icon to select the graphics and select the **Fixed** group under **Cell Items** group. It will turn its color to green if the operation is successful and then by hitting the [ENTER] key the assignment will be confirmed. Click on **Assign Graphics** once you see the item was selected and you will see the assigned items in the list.

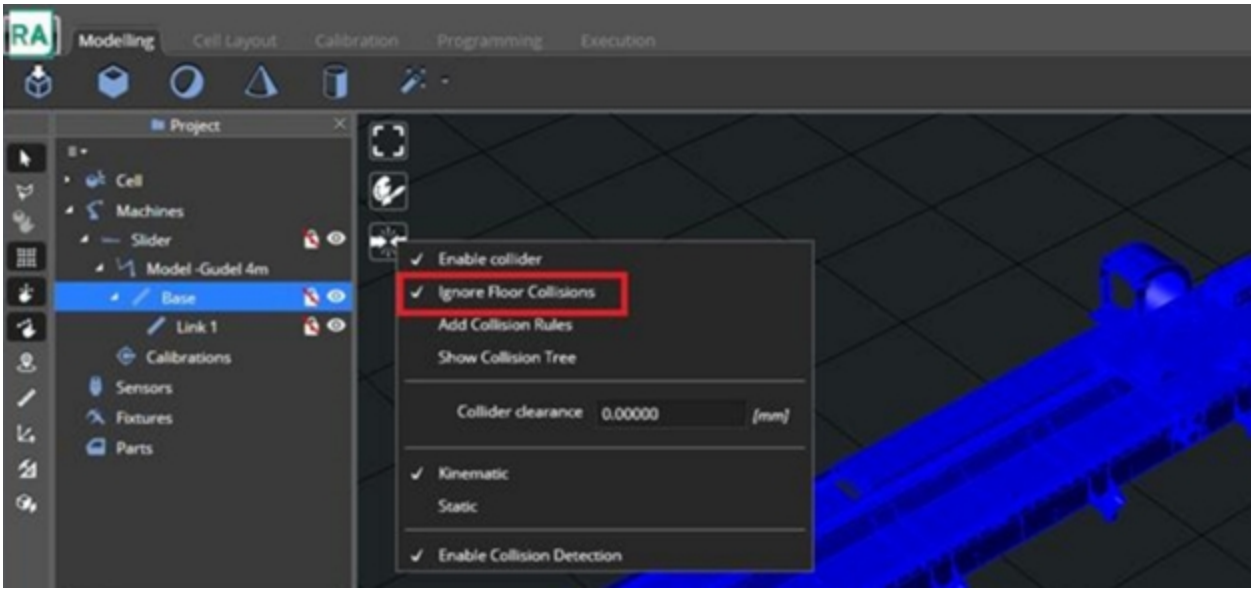


You can continue with the same steps for the other joint (**Joint 2 External**) by selecting it from the **Select Joint** drop-down menu and clicking on the Finish icon ( ) to complete the process.

At the end of the process under **Machines** in the project tree you will find the new machine. You can rename the Robot extension and Model names as you prefer.

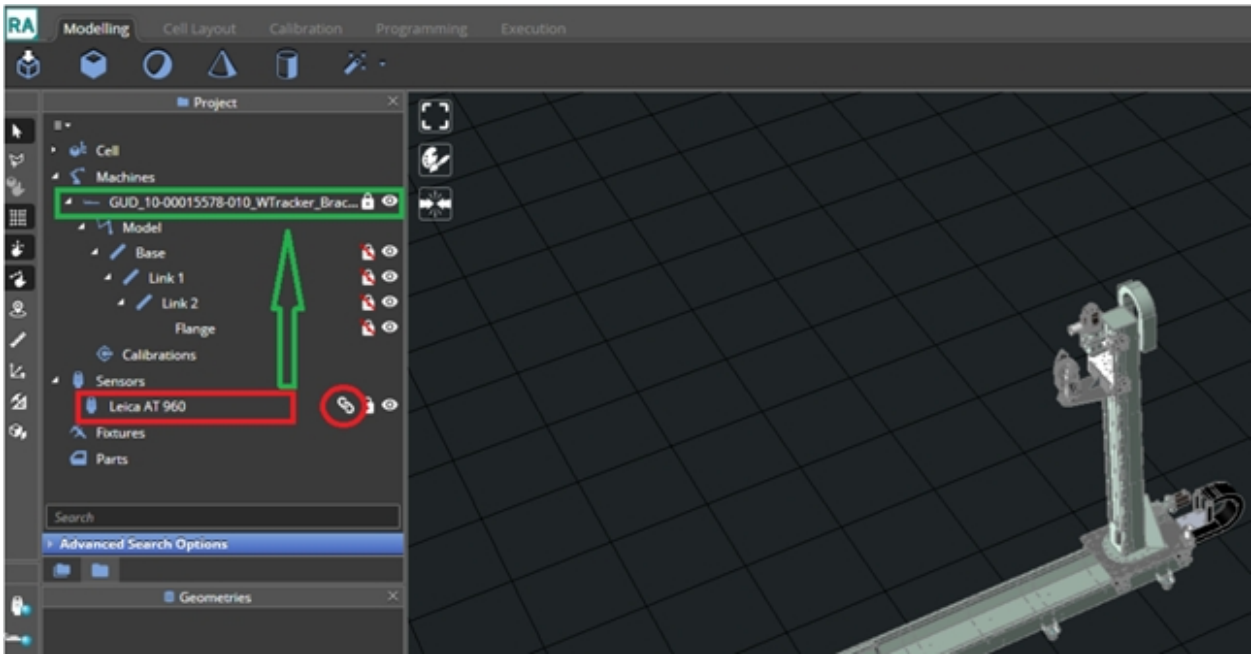
5. 


In case you have a collision with the floor, you can select the slider **Base**, click on **Collision** icon and check **Ignore Floor Collisions**.

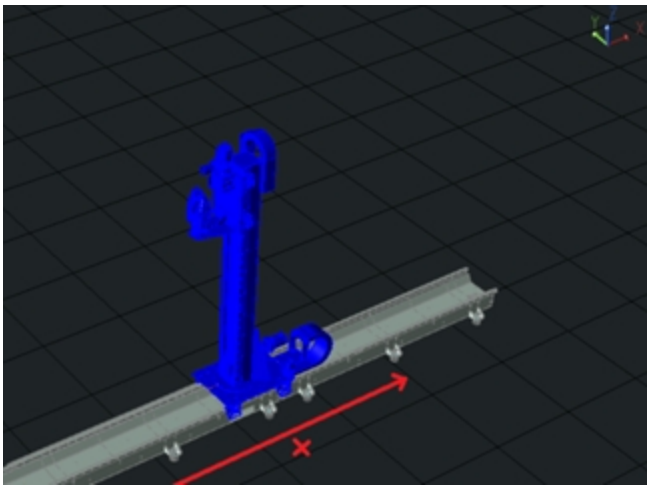


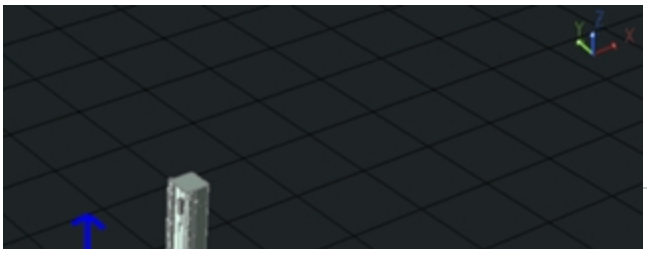
## Tracker on slider

It is possible to model a slider (either with one or two axes) and link the Tracker to that slider by dragging and dropping it to the slider as depicted below.

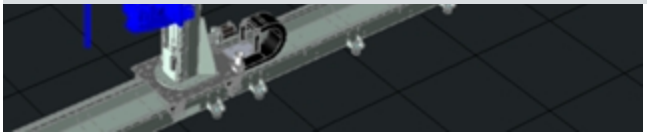



In this way the tracker can move together with the slider in X and Z directions.



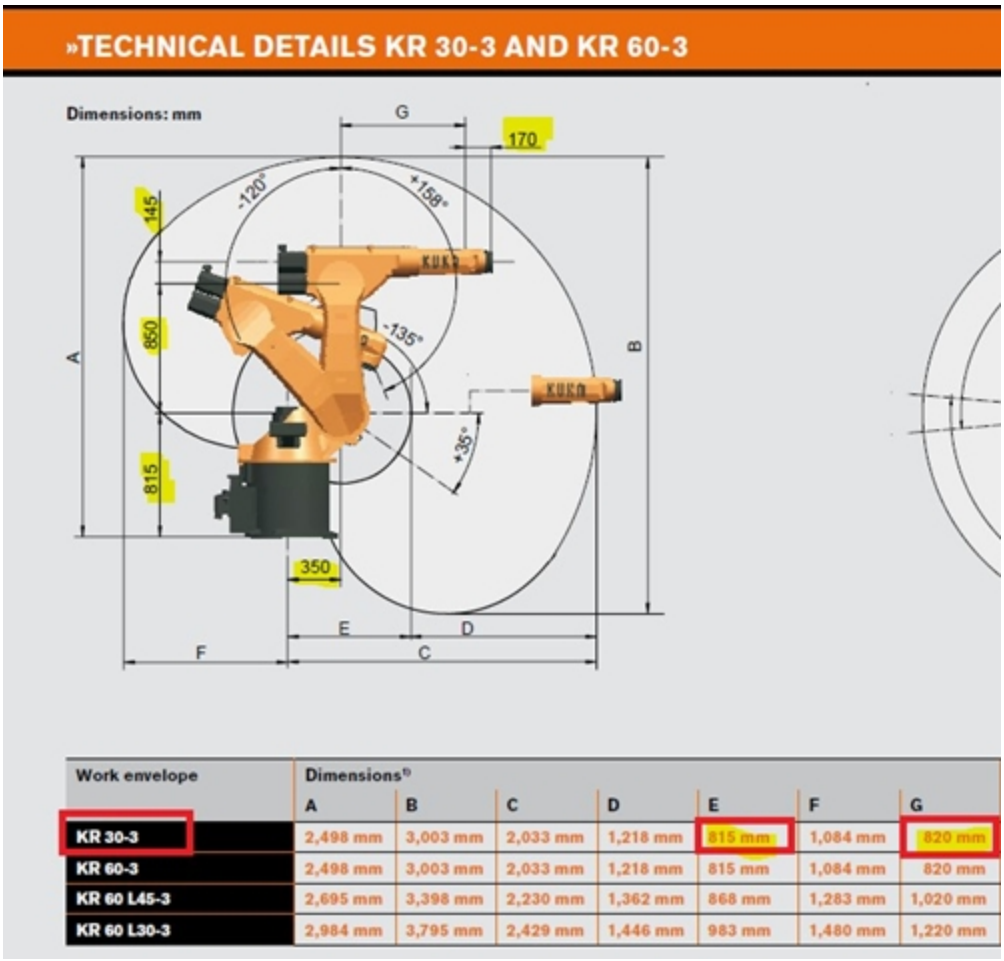







## Robot

Prior to Robot creation, it is needed to prepare the CAD and the specification data sheet to extract the D-H parameters. In the following, an example using a KUKA KR 30-3 robot will be used to show how to extract the D-H parameters from the robot data sheet.

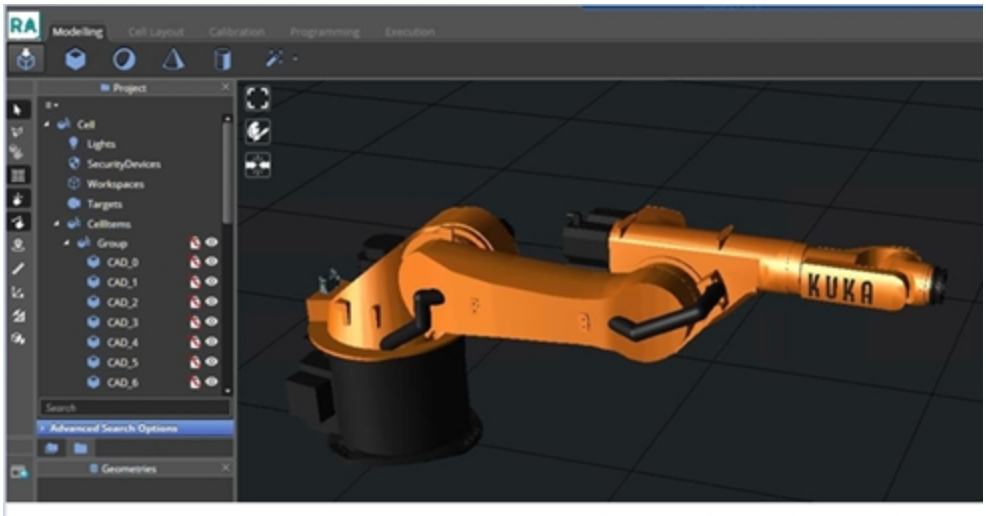



| Joint/Link | R(A) | D | α | θ (zero offset) |
|---|---|---|---|---|
| Base – Link 0 | 0 | -815 | 180 | - |

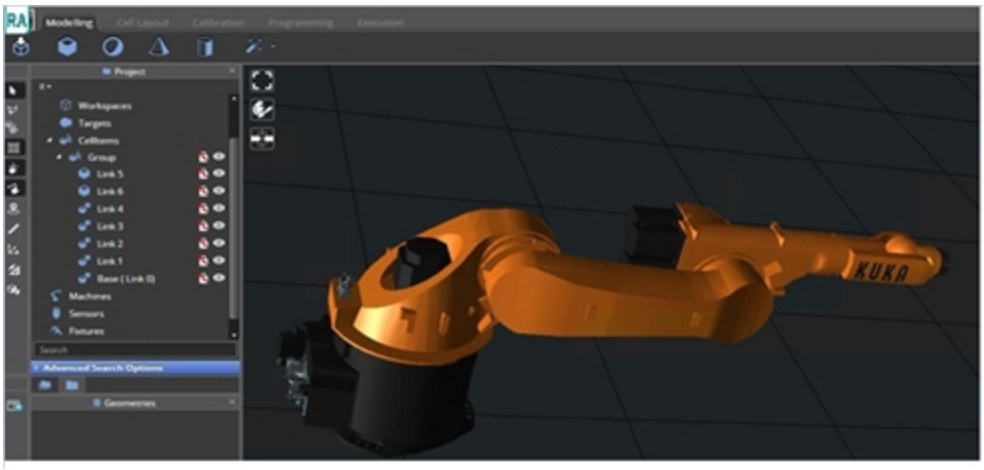
| Joint/Link | R(A) | D | α | θ (zero offset) |
|---|---|---|---|---|
| J1 – Link 1 | 350 | 0 | 90 | 0 |
| J2 – Link 2 | 850 | 0 | 0 | 0 |
| J3 – Link 3 | 145 | -820 | 90 | -90 |
| J4 – Link 4 | 0 | 0 | -90 | 0 |
| J5 – Link 5 | 0 | -170 | 90 | 0 |
| J6 – Link 6 | Transform offset | - | - | - |

Open a new project with Robotic Automation application and under **Modelling** tab, click on the import icon to add the robot's CAD files.

Delete unnecessary CAD, separate, and classify them into merged cad group names as follows: Base (Link 0), Link 1, Link 2, Link 3, Link 4, Link 5, Link 6.



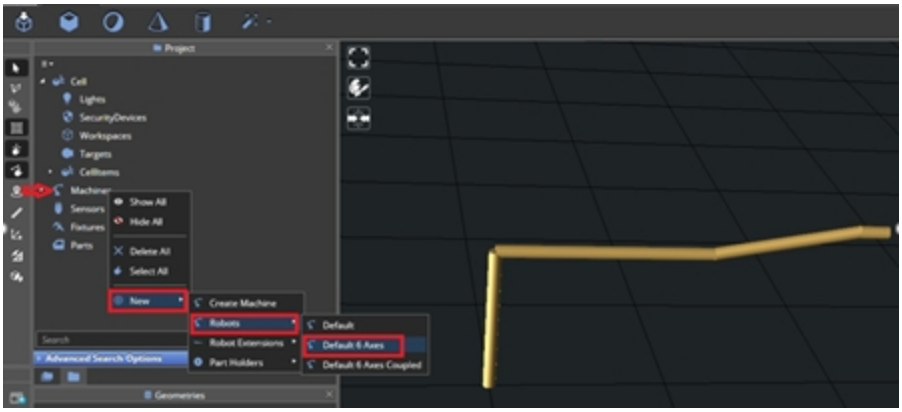

---

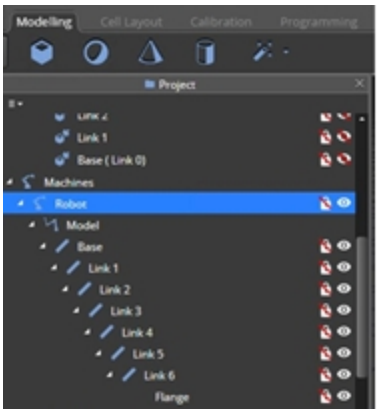



A robot can be created manually or following the robot creation wizard.

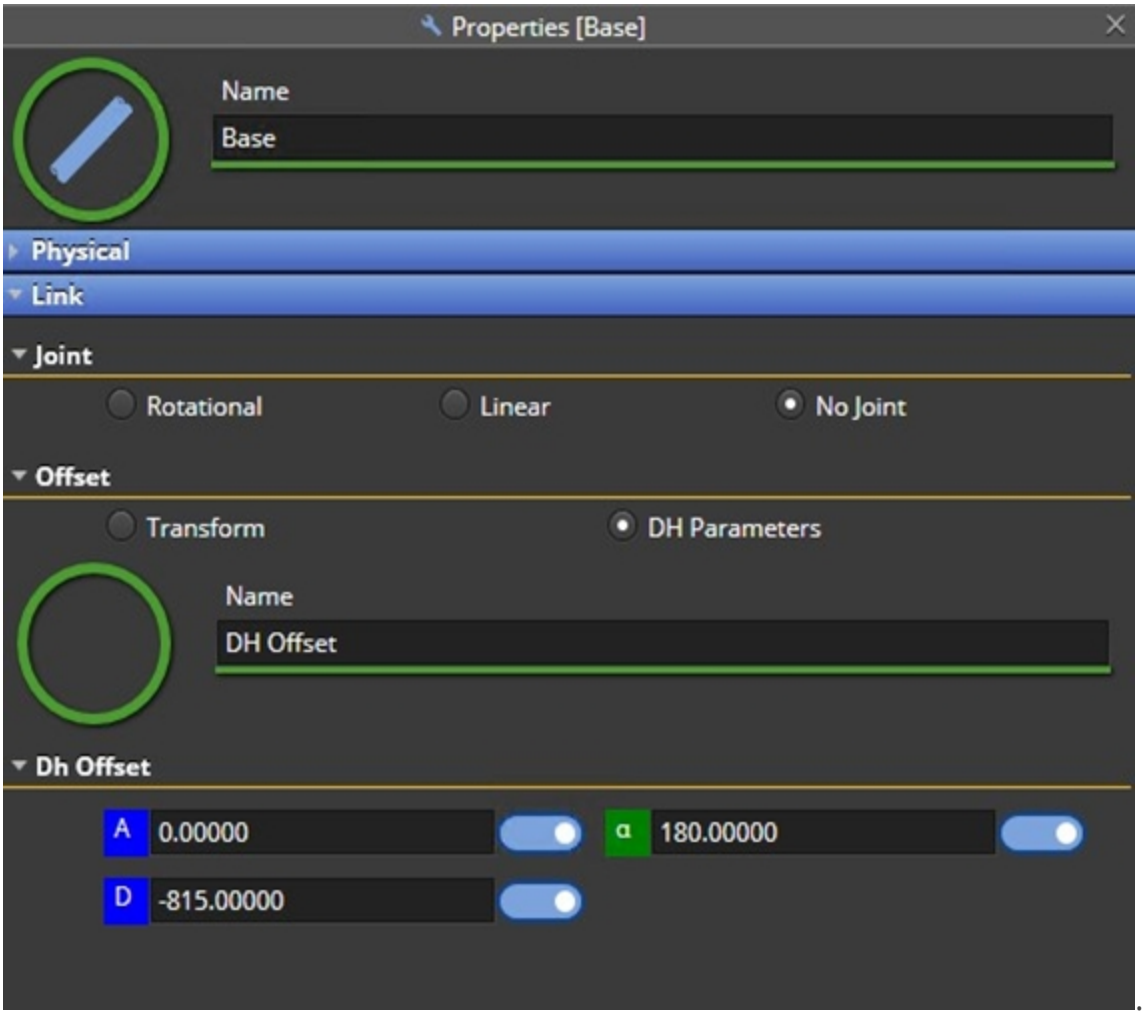
The following example will show how to model a Kuka robot manually.

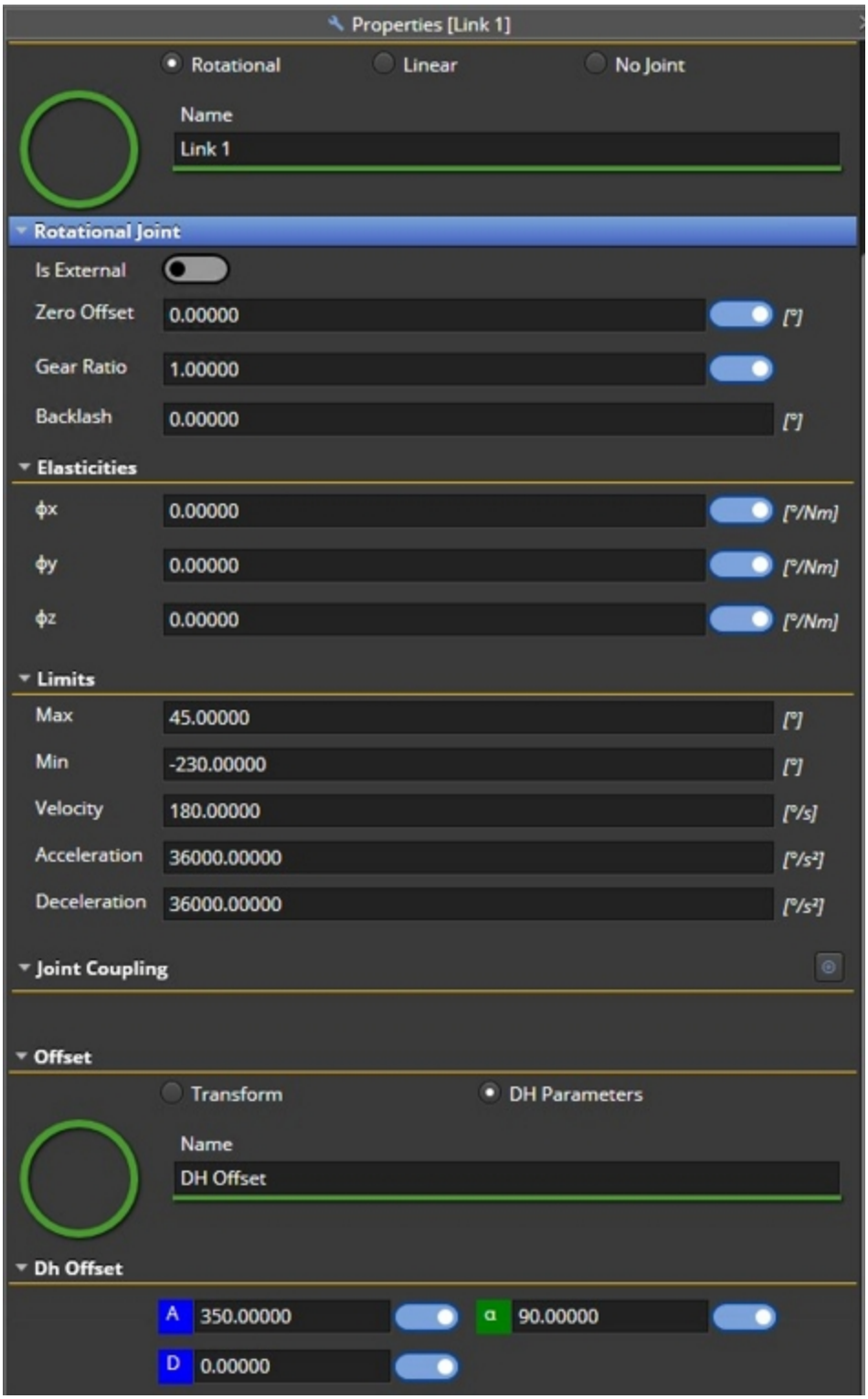
Right click on **Machines** and select New | Robots | Default 6 Axes. A robot will be created with cylinders used as links.



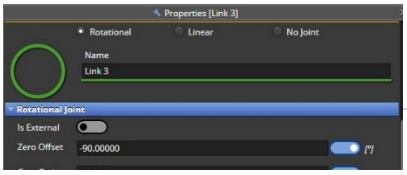



The properties of each link can be changed as follows by defining the DH parameters.

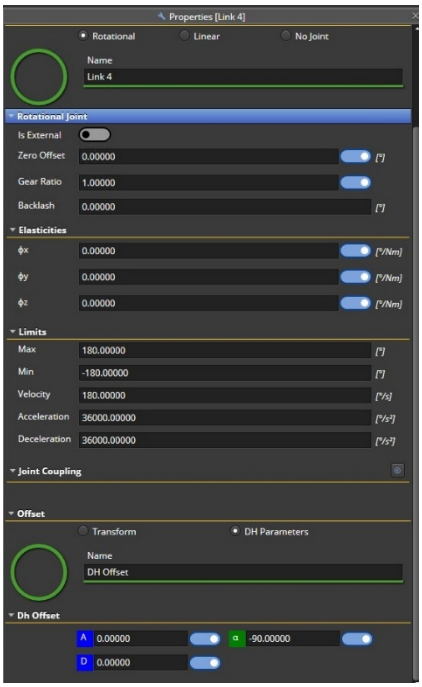


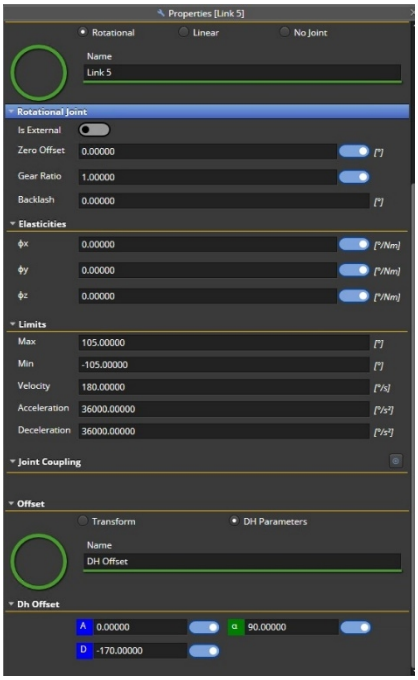

Properties [Link 1]
Rotational
Linear
No Joint
Name
Link 1
Rotational Joint
Is External
Zero Offset
0.00000
[°]
Gear Ratio
1.00000
Backlash
0.00000
[°]
Elasticities
ϕx
0.00000
[°/Nm]
ϕy
0.00000
[°/Nm]
ϕz
0.00000
[°/Nm]
Limits
Max
45.00000
[°]
Min
-230.00000
[°]
Velocity
180.00000
[°/s]
Acceleration
36000.00000
[°/s²]
Deceleration
36000.00000
[°/s²]
Joint Coupling
Offset
Transform
DH Parameters
Name
DH Offset
Dh Offset
A
350.00000
α
90.00000
D
0.00000

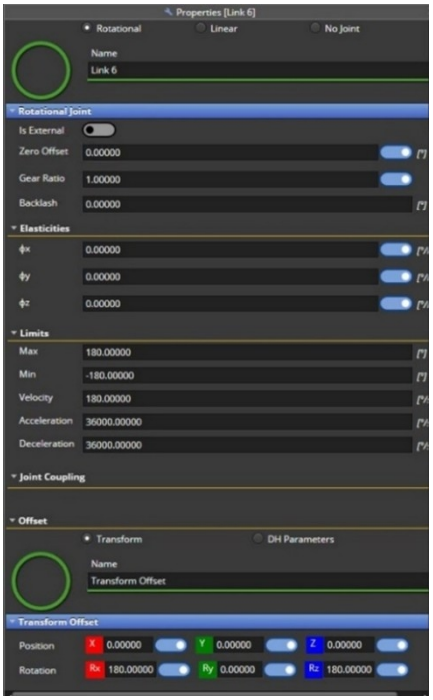
Properties [Link 2]
Rotational
Linear
No Joint
Name
Link 2
Rotational Joint
Is External
Zero Offset
0.00000
[°]
Gear Ratio
1.00000
Backlash
0.00000
[°]
Elasticities
ϕx
0.00000
[°/Nm]
ϕy
0.00000
[°/Nm]
ϕz
0.00000
[°/Nm]
Limits
Max
35.00000
[°]
Min
-135.00000
[°]
Velocity
180.00000
[°/s]
Acceleration
36000.00000
[°/s²]
Deceleration
36000.00000
[°/s²]
Joint Coupling
Offset
Transform
DH Parameters
Name
DH Offset
Dh Offset
A
850.00000
α
0.00000
D
0.00000


Properties [Link 3]
Rotational
Linear
No Joint
Name
Link 3
Rotational Joint
Is External
Zero Offset
-90.00000
[°]

ϕz
0.00000
[°/Nm]
Limits
Max
158.00000
[°]
Min
-30.00000
[°]
Velocity
180.00000
[°/s]
Acceleration
36000.00000
[°/s²]

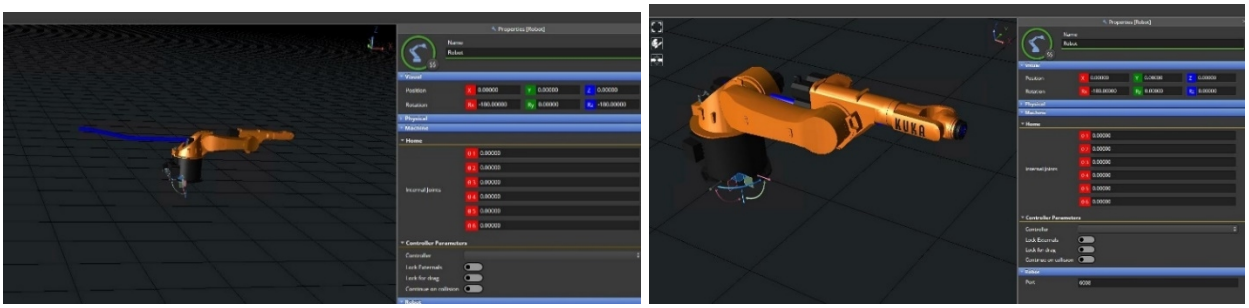


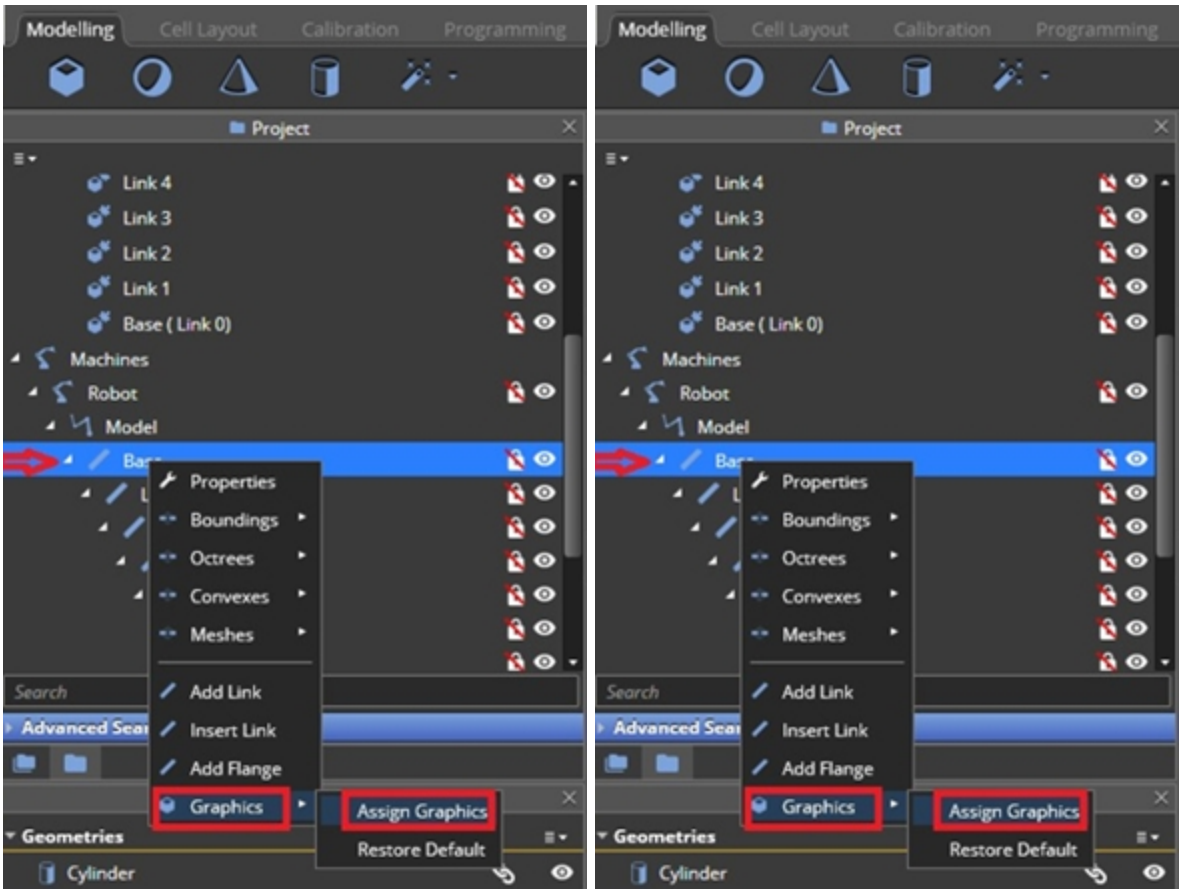




After finishing to edit the DH parameters, it is possible to assign the CAD files to the robot links.

First it is needed to unhide the Robot CAD under Cell Items and change robot cylinders representation to match the CAD files by changing RZ value from -180° to 0°.



To assign a graphics, right-click on the robot part and select Graphics | Assign Graphics and pick the related CAD item from the list.



### Robot creation wizard

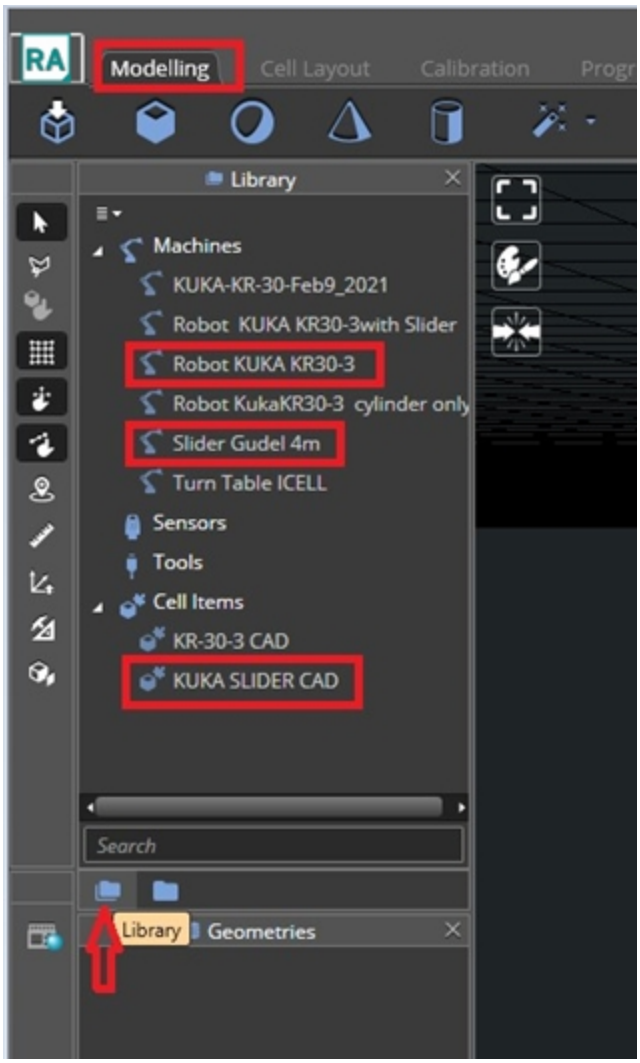
After clicking on the **Machine Creation** wizard, on the **Properties** window:

1. Select **Robot** as machine type, select Create as machine dimensions property and click **Add Joint** and select **No Joint** from the drop-down menu.
2. Add another 6 joints (Rotational) and click **Next**, leaving the internal joints parameter as default.
3. Continue as explained in Slider creation wizard following the next steps of the wizard defining the minimum and maximum values of the joint limits and assigning the graphics (or skipping this part).

At the end of the process under **Machines** in the project tree you will find the new machine. If the graphics step was skipped, no graphics will be displayed.

### Robot on slider

Make sure that you have the Machine model of the desired Slider and Robot in your library. Sometimes, the orientation of Robot and Slider CADs do not match, therefore you also need the Slider CAD in your library, separated already to Moving and Fixed groups as explained in chapter Slider.

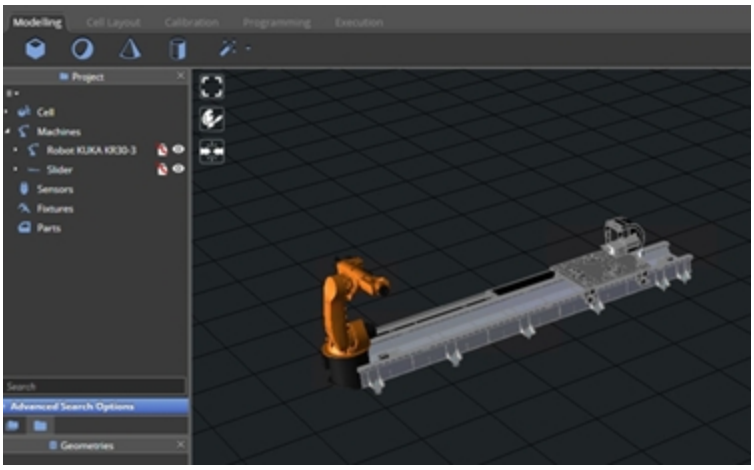


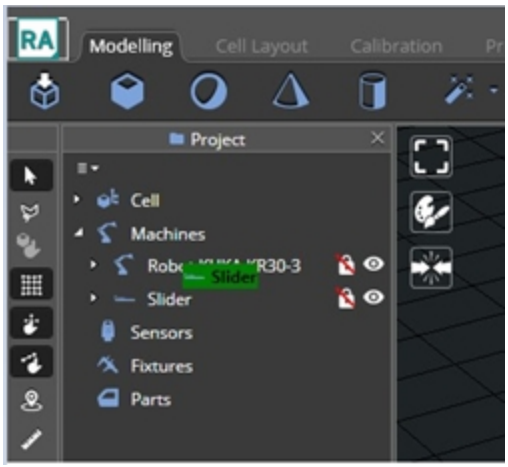

To combine the Robot and Slider into one Machine, drag and drop the desired robot from the library into the project. Select the robot, left click with mouse and drag it to the right. Repeat the same for the Slider (e.g., **Slider Gudel 4m** depicted above).

Within the project tree, under **Machines** you will have the robot and the slider.

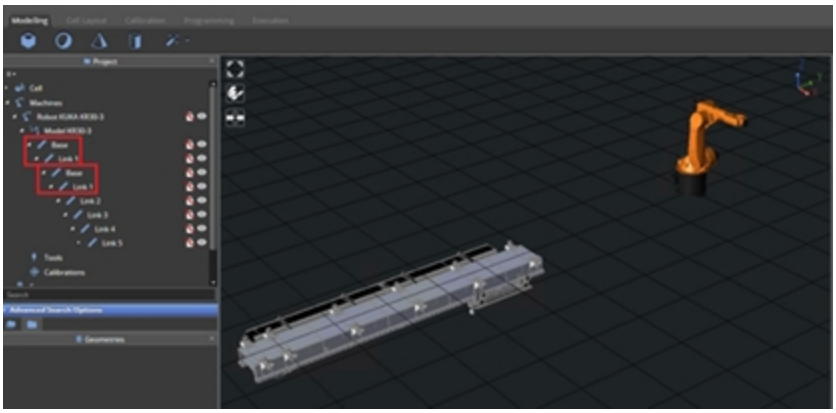
Select the slider with the mouse and drag it under the Robot. Once it is under the Robot, it will turn into green allowing you to drop it there.

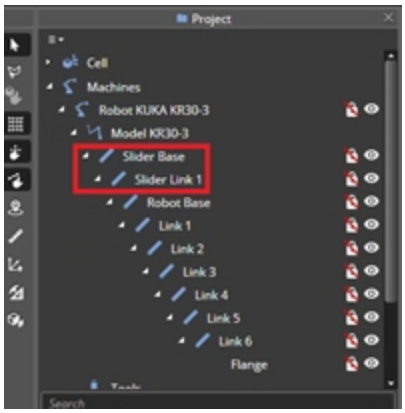
---




It is highly likely that the Robot model and Slider will not be in same coordinate system: you will notice there are two **Base** and two **Link1** items in the project tree as depicted in the picture below.





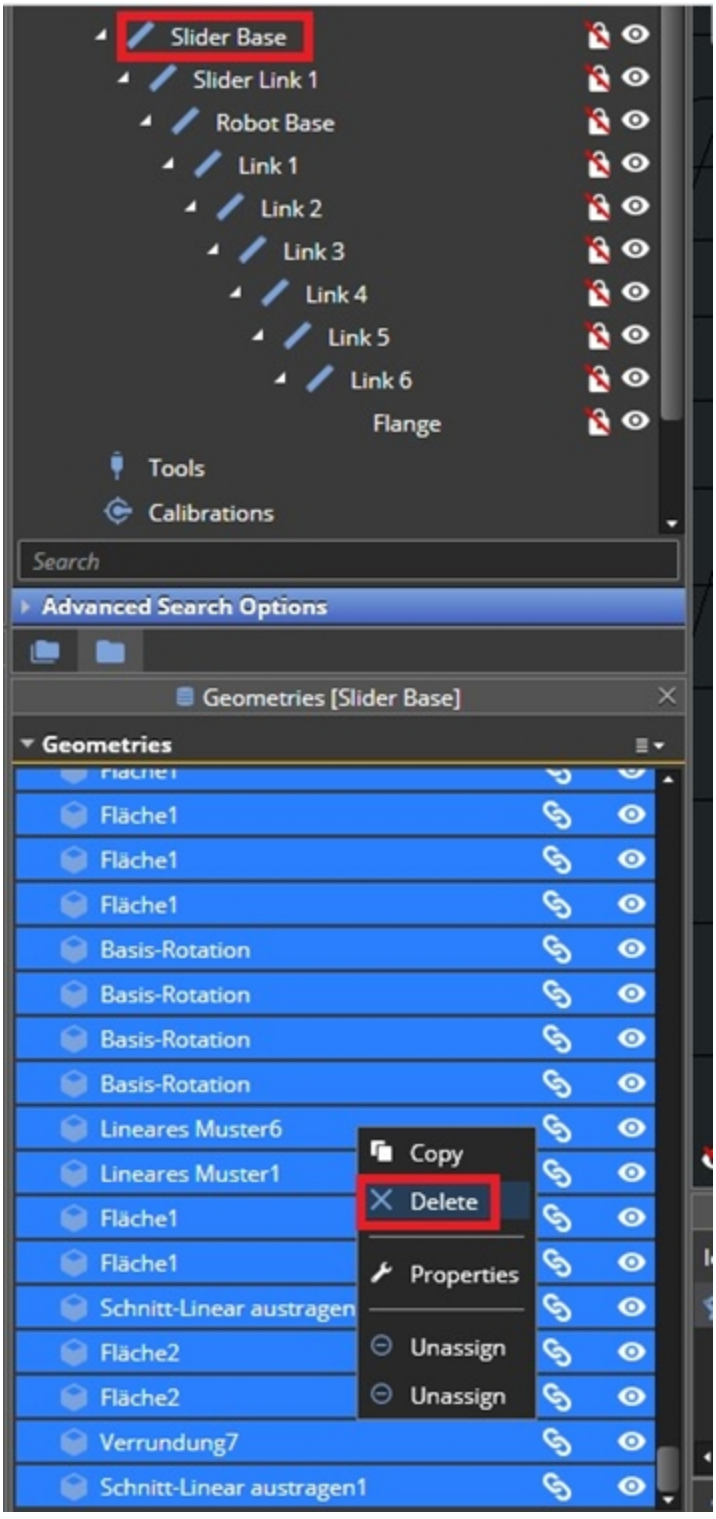

## Align Robot and Slider orientation and Position.

Select the robot properties and set the robot **Home** values for the slider (**E1**) and robot Joints (**θ1**-**θ6**) as they are in the real cell.

The next step is to correct the Robot Orientation in its home (we need to Rotate it 90° counterclockwise) so in robot Base properties it is needed to change the **Offset** parameter from **DH parameters** to **Transform**: The RZ must be changed to -90°.

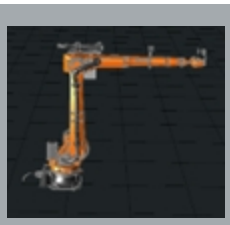
Make sure also to change the robot **Base** CAD accordingly.

The next step is to remove the slider assigned graphic CADs which are in wrong orientation and the same is to be repeated also for slider **Link 1**.


Slider Base
Slider Link 1
Robot Base
Link 1
Link 2
Link 3
Link 4
Link 5
Link 6
Flange
Tools
Calibrations
Search
Advanced Search Options
Geometries [Slider Base]
Geometries
Fläche1
Fläche1
Fläche1
Basis-Rotation
Basis-Rotation
Basis-Rotation
Basis-Rotation
Lineares Muster6
Lineares Muster1
Fläche1
Fläche1
Schnitt-Linear austragen
Fläche2
Fläche2
Verrundung7
Schnitt-Linear austragen1
Copy
Delete
Properties
Unassign
Unassign

The next step is to align the slider CAD group in its properties under the robot **Base**. Before you do that, make sure you have the correct slider E1 values as its position. In our example it is in 4000 mm.

### List of supported robot models

| Fanuc robot model using R-30ia controller | KUKA robot model using KR C4 controller |
|---|---|
| LR MATE 200iD-7C | KR 6 2 |
| CRX-10iA/L |  KR 20 R3100 |
| M-20iD/12L | KR 22 R1610 |

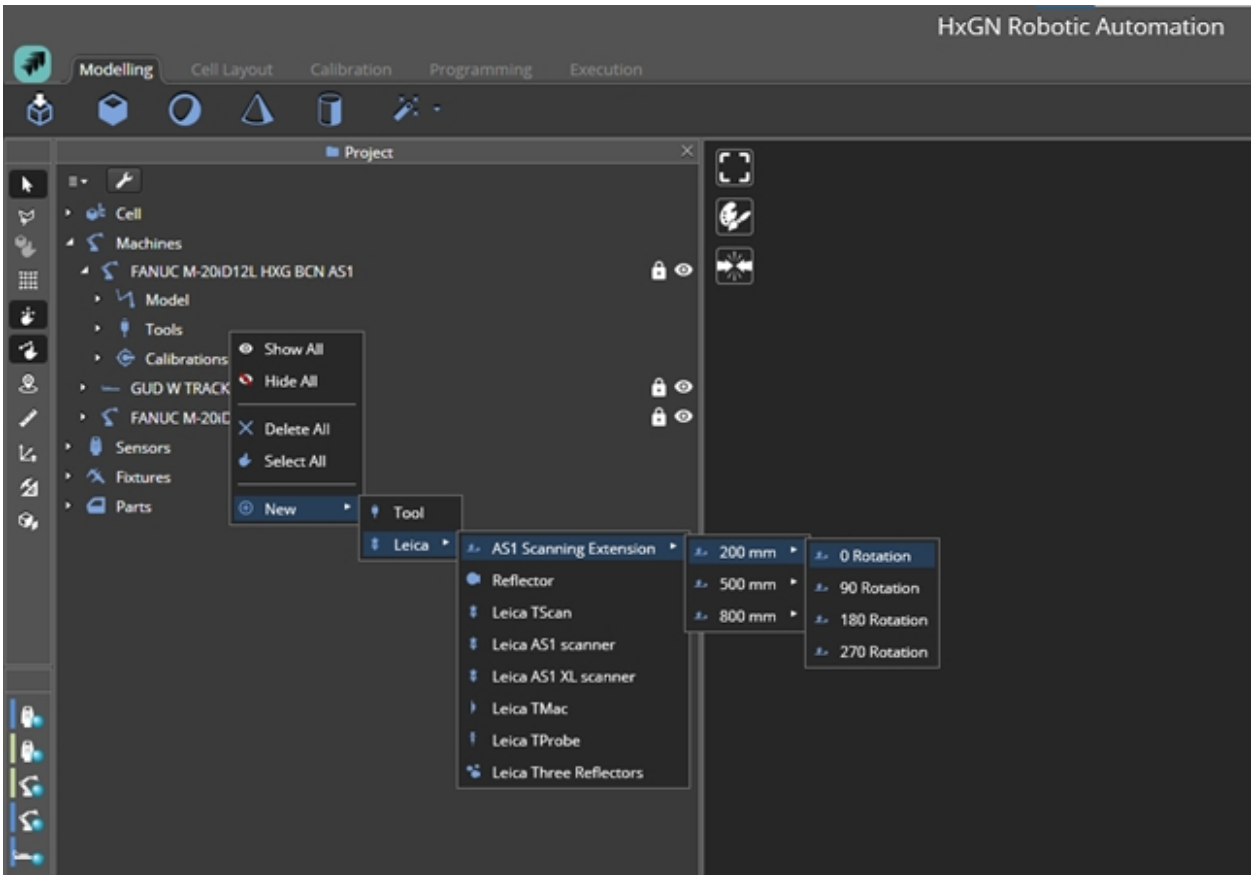
| Fanuc robot model using R-30ia controller | KUKA robot model using KR C4 controller |
| --- | --- |
| M-20iD/25 | |
| M-710iC/12L M-710iC/20L M-710iC/45 | KR 30-3 |

| Fanuc robot model using R-30ia controller | KUKA robot model using KR C4 controller |
| --- | --- |
| <br>M-710iC/50 | |
| <br>R-2000iC/125L<br> <br>R-2000iC/165F<br> <br>R-2000iC/210L | <br>KR 90 R2700 Pro |

| Fanuc robot model using R-30ia controller | KUKA robot model using KR C4 controller |
| --- | --- |
| | KR 210 R2700-2 |

### Robot tools

Once robot is modelled, a new tool can be added to the robot under the robot tree. The tool will be automatically attached to the robot joint 6 where robot TCP is defined. This tool could be generic or can be chosen from the range of tools offered by Leica. For example, AS1 scanner. See options in the image below.




A last step to align the tool with the robot may be needed if adapters are used to place the tool at a different angle or orientation relative to the robot.

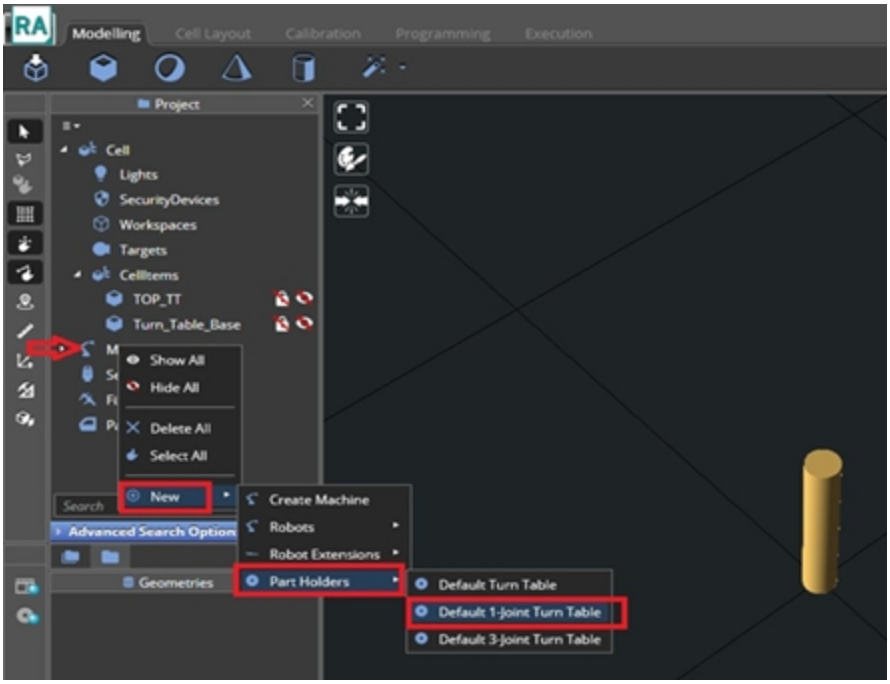
## Rotary Table

Following the same steps explained in chapter Slider it is possible to import the Rotary Table's CAD files. The CAD will show under **Cell Items** as a **Group**. The CAD elements can be separated and merged into two groups: **Turn_Table_Base** (fixed), and **Top_TT** (rotating).

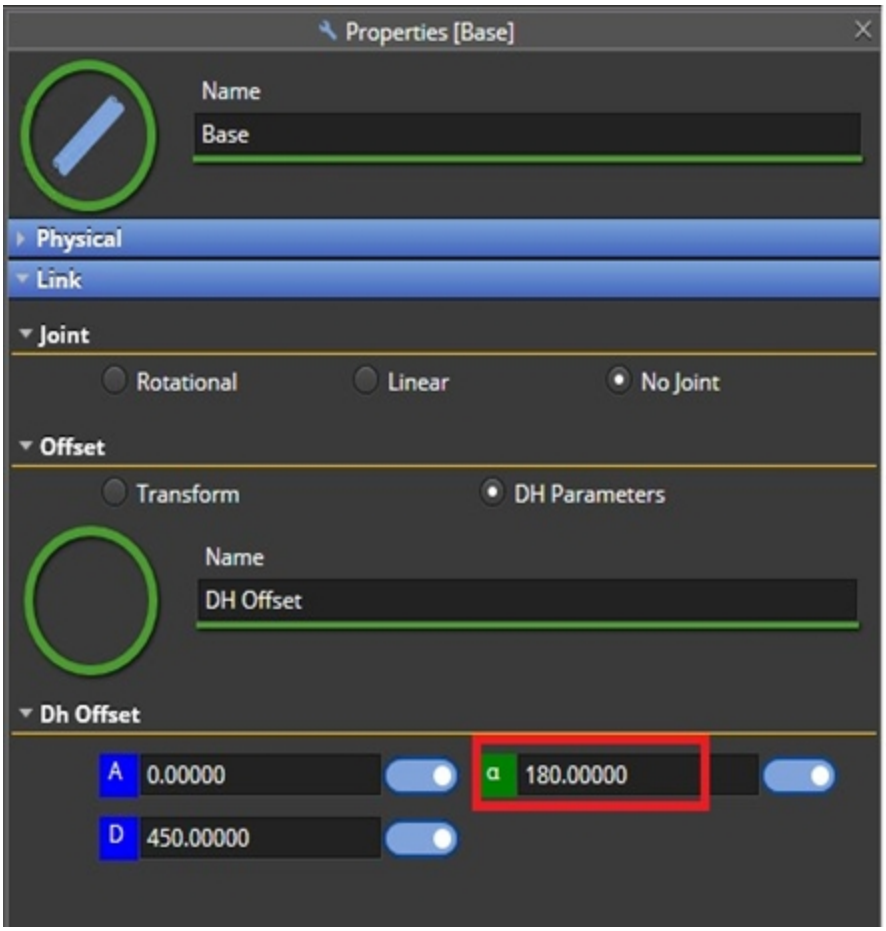
It may be needed to change the CAD properties to make sure Rotary Table Cad orientation facing up to Z axis.

A Rotary Table can be created manually of following the Rotary Table creation wizard.

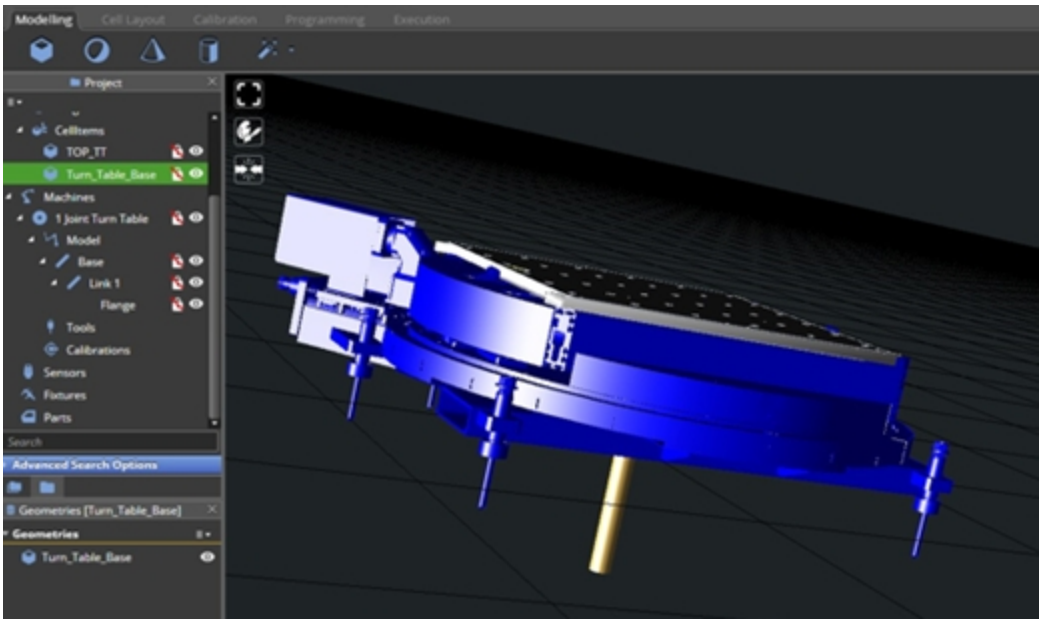
Under Cell items group, right click on **Machines**, then select *New* | *Part Holders* | *Default 1-Joint Rotary Table*. A cylinder will be created.



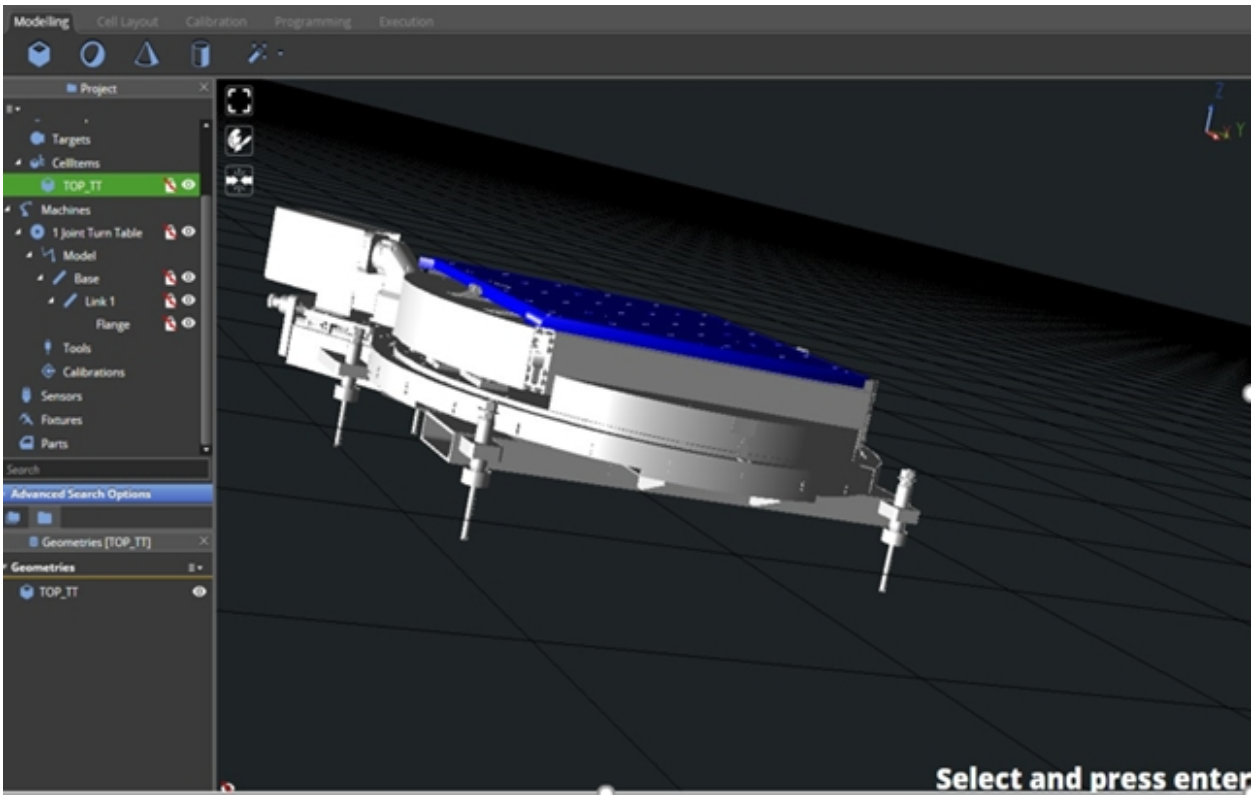

On the cylinder properties, depending on the rotation direction, it is possible to set the correct DH parameters, e.g., changing the α value to 180° will make the cylinder rotating in positive direction in clockwise direction.




Now it is possible to assign the graphics to **Base** and **Link 1**. By right-clicking on **Base** and selecting *Graphics* | *Assign Graphics* and selecting with the mouse the Rotary Table CAD, it will turn to green. After pressing the [ENTER] key, clicking on the Base again, you will notice that the **Geometries** will include the Rotary Table base CAD that will no longer be under Cell Items in the project tree.




The same can be done for **Link1** and **TOP_TT** CAD can be assigned.




---

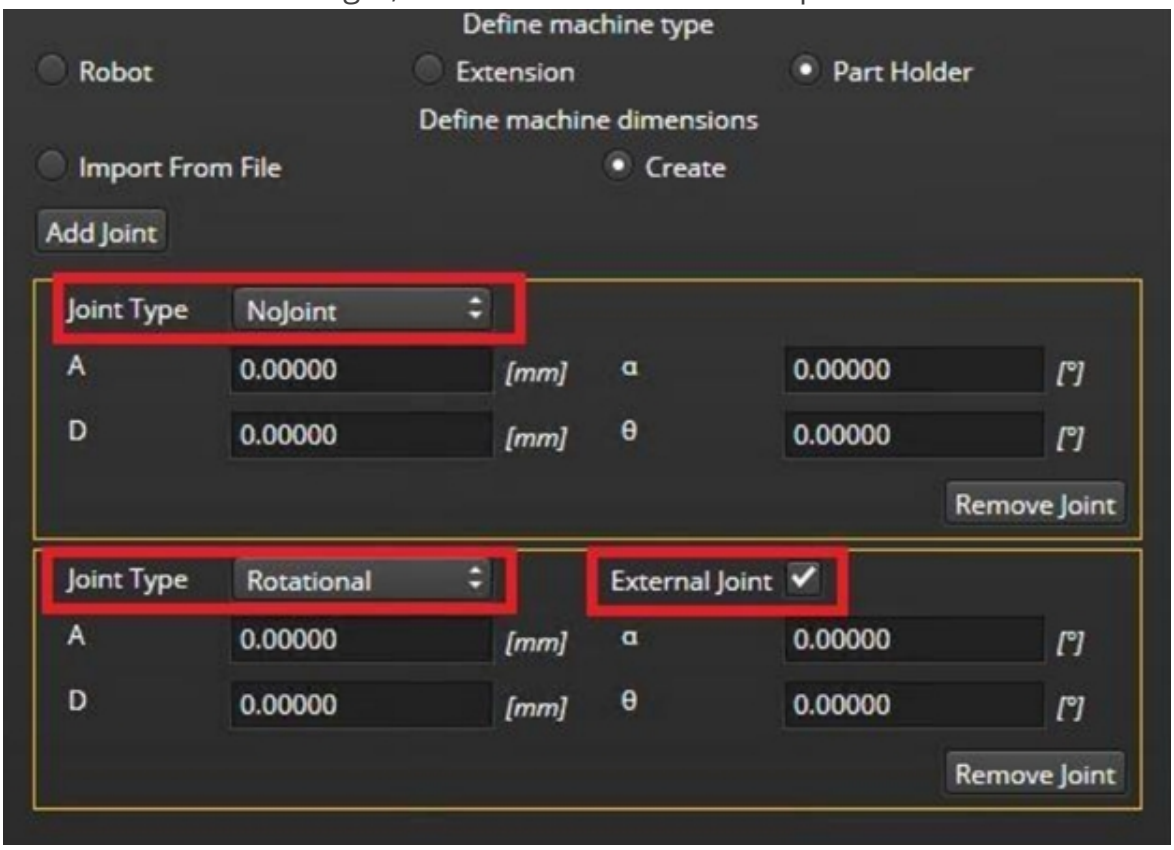
### Rotary table creation wizard

The Rotary Table creation wizard reflects the same steps explained in chapter Slider, but in this case the second joint to be added will be a Rotational joint.

After clicking on the **Machine Creation** wizard, on the **Properties** window:

1. Select **Part Holder** as machine type, select Create as machine dimensions, and click **Add Joint** and select **No Joint** from the **Joint Type** drop-down menu.

   Add another joint selecting **Rotational** for the moving joint and checking **External Joint**, leaving the D-H parameters unchanged. Once done, click on the arrow on bottom right, to continue to the next step.

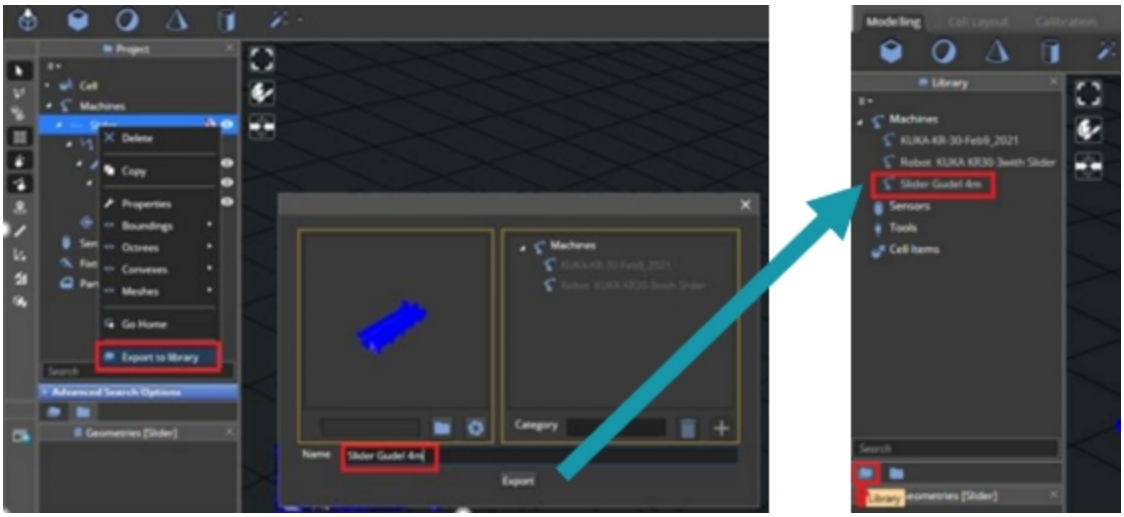
2. 

3. Continue as explained in Slider following the next steps of the wizard defining the minimum and maximum values of the joint limits and assigning the graphics.

At the end of the process under **Machines** in the project tree you will find the new machine.

## Exporting an object to the library

Select the object to be exported, right-click and select **Export to library**. A dialog window will appear where it is possible to specify the object **Category** and **Name**. After clicking Export, the object will be added to the **Library** for future use.



## Changing the machine properties

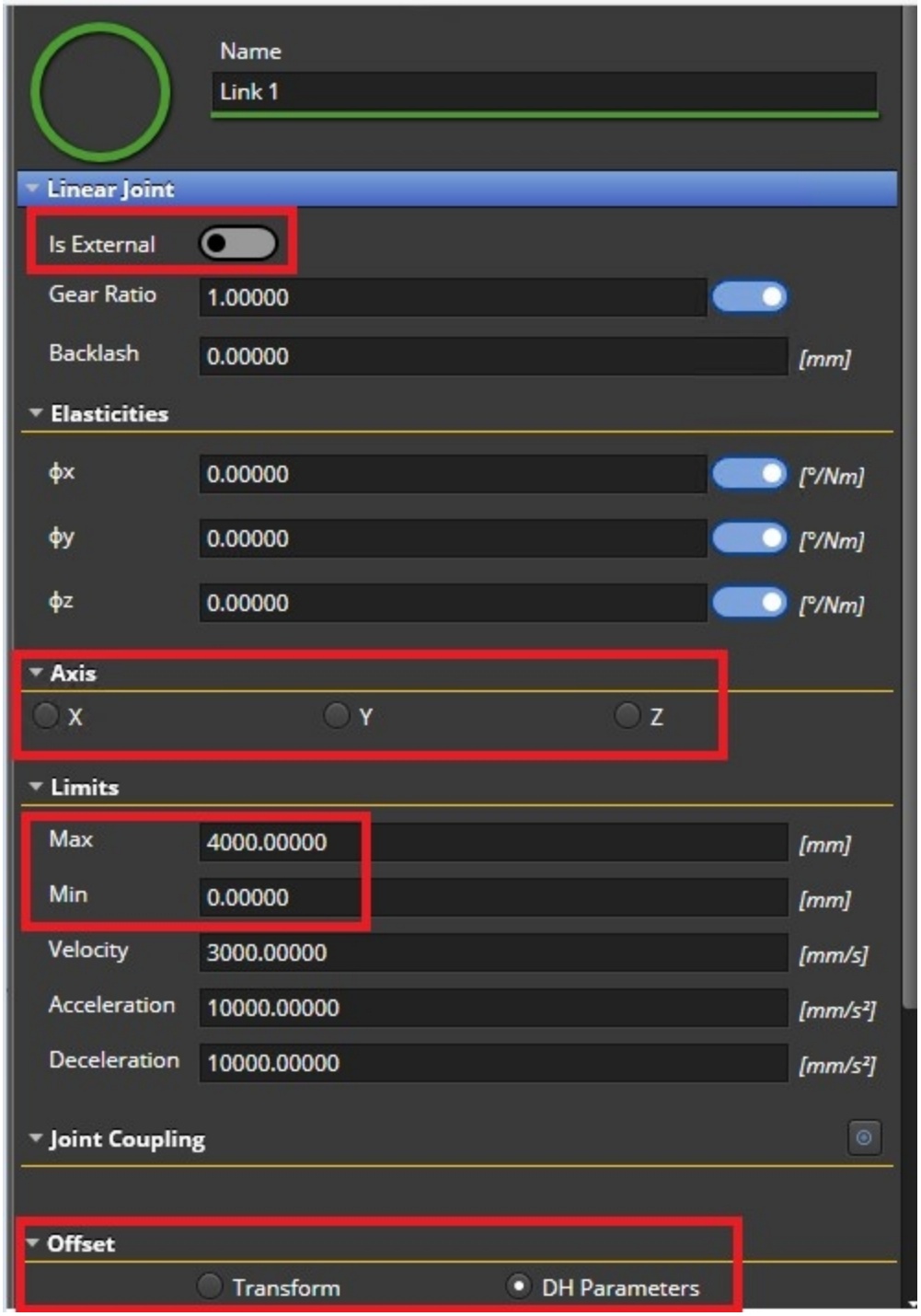
In the project go to cell item, right click, and check the properties.

Depending on the machine type, different parameters can be changed.

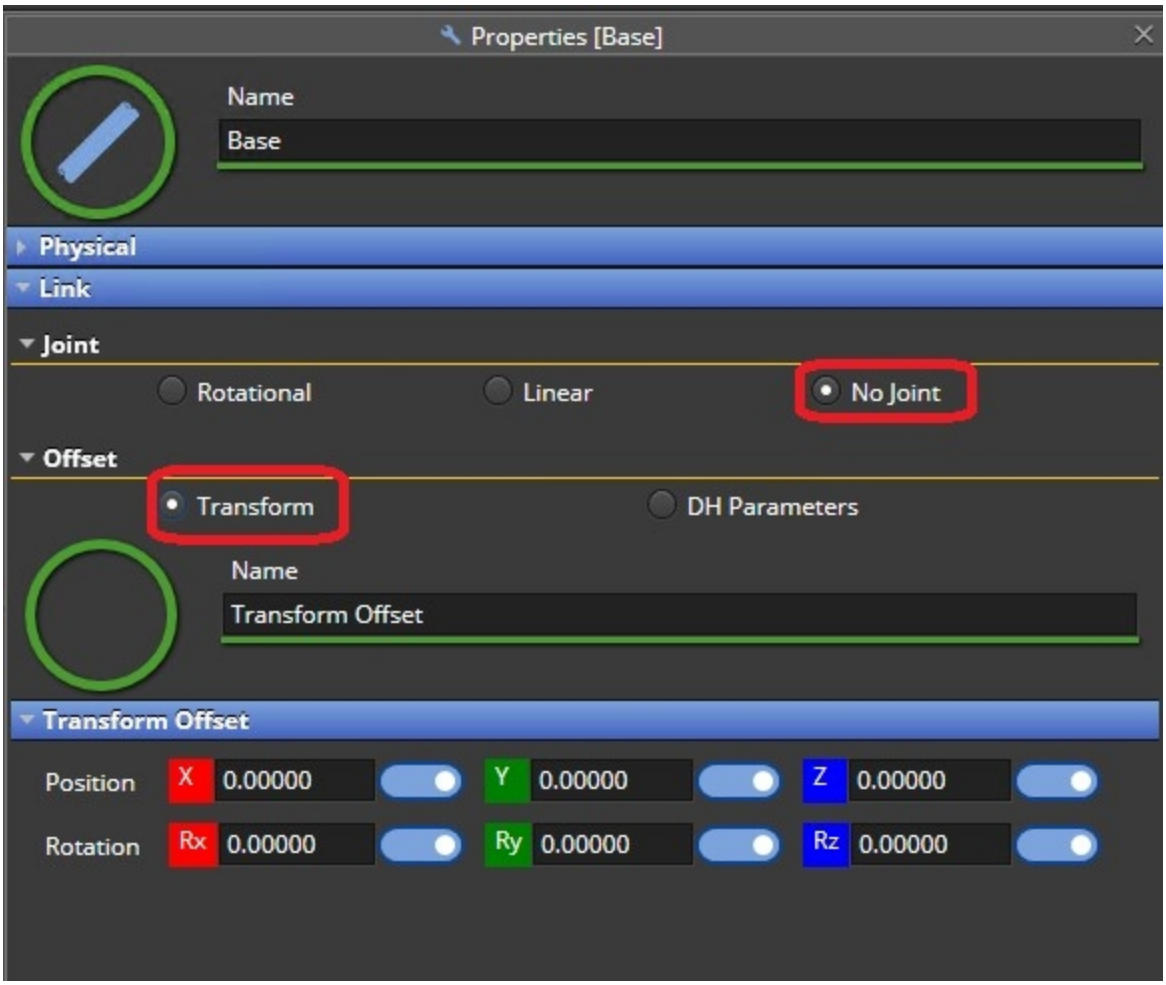
E.g., considering the slider modelled in chapter Slider the default values must be changed to:

1. For the **Link 1**:
    a. **Linear Joint** should change to **Is External**
    b. **Axis** to be check to Y (in our example)

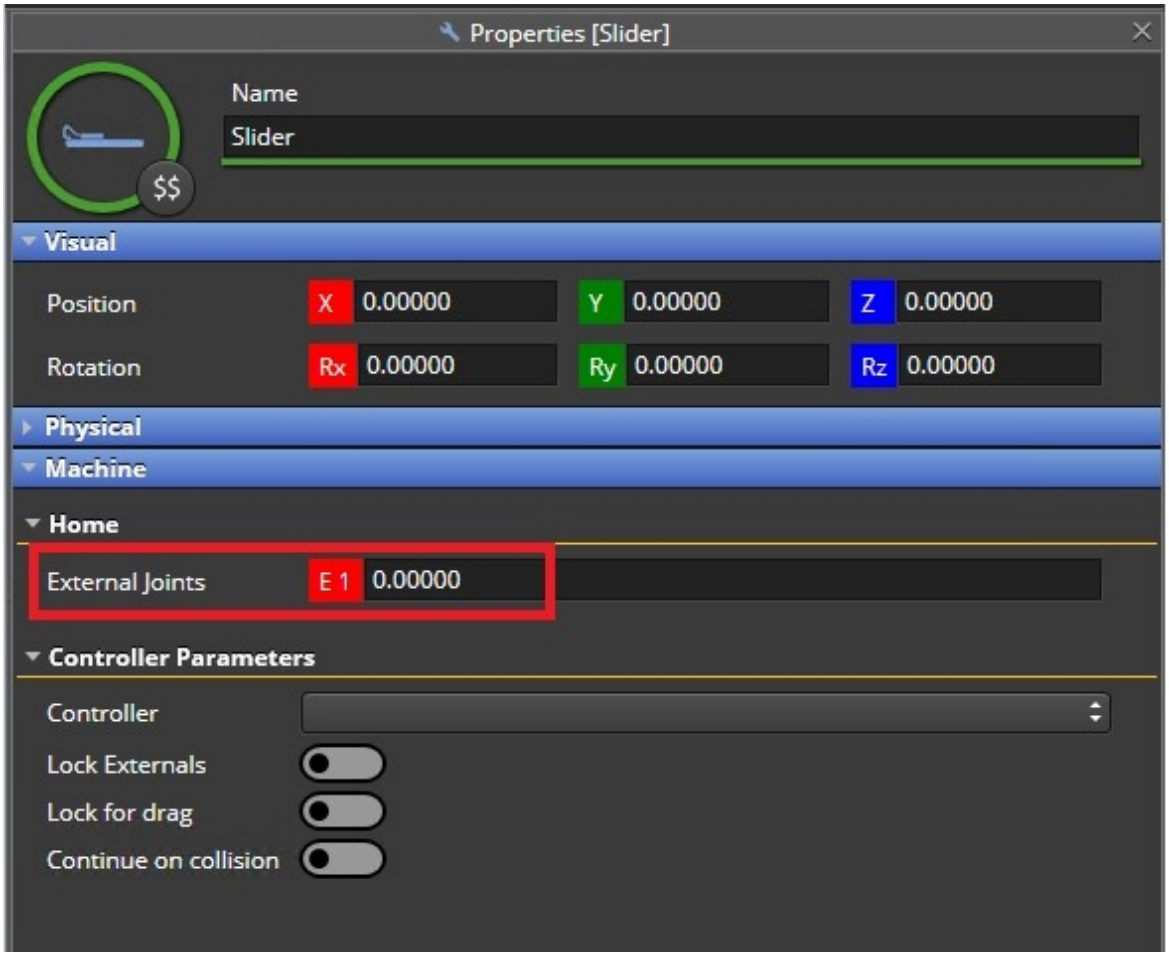
c. **Offset** change to Transform Limits


Name
Link 1
Linear Joint
Is External
Gear Ratio
1.00000
Backlash
0.00000
[mm]
Elasticities
ϕx
0.00000
[°/Nm]
ϕy
0.00000
[°/Nm]
ϕz
0.00000
[°/Nm]
Axis
X
Y
Z
Limits
Max
4000.00000
[mm]
Min
0.00000
[mm]
Velocity
3000.00000
[mm/s]
Acceleration
10000.00000
[mm/s²]
Deceleration
10000.00000
[mm/s²]
Joint Coupling
Offset
Transform
DH Parameters

2. For the **Base** verify that Slider Base is set to **No Joint** and **Offset** to **Transform**.

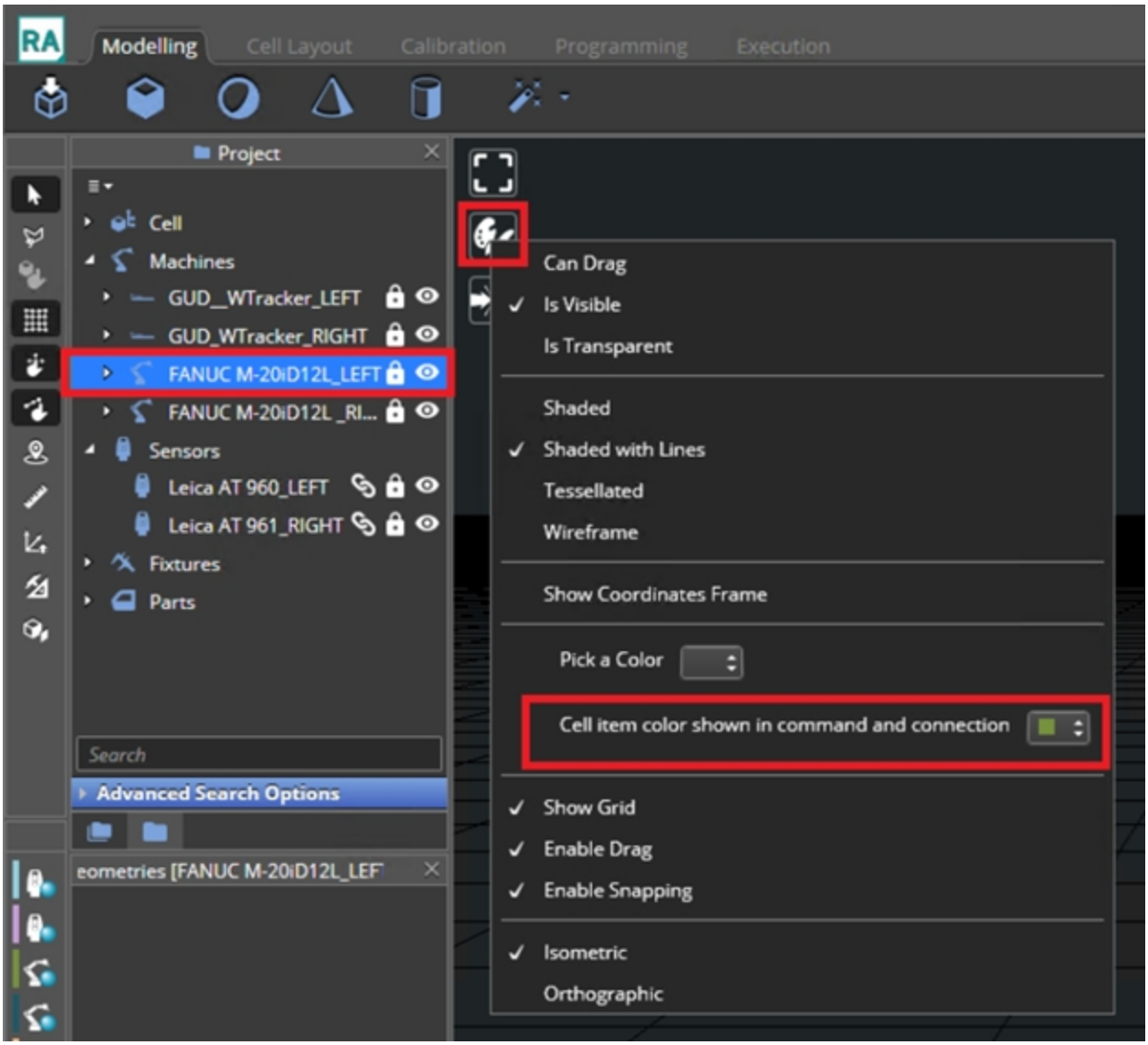


Set **Home** position.

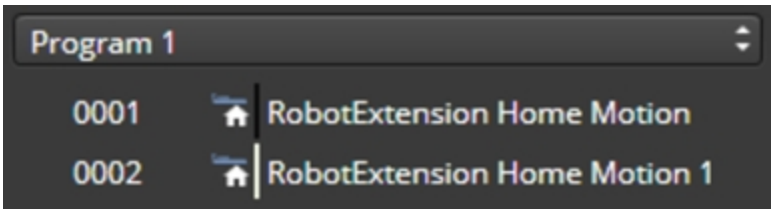
3. 


4. Assign a color to each cell device within the **Visual** field by picking a color from the available ones or by specifying the RGB or Hex code.

It’s also possible to select the cell item in the 3D Window, first right-clicking on the item and then selecting the color from the drop-down menu by clicking on Cell item color shown in command and connection.




By assigning a specific color to all the cell items, it will be easier for the user to easily distinguish between which cell items take part in which command of a program. E.g., in the picture below, two different sliders can be easily distinguished because one is depicted in black and the other one in white.




## Configuring machine saved positions

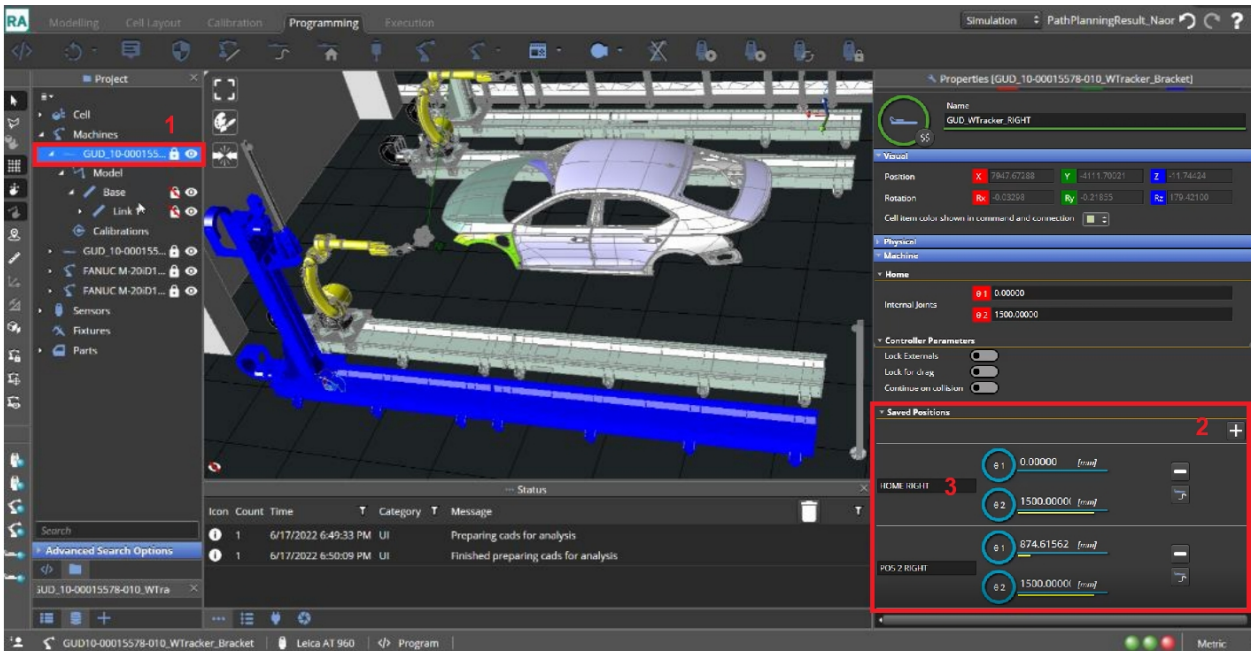
It is also possible to save some relevant positions to be used by the automatic path planning algorithm (see chapter Slider positions for more details) to move the machine to one of the saved positions to perform a proper scan, e.g., measuring the back of a door mounted on a fixture could require rotating the Rotary Table by 180°.

This would reduce the effort of the path planning algorithm because the sampling would be done among the saved positions instead of the infinite positions available within the complete joint space of the machine, e.g., 360° for a Rotary Table or 4m for a 4m-slider.

These positions are stored under Saved Positions property.

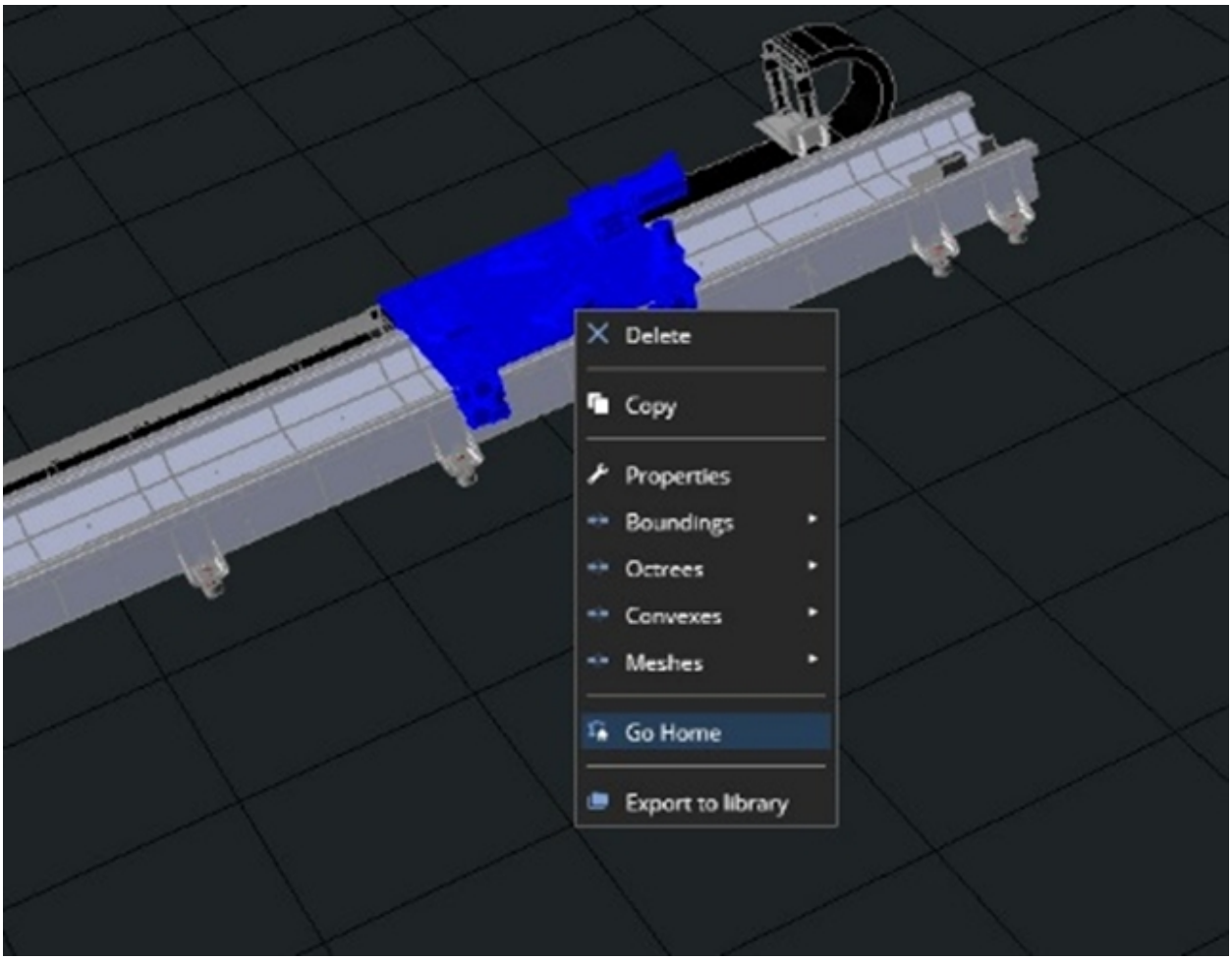
To save a position:

1. Move the machine in the desired position.
2. Click the plus button ( ) to store it. The minus button ( ) will remove the corresponding position.
3. Rename the position conveniently.




## Machine testing

Please refer to chapter Manual Programming Mode.htm for more information on how to move the machines within the 3D environment to test the correct movements of the machines (slider linear movement, Rotary Table rotational movement, robot cartesian or joints movements. To send the machine to home position it's just needed to right-click on the machine in the 3D View and click Go Home.

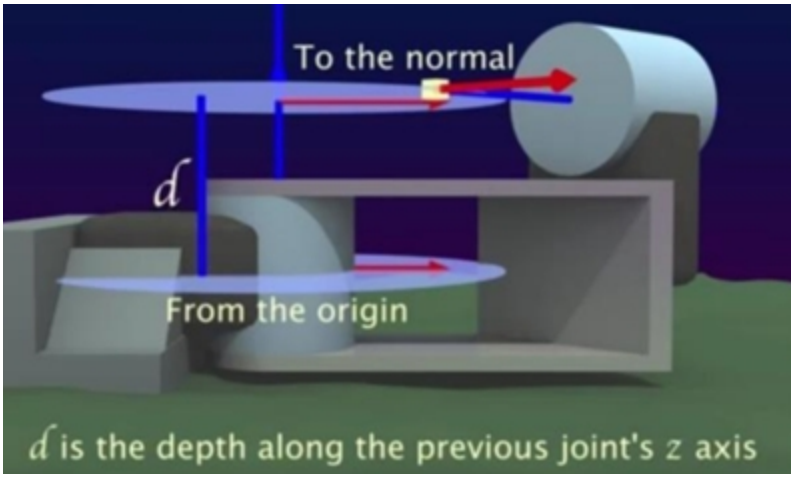



## D-H Parameters

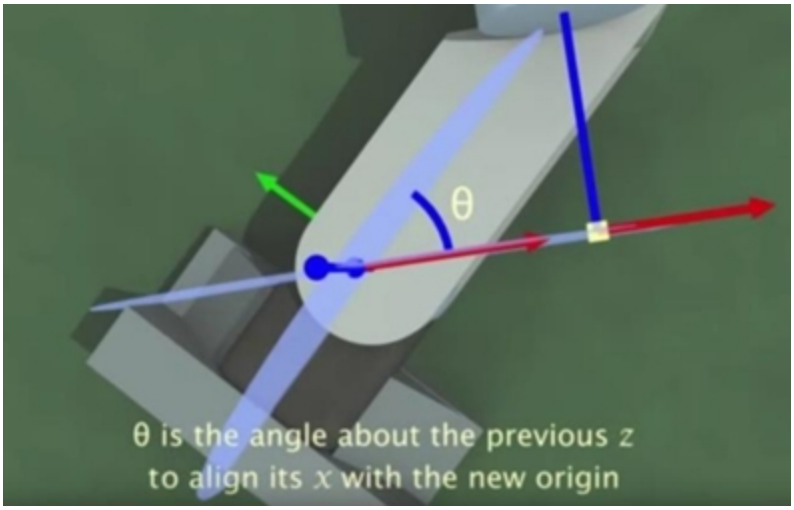
D-H (Denavit-Hartenberg) parameters are the four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain or robot.

The D-H parameters are the following:

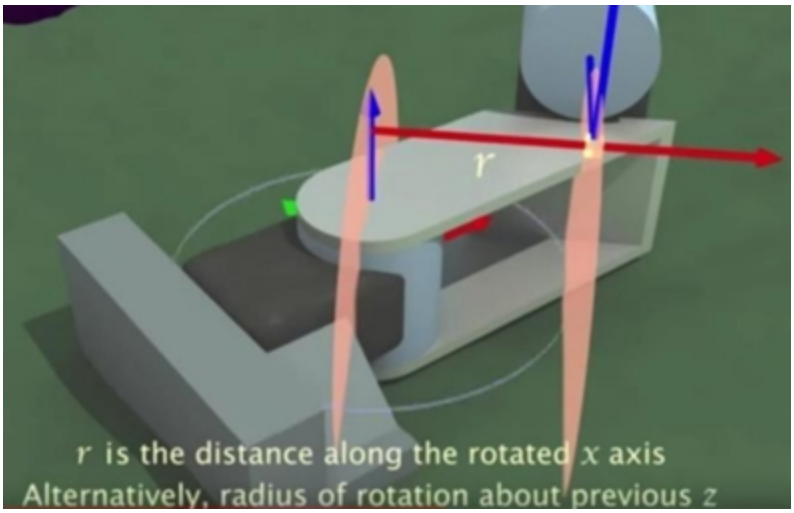
**D**: the depth along the earlier joint's Z axis

1. 

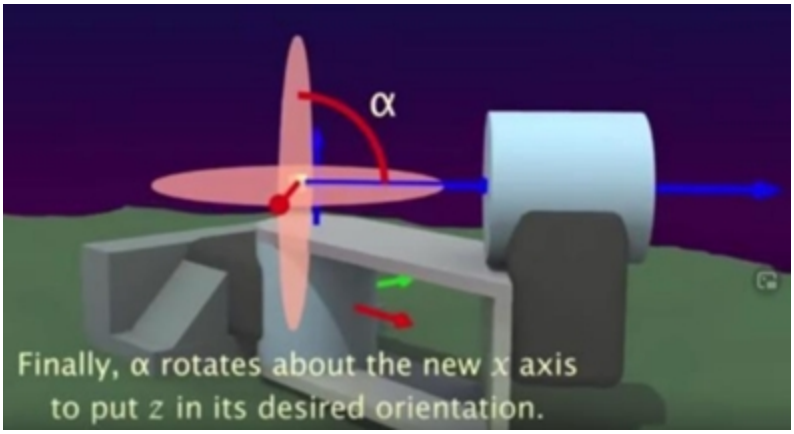

**θ**: the angle about the earlier Z axis to align its X with new origin. In HxGN Robotic Automation this is called **Zero offset**.

2. 

**R**: the distance along the rotated x axis. In HxGN Robotic Automation this is called **A**.

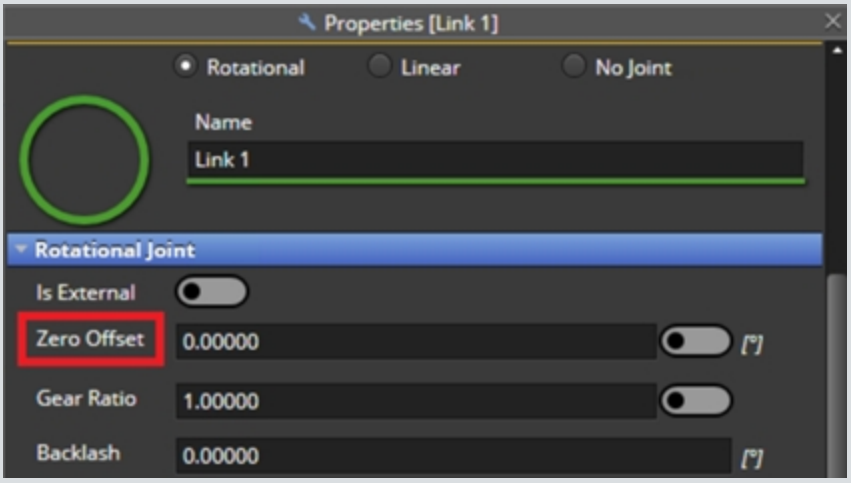
3. 


**α**: the rotation angle about the new x axis to put Z in its desired orientation.

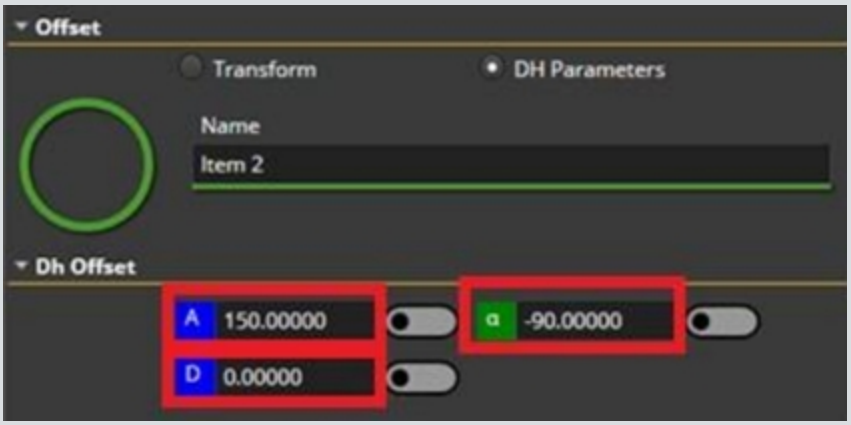
4. 


Example: Properties of Link 1 of Fanuc robot

θ is configured as **Zero Offset** under the **Rotational Joint properties**:




while **R** (A in HxGN Robotic Automation), **α** and **D** are configured under **Dh Offset properties**:




## Cell Modelling

Robotic Automation application has the capability to model a cell layout which includes the following:

- **Machines**: such as robots, robots on a slider (robot extension), Rotary Tables – part holders.
- **Tools**: predefined tools such Leica T-Scan or AS1 and reflectors.

- **Sensors**: predefined sensors such as Leica AT960.
- **Fixtures**.
- Any static object.

Prior to building the model you will need to prepare the following:

- CAD file (format such as STP, JT, etc.) for any object.
- Existing library should have machine model names created according to chapter Machine_Creation
- Tools and sensor model names used.
- Cell dimensions or drawings.

## Adding items to the cell

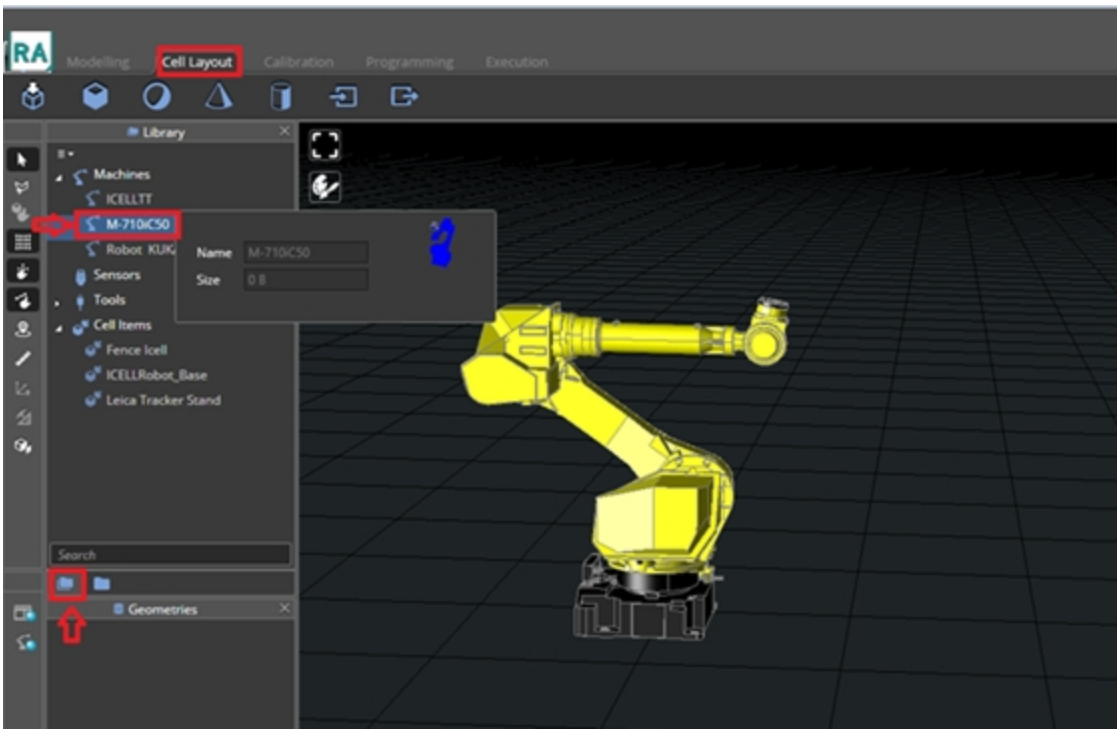
When starting Robotic Automation, and creating a New Project, an empty cell layout (named Untitled) is opened in the 3D Viewer. In the Cell Layout the user can add items to set up a Cell either by importing items from the library or by importing them as CAD models. The two modes are explained in the following sections.

Note that the collision engine will use a less optimized algorithm in case the cell has over 120 different items!

### Importing items from the library

Cell Items from the Library as: Robots, Rotary Tables, Fences, Cabinets, Robot, Controllers can be added to the cell from the library if they were previously created and exported.

On the Items Tree click on the **Library** Icon and expand the Machines Folder. E.g., if you want to use the FANUC M-710iC50, select it, drag it, and drop it in the 3D Window.



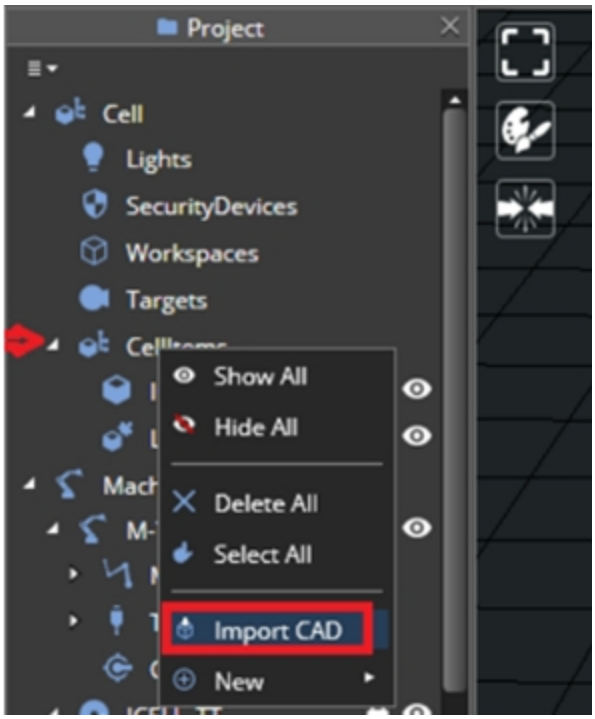

You can continue to add other items from the **Machines** Library, or static object from **Cell Items** library in the same way.

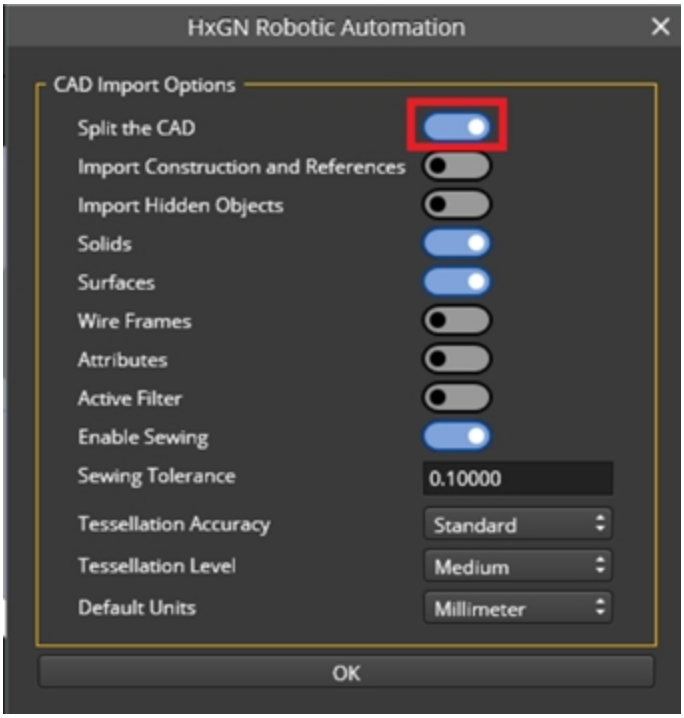
### Importing items as CAD models

User can add Items to a Cell by importing CAD Models and editing their properties, right-click on **Cell Items** and select **Import CAD** and browse the CAD file name needed.

The **CAD Import Options** dialog will pop up. The **Split the CAD** option is disabled by default and can be enabled in case it is needed.

After clicking **OK**, you will have the cell with its components positioned according to the CAD coordinate system.


Project
Cell
Lights
SecurityDevices
Workspaces
Targets
Show All
Hide All
Delete All
Select All
Import CAD
New


HxGN Robotic Automation
CAD Import Options
Split the CAD
Import Construction and References
Import Hidden Objects
Solids
Surfaces
Wire Frames
Attributes
Active Filter
Enable Sewing
Sewing Tolerance
0.10000
Tessellation Accuracy
Standard
Tessellation Level
Medium
Default Units
Millimeter
OK

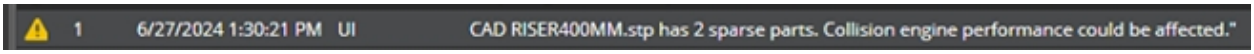
## Analysis of CAD sparse parts

It is possible to analyze how sparse the CAD it, that is, if the CAD is composed of separated parts. This can be detrimental to the performance of the collision engine.

To evaluate if a CAD has sparse parts activate it in file "*C:\ProgramData\Hexagon\RA\Config\RAconfig.xml*", Section *CADImportConfigurationSettings*, by setting *EnableSparseParts* to True. (The default setting is False.)

```
<ConfigSection Name="CADImportConfigurationSettings" DisplayName="CAD Import Configuration Settings" Description="RA
CAD import configuration settings">
    <ConfigSetting Name="EnableSparseParts" Value="True" DefaultValue="True" DisplayName="Enable Sparse Parts"
    Description="Enable/disable Sparse Parts." TypeName="System.Boolean" />
</ConfigSection>
```

It the setting is set to True, the first time each CAD is loaded on its project startup the CAD will be analyzed for sparse parts, and if any sparse parts are found the following warning message will be displayed per CAD:



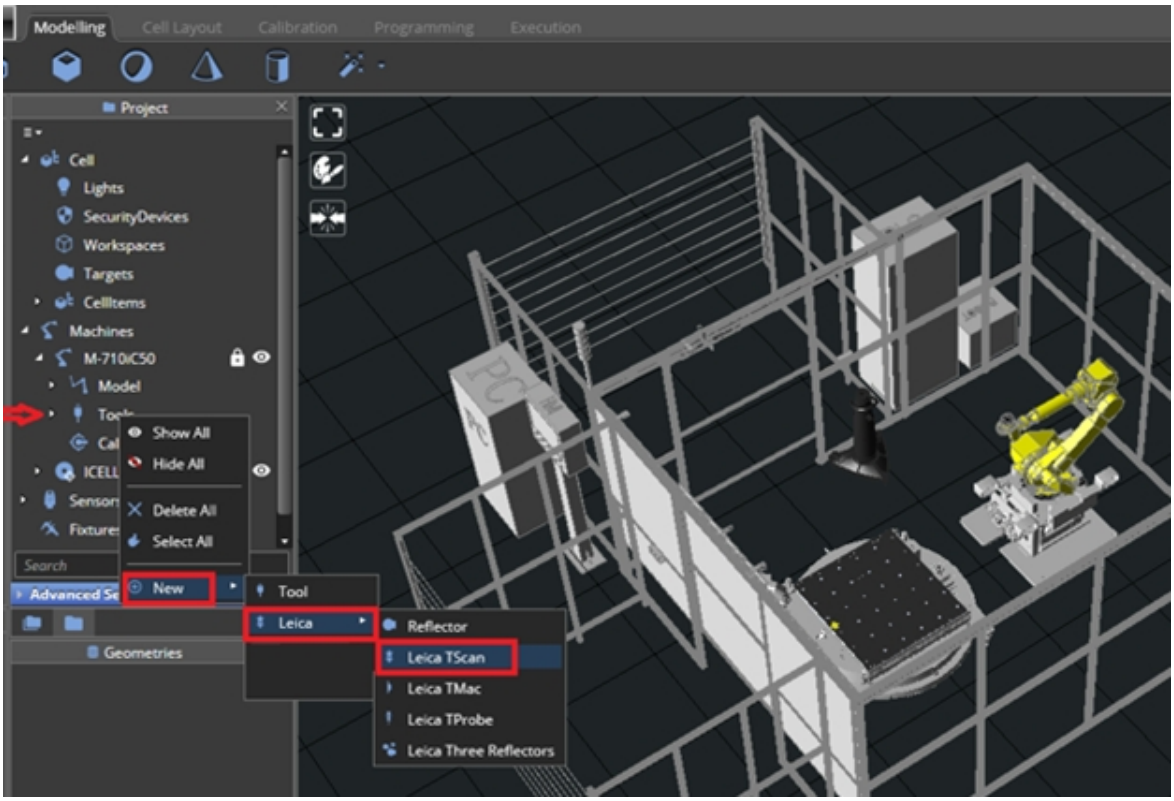

Note that the analysis of a very big CAD can take up to minutes.

## Loading and attaching a Tool to the Robot

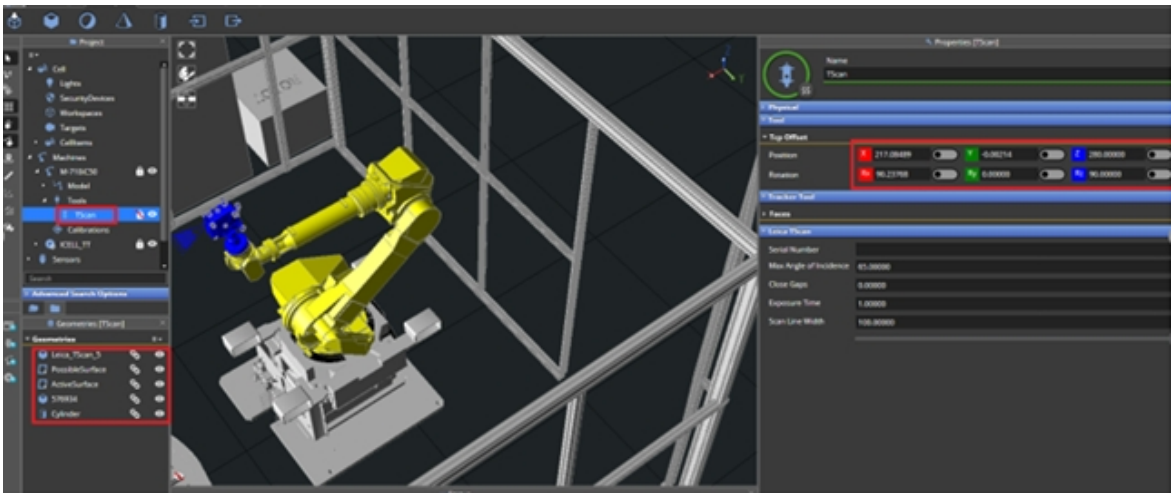
Once the Robot is added to the layout, a Tool to a Robot can be added as an existing Tool from the Library and Editing its position and activating it.

Once the project is loaded and Robot is displayed in the 3D Window:

1. Expand the **Machines** Tree.
2. Select the robot (e.g., M-710iC50) and expand it.
3. Right-click on the Tools section and select the tool, e.g., a Leica T-Scan 5 scanner (*New* | *Leica* | *Leica TScan*).



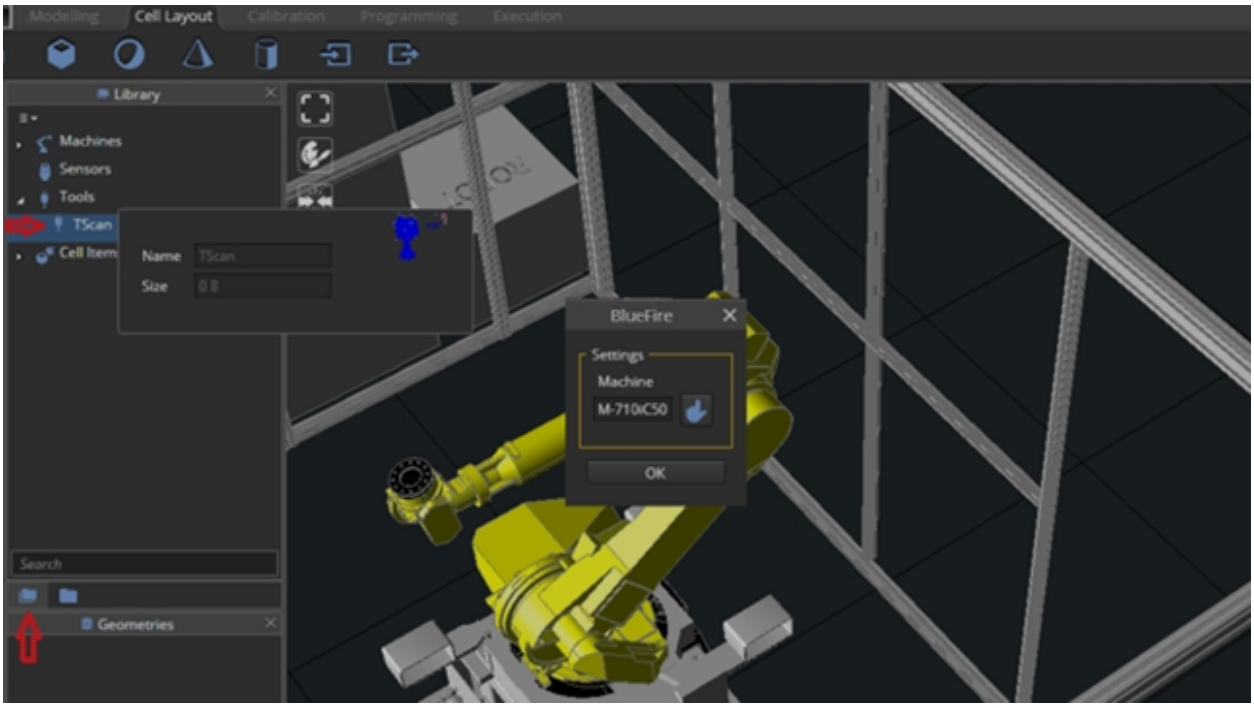

The TScan tool will be added to the robot in the 3D Viewer. You can use the align tool and position the TScan correctly on Robot J6 and add any Adaptor CADs as needed.



Alternatively, it is possible to add a tool from the library.

The same steps followed to add other items from the library can be followed, see chapter Adding_items_to_the_cell. The only difference is that the tool will be found under **Tools**.

After dragging and dropping it into the 3D Window, a pop-up window will appear to select the Robot to which the tool will be attached where it is possible to verify that the Robot model type assigned is correct or select the right one by clicking on the hand icon and pressing **OK**.

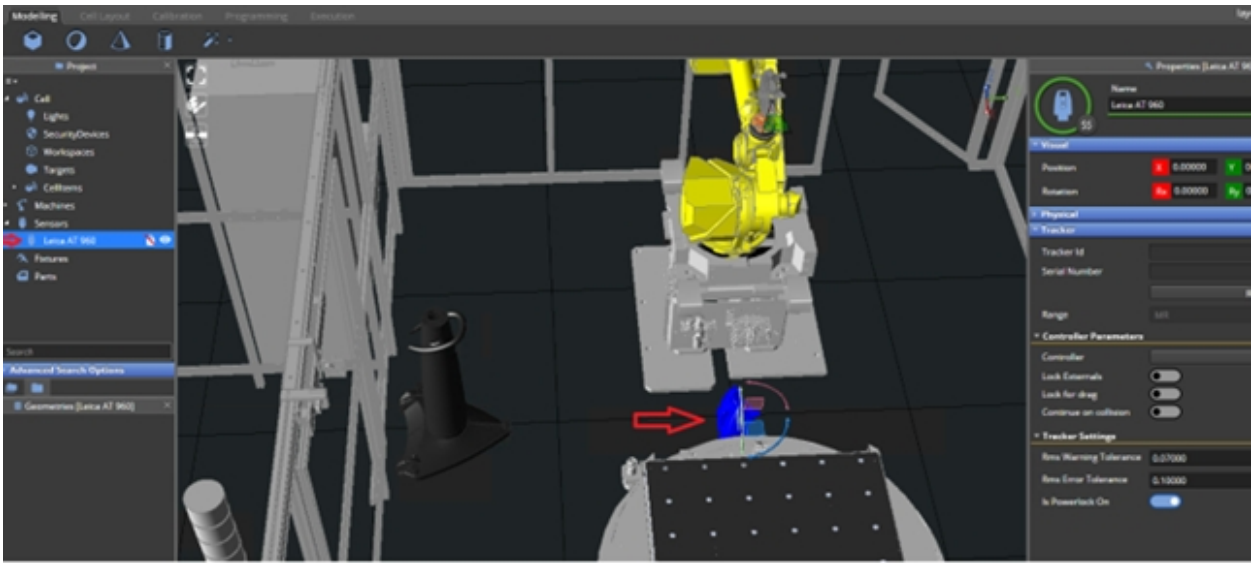



## Adding a Sensor to the cell

Right-click on the Sensors icon ( ) and select *New* | *Tracker* | *Leica AT960*.

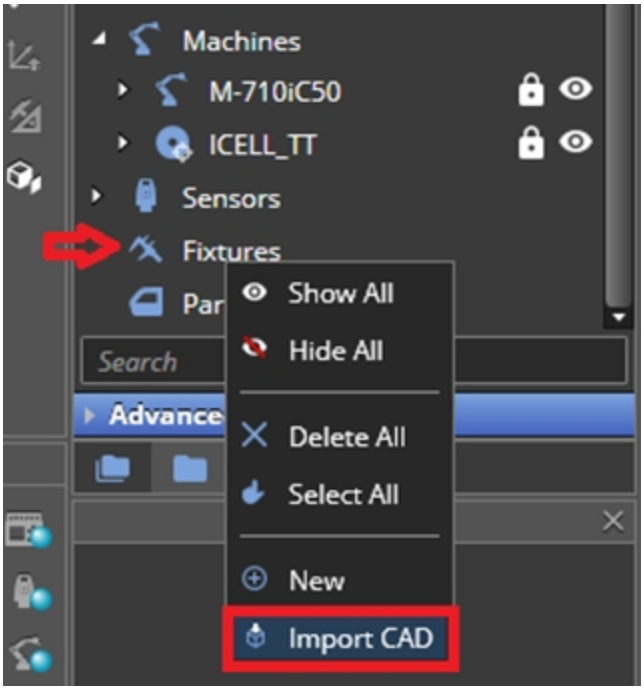
The Leica AT960 will be added in its default position as shown on its properties.

The alignment tool (see chapter Arranging_items_into_the_cell) can be used to position it as it is in real on top of the Leica Tracker stand under **Cell Items**.

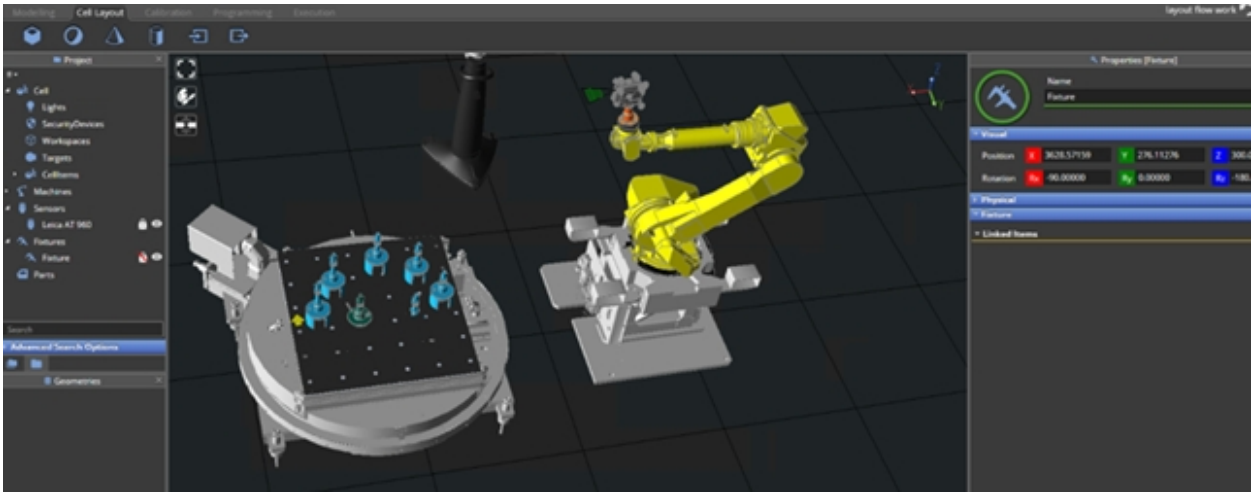


## Adding a Fixture to the cell

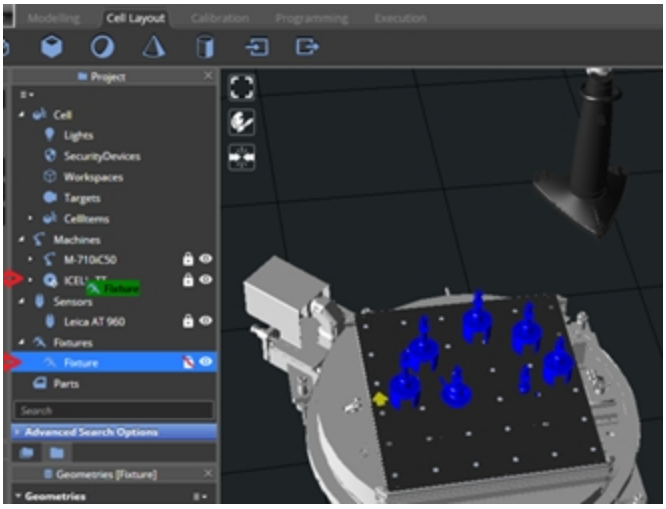
Right-click on **Fixtures** and **Import CAD**. Alternatively, you can use Import CAD icon ( ) in Cell Layout tab and once the CAD appears under cell item, you can assign graphic to it.

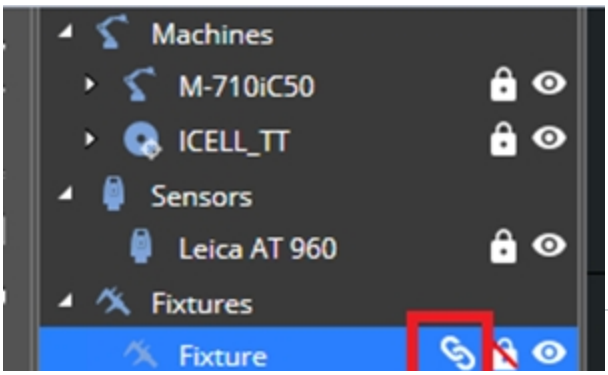


Align the fixture using the alignment tool (see chapter Arranging_items_into_the_cell) to position it on the Rotary Table.



Link the Fixture to the Rotary Table by selecting the fixture with the mouse and dragging it under the Rotary Table until it will turn to green, then drop it. The linked icon ( ) will appear next to the fixture, meaning that it is linked.





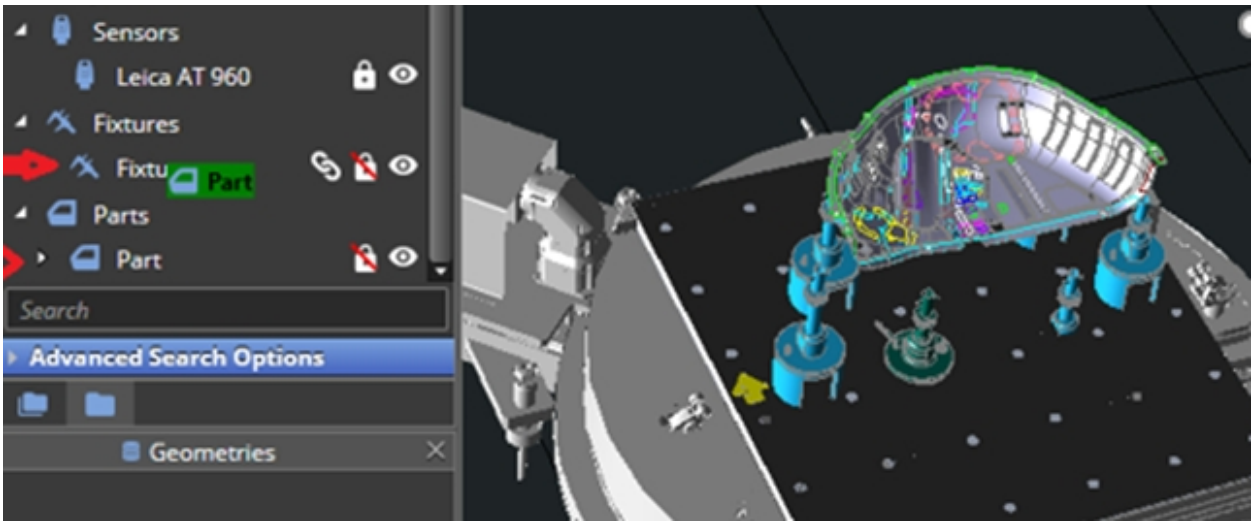
## Adding a Part to the cell

It is recommended to import the part from the metrology software.

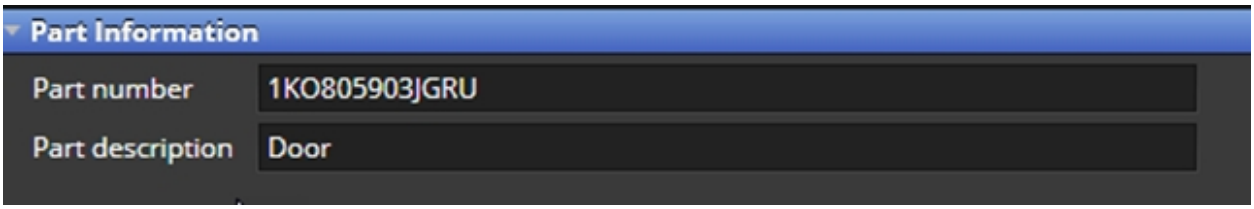
Once the data from the metrology software project is transferred to the HxGN Robotic Automation project, the part's CAD will be added to a new part element in the tree view.

In alternative, the part can be directly imported in HxGN Robotic Automation by right-clicking on Parts and **Import CAD** and use the alignment tool (see chapter Arranging_items_into_the_cell) and place it on the fixture.

Link the Part to the Fixture by selecting the part with the mouse and dragging it under the Fixture until it will turn to green, then drop it. The linked icon ( ) will appear next to the fixture, meaning that it is linked.

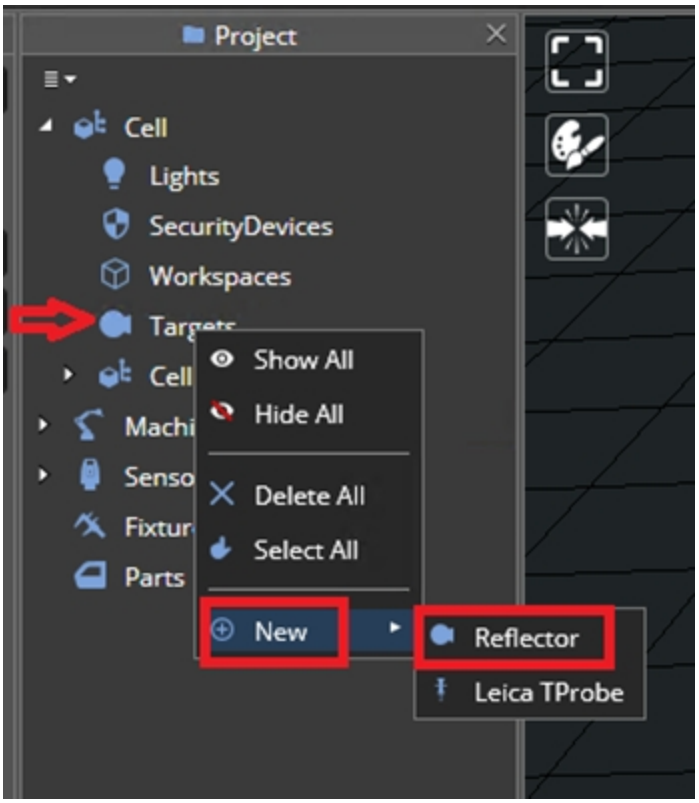



Within the part properties, there is a section to add the part number and optionally a description. The part number is mandatory and will be requested upon exporting to Execution Manager in case it has not been added here.

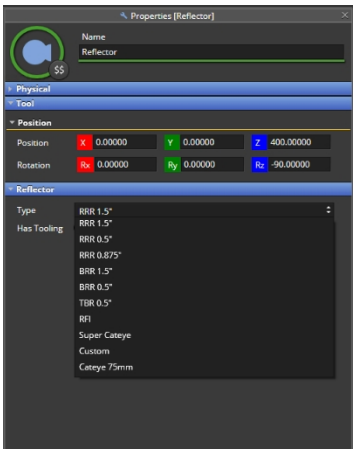


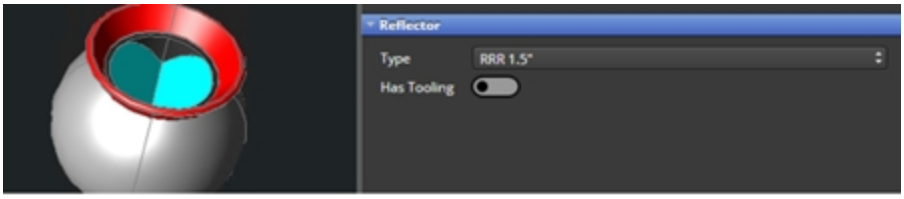

### Adding Targets to the cell

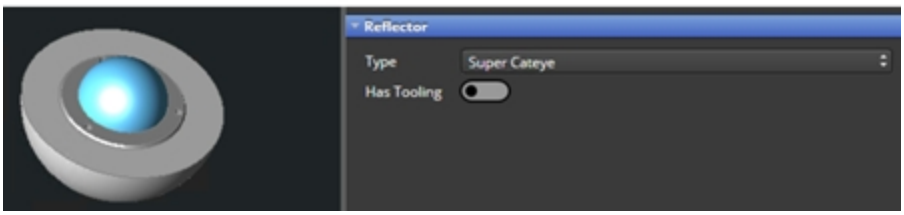
Under Cell after expanding, right-click on *Targets* | *New* | *Reflector*. Reflectors are used for Alignment and Rotary table compensation.




Add as many reflectors as the real cell has, double click on them and set their properties, e.g., by setting their position on the Rotary Table. Their type can be selected from the Type drop-down list.




## Defining Collision rules among cell items

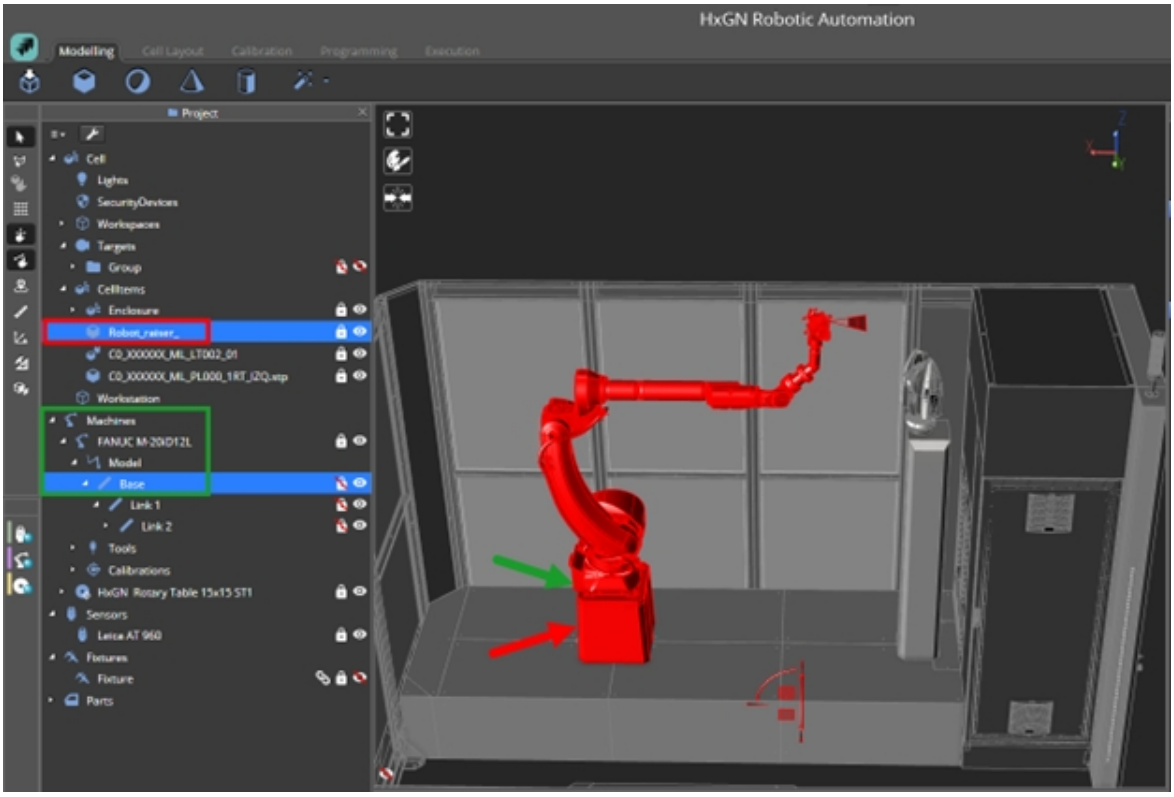
HxGN Robotic Automation provides a collision engine that is checking if cell items are colliding with each other in the 3D view. Collision rules are set up by default among all cell items but can be freely defined by the user.

Between dynamic machines that should ignore collisions with each others, it is important to **ignore collisions between the individual links** rather than the machine. In case you add an ignore collision exception to the machine of a whole, this will display in the UI, but won't have any effect! Similarly if you ignore a link with a whole machine, this will display in the UI, but won't have any effect.

To create a collision rule among two cell items, the user needs to:

1. Select the cell item where he needs to create a collision rule for, e.g., **Robot M-20iD12L** as shown below.

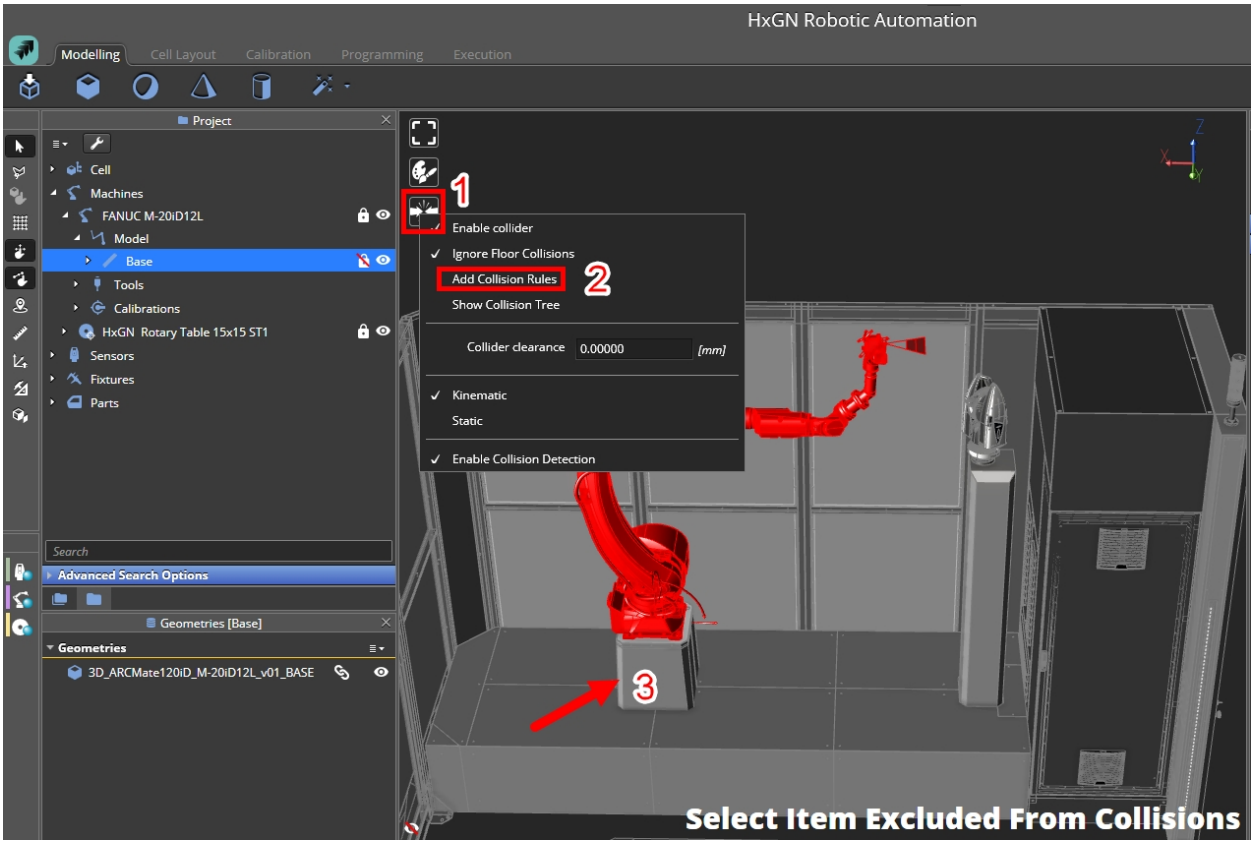
   Select a part of it, e.g., its Base can be selected to create a collision rule between it and the robot raiser (**Robot_raiser_** in the example).

2. 

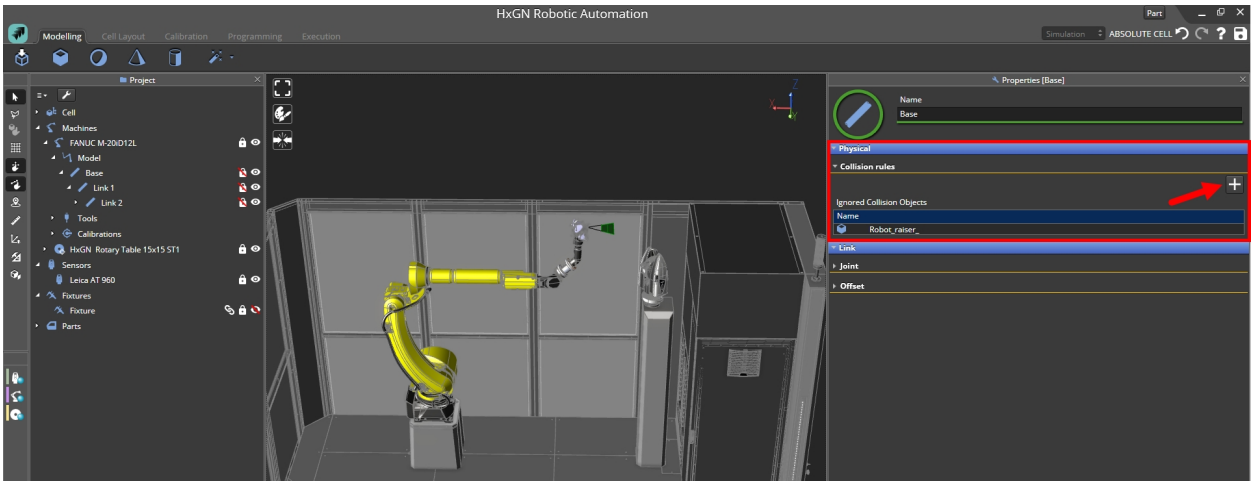

Then the user can either:

1. Click on the **Collisions Overlay** Icon in the 3D View
2. Select **Add Collisions Rules** from the Context Menu
3. In the 3D Viewer click on the CAD of the item to be excluded from the collision check for the Robot Base, in this example, the robot stand is excluded.

Alternatively, the user can select the same item in the project tree. In both cases, it is needed to press the [ENTER] key to confirm.




Collision rules can also be added inside the item properties by clicking the + button and following the same as described above.




The collision rule will appear within *Physical* | *Collision rules* | *Ignored Collision Objects*.

This means that the collision engine will not check for collisions for the items that are available within the list of **Ignored Collision Objects.**

## Arranging items into the cell

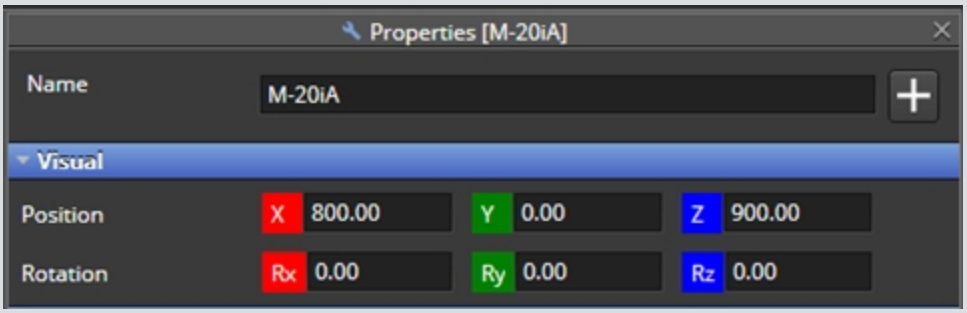
### Alignment Tool

Using the Alignment tool user can place/align items in the 3D Viewer relative between each other, based on alignment constraints as:

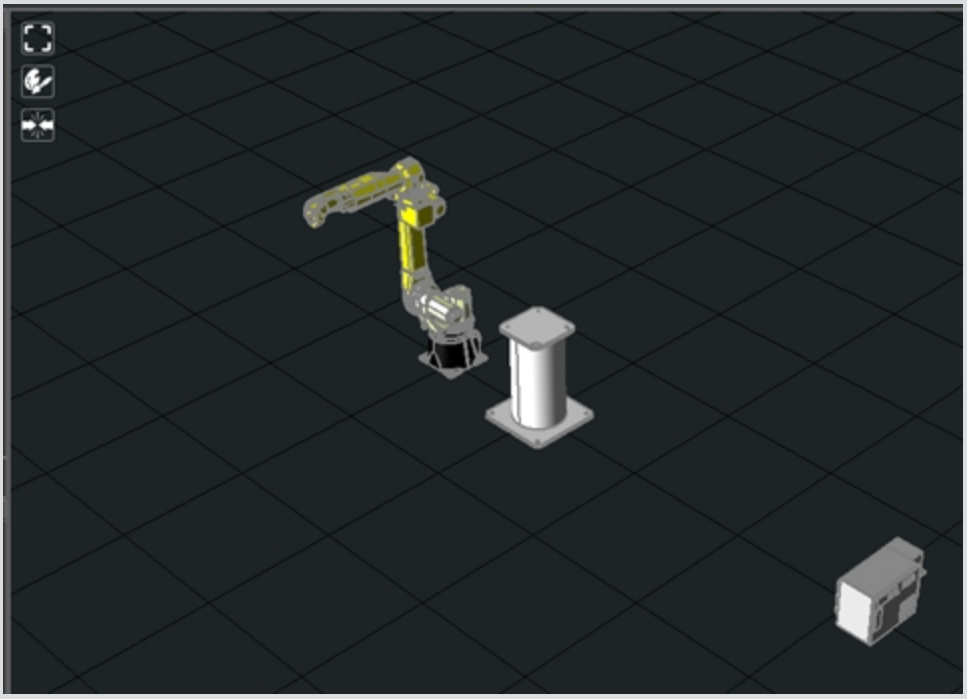
- Parallelism
- Concentricity & Coaxiality

Example: How to Align a Fanuc M-20iA Robot to a Robot Stand

On the Items Tree click on the Library Icon and expand the Machines Folder. Expand the FANUC Folder, select the M-20iA Robot, drag it, and drop it in the 3D Window. Change its position to the initial coordinates:

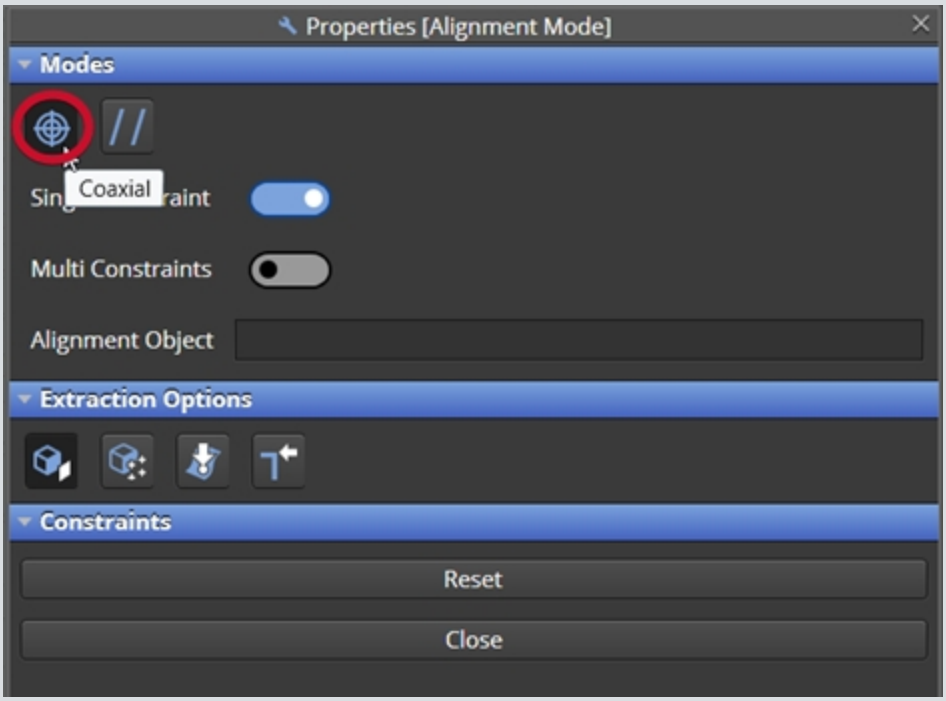



And the 3D Window should look like the image below:

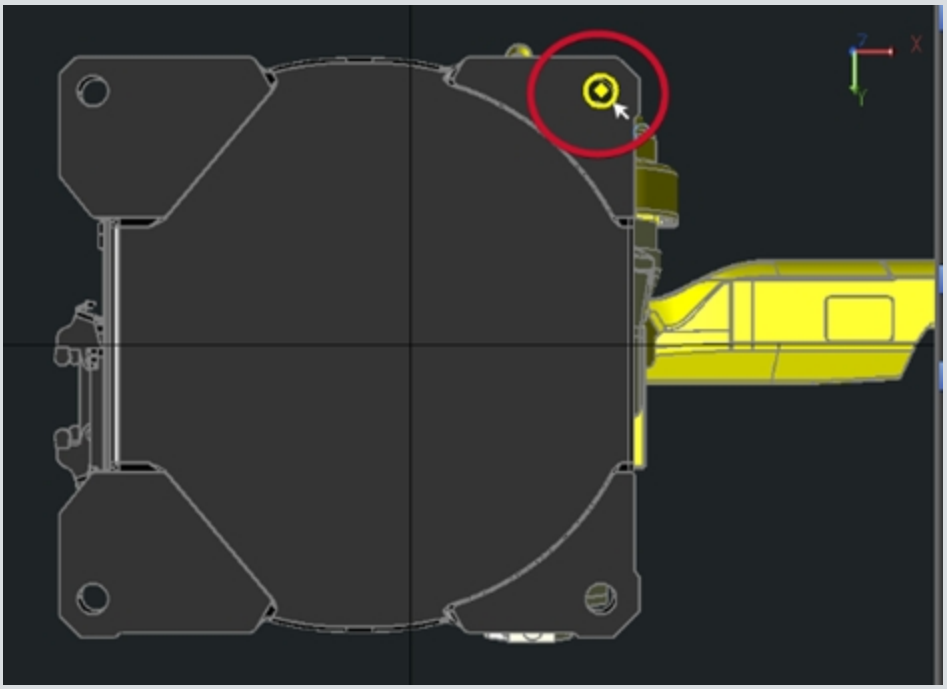


Next step is to align the robot to the top of the robot stand. To do this it is needed to activate the alignment mode by clicking on the Alignment Mode Icon in the Left Panel and generate a **Coaxial** constraint between the Robot Stand and the Robot base to place the Robot on top of the Stand within the Properties window.

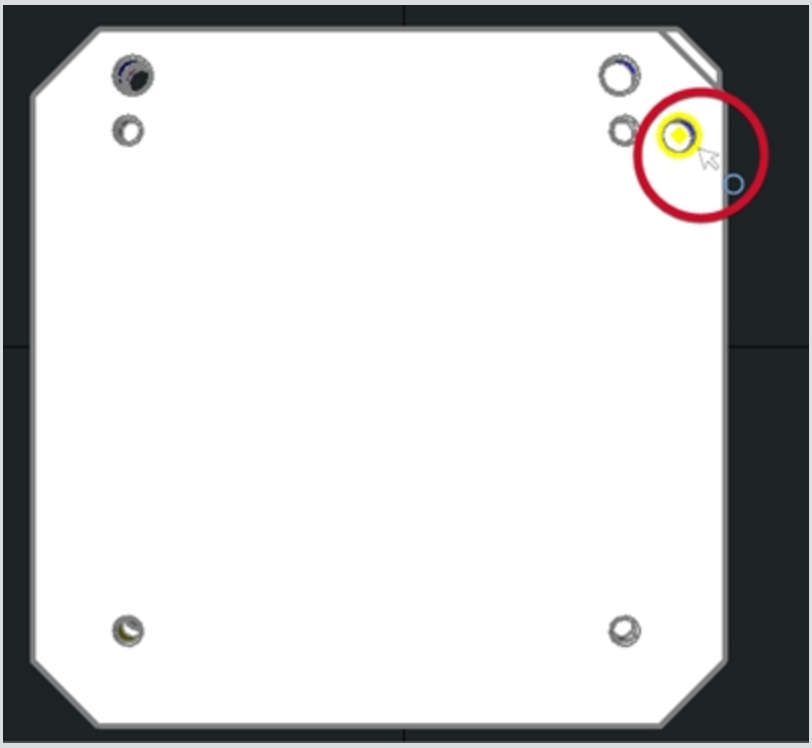
The Robot is the item that will be moved to the Robot Stand (which is the item that is fixed).



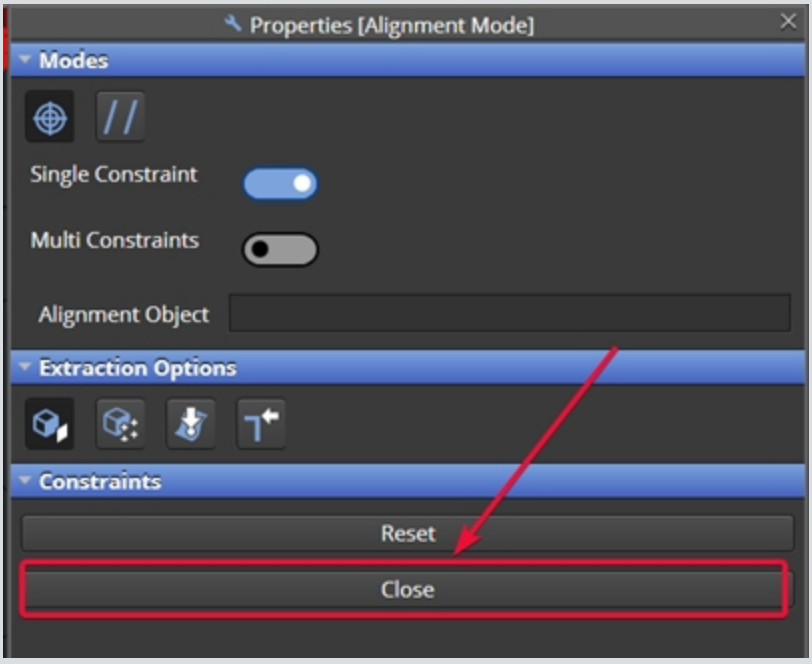

Rotate the 3D Window view mouse over the Robot Base bottom face and select the circle displayed on the image below:



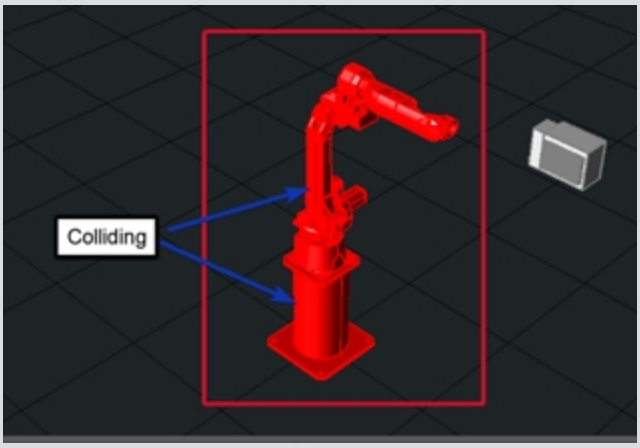
And then rotate the 3D Window view, mouse over the Robot Stand top face and select the Circle displayed on the image below:



Once done we can close the alignment mode by clicking on Close button.



After clicking on the base of the alignment the robot will move on top of the robot stand and will show collision with it.



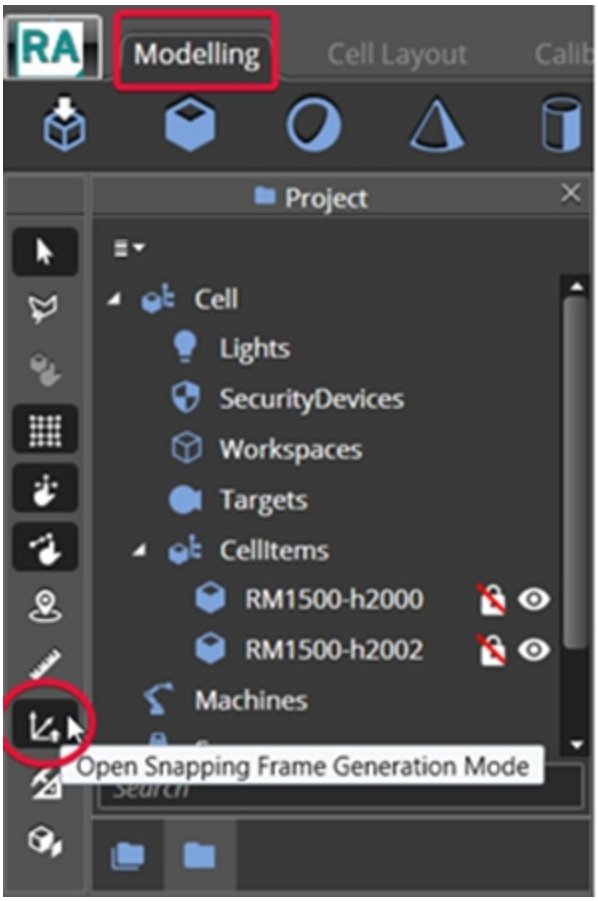

A new collision rule between the Robot Base and the Robot Stand should be created as explained in chapter Defining_Collision_rules_among_cell_items so these collisions are filtered.

## Snapping frames

A snapping frame is a point added to a cell item.

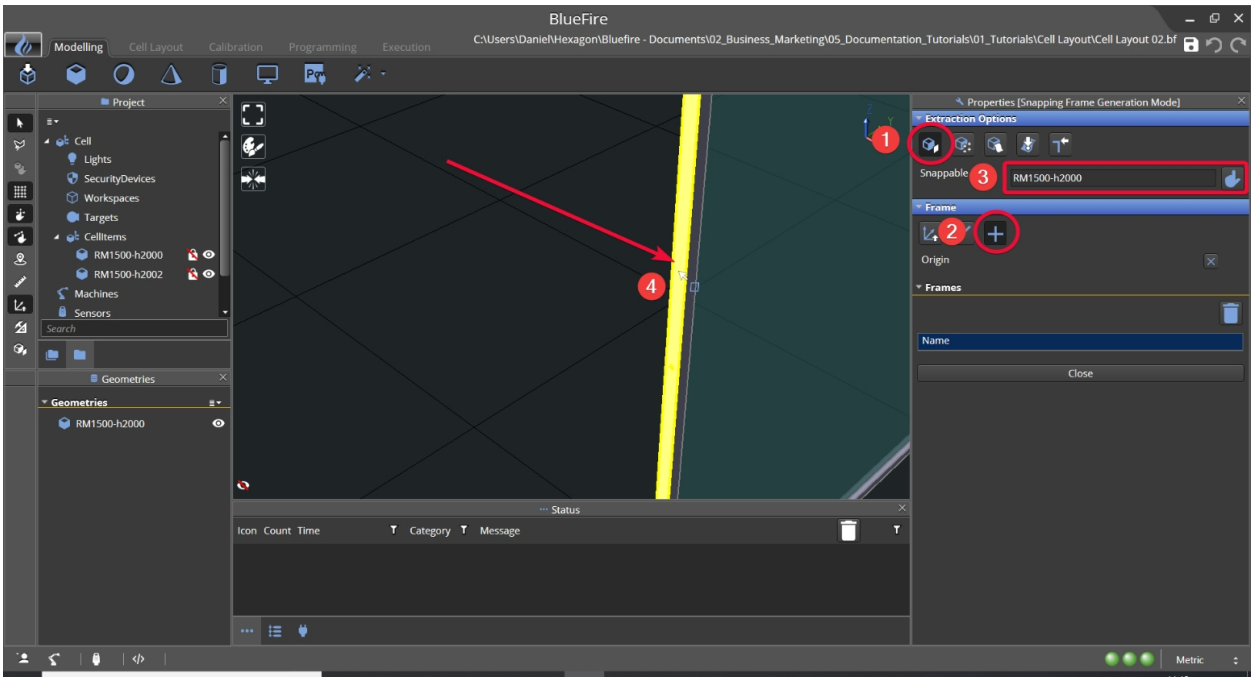
Once two objects with snapping frames get closer, a bubble appears and if we bring them close enough and stop dragging, the items will be aligned by the snapping frame.

The snapping frame generation mode can be opened in the **Modelling** tab.

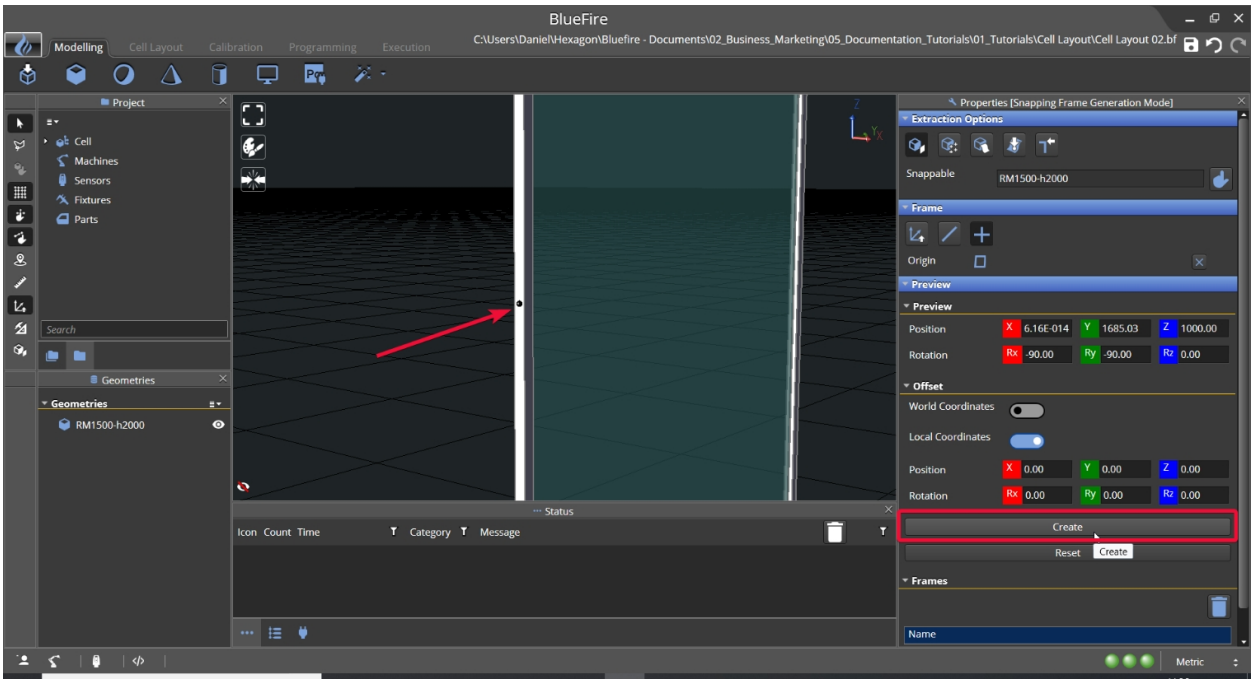



The following steps must be followed:

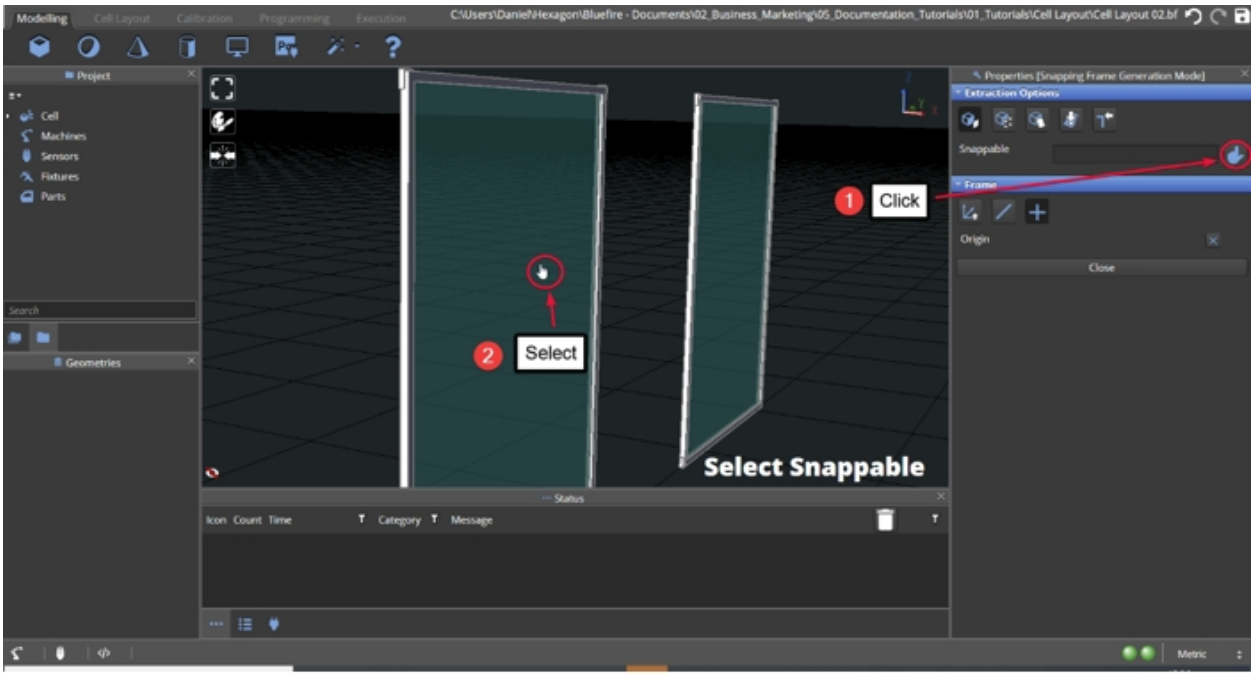
1. Select feature on extracting options.
2. Select point on frame.
3. Select the fence where you want the snapping frame to be generated.
4. Zoom in until you can select the side of the fence.



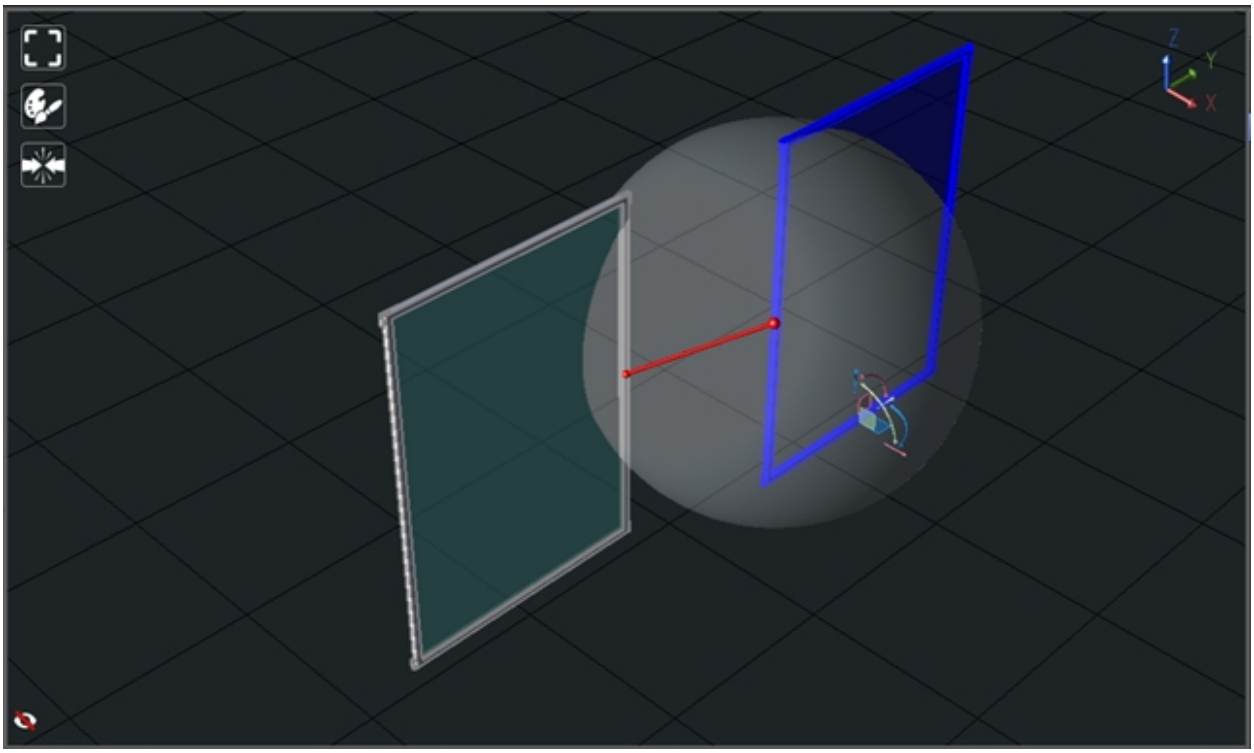

Once selected, a point will appear, and it is just needed to click on it and the frame will be created.



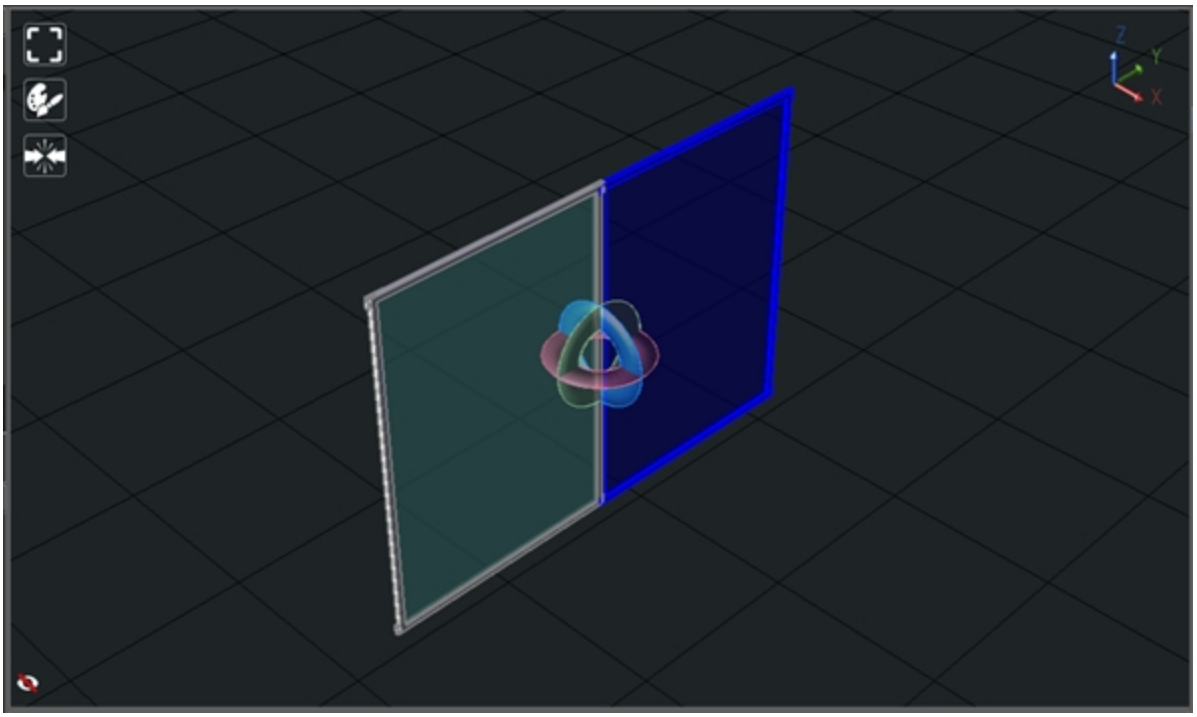
Then click on the other fence and create another frame on the opposite fence:




Then drag it near the other fence.



After releasing the fences will be automatically attached by the snapping frame point.



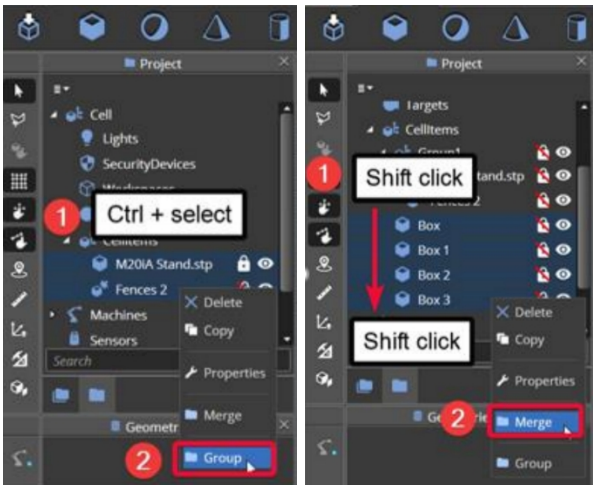
## Grouping and Merging cell items

When Modelling a Machine or Building a Cell Layout there is often the need to **Group** Items to keep them under a same section or folder or **Merge** Items to make them a single assembly. After selecting the items and right-clicking on the selection, it is possible to either merge or group them by clicking the respective button.

### Items selection

Items within the project tree can be selected in two ways:

1. Selecting them from the 3D Window by clicking and pressing the [CTRL] key for multiple choice.
2. Selecting them in the Project Tree using the range choice, clicking on the first element of the selection, and then pressing [SHIFT] key clicking on the last element of the selection.

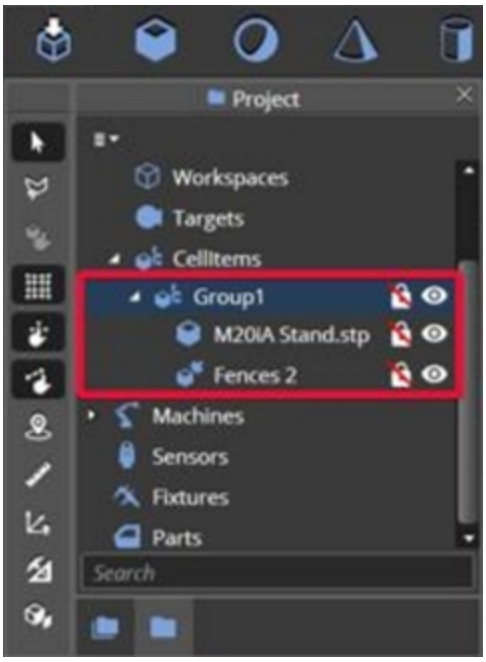



## Grouping items

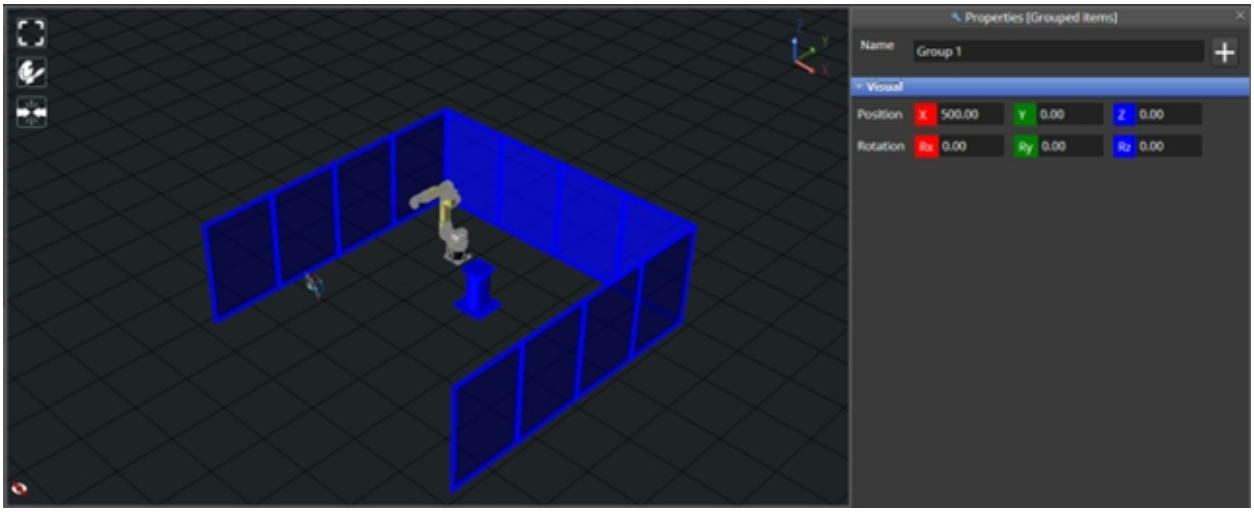
Grouped items properties can be edited individually or all together as a Group.

Consider for example the following project where two items are grouped under **Group1**:

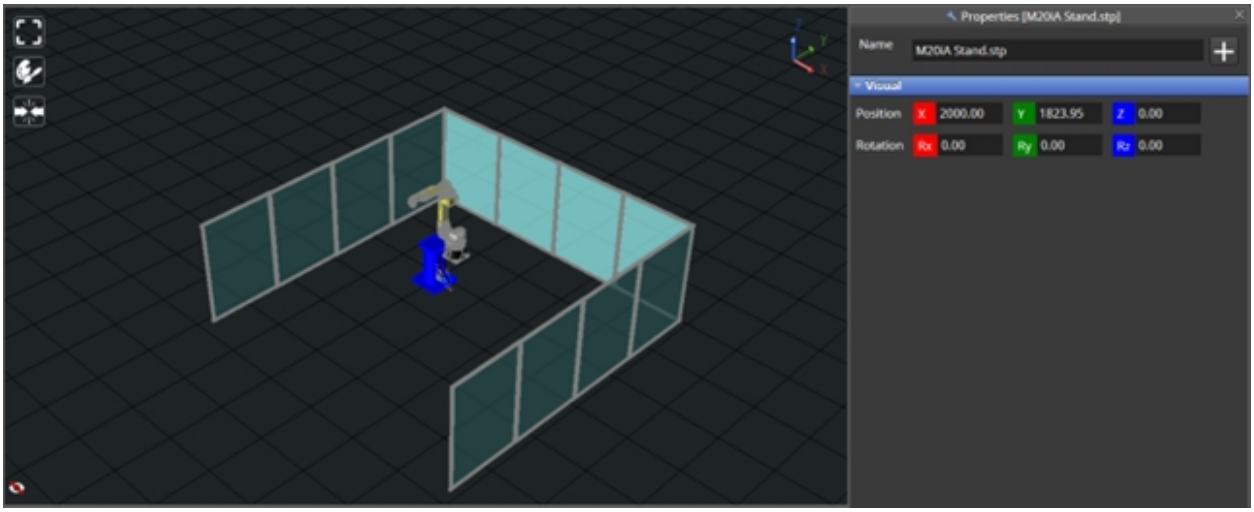
- M20iA Stand.stp
- Fences 2




It is possible to move the entire Group1 by double-clicking on the group and changing the group position in the Properties, e.g., by moving the group along the X axis by 500 mm.



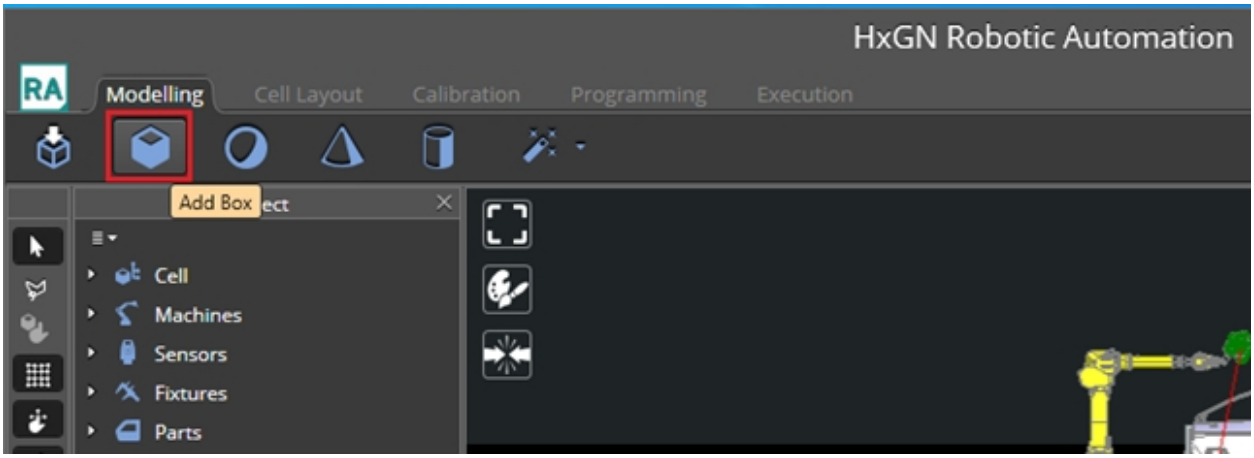
Even though the Robot Stand is in the Group1 it can be edited individually without changing the rest of the Items in the Group1 position, e.g., by selecting the robot stand and moving it along the X axis by 2000 mm.



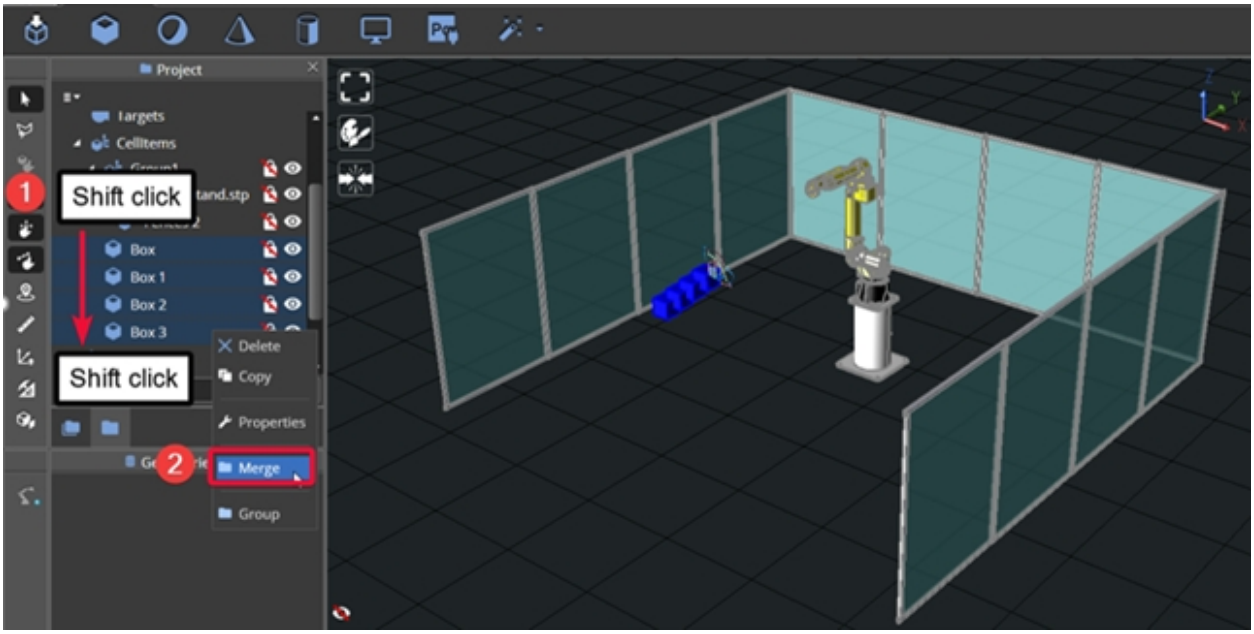

## Merging items

Merged items will belong to the same assembly and cannot be moved individually.

To show the difference between Merging and Grouping it is possible to add four boxes to the Cell by clicking four times on the **Add Box** icon on the top toolbar.



Select the four boxes in the Items Tree using the range selection. Once all the boxes are selected, right-click and select **Merge** from the context Menu.




A New Merged Cell Item is created that has the 4 Boxes in a Single Item. None of the boxes can be translated individually.

# Alignment

This section is aimed for **Hexagon Application Engineers**. Standard users do not have to calibrate robotic cells; they usually start using HxGN Robotic Automation already configured to work on the robotic cell where it is installed, with the cell aligned and ready to operate.

Cell Alignment is a critical step of performing the fine tuning of the cell model that was created virtually, and its goal is to match real cell with virtual cell.

Please select the relevant chapter for your cell alignment. This will depend on your hardware

- Alignment - Leica.htm
- Alignment - ZG.htm

## Alignment - Leica

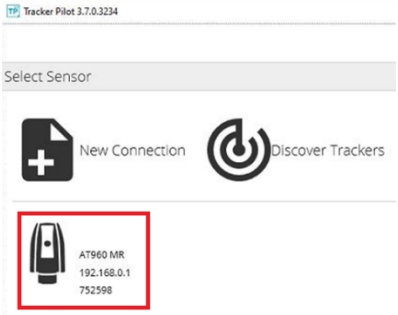
### Prerequisites

- Leica Metrology Foundation (LMF) and Tracker installed on the system (AT 960-MR)
- A Reflector (target) Type RRR (Red Ring Reflector) 1.5” (Check if it has tooling and its type).
- Cell model imported to the online Robotic Automation system.

The cell alignment will use the tracker as a reference for aligning the cell, therefore, make sure to position the Tracker in X, Y, Z coordinates as much as possible to match the real cell.

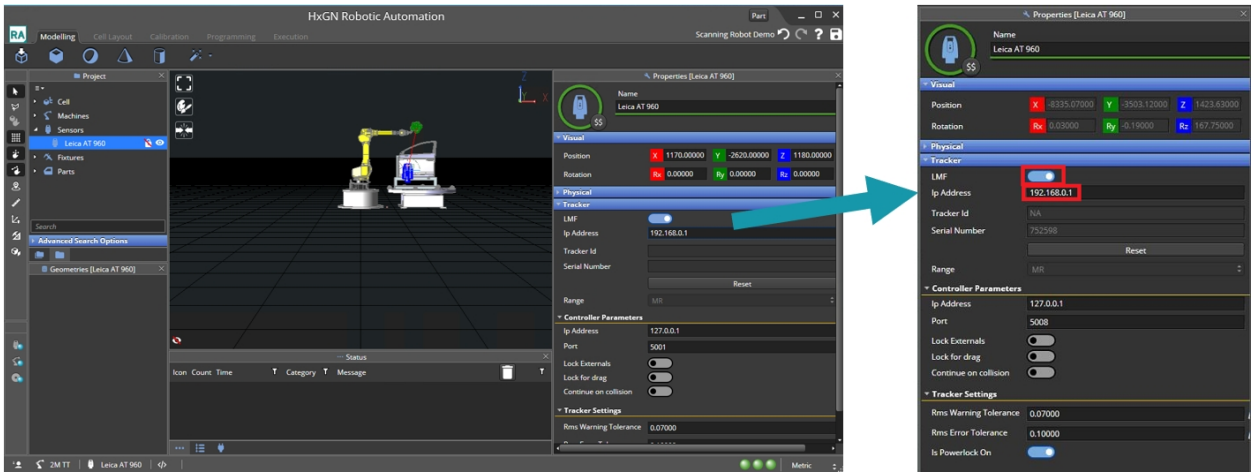
Alternatively, you can position it in 0,0,0 coordinates and make sure not to move it during the complete process to complete alignment of all cell items. In this case the origin of the tracker reference system will coincide with the World reference system.

Execute the Tracker Pilot software in the PC, select the Sensor and obtain the IP address. If not visible, clicking on Discover Trackers will show the available tracker information.




Open HxGN Robotic Automation application and import the cell model that needs be aligned.

Double click on the Leica AT960, update the IP address, and enable the LMF with the relative toggle switch:



## Manual alignment workflow

The manual alignment process is done clicking on Open Alignment Mode under the Special Mode Toolbar (see chapter UI Overview.htm).

This process can be used to align any cell items, machines, sensors, fixtures, and parts in the project and allows the user to manually select and align them in the 3D View.

During the alignment process the user needs to:

1. Measure the real position of any item in the project
2. Align the virtual item positions in the project to the real ones

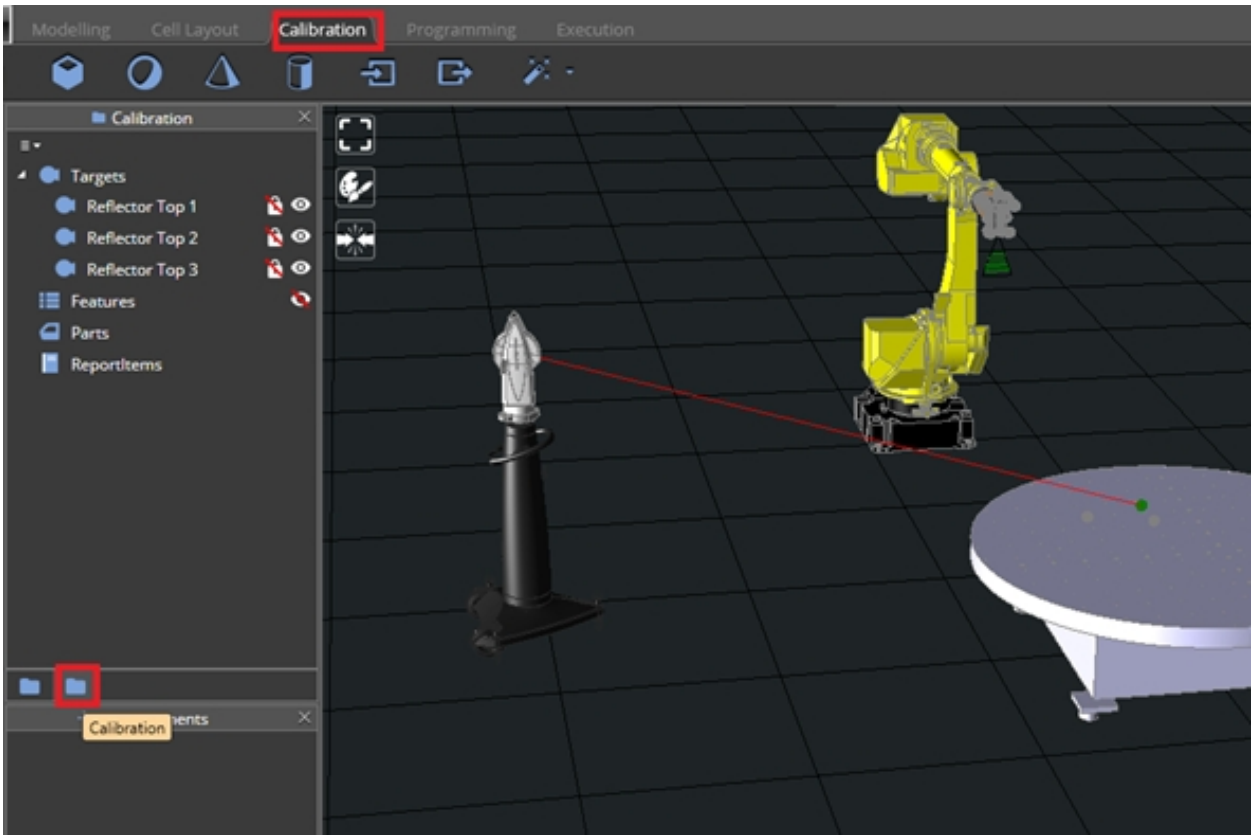
To be able to measure the real item positions, the user needs to connect to the cell tracker (using LMF) and get the points to construct the required feature with the help of any reflector or T-Probe device.

Depending on the cell configuration, the user will have to deal with one or more trackers, which are explained in the next chapters. This chapter covers the alignment workflow for single tracker cells. If the cell configuration has more than one tracker, please refer to chapter Dual_tracker_cell_alignment_workflow.

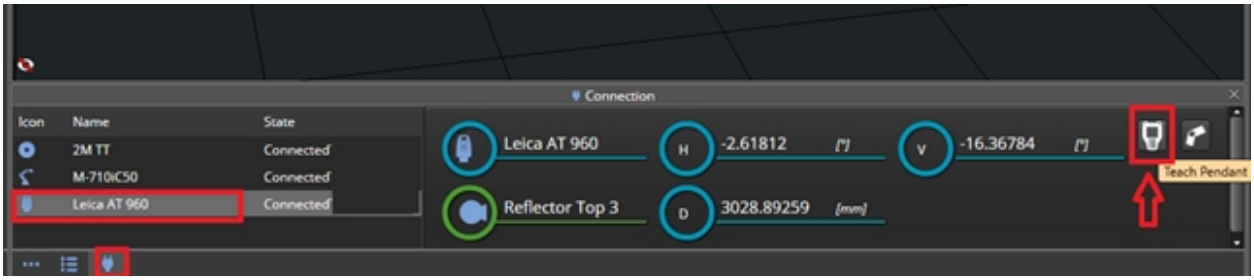
Working with one tracker simplifies the solution as all the measured features are in the same tracker coordinate system.

### Rotary Table alignment using three planes

In this use case, reflectors are used to align the Rotary Table, so they are placed on it.



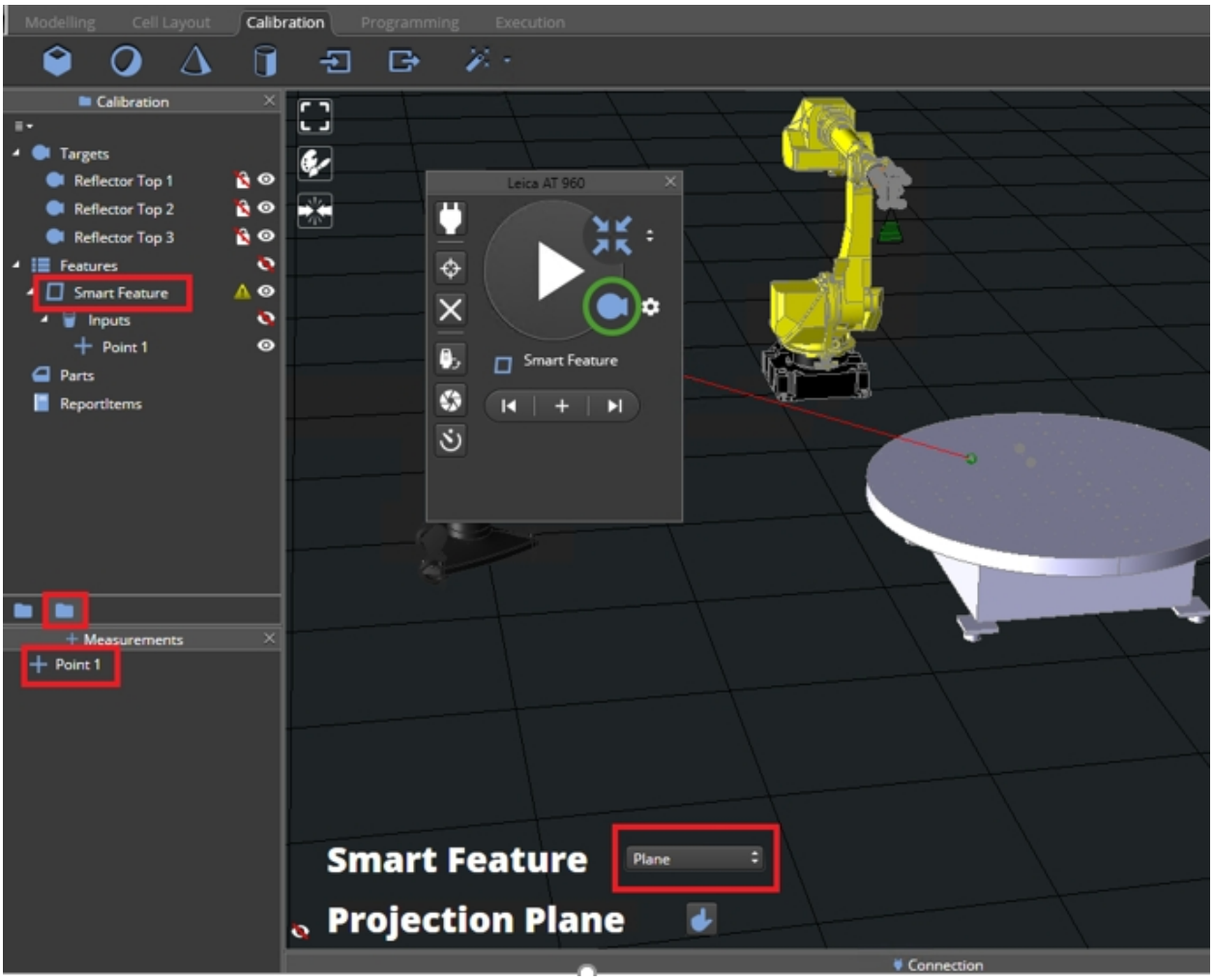

Connect tracker with LMF and open the tracker Teach Pendant, then lock the Tracker on the first Reflector position. In this case, first plane will be the Rotary Table plate and it is recommended to measure at least three points trying to cover the largest surface as possible to minimize the error.



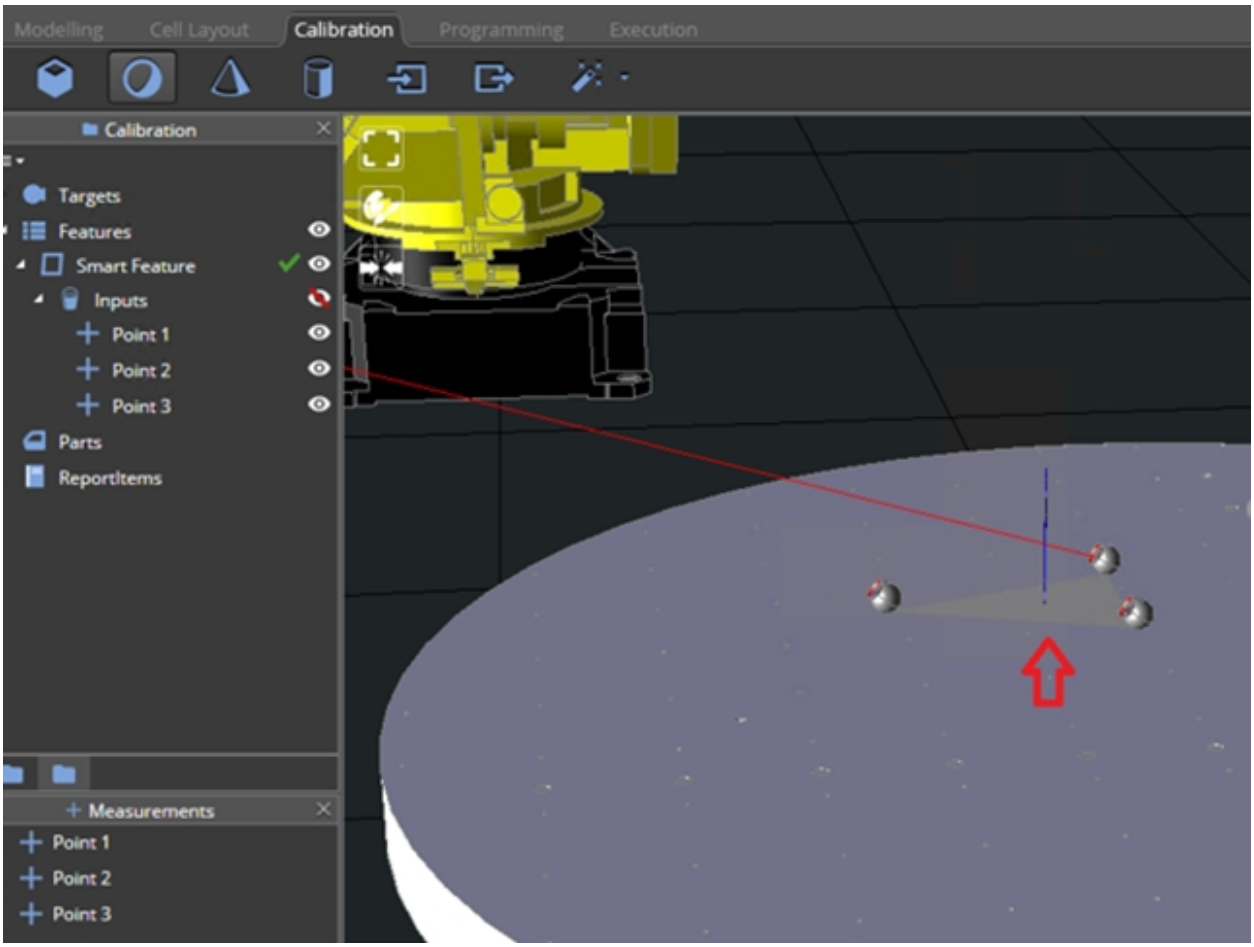

Click on the Measure button from the tracker teach pendant:

A Smart Feature will be created (**Point 1**), make sure to change its type to **Plane** from its drop-down menu.

| | |
|---|---|
| | **Info**: Smart features are specific features that allow the user to collect measurement points (inputs) within a group that automatically identify its type (e.g., two points are needed for a smart feature of type line, three points are needed for a smart feature of type plane, etc.). The type can be also specified by the user using the relative drop-down menu in the properties. |




Lock on the Tracker on second and third Reflector and repeat the measure to get three points, making sure it is computed .

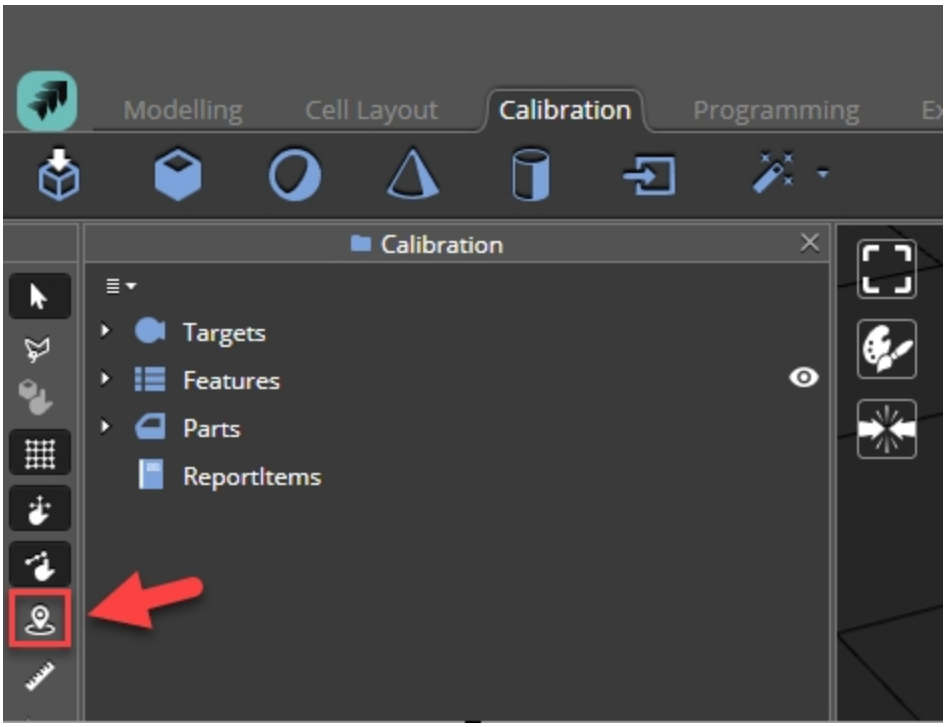


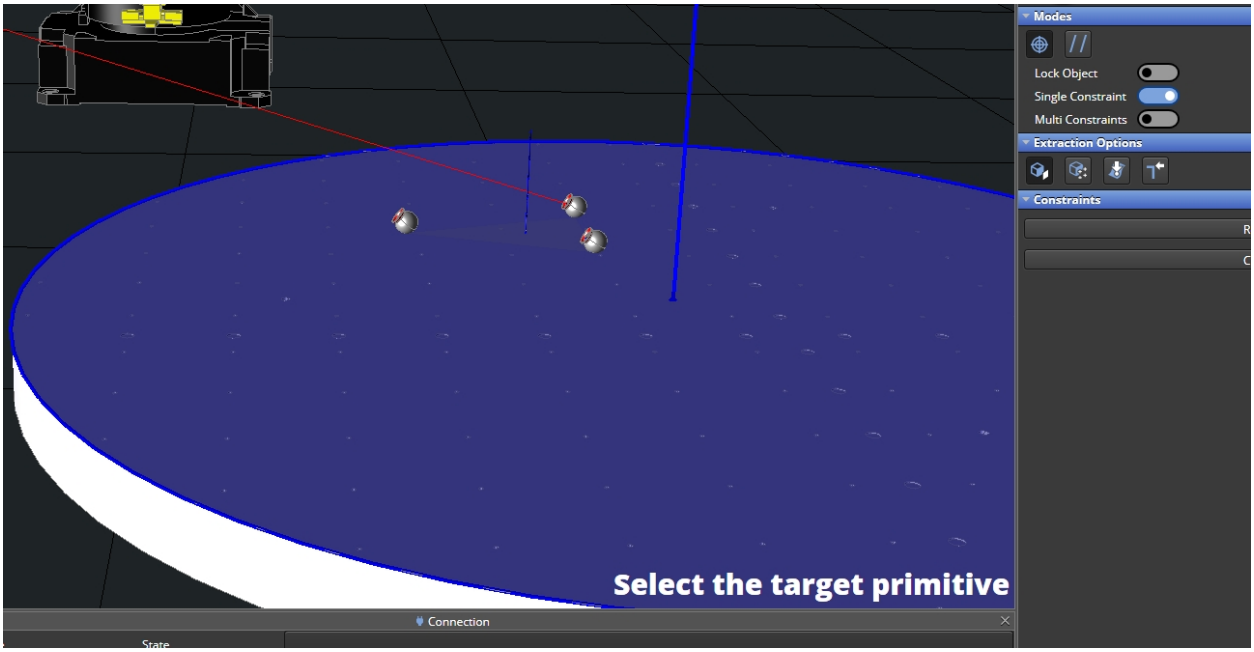

Double clicking on the Smart feature, it is possible to rename it to Top Plane, and verify if the plane setting is set to Below.

Open the Alignment Mode and align the Top of the Rotary Table CAD to the Top Plane feature you created before by first selecting the Rotary Table CAD that you want to move first, and then selecting the target.

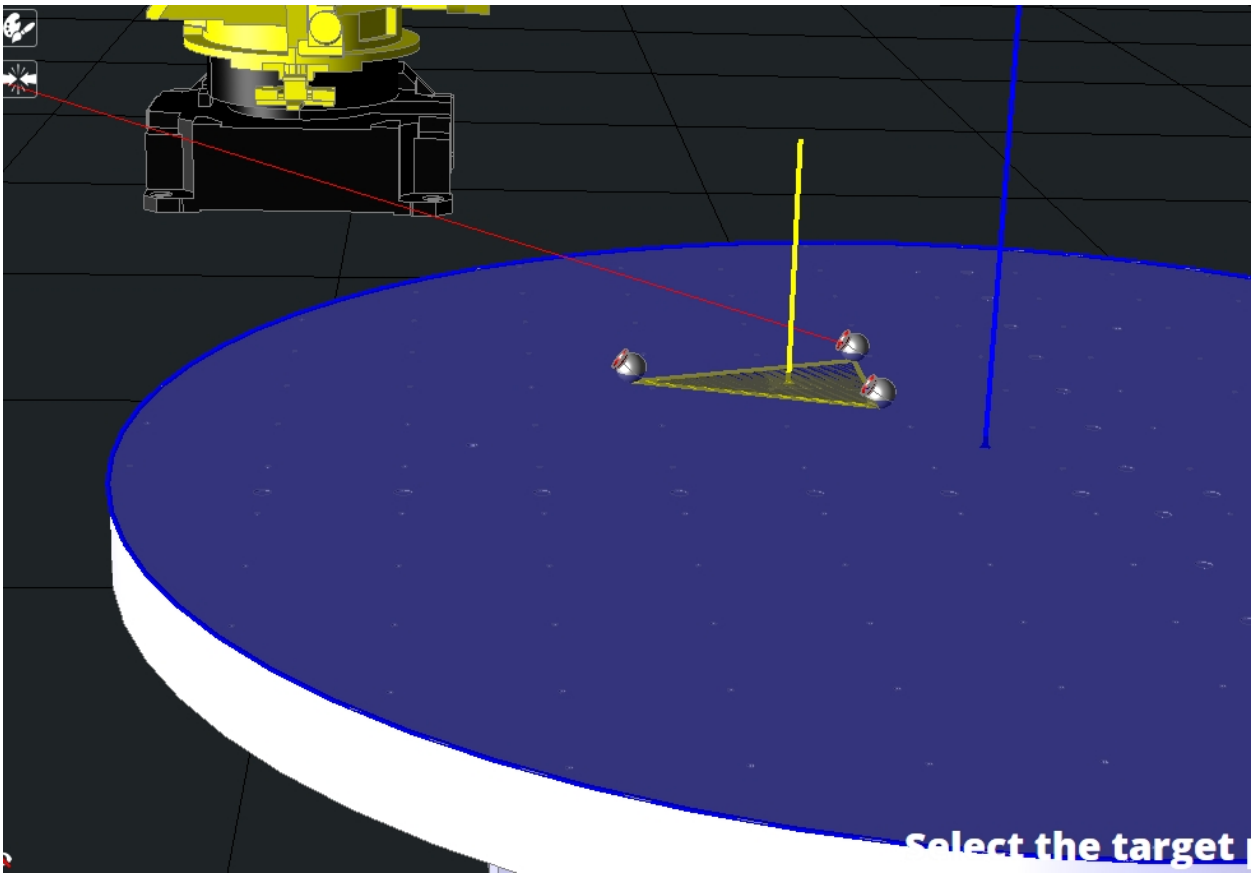
---

| | |
|---|---|
| Alignment | 280 |

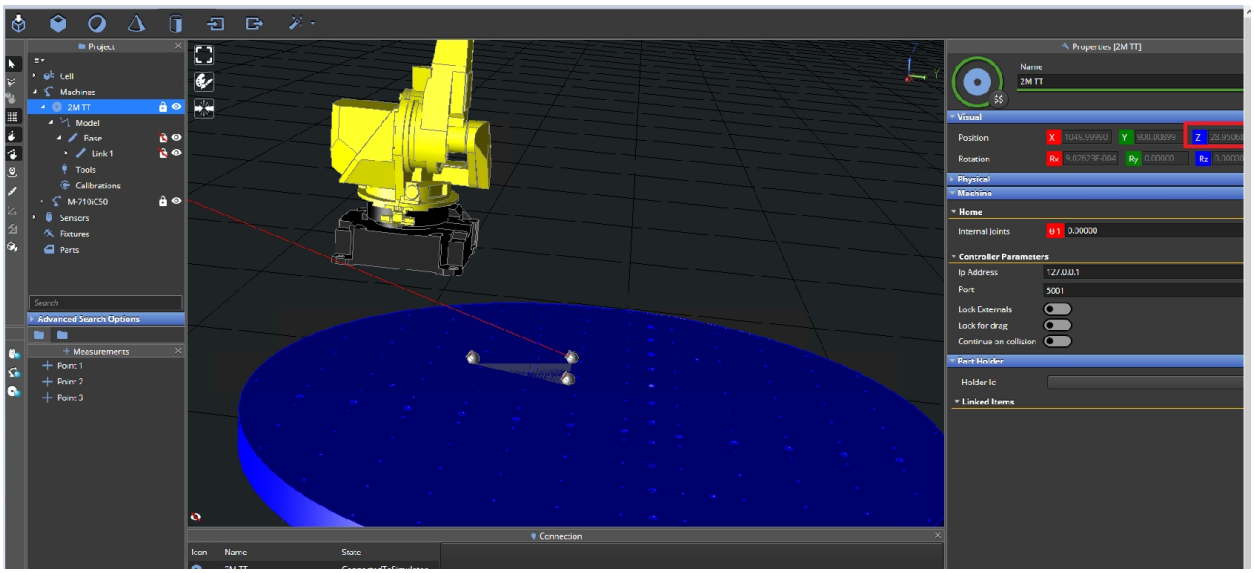



| | |
|---|---|
| Alignment | 281 |

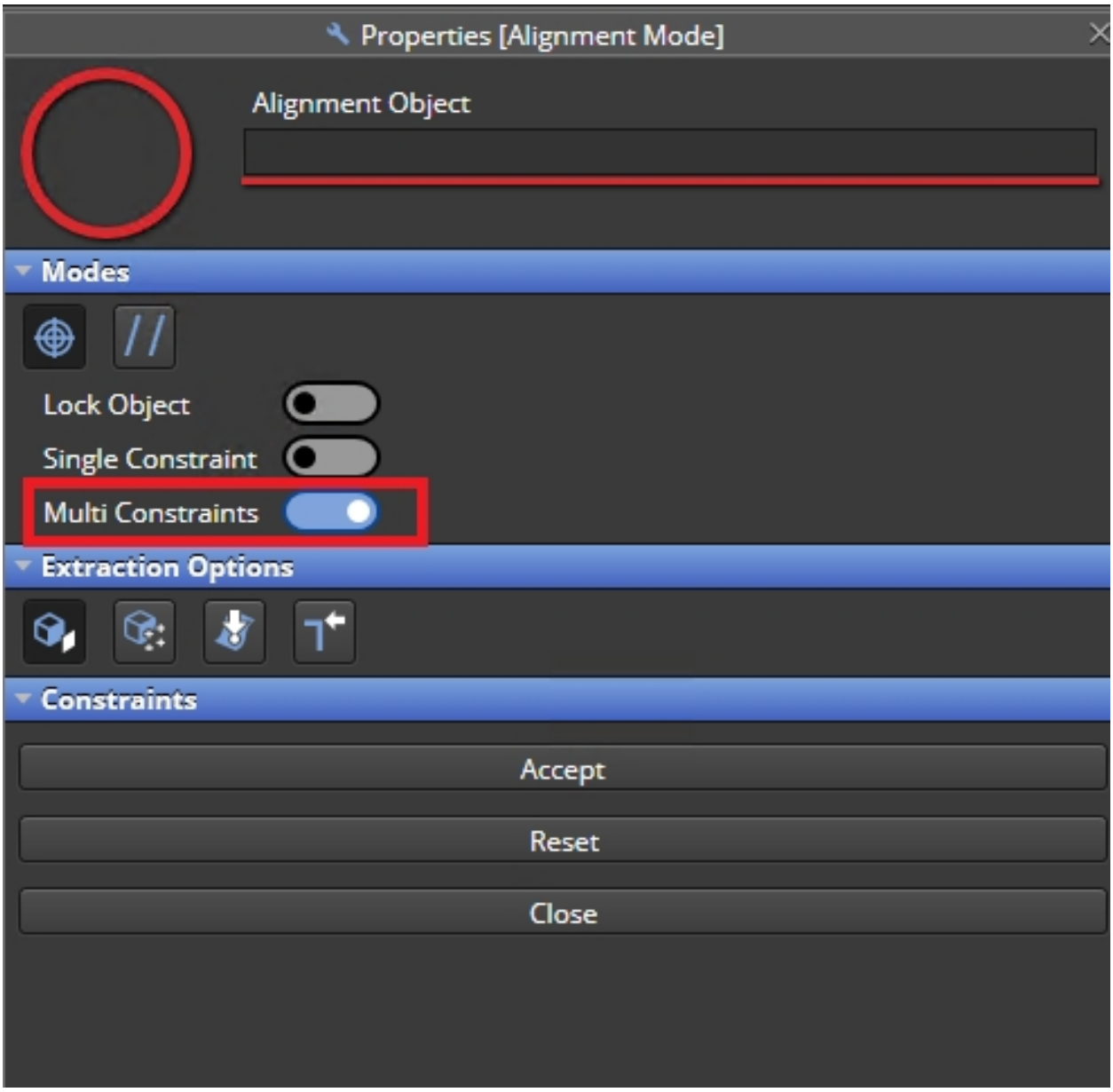


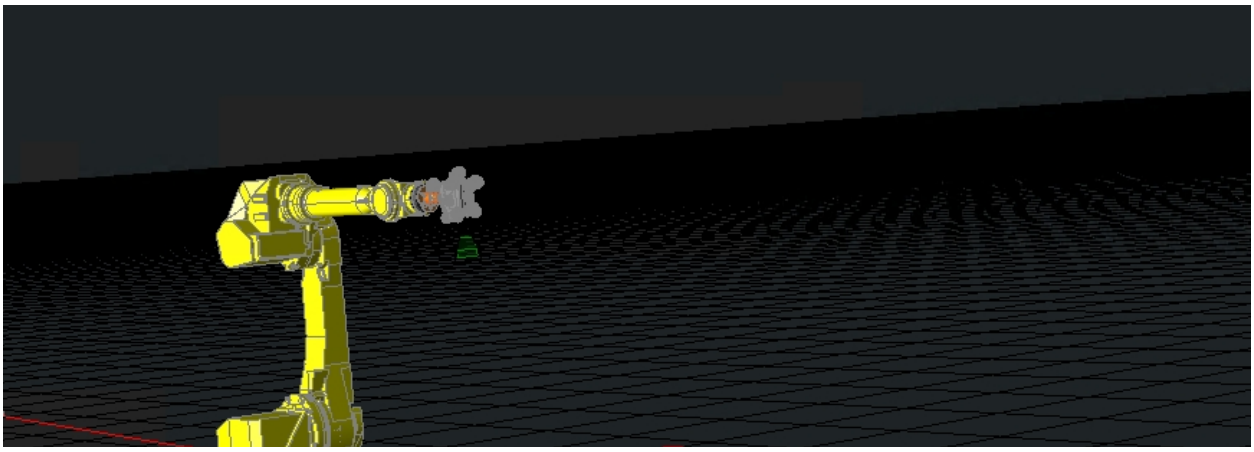

The Rotary Table cad will align with the top plane, and therefore its reference frame will be moved to match the real one that was measured by the three reflectors (in this example only the Z axis will be changed in a relevant way).




Accept when done and repeat the process again with two more planes.

It is recommended to create at least three measured planes, and then perform an alignment using **Multi Constraints** option (other extraction options are explained in chapter Alignment Mode.) by repeating the steps described above to align a plane three times, one for each plane to be measured.



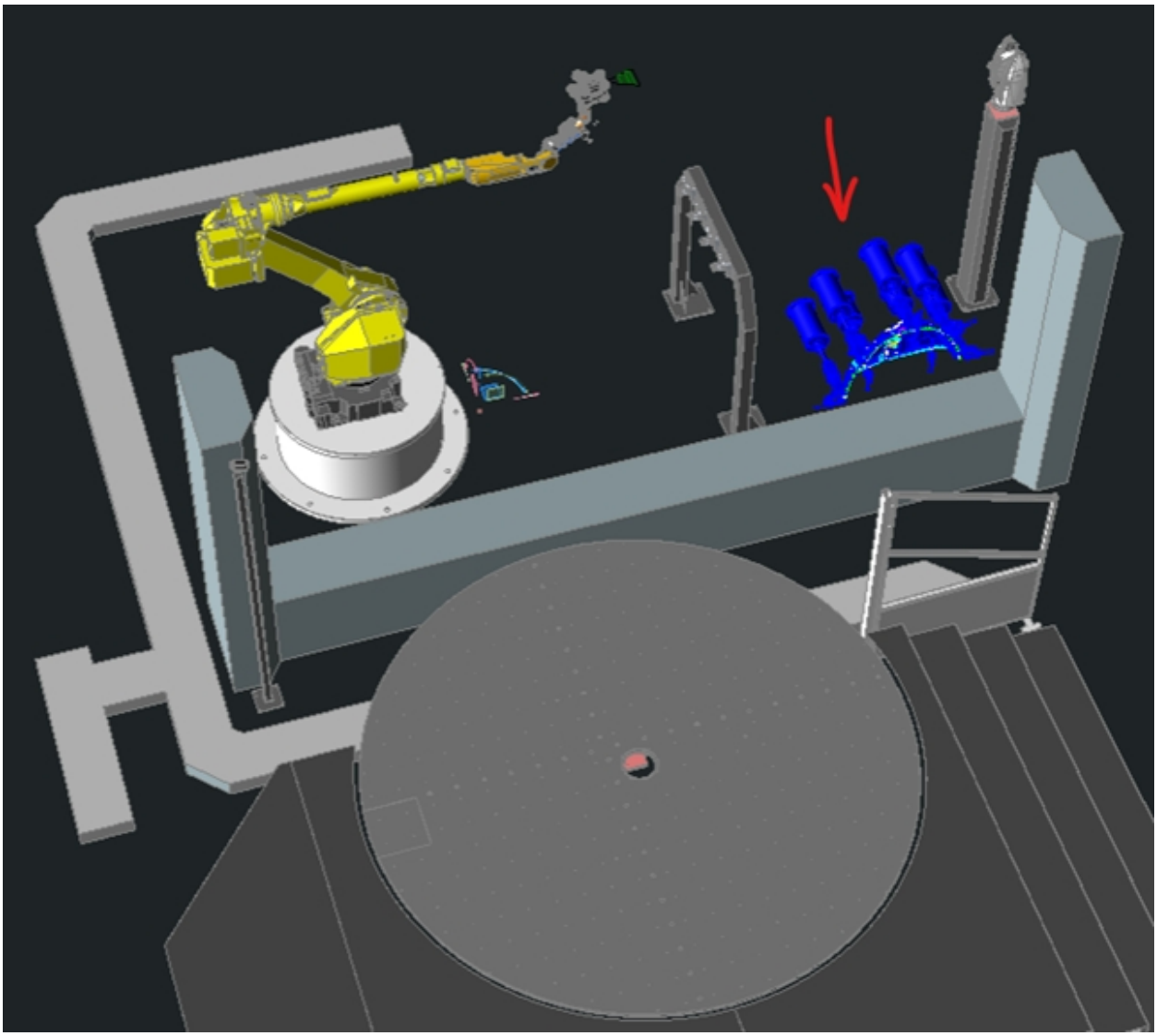



| | |
|---|---|
| Alignment | 284 |

### Part alignment using features

In this use case, the manual alignment is used to align the part to the fixture. The fixture and the part can be added to the cell as described in chapter Adding a Fixture to a cell .

After adding the fixture and the part, they can be moved to the correct position in the cell. In most cases the imported part will be positioned in the right location in relation to its fixture but they both will be in their original design coordinates (e.g., car coordinate system).



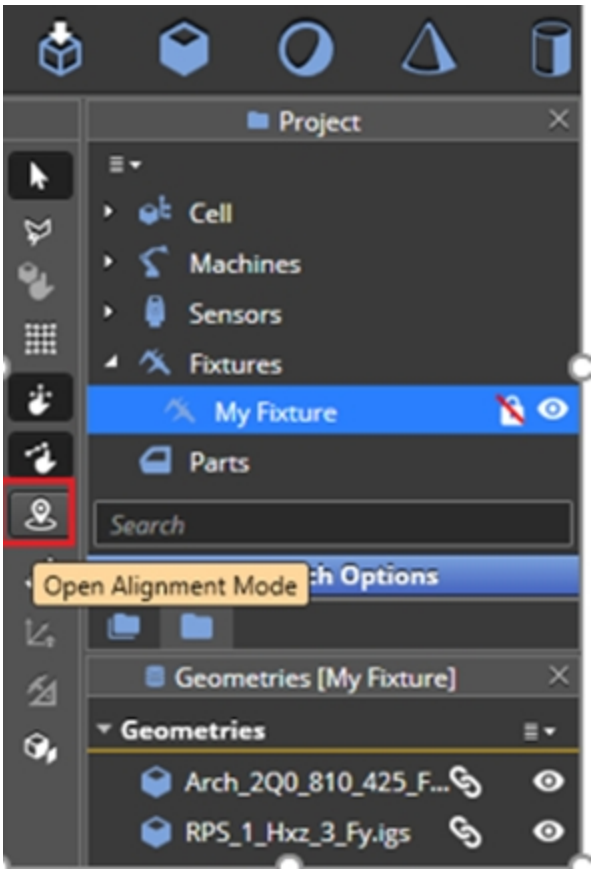
The process of aligning the fixture in the cell consist of selecting geometries on the fixture and align them with the corresponding geometries in the real cell. The fixture coordinate system will be aligned to the reference frame of the corresponding geometries.

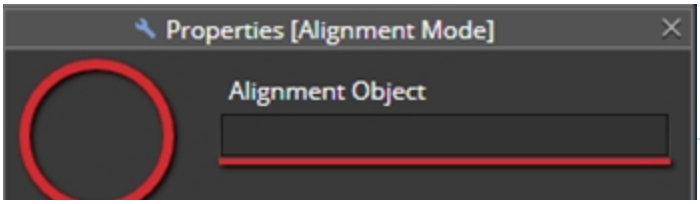
There are two types of corresponding geometries:

---

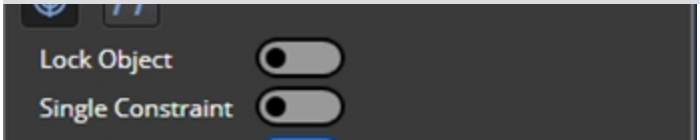
1. **Nominal CAD geometries** – This option is used when the fixture has a distinct position in the cell. In our example, the distinct position is shown by holes in the turntable into which the fixture columns will be fitted.
2. **Actual geometries** – These are geometries that must be measured on the real fixture in the cell. In most cases you would use a T-Probe or a reflector to measure these geometries. To learn more on how to measure actual geometries refer to the section Measuring Mode on how to measure cell geometries.

After opening the Alignment mode, in the Properties it is needed to switch on the **Multi Constraints** mode. Other extraction options are explained in chapter Alignment Mode.

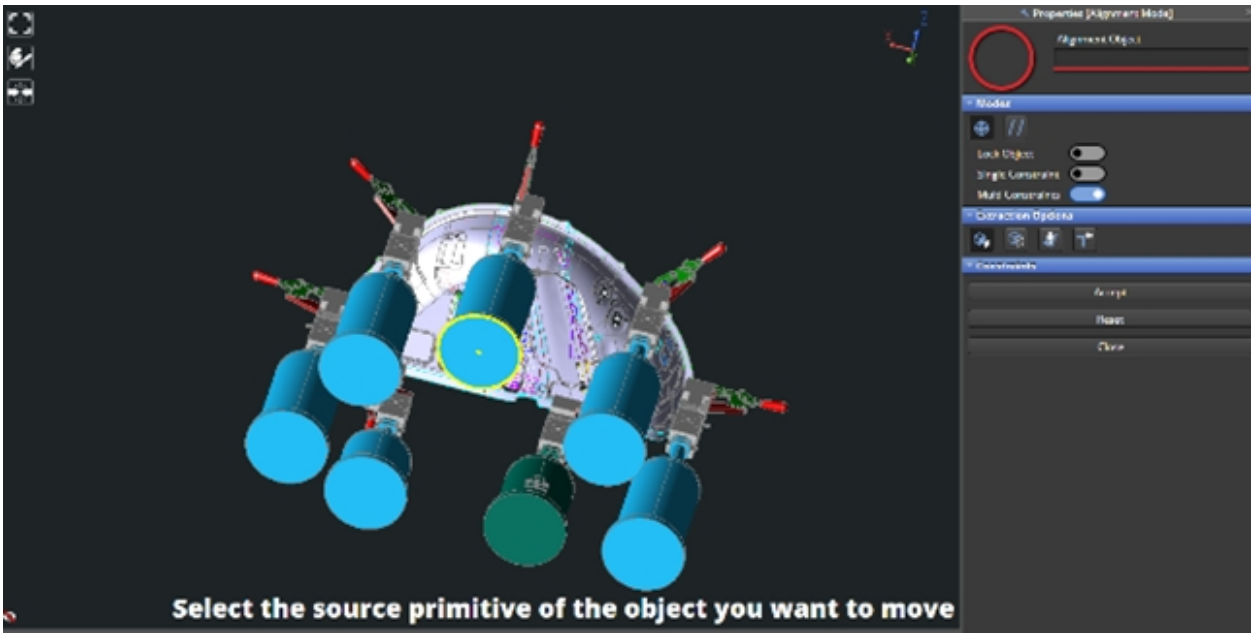


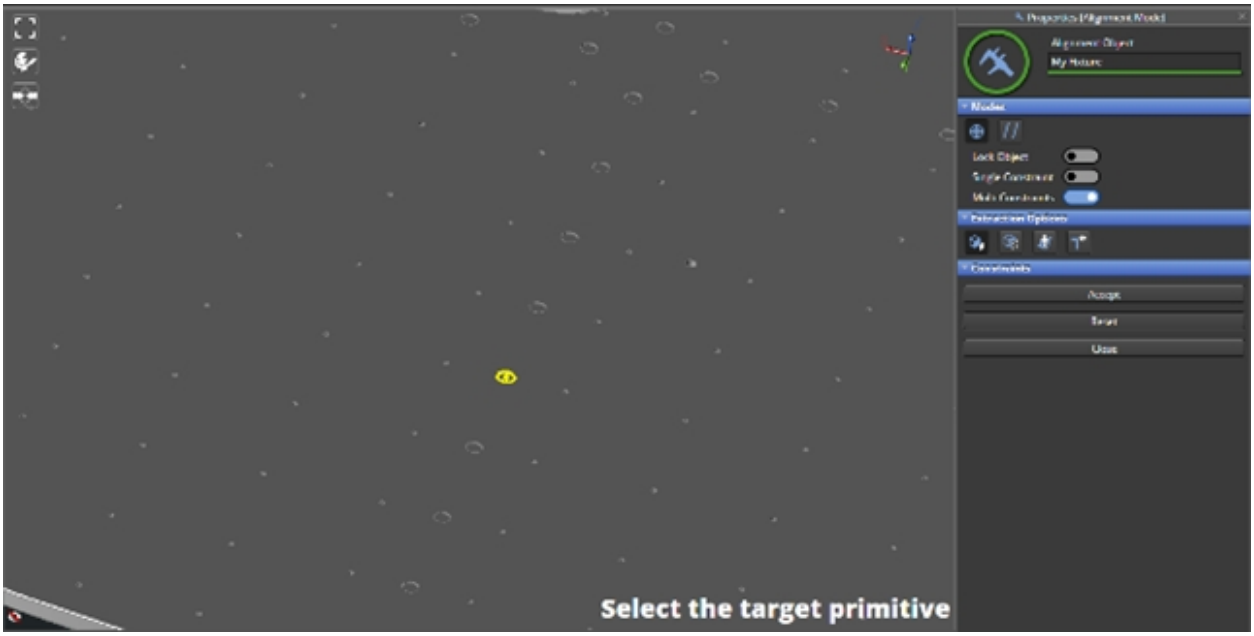





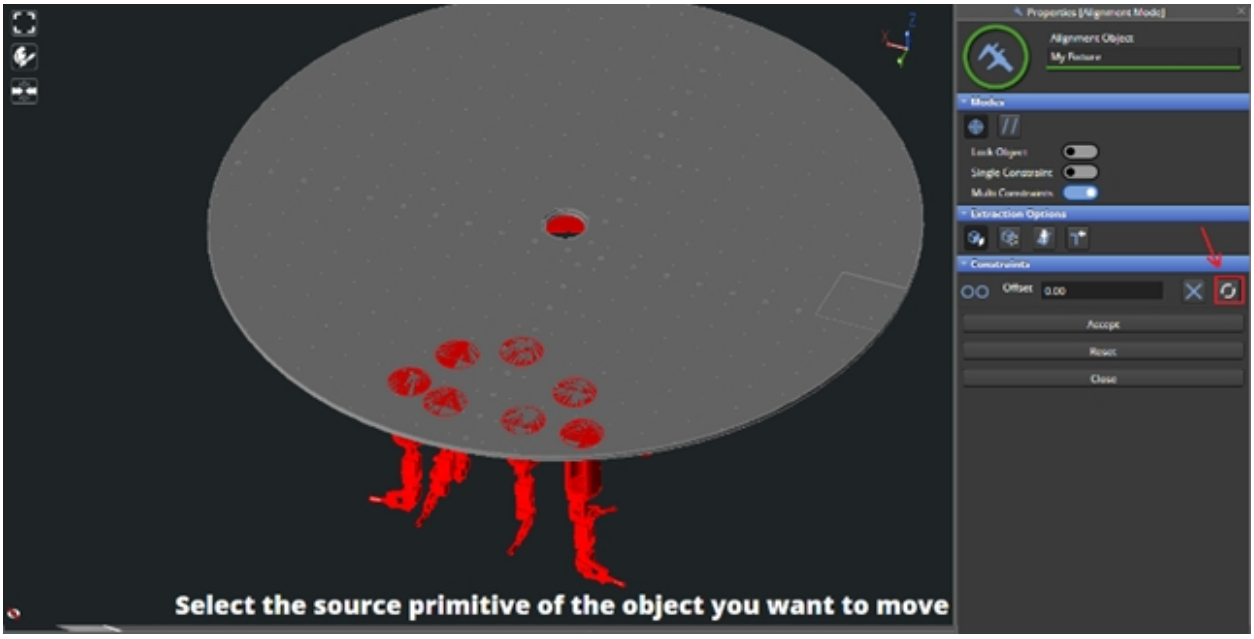



It is possible to go ahead by picking a feature on the element which will move (fixture) and then pick the corresponding element on the destination (Rotary Table), continuing picking elements on the fixture and on the turntable to set all the alignment constraints.



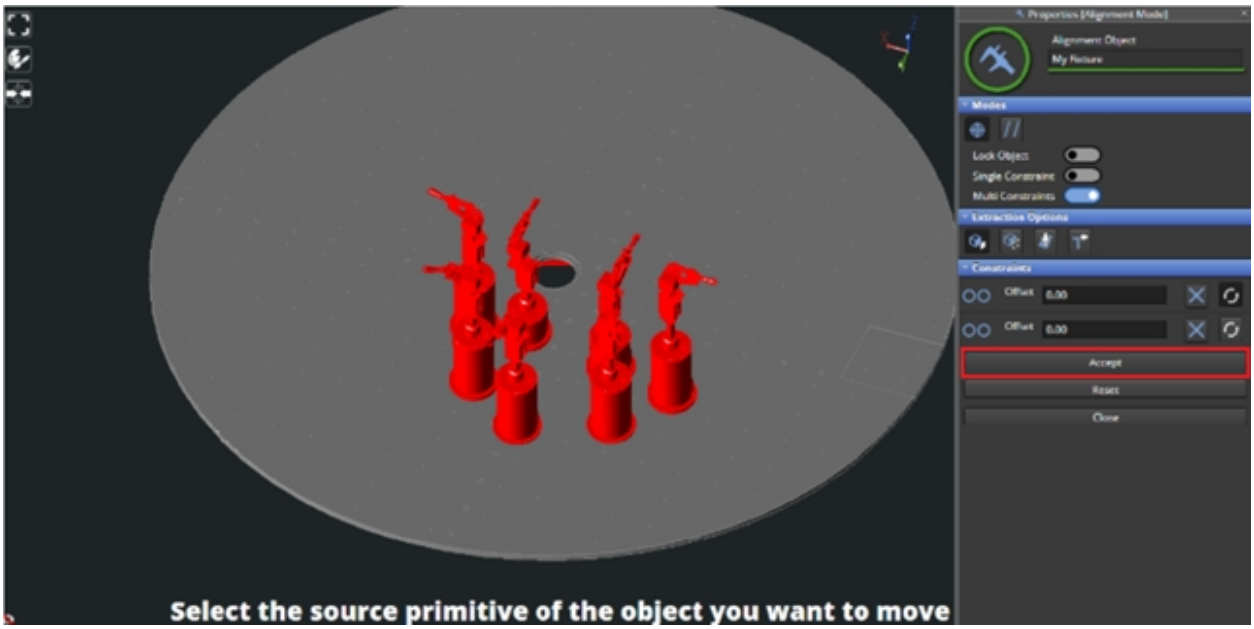



After clicking on the target primitive, the fixture will be aligned to the Rotary Table position. In case the fixture is aligned in a flipped position, you can flip it back using the **Reverse** button.




The user can continue adding other constraints and once the fixture is in the correct position, press on the **Accept** button, and close the alignment mode.

| | |
|---|---|
| | **Info**: In the following picture only two constraints are used but it is strongly recommended to use three. |




**Info**: In case you want to use actual geometries for the fixture alignment, you need to create them extracting from the CAD before you start the alignment process. In the alignment process, instead of selecting elements on the turntable as the destination, select the corresponding actual geometries.

The fixture is now in the correct position, but the part is not.

To align the part to the fixture, you can follow the same steps described above, selecting:

- The part elements instead of the fixture elements
- The fixture elements instead of the Rotary Table elements.

At the end of the alignment process the part will also be in its correct position so both the fixture and the part will be aligned to their real position in the cell.

---

At this point it is needed to lock the part and fixture positions as described in chapters Adding fixture to a cell

If you would like to place the part on the Rotary Table without a fixture, you will still need a fixture object as you cannot assign the part directly to the Rotary Table.

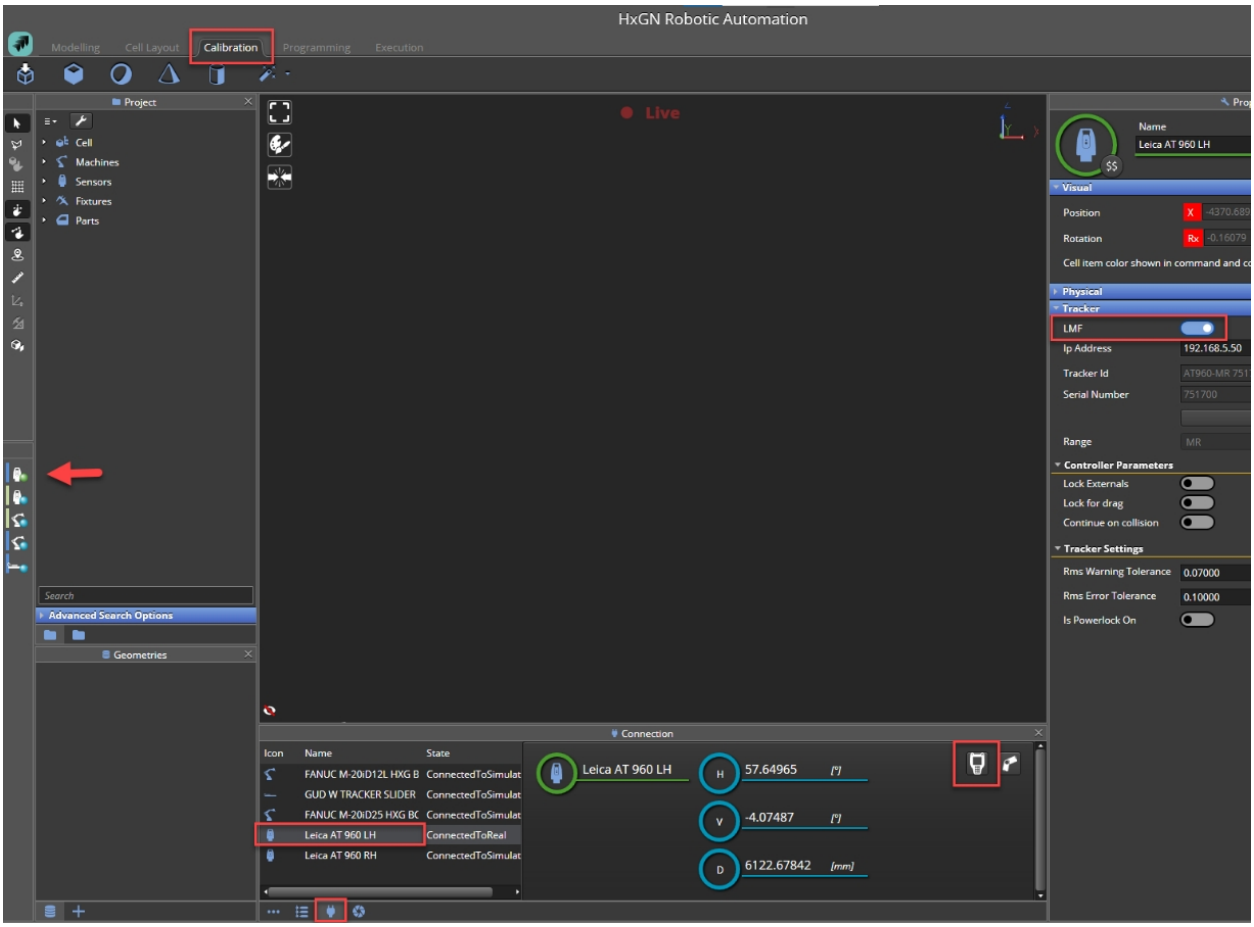
You can create a new fixture without assigning any CAD to it.

Next, drag the empty fixture onto the turntable, and the part onto the empty fixture.

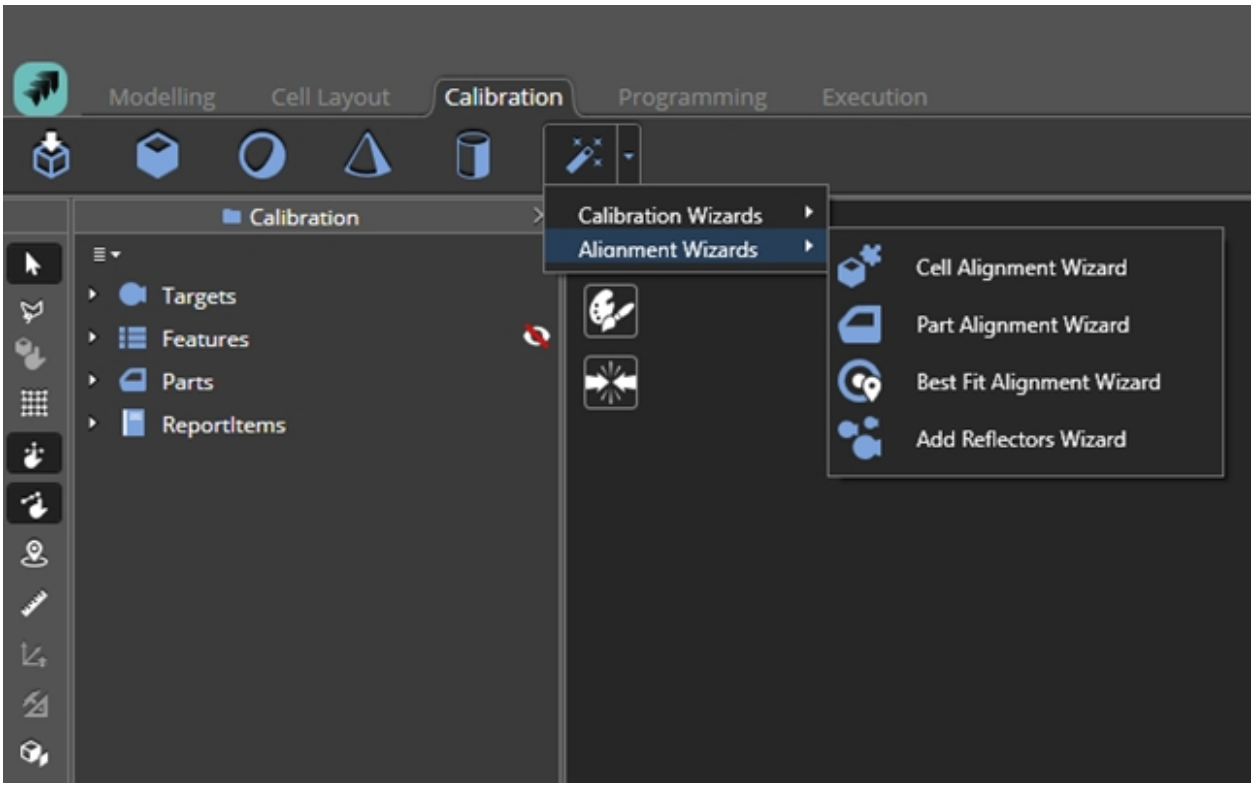
## Alignment wizard workflow

Alignment is done through wizards under the Alignment tab.

Click on Alignment mode tab and then click on sensors in project tree and open the tracker properties. On tracker section enable the toggle button to connect with tracker through LMF. The LED will switch from blue to green which means the tracker is connected and ready to be used.




Once tracker is connected with LMF, open the wizard panel on upper right in Alignment tab. There is a wizard for each type of alignment.




## Cell alignment wizard

The cell alignment wizard allows the user to align and calibrate items in the virtual cell to make it match the real cell, starting from the tracker to every cell item and create reference targets for best fit alignment.

| | |
|---|---|
| | This wizard is designed for single tracker cell configurations.<br>For dual robot configurations, please follow the steps described in chapter Dual_tracker_cell_alignment_workflow |

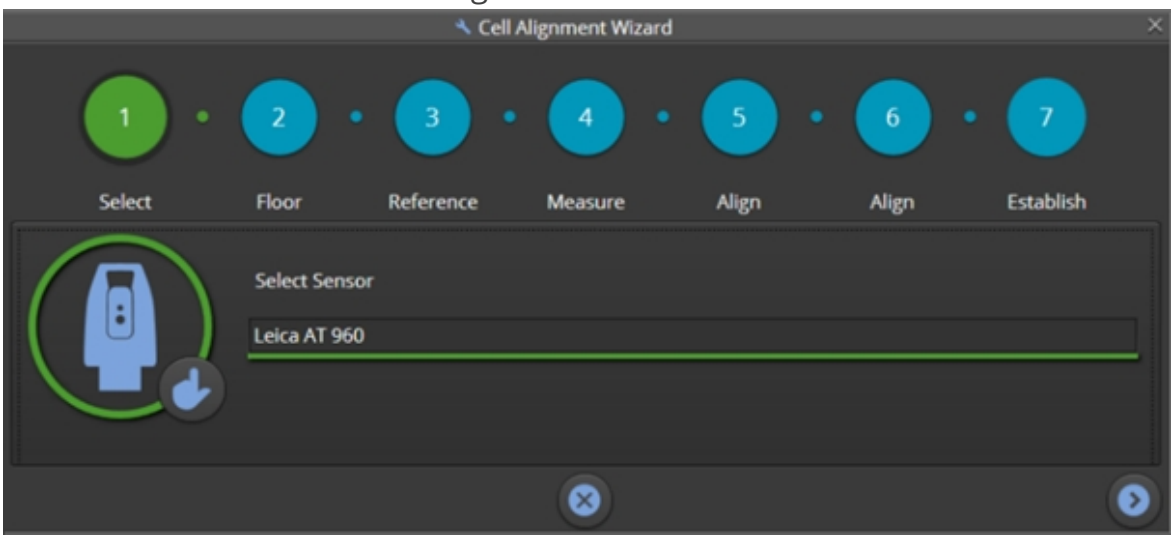
| | |
|---|---|
| | In case the tracker is mounted on a slider, unassign the tracker from its slider before starting the alignment process and reassign it at the end of the alignment process. |

This wizard currently forces the user to go through the following flow:

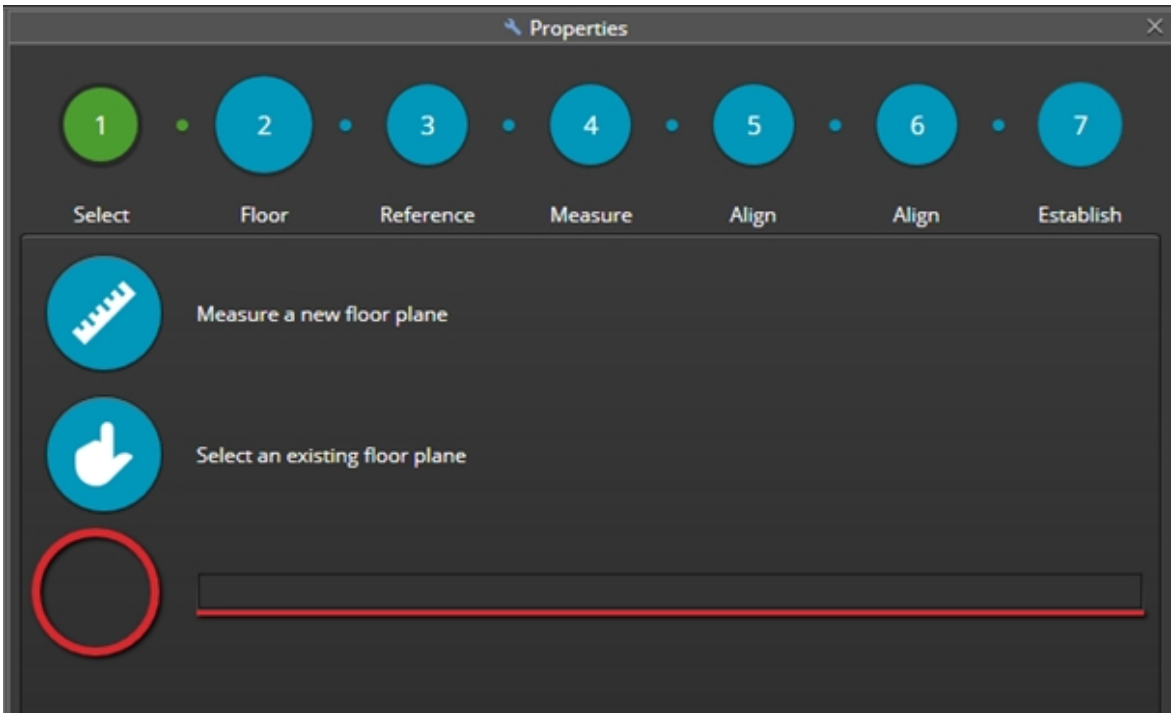
**Floor → Reference → Measure → Align → Establish**

Following the wizard steps:

1. **Select**: select sensor used to align the cell

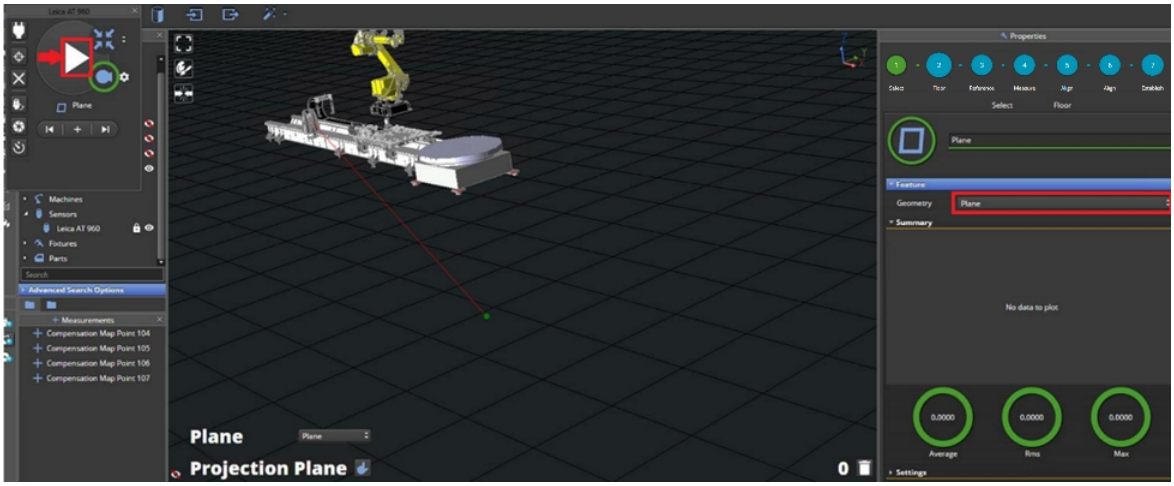



2. **Floor alignment**: The purpose is to align the Tracker reference frame with the floor reference frame.
    a. **Measure a new floor plane** to open a teach pendant to measure a plane feature.
    b. **Select an existing floor plane** to choose a previously feature being the floor plane.




In the following we will continue selecting **Measure a new floor plane**

Locate the Reflector on the floor and lock the tracker on it.



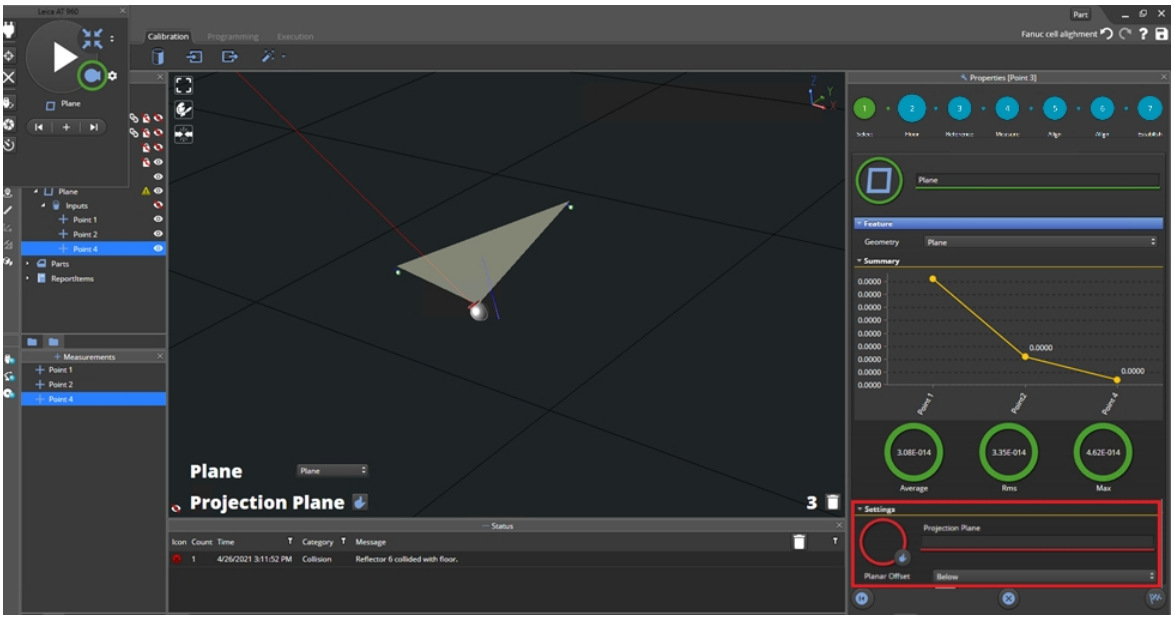

.

The Teach pendant will open, select plane geometry on properties, click on the Play icon ( ) and measure at least three points to define a plane

Once your click, you will see under *Features* | *Plane* | *Inputs* the point that was created (Point1)..

Move the Reflector to another position on the floor and click to measure again. Repeat again at least three points until you see the plane in the 3D Window.

If you are using a tool to place the Reflectors on (e.g., a nest) and it is below the reflector, select **Below** from the drop-down menu in *Properties* | *Settings* | *Projection Plane* within properties.




Floor and plane have different heights. After clicking the Finish icon ( ) the plane height will be adjusted automatically to the floor grid, and the tracker position value will be adjusted to the actual height in the real cell.

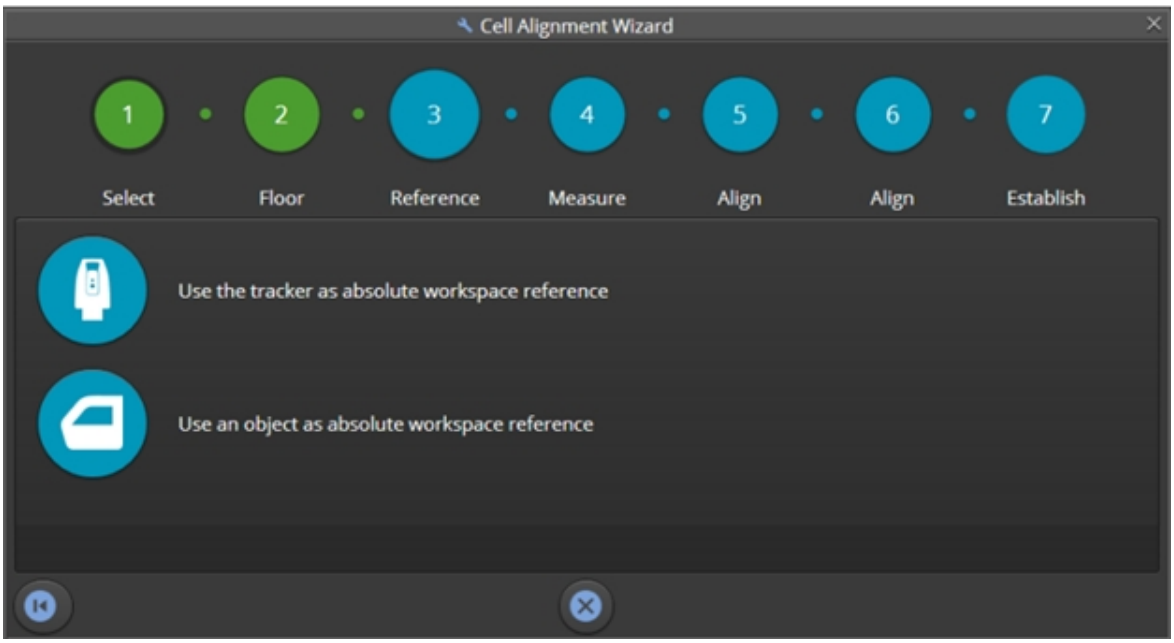
It is recommended to record and write down that position value for future reference.

3. **Reference**: select the reference frame:
    a. **Use the tracker as absolute workspace reference**: cell alignment will use tracker as reference for aligning the cell.

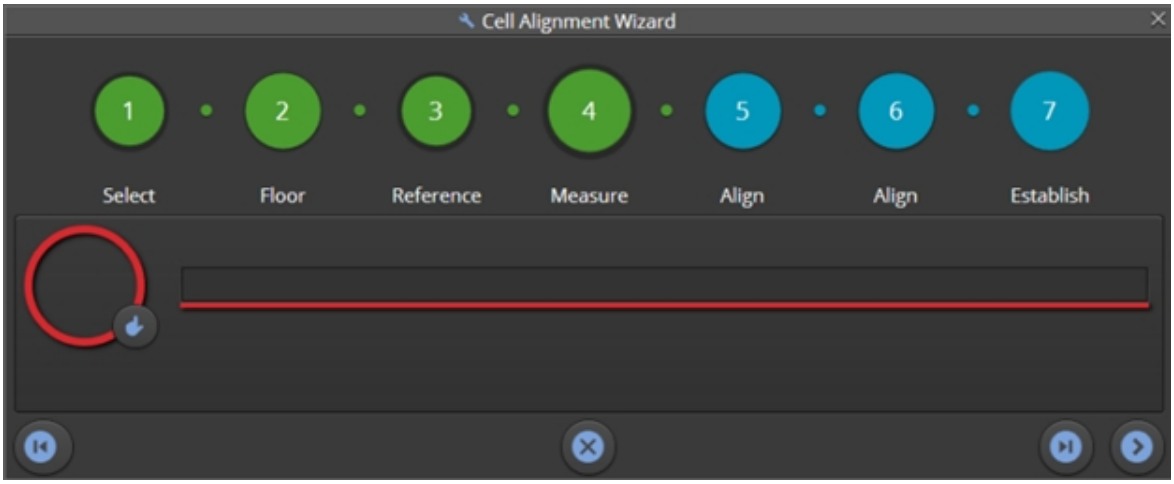
If tracker is going to be the reference object, it is a good practice to set position on (0,0,0).

    b. **Use an object as absolute workspace reference**: cell alignment will use an object as reference for aligning the cell.

In case the Tracker is used as absolute workspace reference, the user can continue selecting the first option. If an object is used as absolute workspace reference, the user can select the second option. See chapter Rotary_Table_used_as_absolute_workspace_reference to see how a Rotary Table is used as an absolute workspace reference.

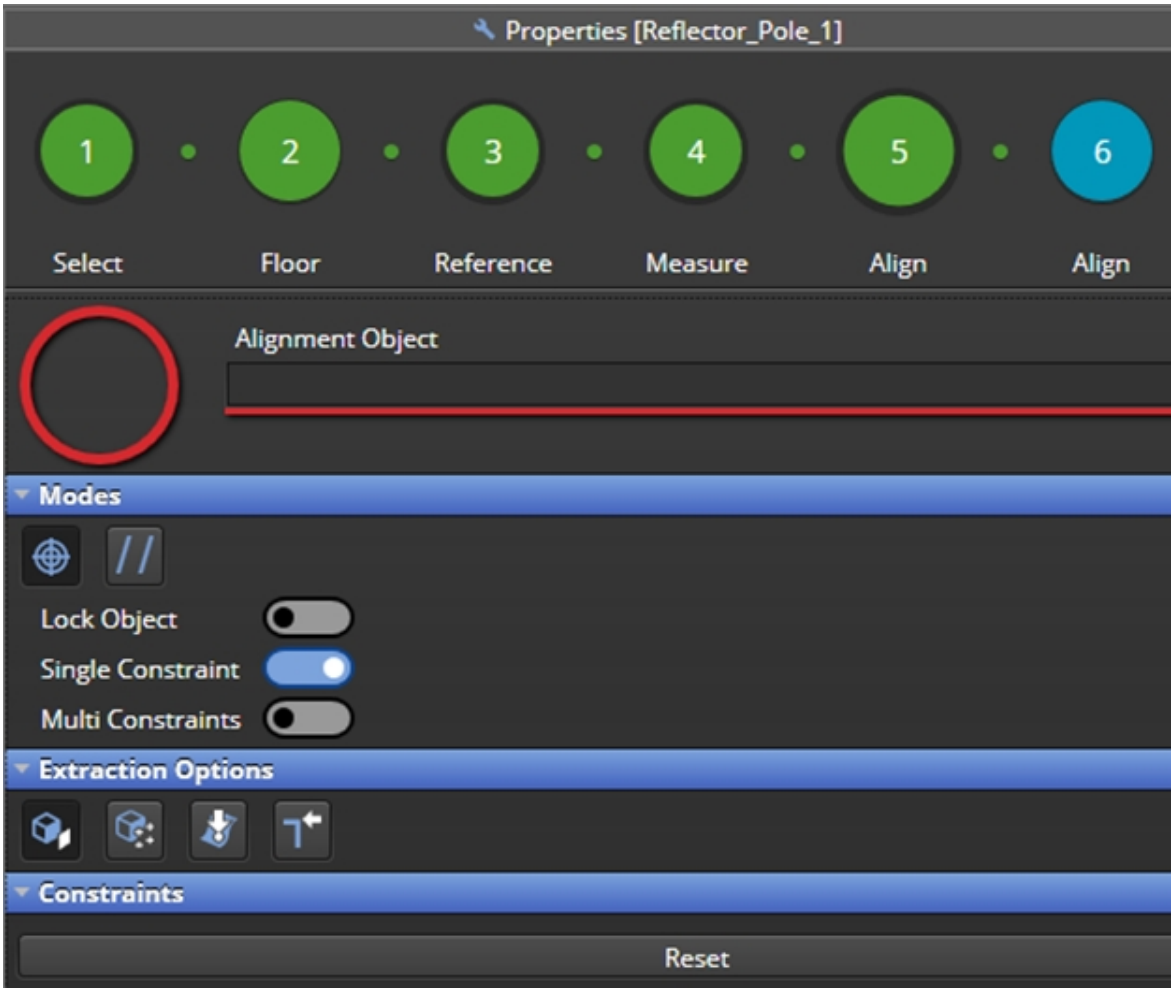



**Measure**: the tracker teach pendant will open to measure all the features considered by the user to align the cell items.

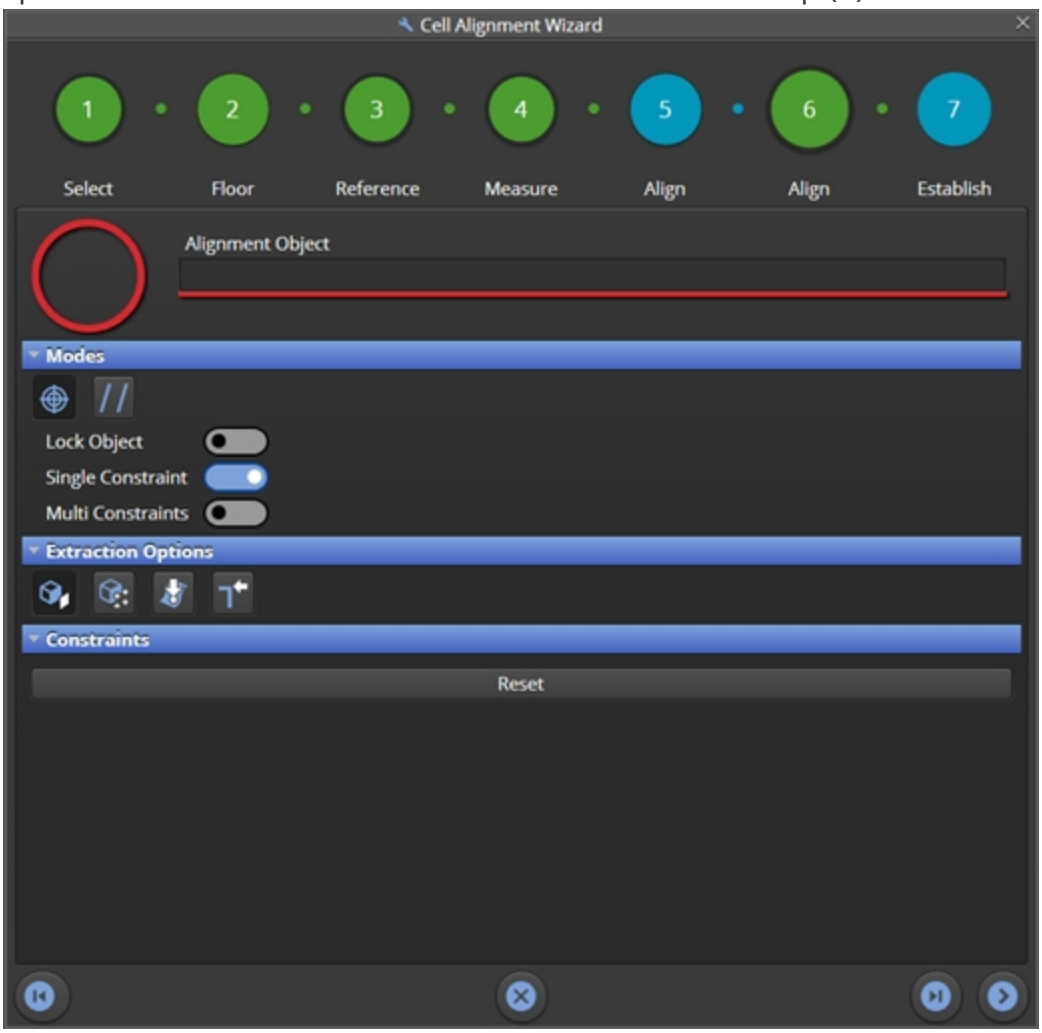
4. 


5. **Align**: this step is skipped in case the tracker is used and absolute workspace reference so there is no need to align the tracker to the object. If an object is used as absolute workspace reference, in this step the tracker will be aligned

to the selected object, compensating for alignment error.

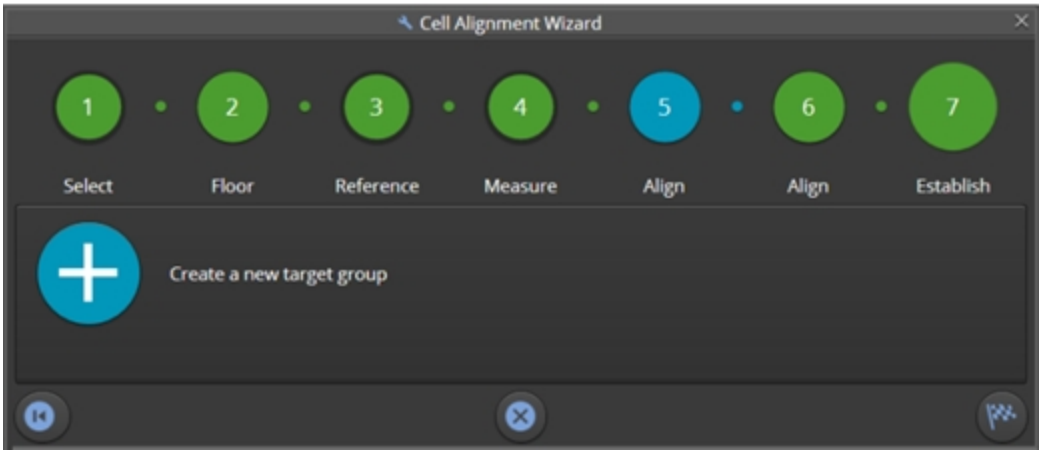


**Align**: in this step, it possible to align all items in the cell to the absolute work-space reference that was defined in the Reference step (3).

6. 

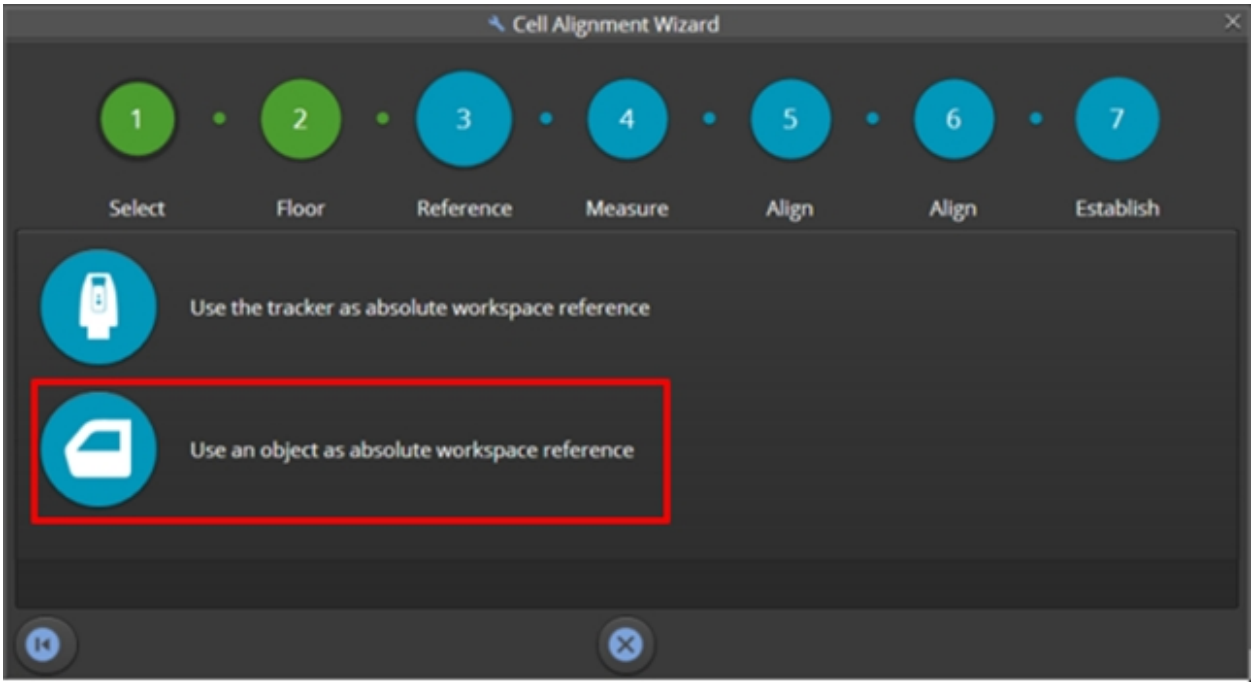

7. **Establish**: in this step it is possible to define a target group of reflectors (at least three are required, see chapter Add_reflectors_wizard) to be used in Best Fit alignment wizard to re-align the tracker in case of movement from the pre-

vious location (e.g. in case of maintenance or replacement).



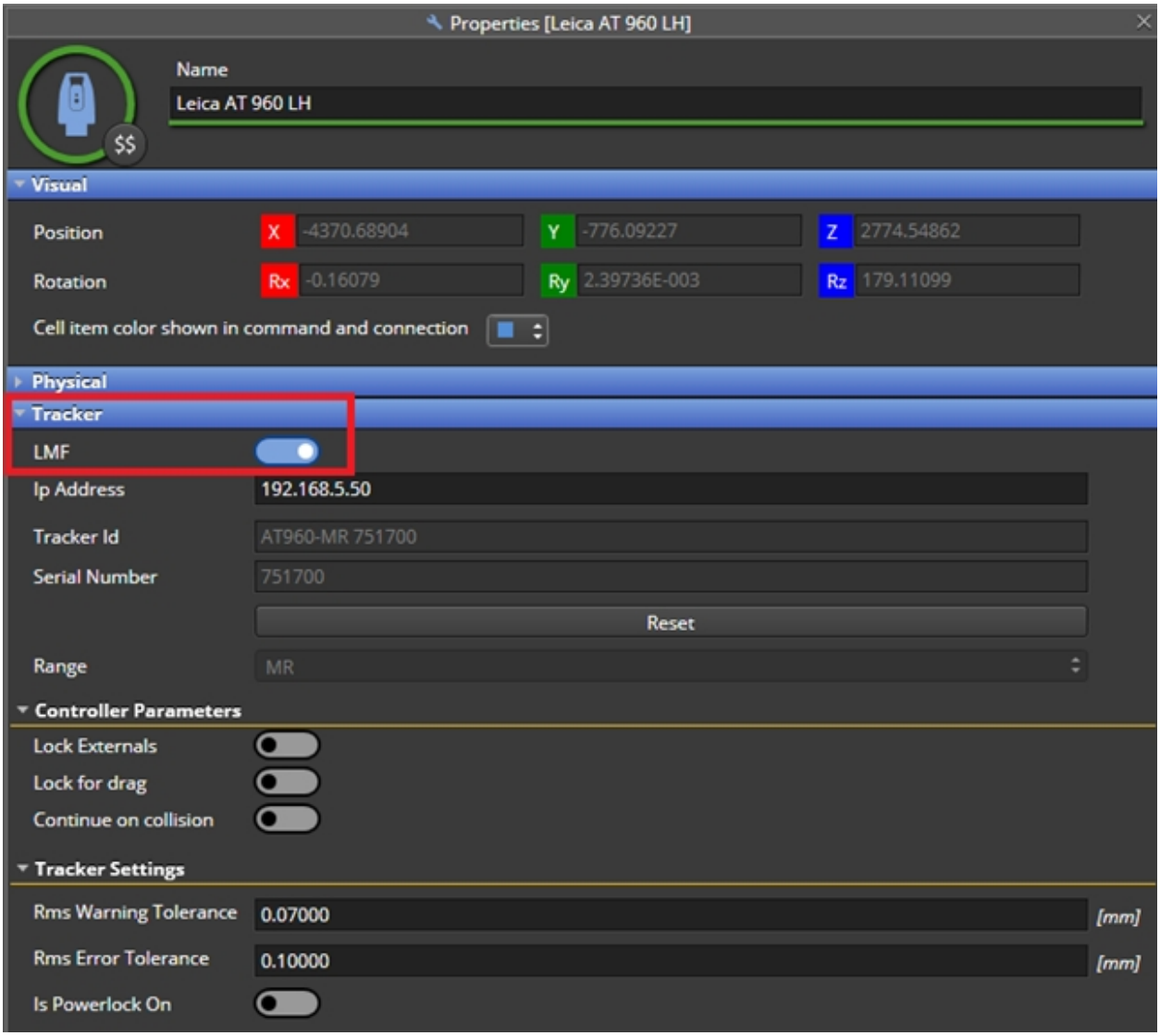

**Rotary Table used as absolute workspace reference**

In this specific case, the tracker is aligned to the Rotary Table which is used as reference object.



Place at least three reflectors on the top plane of the actual Rotary Table or you may use the same reflector and move it to create a plane on the top.

Connect to the Tracker by enabling the LMF toggle switch and measure the three points on the top of the Rotary Table.




Once finished measuring the Rotary Table the wizard will ask for aligning the sensor to the reference object, then user will align the plane measured to the Rotary Table

object in the project. Tracker position will be updated to match the Rotary Table coordinates.

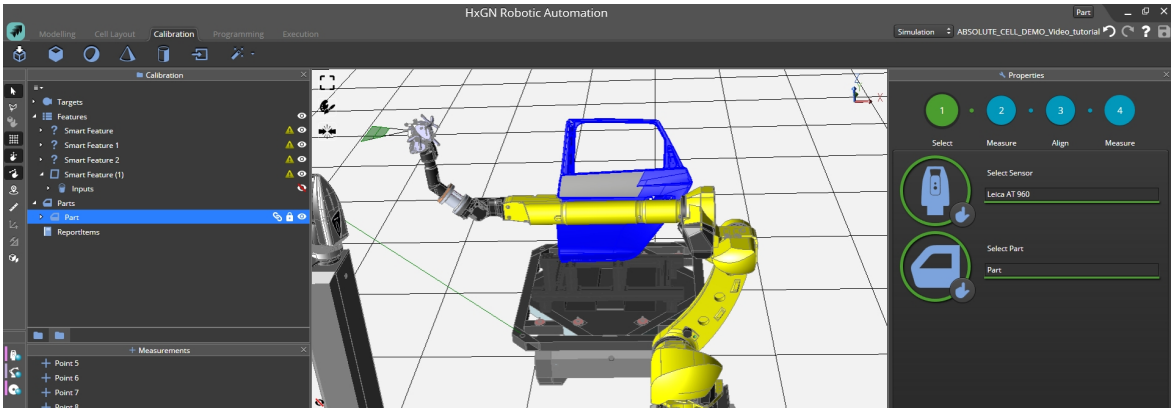
User can continue to follow the remaining steps described in the cell alignment wizard to complete the alignment using the tracker to measure the rest of the cell items which will be now in the same Rotary Table absolute workspace reference.

### Part alignment wizard

This wizard is used to align a part to its real cell position forcing the user through the following flow:

**Select** → **Measure** → **Align** → **Measure**

1. **Select** the part from 3D view or the project tree and follow the steps.




2. **Measure** the feature positions needed to align the part, see similar step in chapter Cell_alignment_wizard
3. **Align** the CAD to the measured positions, see also similar step chapter Cell_alignment_wizard

The fourth step can be skipped, and user can click **Finish** to complete the wizard.

## Best fit alignment wizard

If the tracker is mounted on a slider (see chapter Tracker on slider), the user must first unassign the tracker from the slider, align it, and then assign it again once aligned.

The goal is to align a new Laser Tracker in an existing project for the following three use cases:

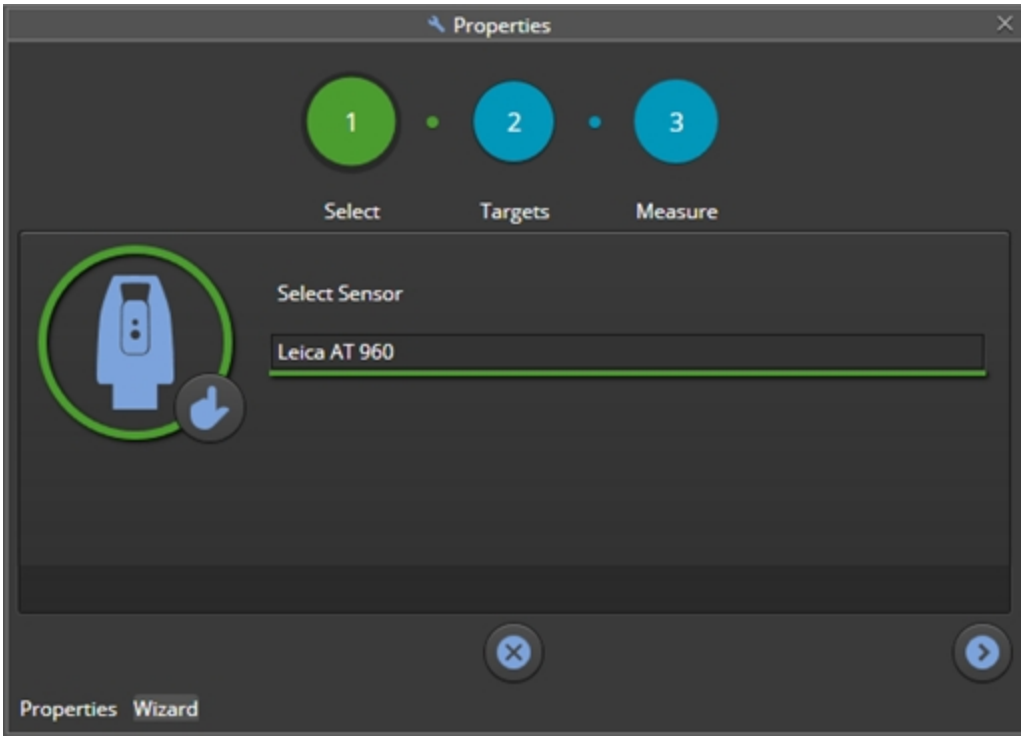
1. A new tracker is **added**.
2. The existing Tracker is **replaced** with a new one and the existing programs must be updated.
3. The existing Tracker is **moved** to a new position and the existing programs must be updated.

This alignment is usually applied after cell alignment was performed and measurement programs were generated. If for any reason it is necessary to move the Tracker to a new position, the goal of Best fit Alignment is to perform the alignment of the Tracker in the new position, using some reference points measured by the Tracker in the original position.

A preliminary requirement is to have a group at least 3 Targets (reflectors) measured with the Tracker in the original position.

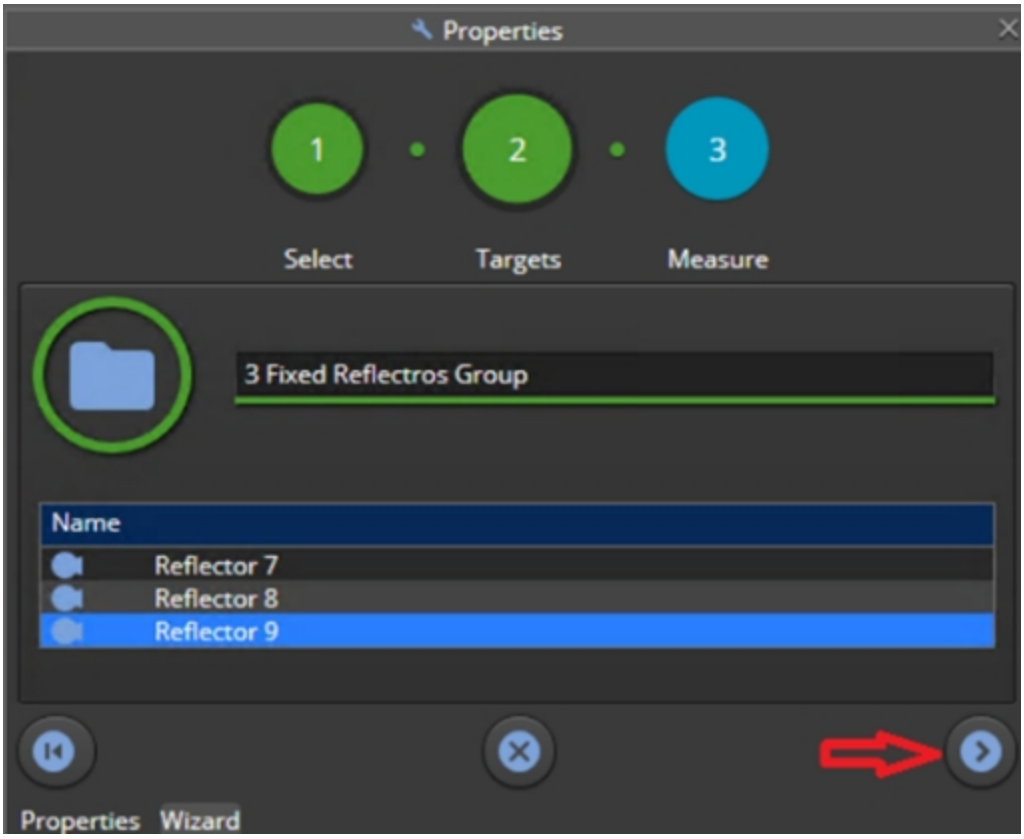
At this point is possible to follow the Best Fit Alignment wizard steps:

Select the Tracker (e.g., Leica AT960) and click Next.

1. 

2. Select the existing target group in the project tree and the reflectors will appear into the Wizard.

Click **Next** once the existing target group is selected and shown.

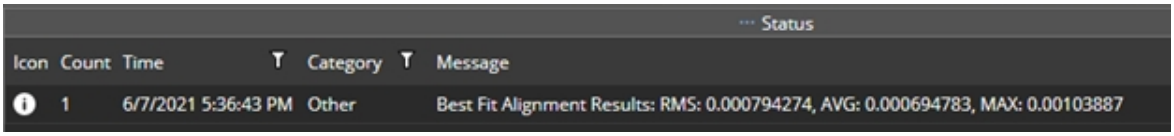



3. Click on **Auto Measure** and the system will automatically measure the targets with the new tracker. In case the automatic measure is not successful, it is also possible to measure manually the reflectors using the tracker teach pendant as explained in chapter Add_reflectors_wizard

Click **Finish** once the summary is displayed.

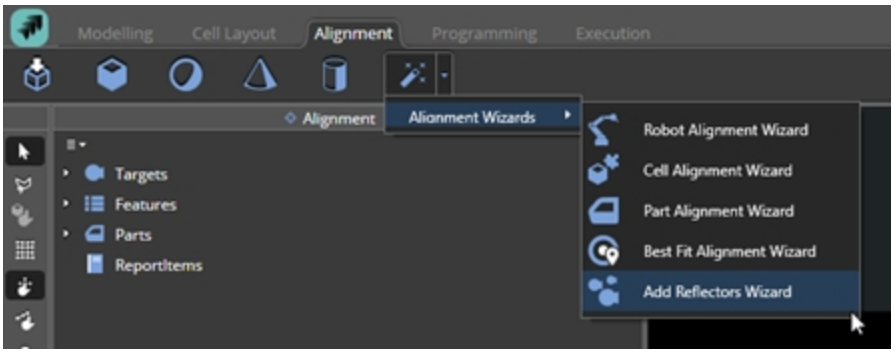
4.


You can also see the result in the **status window.**

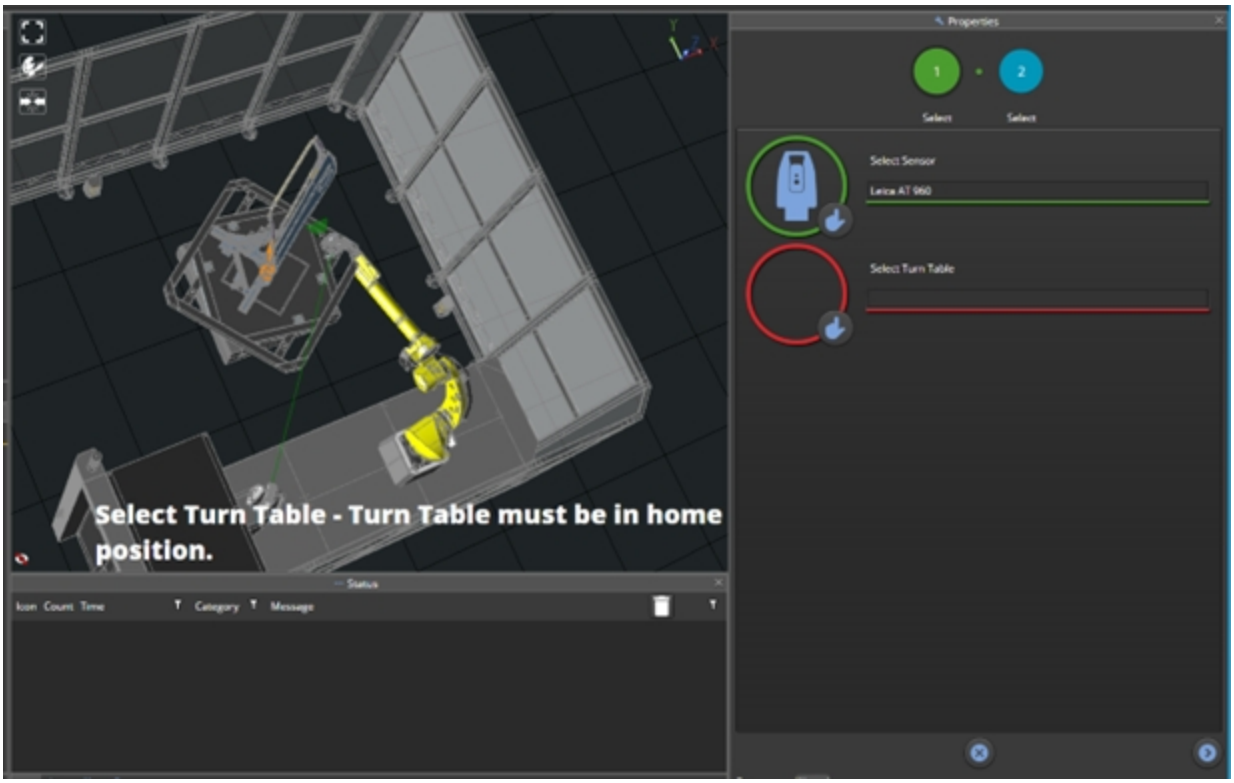
5. 

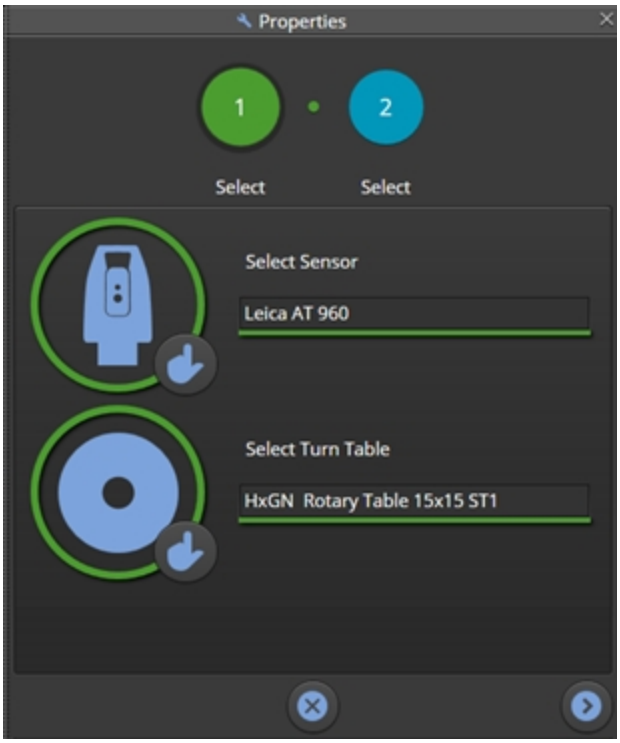

## Add reflectors wizard

Under Alignment Tab, there is a wizard to help user insert Reflectors:

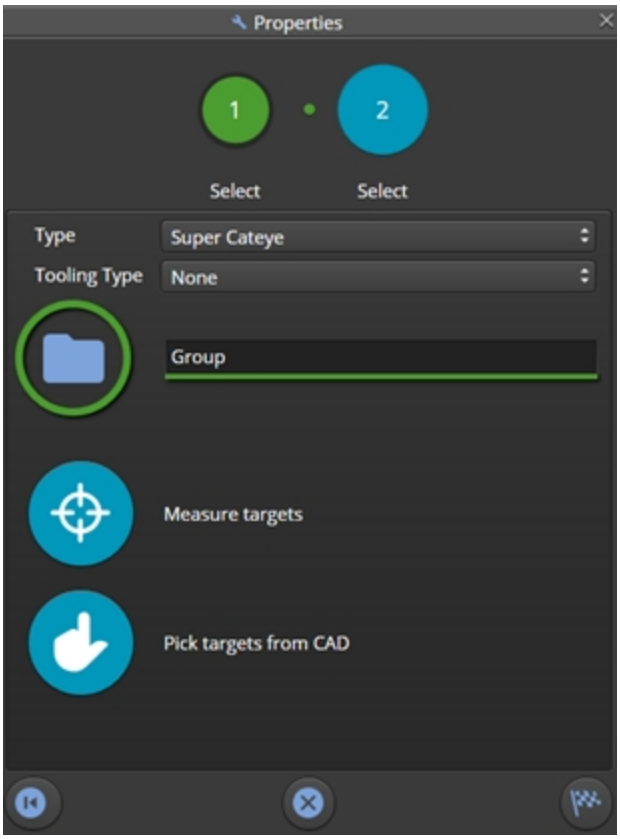



Once the wizard properties open on the right, you may select Rotary Table if reflectors are on the Rotary Table (make sure the Rotary Table is in the home position): the reflectors will move together with the Rotary Table or skip and click **Next** (button right): the reflectors will be static.



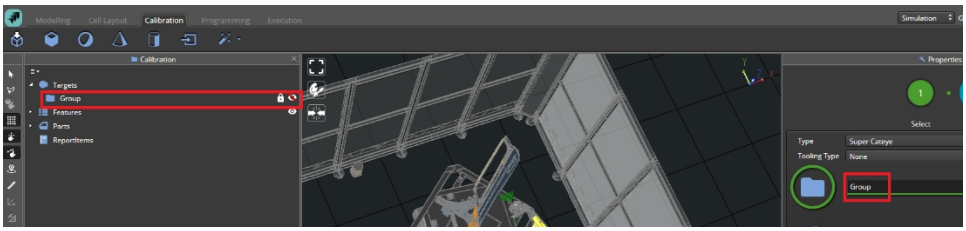



Next step is to select the reflector type, its tooling type, and it is also possible to rename the group as preferred in case multiple groups are present in the project:

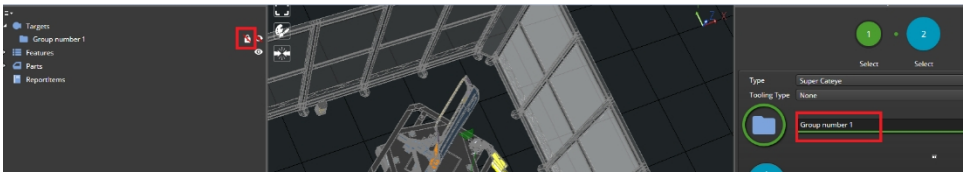



Select the reflector type from the list of supported reflector types in the drop-down list:

Select the Tooling Type and define a name for the Group:

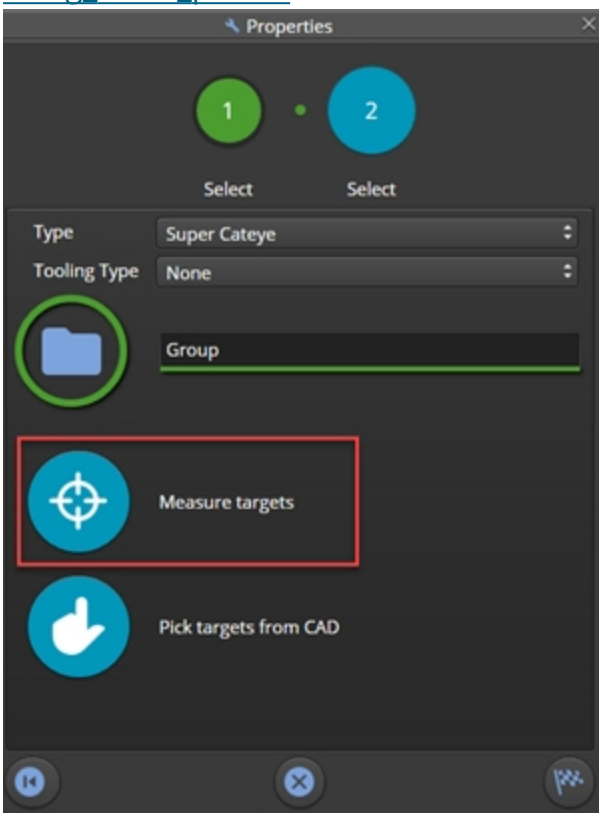


To change group name, it is needed to unlock the target group name from the project tree:

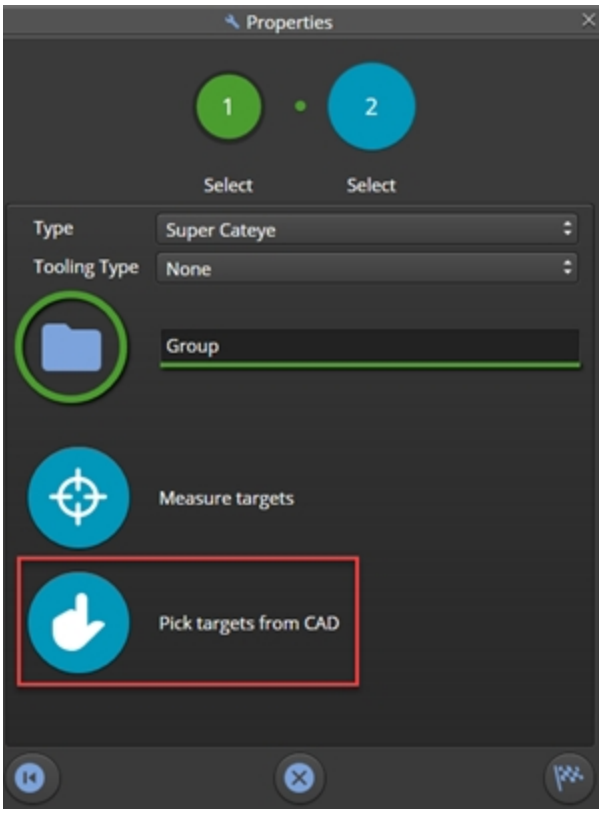


It is now possible to choose either to:

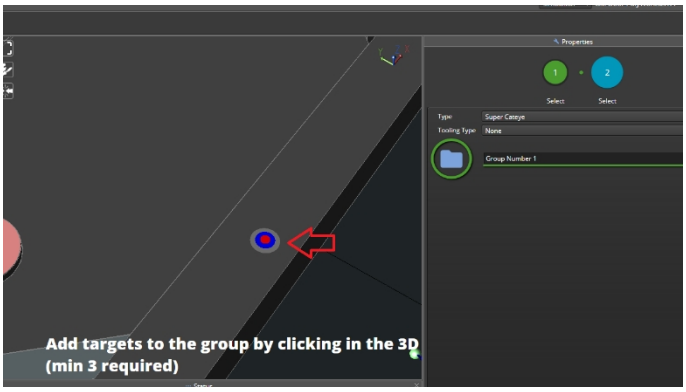
**Measure targets**: for measuring existing targets in the real cell by using the tracker teach pendant as already shown in chapter [Rotary_Table_alignment_using_three_planes]

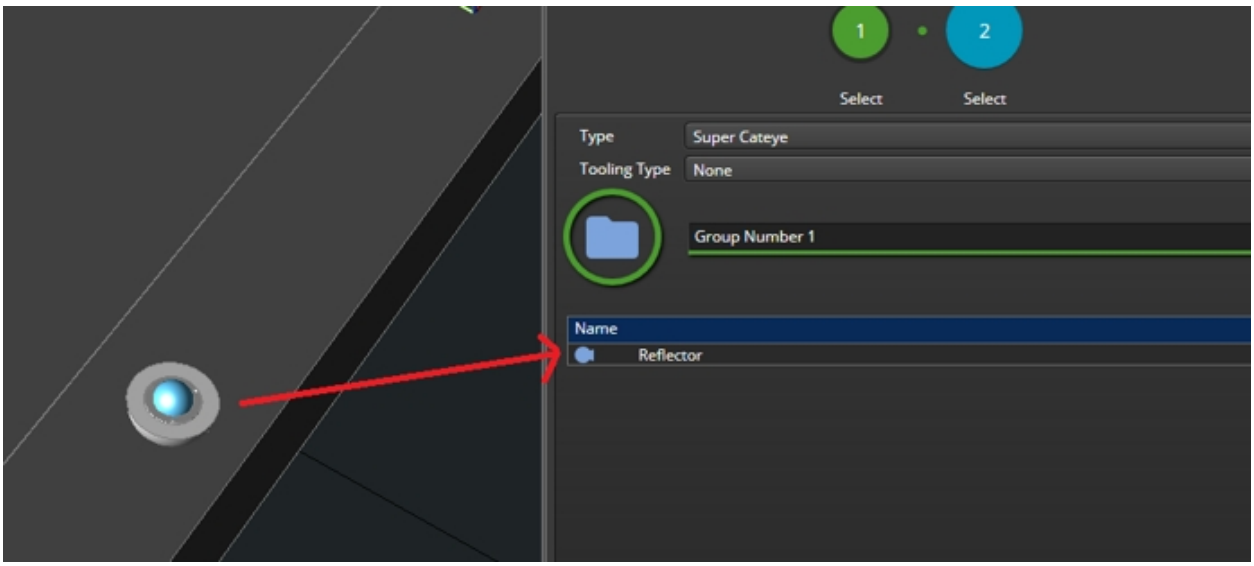
1. 

**Pick targets from CAD**: for picking the position of reflectors from the 3D View (used for offline mode).

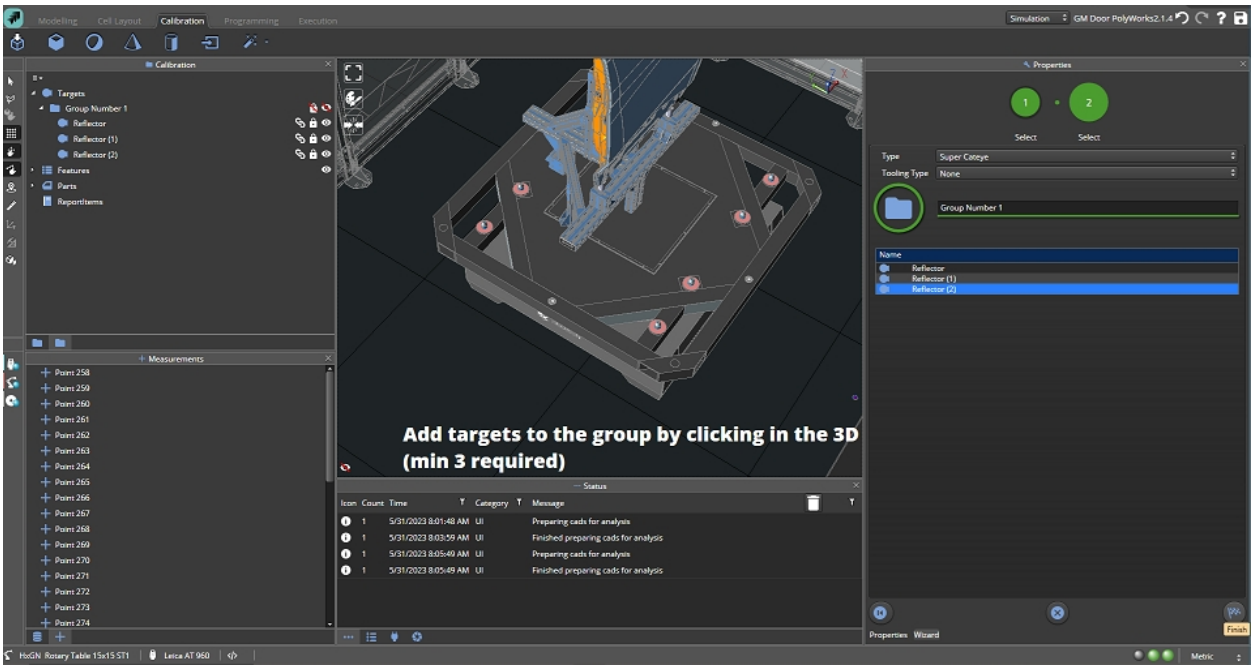
2. 


By clicking with the mouse, the reflector will be added within the group:




Continue with at least three reflectors. Once all reflectors are added, click **Finish** to complete the wizard.




## Dual tracker cell alignment workflow

In cells where more than one tracker is available to measure, it is necessary as a first step to align them each other.

In this way it does not matter which tracker is used to measure the features as all of them will be referred to the same coordinate system.

The following steps must be followed:

1. In case the trackers are mounted on a slider, unassign both trackers from their sliders before starting the alignment process.
2. Measure all features with the main tracker
3. Execute the cell alignment wizard following all the steps (see chapter Cell_alignment_wizard)
4. Align the secondary tracker using the best fit alignment wizard (see chapter Best_fit_aligment_wizard)

5. Measure with the secondary tracker all the features that could not be measured by the main tracker.
6. Use the alignment mode to align the new features measure in the earlier step.
7. In case the trackers are mounted on a slider, they need to be linked again to the respective slider.
8. Be sure that all the features are locked in the project tree to avoid any unwanted movements.

## Robot on slider and tracker on slider workflow

In this cell configuration the challenge is align the tracker slider. In general, if tracker is already mounted on the slider is going to be hard to measure the necessary features to align properly the cad model.

There are two possibilities, it depends if the user can use a second tracker to help in the alignment or dismount the tracker on slider to measure the necessary items from other tracker position.

As a first option, the user can follow the next workflow explained below:

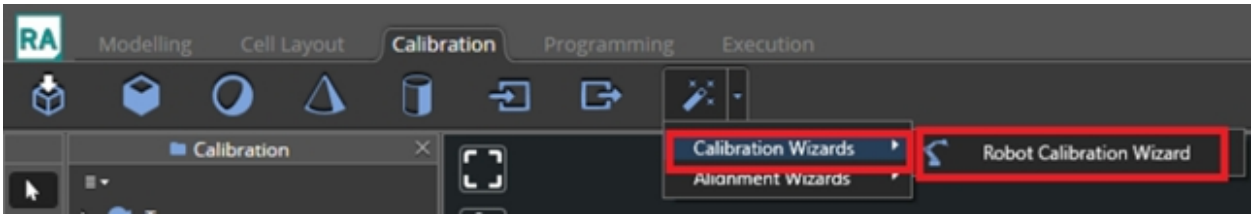
1. With tracker on slider generate a target group to use it later in the tracker alignment process
2. Move the tracker to a new position in the cell (use a Leica tripod) and align it using the best fit alignment wizard.
3. Measure all the necessary items on the tracker slider that are missing when tracker is on it.
4. Mount again the tracker on the slider and use alignment mode to align all the items.
5. Link the tracker to the slider.
6. Lock all the features after making sure the alignment was done properly.

In case there is no possibilities to dismount or measure with a secondary tracker, the alignment process can be done manually following the next steps:

1. Measure and align the base of the tracker slider.
2. Manually adjust the tower in the horizontal and vertical axis to be aligned with the tracker.
3. Link tracker to the slider
4. Lock the features.

## Robot alignment

The Robot alignment process must be done to adjust the robot theoretical model to the real robot used in the cell and to align the robot base with the tool TCP.




A basic alignment is done using the Robot Alignment wizard for the following:

- Robot base – 6 DOF (degree of Freedom) position. In case the Robot in mounted on a slider, the slider 6DOF position must be considered.
- Tool TCP (T-Scan or AS1 laser scanners) – 6DOF position. With a basic alignment, it is expected to get good accuracy results of RMS below 2 mm.

| | |
|---|---|
| | This robot alignment is not focused on advanced alignment of robot parameters (e.g., motor encoder calibration) that must be executed according to manufacturers guidelines. |

### Prequisites:

1. Leica Metrology Foundation (LMF) and Tracker installed on the system (AT 960-MR)
2. LMF is the only tool that can be used for robot alignment (not possible using T-Collect or RDS)
3. Cell model imported to the online HxGN Robotic Automation application.
4. Cell.json file changed to have a workstation configured for robot alignment purposes (meaning that the tracker must not be included within the work-station as it must be controlled by LMF and not by HxGN Robotic Automation) with unique ID (e.g., 100). This workstation ID must be used for the robot align-ment process as only the correct cell elements will be used.
5. Imported Cell is already aligned (all elements) as described in earlier chapters.
6. Only a single alignment per robot is needed. In case a dual robot cell is used, every robot must be aligned separately.

The alignment is done in the context of the workspace you intend to use, i.e., the area where the part is placed to be measured. E.g., the workspace can be the area above the Rotary Table when it is used as a part holder where to place the part and the relative fixture.

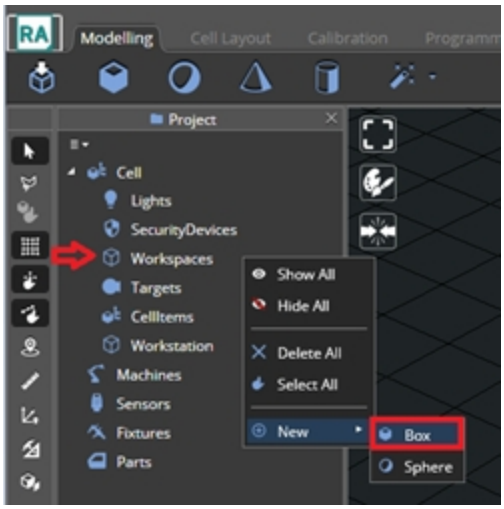
In a cell where the robot is used in two distinct workspaces, e.g., two Rotary Tables, it is recommended to do two alignments in two different projects.

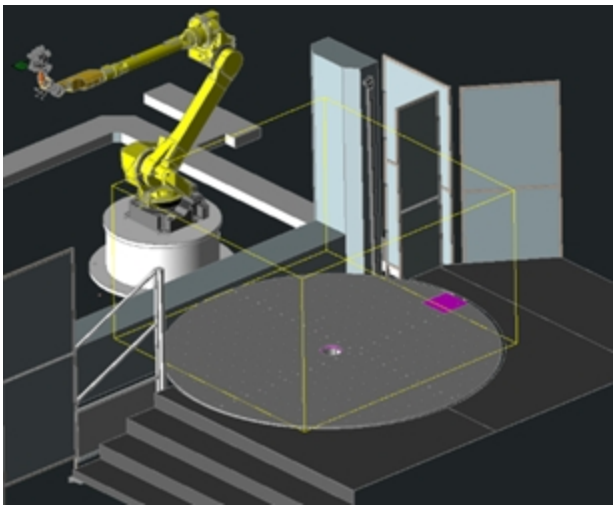
To create a workspace, in Modelling mode, select a new **Box** workspace and adjust the workspace size and position to reflect your working area. This is the area which your largest part to be measured will occupy (yellow wireframe box). Make sure to define workspace before you start the wizard.

| | |
|---|---|
| | **Pro Tip**: users need to consider that: |

- the minimum distance between tracker and scanner is 1.5 m
- the solution provided by the path planning algorithm will always try to place the scanner parallel to the workspace face and with a face pointed towards the tracker for it to lock on that face of the scanner.

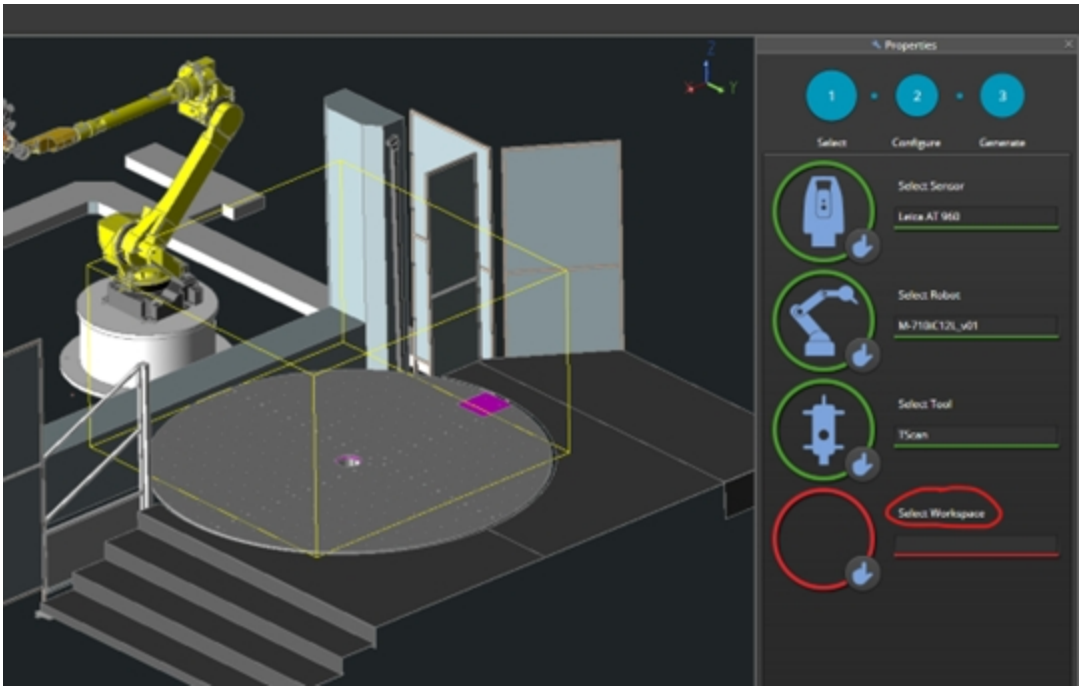
If the conditions above are not considered, it may be that the face on which the tracker needs to lock will not be visible and the algorithm to fail.



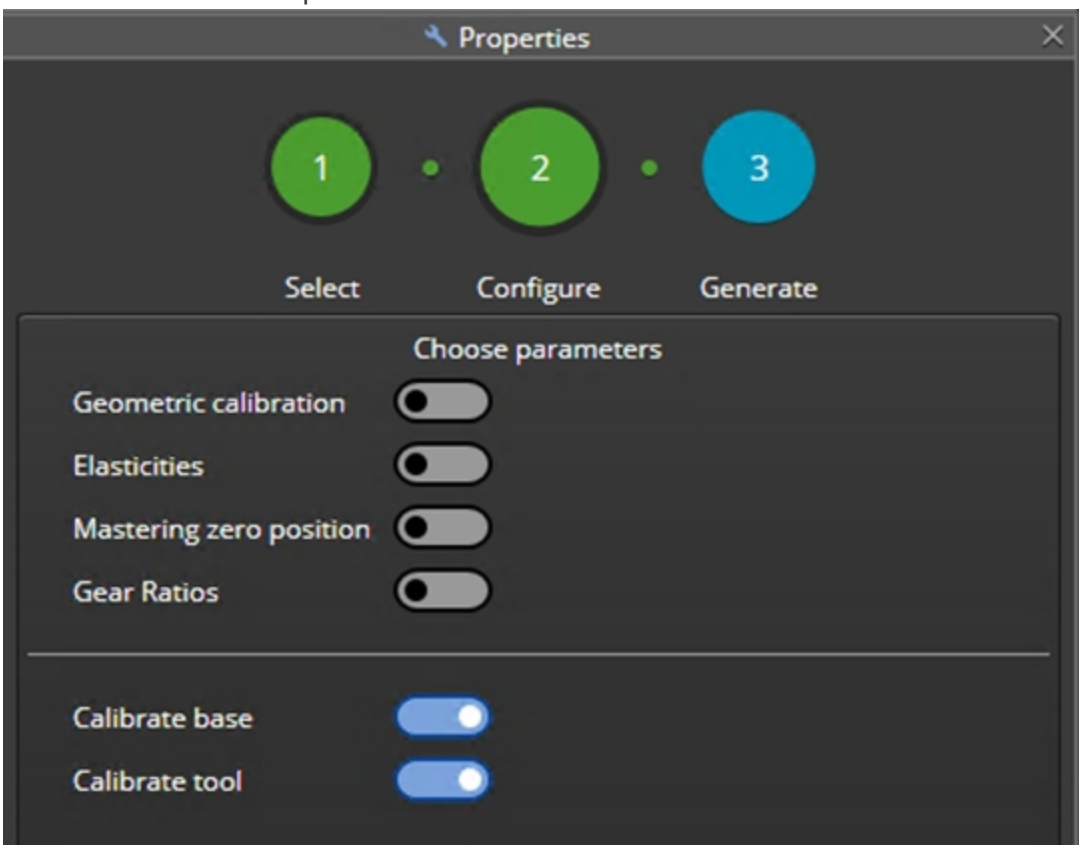
The **wizard** can be executed, and the steps followed:

Select your workspace and press **Next**.

1. 

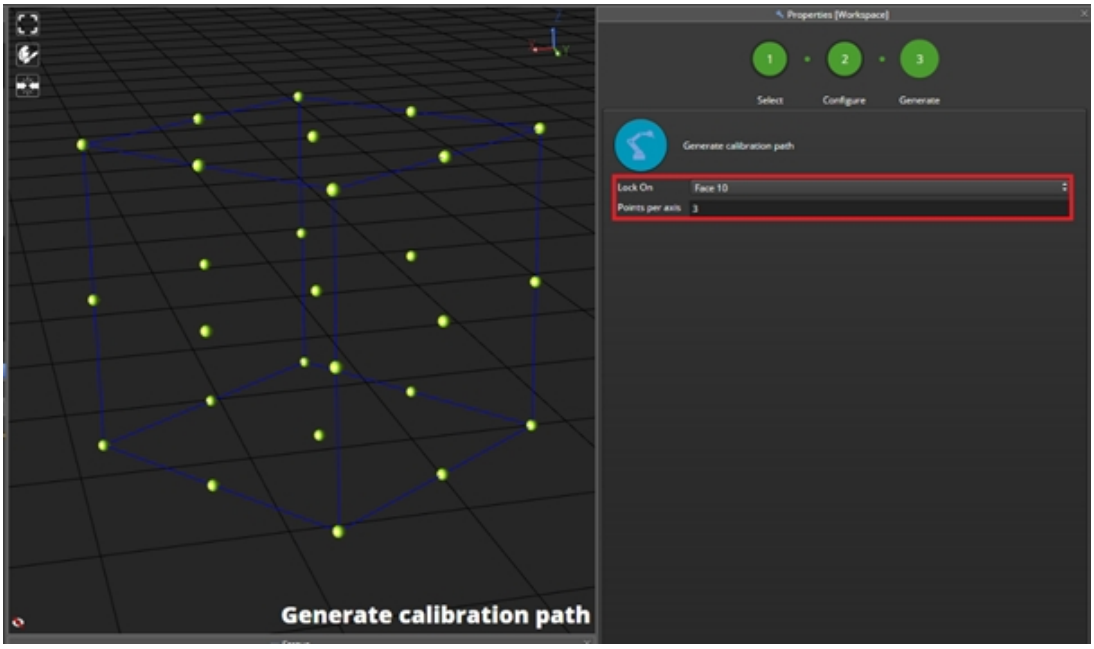

| | |
|---|---|
| Alignment | 320 |

Set the parameters as default values shown in the following picture and continue to the next step.

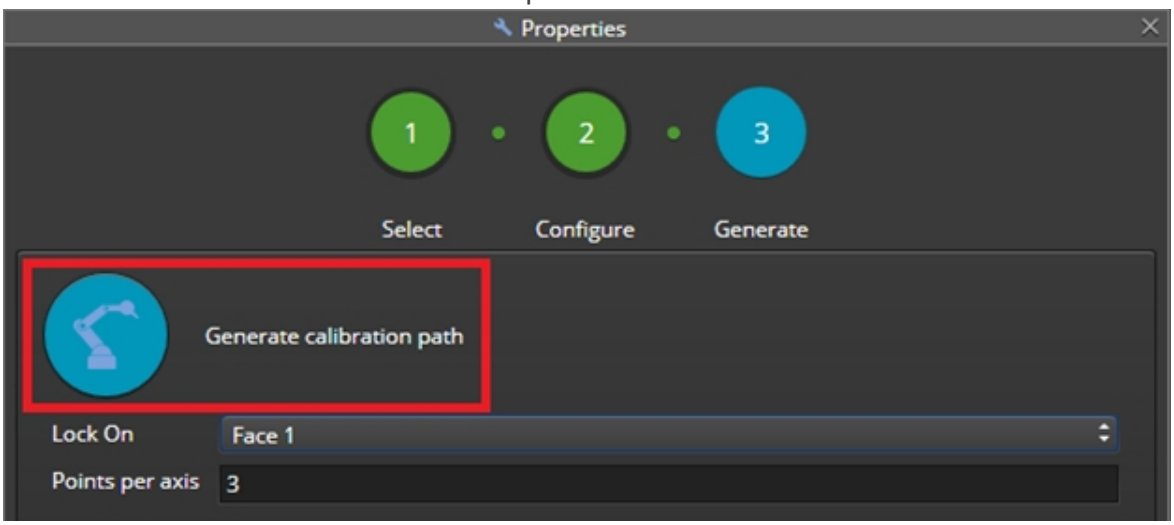
2. 


3. In this step, we will generate the robot alignment program by setting two parameters:
    a. **Lock On**: the scanner face to be used to lock on it. It is recommended to use first one side face of the scanner for the most stable results. It is usually suggested to use the left face of the scanner (Face 1)
    b. **Points per axis**: the number of points which will be used for alignment. This number decides the discretization of the workspace. It is suggested to use the default value (3). It will create a grid of 3x3x3 (total of twenty-seven points per face). This is good to have redundancy as many points

will not be reachable.




Click on "**Generate calibration path**" and the program will be automatically created. Click on Finish icon to complete the wizard.

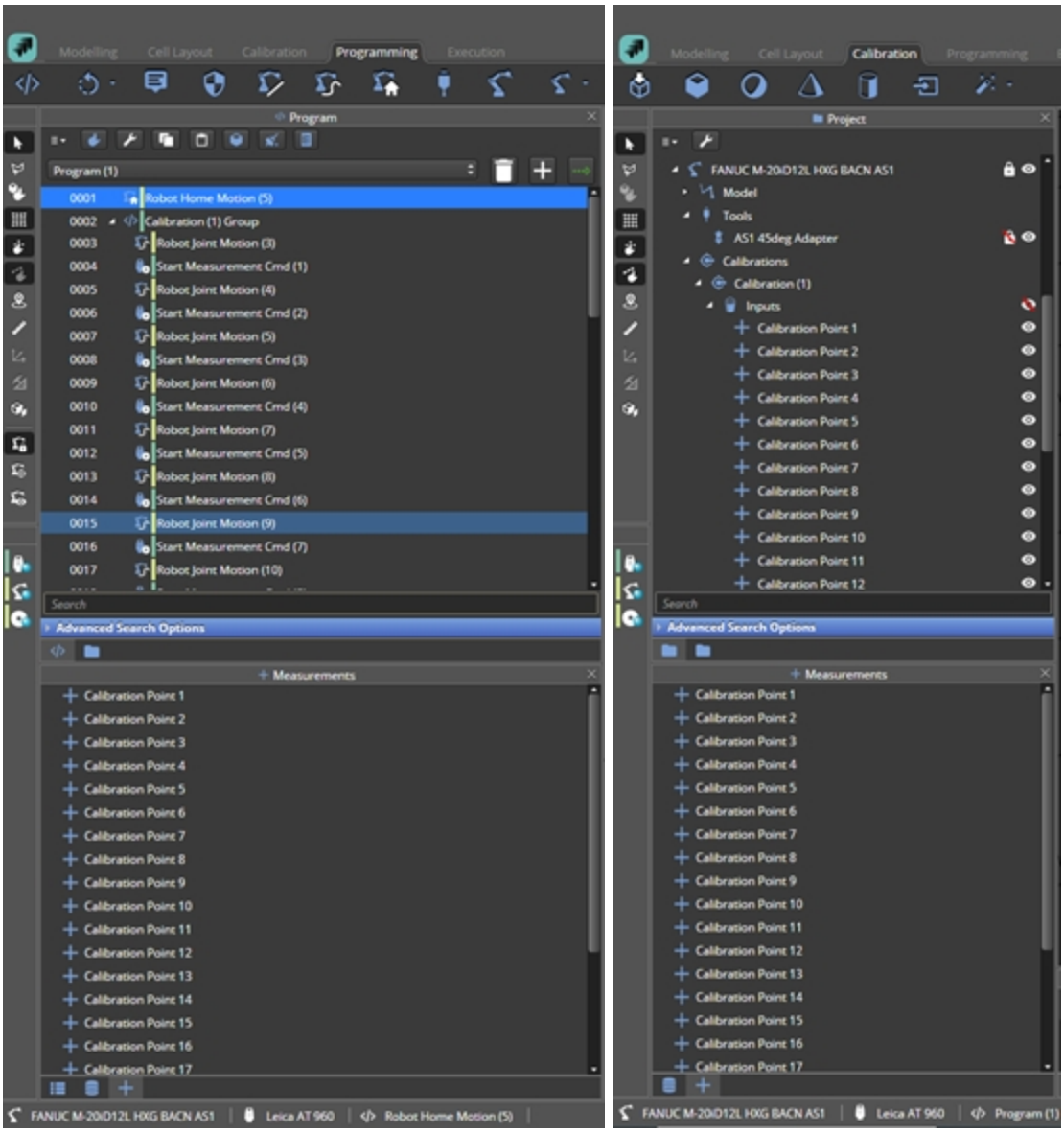
4. 


The alignment wizard generated a program for the face we chose (Face 1).

5. User now needs to validate the program and execute it in the real cell. After the execution is completed, the calibration points will be available in the project tree under **Alignments** inside the robot that was used in the alignment wizard.

| | |
|---|---|
| | To validate the program and run it an online environment within HxGN Robotic Automation Execution mode, it is strongly suggested to use slow speed. |

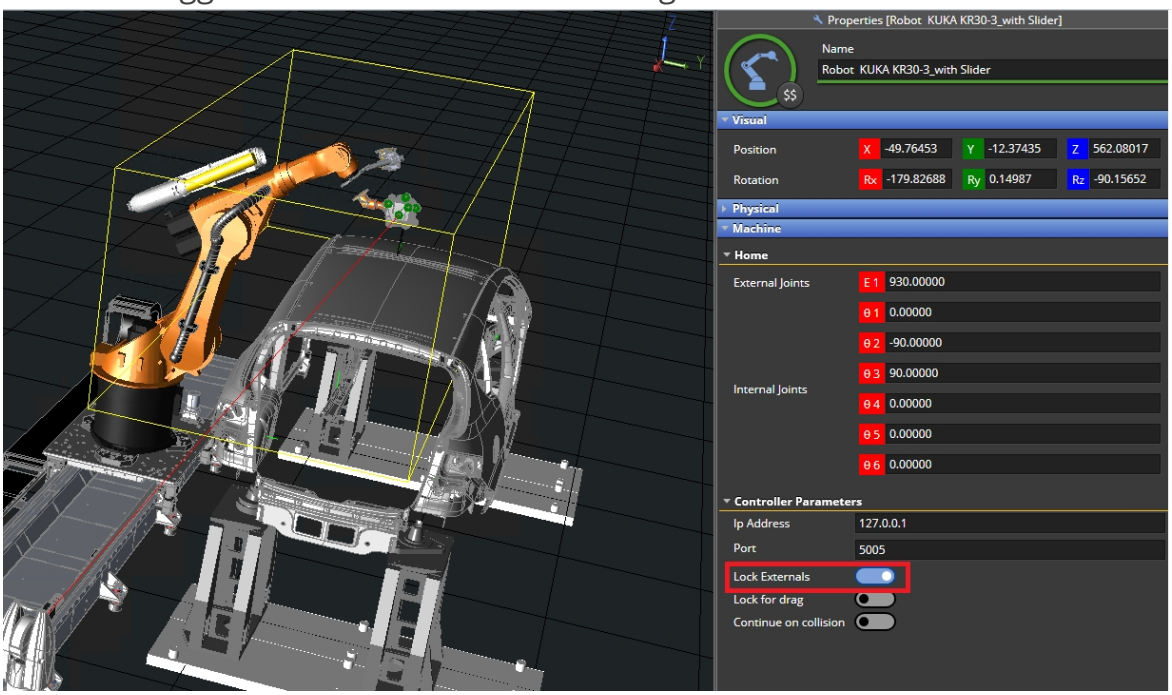
| | |
|---|---|
| | Make sure there are no Calibration Points in the project before executing the program in online mode otherwise the results may be wrong. |



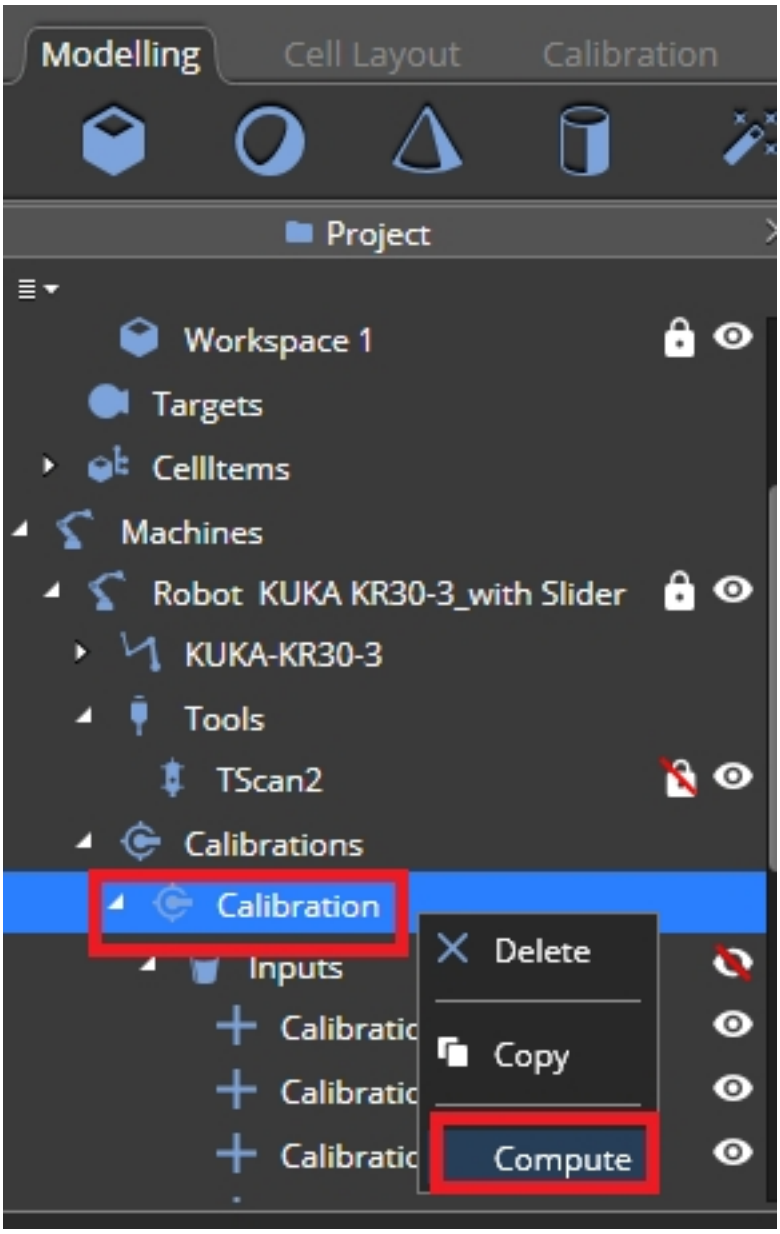

It is needed to measure at least ten points in the workspace. It is also needed to get a variety of robot orientations, configurations, and joint values. In case of a slider is used, one slider position is to be used making sure to create a workspace box big enough to cover the part to be measured.

---

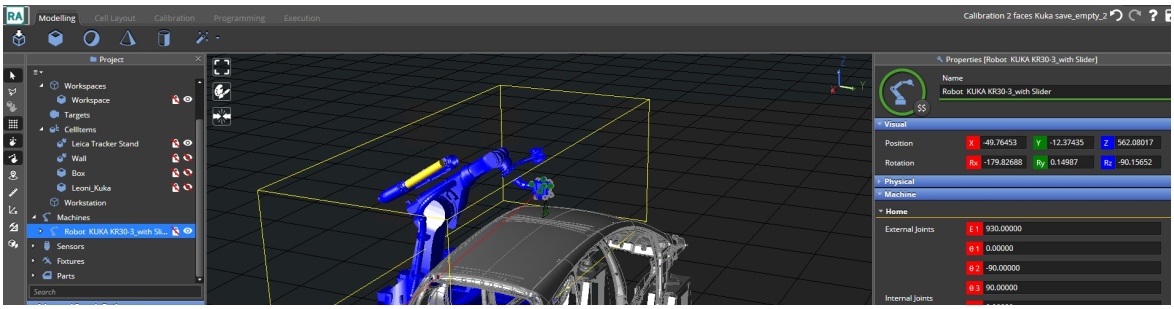
Robot alignment supports both 6 and 7 axes configurations. If the robot is moving on a slider (7 axes configuration) but when it reaches the workspace, the slider is not used to perform the scanning, there is no need to do the robot alignment considering that axis. For this use case, move the slider to the position where you want to perform the alignment on the workspace, enable Lock Externals toggle switch and use the robot alignment wizard.

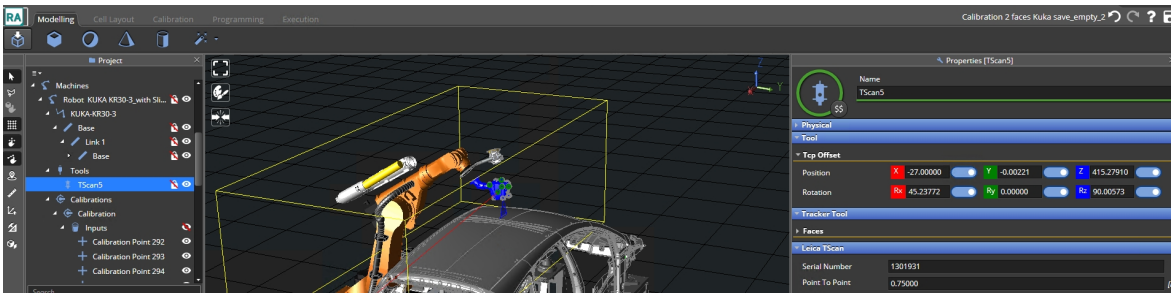



6. To execute the calibration, right click on *Calibration* | *Compute.*

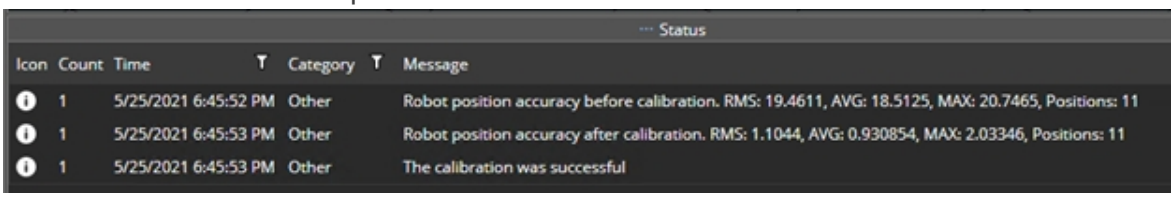



The alignment will update the position of the robot and the Tool.

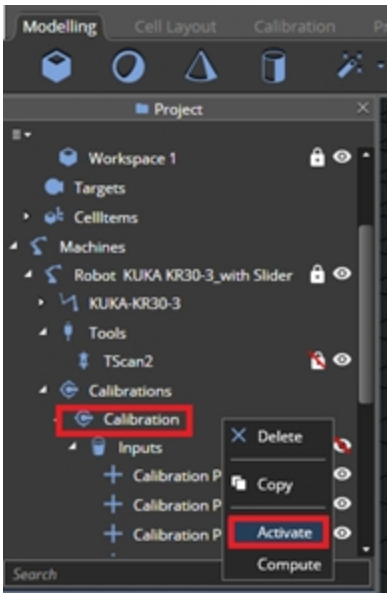




The computation results will be displayed in the Status window. The alignment can be considered acceptable if the RMS is under 2 mm.

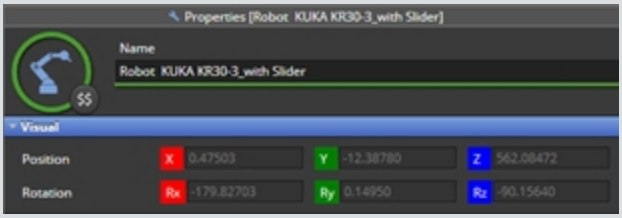


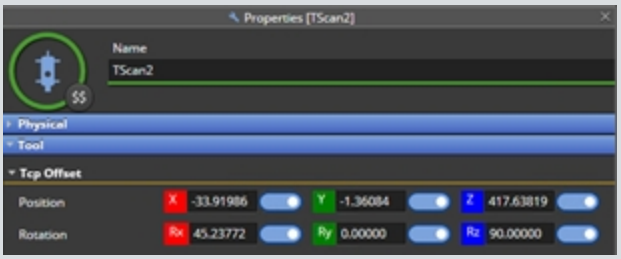

Next, activate the calibration, by right-clicking on *Calibration* | *Activate*.




Note that the robot base changed slightly when the calibration was activated.

Before

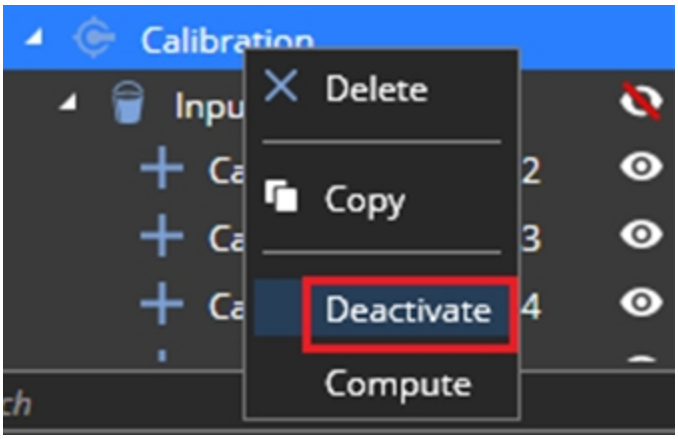


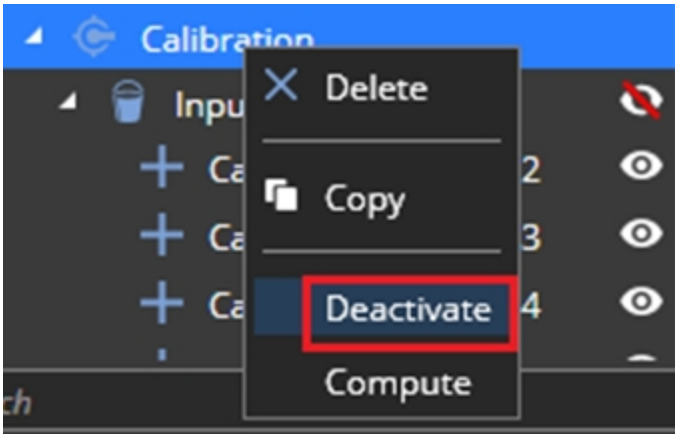
After

Now your robot is calibrated, and you can start using it to program and scan new parts.

Note that after you activated the calibration, you can deactivate it, and this will remove all the computed offsets. However, it will not revert the robot to its original position.



After you activated the calibration, you can deactivate it, and this will remove all the computed offsets. However, it will not revert the robot to its original position.




# Alignment - ZG

## Manual Tracker And Sensor Calibration

Calibration is divided into two types: factory calibration and user calibration.

1. **Factory Calibration**: This calibrates all parameters of the scanner, restoring it to its optimal state. It is recommended to send the entire scanner system to the after-sales service center for factory calibration every 12 months.
2. **User Calibration****: This calibrates some parameters of the scanner. It is recommended to perform user calibration when there are changes in the environment, temperature, or humidity, or when poor surface quality is observed.

### Calibration Precautions

- Ensure there are no other marker points near the calibration board.

- After use, please store the calibration board in its protective case.
- Ensure there are no reflective objects near the calibration board.
- Stay relaxed during calibration and move your gestures slowly.
- If errors, damage, or misplaced marker points are detected, the calibration process may fail.

## Manual Calibration Procedure

#### Tracker Calibration

When calibrating the Track, please ensure that there are no obstacles within the Track's field of view. Hold the calibration rod and move it to the position indicated on the software according to the software prompts.

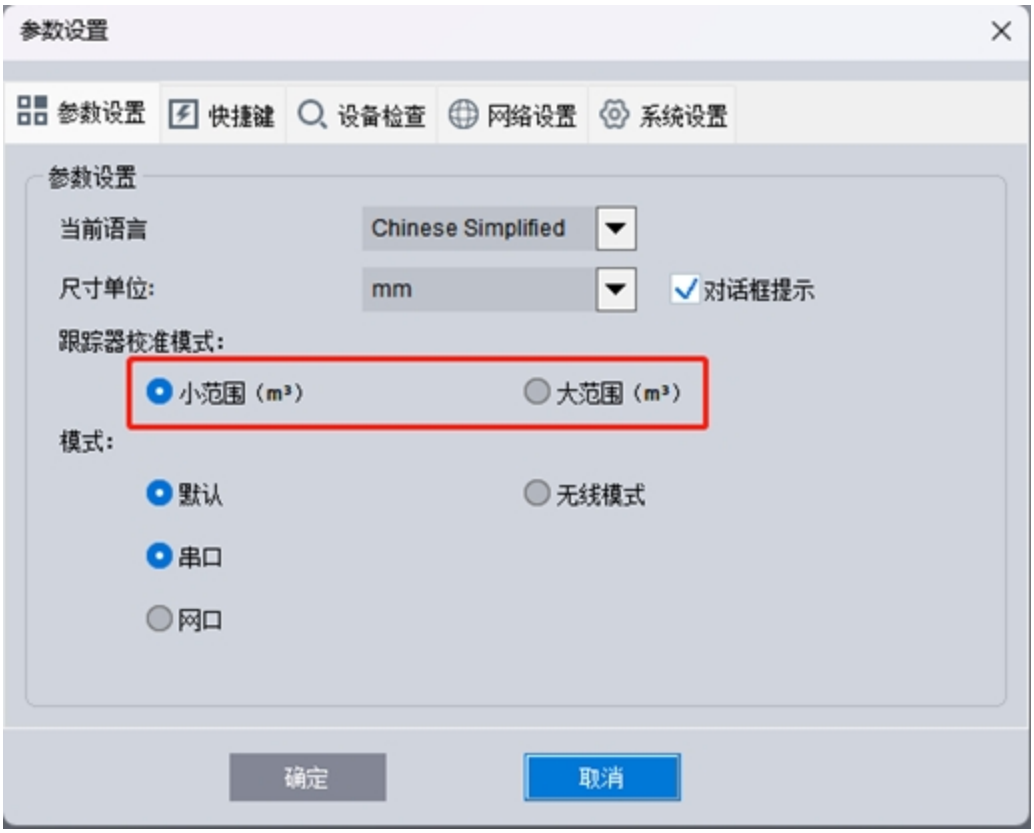
Small-scale calibration: There are a total of 36 specified poses. Once the current pose is aligned, it will automatically jump to the next specified pose. The calibration is completed when all 36 poses are aligned.

Large-scale calibration: There are a total of 72 specified poses. Once the current pose is aligned, it will automatically jump to the next specified pose. The calibration is completed when all 72 poses are aligned.

Setting method:

Click on "Options"(选项) in the upper left corner of the software.

Tracker Calibration Mode Selection: Small Range(小范围)/Large Range(大范围)




Calibration Steps:

---

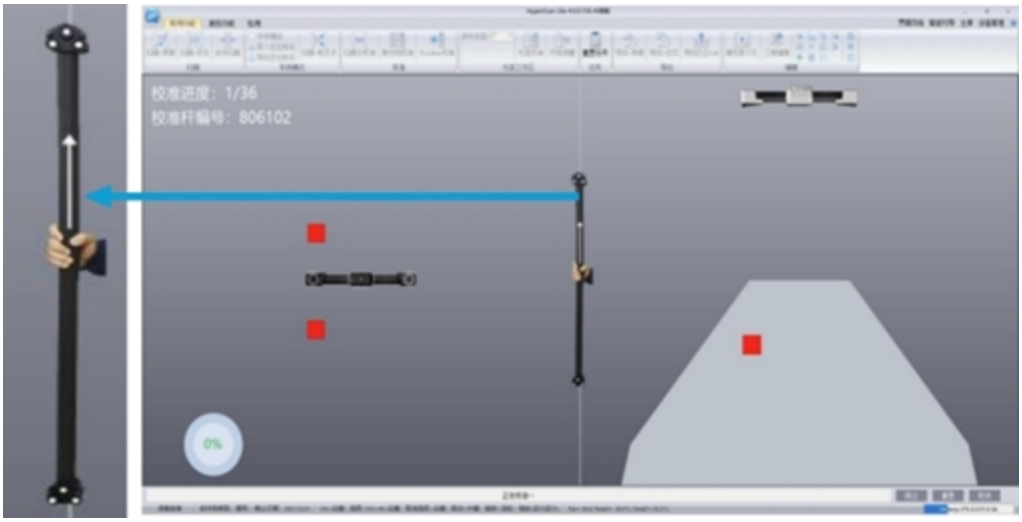
| | |
|---|---|
| Alignment | 332 |

Place the Track as shown in the following picture, and make sure there are no obstructing objects within the frontal field of view.

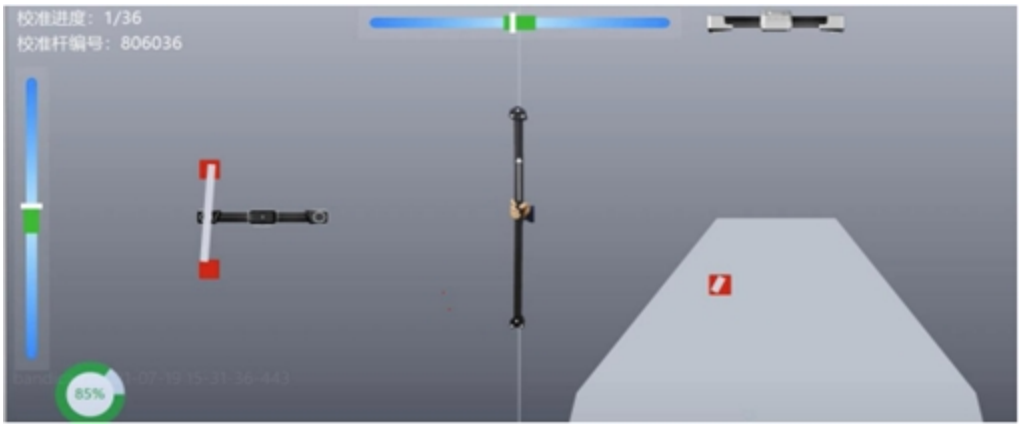
1. Click "Tracker Calibration"(跟踪器校准) in the software.

2. Place the provided calibration rod within the Tracker's field of view (position the calibration rod vertically in the field of view with the flat side facing forward to the Tracker). The interface will appear as shown in the figure below:

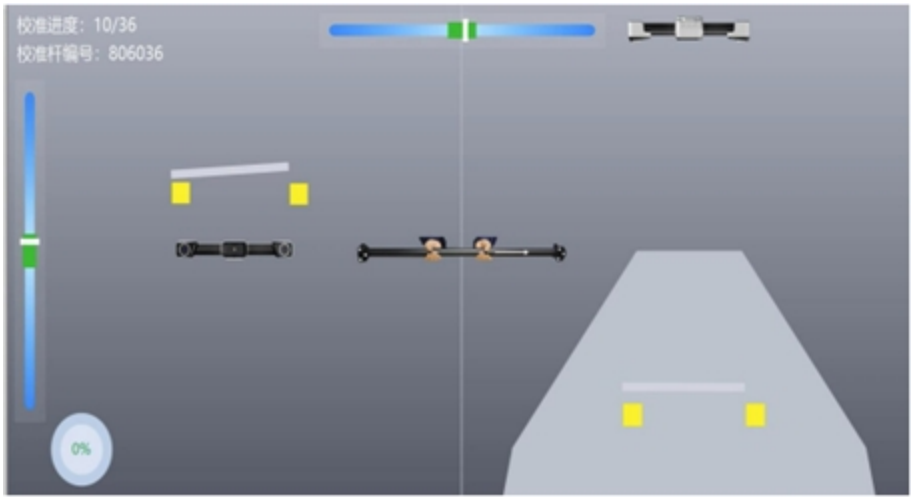
The six points on the flat side will point in one direction, which should be pointed straight up at the beginning of the calibration



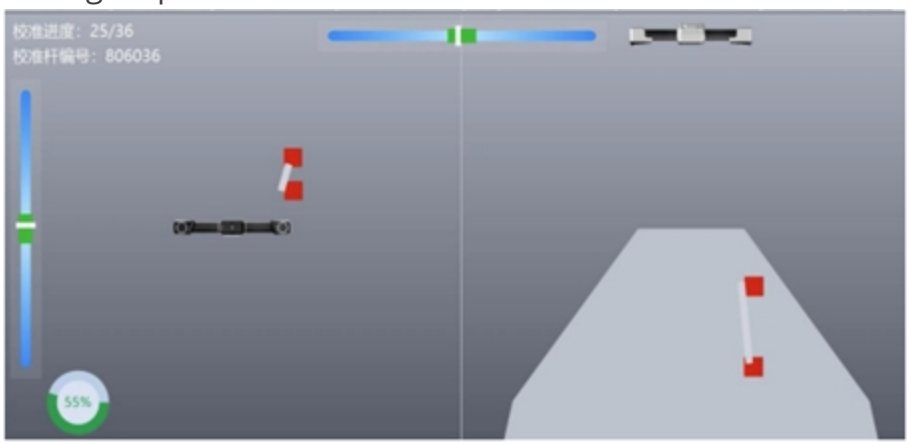

Move the calibration rod to the position indicated by the red box in the software according to the prompts. Continue adjusting until the view turns green which means the calibration for this position is completed. Then the calibration rod needs to be moved to the next position.

4. 

Calibrate to step 10, and place the calibration rod horizontally (refer to the calibration interface instructions) for calibration, as shown in the right figure.

5. 

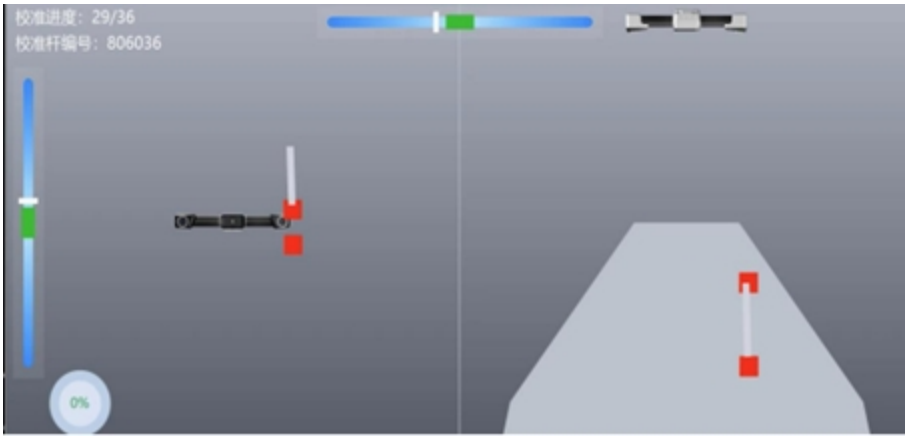

Calibrate the progress to step 19. Change the grip of the calibration rod again (refer to the calibration interface instructions), and place the inclined side with an angle upwards to start the calibration.

6. 

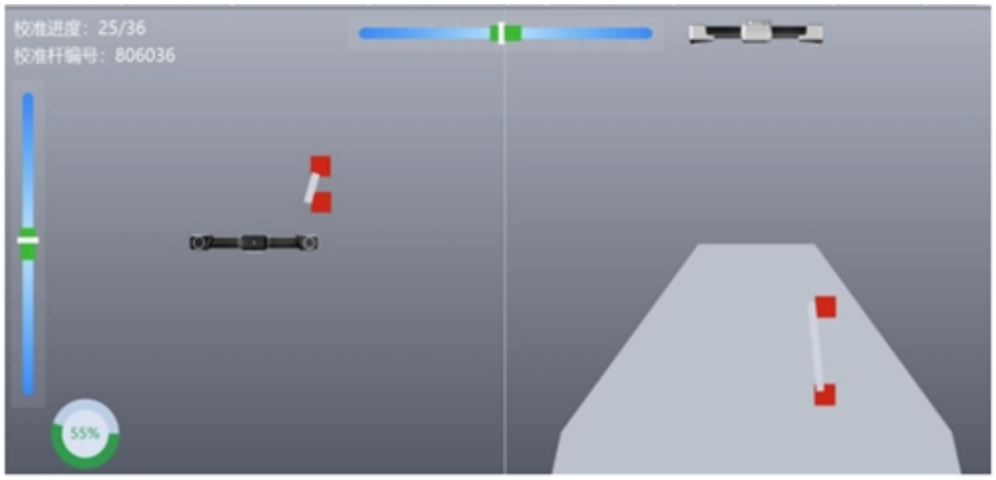

7. Calibrate the progress to step 28. Change the grip of the calibration rod again (refer to the calibration interface instructions), place the inclined side with an

| | |
|---|---|
| Alignment | 335 |

angle facing down, and start the calibration.




The calibration accuracy is popped up after the calibration is completed, and the normal accuracy of calibration RMS value need to be within 0.05mm.

8. 

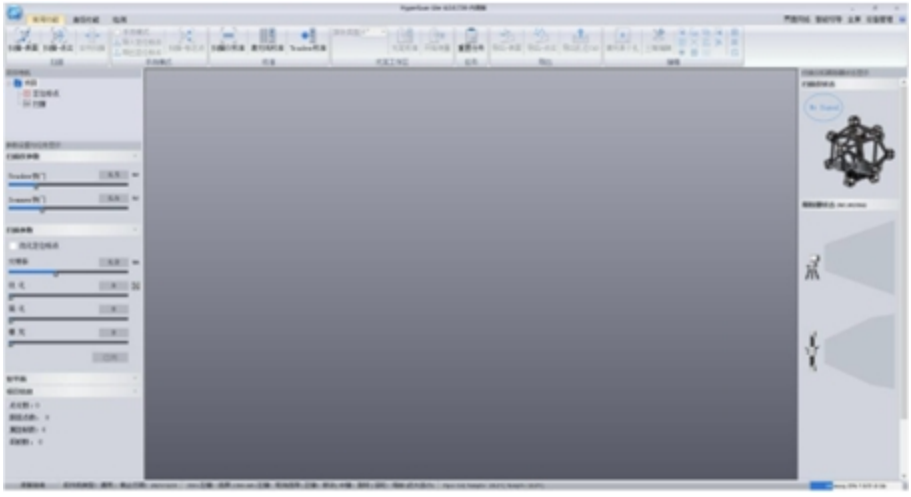

Shutter setting is 0.5ms when the tracker is calibrated

Position the calibration rod as indicated and move it back and forth, keeping the position consistent with the shadow position in the 3D viewer.

**HyperScan Scanner Calibration**

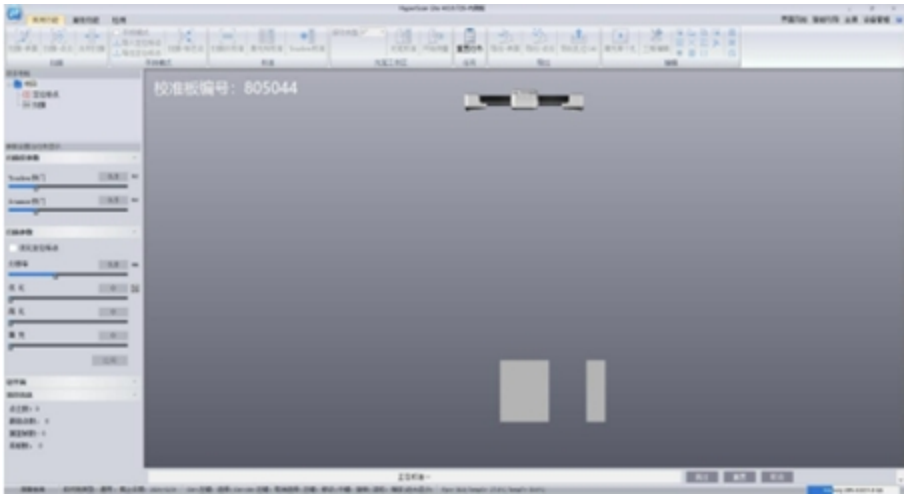
1. Take the calibration board out of the protective box and open it, place it as shown below, and place the calibration board stably in the field of view of the tracker:

When calibrating the Sensor, the three points circled on the calibration plate should be directly facing the Tracker

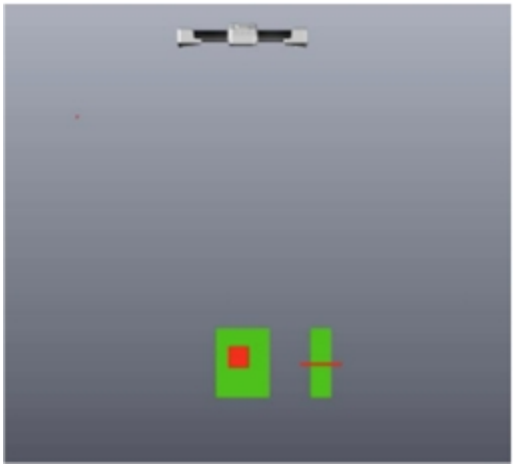


2. Open the scanning software, as shown below:

   Click the scanning software "Scanner Calibration", and the calibration interface will pop up as shown below;

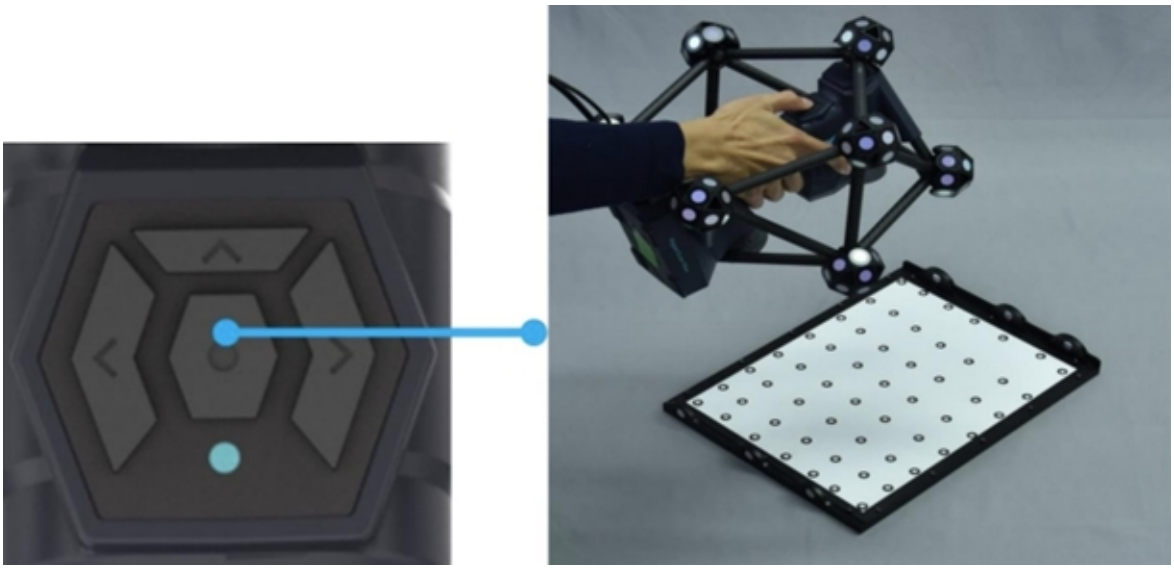
3. 


   Move the calibration board so that it displays the position in the green area on the software;

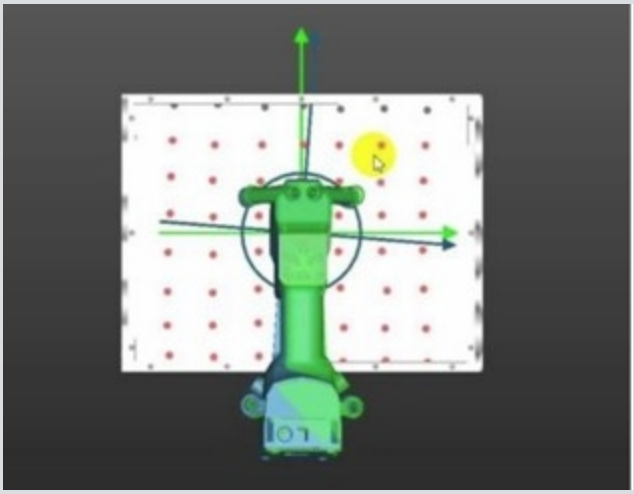
4. 

5. Align the device with the calibration plate and press the laser trigger key;

---



If you move the device alignment board, you can see that the indicator bar on the interface changes.



- During calibration, the scanner position posture is moved so that it coincides with the corresponding three-dimensional position in space. The current position is automatically calibrated and jumps to the next position. When all the positions are

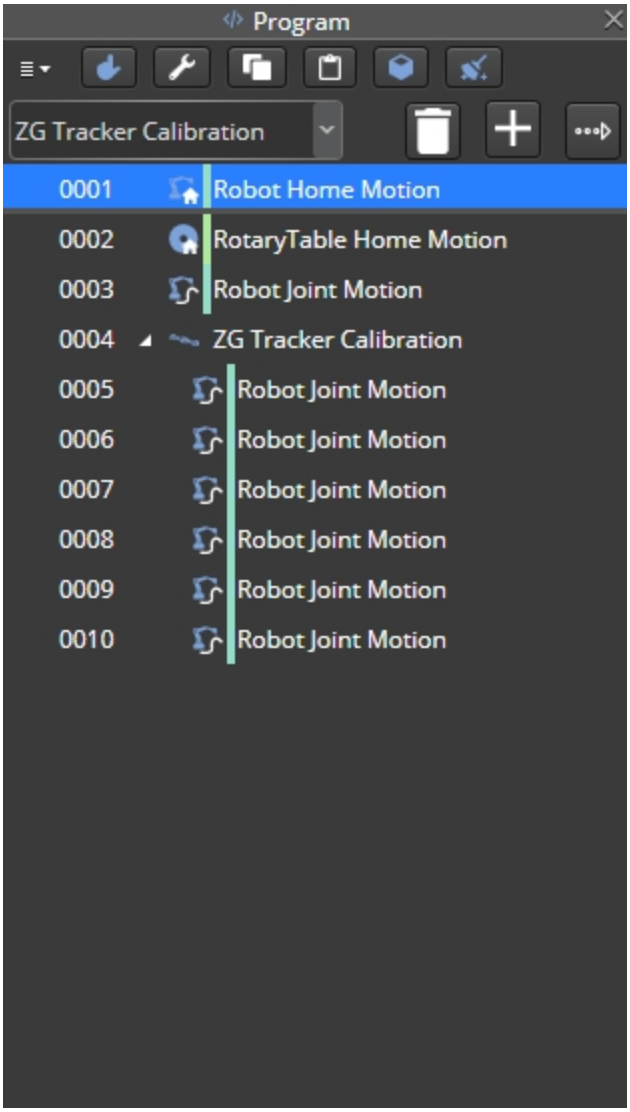
| | |
|---|---|
| | aligned, the scanner calibration is completed.<br>• When calibrating, the shutter is set to 0.6ms;<br>• Do not overshade the scanner while calibrating, and keep ZG-Track to see as much of the spherical target on the scanner as possible. |

## Tracker And Sensor Calibration using HxGN Robotic Automation

### Tracker Calibration

| | |
|---|---|
| | The calibration rod will need to be mounted on the robot before starting the program |

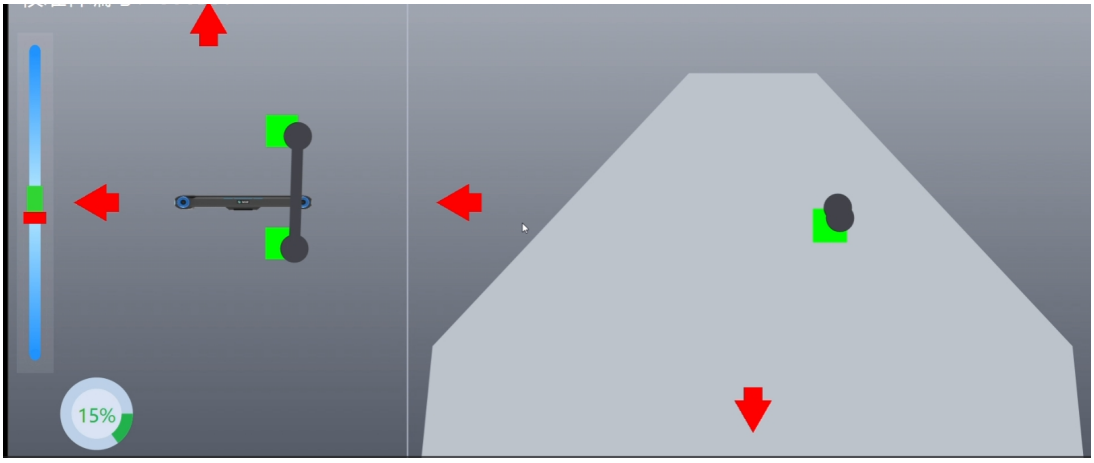
The program will look like this:



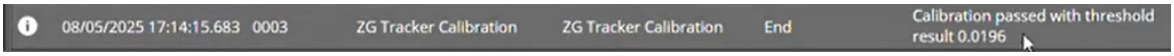

1. Start with all cell elements at home (Rotary table, Tracker Slider, Robots)
2. Move the robot with the control rod mounted on it towards the tracker vision
3. Add the tracker calibration command

---

- Add the necessary robot join motions to arrive at all the positions requested by the HyperScan software



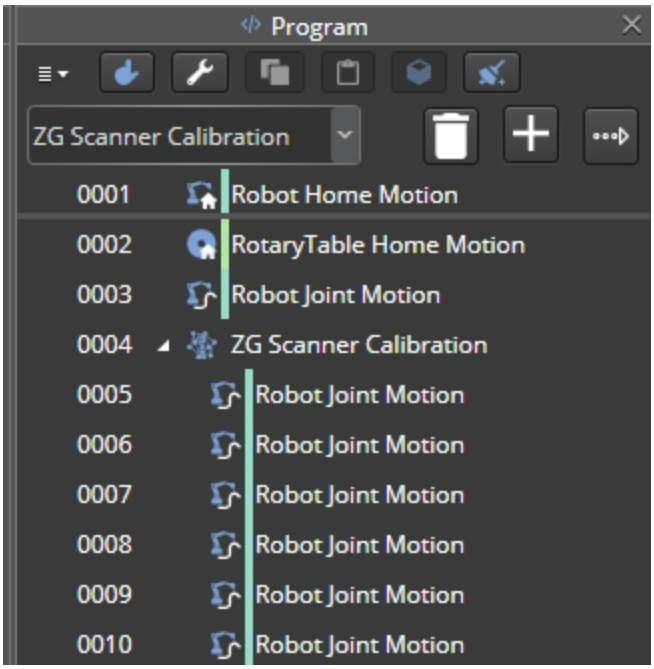

After the program has run in HxGN Robotic Automation, the Execution Status window will show a message with the results of the calibration.

4. 

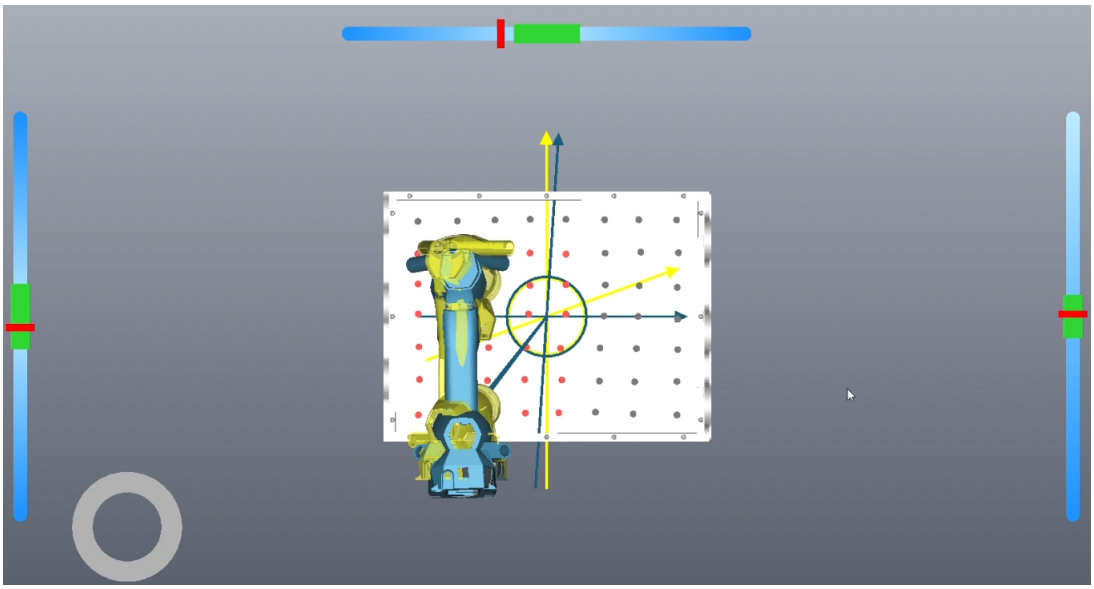

## Scanner Calibration

| | |
|---|---|
| | The scanner calibration plate will need to be available in the cell before starting the program |

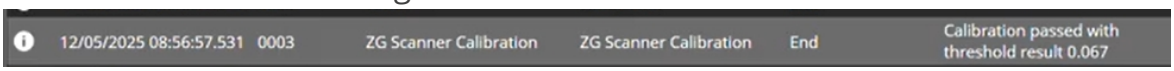
The program will look like this:




1. Start with all cell elements at home (Rotary table, Tracker Slider, Robots)
2. Move the robot with the tool mounted on it towards the scanner calibration plate
3. Add the scanner calibration command
    - Add the necessary robot join motions to arrive at all the positions requested by the HyperScan software



After the program has run in HxGN Robotic Automation, the Execution Status window will show a message with the results of the calibration.

4. 

# Programming

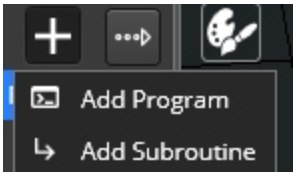
When the cell is ready for work, i.e., the **fixture** and **part** are loaded, properly aligned and the **features** to be measured are available in the project tree, it is possible to program the robot movements needed to inspect the part.

| | |
|---|---|
| ⚠ | Make sure to create or import all other objects in your cell. Avoiding collisions is a major part of the planning and so it is crucial for the cell to be precisely modelled. |

| | |
|---|---|
| ⓘ | This section will mainly talk about all the programming commands that are generic to the sensor used. |

## Program versus subroutine

The Add button now has an option to either create a new program or to create a new subroutine.




Subroutines are programs that can be called from another program.

Subroutines abide by the following rules:

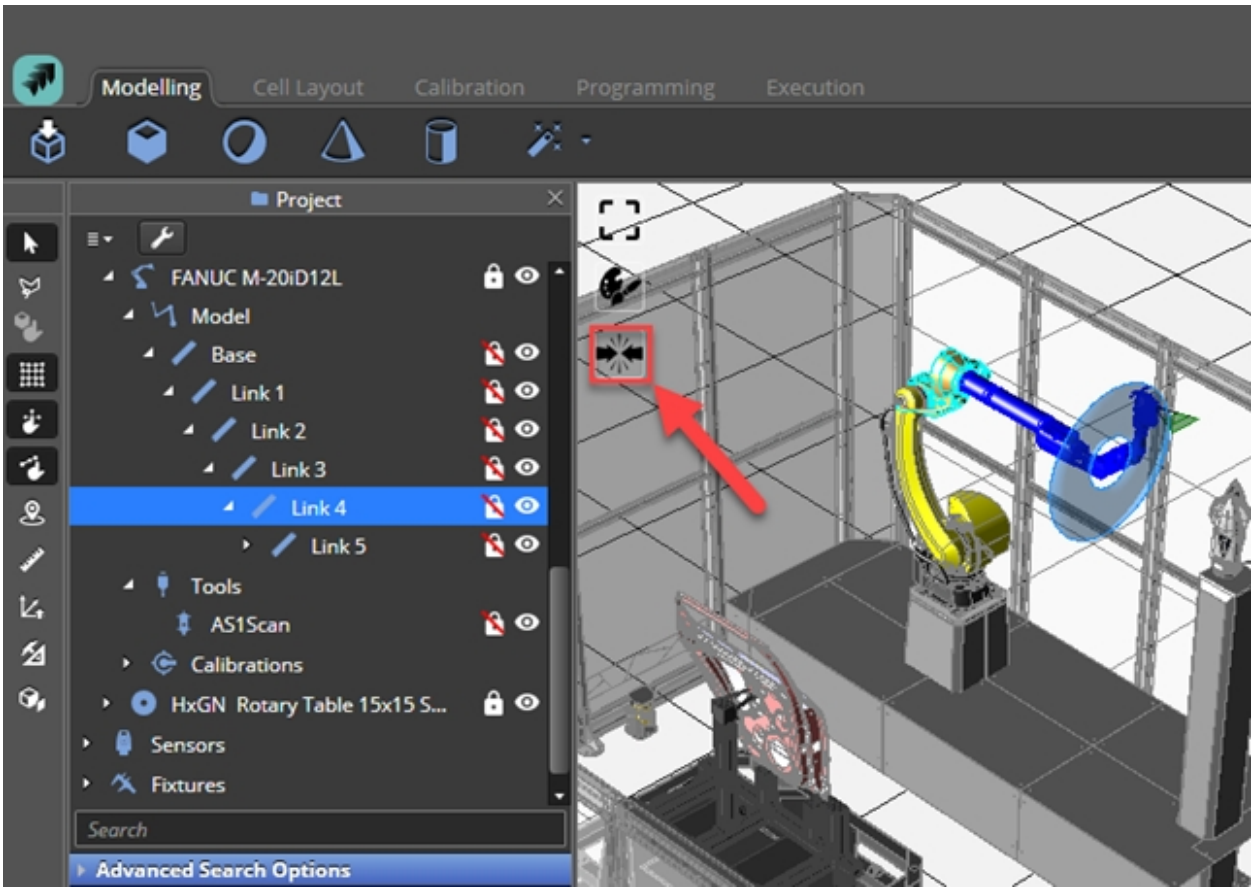
- Adding a subroutine to a program will show the subroutine as read only. No modifications can be made! Only by going to the subroutine itself you can change the commands.
- Changing a subroutine will affect all the programs that contain that subroutine. All those programs will need to be validated again.

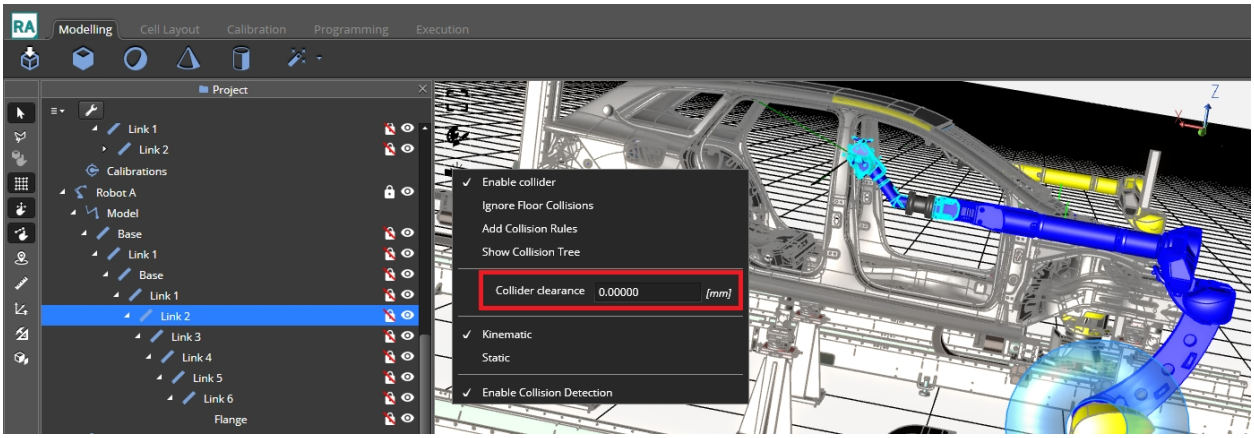
- A subroutine cannot have any recursion. (E.g. subroutine 1 refers to 2, which in it's own refers back to 1
- A subroutine cannot be deleted if it has been added in a program. First it will need to be removed from the program.

A program cannot contain 2 identical subroutines which both include the exact same measurement routine! Moving cell items or interacting with the PLC is fine however.

## Clearance settings

To generate and achieve a safe path, user may use the clearance option to virtually expand the object volume by increasing clearances. User can set a clearance value per each object by clicking on the object and set its value in [mm] by selecting the machine link in the project tree or in the 3D viewer and clicking on the Collisions button. The default setting for all elements is 0 [mm].

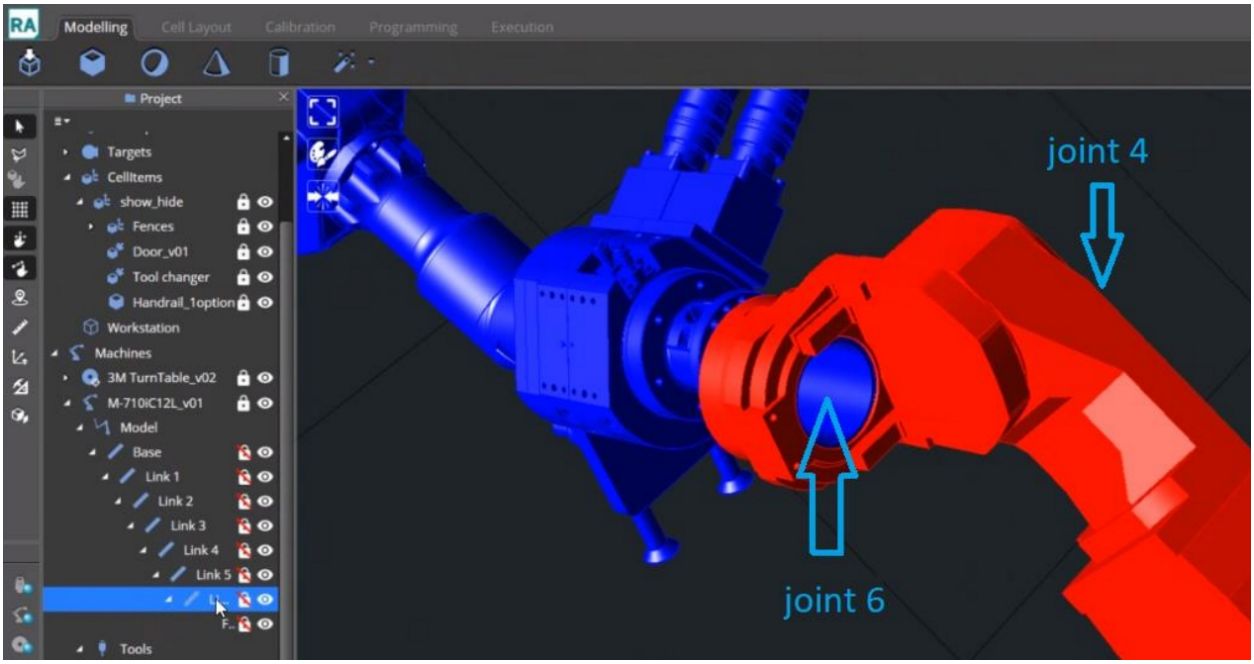



Adding clearance can be beneficial for collision but should be done carefully since it will limit the motion and area for the planning path. It is

| | |
|---|---|
| | recommended to add clearance only for the Tool with value of 20-25 [mm]. |
| | Both Leica trackers have a minimum clearance. This cannot be reduced due to a risk of collision. The AT960 has a minimum clearance of 110 (mm) and the ATS800 of 90 (mm). |

It is not recommended to add clearance neither on the part, nor the fixture nor the Rotary Table nor the robot joints. The main risk with adding clearance to Robot joints is internal collision, meaning collision with its own joints. In some cases, user will need to test if there is a collision by moving the joints e.g., if user add clearance of 20 mm on joint 4 and in some positions, there is a collision between joint 4 and joint 6 as shown figure below.




*Collision between Joint 4 and Joint 6 due to an excessive clearance.*

During the automatic path planning, HxGN Robotic Automation adds 10 [mm] clearance to the Tool in addition to the user clearance definition to achieve a safer program and solve discretization issues.

The robot joints clearances are also added 10 mm but only for joints of which the clearance value was changed to a value greater than zero (e.g., if joint 4 clearance is as default zero, it will stay zero, but if user changes it to 2 mm, then actual clearance will be 12 mm).

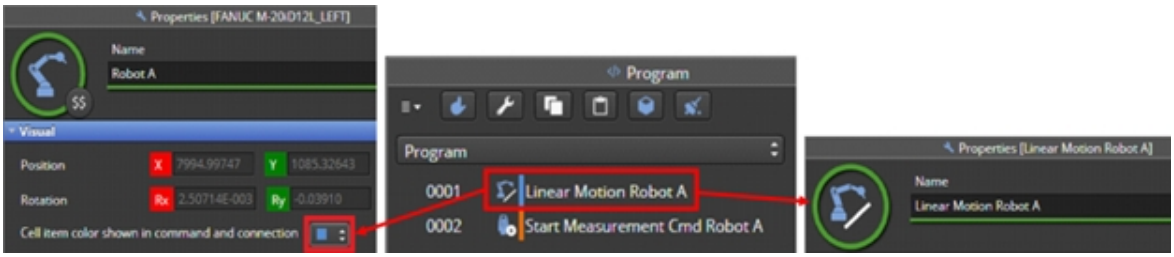
In all cases where user changes the default clearances value to a different value than the recommended values, user should move the robot joints within their range and verify there is no internal collision.

## Language syntax

A program in HxGN Robotic Automation is a list of commands that will be executed by the involved cell items either sequentially or in parallel.

Every command is represented by:

- **Program line number**: automatically assigned when the command is added
- **Icon**: automatically assigned depending on the command type.
- **Color bar**: automatically assigned but can be changed in the item properties (see chapter Changing machine properties).
- **Name**: automatically assigned but can be changed in the command properties.

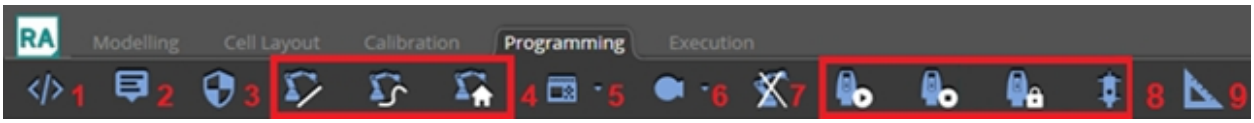



*Example of two commands and related syntax: the first command has index "0001", a "Linear Motion Command" icon, the color line as defined in Robot A properties, and a "Linear Motion Robot A" name defined in the command properties*

## Supported commands

The complete list of supported commands is available in the ribbon toolbar within Programming tab.

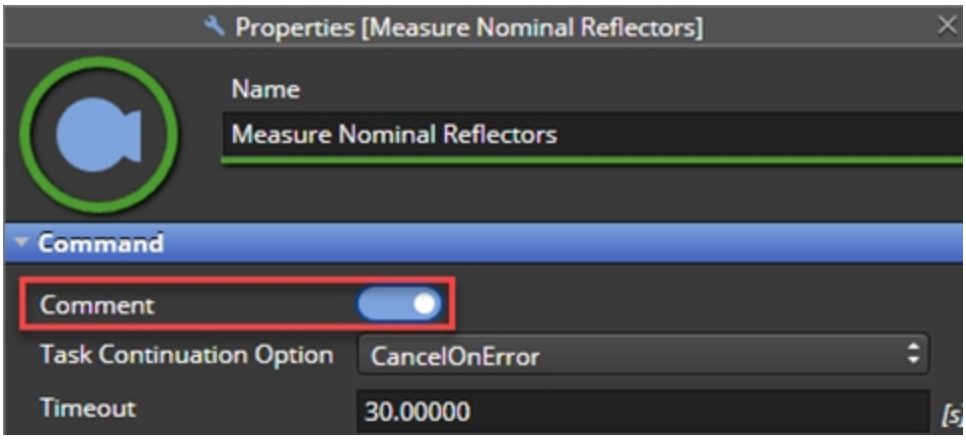
If there are multiple commands available within a specific command type, a drop-down menu is used to list them.

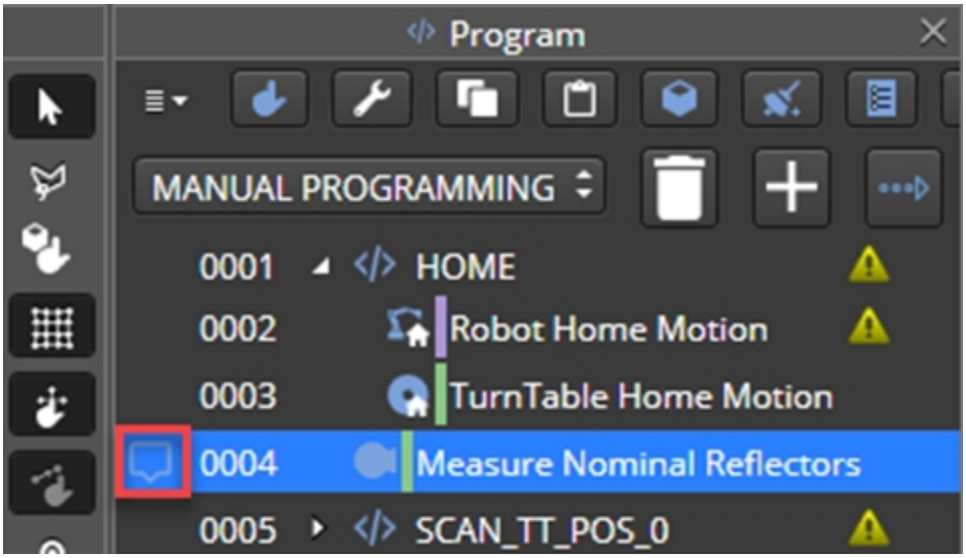


Some commands are only visible if the applicable sensor (e.g. ZG or Leica) or the applicable Metrology (e.g. PC-DMIS or PolyWorks)

All commands share some common properties:

- **Name**: will be visualized in the program tree
- **Command** area having:
    1. **Comment**: if the toggle switch is enabled, the command will be skipped in the execution. It is also possible to comment a command by double-clicking it in the program tab.



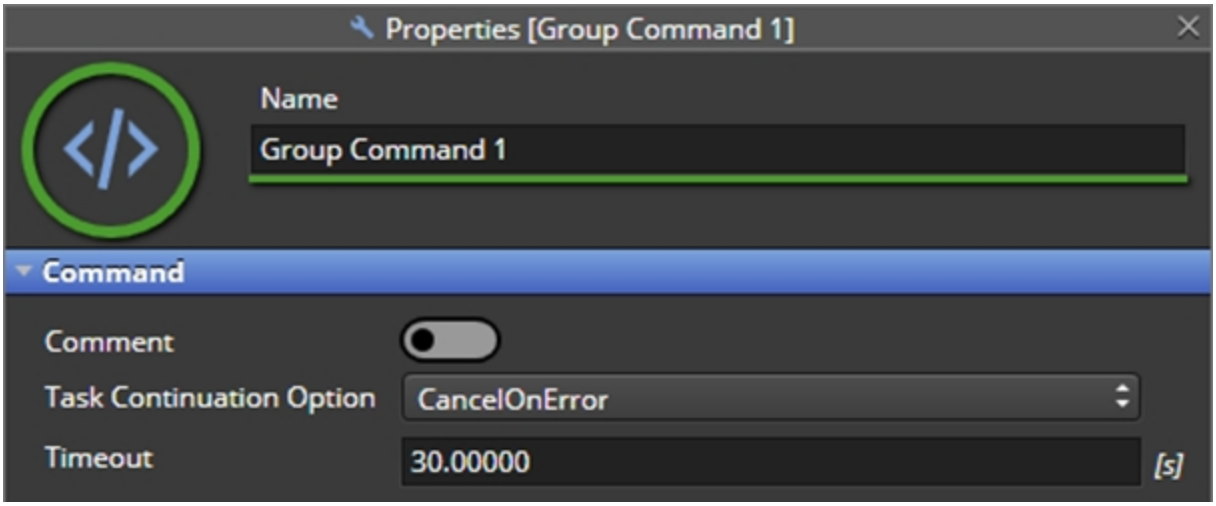

To set a debug point instead of a comment it is needed to single-click the command line where the execution must stop.

2. **Task Continuation Option**: (not available for Motion commands)
    a. **CancelOnError**: if the commands fail, the program will be immediately stopped with an error

b. **ContinueOnError**: if the commands fail, the program will continue

c. **Undefined**: if the commands fail, the program will continue

3. **Timeout**: expressed in seconds. If the time defined elapses before the command execution is finished, the execution is stopped with a timeout error.

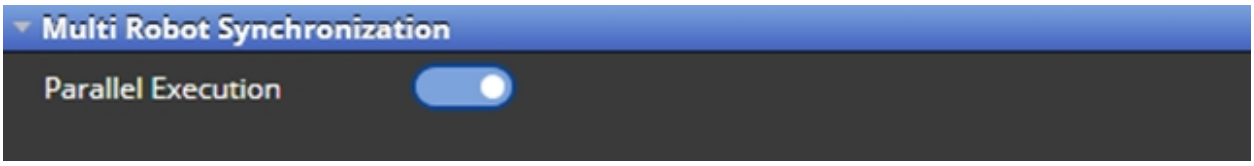


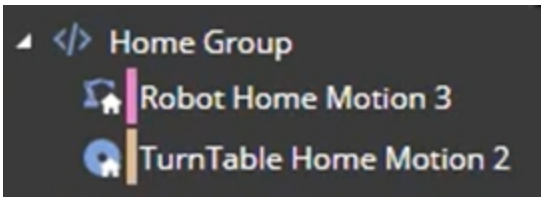

Depending on the command type, there will be dedicated properties as described in the following sections.

## Grouping

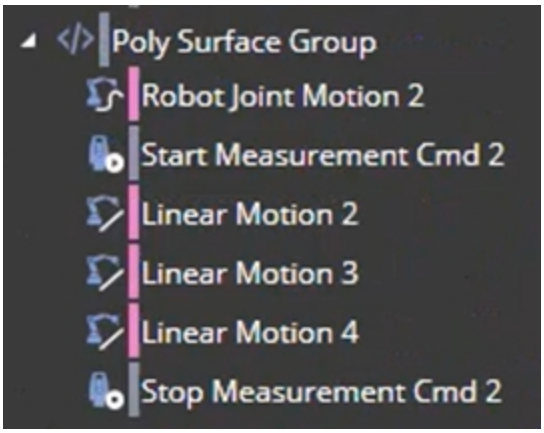
Grouping ( ) can be used to improve the readability of a program by putting together commands according to a common criterion, e.g., related to a feature scan or to homing movements of all the machines used in the cell.

Once grouped, the commands will be executed in sequence, unless the **Parallel Execution** switch is turned on. See chapterParallel Programming for more information on parallel execution scenarios.




Example – Grouping homing commands

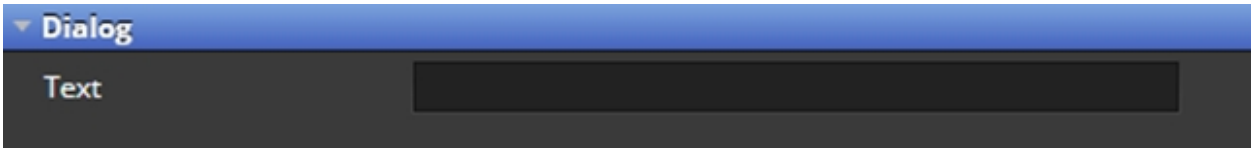



Example – Grouping feature related commands

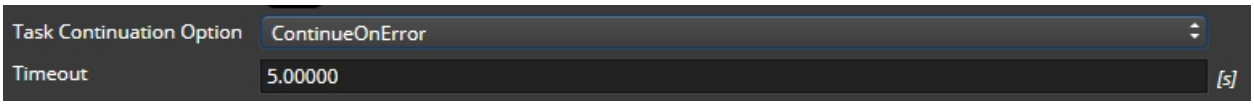
## Dialog

A dialog command ( ) can be used to display a pop-up window with the text specified in the **Text** property..

User will need to click on **OK** to continue the program execution.




The dialog command will take into account 2 options:

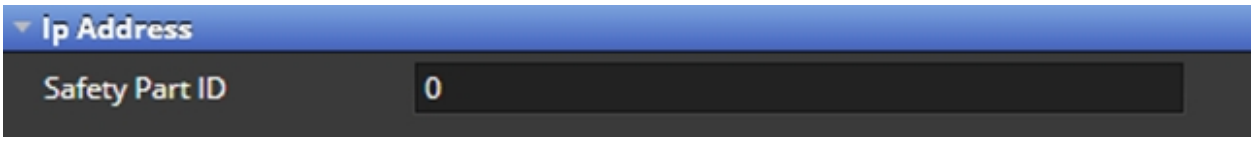



- **Timeout**. The time the user has to acknowledge the Dialog Box
- **Task Continuation Option**:If not acknowledged within the timeout, it will either stop the program (CancelOnError) or continue to the next command (ContinueOnError).

Dialog commands will not be used in programs executed with the Execution Manager.

## Safety

A safety command ( ) can be used to trigger a specific program represented by its ID (previously configured on the safety device) in the command properties.




Currently HxGN Robotic Automation only supports Keyence Pokayoke camera system.

## Motion

Depending on the selected machine (robots, rotary tables, sliders), the related commands will be shown in the programming bar in this area as explained in the next sections.

All motion commands share the same properties' structure that differs one from another depending on the machine type:

1. **Motion** part has the machines involved in the motion command (e.g., the robot and its tool) that can be selected using the hand icon ( )
2. **Target** part has the target position to be reached either in cartesian (**Target Transform**) or joint space coordinates (**Target Joint Values**)
3. J**oint/Linear** part has the velocity and acceleration settings for both a joint

and a linear movement.

Motion
Dependencies
1
Select Machine
FANUC M-20iD12L
Select Tool
AS1Scan
Target
2
Target Transform
Position
X
-309.82785
Y
0.01790
Z
2870.30768
Rotation
Rx
93.00000
Ry
1.55257E-003
Rz
-90.00135
Target Joint Values
Internal Joints
θ 1
0.00000
θ 2
-24.00000
θ 3
0.00000
θ 4
0.00000
θ 5
42.00000
θ 6
0.00000
Joint
Motion Settings
3
Velocity
Acceleration
40.00000
[%]
100.0000
[%]

### Robot

When a robot is selected it is possible to add three types of commands according to the picture below:

1. **Linear Motion**, to specify a cartesian position of the robot.
2. **Joint Motion**, to specify the position of the robot in the joint space- either 7 DOF or 6 DOF depending if the robot is mounted on a slider (see chapter Robot)

### Rotary Table

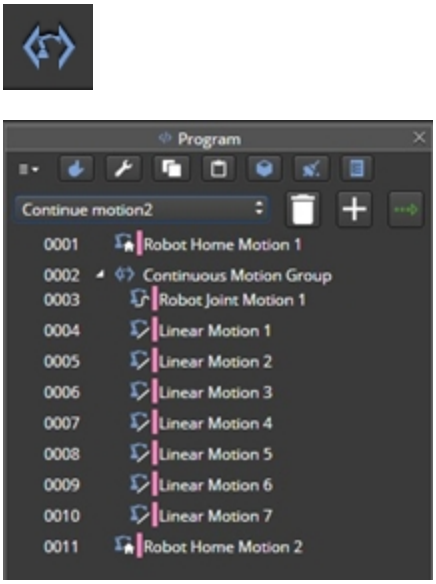
The same applies when a rotary table (see chapter Rotary table) is selected but linear motions are not available, and the joint space is 1 DOF.

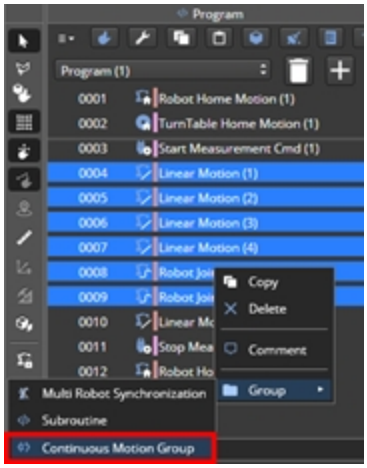
### Slider

The same also applies for sliders (linear joints) where usually a tracker is mounted. In this case depending on if the slider is only horizontal or vertical the joint space will have only 1 DOF, if the slider can move in both vertical and horizontal direction, the joint space will have 2 DOF (see chapters Slider).

### Continuous Motion group

It is possible to speed up the execution of robot motions reducing drastically the stopping time between different linear or joint motion commands by placing them inside a Continuous Motion Group, either adding first the group command in the program and then adding the other commands inside (left picture below), or by selecting motion commands already present inside the program and adding them by right clicking and selecting the correct option (right picture below).




In case the user executes the program non from the beginning but selecting a program line inside a continuous motion group, the behavior of all the following commands within the group will be the standard one, as the commands were executed externally to the group.

## PLC

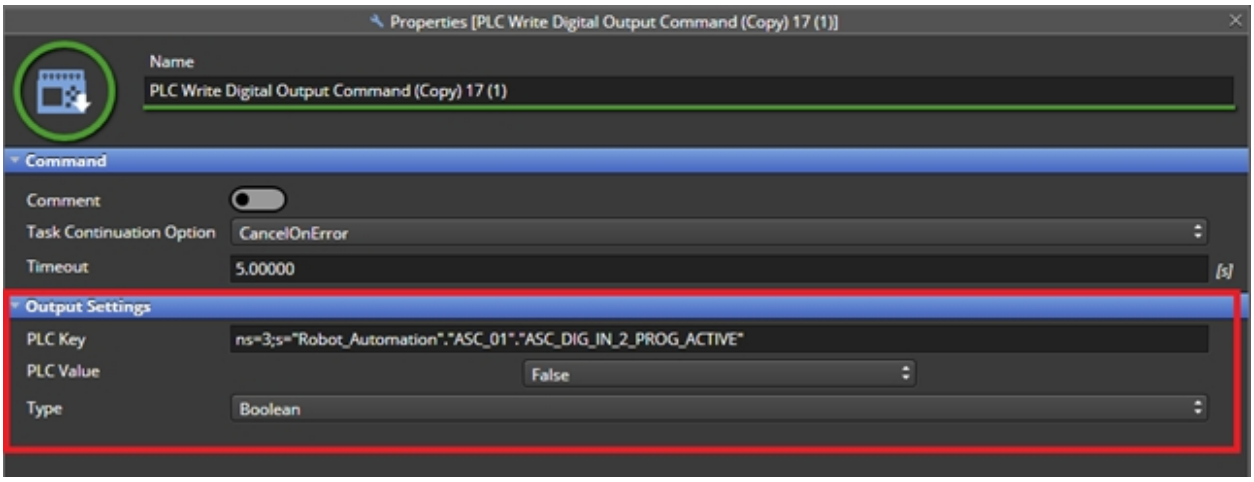
PLC commands are used to communicate with the cell PLC that is responsible to control the overall main logic of the robotic cell and it is the interface with all the other devices connected to it, e.g., motion and safety devices. The communication

with the PLC happens through and OPC UA interface where the PLC acts as an OPC UA server and HxGN Robotic Automation as an OPC UA client.

The available commands are:

1. **PLC Write Digital Output**: to be used to set a Boolean value (true or false) to a variable defined in the PLC.
2. **PLC Wait**: to be used to wait that the PLC has received the output signal and the consequent action was successful (e.g., an action is performed on a linked HW device). The program will wait until the PLC communicates the successful handling of the signal.

In both cases the variable settings are done within the **Output Settings** section within the command properties where it is needed to specify the PLC tag to communicate with over OPC UA in the PLC Key field.



### Acquire/release Lock for Multi Robot Synchronization

In multi-robot configurations, there may be situations when both robots running in parallel are only allowed to enter a zone one at the time to avoid collisions. To deal with such cases HxGN Robotic Automation makes available for the user the **Acquire/release Lock** for **Multi Robot Synchronization** command ( ).

The essence of this command is that the children of only one of such commands (by means of resource name) must be executed at the same time so whichever command (e.g., Lock Resource A) locks a resource which is specified in the *Multi Robot Synchronization* | *Resource Name* field will execute; the other commands sharing the resource (e.g., Lock Resource B, Lock Resource C, and so on) will wait for the Lock Resource A execution to be over.

**Pro Tip**: programmers can see the analogy between this command and a C#'s "lock" or a C++ "critical section" keywords, which allow only one thread to enter the part that is locked, and the lock is not shared with any other processes.

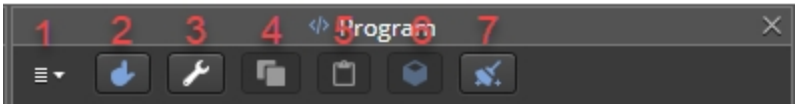
### Metrology

Metrology commands group has a list of commands which are specifically developed for special functionalities available in the interface with PC-DMIS, Inspire and PolyWorks.

- Measure **PC-DMIS Static Points** is specific to PC-DMIS, and it is explained in chapter Programming - Leica.htm
- **Execute PolyWorks Macro** and **Measure** PolyWorks **Reference Points** are specific to PolyWorks. Please refer to chapter Compensation for the explanation on this command usage and properties.

## Programming tools

A toolbar above the program can help the user with complex programs.



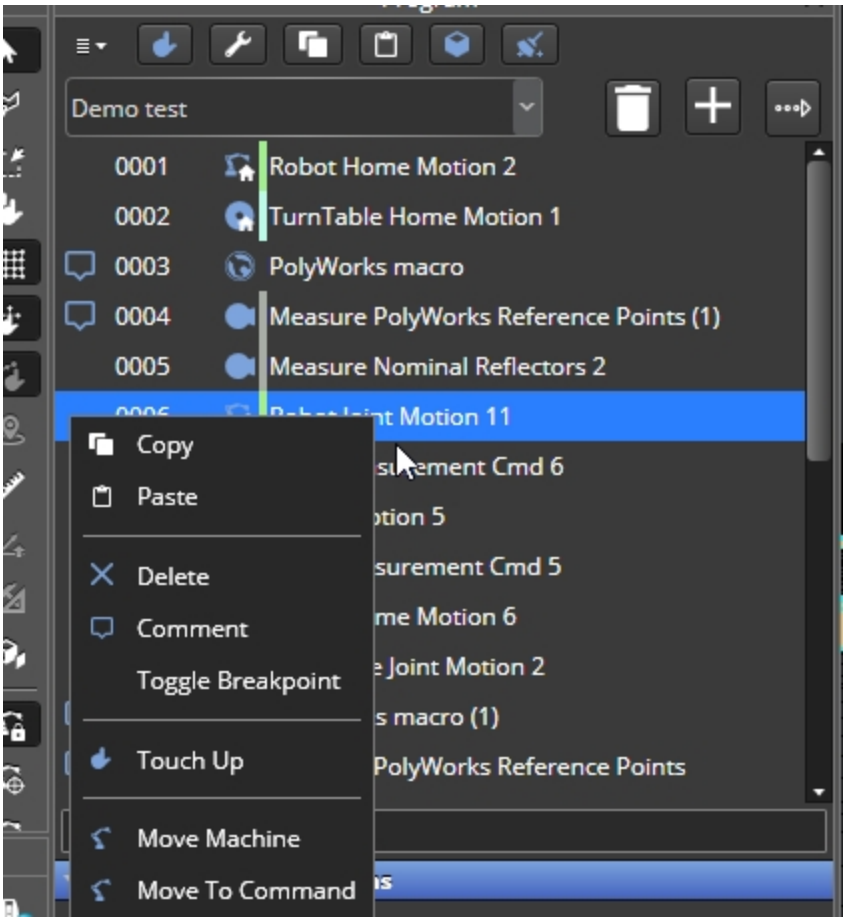

1. **Collapse/Expand** all the commands in the program.
2. **Select All** the program and all its commands and subroutines.
3. **Properties** show the program or project properties (depending on which tab you are)
4. **Copy** the selected commands (and subroutines) of the program (to afterwards paste them in another program).
5. **Paste** previously copied commands and subroutines into a new program (will be pasted after the current program if there are already commands and subroutines in place).
6. **Paste After** the previously copied commands and subroutines into a new program (will be pasted after the selected command or subroutine). This choice will allow you to paste a program to a specific subroutine folder for example.
7. **Clear Highlighting** in case the program stopped in error and the last active command will stay highlighted after debugging.

## Generic command menu

By right clicking on a command, it is possible to see the command menu that will show for all commands the following options:

1. **Copy** the command
2. **Paste** a copied command
3. **Delete** the command
4. **Comment**: disable command execution, see chapter Supported_commands

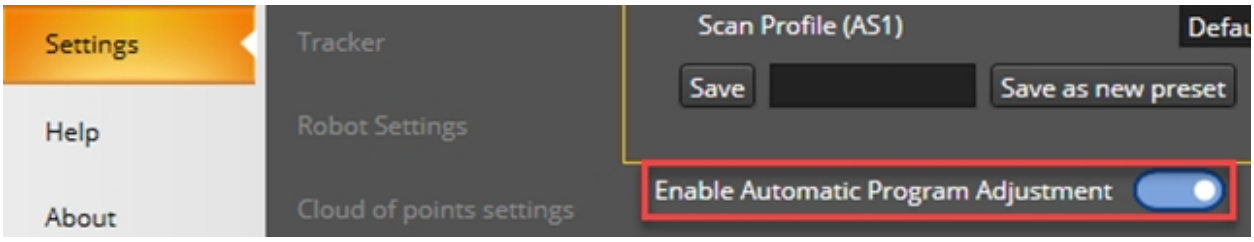
5. **Toggle Breakpoint**: set a debug breakpoint, see chapter Supported_commands
6. **Touch up** only available when selecting a motion command. this will set machine selected to it's position in the 3D world
7. **Move Machine** only available when selecting a motion command. this will move the machine in the 3D world to the target position of the command
8. **Move to command** will move the whole 3D world to the point in time where this command will be executed in the cell.
9. **Resolve error**: only available in case we detect a collision on a robot join motionpath planning will try to find an additional robot join motion(s) to avoid the collision



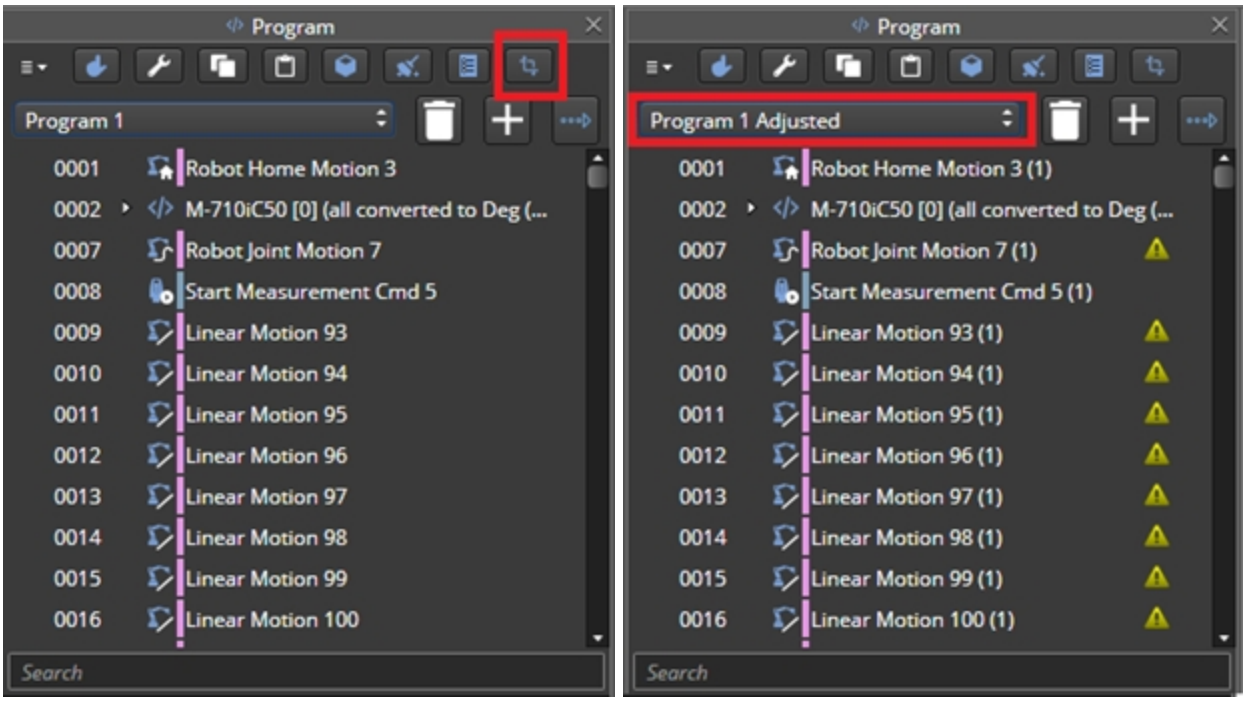

## Automatic Program Adjustment

This feature re-calculates an existing program when there is a change in the model. This allows the user to migrate old programs according to a new model. It may happen that a program cannot be fully migrated due to the new model (e.g.: collisions, parts out of reach, etc.). In that case, the new program shows which motions could not be migrated properly and must be resolved manually by the user.

The choice can be activated under *Settings* | *Planning* with the relative toggle switch. By default, it is **disabled**.




In the Programming tab, you can find the Automatic Program Adjustment button. After selecting an existing program, if you click the button a new program is created with the same name of the original program plus “Adjusted.” If there was a modification in the model, the new program will adjust the earlier one to scan the same area if it is possible. Otherwise, the resultant program will be the same.

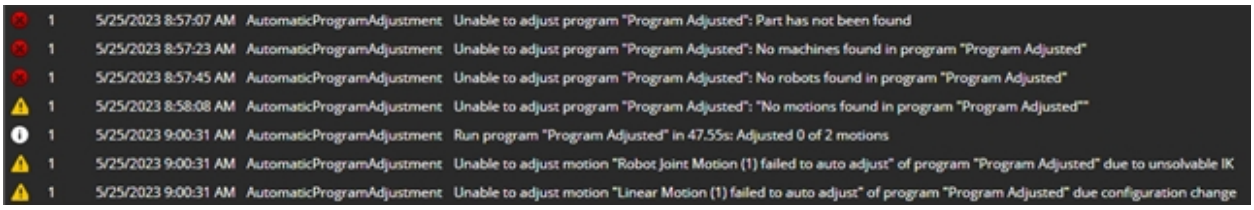



The execution time of adjusting automatically a program should be within the order of a few seconds at most for long projects. For simple robots, even less than a second.

| | |
|---|---|
| | It is always recommended validating the newly adjusted program to avoid potential |

| | |
|---|---|
| | collisions. |

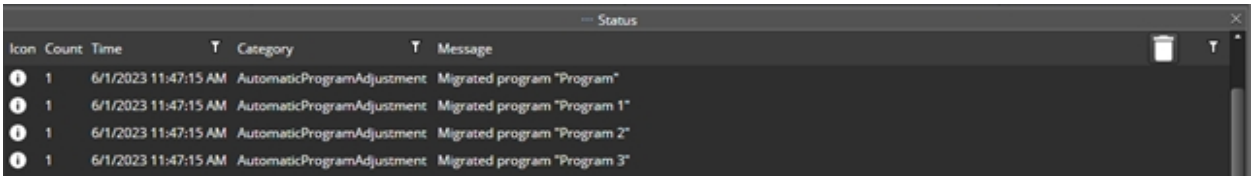
All information about the automatic adjustment like errors, warnings, and info messages are shown to the user in the Status window as per example below.




## Migration of old Programs

Since the old programs do not have the current parameters, we create those parameters when loading the project. That allows us to adjust the program automatically if it is needed. Recall that to preserve those changes it is mandatory to save the project before closing RA.

Every time an old program is migrated, the following message is shown to the user in the Status window:




## Added notes

1. If a part is located on another machine, the original program will not be adjusted to the new machines, since exclusively the machines of the original program will be considered. The programs have information on the machines by their ID, which is never changed when the program is adjusted.
2. If a motion cannot be adjusted, check out that it is reachable by doing a Pick On Cad (see chapter Pick on CAD for further explanation of this functionality).

3. For an adjusted program, the adjusted motions may lead to collisions. For linear motions, collisions can occur between the internal structure of the robot and the environment; for joint motions, collisions can occur also with the tool. This is because the new program is not checked for collisions, this is for the user to fix with a manual program. Consider manually changing the configuration of the robot at the motions with collisions to find valid adjusted motions.
4. If a program is generated with no motions successfully adjusted, the adjusted program will be the same as the original.

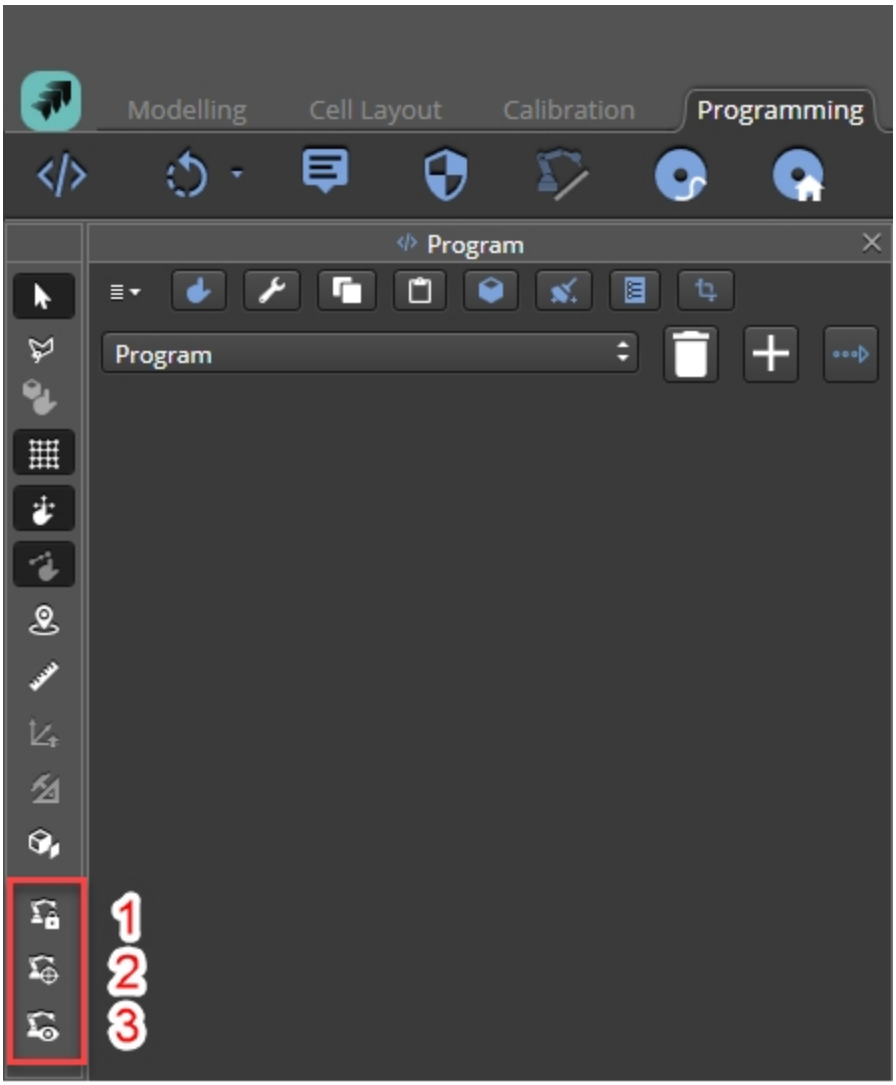
## Manual Programming Mode

In manual mode, the user would need to define the position where to move the robot to start a scan and the linear movements needed to scan the feature or area of interest. It is also needed to add all necessary compensation and homing commands to create a valid inspection program.

To do it, HxGN Robotic Automation have different modes that the user can choose depending on their needs.

### Programming mode special toolbar commands

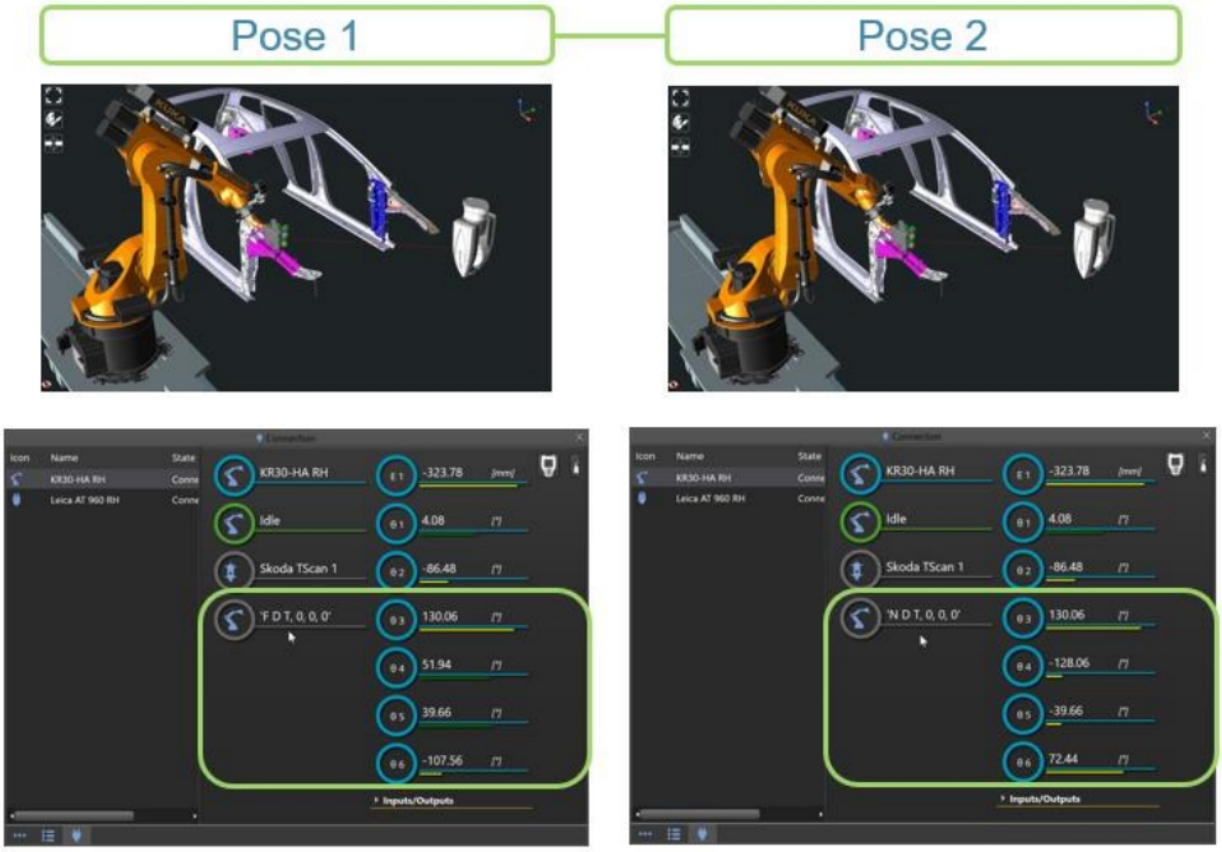
When programming mode is selected the special modes, toolbar will be expanded with three more commands.




1. **Lock Machines**: when enabled, allows coupled machines to be moved together, if it is disabled, those machines will be moved independently..
2. **Hold Cartesian Position**: when enabled, robot will move without changing the position of the TCP in the cartesian space, if it is disabled the robot will move with no restrictions.
3. **Change Machine Pose**: possibility to switch from the Multiple Robot Valid Poses (different Inverse Kinematics result) to reach a specific target TCP

---

position. Useful when doing Manual Programming or program fine tuning to avoid Robot Configuration changes or singularities.




## Controlling a cell item using the teach pendant

All machines that can be moved by the user have their own virtual teach pendants. The user can move them using the different settings available. To activate a teach pendant:

1. Select the **Connection** tab on the bottom window.
2. Choose the relevant machine.
3. Click on the **Teach Pendant** icon and the teach pendant will open.

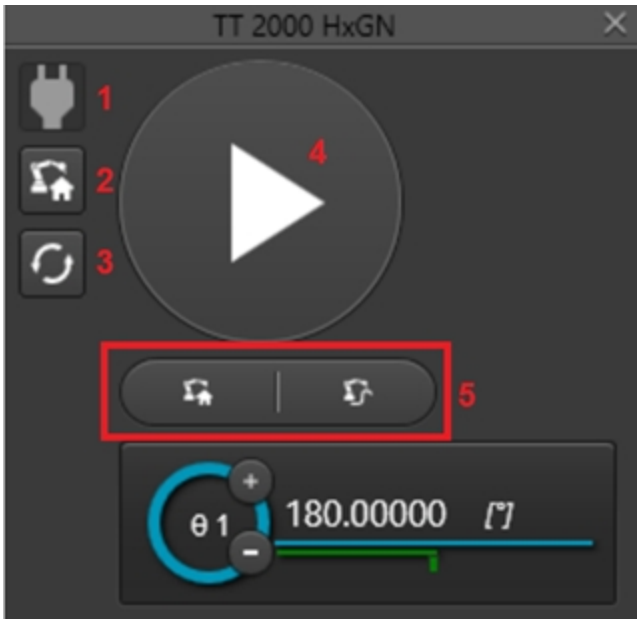
## Robot teach pendant

The following settings are available:

1. **Connect/Disconnect**: connects or disconnects from the real hardware.
2. **Go Home**: takes the robot to home position.
3. **Show GoPro** shows the camera of the tool (if one is attached to the machine).

4. **Position**: switches between different robot coordinate systems (Joint, World, Tool) to show the relative values accordingly.
5. **Motion Mode**: selects the way the robot will move (joint or linear motions).
6. **Refresh**: updates the robot position to fit the 3D environment.
7. **Play/Stop** moves/stops robot.
8. **Program Override**: the percentage of the maximum velocity used.
9. **Velocity**: maximum speed of the robot.
10. Buttons to add **linear**, **home**, or **joint motion** commands to the active program.

### Rotary Table/slider teach pendant




The following settings are available for external axes controlled by the PLC like a Rotary Table or a linear slider:

1. **Connect/Disconnect**: connects or disconnects from the real hardware.
2. **Go Home**: takes the machine to home position.
3. **Refresh**: updates the machine position to fit the 3D environment.
4. **Play/Stop** moves/stops the machine.
5. Buttons to add **home** or **joint motion** commands to the active program.

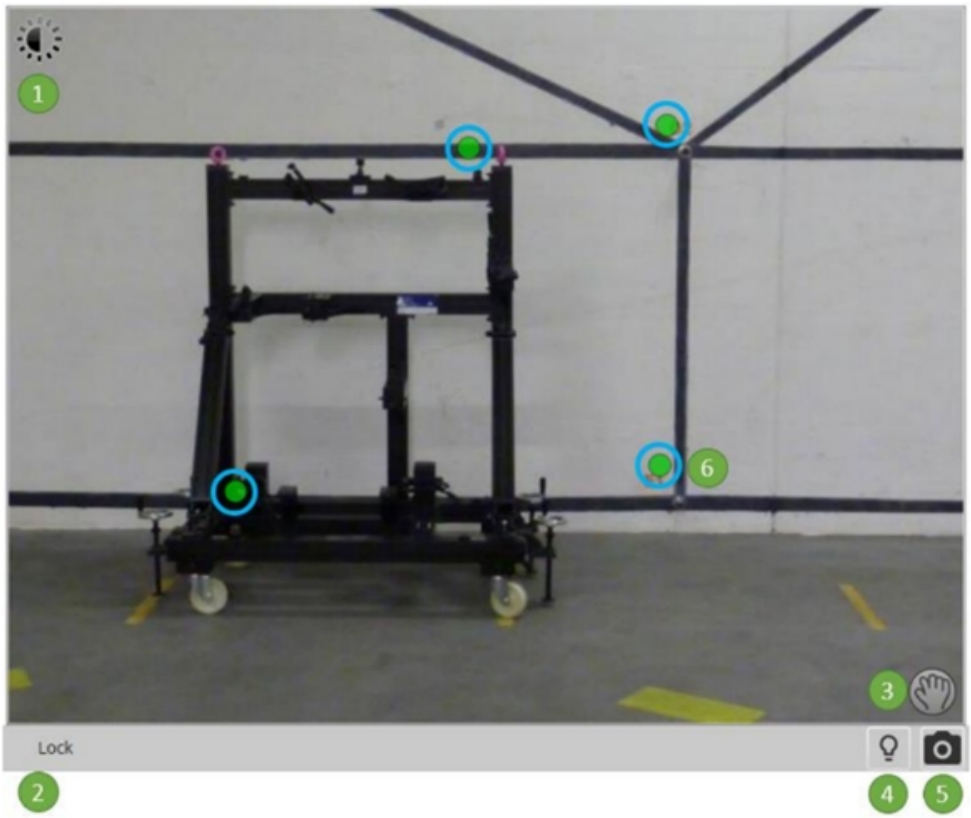
### Tracker teach pendant for Leica only

The following settings are available:

1. **Connect/Disconnect**: connects or disconnects to the tracker.
2. **Lock on** locks on target tool, e.g., AS1 scanner or a reflector.
3. **Delete last point** removes last point measured.

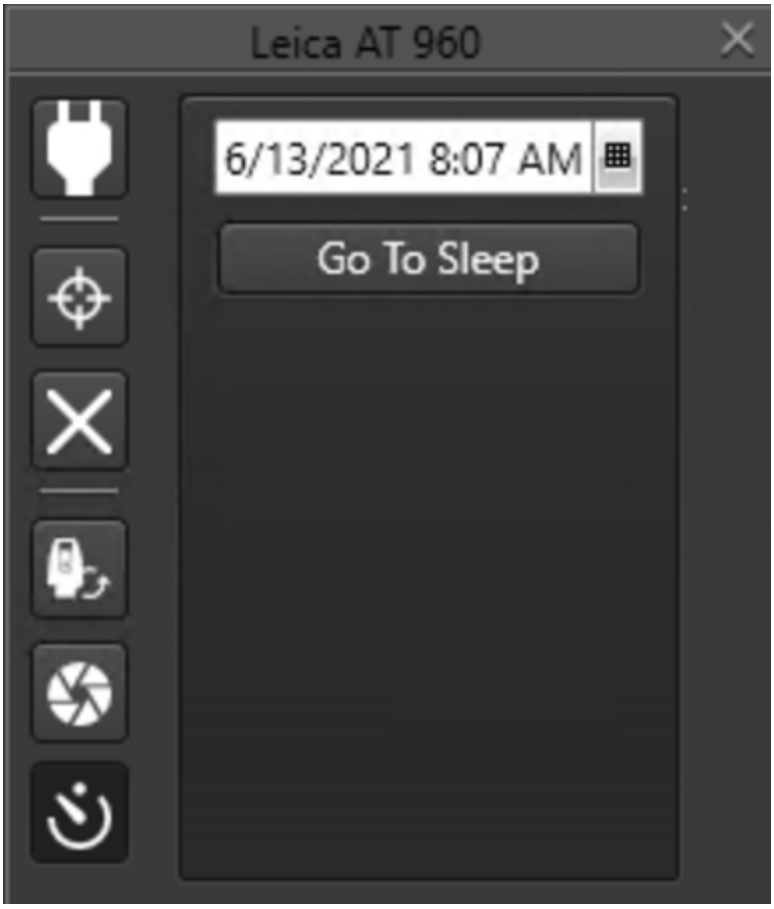
4. **Initialize**: initializes the tracker.
5. **Show GoPro**: click to open OVC of tracker.
    a. User can move tracker joints with the manipulator.
        i. Brightness: user can change brightness
        ii. Enable/disable Lock: enable to lock on target /disable to lock on target.
        iii. Manipulator: use to mode joints of tracker
        iv. Help button: click to get help on commands.
        v. Take snapshot: click to take a snapshot. Immediately a folder window will appear to save it.

Targets will glow in green as seen on the image below.

vi. 


6. **Turn Off**: turns off the laser and set the wake-up time. To be used to set the wake-up time by clicking on the date and time field and put tracker to sleep by

clicking on the Go To Sleep button.




7. **Measure/Stop** starts/stops measurement.
8. **Measure mode**: click on four arrows to access measurement mode options.

   User can select one of the five modes available from the dropdown.

   a. **Scanning**: starts collecting points from a laser scanner.

      i. Min Step Scan To Scan: this value defines the minimum distance between two scan lines.

ii. Point To Point: this value decides the average point distance on the scan line. It refers to the canter of the scanner's measurement range and changes with increasing or decreasing distance to the object. The value can only be changed to 0.075 mm steps.

b. **Stationary**: stationary measurement is a mode of measurement where the tracker spends set time (set up by the measurement profile) taking measurements and then returns a point.

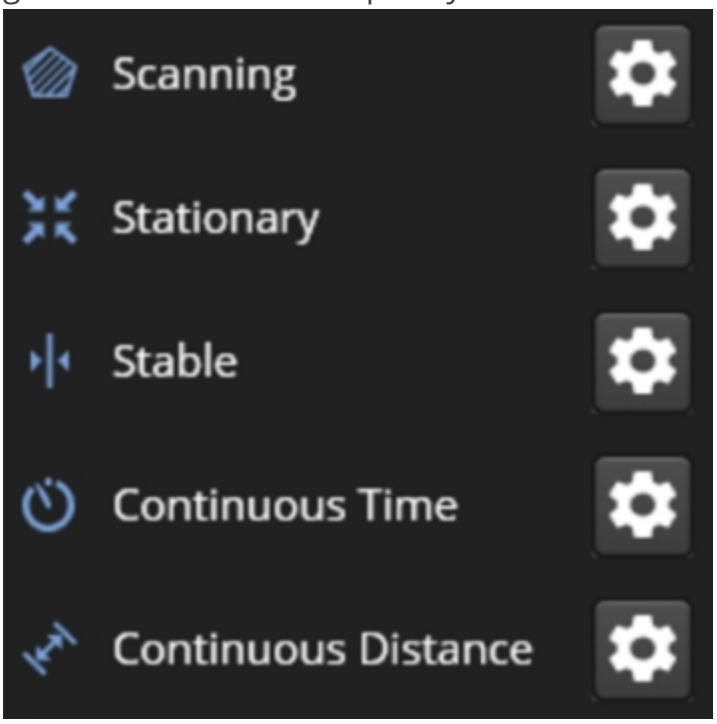
i. Measurement profile: Fast, Standard, Precise according to the following table:

| Profile for measurement type | Use Case | Measurement Time |
|---|---|---|
| Fast | Controlled Environment | 500 ms |
| Standard | AT960 achieves specified accuracy in a controlled environment | 2000 ms – 4000 ms<br>(distance dependent) |
| Precise | Highest accuracy – Sensor checks – Compensations | 10000 ms |
| Individual | Set each reflector with different type | N/A |

| Profile for measurement type | Use Case | Measurement Time |
|---|---|---|
| | individually (Fast Standard or Precise) | |

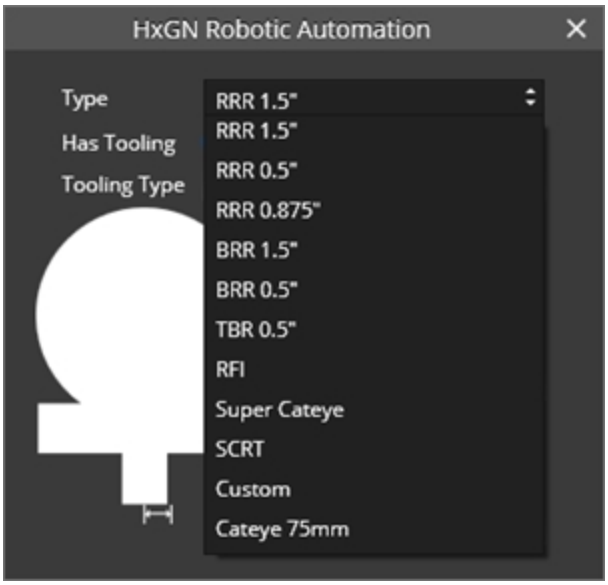
c. **Stable**: stable measurement will trigger measurement shot when conditions on stable time and space are match.
    i. Stable time: change the time probe must be stable to take a measurement shot.
    ii. Stable Space: Change the space probe must be stable to take a measurement shot.
d. **Continuous Time**: continuous time mode takes measurement shots based on the frequency given on continuous frequency.

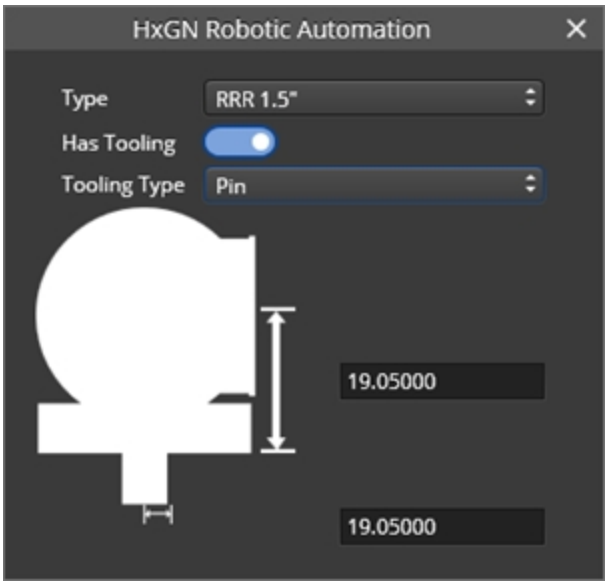
**Continuous Distance**: takes measurement shots based on the distance given on continuous frequency.

e. 


9. **Tool Settings**: shows the tool attached and its options (if any).

   Tool settings are only available if tool attached is a reflector (and the Tracker is locked on that reflector). The user can choose the type of reflector from the list or create a custom one and define if there is a tooling and the tooling type (Flat, Edge, Pin) and enter the correct values of the reflector.



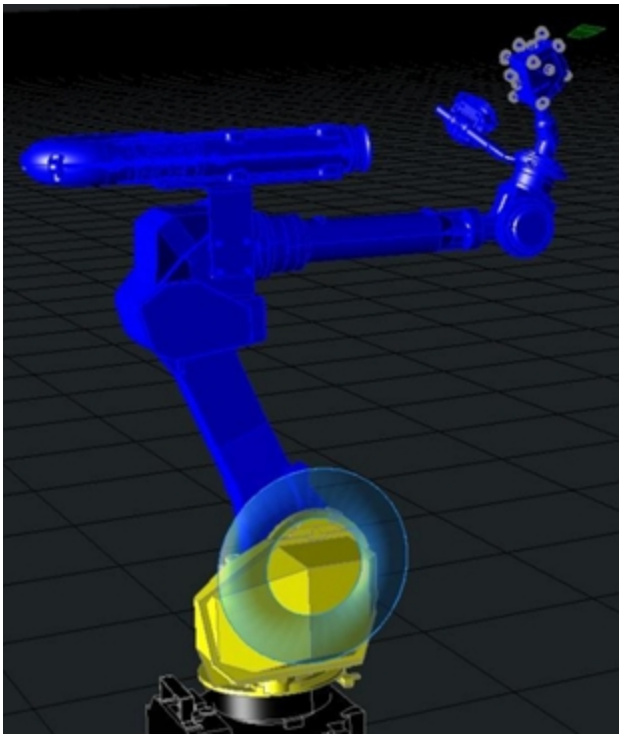



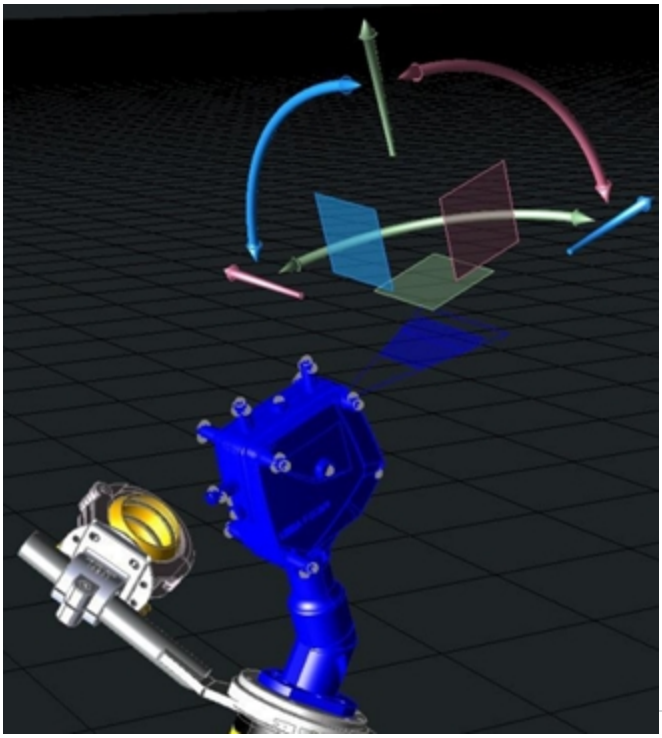

10. Buttons to manage smart features (see chapter Smart Feature - Rotary Table alignment): go to previous, reset current, go to next.

## Moving a cell item dragging it in the 3D environment

To move the machines using the 3D environment you can click and drag the different joints (either in general or using the drag wheel that appears). You can also

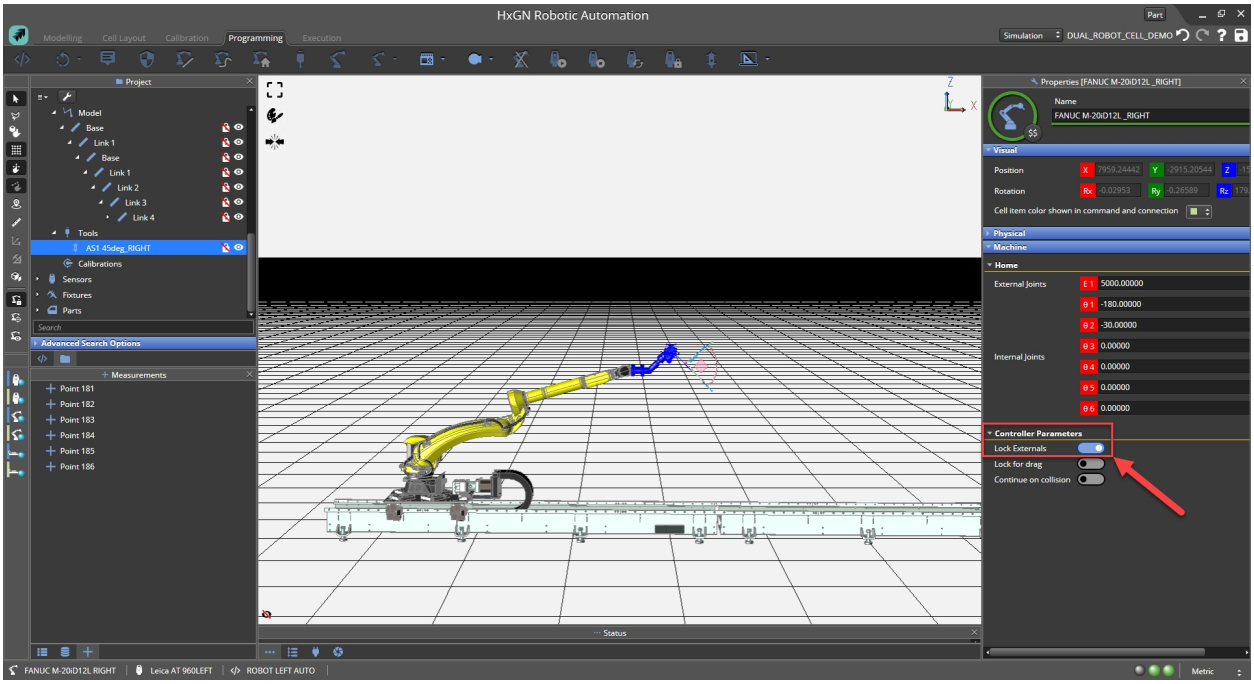
move the tool itself via the same method or by using the gizmo that appear when choosing it. By moving the tool, the rest of the joint will adjust automatically according to the machine kinematics to reach the desired position.

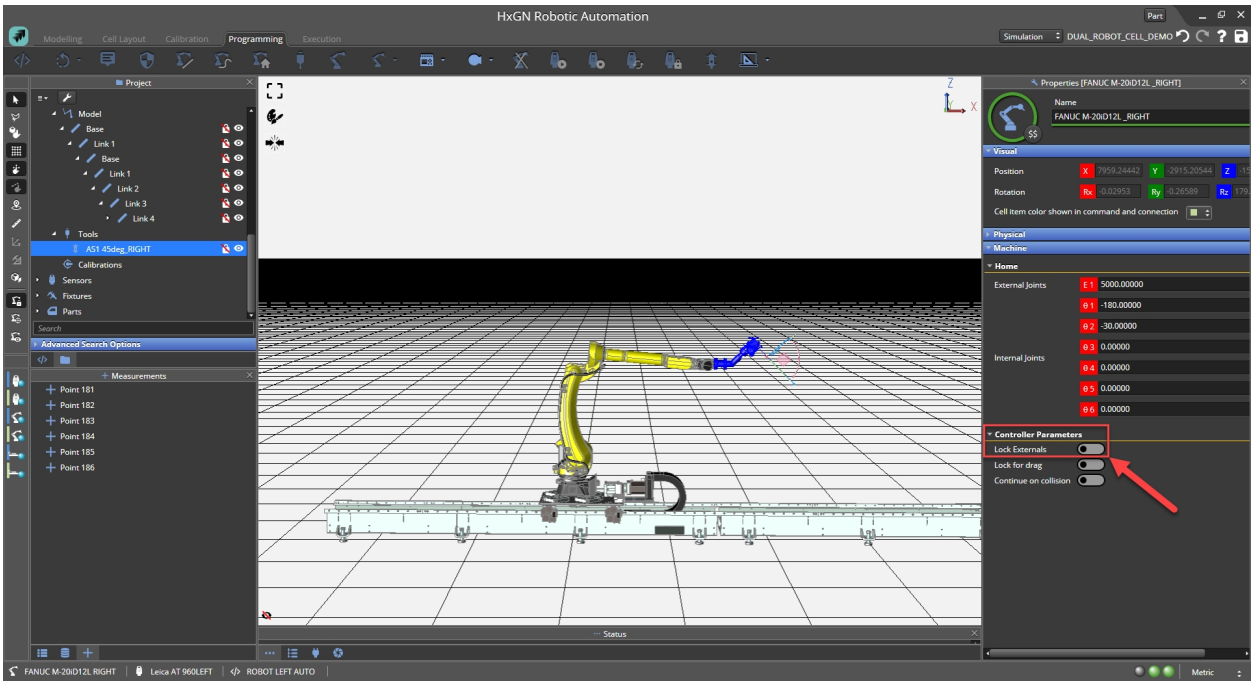




If a robot is mounted on a slider (see chapter Robot on slider for modelling information), it is possible to 3D drag the tool gizmo considering the overall 7 axes kinematics of the robot including the external axis (slider) or to lock it.

Within **machine Properties**, if the **Lock Externals** toggle switch is enabled, when the robot is moved in the scene, the external axis will be locked so it will not move, if it is disabled also the external axis will be moved to bring the robot to the desired position.





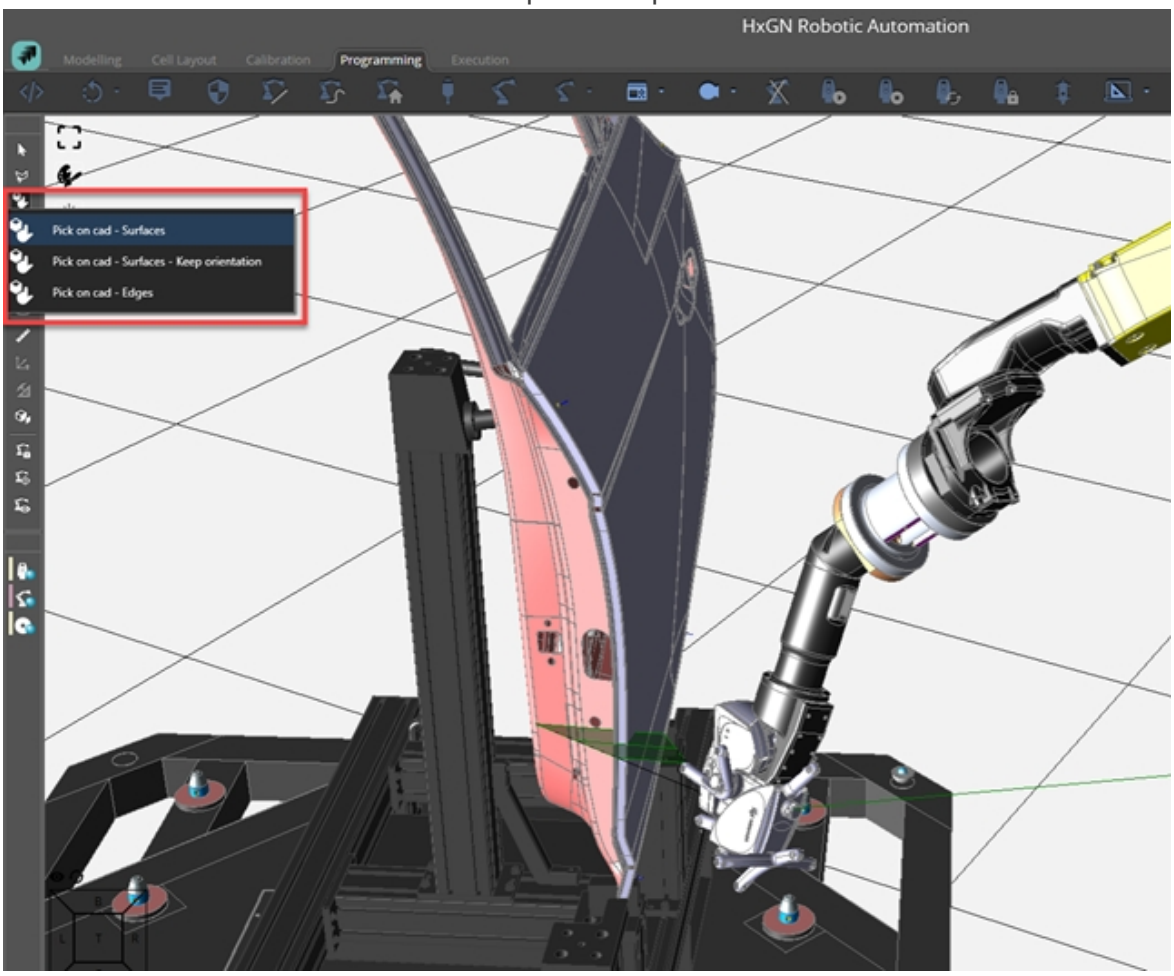

## Pick on CAD

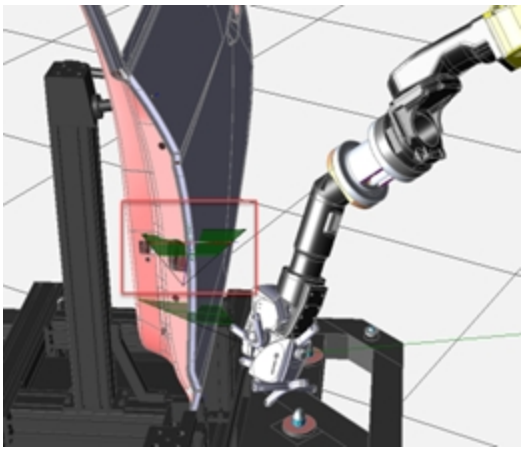
Pick on CAD is a special mode that allows the user to select a point either on the surface or on the edge of a CAD model by simply clicking on it.

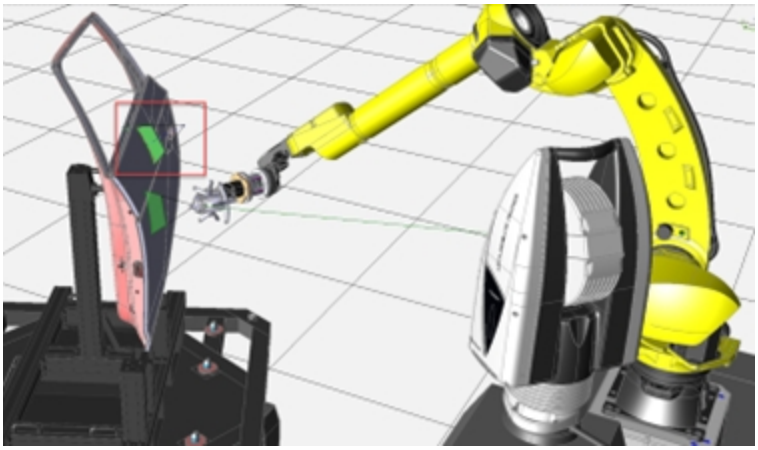
Pick on CAD offers three ways of picking a point on the CAD model:

1. **Pick on CAD** – Surfaces allows to visualize the tool TCP on the CAD surface point and click to move the robot in the specified position.
2. **Pick on CAD – Surfaces** – Keep orientation: same as above but the target orientation is set to the current tool orientation.

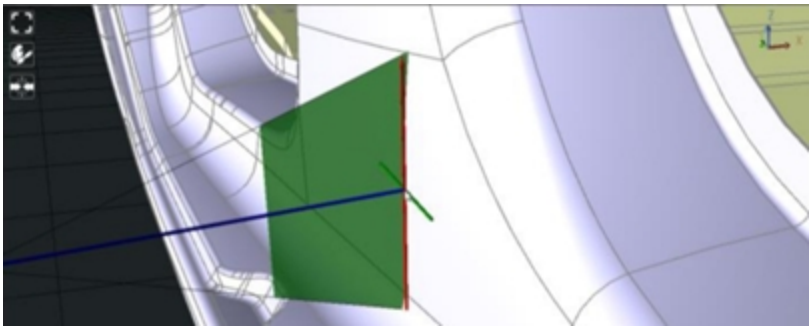
**Pick on CAD – Edges** allows to visualize the tool TCP on the CAD surface point and click to move the robot in the specified position.

3. 

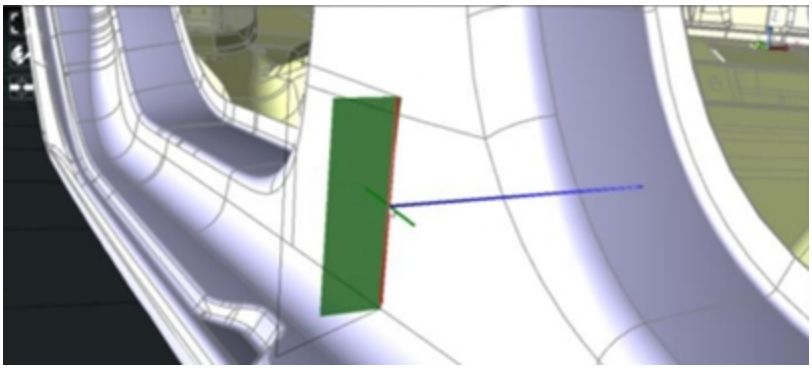




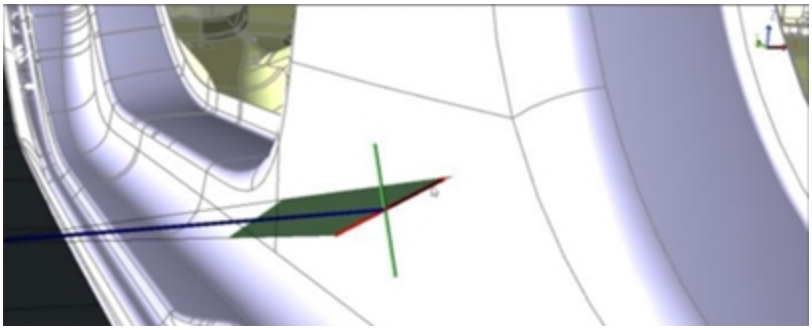
When activated, the scanning range will appear perpendicular to the surface.



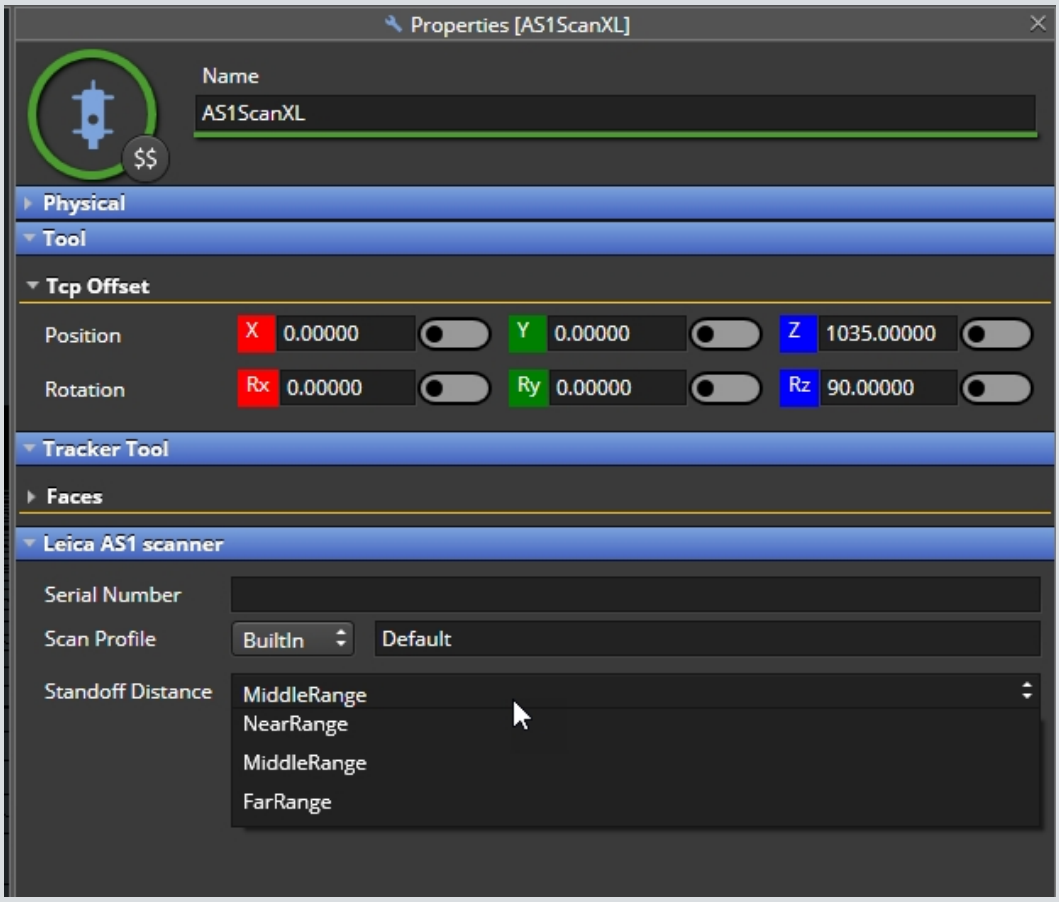
To flip the normal [Alt] button needs to be pressed.



To rotate direction [Shift] button needs to be pressed and mouse dragged.



| | |
|---|---|
| | For some scanners (AS1XL or ZG Scanner) the distance of the pick on CAD will depend on the selecting working range. This can be set in the properties of the scanner. |

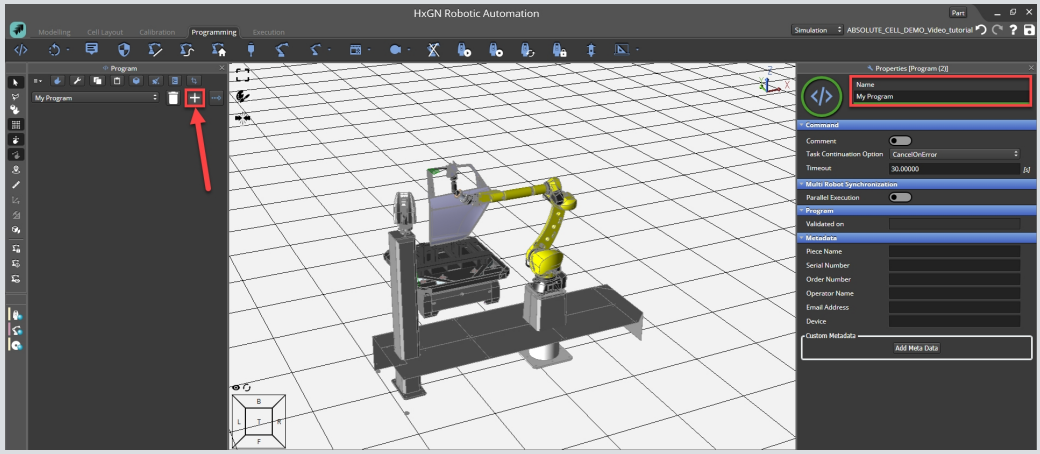



## Adding commands

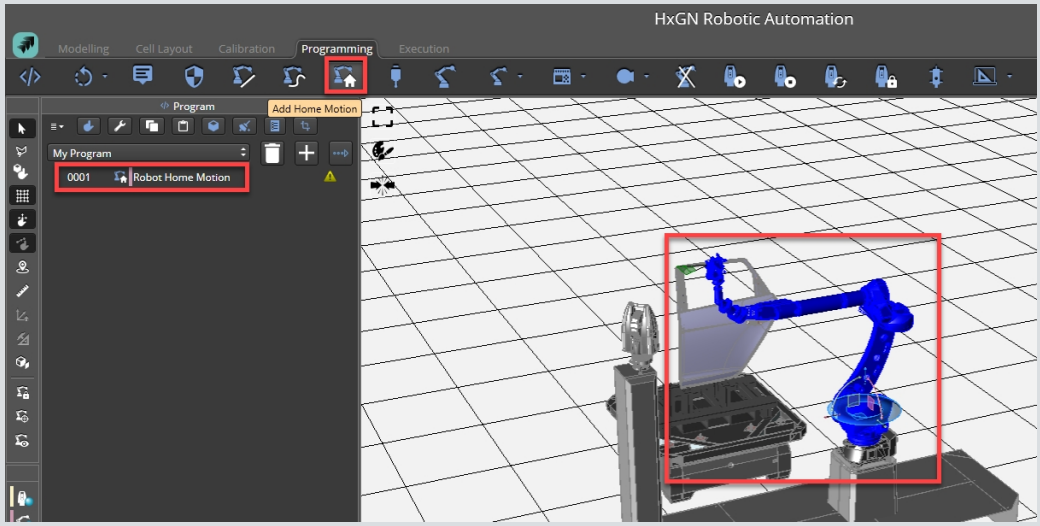
After choosing the position of the robot you can start adding commands from the upper toolbar and begin creating your manual program. When choosing the command notice the right-side window displaying the commands properties in case of any changes you need to make. All available commands are explained in detail in chapter supported commands In the following example, a basic program is shown where the user creates a simple program to scan a feature by adding manually the necessary commands from the ribbon toolbar.

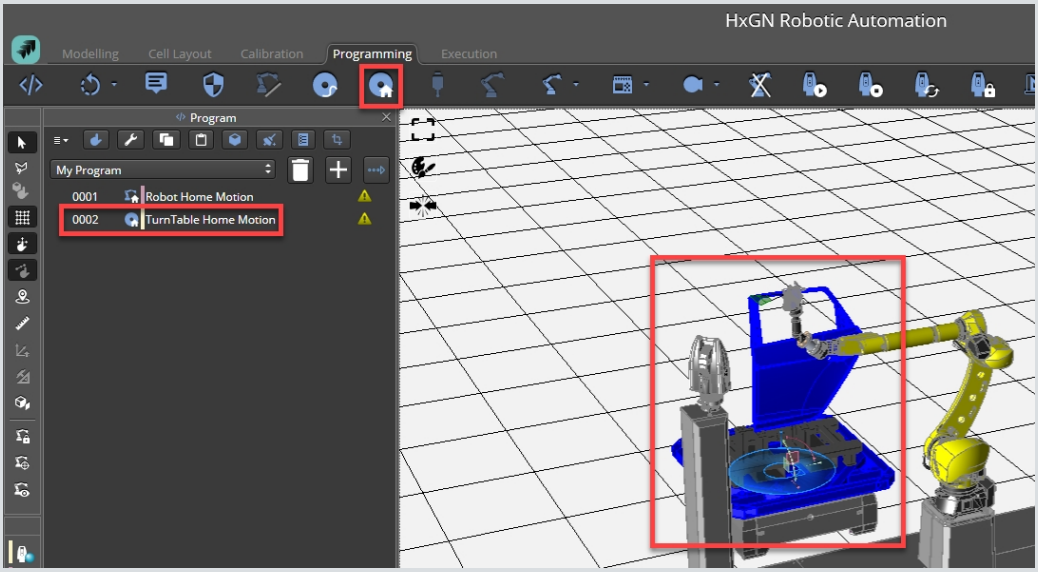
Example: Single robot cell manual programming - Leica tracker and scanner

First, it is needed to click on the plus button, create a new program, and call it e.g. “My Program.”

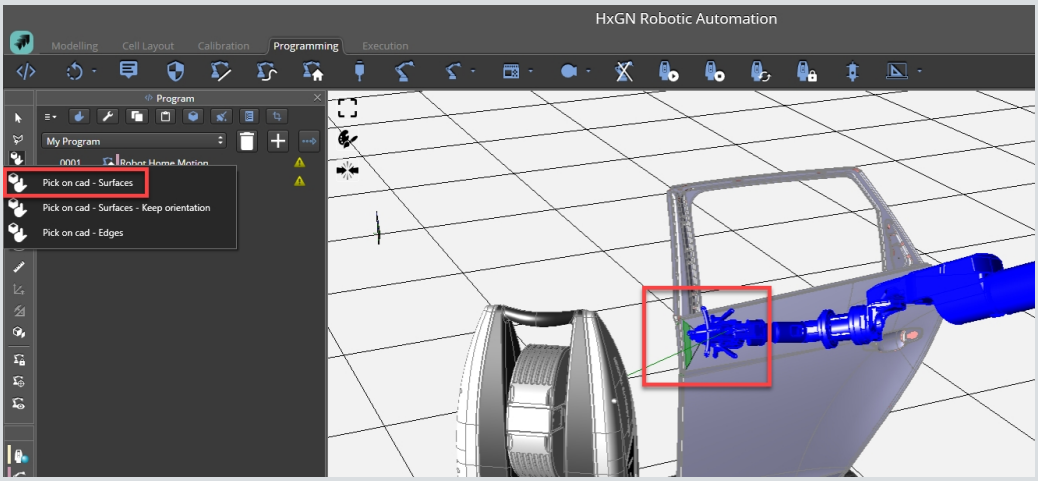
1. 

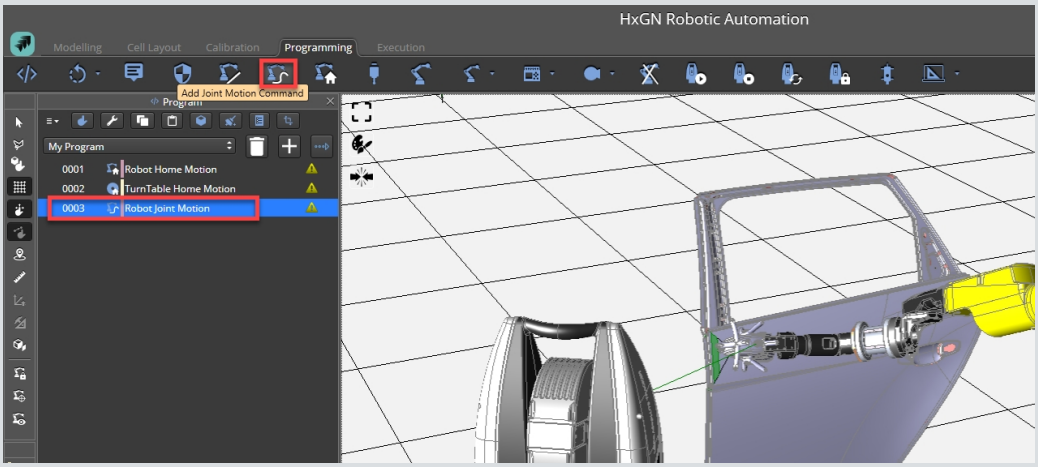

2. Then it is needed to add homing motions for the robot and the Rotary Table by selecting them and adding the relative homing commands.



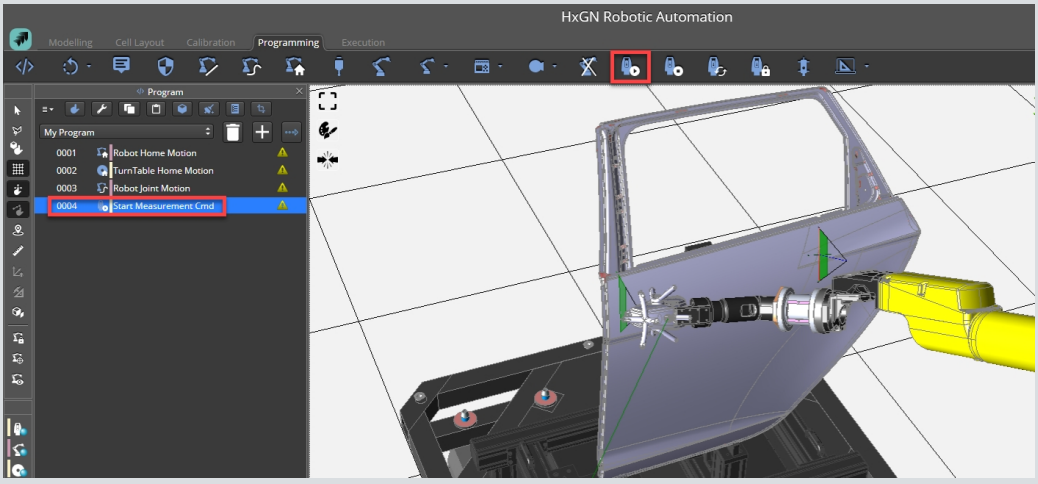



3. At this point, it is possible to move the robot e.g., using the Pick on CAD functionality to a specific point on the CAD surface where the scan will start and then add a joint motion command to take the robot to that position.

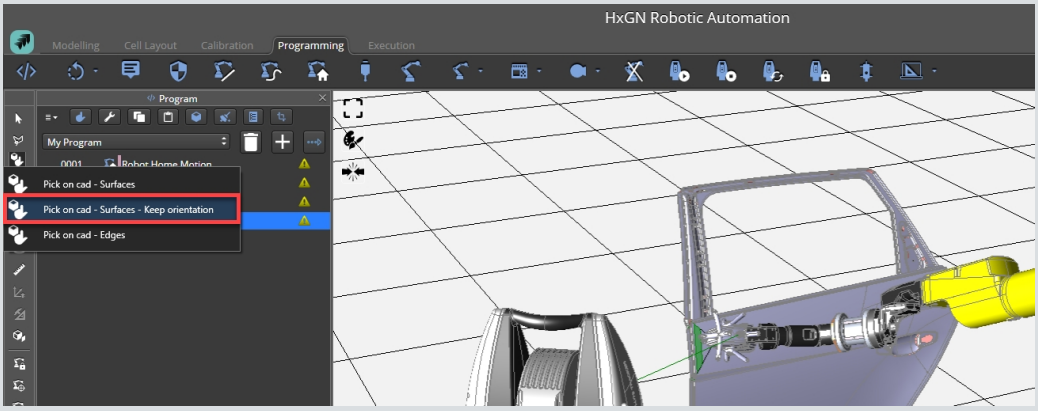


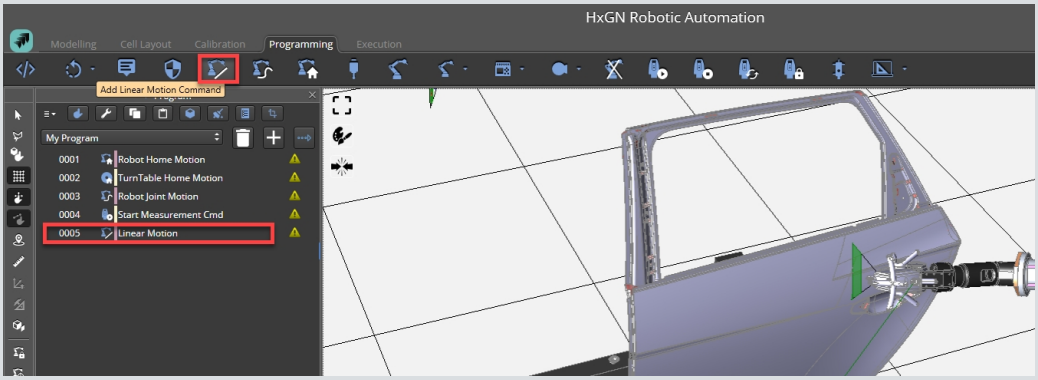



Now it is possible to add a Start Measurement command to start the scanning from this position.

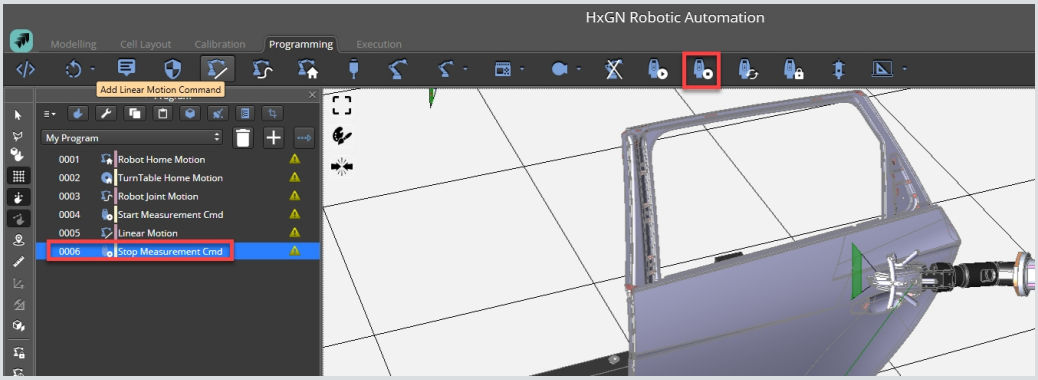
4. 

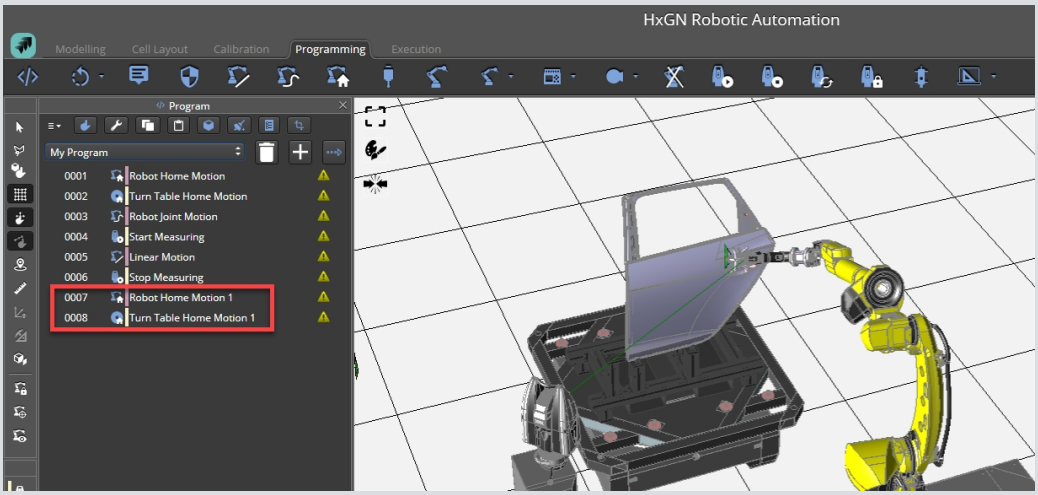

5. Then using the Pick on CAD – Keep orientation functionality is then possible to specify the next position that the robot will reach using a linear movement while scanning. After clicking on the next position on the CAD, a linear motion command to can be added to the program.




6. Finally, a Stop Measurement command can be added to stop the scanning, as well as homing motions to take the robot and the Rotary Table to their starting positions.





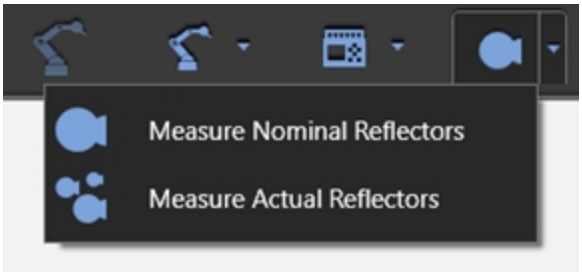

# Programming - Leica

## Supported Commands

### Compensation (HxGN Robotic Automation Compensation)

The compensation commands are usually needed in the following cell configurations which are examples of the relationship change between tracker and part:

1. Cell including one or more Rotary Tables, with reflectors mounted on the Rotary Table.
2. Cell including one or more trackers mounted on a slider, with reflectors mounted on some static reference.
3. Cell including both cases above simultaneously.




In general, the compensation flow is composed by two steps: a nominal process measurement followed by an actual process measurement for which dedicated commands are available.

See chapter HxGN Robotic Automation Compensation for further information on compensation.

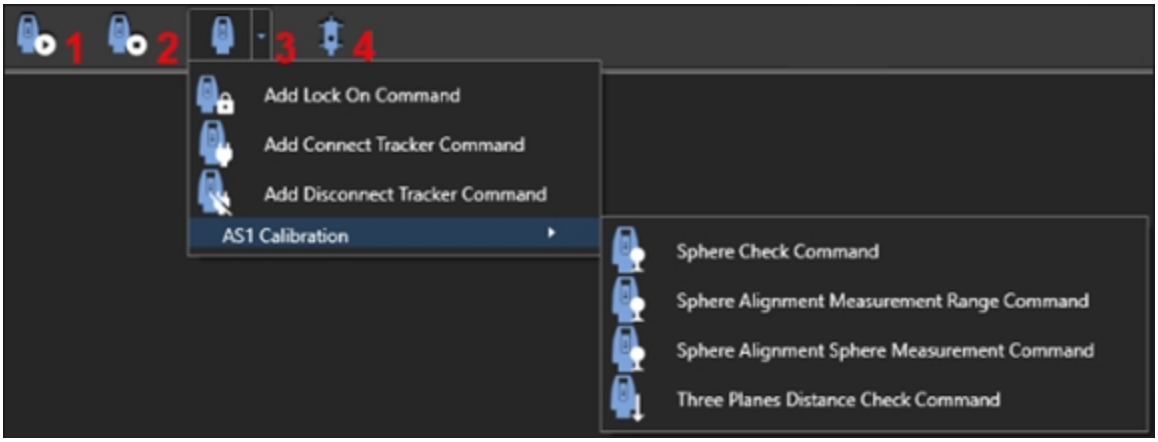
### Compensation (PolyWorks Reference points)

An alternative to use HxGN Robotic Automation nominals and actuals, if the Metrology application is PolyWorks, Reference points will be also available.

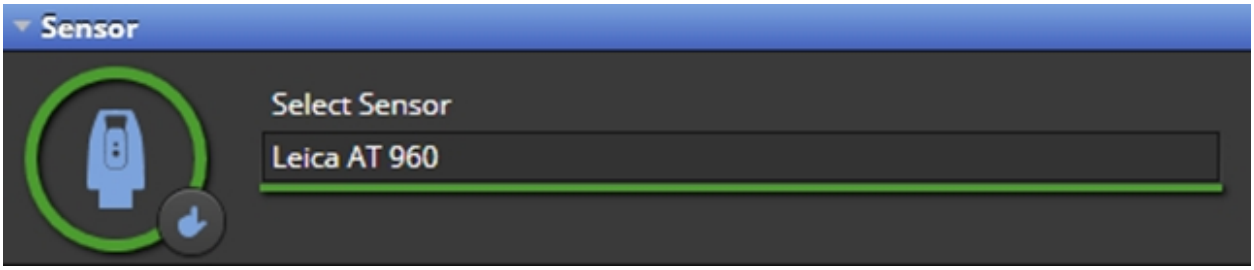
See chapter PolyWorks reference points for further information on PolyWorks .reference points

### Sensors

In this area the user can select the commands to control the Trackers (e.g., AT960) and Laser scanners in use (e.g., AS1) in the project.



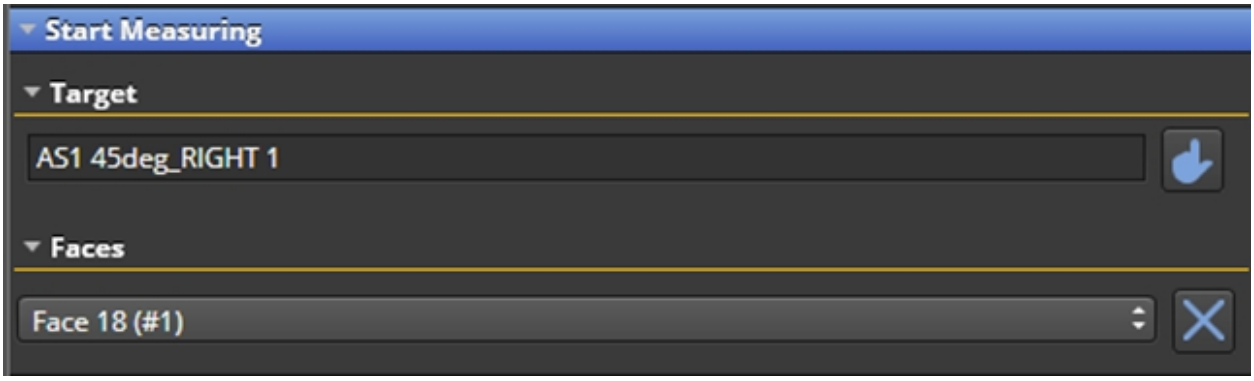
All tracker commands (start/stop measurement, lock on, etc.) must specify the sensor to be used in the **Sensor** section of the properties (e.g., AT960).




Depending on the command, they have specific parameters to define the behavior as explained below.

#### Start Measurement - AT960

In the Start Measurement command the tool to scan must be specified under *Start Measuring | Target section*. There is also the possibility for the user to force the face number where the tracker must be locked to start scanning.

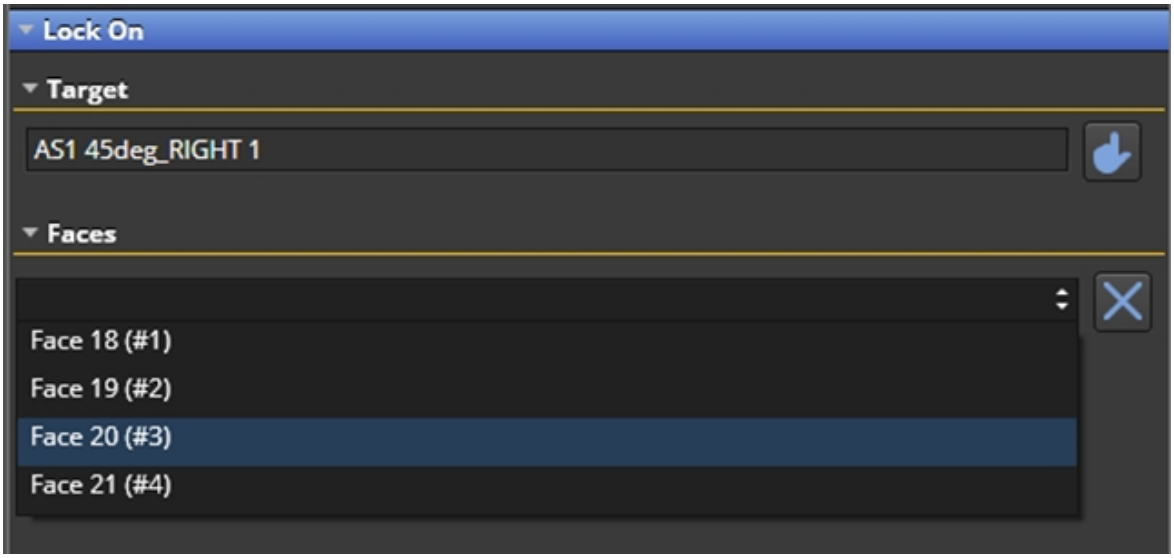



#### Stop Measurement - AT960

In the Stop Measurement there is no need to specify any properties as the stop command will refer to the earlier Start Measurement command settings.

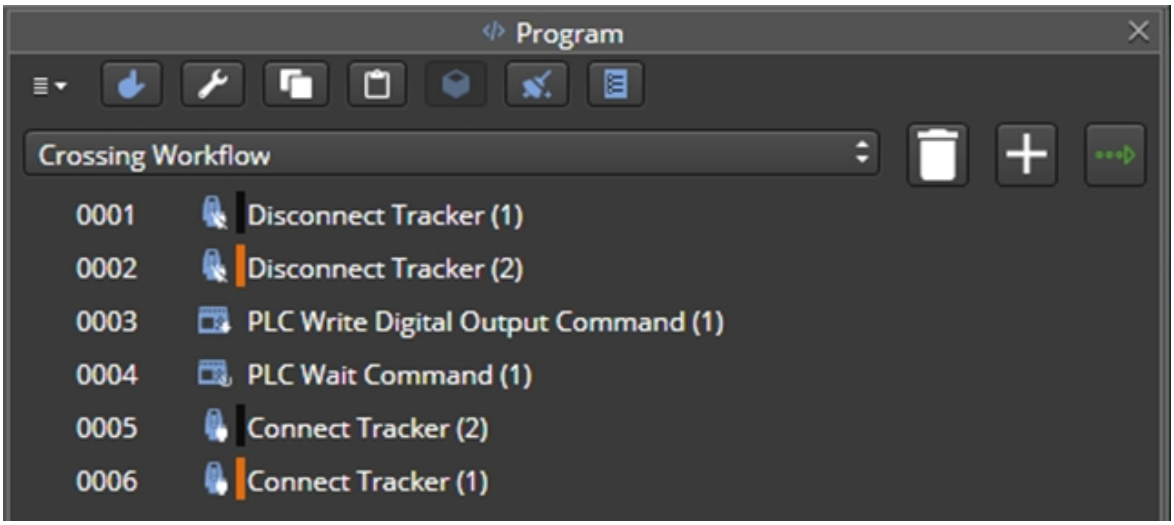
#### Other Tracker Commands

- **Lock On** command can be used to force the tracker to lock on a specific target that can be either the face (to be defined under Lock On | Faces using a drop-down menu) of a scanner (to be defined under *Lock On* | *Target*) or a reflector.




- **Connect Tracker** command is used to connect the selected tracker using the IP configured in RDS Application
- **Disconnect Tracker** command is used to disconnect the selected tracker from the IP configured in RDS Application

  These two commands are used together with the PLC Write and Wait commands to cross the trackers with the scanners in a Dual robot cell setup following the next working flow:




- **Read Sensor Data** command reads the data from RDS and stores it into a variable. See section Reading_Sensor_data
- Inside the **AS1 Calibration** title, four new commands can be found related to the checking and alignment of the AS1 sensors.

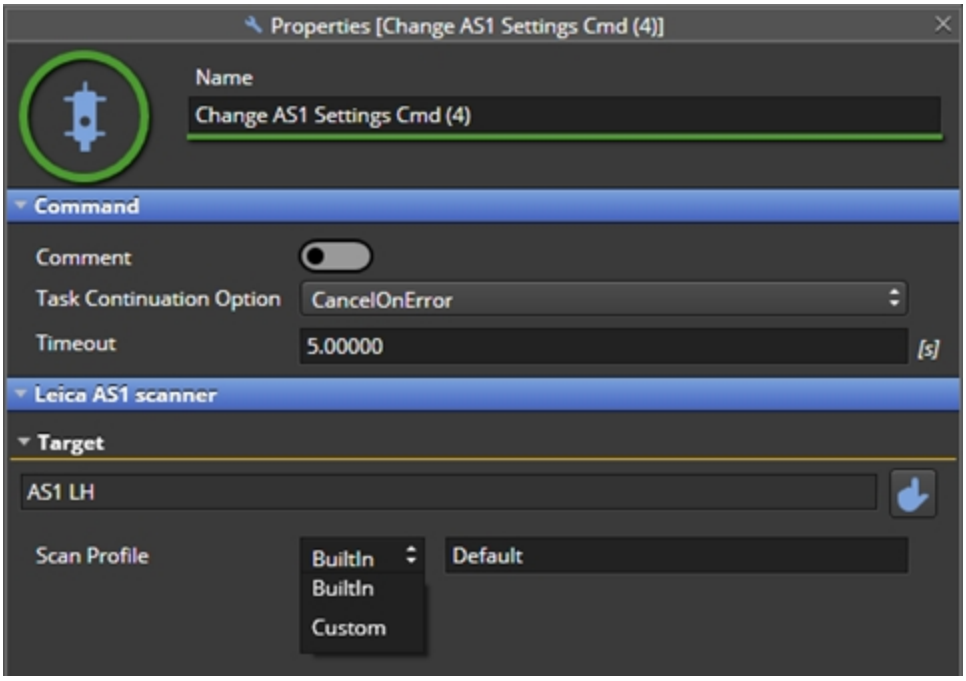
  **Sphere Check Command**. Verifies the current scanline to probe face alignment. Calibrated sphere is needed.
- **Sphere Alignment - Range Measurement** Command. Calibrates only the measurement range of the scanner.
- **Sphere Alignment - Sphere Measurement** Command. Aligns the scan-line to the probe face of the AP21.
- **Three Planes Distance Check** Command. Verifies if the measurement range calibration is good.

#### Change AS1 Settings

A Change AS1 Settings command ( ) can be used to change the scanning profile of a laser scanner by specifying the Scan Profile in the corresponding fields.

Users need to define the full profile name by choosing first the prefix (BuiltIn/Custom) using the drop-down menu and then adding the profile name suffix in the text input box. If user leaves it empty, “BuiltIn.Default” profile will be used automatically as shown in the picture below.

The AS1 settings change will be applied at the next **Start Measurement** command.

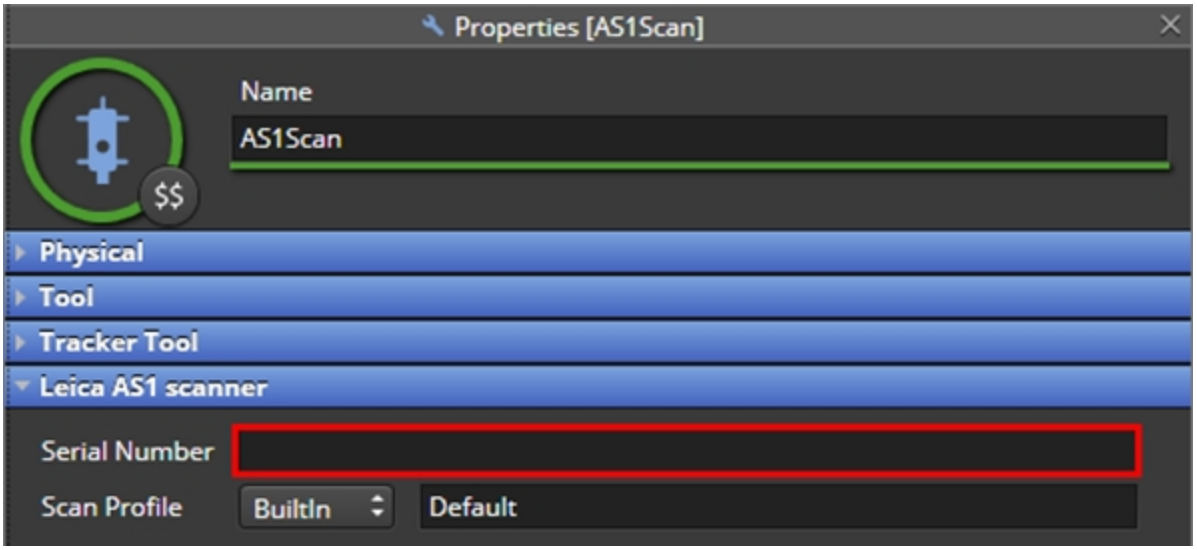



| | |
|---|---|
| ⚠ | When using a Custom Scan Profile, the profile must be set also in RDS, and it must match exactly the one specified in the Scan Profile property. |

| | |
|---|---|
| ⓘ | AS1 XL built in options are Default and Sensitive, while for AS1 the only built in option is Default. In these cases, the profiles must be defined as BuiltIn.Default and BuiltIn.Sensitive (only for AS1 XL) in the Scan profile property. |

| | |
|---|---|
| | The user can define as many Custom profiles as needed and call them using the Custom prefix (e.g., Custom.MyProfile).<br>In case a “Sensitive” profile is defined for AS1, it must be called as Custom.Sensitive. |

| | |
|---|---|
| ⚠ | If there are more then one tools defined for a robot, it is mandatory to define the serial number for each tool |




## Programming

### Compensation

The idea of the compensation approach is to make it as agnostic as possible to the cell setup. This is the generic compensation used in RA to compensate for relative motions of cell items causing the COP collected in different scans to be misaligned with each other whenever the relationship between tracker and part is changed.

This situation can happen in two cases:

1. Dynamic reflector scenario: the part moves, so the reflectors change position accordingly (e.g., reflectors linked to the part). See chapter Notable use cases.htm for a notable example.
2. Static reflector scenario: the tracker moves from its earlier position (e.g., tracker moving on a slider). See chapter Notable use cases.htm for a notable example.

HxGN Robotic Automation Compensation has been built with the following flow:

- The reflectors are measured at home position (**nominal reflectors**): this will allow the tracker to create an internal compensation map and adjust for any changes in the relationship between the tracker and the part
- Weather there is a change in the relationship between the tracker and the part, e.g., either the Rotary Table or the slider move, this requires the tracker to understand what was changed compared to its earlier compensation table with respect to the current Rotary Table or tracker position.
- The reflectors are measured in the new position (**actual reflectors**): this will allow the tracker to update the compensation table that it was created with the nominal reflector positions before and the newly measured actual reflector positions and apply a best fit alignment.

It is always recommended to scan all the reflectors in the nominal position. It is not possible to measure all the reflectors from a single position, it is needed to move the Rotary Table (or the slider) and perform another nominal measurement to collect the measurement of all the reflectors and include them into the compensation map together with the reflectors already measured.

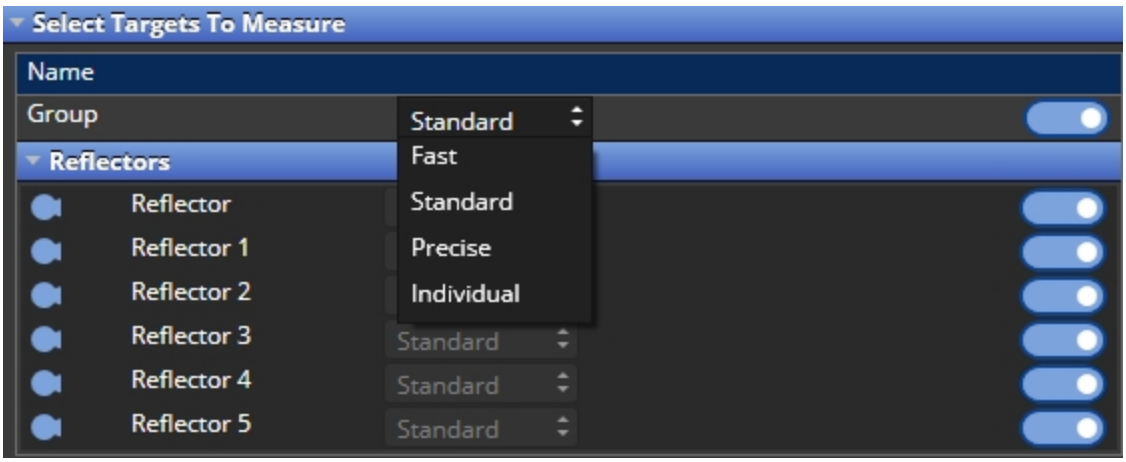
After each move, either measuring nominal or actual reflectors, at least three of the original nominal reflector positions need to be included to reduce the best-fit alignment error at minimum. The more moves are used to measure the reflector positions, the more noise will be introduced in the best-fit alignment process.

Path planning algorithm already embeds the above logic and create programs that include this compensation flow.

PolyWorks Compensation has been built with the following flow: See chapterReference points

**Type of reflector measurement**

Both in nominal, actual and reference point measurement, different type of measurements of those reflectors can be selected.




| Profile for measurement type | Use Case | Measurement Time |
|---|---|---|
| Fast | Controlled Environment | 500 ms |
| Standard | AT960 achieves specified accuracy in a controlled environment | 2000 ms – 4000 ms<br>(distance dependent) |

| Profile for measurement type | Use Case | Measurement Time |
|---|---|---|
| Precise | Highest accuracy – Sensor checks – Compensations | 10000 ms |
| Individual | Set each reflector with different type individually (Fast Standard or Precise) | N/A |

**Measure Nominal Reflectors workflow**

1. The home position is first considered to measure the reflectors as the first pose.
2. The user adds a **Measure Nominal Reflectors** command and HxGN Robotic Automation tries to measure all the nominal reflector positions. If in the home position it is not possible to measure enough reflectors, a warning is shown during the program execution in the status window. This would imply that the user needs to create a new position where the reflector is visible or move the reflector to be able to be measured from the home position.
3. If all reflectors can be measured in the home position, then the overall program will be made of the following instructions:
    a. Move the Rotary Table or Slider to Home
    b. Measure Nominal Reflectors
4. If in the home position only some of the reflectors can be measured, a set of Rotary Tables or slider positions need to be added in the program to measure all reflectors so the program will have a sequence of instructions to execute more Rotary Table or slider positions and measure nominal reflector positions in the new Rotary Table or slider position

---

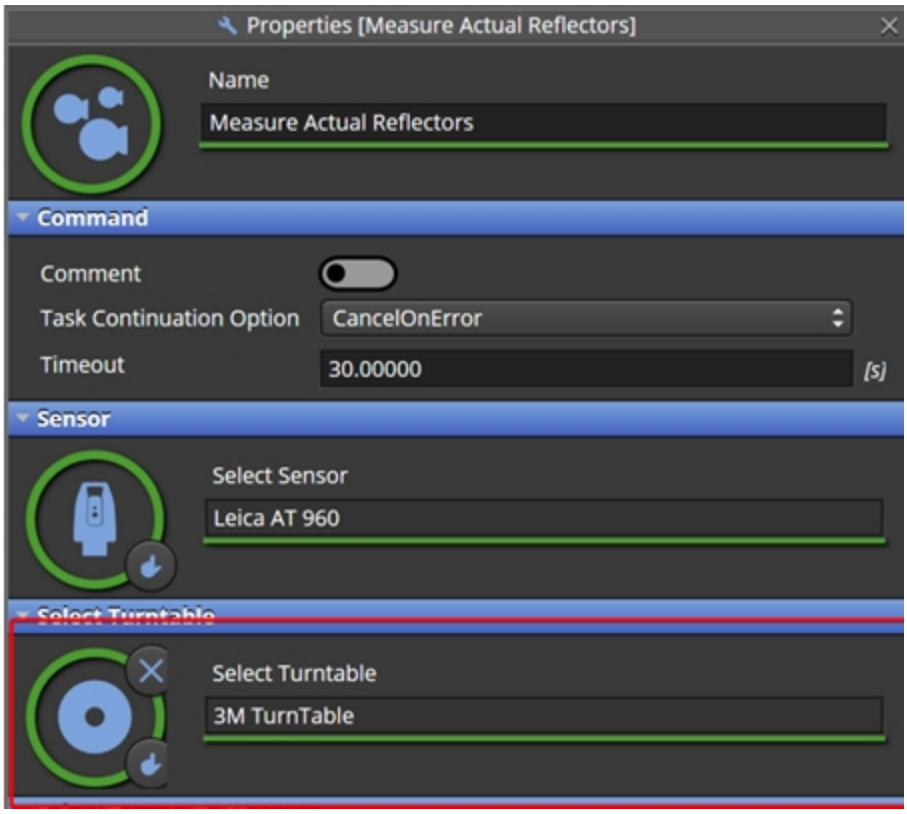
5. The amount of added Rotary Table or slider position depends on the specific situation, e.g., if three pose are defined either for a Rotary Table or a Slider, the sequence of instruction will have the following instructions:
    a. Move the Rotary Table or Slider to Home
    b. Measure Nominal Reflectors
    c. Move the Rotary Table or Slider to Pose 1
    d. Measure Nominal Reflectors
    e. Move the Rotary Table or Slider to Pose 2
    f. Measure Nominal Reflectors
    g. Move the Rotary Table or Slider to Pose 3
    h. Measure Nominal Reflectors

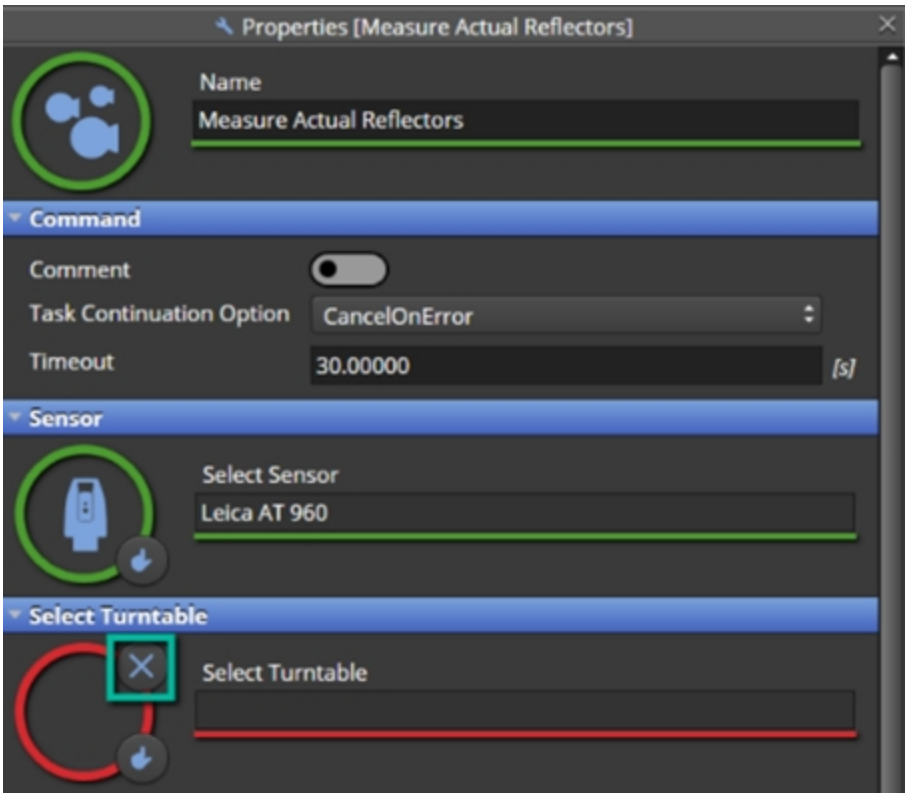
Notable use cases of the compensation usage are described in chapter Compensation

#### Measure Actual Reflectors workflow

Before starting any measurement routine, if the reflectors have moved compared to the position they had when measured by a **Measure Nominal Reflectors** command, it is needed to measure the actual position of at least three of them. For this purpose, the **Measure Actual Reflectors** command must be used with the same considerations described in the earlier chapter.

Among the properties it is possible to select a Rotary Table where the reflectors are mounted on. Otherwise, the field must be kept empty (e.g., when the Rotary Table does not move, and the measure actual command is needed because of a tracker movement instead).


Properties [Measure Actual Reflectors]
Name
Measure Actual Reflectors
Command
Comment
Task Continuation Option
CancelOnError
Timeout
30.00000
[s]
Sensor
Select Sensor
Leica AT 960
Select Turntable
Select Turntable
3M TurnTable




It is always recommended to measure as many reflector positions as possible in every Rotary Table or slider position to have the optimal best-fit alignment process.

#### Evaluating the Quality of the Compensation

The assessment of the quality of the compensation is done at two stages:

1. If the program has several Measure Nominal Reflectors commands, it assesses the quality of the newly measured reflectors with respect to the reflectors measured in the earlier Measure Nominal Reflectors commands.
2. Assesses the quality of the reflectors on each Measure Actual Reflectors command.

At each stage, the outcome of the assessment is either that a particular reflector needs to be disabled from the command (Measure Nominal Reflectors command or

Measure Actual Reflectors command) because it may add too much noise during measurement or discard the command and measure at a different pose.

#### Measure Nominal requirements

The first Measure Nominal Reflectors command must measure at least three reflectors.

- The first Measure Nominal Reflectors command must measure at least three reflectors.
- The next Measure Nominal Reflectors command must measure at least three reflectors in common with the first Measure Nominal Reflectors command.
- It is suggested to use as few Measure Nominal Reflectors command as possible to measure as many reflectors as possible.

The procedure is to execute the command to assess the quality of compensation data.

Once executed, the first Measure Nominal Reflectors command will not show any information about the quality of the best fit (is not computed on this case) and must be ignored.

The rest of Measure Nominal Reflectors commands will show a chart with information about the result of the best fit.

This chart must be used to assess the quality of the measured data as explained in chapter Understanding_alignment_data

Properties [Measure Nominal Reflectors 1]
Name
Measure Nominal Reflectors 1
Command
Comment
Task Continuation Option
CancelOnError
Timeout
30.00000
(s)
Sensor
Select Sensor
Leica AT 961
Select Targets To Measure
Name
TRACKER1
TRACKER1
Reflector 6
Reflector 7
Reflector 8
Reflector 9
Reflector 10
Reflector 11
Reflector 12
TRACKER2
TRACKER2
Reflector 13
0.0003
0.0003
0.0002
0.0002
0.0001
5.00E-005
0.0000
0.0002
0.0002
0.0003
0.0002
0.0000
0.0000
1
2
Reflector 6
Reflector 7
Reflector 8
Reflector 9
0.0002
Average
0.0002
Rms
0.0003
Max
3

Let's evaluate the following areas

1. **Region 1**: given a particular reflector, this area of the graph shows the absolute error (distance) between its measured position at the first Measure Nominal Reflectors command pose and the estimated position resulting from applying the computed compensation to the reflector measured at the current Measure Nominals Reflectors command pose.
2. **Region 2**: this area shows the new measured reflectors that will be available for future Measure Nominal Reflectors command. No error can be computed on them because they are not used for best fit computation.
3. **Region 3**: this area gives some statistic summary of the best fit computation.
    a. Average between the absolute errors of all the reflectors of Region 1.
    b. RMS between values of Region 1 values.
    c. Max absolute error (distance) of Region 1 values.

If the result doesn't fit with the expected quality, there are two main actions that can help to improve the quality of the compensation:

1. The absolute error of one of the measurements is significantly greater than the Average/RMS: in this case it is possible to disable the measurement of this reflector.

2. High Average/RMS values but there is not a clear measurement with a significantly greater error than the Average/RMS: in this case it is recommended to try another position.

Once the corrective actions are performed it is necessary to execute and assess the quality again.

#### Customizing RMS tolerance

To customize the default RMS tolerances on the actual measurement, modify the values:

- For Robotic Automation:
  - RMSError node (C:\ProgramData\Hexagon\RA\Config\RAconfig.xml)

```
<ConfigSetting Name="RMSError" Value="8" DefaultValue="8" DisplayName="RMS compensation error threshold"
Description="RMS compensation error threshold" TypeName="System.Double" />
```

- For Execution Manager
  - RmsErrorLimit (C:\ProgramData\Hexagon\HxGN.RA.Gen-

ericHost\ExecutionManagerSettings.json)

```
{
  "Compensation": {
    "RmsWarningLimit": 5,
    "RmsErrorLimit": 8
```

#### Measure Actual requirements

Each Measure Actual instruction should measure at least three reflectors in common with the measured reflectors with Measure Nominal instructions. It is recommended to try to find positions where there are as much reflectors as possible. On each Measure Actual command will show a chart with information about the result of the best fit. You can use this data to assess the quality of the measured data as explained in chapter Understanding_alignment_data

#### Measure Reference points workflow

Reference points are used in PolyWorks to compensate the cloud of point as it is done using internal HxGN Robotic Automation compensation.

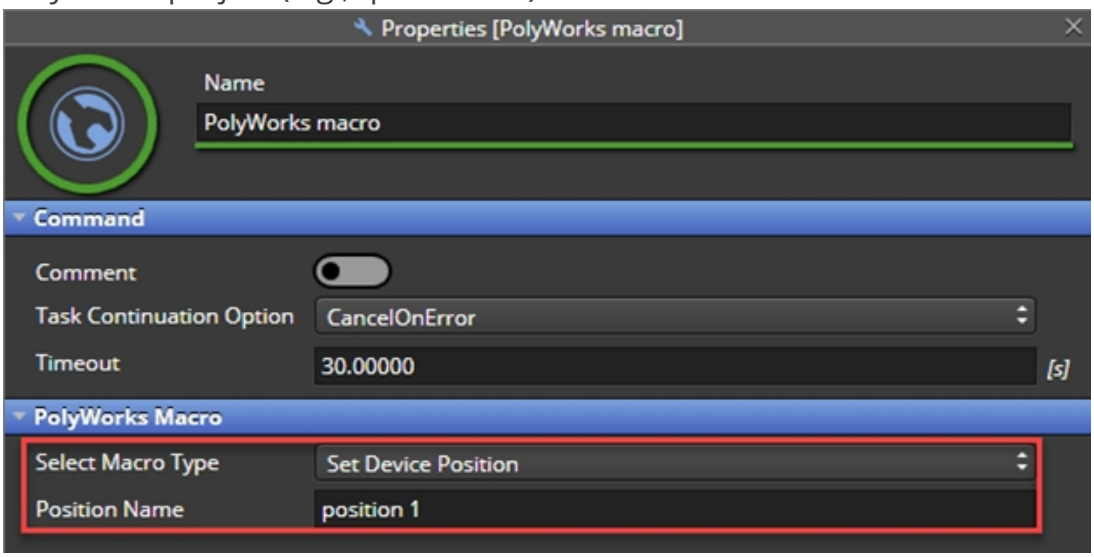
The functionality is similar to PC-DMIS Static points, used for part alignment workflow with PC-DMIS (PC-DMIS Static Point measurement). Reference points workflow with PolyWorks consists of HxGN Robotic Automation sending reflector positions to PolyWorks so it can either use them to align the various COPs received during the scans or to check part alignment.

| | |
|---|---|
| ⚠ | When using reference points, COP will appear in a wrong place in RA because data compensation is managed in PolyWorks side.<br>To set the reference points, it is first necessary to: |

1. Create device positions (if they do not already exist) in the PolyWorks project to be used either using Macro programming directly in PolyWorks or using an

Execute Macro command within the HxGN Robotic Automation program.

2. Set the active device position by using the predefined Set Device Position macro available in HxGN Robotic Automation. This predefined macro allows the software to automatically ensure the COP name is not re-used in various positions:
    a. add an **Execute PolyWorks Macro** command
    b. select **Set Device Position** as Macro Type from the relative drop-down menu

       set the device position name (case sensitive) as defined in the relative PolyWorks project (e.g., “position 1”)
    c. 


The workflow using Reference points within an HxGN Robotic Automation program is shown in the following two examples below.

Example: Rotary table moving to different scan positions

1. Rotary Table and Robots move to home positions
2. Create the positions in PolyWorks through a macro in case they don’t exist yet in the project.

3. Set Device Position Macro
4. Measure the reflectors.
5. Measure a COP
6. Rotate the Rotary Table
7. Set Device Position Macro
8. Measure the reflectors
9. Measure a COP
10. Repeat steps 6) to 9) to add more rotary table scan positions

Example: Tracker slider moving to different scan positions

1. Tracker slider and Robots move to home positions
2. Create the positions in PolyWorks through a macro in case they don't exist yet in the project.
3. Set Device Position Macro
4. Measure the reflectors
5. Measure a COP
6. Move the tracker slider
7. Set Device Position Macro
8. Measure the reflectors
9. Measure a COP
10. Repeat steps 6) to 9) to add more tracker slider scan positions

The order of either doing first the COP measurement (step 5) or the reflector measurement (step 4) is not important.

Due to a PolyWorks limitation it is required that the COP names cannot be re-used in various positions. E.g.:

Position1

Start measurement (COP NAME: BIW_LEFT)

Start measurement (COP NAME: BIW_RIGHT)

Position2

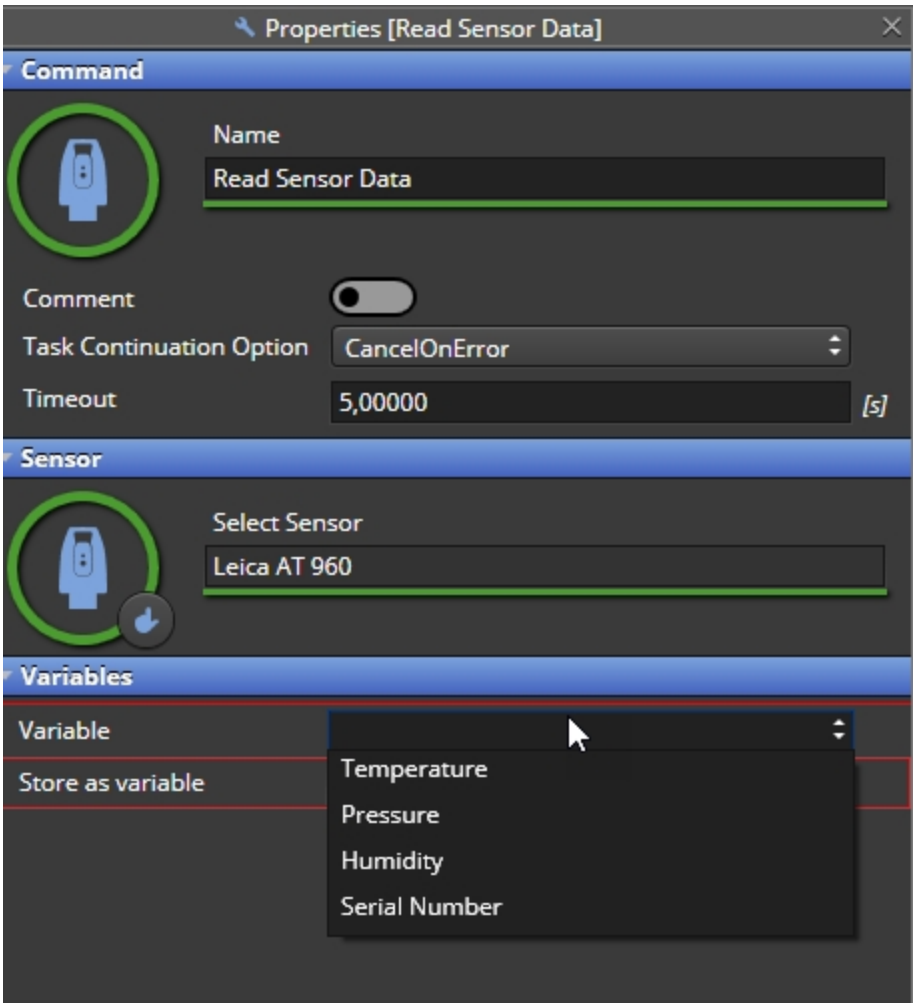
Start measurement (COP NAME: BIW_LEFT) → **Error**!

PolyWorks does not support multiple positions at the same time. This means reference points cannot be used for a dual robot, dual tracker cell where it is needed to have two parallel scans at the same time. Using HxGN Robotic Automation compensation (Nominal vs Actual reflector positions, see chapter Compensation_(HxGNRobotic_AutomationCompensation) ) is recommended for this use case.

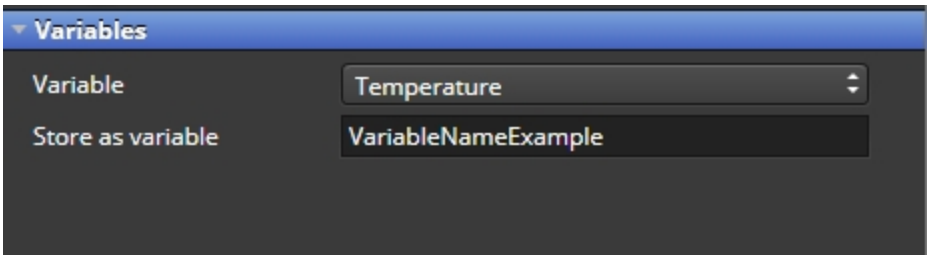
## Reading Sensor data

In the **reading sensor data** command properties, the following items can be selected to be read from RDS:

- Temperature
- Pressure
- Humidity
- Serial

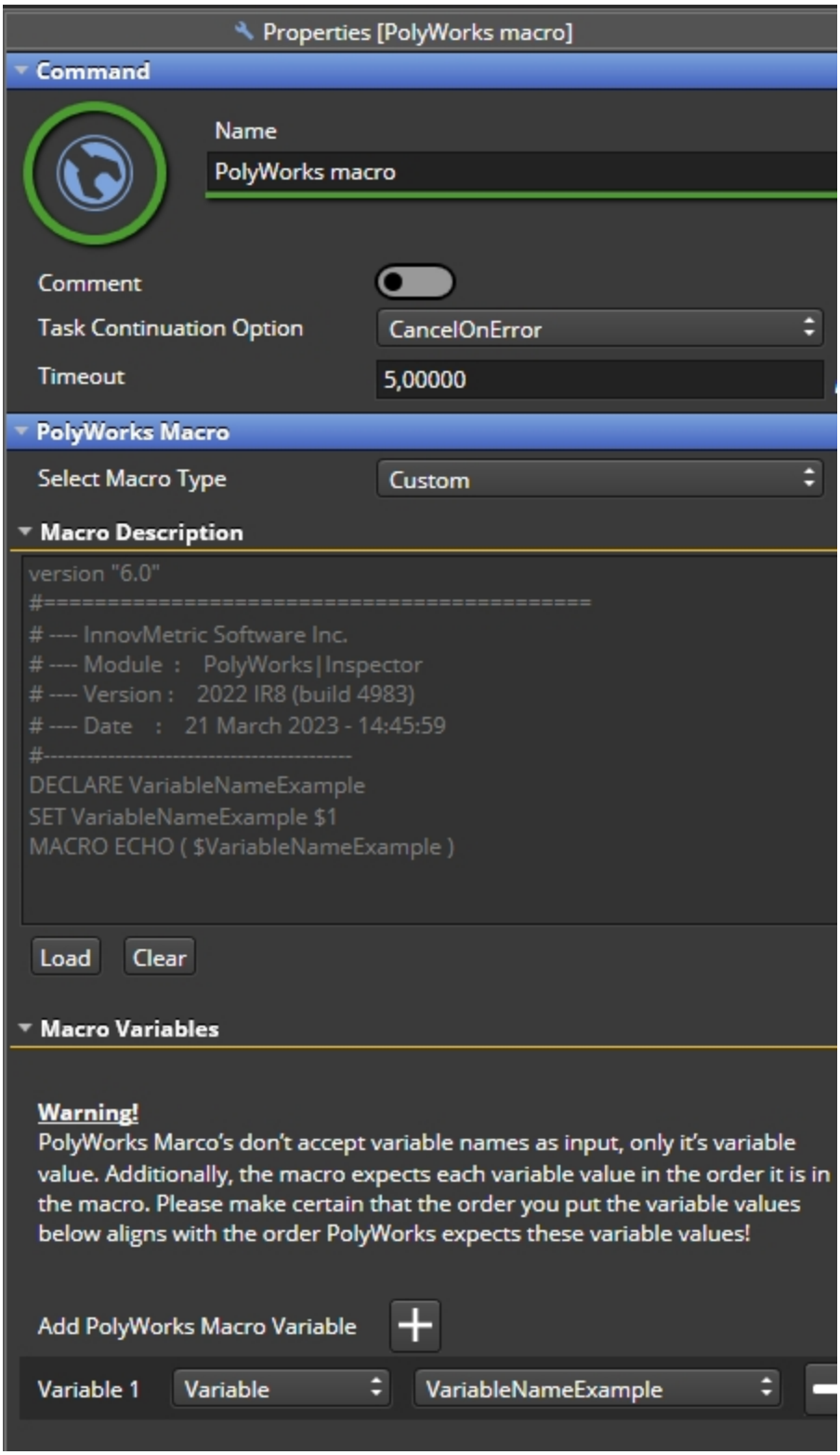



Once selecting an item to be read from RDS, it will need to be stored in a variable name




At the moment only PolyWorks has the ability to accept new variables in the middle of a program execution, so we can then send this variable with a PolyWorks Macro:

1. **Add a PolyWorks** macro command
2. Select **Custom**
3. In Macro Variables add your sensor data variables
    - In the example this was "VariableNameExample"
4. Create in PolyWorks a macro that will call the first variable
    - DECLARE VariableNameExample
    - SET VariableNameExample $1
5. In case there are multiple sensor reading commands in the same program, multiple variables will need to be added to the Macro variables. Each variable is referenced in order (e.g. $1, $2, etc....)


Properties [PolyWorks macro]
Command
Name
PolyWorks macro
Comment
Task Continuation Option
CancelOnError
Timeout
5,00000
PolyWorks Macro
Select Macro Type
Custom
Macro Description
version "6.0"
#==========
# ---- InnovMetric Software Inc.
# ---- Module : PolyWorks|Inspector
# ---- Version : 2022 IR8 (build 4983)
# ---- Date : 21 March 2023 - 14:45:59
#----------
DECLARE VariableNameExample
SET VariableNameExample $1
MACRO ECHO ( $VariableNameExample )
Load
Clear
Macro Variables
Warning!
PolyWorks Marco's don't accept variable names as input, only it's variable value. Additionally, the macro expects each variable value in the order it is in the macro. Please make certain that the order you put the variable values below aligns with the order PolyWorks expects these variable values!
Add PolyWorks Macro Variable
Variable 1
Variable
VariableNameExample

In the example we added MACRO ECHO to check the result in PolyWorks.

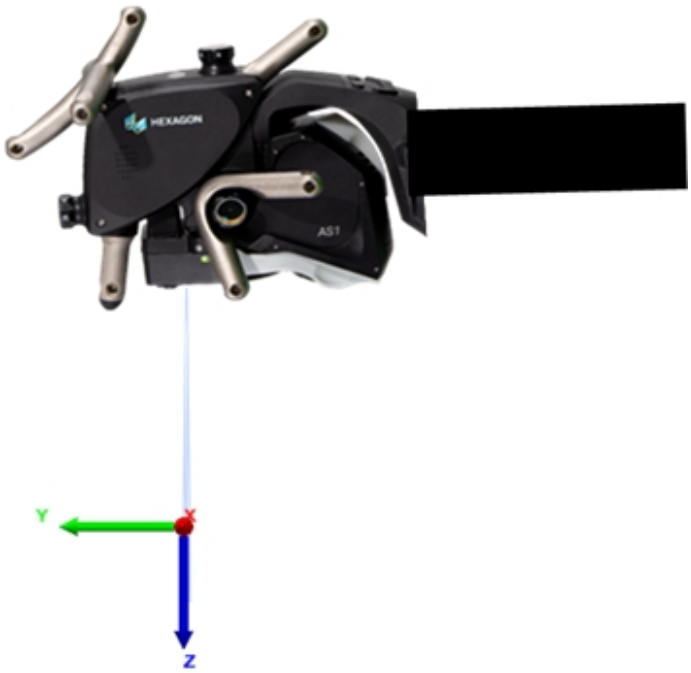
## Programmatic checks and alignments

AS1 programmatic checks and alignments allows the cell operator to easily create and execute robotic routines to verify the scanners' status and perform the necessary alignments to have the system up and running, ready to start scanning the parts that will be loaded into the robotic inspection cell.

In the following chapters, it will be explained how to create programs in HxGN Robotic Automation to perform the sphere and three plane distance checks and the sphere alignment process.

For better understanding of the robot movement recommendations described in the next chapters, the following robot tool definition (TCP) is used for the AS1. TCP is defined in the middle of the scan line and working range of the AS1 with the following axis directions:

Front view (Face 2)




Side view (Face 1)

## Sphere check

This check verifies the current scanline to probe face alignment. A calibrated sphere is needed (provided with each scanner). The sphere can be measured with all four faces combined, the important to achieve the recommended sphere coverage (80%).

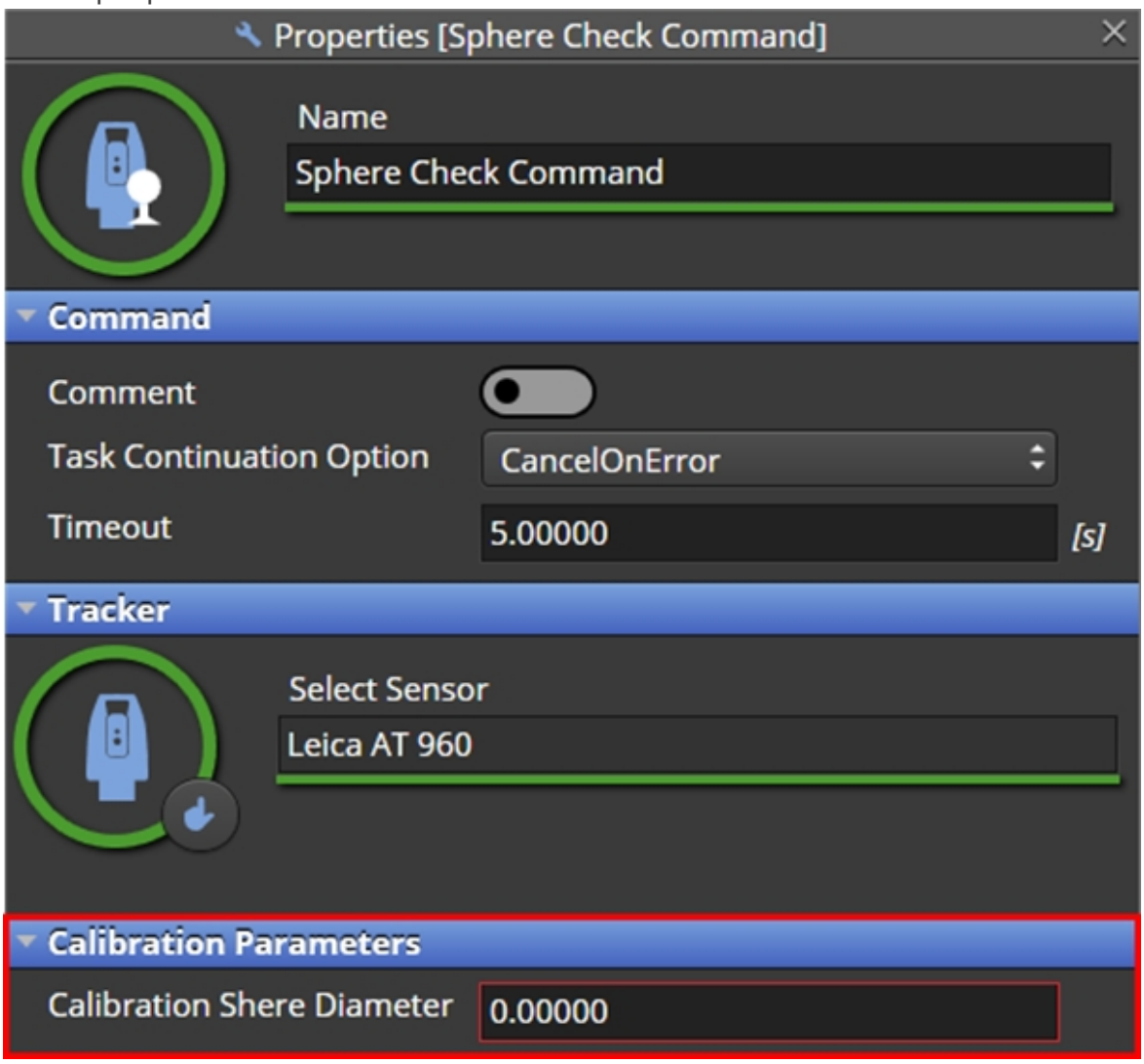
#### Workflow

The workflow is the following:

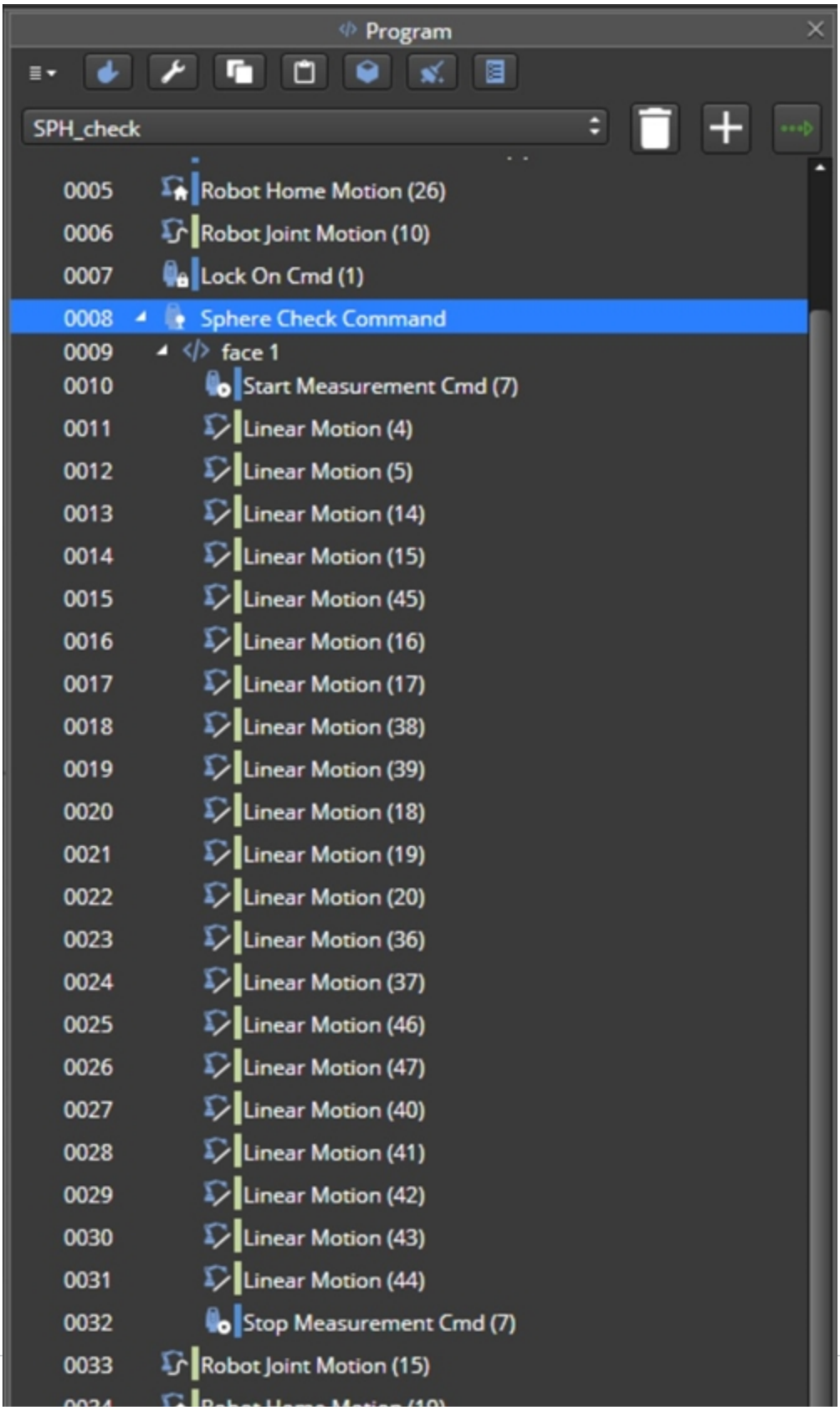
1. Move the robot to the starting position using a **Robot Joint Motion** command.
2. Lock on the AS1 using a **Lock On** command.
3. Use the **Sphere Check** Command in RA, this will create a group.

Within this group, the user can add the necessary commands as explained in the following steps.

First you need to add the sphere diameter value within the sphere check command properties as shown below.



4. Within the **Sphere Check** Command you can use the various faces of the tool to get a minumum 80% coverage of the sphere.


Program
SPH_check
0005 Robot Home Motion (26)
0006 Robot Joint Motion (10)
0007 Lock On Cmd (1)
0008 Sphere Check Command
0009 face 1
0010 Start Measurement Cmd (7)
0011 Linear Motion (4)
0012 Linear Motion (5)
0013 Linear Motion (14)
0014 Linear Motion (15)
0015 Linear Motion (45)
0016 Linear Motion (16)
0017 Linear Motion (17)
0018 Linear Motion (38)
0019 Linear Motion (39)
0020 Linear Motion (18)
0021 Linear Motion (19)
0022 Linear Motion (20)
0023 Linear Motion (36)
0024 Linear Motion (37)
0025 Linear Motion (46)
0026 Linear Motion (47)
0027 Linear Motion (40)
0028 Linear Motion (41)
0029 Linear Motion (42)
0030 Linear Motion (43)
0031 Linear Motion (44)
0032 Stop Measurement Cmd (7)
0033 Robot Joint Motion (15)

5. For every face used (one is needed, the rest are optional) use one **Start Measurement** command at the beginning, add **Linear Motions** as needed, and add one **Stop Measurement** command at the end of the group.

   | | |
   |---|---|
   | ⚠ | It is possible that in complex scenarios (AS1 scanning extension on a robot without slider) during the scanning the LOS is obstructed, and the measurement must be stopped and started again after repositioning the robot in a reachable position. |

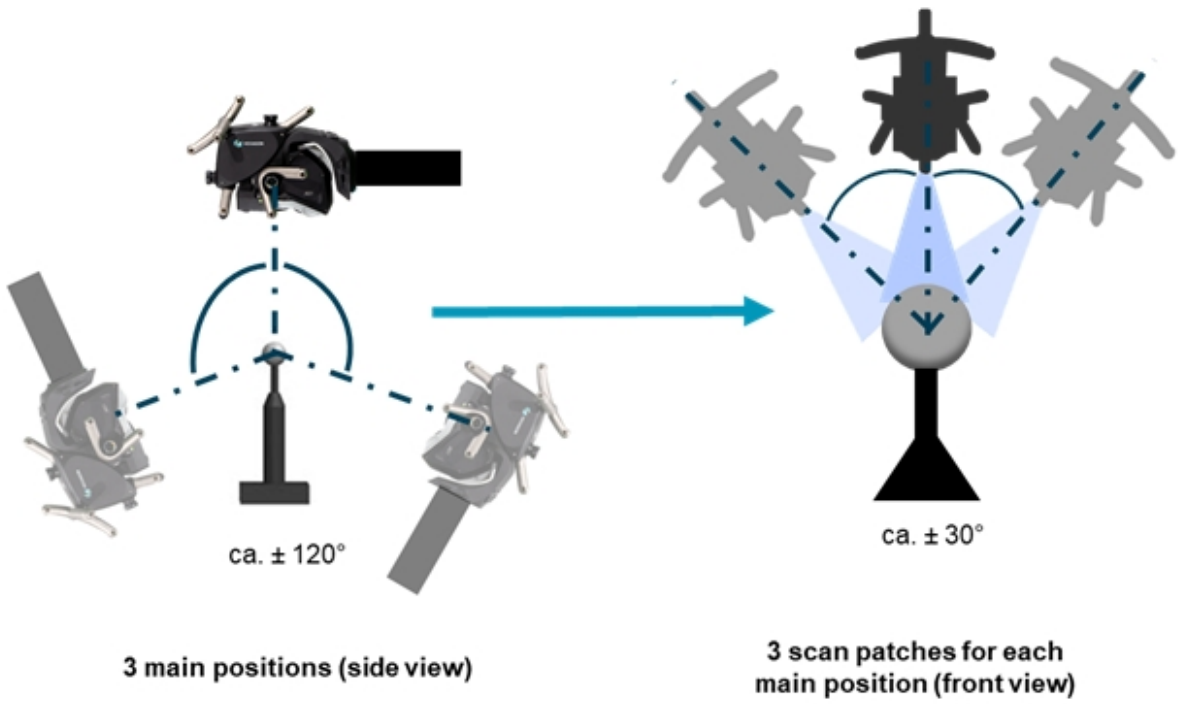
6. When exiting the group, RA will send a sync script to RDS and RDS will start calculating the results.
7. Move the robot to the homing position using a **Robot Home Motion** command to complete the program.

#### Recommended robot motions for sphere measurements

The recommended robot positions are depicted below.

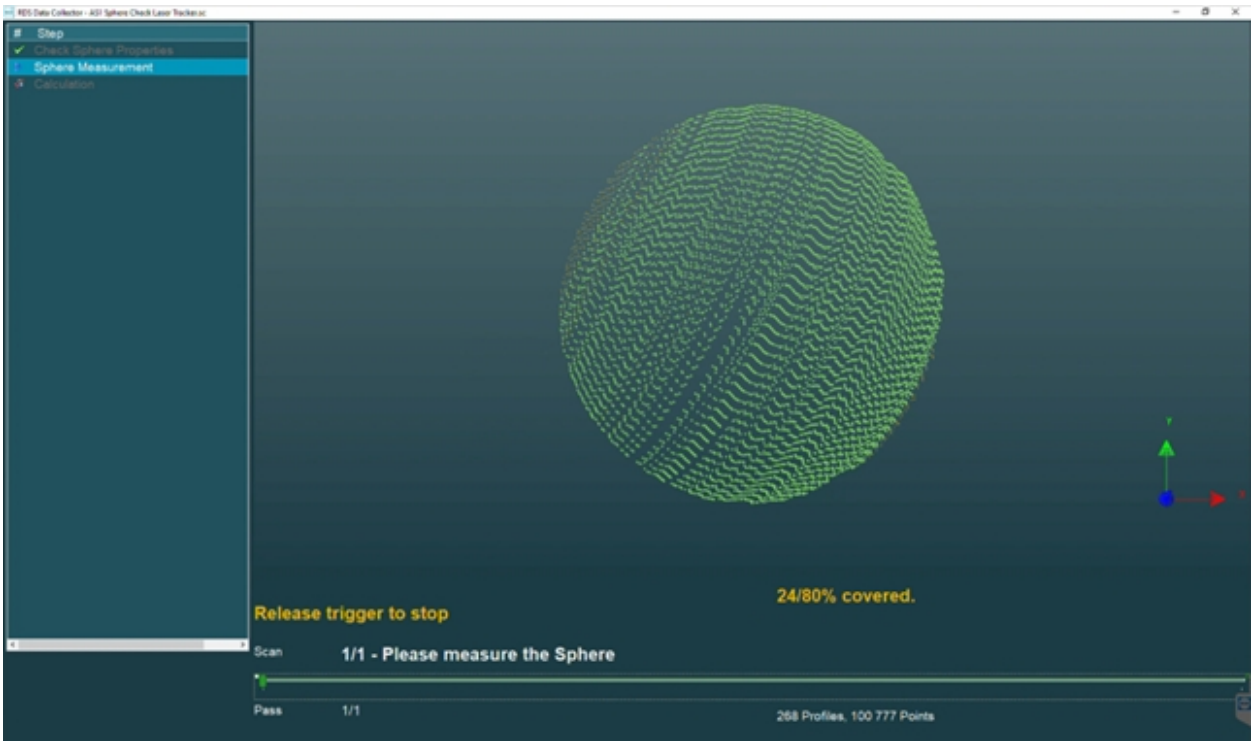
The scanner must have the center of the scan line at standoff distance from the sphere, using a scanning speed of around 200 mm/s.

Using the three main positions and three scan patches for each main position described in the picture below, a coverage of 87% can be achieved.




#### Execution

During the execution of the **Sphere Check** Command group, the RDS Data collector will appear and show the coverage percentage.




After the execution of the group is completed, the RDS Data collector will close.

- In case the script is not successful, an error message will be displayed in the status view of HxGN Robotic Automation, and the program execution will be aborted.
- In case the result of the check is successful, it is possible to retrieve the log file from the "*C:\User\Public\Documents\Romer\RDS\CalibData*" folder.

## Three plane distance check

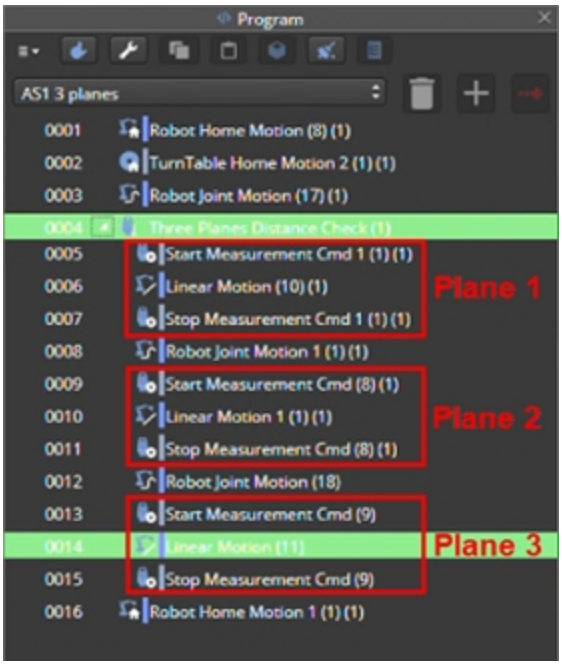
This check verifies if the measurement range calibration is good. Any flat surface can be used for the check.

#### Workflow

The workflow is the following:

1. Move the robot to the starting position (e.g., standoff) using a **Robot Joint Motion** command.
2. Lock on the AS1 using a **Lock On** command.
3. Use the **Three Plane Distance Check** command in RA, this will create a group.
4. Add a **Start Measurement** command at the beginning, add **Linear Motions** as needed, and a **Stop Measurement** command at the end of the scan when the first plane is covered. This can be verified using the RDS Data collector.

   It is also possible that a single Stop Measurement command is used at the end of the complete scan of all the three planes.
5. Do the same for the second and third planes as shown in the picture below, adding Robot Joint Motions to move the robot to the new starting positions to scan the remaining two planes within the other two working range fields.
6. When exiting the group, RA will send a sync script to RDS and RDS will start calculating the results.
7. Move the robot to the homing position using a **Robot Home Motion** command to complete the program.

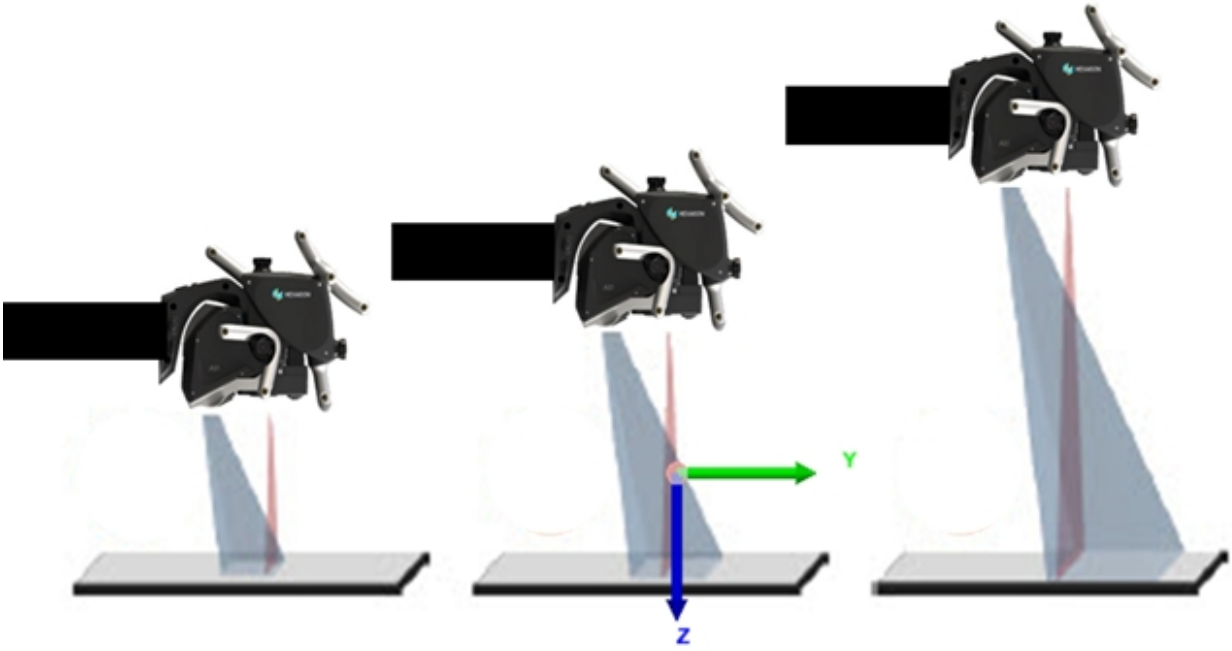



**Recommended robot motions**

The recommended robot positions are depicted below.

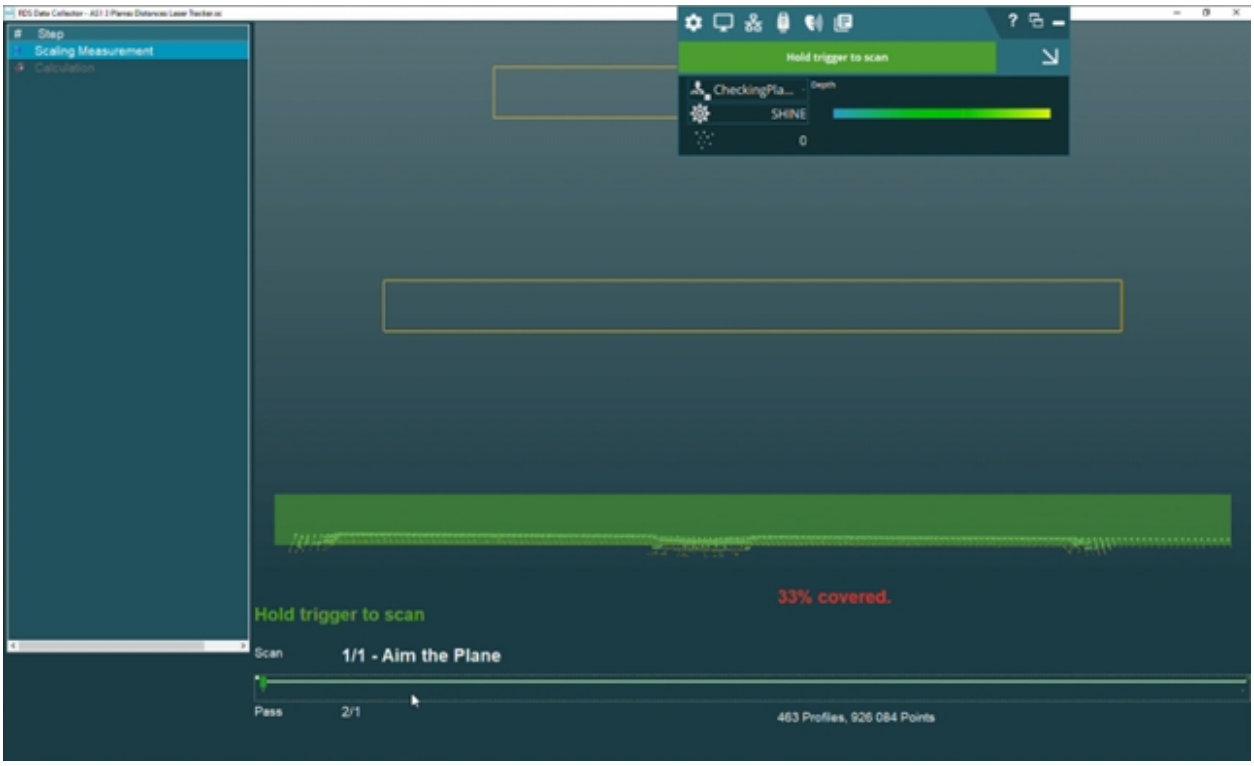
Three scan patches (Near, Standoff, Far) on the same area of the plane are needed, moving the scanner along the Z-axis around ± 40 mm from standoff to reach the near and far scanning positions.

The scan direction doesn't need to be the same for all three scan patches, back and forward movements are possible. The scan speed should be around 100 mm/s and the scan patch length should be 100 -150 mm.




**Execution**

During the execution of the Three Plane Distance Check group command, the RDS Data collector will appear and show the coverage for each plane.




After the execution of the group is completed, the RDS Data collector will close.

- In case the script is not successful, an error message will be displayed in the status view of HxGN Robotic Automation, and the program execution will be aborted.
- In case the result of the check is successful, it is possible to retrieve the log file from the "*C:\User\Public\Documents\Romer\RDS\CalibData*" folder.

### Sphere Alignment

This process aligns the scanline to the probe face of the AP21 and needs to be done with each face of the AP21 separately. It also calibrates the measurement range and the laser line itself. A calibrated sphere is needed (provided with each scanner).

This alignment is enough for the proper Scanner operation.

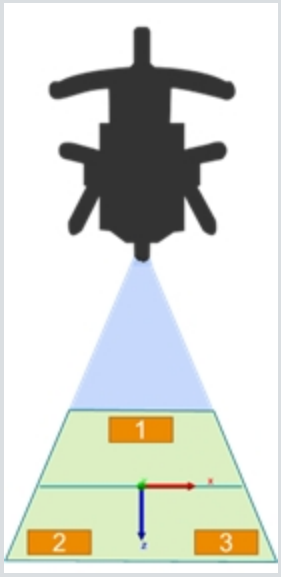
The face used in the **Sphere Alignment Measurement Range** and the subsequent **Sphere Alignment Sphere Measurement** need to be the same!

#### Workflow

The workflow is the following:

1. Move the robot to the starting position using a **Robot Joint Motion** command.
2. Lock on the AS1 using a **Lock On** command.
3. Use the **Sphere Alignment Measurement Range** command in RA, this will create a group.
4. With not specific order between the three FOV boxes depicted below:
    a. Measure the sphere with the top of the tracker FOV cone (Box 1)
    b. Measure the sphere with the left bottom side of the tracker FOV cone (Box 2)
    c. Measure the sphere with the right bottom side of the tracker FOV cone (Box 3)

| | |
|---|---|
| ✓ | Example: Reference position: center of scanline at standoff distance |




**Box 1 (near):**

Center_x = 0 mm

Center_z = -37.5 mm

**Box 2 (far left):**

Center_x = -70 mm

Center_z = +37.5 mm

**Box 3 (far right):**

Center_x = 70 mm

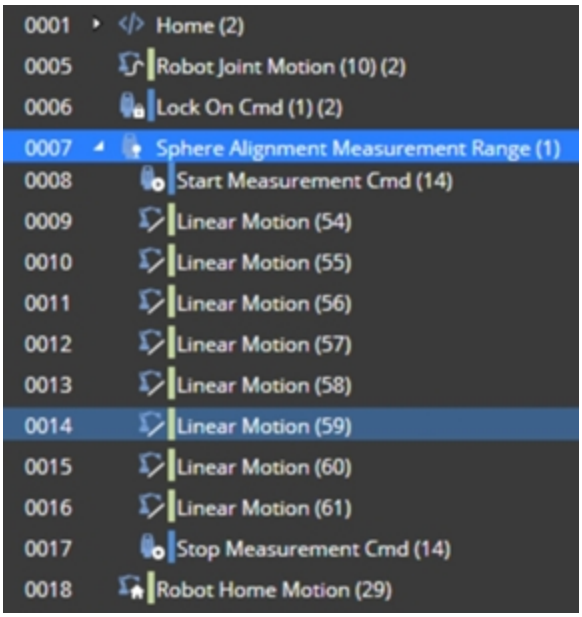
Center_z = +37.5 mm

**Box dimensions:**

Length_x = 30.5 mm

Length_z = 12.7 mm

:The program in HxGN Robotic Automation will like the following:



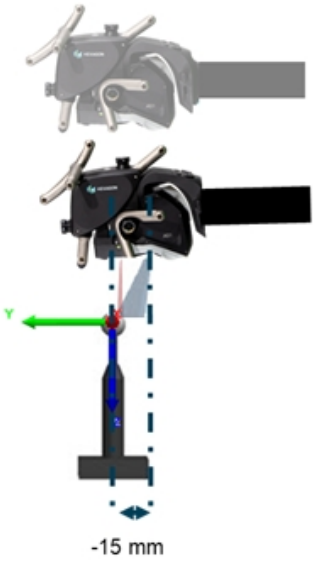

5. When exiting the group, RA will send a sync script to RDS and RDS will start calculating the results.
6. Use the **Sphere Alignment Sphere Measurement** command in RA, this will create another group.
7. Within the **Sphere Alignment Sphere Measurement** group you will need to cover the sphere with only 1 face of the tool.
    a. to achieve the following recommended minimum sphere coverage
        i. 80% for Faces 1, 2 and 4
        ii. 55% for Face 3 (Back Face)
    b. This process will need to be repeated for each face of the tool in a separate **Sphere Alignment Sphere Measurement** group
8. When exiting the group, RA will send a sync script to RDS and RDS will start calculating the results.

9. Move the robot to the homing position using a Robot Home Motion command to complete the program.

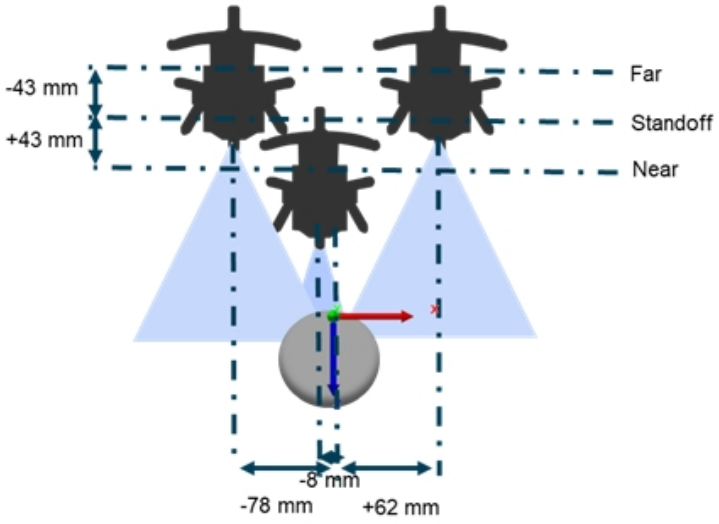
#### Recommended robot motions for range measurements

The recommended starting positions to perform the range measurement on the sphere are the following:

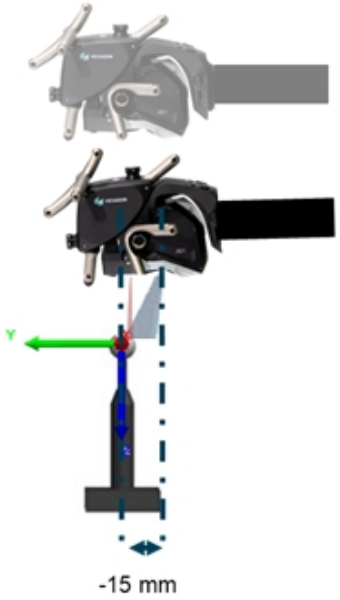
- **Reference position**: the middle of the laser line is in standoff distance to the sphere, where the laser line cuts the sphere in half.
- **Main starting positions with respect to TCP coordinate system:**
  - Near range: X = -8 mm, Y = -15 mm, Z = +43 mm
  - Far left range: X = - 78 mm, Y = -15 mm, Z = -32 mm
  - Far right range: X = + 62 mm, Y = -15 mm, Z = -32 mm



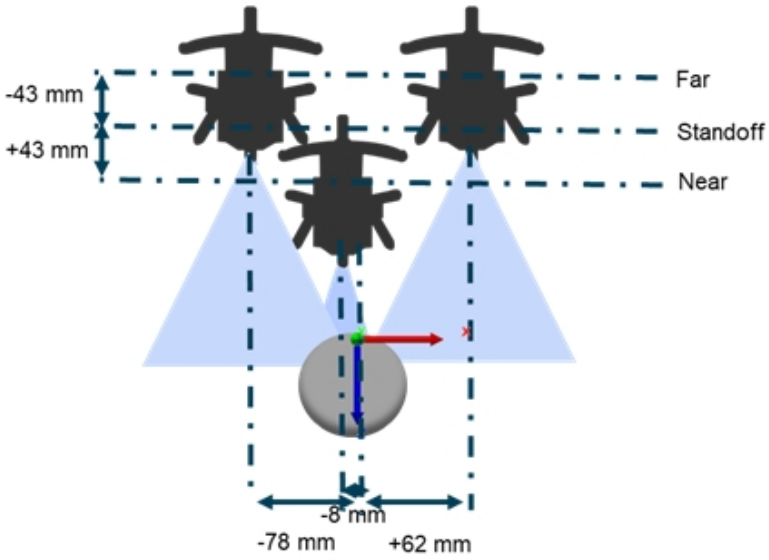

Main positions side view




Main positions front view



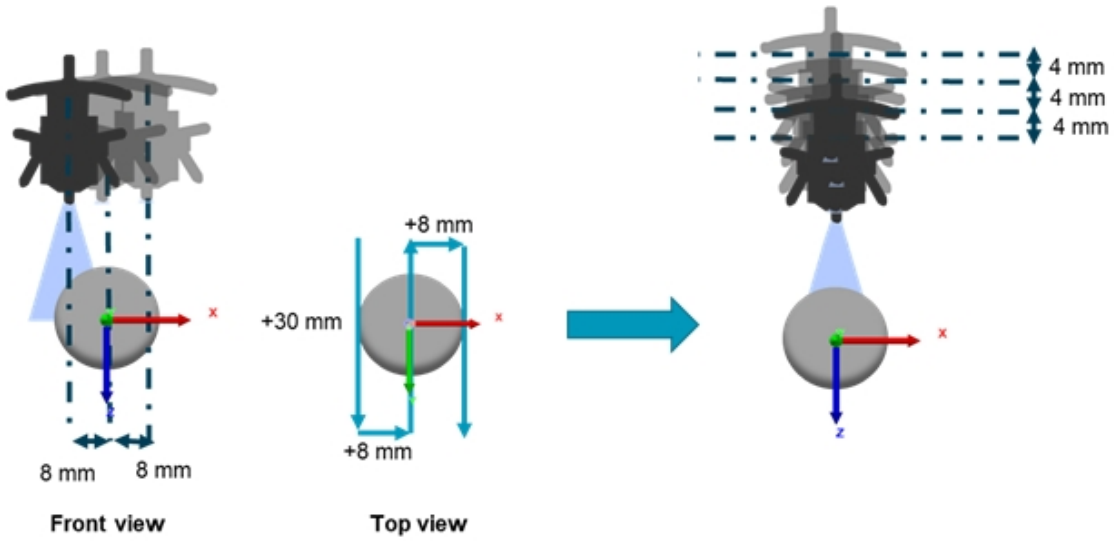

Main positions side view




Main positions front view

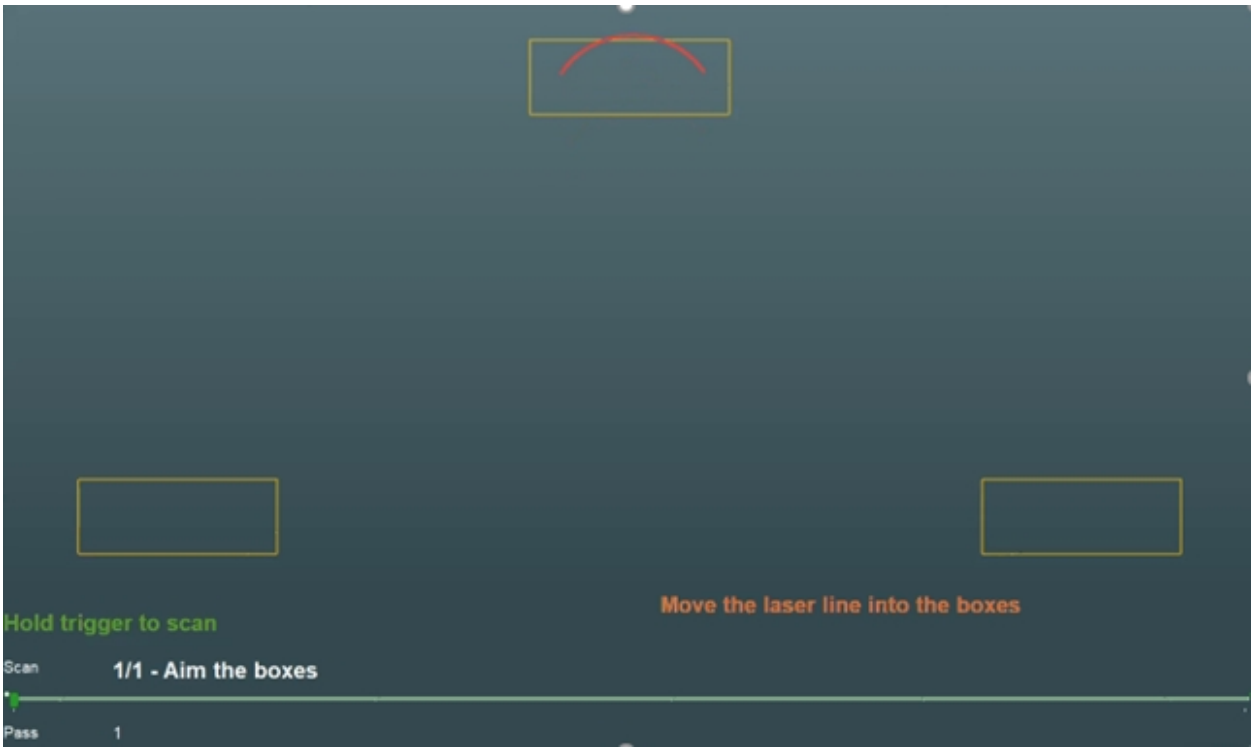
To perform the range measurement, it is recommended to perform the following robot movements starting from each of the three main starting positions, using a scanning speed of around 200 mm/s:

- Start scan at starting position, move +30 mm in y-direction to measure the sphere.
- Shift + 8 mm in x-direction and move -30 mm in y-direction
- Shift another + 8 mm in x-direction and move +30 mm in y-direction
- Repeat this scan path in three more different heights, each shifted -4 mm more in z-direction




**Execution**

During the execution of the group command, the RDS Data collector will appear and show the coverage for each FOV cone of the AS1.



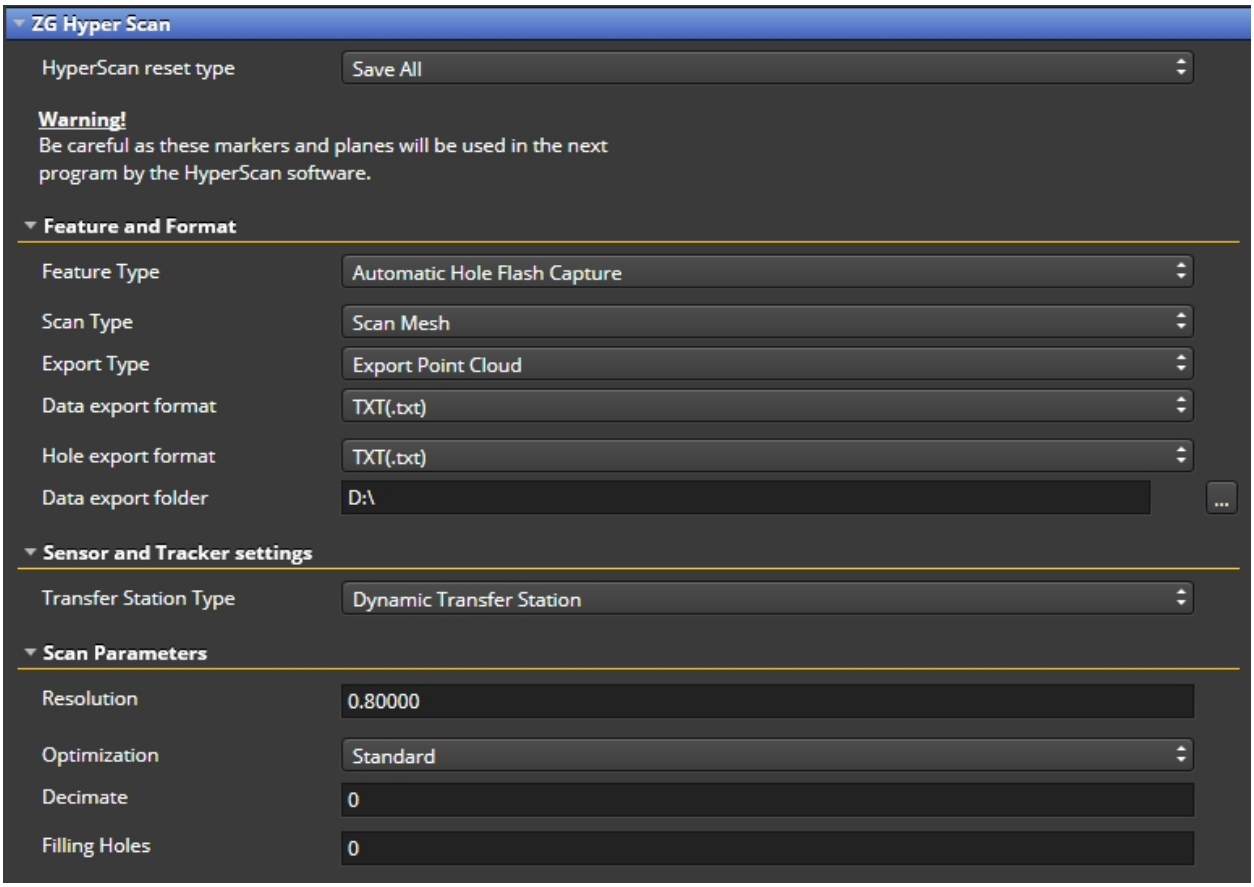

If sufficient data are recorded the boxes will turn green and the calculation will start, otherwise if the sphere was not measured with the correct part of the FOV cone, a time out error will be reported.

After the execution of the group command is done, the RDS Data collector will close.

- In case the script is not successful, an error message will be displayed in the status view of HxGN Robotic Automation, and the program execution will be aborted.
- In case the result of the check is successful, it is possible to retrieve the log file from the "*C:\User\Public\Documents\Romer\RDS\CalibData*" folder.

# Programming - ZG

## Program properties




1. **Hyperscan reset type** options:
    a. Save all: In this mode, the markers and planes obtained from the scan will not be reset.

    **All the markers and planes obtained from previous scans will be retained**.

b. Save Planes: In this mode, the markers obtained from previous scans will be deleted, but the planes will not be clipped.

c. Save Makers: In this mode, all the markers obtained from previous scans will be retained, but the planes will be clipped.

d. Reset Markers & Planes: In this mode, all the markers and planes will be deleted and clipped.

2. **Feature Type** options:

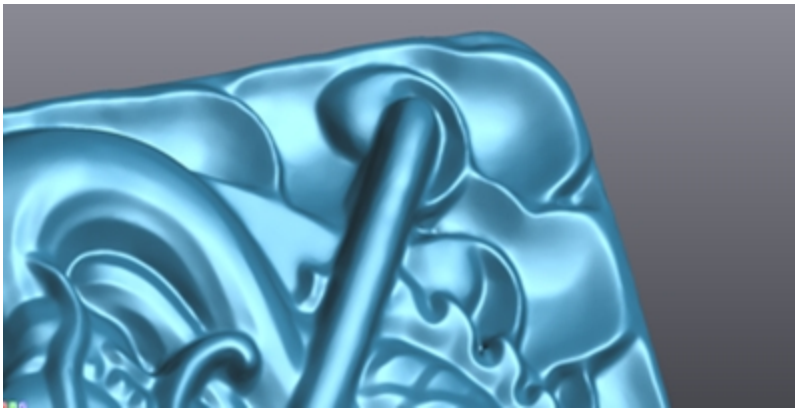
a. Automatic Hole Flash Capture: Used to capture both surfaces and features. It will export 2 files, ones for the point cloud or mesh and one for the hole capture.

b. Surface Scan (No Hole Flash Capture): Only surface scan, a single export containing a mesh or point cloud can be selected.

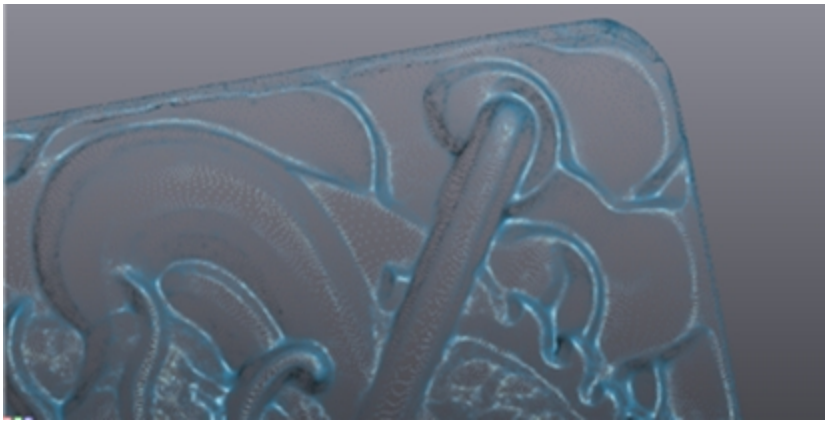
c. Only Hole Flash Capture (No Surface Scan): Only holes can be captured, a single export containing the hole data can be selected.

3. **Scan Type** options:

a. **Scan mesh** mode: The scanner will collect the surface and output it as a mesh grid. The effect is shown in the figure below.



b. **Scan Point Cloud** mode: The scanner will collect the point cloud and output it as a point cloud. The effect is shown in the figure below.



There is no difference for the scanner device itself when scanning the surface or the point cloud. The scanning process of the scanner is exactly the same, only the scanned data is different. It is recommended to perform the scanning in the surface scanning mode, i.e., the **Scan Mesh** mode. After the processing of the scanned data is completed, you can choose to export the surface or the point cloud.

4. **Export Type**: Here can select according to their own needs. Export Mesh or Export Point Cloud.
5. **Data export format**:Here can select according to their own needs. There are six data formats in total, namely txt, asc, xyz, pnt, igs, and csv.
6. **Hole export format**: Here can select according to their own needs. There are four data formats in total, namely txt, stl, stp and iges
7. **Data export path**:When adding a **Scan** command, it is necessary to set the **Data export path** to specify the storage location of the data obtained from the scan.

8. **Transfer Station Type**:Here the type of transfer station that will be used in the program can be selected. See chapter: Transfer_Station
9. **Markers import path**: This is used with the **Transfer Station** command. It is necessary to set the **Markers import path** to specify the file path for reading the pre-collected marker position information.

The following scan settings can only be set once per program, this is why they are located in the program properties. During the scan process, the user can only reset the Laser Line Type parameter.

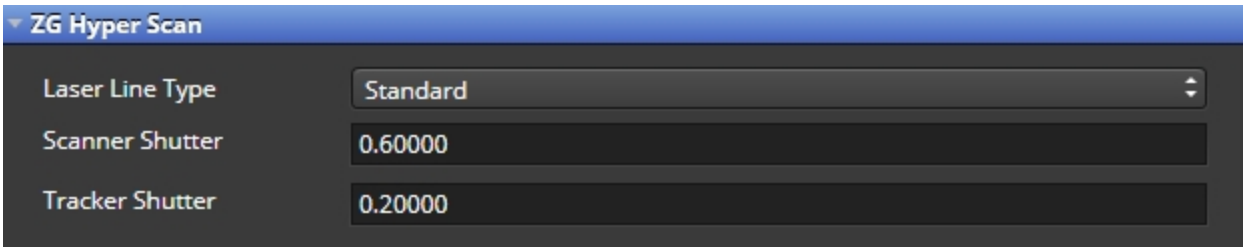
- **Resolution**: This determines the point cloud spacing of the model generated by the scanner. Different device models have different setting ranges. A smaller resolution value results in denser point clouds, larger data volume, slower scanning speed, and better scanning details. Conversely, a larger resolution value results in sparser point clouds, smaller data volume, faster scanning speed, and reduced object detail.
- **Optimization**: This is an optimization option that offers three levels of optimization to make the mesh surface smoother. The parameters include None, Standard, and High, which progressively increase the mesh optimization intensity. The resulting mesh data becomes smoother while maintaining a certain level of accuracy.
- **Decimate**: This is a simplification option that reduces the data volume of the final generated model. The setting range is 0 to 100, where the value represents the percentage of simplification. For example, a value of 16 means simplifying 16% of the data, leaving 84% of the data remaining.
- **Filling Holes**: This is a filling option that can fill small holes in the model. The setting range is 0 to 100, where the value represents the size of the holes to be filled. The default value is 50, which can fill holes up to φ5mm in the model.

  After completing the final Scan command, RA will automatically transmit the scanned data to Metrology.

## Scan command

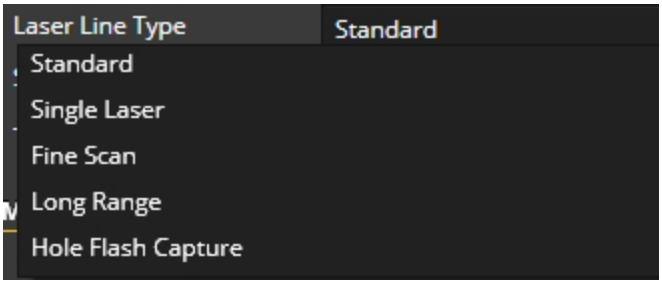
Typically, the Scan command is used to add scanning instructions for the ZG H-Track Ultra.

The scan command has the following properties:




**Scanner Shutter**: This represents the shutter exposure time during the scanner's operation, measured in milliseconds. A larger value results in brighter laser illumination, enabling the recognition of workpieces with darker colors.

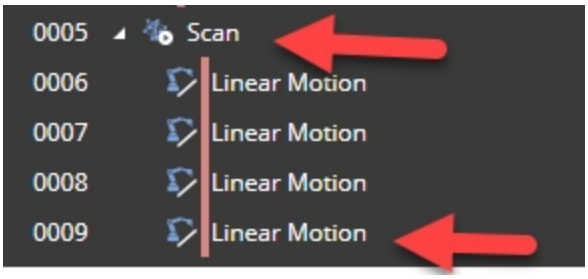
**Laser Line Type**:This will select the type of laser used by the scanner.




| Laser Line Type | Divergence Angle | Min. Distance | Max. Distance | Ideal Distance |
|---|---|---|---|---|
| Standard | length: 53º<br>With: 40º | 200 mm | 450 mm | 300 mm |
| Fine Line | length: 53º<br>With: 25.4º | 150 mm | 250 mm | **180 mm** |

| Laser Line Type | Divergence Angle | Min. Distance | Max. Distance | Ideal Distance |
|---|---|---|---|---|
| Long Range | length: 53°<br>With: 40° | 200 mm | 750 mm | 300 mm |
| Hole<br>Flash Capture | | | | |

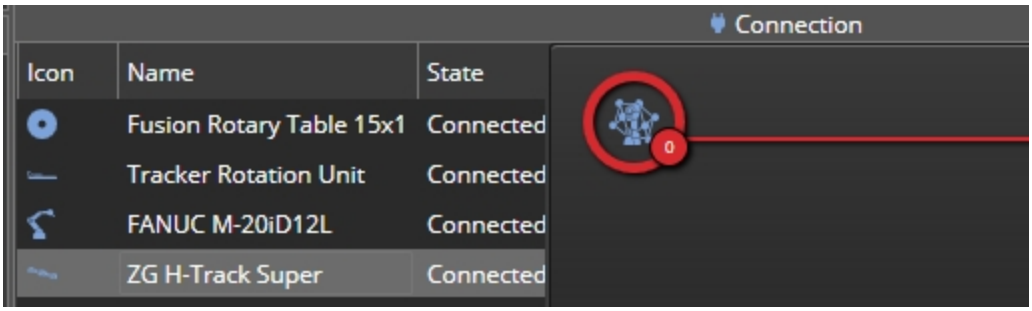
The tool will start scanning at the start of the scan command and stop when the last motion within the composite command has been reached.

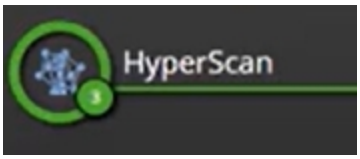



To use Pick on CAD functionality with the correct scanning cone visible. Select the tool in the modelling tab and set the laser line type to the desired one.

## Tracker Visibility on the Scanner

When selecting the tracker in HxGN Robotic Automationconnection window it will show how many tracker "balls" can be seen on the scanner from the tracker points of view. There should be at least 2 visible for a correct scan!

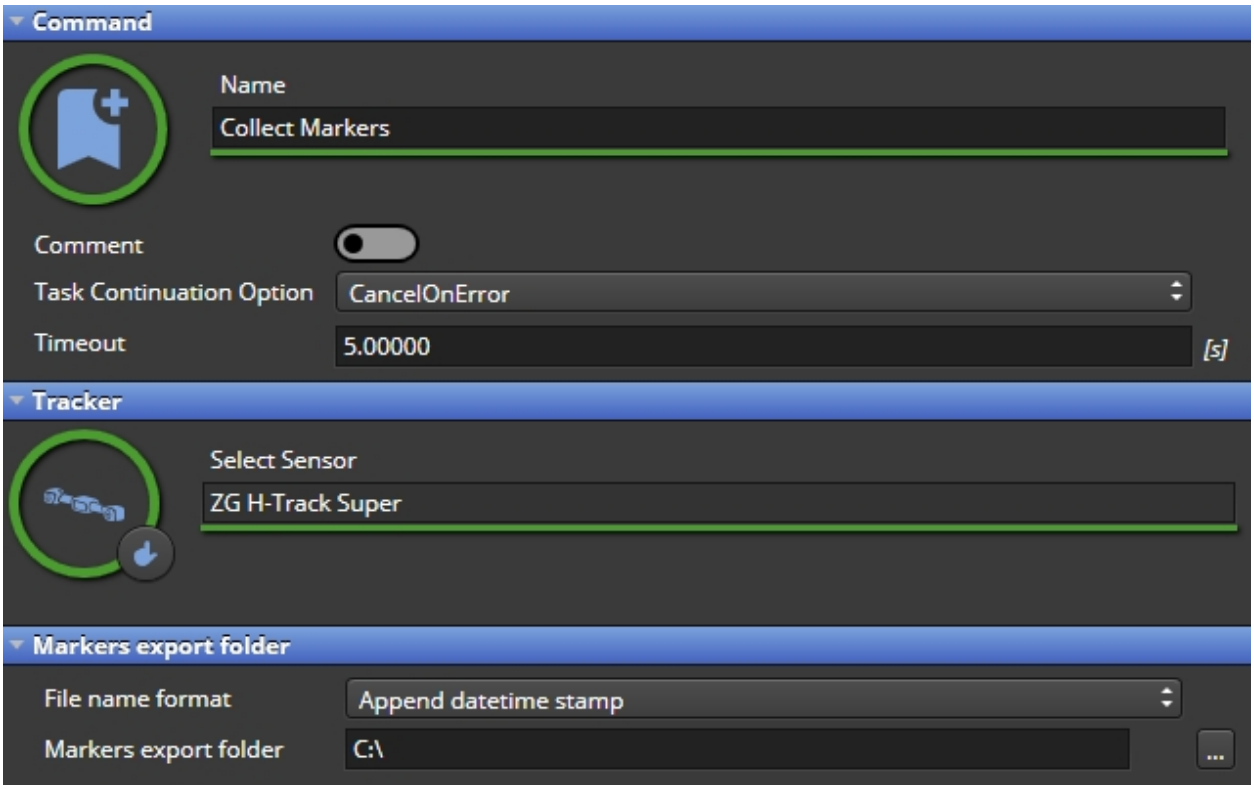



## Collect markers

The primary function of the Collect Markers command is to gather the pose information of the marker points required for the Transfer Station command. It is necessary to run the Collect Markers command before executing the Scan command and the Transfer Station command.

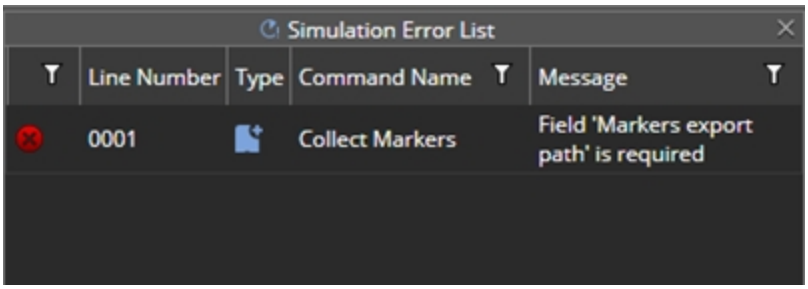
1. Capture Marker Points: The Tracker captures the marker points at the current pose.
2. Adjust Position: Perform the corresponding turntable rotation or movement of the robot or slider.
3. Export Pose Information: Export the pose information of the marker points in a text file (txt format).




This command can be configured through Program Properties, where the Markers export path must be set.

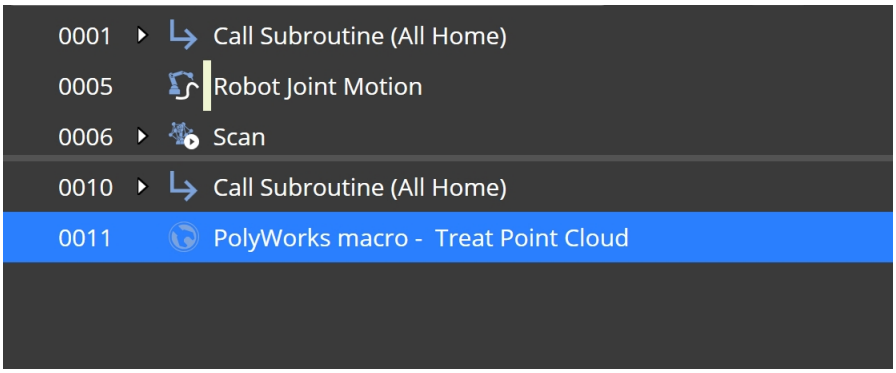
- **Append datetime stamp**: Used in case specific marker files go with specific programs
- **Used fixed name**: Will overwrite any existing markers file in that directory. For example when running a collect marker program in Execution Manager and afterwards a program using transfer station, this 2nd program will then use the newest marker file.

This path specifies the export location for the txt file containing the pose information of the marker points. This parameter is mandatory, and failure to set it will result in an error like follow:



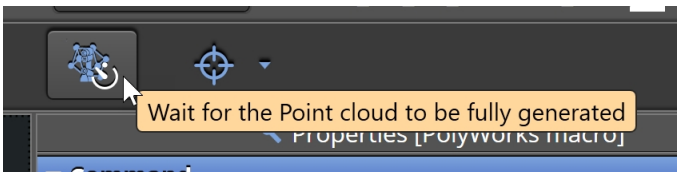

## Wait for the Point cloud to be fully generated

If after the execution of a program, the point cloud needs to be treated in PolyWorks through a command like the example below.

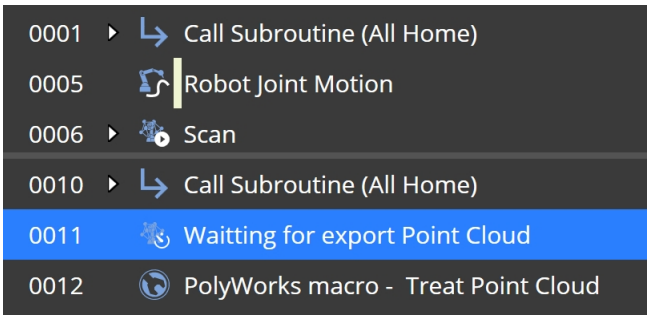



This PolyWorks command might fail as the point cloud has not been generated by the Hyperscan software and sent to PolyWorks.

To avoid this scenario a command has been added that will pause the program untill the point cloud has been absorbed by the Metrology application.




This can be used in the example program in the following way:




## Transfer Station

The Transfer Station command functions by reading the position information of the Markers to perform coordinate transformation, thereby assisting in the alignment and stitching of point clouds.

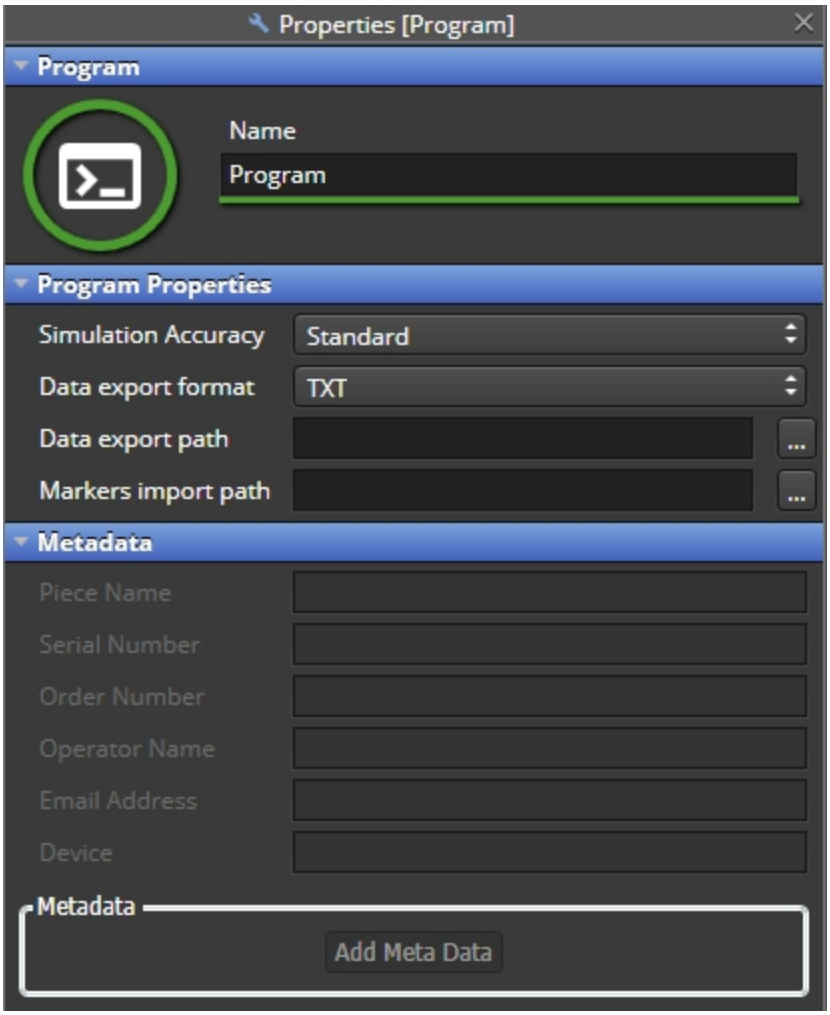
Before using this function it is first advisable to understand which type of transfer station is applicable for the program.

- **No Transfer Station**: Used if the tracker and part relationship doesn't change. (No rotary table movement, no tracker movement)
- **Static Transfer Station**: When there is a need for high precision and there is no strict requirement for the cycle time or the markers have a very little overlap between the various positions. For example **high requirements for the precision**, more effected by vibrations.
- **Dynamic Transfer Station**:(**Recommended Option**) When there is significant vibration and real-time stitching is required or the markers have a big overlap between the various positions.

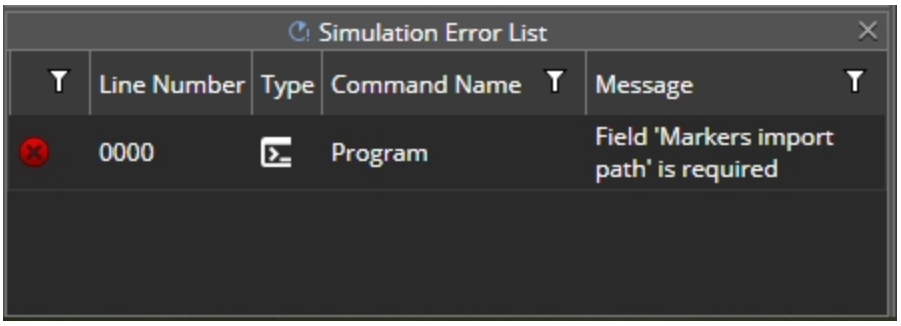
This command is primarily divided into three parts, as follows:

1. Pre-command: If the current station transfer operation is the first one in the program, the pre-command corresponds to reading the Markers data. If the current station transfer operation is not the first one in the program, the pre-command corresponds to communicating with Hyperscan to initiate the station transfer.
2. Execution Command: Execute the turntable rotation or the movement of the robot or slider to adjust the position and orientation of the object being measured. This can be achieved through previous Robot, Rotary Table, or Slider commands.
3. Post-command: Conclude the station transfer operation

This command can be configured through "*Program Properties*", with details as follows:




The parameter Markers import path specifies the required Markers position information, which is typically generated by the Collect Markers command through export. This parameter is mandatory, and failure to set it will result in an error like follow:



Both Dynamic Transfer Station (real time tracking) and Static Transfer Station are set through the ZG configuration. On their use case see the following chapter Dynamic_Transfer_Station_(Real_Time_Tracking) & Static_Transfer_Station.

## Dynamic Transfer Station (Real Time Tracking)

Use case:
When there is significant vibration and real-time stitching is required.

or

The markers have a big overlap between the various positions. For example a rotation of markers with less than 20º.

In the Dynamic Transfer Station it is only required to add 1 **Transfer Station** command at the start where the software will import the markers. Other **Transfer Station** commands can be added, but will serve no purpose.

## Static Transfer Station

Use case:
When there is a need for high precision and there is no strict requirement for the cycle time

or

The markers have a very little overlap between the various positions. For example a rotation of markers more than 20º.

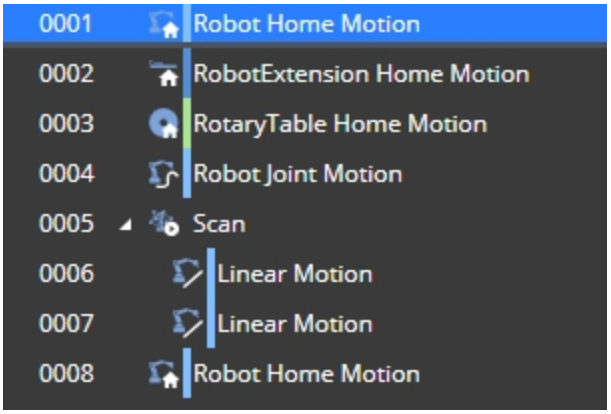
In the Static Transfer Station it is required to add 1 **Transfer Station** command at the start where the software will import the markers and additionally every Rotary Table movement or Tracker Slider movement will need to be hosted under a **Transfer Station** command. See example programs. Example_Programs

## Example Programs

See below an example of a Hyperscan programs in HxGN Robotic Automation (Showing collect markers, device position and transfer station)

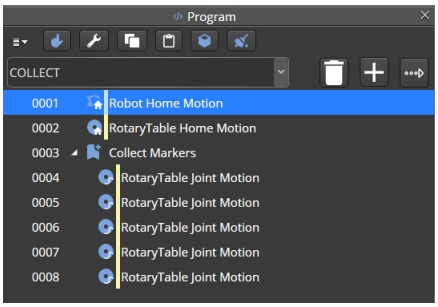
### No Transfer Station example:

No rotary table nor tracker slider movement with direct scans only.




### Collect Makers (required for Static or Dynamic Transfer Station)

- To avoid the need to collect markers again for each *Scan* program, we usually edit the collect markers command as a separate program, as shown in the following example:

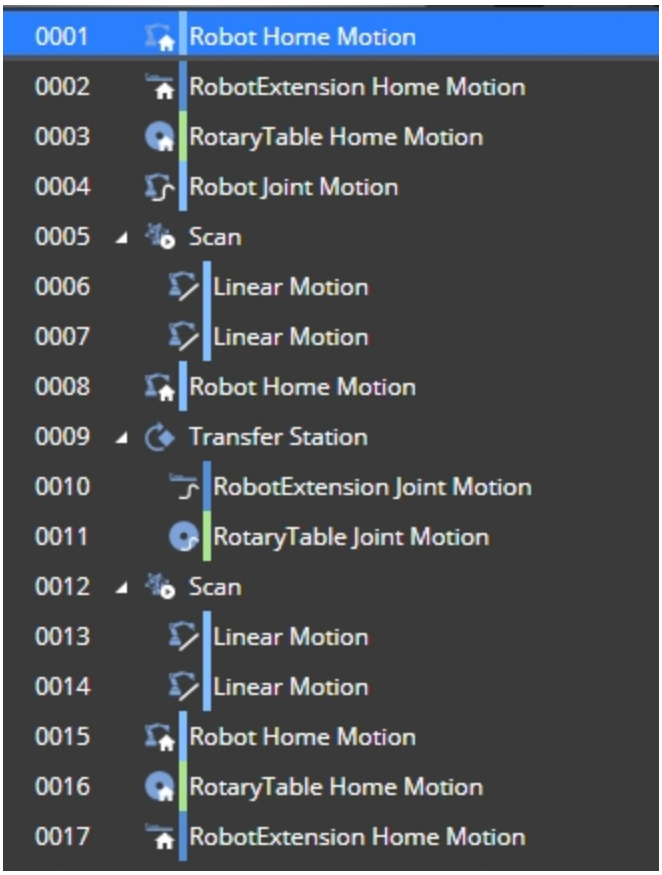



When performing **Collect markers**. You need to set its *Markers export path* in *properties*. This path is used to store the collected markers data. The data is usually stored in txt file format.

Then the scanning program can be created.

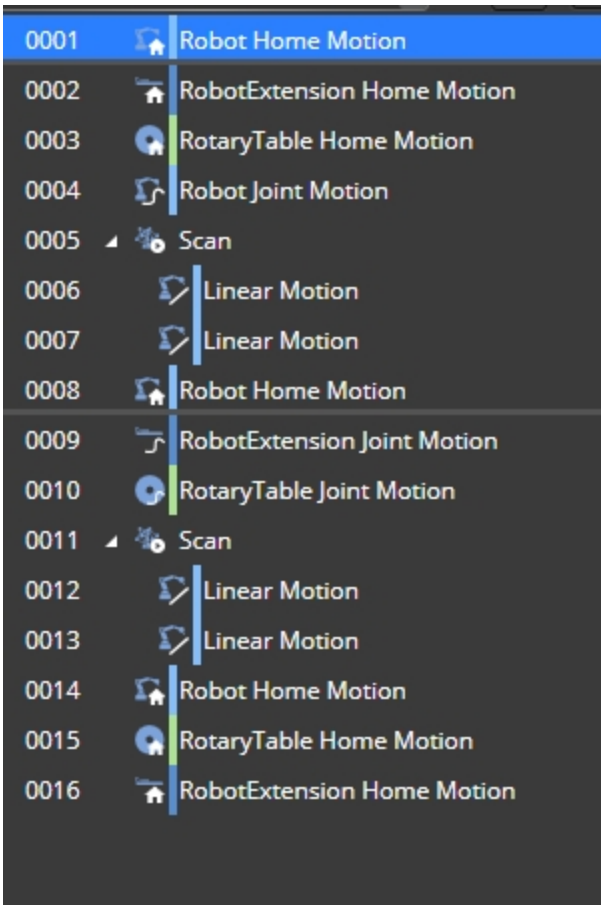
## Example Static Transfer Station Program:

The part and tracker relationship between scan commands need to be captured in a transfer station command.




### Example Dynamic Transfer Station Program:

Example Dynamic Transfer Station Program, where the Rotary Table or Tracker Slider movements are not required to be added under a Transfer Station command.




# Path Planning

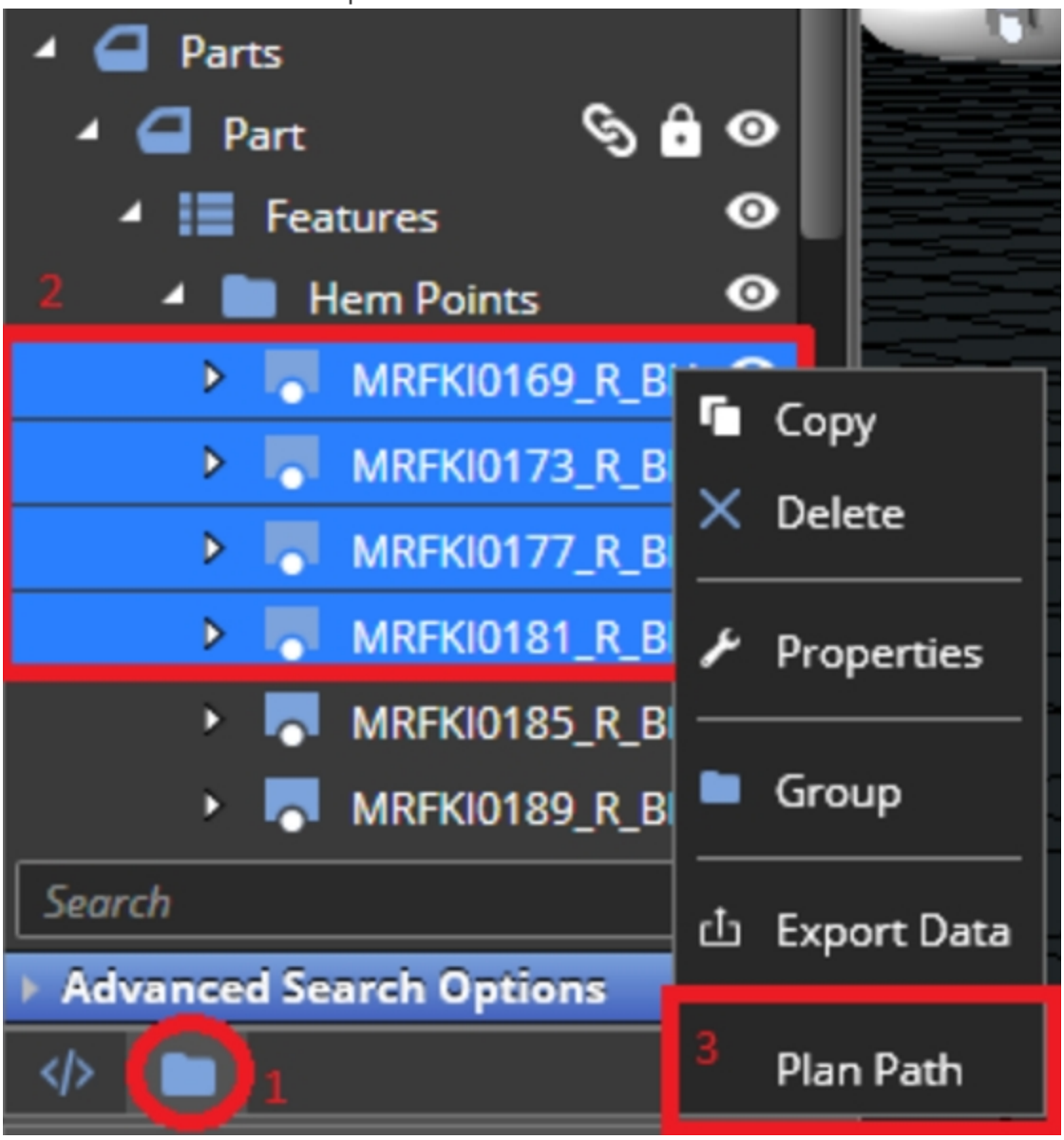
Fully automated Path planning is currently only available for Leica sensors. Once this is available for other sensors like ZG this will be noted.

## Automatic mode programming – Feature path planning

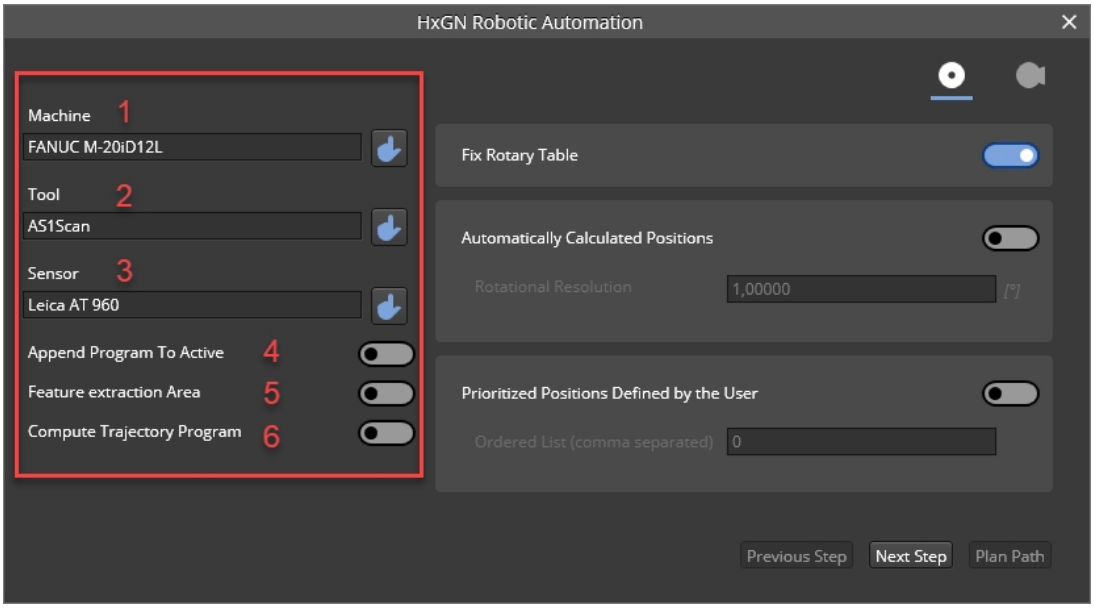
Automatic mode allows the user to automatically create a program based on the metrology features to be measured.

To plan a path, the following actions must be done:

1. Select the Project tab in the upper left window.
2. Select the features you want to measure and right click.

   Click on the **Plan Path** option.
3. 

4. A pop-up window will appear where path planning process settings can be defined. This window will be divided in various sections:
    a. Fixed SectionThis section will show on the left of any of the future sections below



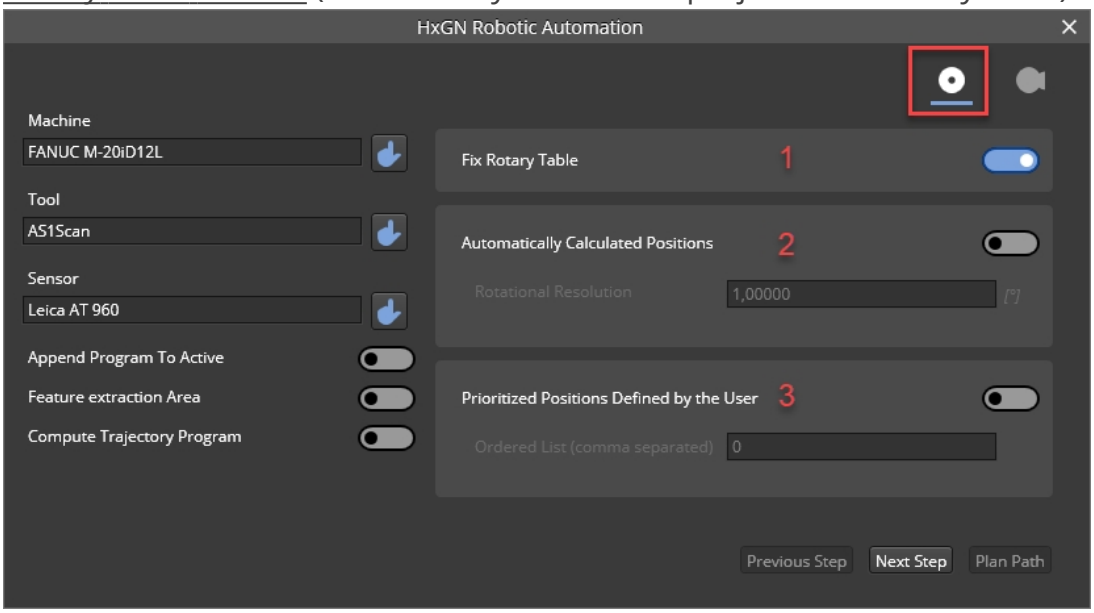

1. **Machine**: Showing the current selected machine
2. **Tool**: Showing the current selected tool
3. **Sensor**: Showing the current selected sensor
4. **Append Program to Active**:
    i. By default, it is checked out. Planning path will generate new program.
    ii. If checked in, the generated planning path will not generate a new program, instead it will append it to the Active program if there is one (the user can activate a program by selecting it

using the relative drop-down menu within the Program window). In this way the path planning algorithm will try to combine different scans covering the relative features into one single scan.

5. **Feature extraction Area:**
    i. For a patch defined by 4 points: selecting this mode, the patch scan planning will be used.See chapter Featureless_path_planning
6. **Compute Trajectory Program:**
    i. By default, it is checked out. If checked in, the user can select a list of features (e.g. edge points, surface points, circles), and the path planning algorithm will try to find the path to move between these features. See chapter Featureless_path_planning

Rotary Table Section (this will only show if the project has a rotary table)

b.

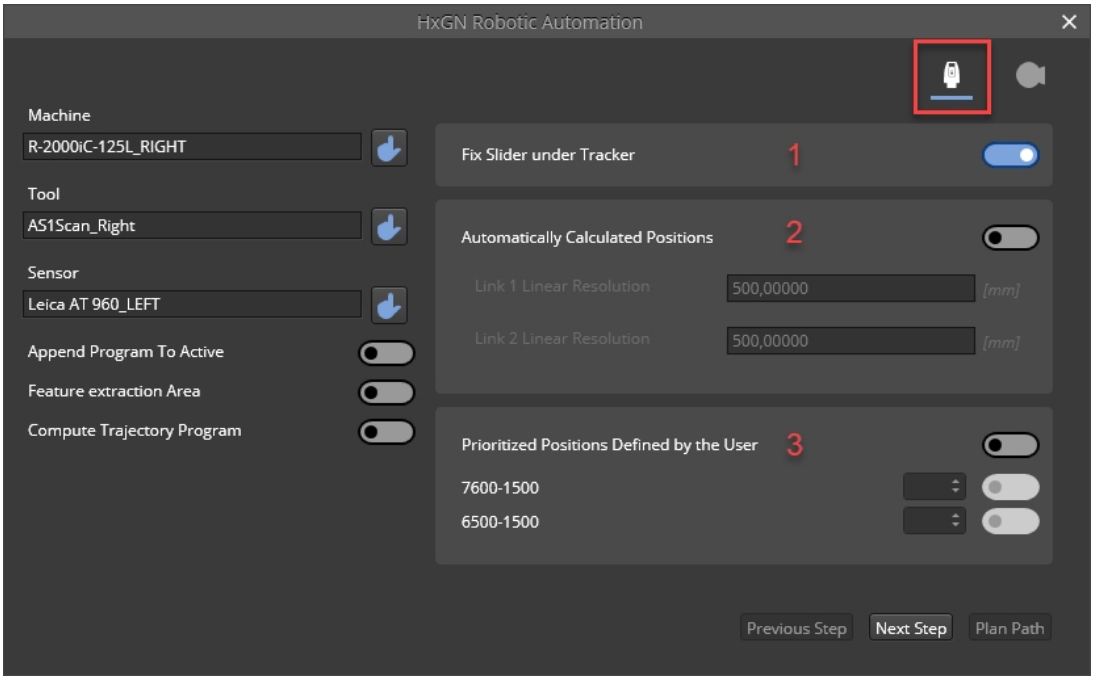


1. **Fix Rotary Table:**
    i. By default, is checked out. The planning path for the selected features will try to find best positions for the Rotary Table, with minimal positions for covering the features.
    ii. If checked in, HxGN Robotic Automation will lock the rotary table holding the part and the path planning for the selected features will be done only for the current rotary table position so the path planning algorithm will not try to find the best rotary table positions for the scan. Before clicking on Plan Path, user can change the current Rotary Table to the desired position.
2. **Automatically Calculated Positions.**
    i. The algorithm will try to find the best Rotary table positions to create the part inspection program.
    ii. Alternatively, the user can specify in the **Rotational Resolution** field the smallest angle resolution used by the algorithm to compute the positions.
    iii. E.g., if 45° is used as rotational resolution, the algorithm will allow 0°, 45°, 90°, 135°, 180°, etc. positions. If the user defines 60° as rotational resolution, only 0°, 60°, 120°, 180°, etc. positions will be allowed. By default, the path planning algorithm will consider any position between 0° and 360° with a rotational resolution of 1° thus considering all positions within that granularity (e.g., 42°, 173°, and so on).
3. **Prioritized Positions Defined by User**

i. Toggle switch: by default, it is checked out and will be exclusive with the earlier option. If selected, the user can specify within the Ordered List (comma separated) field specific positions to be used by the algorithm to create the part inspection program, ordered by priority.

ii. E.g., if the user inserts 42°, 87°, 123° degrees, the algorithm will not calculate the optimal Rotary Table positions, but it will use only the positions provided by the user (or part of them) to compute the program, trying to use first 42°, then 87° and lastly 123°.

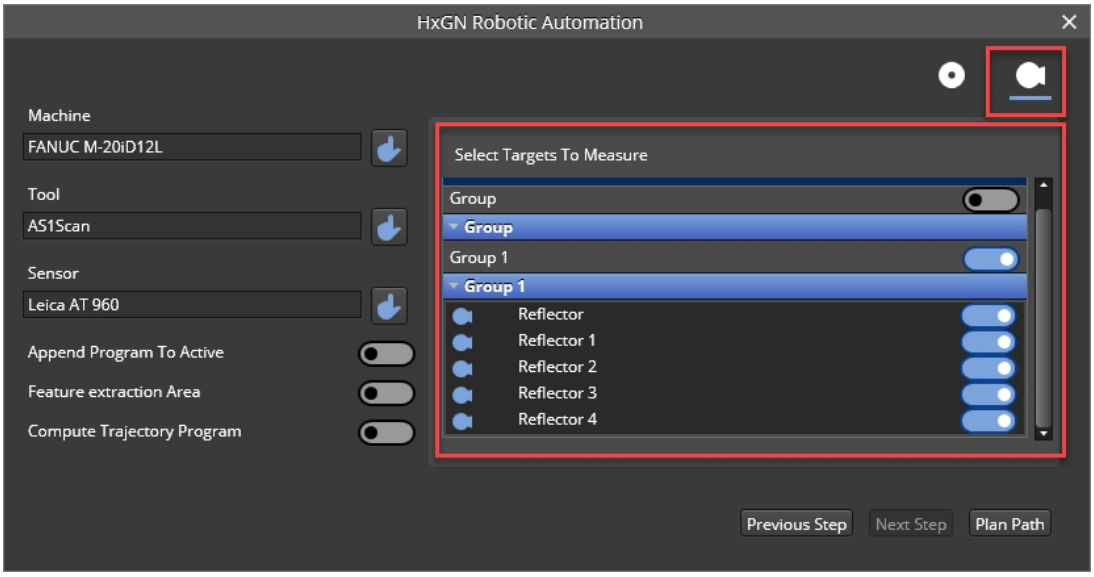
Tracker Section:(This will only show if the project has a tracker slider)

c. 

1. **Fix Slider under Tracker:**
    i. By default, is checked out. The planning path for the selected features will try to find best positions for the tracker on the slider, with minimal positions for covering the features.
    ii. If checked in, HxGN Robotic Automation will lock the slider with the tracker and the path planning for the selected features will be done only for the current slider position so the path planning algorithm will not try to find the best slider positions for the scan. Before clicking on Plan Path, user can change the current slider position to the desired position.
2. **Automatically Calculated Positions.**
    i. The algorithm will try to find the best Tracker Slider positions. same as above applies, with the difference that the resolution is now linear, and the default value is 500 mm.
3. **Prioritized Positions Defined by User**
    i. Toggle switch: by default, it is checked out and will be exclusive with the earlier option. If selected, the user can specify positions of the tracker slider can be added and in which order
    ii. The saved positions can be enabled/disabled with a toggle switch and prioritized by entering the priority for each position using the drop-down menu (the smaller the digit, the higher the priority, e.g., 1 has higher priority than 2). If two positions have the same priority, the top one will be used first. The path planning algorithm will try to generate a program that scans as many features as possible using the slider pos-

ition with highest priority and continue with other positions to scan the remaining features (if any).

Reflector Section

d. 


1. Under Select targets to measure, the user can select the reflector groups to be used for compensation. See chapter add reflector wizard for an explanation on how to create reflector groups.

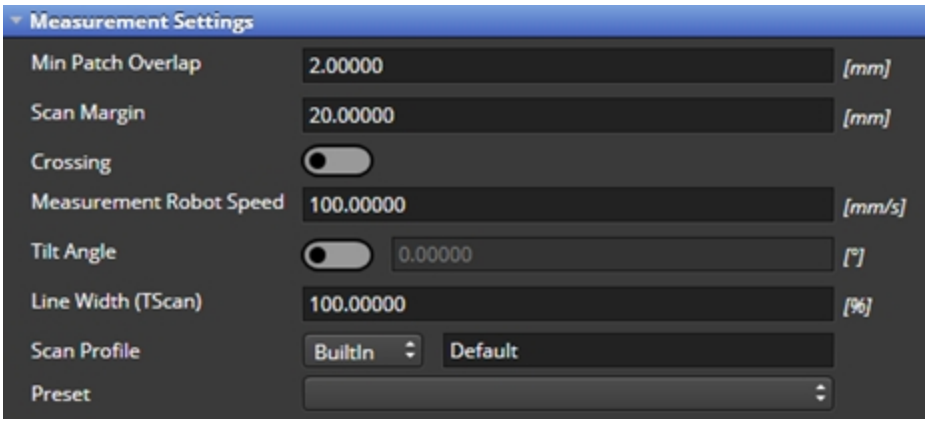
5. After defining the options for the path, the user can click on the **Plan Path** button and wait for the program to be created.

   When the program is generated, it will be visible within the Program window.

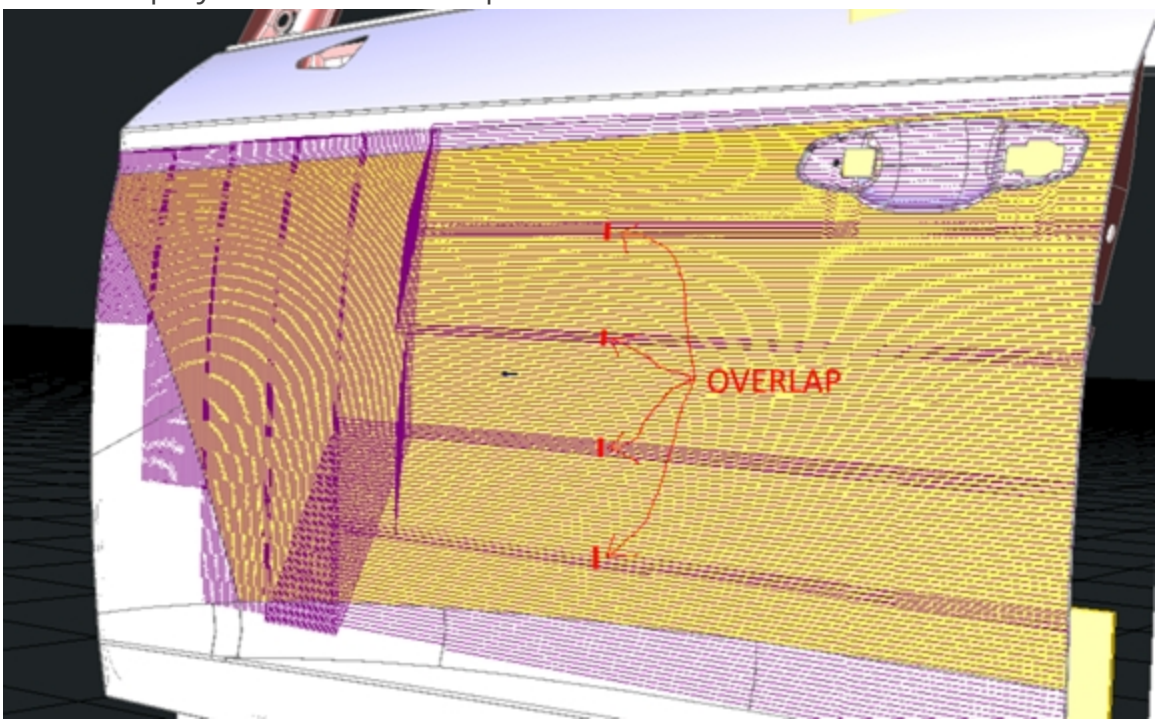
   If no solutions are found or in case of a partial solution, the information will be reported in the status window.

## Feature measurement settings

For each feature that is being measured and thus used to compute an automatic path to scan it, it is possible to right-click on it in the project tree and set the following properties:

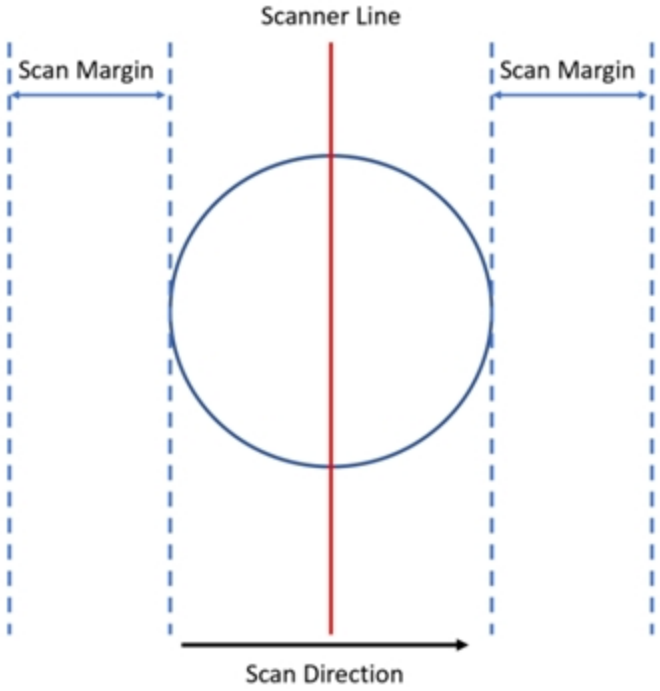



**Min Patch Overlap**: minimum overlap between parallel stripes of a scan of a feature. Note that the overlap can be bigger if more stripes are needed to cover the poly-surface with the specified minimum value.

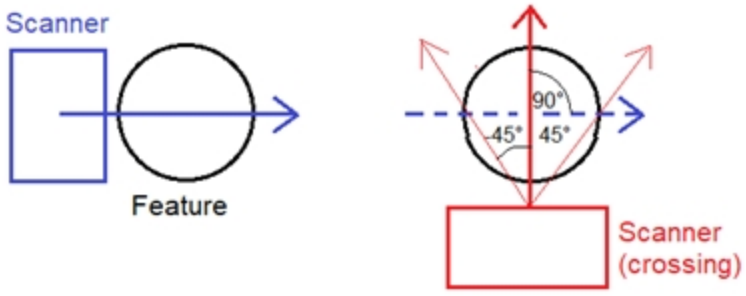
1. 


2. **Scan Margin**: the extra margin of the scan around the feature. If it is a point, the scan margin consists of a circle around the point perpendicular to the scan

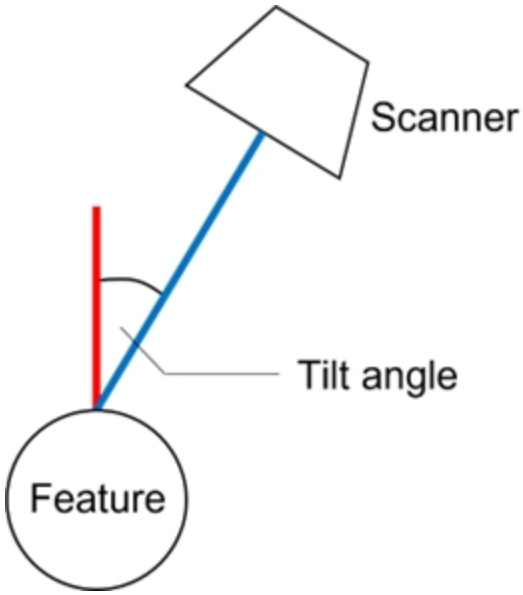
normal within which the scanning will continue to happen. If it is not a point-like feature (e.g., a shape, or a poly surface, or a circle) the margin is to be considered around the perimeter defined by the feature.




3. **Crossing**: if this is enabled, the feature is first scanned and, if possible, a second scan (the crossing) will be computed considering an angle of 90° ± 45° with respect to the direction of the first scan to use the same scanner face. For the crossing to be considered valid, the same sensor face must be used. If it is not possible to compute the crossing, only the first scan is delivered. No visual keyword in the program informs the user of the success of the crossing. It must be visually verified by the measurement "painting" over the feature.



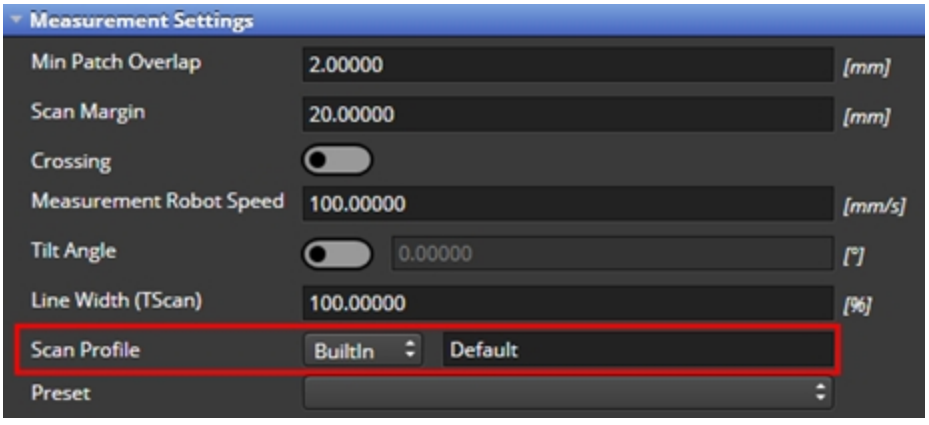

4. **Measurement Robot Speed**: the speed of the robot during the measurement of the feature, in percentage to the maximum speed:
    a. The default value is 40, meaning that the final robot speed will be set to 40% of its maximum speed.
    b. The user can only change the value to 100, meaning that the final robot speed will be set to 100% of its maximum speed.
5. **Tilt Angle**: it allows the user to specify that the feature is intended to be scanned with a specific tilt along the scan direction as depicted in the picture below:
    a. the red line is the feature normal.
    b. the blue line is the feature normal with the tilt.

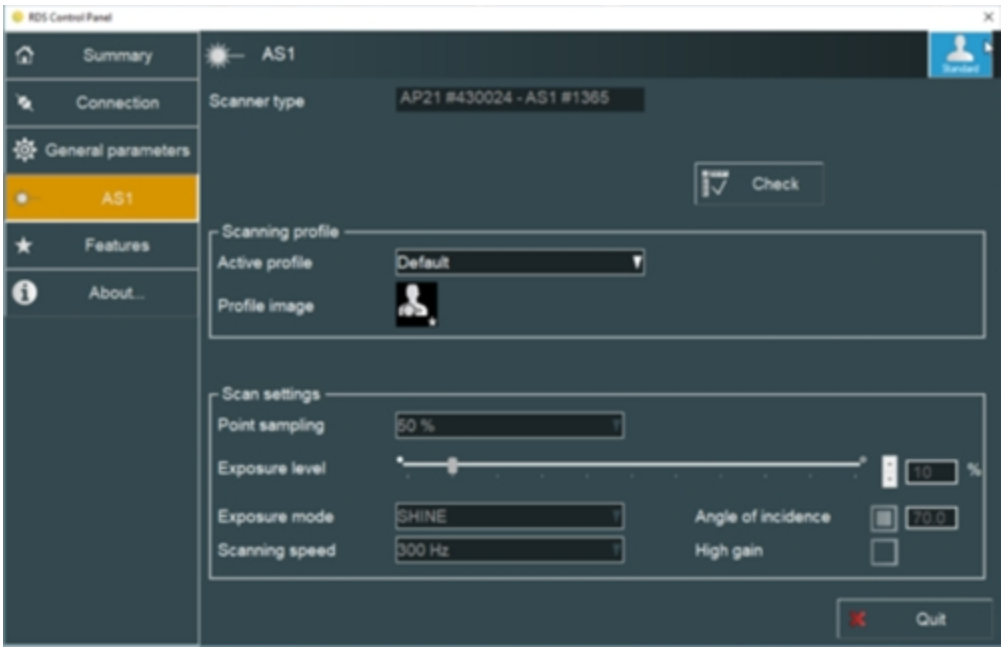



c.

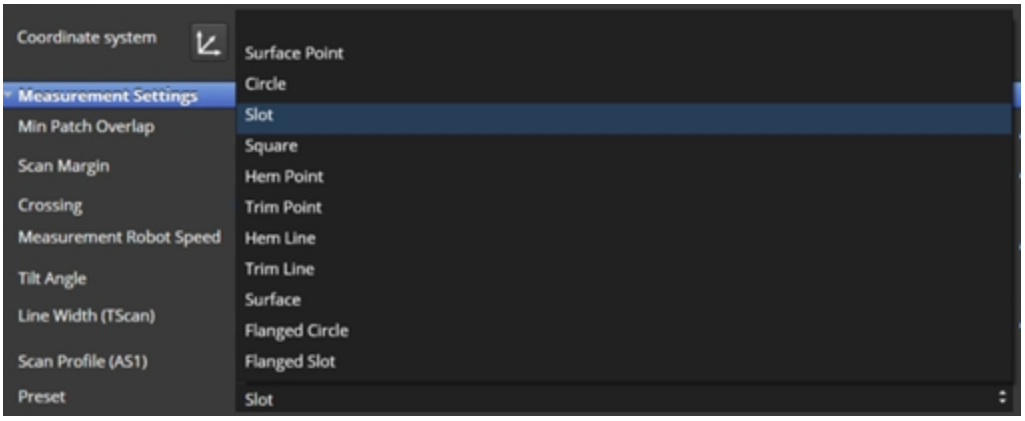
A set of scans is generated over a feature. If the scan with the feature cannot be computed with the tilt angle defined by the user, other tilt values will be tried by the algorithm to provide a solution.

6. **Line Width** (T-Scan): allows the user to set the T-Scan 5 line width (from 40% to 100%).
7. **Scan Profile** (AS1): allows the user to activate a Scanning profile defined in RDS.






**Preset**: allows to specify values for the parameters above, grouped under a specific label that can be defined within the HxGN Robotic Automation settings

8. 


## Save fixed positions

It is possible to define fixed position to reduce the range of positions of external axes that the automatic path planning will consider creating the robot program.

### Path planning algorithm flowchart

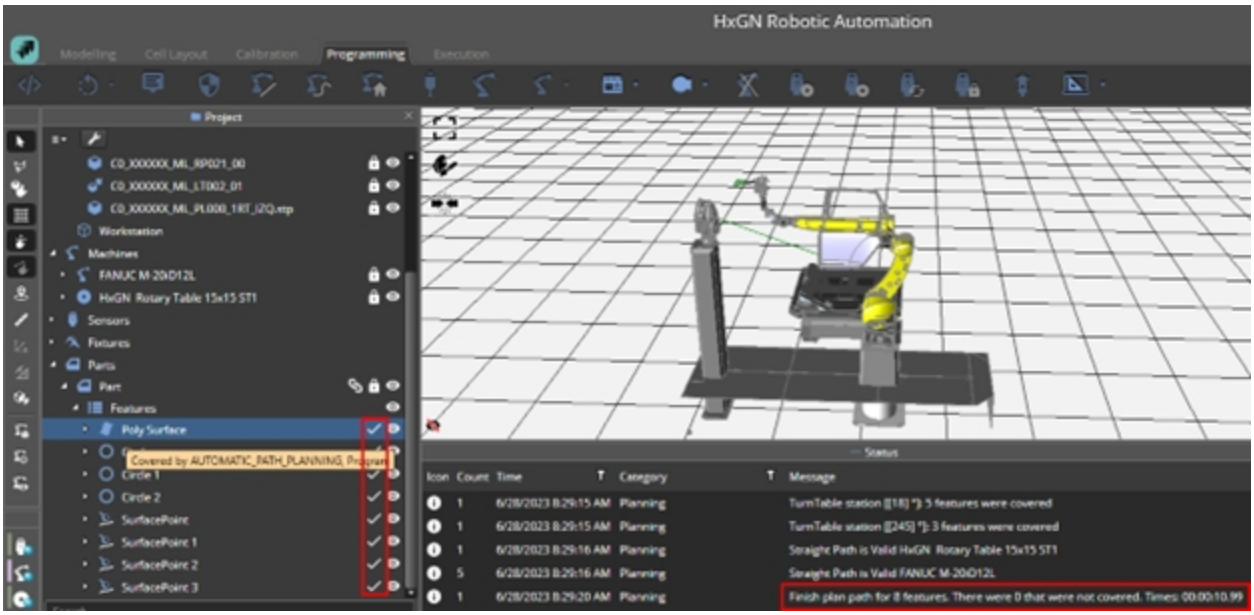
After the user selects the features and click on plan path after specifying all the required settings as explained in the earlier chapters, the path planning algorithm will create a program according to the flowchart depicted in Figure Automatic path planning flowchart, that can be summarized in the following steps:

1. Homing motions are added for all the machines in the cell to start the program from the starting positions.
2. Commands to measure the nominal position of the reflectors are added.
3. The command groups to measure the features are created. In case it is not possible to scan the features from the current machines' positions, the machines are moved to a convenient position and the actual positions of the reflectors are measured.
4. After all groups are created and all features are covered, the groups are sorted to supply a best sequence of grouping to have the fastest scan of all the features.

5. If it not possible to move the machines between any groups, more movements are added to allow it.
6. Homing motions are added for all the machines in the cell to end the program correctly by sending the machines back to the starting positions.
7. In case some features cannot be covered by the path planning, the user is informed both in the status window and in the project, tree

### Evaluate the quality of the scan

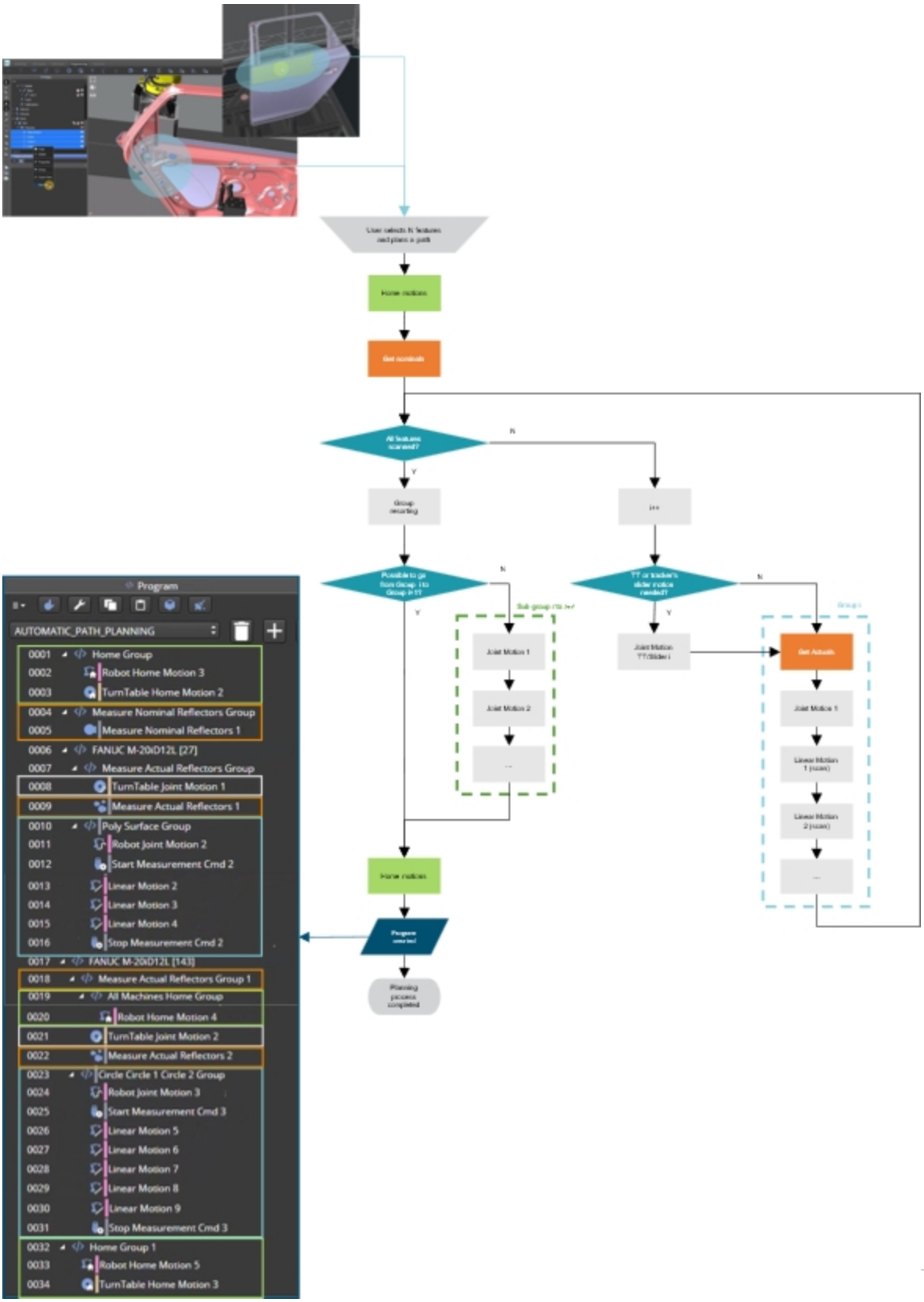
After the program generated by the automatic path planning is validated and executed, it is possible to evaluate if the features were properly covered, i.e., if enough points were collected during the scan of the features. The automatic path planning algorithm tries to scan a feature in the best way. In case the features are scanned properly and covered properly, a checkmark icon is displayed near the feature.




If this is not possible to cover the feature, the automatic path planning tries to scan it with lower-quality scans. This situation can happen e.g., measuring edge points that could be scanned not with the highest quality.

In this case the user can decide to adjust the scans to cover the partially covered feature or try to execute the feature computation in the metrology SW with the acquired data.

The concept of coverage is not to be confused with a metrology computation on the features and provides only a sign on COP quality for a specific feature. Any metrological analysis must be done in the metrology software used in the project (see chapter Execution.htm).


Program
AUTOMATIC_PATH_PLANNING
0001 Home Group
0002 Robot Home Motion 3
0003 TurnTable Home Motion 2
0004 Measure Nominal Reflectors Group
0005 Measure Nominal Reflectors 1
0006 FANUC M-20iD12L [27]
0007 Measure Actual Reflectors Group
0008 TurnTable Joint Motion 1
0009 Measure Actual Reflectors 1
0010 Poly Surface Group
0011 Robot Joint Motion 2
0012 Start Measurement Cmd 2
0013 Linear Motion 2
0014 Linear Motion 3
0015 Linear Motion 4
0016 Stop Measurement Cmd 2
0017 FANUC M-20iD12L [143]
0018 Measure Actual Reflectors Group 1
0019 All Machines Home Group
0020 Robot Home Motion 4
0021 TurnTable Joint Motion 2
0022 Measure Actual Reflectors 2
0023 Circle Circle 1 Circle 2 Group
0024 Robot Joint Motion 3
0025 Start Measurement Cmd 3
0026 Linear Motion 5
0027 Linear Motion 6
0028 Linear Motion 7
0029 Linear Motion 8
0030 Linear Motion 9
0031 Stop Measurement Cmd 3
0032 Home Group 1
0033 Robot Home Motion 5
0034 TurnTable Home Motion 3

## Automatic mode programming – Featureless path planning

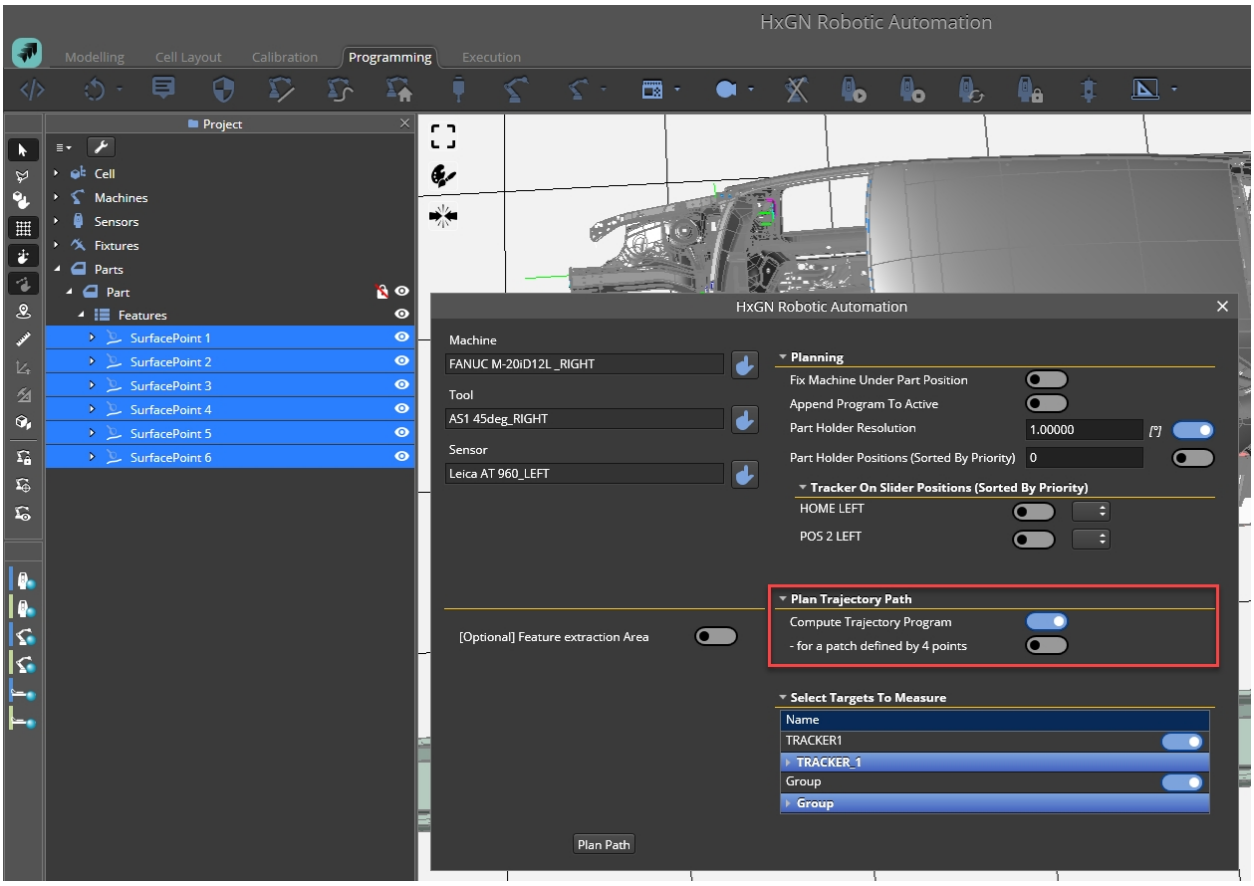
The Automatic mode using featureless path planning allows the user to automatically create a program based on a set of via points defined by the user instead of using metrology features to calculate the path.

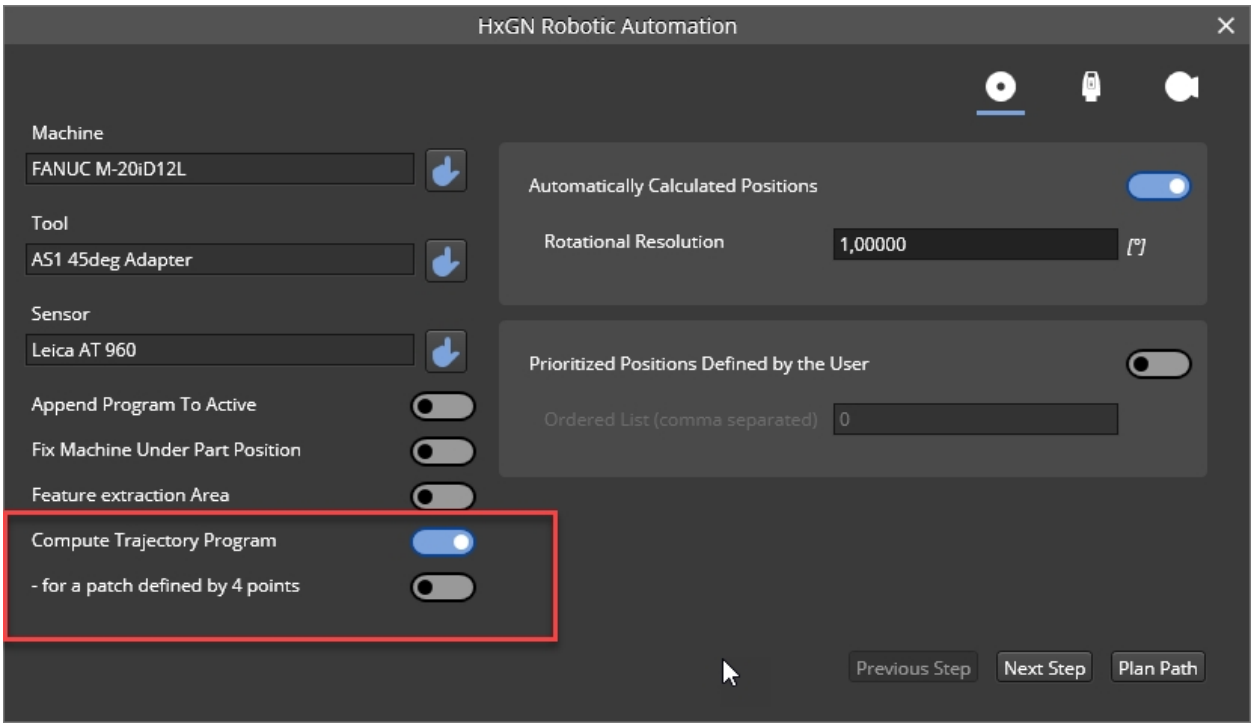
The via points can be defined in two ways:

1. As a piecewise trajectory that the scanner needs to follow to perform the measurement (trajectory scanning) as described in chapter _Trajectory_scanning
2. As four points defining a rectangle that the scanner needs to cover (patch scanning) as described in chapter Patch_scanning

### Trajectory scanning

The same procedure described for the automatic path planning mode can be followed but the features in this this case are defined by the user by inserting features (in this example surface points) with the feature generation mode (see chapter Feature Generation Mode) and planning a path on those points by selecting them from the project tree, right click and then select the **Compute Trajectory Program** option within the Planning dialog box as depicted in the image below.


HxGN Robotic Automation
Modelling
Cell Layout
Calibration
Programming
Execution
Project
Cell
Machines
Sensors
Fixtures
Parts
Part
Features
SurfacePoint 1
SurfacePoint 2
SurfacePoint 3
SurfacePoint 4
SurfacePoint 5
SurfacePoint 6
HxGN Robotic Automation
Machine
FANUC M-20iD12L_RIGHT
Tool
AS1 45deg_RIGHT
Sensor
Leica AT 960_LEFT
Planning
Fix Machine Under Part Position
Append Program To Active
Part Holder Resolution
1.00000
Part Holder Positions (Sorted By Priority)
0
Tracker On Slider Positions (Sorted By Priority)
HOME LEFT
POS 2 LEFT
Plan Trajectory Path
Compute Trajectory Program
- for a patch defined by 4 points
[Optional] Feature extraction Area
Select Targets To Measure
Name
TRACKER1
TRACKER_1
Group
Group
Plan Path



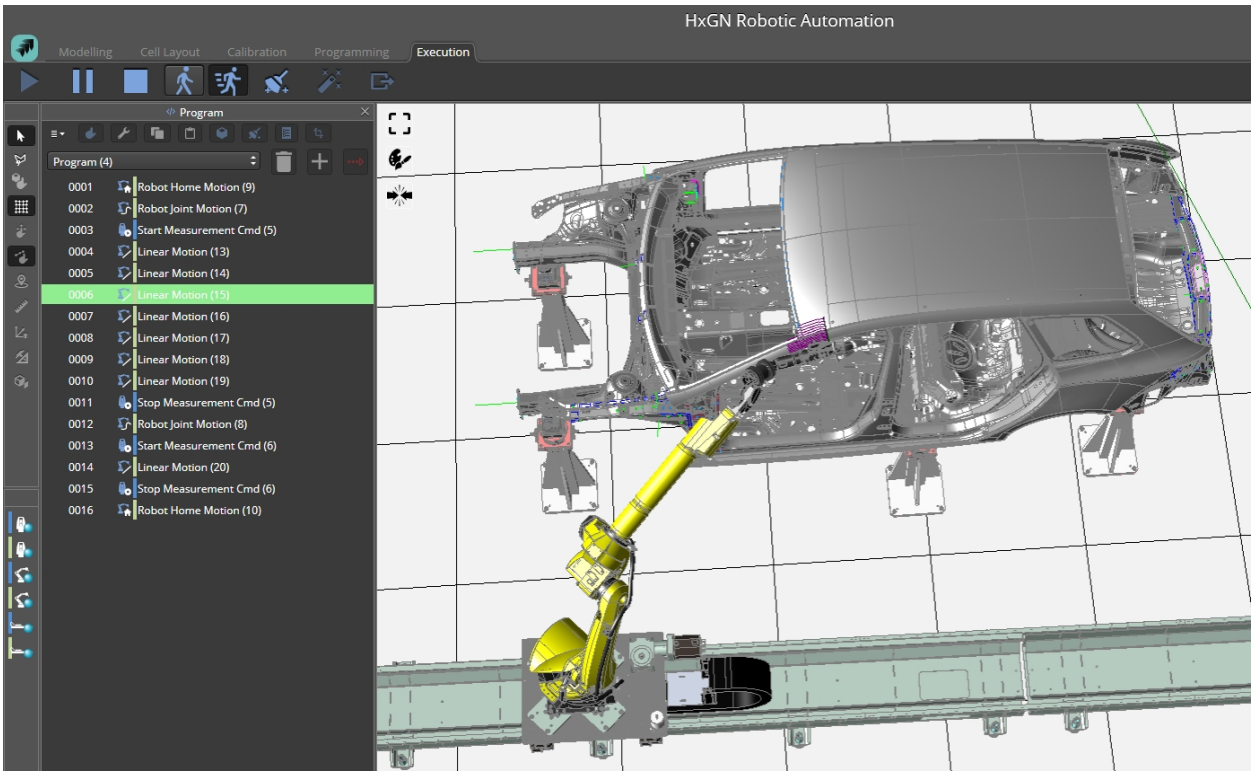

As explained in chapter Feature_path_planning, the user can configure the algorithm to automatically compute the best rotary table position or the best tracker slider position to cover the expected surface. It is also possible to use positions predefined by the user.

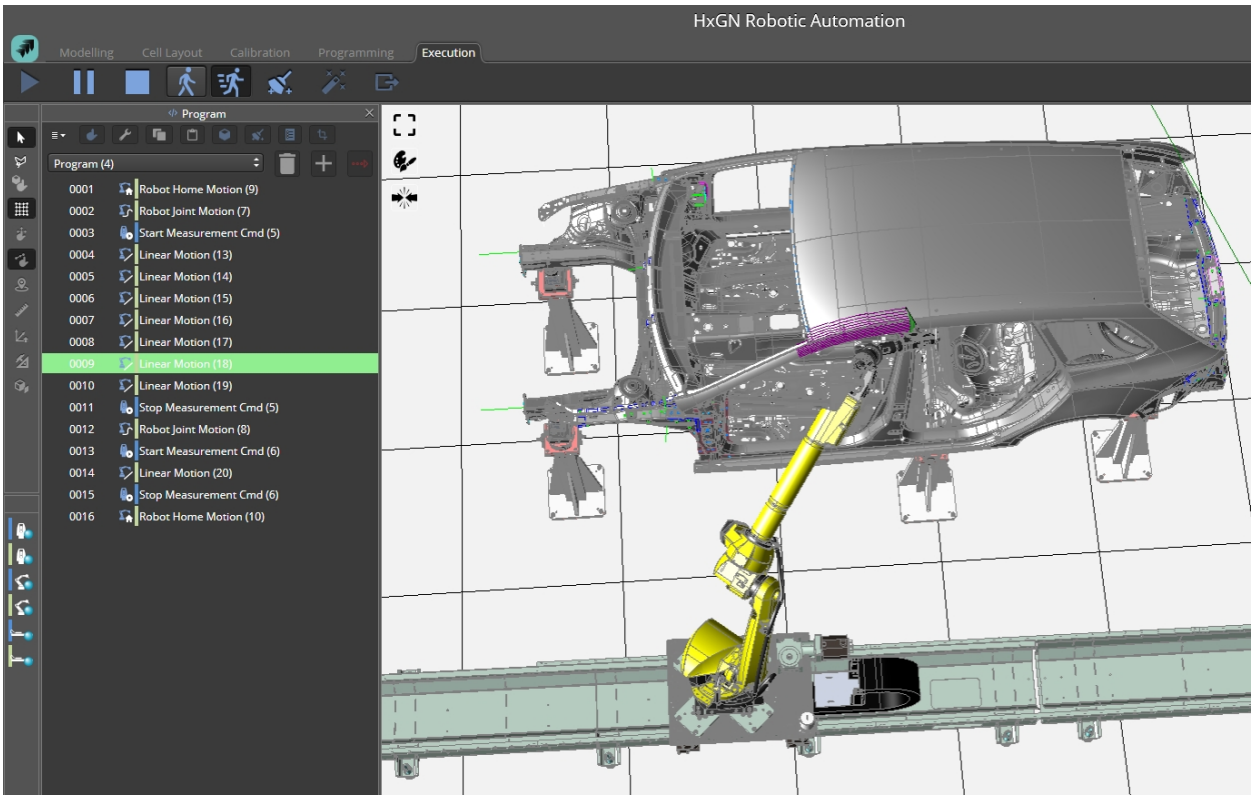
The simultaneous calculation of both rotary table and tracker slider best positions is currently not available.

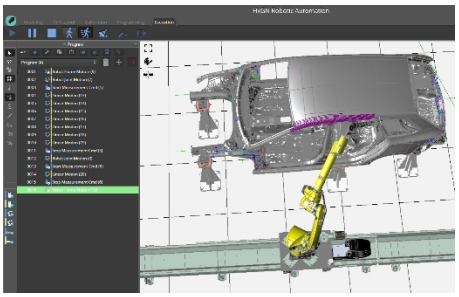
If the project has a tracker on a slider, the user can also configure the algorithm to use a set of predefined slider poses for the tracker slider. After clicking on Plan Path button, it is just needed to wait for the program to be created as for automatic mode programming. When the program is generated, it will be visible within the Program window. If no solutions are found, the information will be reported in the status window.

The path planning algorithm will generate a measurement program having joint and linear motions to scan the area defined by the four surface points as well as homing motions and start/stop measurement commands to execute the scanning.

In the following pictures is shown how a trajectory scanning program will look like for the list of six surface points defined above.

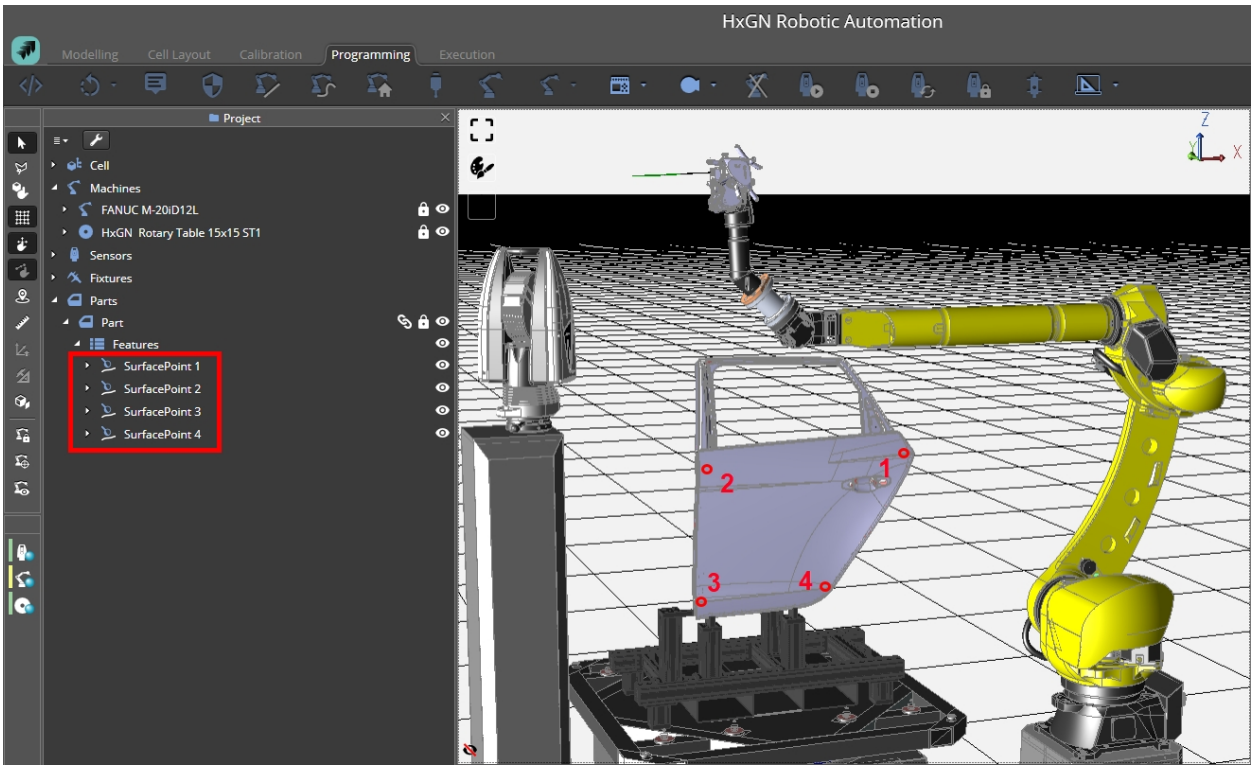

HxGN Robotic Automation
Modelling
Cell Layout
Calibration
Programming
Execution
Program
Program (4)
0001 Robot Home Motion (9)
0002 Robot Joint Motion (7)
0003 Start Measurement Cmd (5)
0004 Linear Motion (13)
0005 Linear Motion (14)
0006 Linear Motion (15)
0007 Linear Motion (16)
0008 Linear Motion (17)
0009 Linear Motion (18)
0010 Linear Motion (19)
0011 Stop Measurement Cmd (5)
0012 Robot Joint Motion (8)
0013 Start Measurement Cmd (6)
0014 Linear Motion (20)
0015 Stop Measurement Cmd (6)
0016 Robot Home Motion (10)


HxGN Robotic Automation
Modelling
Cell Layout
Calibration
Programming
Execution
Program
Program (4)
0001 Robot Home Motion (9)
0002 Robot Joint Motion (7)
0003 Start Measurement Cmd (5)
0004 Linear Motion (13)
0005 Linear Motion (14)
0006 Linear Motion (15)
0007 Linear Motion (16)
0008 Linear Motion (17)
0009 Linear Motion (18)
0010 Linear Motion (19)
0011 Stop Measurement Cmd (5)
0012 Robot Joint Motion (8)
0013 Start Measurement Cmd (6)
0014 Linear Motion (20)
0015 Stop Measurement Cmd (6)
0016 Robot Home Motion (10)



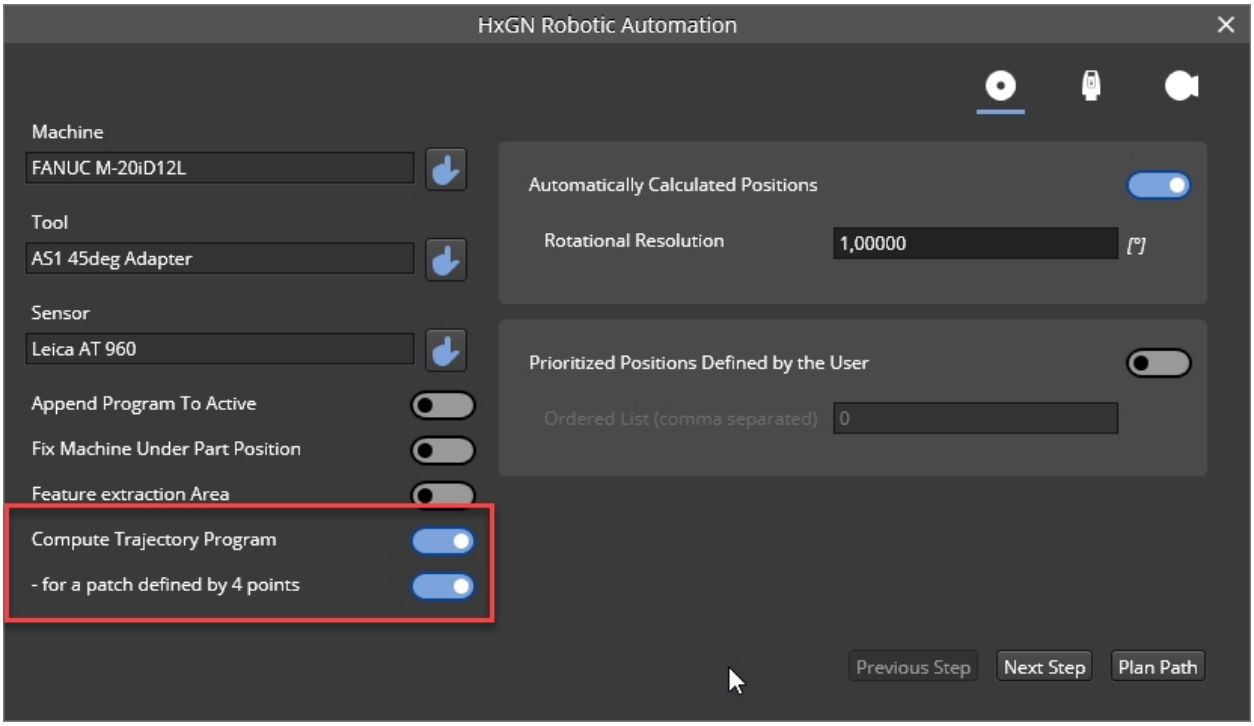
## Patch scanning

This programming mode allows HxGN Robotic Automation to create a trajectory path planning to scan a surface defined by four surface point features.

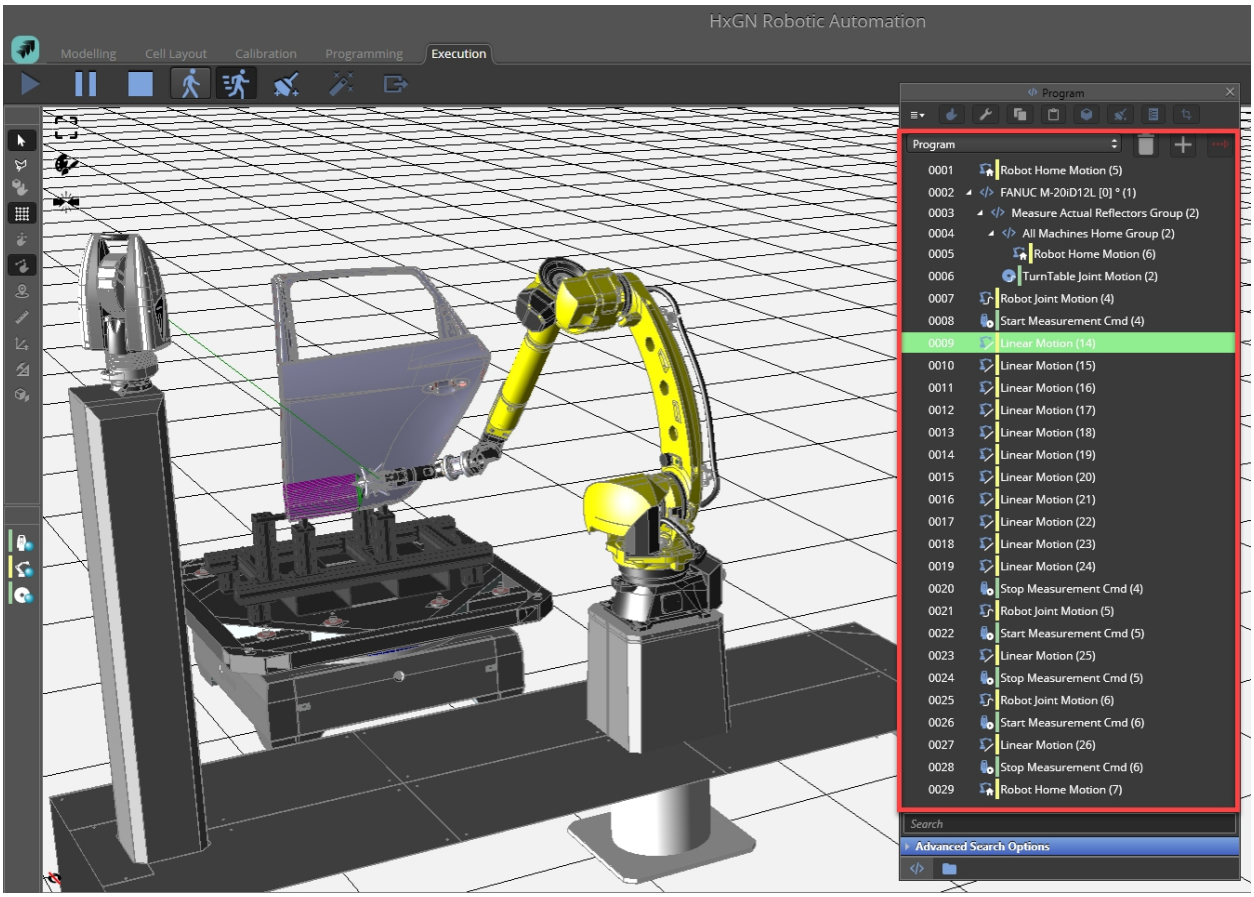



The same procedure described in the previous chapter can be followed but the features in this this case are defined by the user by inserting surface points defining a four-sided polygonal area and generating a program to scan the area by selecting the points from the project tree, right click and then select the **Compute Trajectory**

**Program** option (as done in the previous chapter) but in this case also checking – **for a patch defined by 4 points** within the Planning dialog box as depicted in the image below.

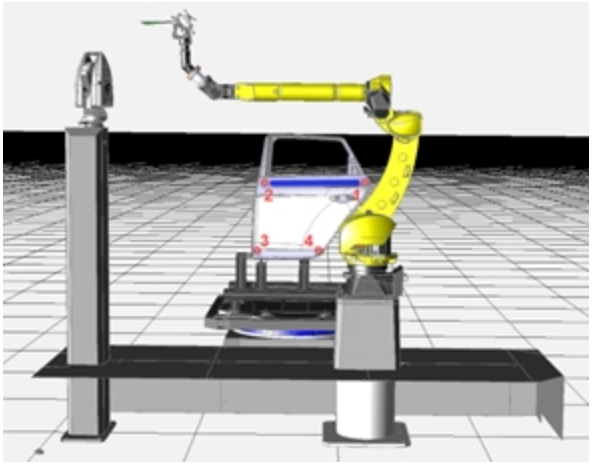
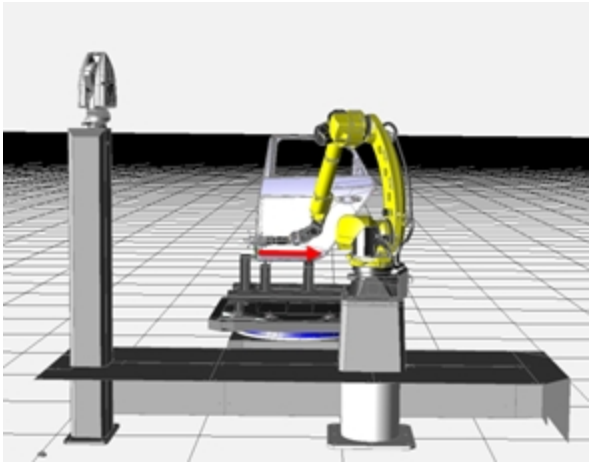



The path planning algorithm will generate a measurement program having joint and linear motions to scan the area defined by the four surface points as well as homing motions and start/stop measurement commands to execute the scanning.

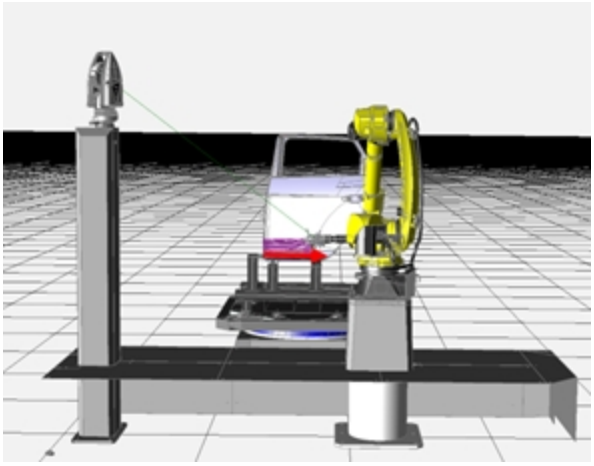
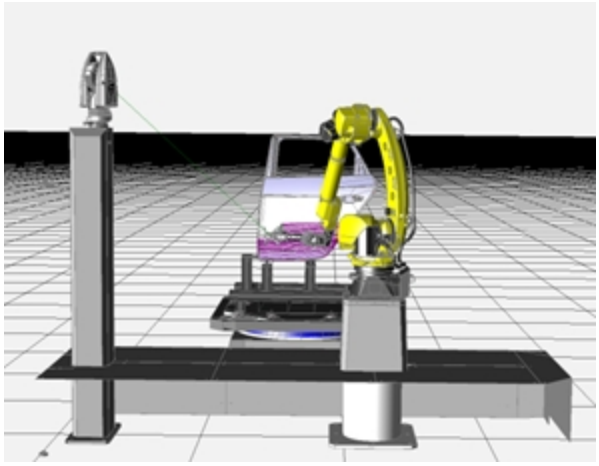


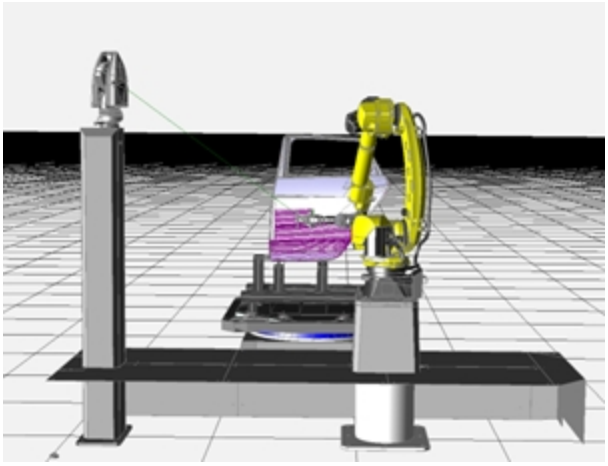

The user must be aware that the scan trajectory orientation is decided by the line defined by the first two points (e.g., “SurfacePoint 1” and “SurfacePoint 2” as shown in the example below, colored in blue).

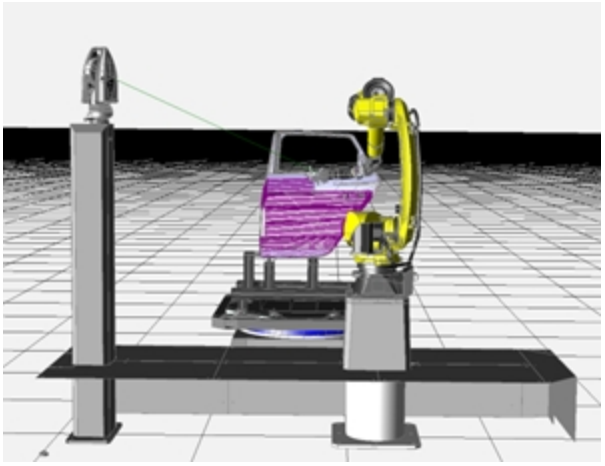
The starting point and the starting direction of the generated program is decided by the algorithm to start from the closest point to the robot starting position (e.g., from “SurfacePoint 3” in the example below).

In the following pictures is shown how a patch scanning program will look like for the list of four surface points defined above.





**Pro Tip**: This approach is a clever way to exploit the high point density of the AS1 scanner in a “featureless” approach where the features are not considered explicitly as in the automatic path planning but rather covered implicitly by having patch scanning which is agnostic on the features contained in the area and will result in the fastest execution time.

## Other Path Planning modules

### Mirroring a program

To enable program mirroring, you will need to set the following feature flag in the RAConfig file:

RAConfig.XML

```
<ConfigSetting Name="EnableMirrorProgram" Value="True" DefaultValue="False" Dis-
playName="" Description="" TypeName="System.Boolean" />
```

Mirroring a program will copy the robot and tracker movements from one side of the mirroring plane to the other side of the mirroring plane. It will then create a new

program with the 2nd robot and 2nd tracker to execute a mirrored motion based on the original program.

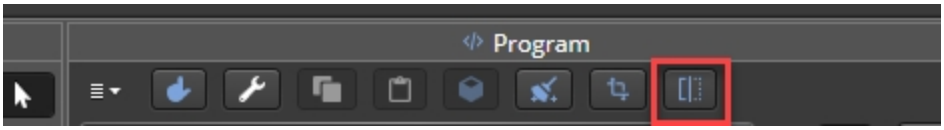
#### Creating the mirroring plane

In the **modelling tab** the mirroring plane can be created. Before clicking the mirroring plane button , there are some extra options to take into account.

- The mirroring plane will have the following properties:
    - Width of 1 mm and height and length of 2 m
    - Transparent
    - No collisions with cell objects
    - **There can only be 1 per project**
- The mirroring plane will be created in the following location:
    - If a CAD has been selected when clicking the button: 1 m above the CAD in the same CAD origin in the Zx axis.
    - If no CAD has been selected, at the 0,0,0 of the cell coordinates.

#### Mirroring a program

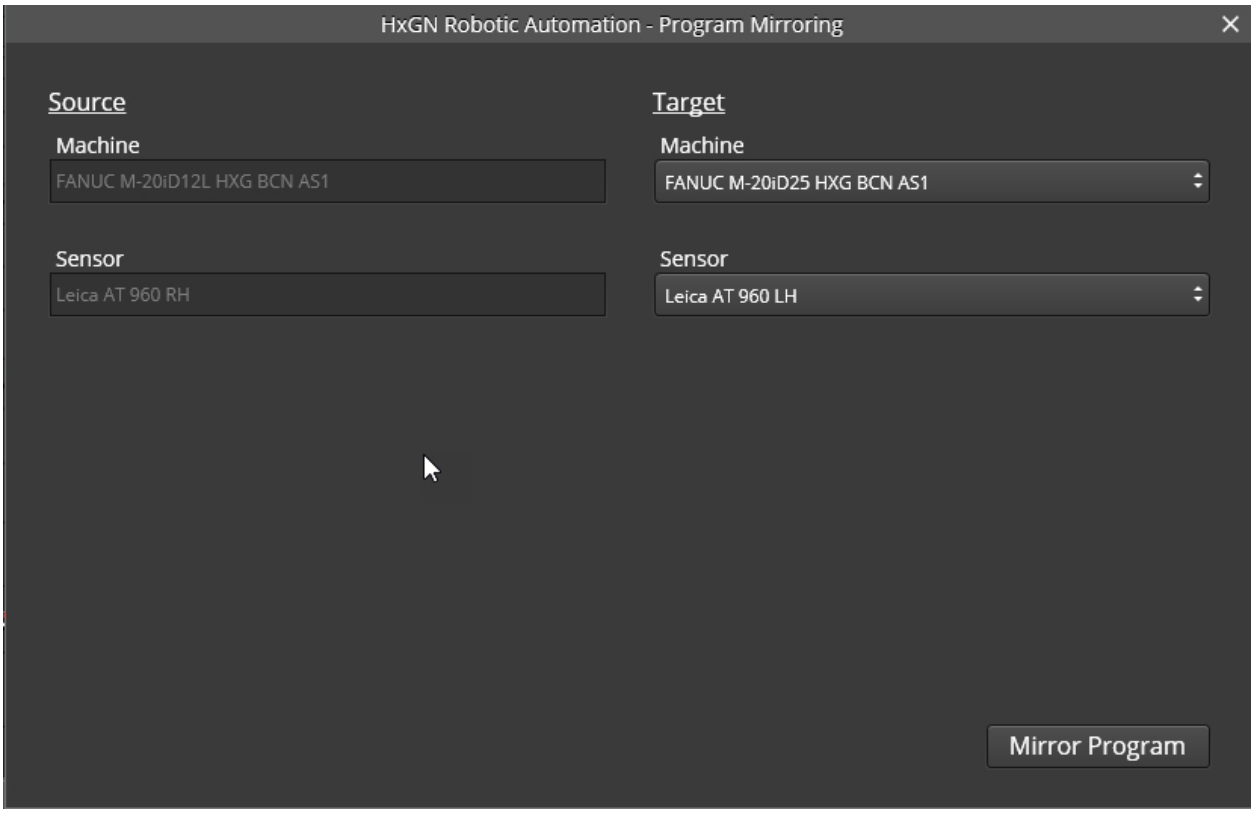
In the programming tab there will be an icon to mirror the current program.




- The program to be mirrored can only contain commands for 1 active tracker and 1 active robot (home motion commands for the other robot and tracker slider are allowed)

- The mirrored program will be created as a copy of the current program. All motions that cannot be mirrored (e.g. out of range) marked as: Failed to mirror with the previous motion coordinates in the motion target.

Only in case of more than 1 robot or tracker in the cell a menu will appear after clicking the button:




Here the mirroring target can be selected.

## Reversing a program

To enable program reversing, you will need to set the following feature flag in the RAConfig file:

RAConfig.XML

```
<ConfigSetting Name="EnableReverseProgram" Value="True" DefaultValue="False" Dis-
playName="" Description="Use ActiveBlock" TypeName="System.Boolean />
```

Reversing a program can be triggered by clicking the reverse program button in the programming tab

It will reverse the program order of the current program taking into account that reversing will work at a robot movement level.

E.g. If a program ends:

- ....
- Linear Motion
- Stop Scan
- Home Motion

It will be revered to:

- Home Motion
- Robot Joint Motion (To arrive at the end of the linear scan)
- Stop Scan Commented out
- ....

By commenting out the program logic of start stop scanning and reflector measurement, HxGN Robotic Automation can ensure that nearly any program can be copied rather than having to adhere to a strict programming logic.

It is then up to the user to adjust the commented scan and reflector measurement commands to their proper location.

---

# Validation and Simulation

From version 2024.2 of HxGN Robotic Automation the software is using a new approach to validating and simulate a program. This new approach allows for a faster validation and more flexible simulation and is broken up into 2 separate components.

1. A direct ad hoc validation on any command or syntax errors.
2. A program simulation that is build and that will be used to:
    a. Check for program issues (loosing tracker lock on, collisions, etc.)
    b. Ability to play this simulation like a video

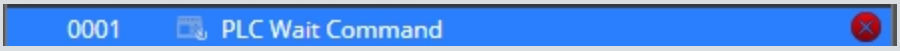
The user workflow would be the following:

1. While creating a program the ad hoc program command validation will highlight any errors in syntax of specific commands.
2. Once you want to either run the program or click the build program simulation button, the software will check for any error in the program as a whole and have the simulation ready to play.
3. The simulation can then be played from either the program tab or the execution tab.
4. Before running a program in real, we recommend running the simulation accuracy in precise mode

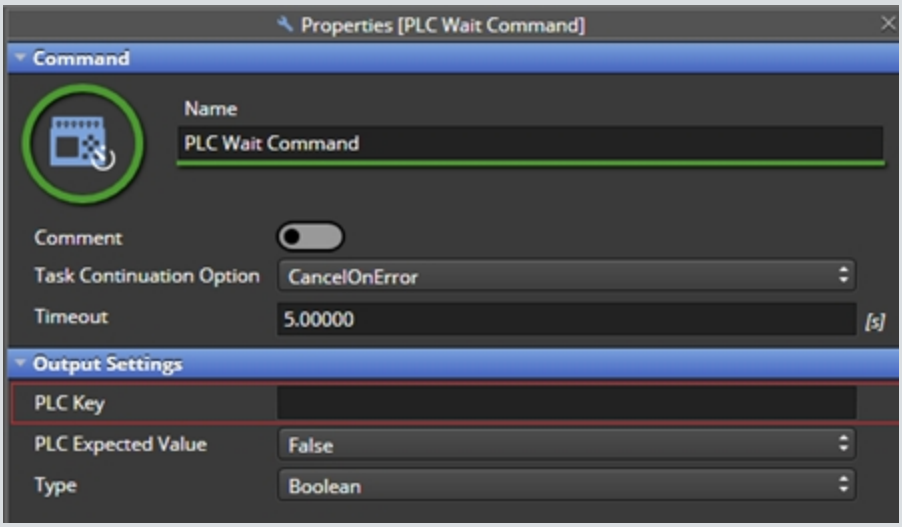
## Direct Command Validation

The software will constantly check the added program commands for any missing information. For example, if we add a “PLC Wait Command”, the command will immediately be flagged as in error as there will be missing information.

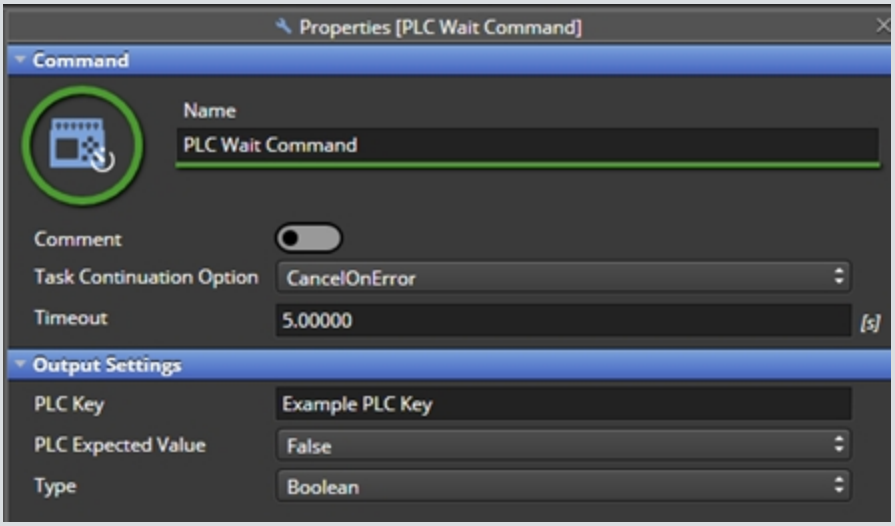
Example: Adding PLC Wait Command



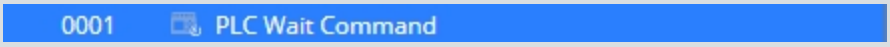

With missing data:




Once we add the missing information:




The red icon will disappear:



## Simulation Errors and Warnings

Simulation errors will block the program execution. They will range from values being out of range, potential collisions, or unsupported use of commands.

A program with errors can be run in simulation but cannot be executed in a real cell!

Simulation warnings are all the other issues that can affect the Point Cloud or quality of the program but will not cause any collision or issue when running the program in a real cell.

A program with warnings can be run in simulation and in the real cell!

## Simulation Error List

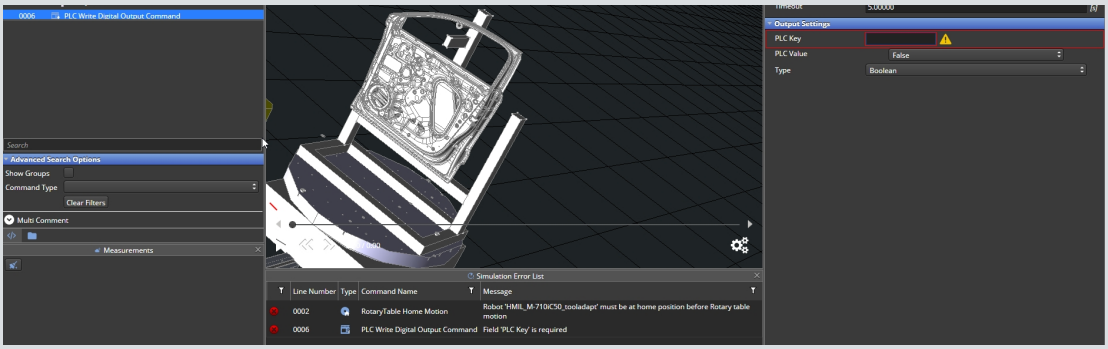
All the Simulation errors will appear in the simulation error list. A new tab that has been implemented that will show all the errors and warnings.

Any issue found in the direct command validation will appear and (and disappear once fixed) from the list immediately.

Any issue that is found from the simulation will appear in the Simulation Error List but will only disappear once fixed and the Simulation has been re-created.

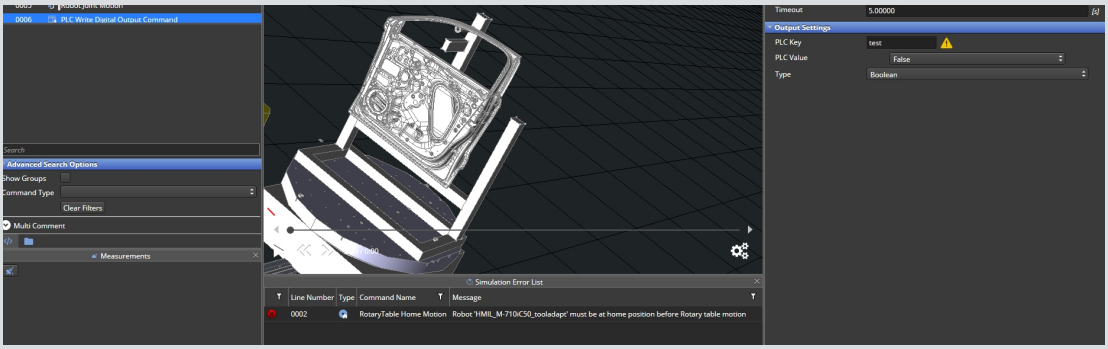
Example: Direct command validation

Add a command with missing information:

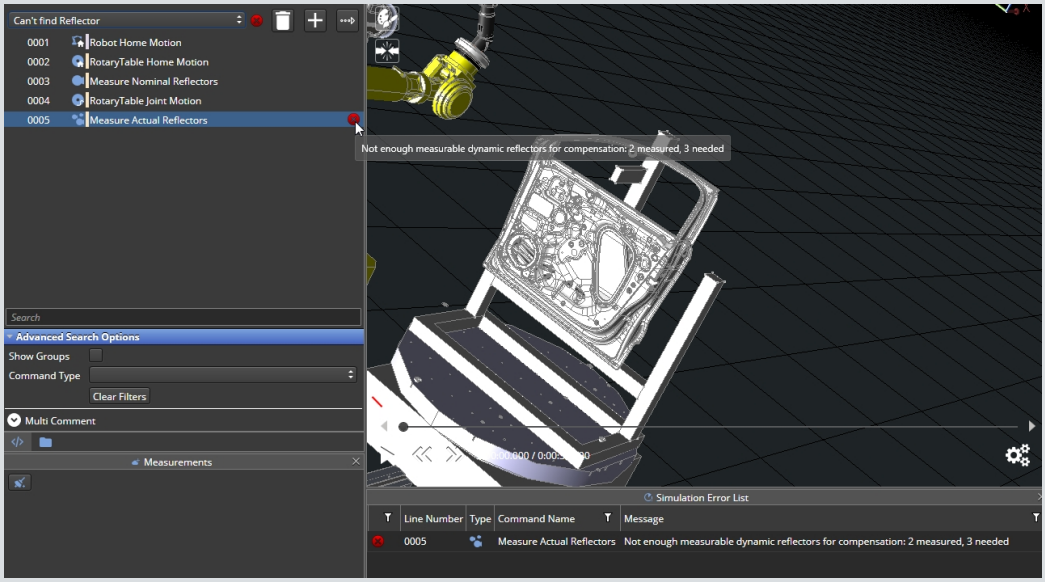


The error appears in the Simulation Error List

Add the missing information and the error disappears from the list.



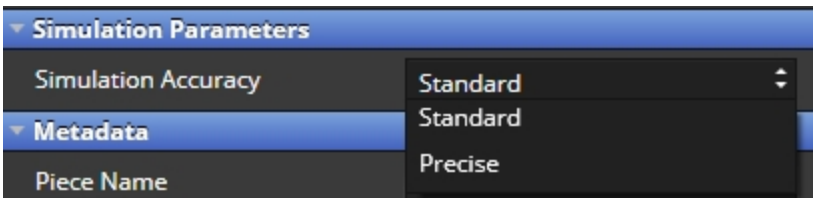

Example; Error detected during simulation



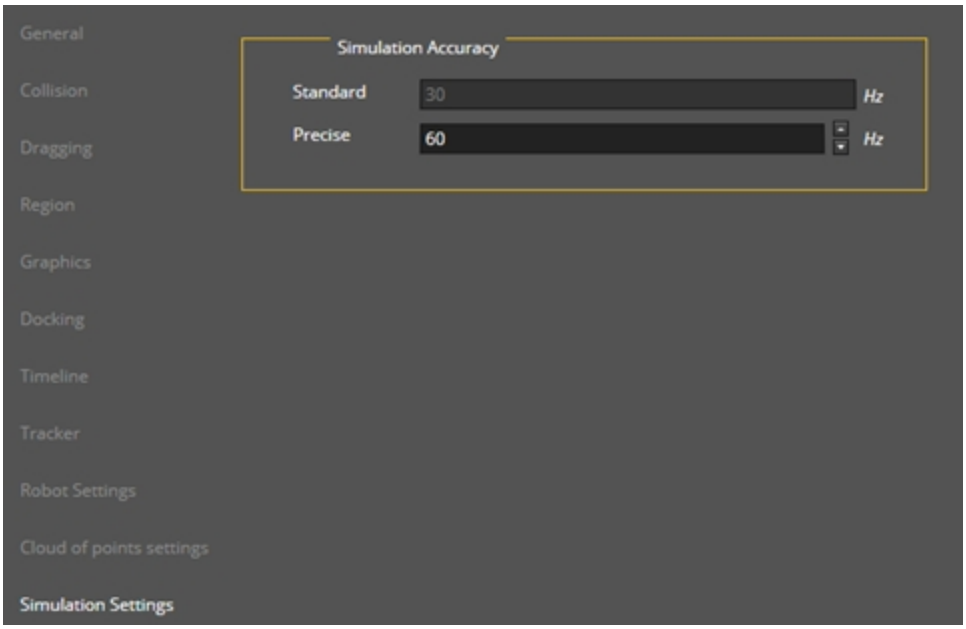
## Program Simulation Accuracy

In the program properties a new header has been added where the program simulation accuracy can be set.

The default value is standard. For fast robot movements or before testing a program in a real cell, it is recommended to run the simulation on precise mode at least once.




In the HxGN Robotic Automation settings, the values of standard and precise can be adjusted, by either a number between 60 or 990 Hertz (least to more accurate, but also taking more calculation time).



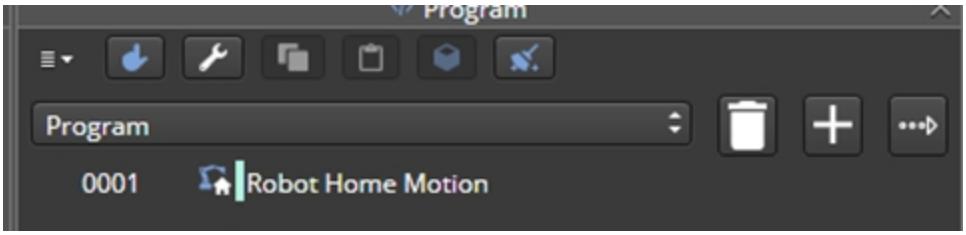

## Build Program Simulation

The simulation building will occur in 3 different scenarios:

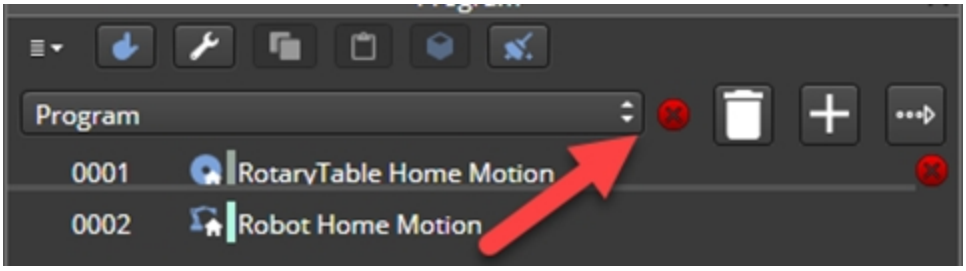
1. On demand when clicking the “Build Program Simulation” button
2. When clicking run simulation, and the program did not build the simulation yet.
3. When clicking the send program data to Metrology button, and the program did not build the simulation yet.

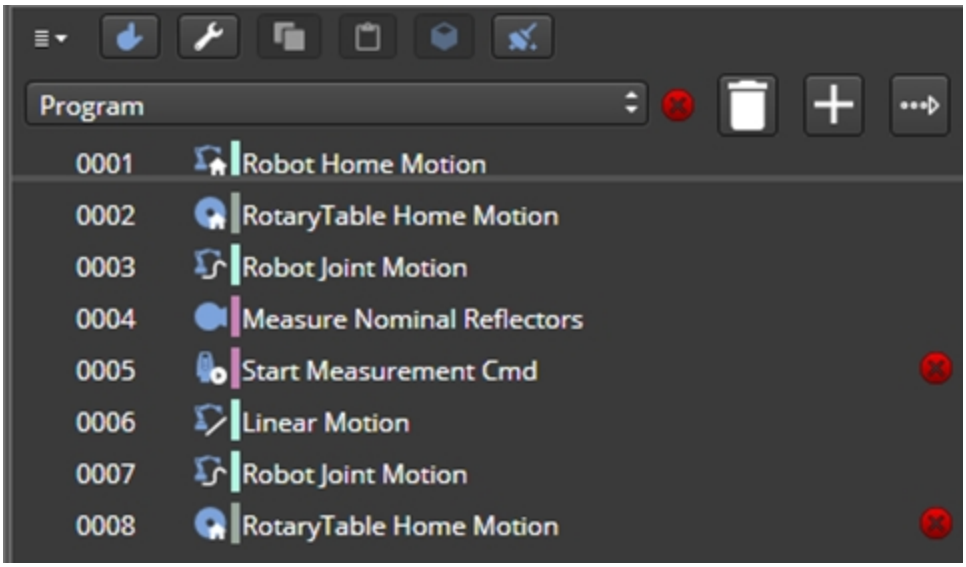
The build simulation button will leave the program in 3 states:

1. Program is newly created; no simulation has been built yet

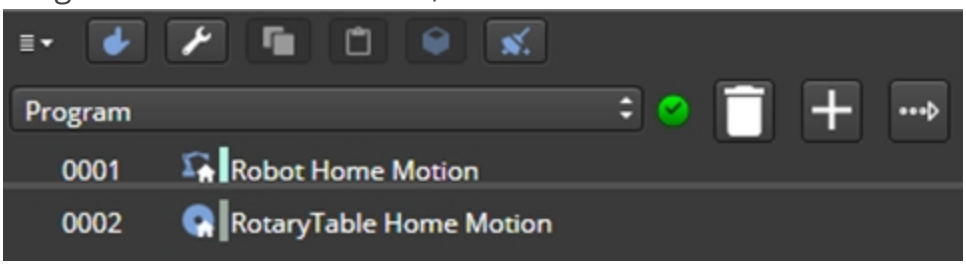



2. Program has been simulated and has errors



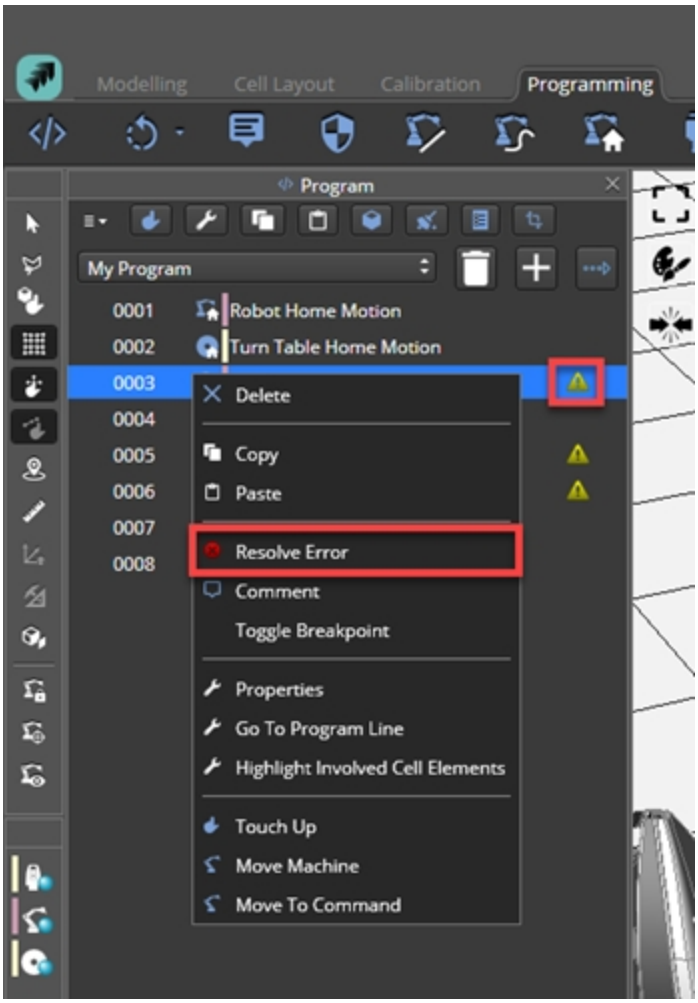



Program has been simulated, no errors have been found

3. 


The program simulation button creates a simulation that checks for collisions and other issues in the program (e.g., a reflector is obstructed and cannot be measured by the tracker) and will let you know by displaying an error sign next to the problematic command in case it finds any.

In case of a collision, you can either fix it manually or use the choice Resolve Error within the properties shown after right-clicking on the problematic program line.




## Simulation Warnings and Errors

Validation will distinguish between Errors and Warnings.

### Warnings

Warnings are the issues that validation detects that could **impact the quality of the program and point cloud**, however they will **not prevent the program** from being **executed** in the real cell.

Examples Warnings: Stop Measurement missing after a Start Measurement, Target is out of range of tracker, Reflector angle of incidence is too big, etc...

---

## Errors

Errors are the issues that validation detects that will **either** prevent the **program** from **executing** due to **missing information** or show a **potential collision**.

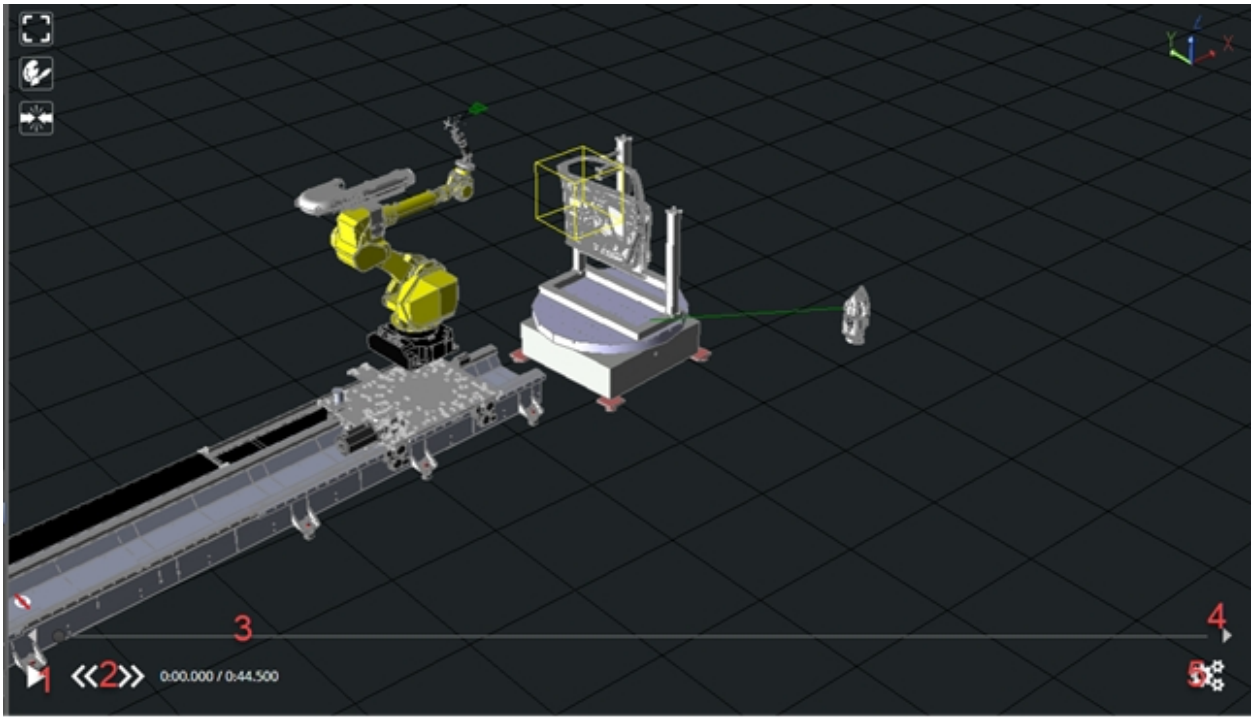
Example of Errors: Using different robots in the same continuous motion group, Collisions, Position not Reachable

## Simulation Error Feedback

The Simulation Errors and Warnings can always be seen in the simulation error list tab.

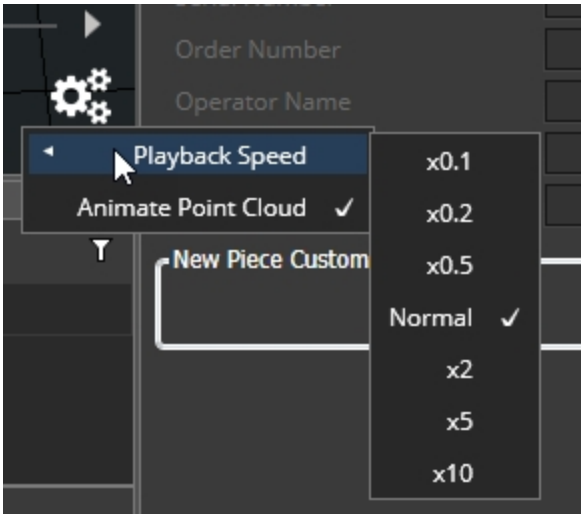
Re-validation might be required to clear some of the items in this list.

## Simulation Player UI




1. **Start/Pause** button. Starts or pauses the simulation video. In case the simulation was not created yet, it will create the simulation before starting the video. In case the simulation has ended, the button becomes a **Replay** icon
2. **Go to previous** or **next** command button. Moves the simulation player to the next command
3. The simulation timeline providing mouseover command information
4. **Next and previous frame** button. Moves the player 1 frame forward or backward

**Settings** allows to configure the playback speed.

5. 


6. **Animate Point Cloud**:
    - When **enabled** this will create the point cloud together with the related robot movements during simulation.
    - When **disabled**, the point cloud will be shown independent of the simulation player animation.

| | |
|---|---|
| ⓘ | The point cloud might take some time to calculate in case the option is disabled and will display parts while it is being calculated and will display fully once done. |

| | |
|---|---|
| ⚠ | For bigger projects (e.g. +4000 lines) it is recommended to disable the Animate point cloud as this has a big impact on the 3D world performance and display. |

| | |
|---|---|
| ⓘ | Adding a dialog to a program will pause the simulation player when arriving to that command. Clicking the OK button will resume the playback. |

| | |
|---|---|
| ⓘ | Switching to the execution tab will hide the simulation player (and pause any ongoing program simulation) |

| | |
|---|---|
| ⓘ | When connecting to a real cell, the player will now longer show and the real cell position of the robots will be shown. |

### Player skipping frames

When running the simulation player on a machine that is not meeting the software requirements, there is a possibility that some frames in the simulation player will be skipped. This is due to the Hoops software needing access to a good GPU to display the cell items in each frame.

## Hiding the Point Cloud from 3D window

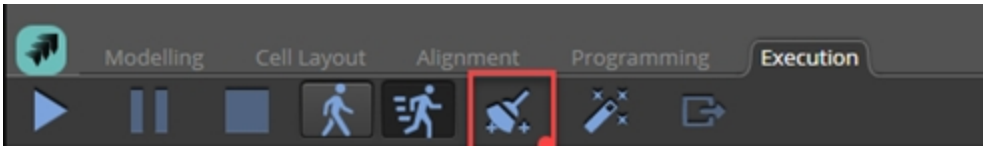
The measurement window will show always the last executed (both in the real cell and in simulation) measurements. By selecting the programs toggle, it is possible to switch showing all the results from the programs that ran.

The eye icon can be used to hide or unhide measurements results or even the whole program from the 3D view.

Things to note:

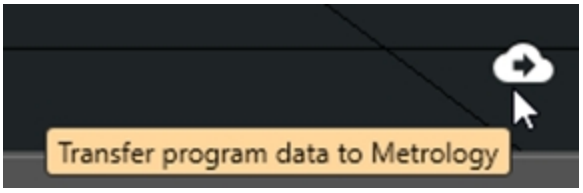
- Real cell measurements will be shown with a full icon
- Program simulation measurements will show with a hollowed out icon

In the past the clean result button was in the execution tab and now has been moved to the measurement window. (Available in the alignment, programming and execution tab)




## Sending data to Metrology

Once connected to Metrology, the button “Transfer data to Metrology” is no longer greyed out.




Clicking the button will send all the Point Cloud and point measurements to the connected Metrology application. The sending of the data and the simulation player are not synced!

---

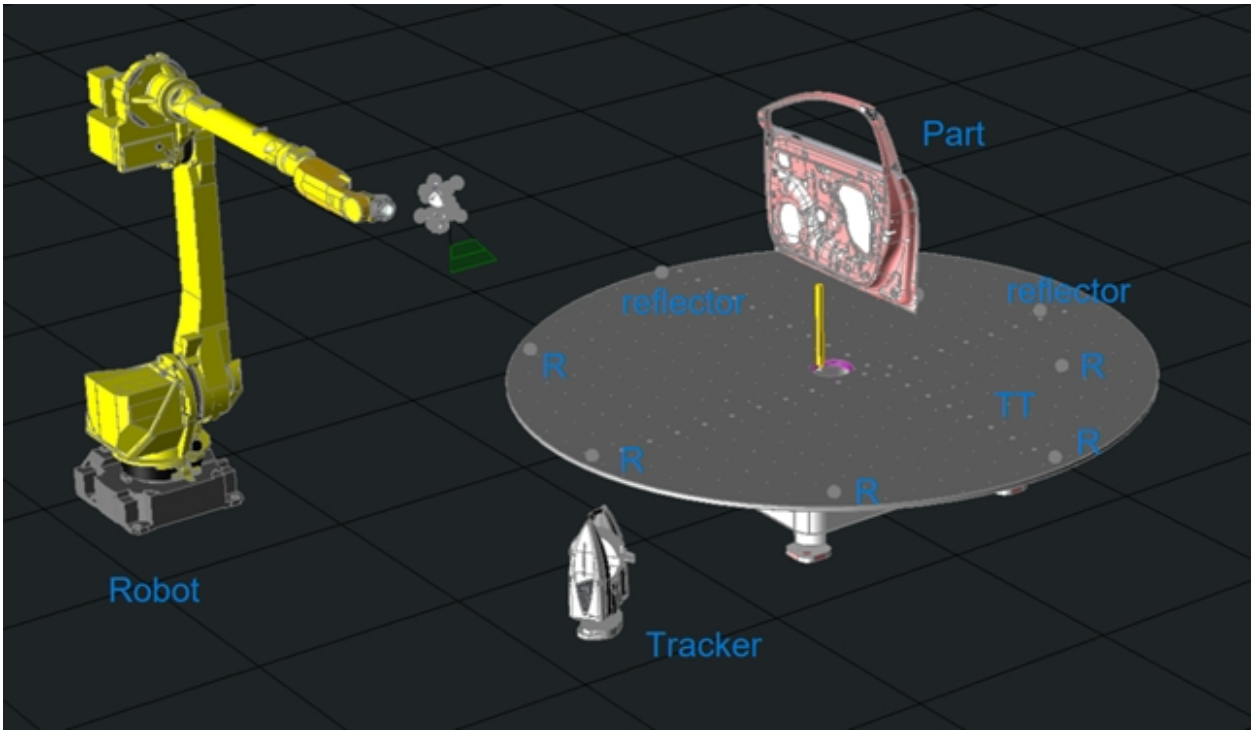
## Notable use cases

In this section five notable scenarios are described together with the best practices to follow to create programs to cover such situations.

### Single robot cell with Rotary Table and stationary tracker

Considering a cell with the following configuration:

- 1 stationary tracker
- 1 robot
- 1 Rotary Table
- 1 group of movable reflectors (attached to Rotary Table)




The user can create a program manually adding the necessary commands or using the path planning both in automatic and semi-automatic mode.

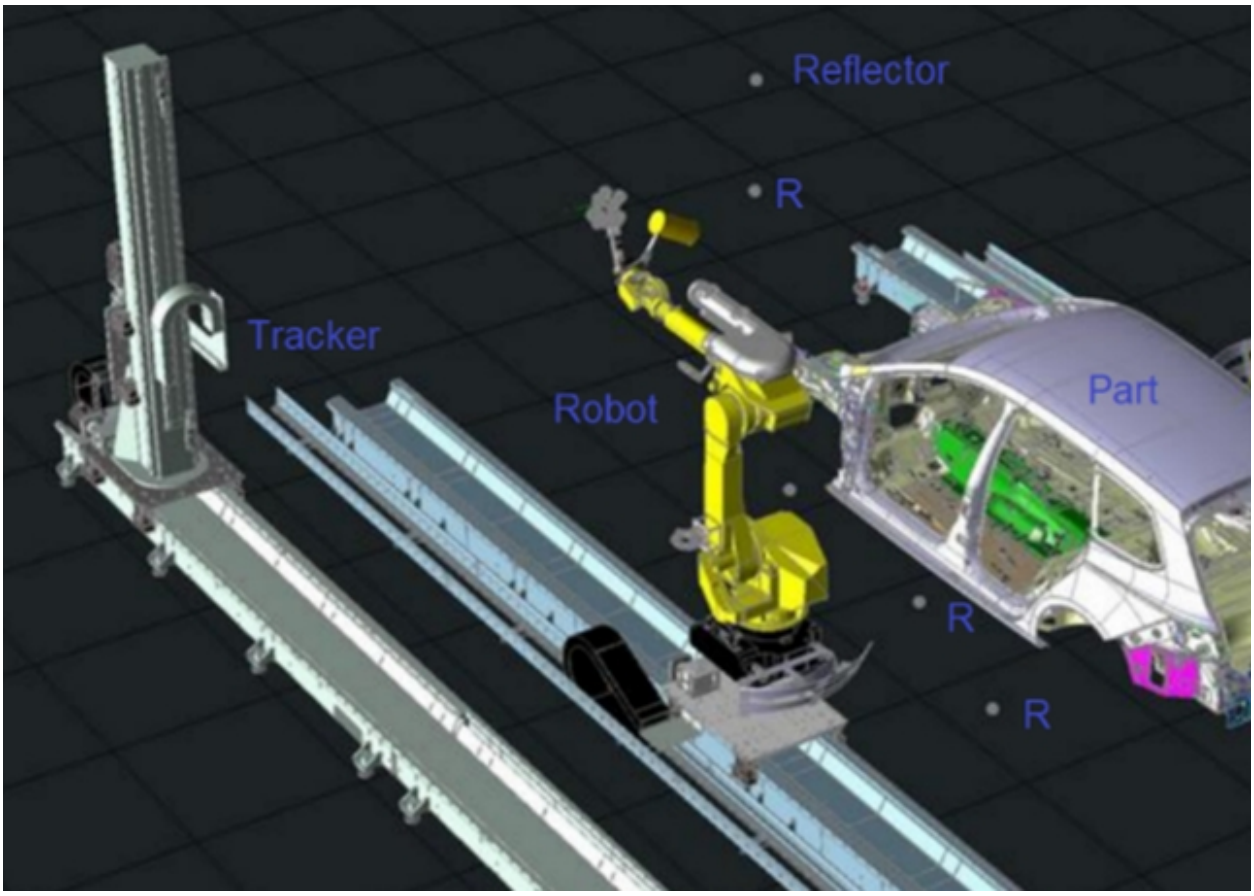
In all cases, the program needs to have the structure depicted in Figure Programming workflow resembling the process that the automatic path planning is computing and according to the rules for compensation defined in chapter Compensation

In automatic or semi-automatic mode, it is necessary to follow the steps defined in the relative chapters.

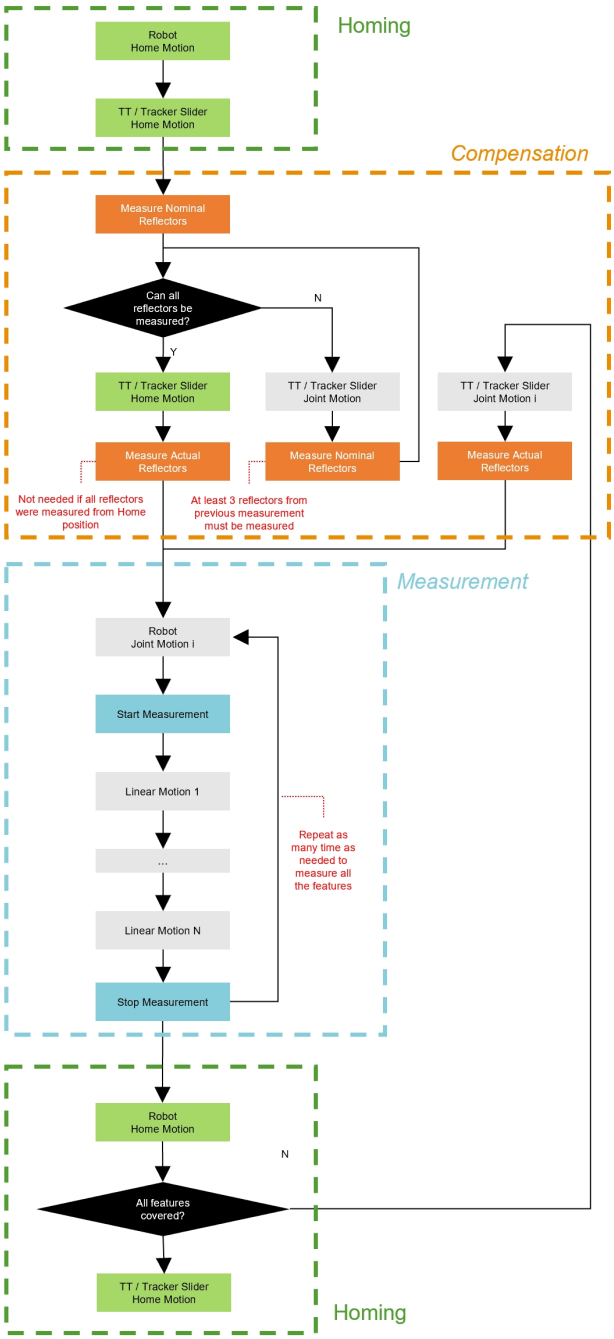
## Single robot cell with stationary part holder and movable tracker

Considering a cell with the following configuration:

- 1 movable tracker over a slider
- 1 robot
- 1 stationary part holder
- 1 group of fixed reflectors




The same considerations above apply to this use case as the movable slider also inserts a change in the relationship between tracker and part, so the same rules apply for the compensation part as described in Figure below.



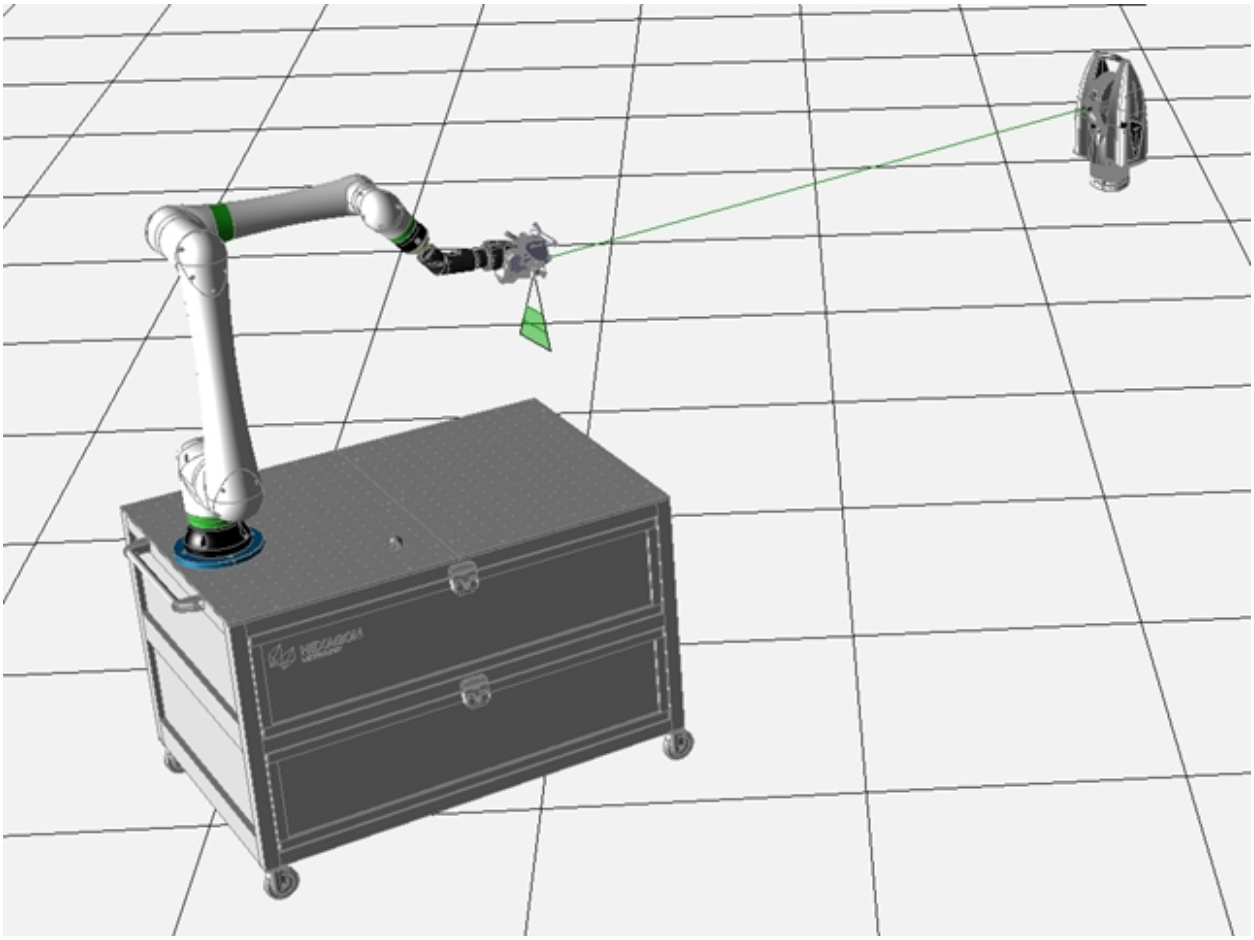

Programming workflow for a single robot cell with Rotary Table and stationary tracker (consider “TT” within the relative blocks) and for a single robot cell with sta-

tionary part holder and movable tracker (consider “Tracker Slider” within the relative blocks)

## Single robot cell with stationary part holder and stationary tracker

Considering a cell with the following configuration:

- 1 stationary tracker
- 1 robot
- 1 stationary part holder
- 1 group of fixed reflectors

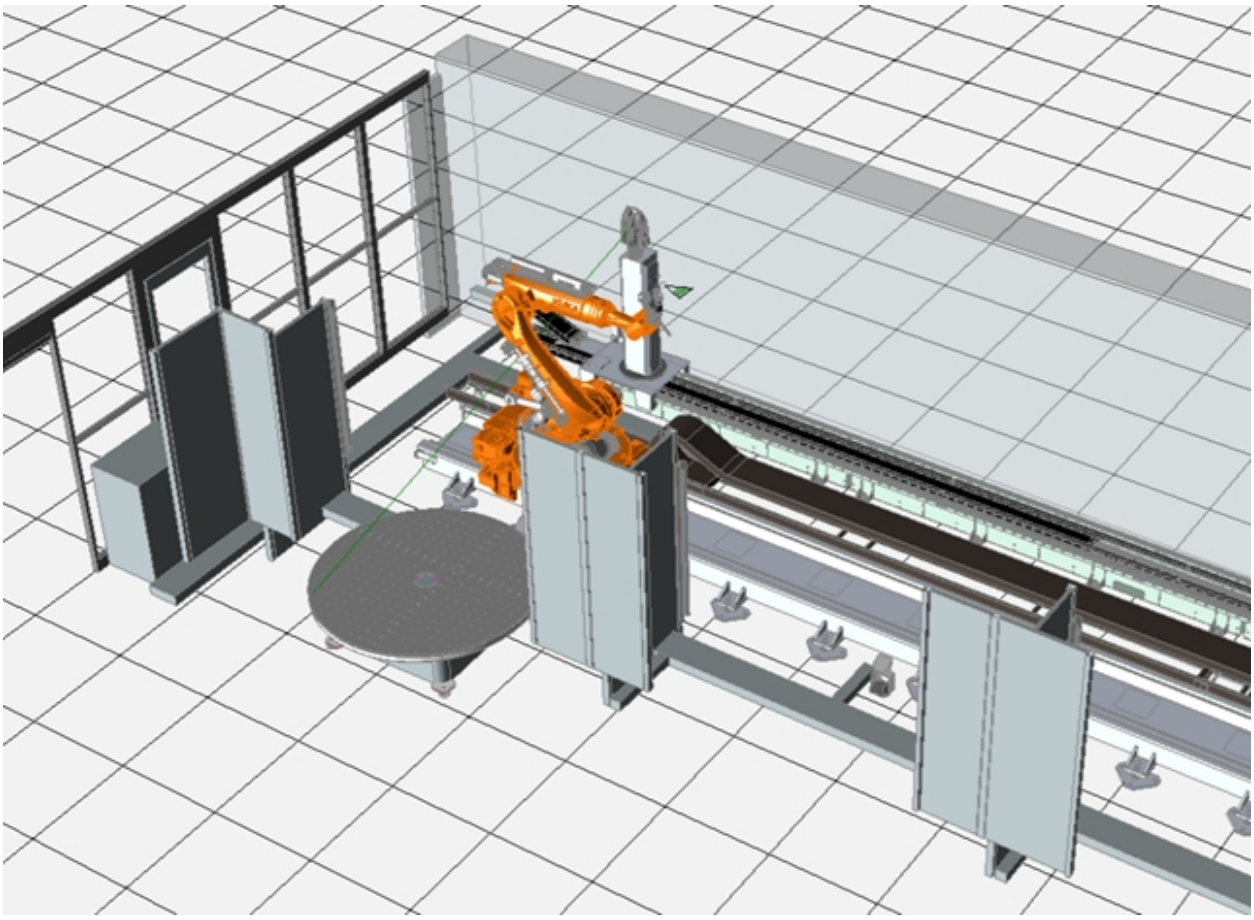


In this case no compensation is necessary as all reflectors must be visible (and therefore measured) from a single tracker position and since the part holder is stationary, there will be no changes in the relationship between tracker and part so the programming workflow is the same as described in the previous sections and depicted in Figure Programming workflow without considering the compensation parts.

## Single robot cell with Rotary Table and movable tracker

Considering a cell with the following configuration:

- 1 movable tracker over a slider
- 1 robot
- 1 Rotary Table
- 2 groups of reflectors:
    - movable reflectors (attached to Rotary Table)
    - fixed reflectors



In this case, there are two reasons why there can be a change in the relationship between tracker and part as both the Rotary Table holding the part and the tracker can move.

Since there are two different changes in the relationship between tracker and part, it is needed to have two reflector groups, one for the reflectors mounted on the Rotary Table and another one for the fixed reflectors.

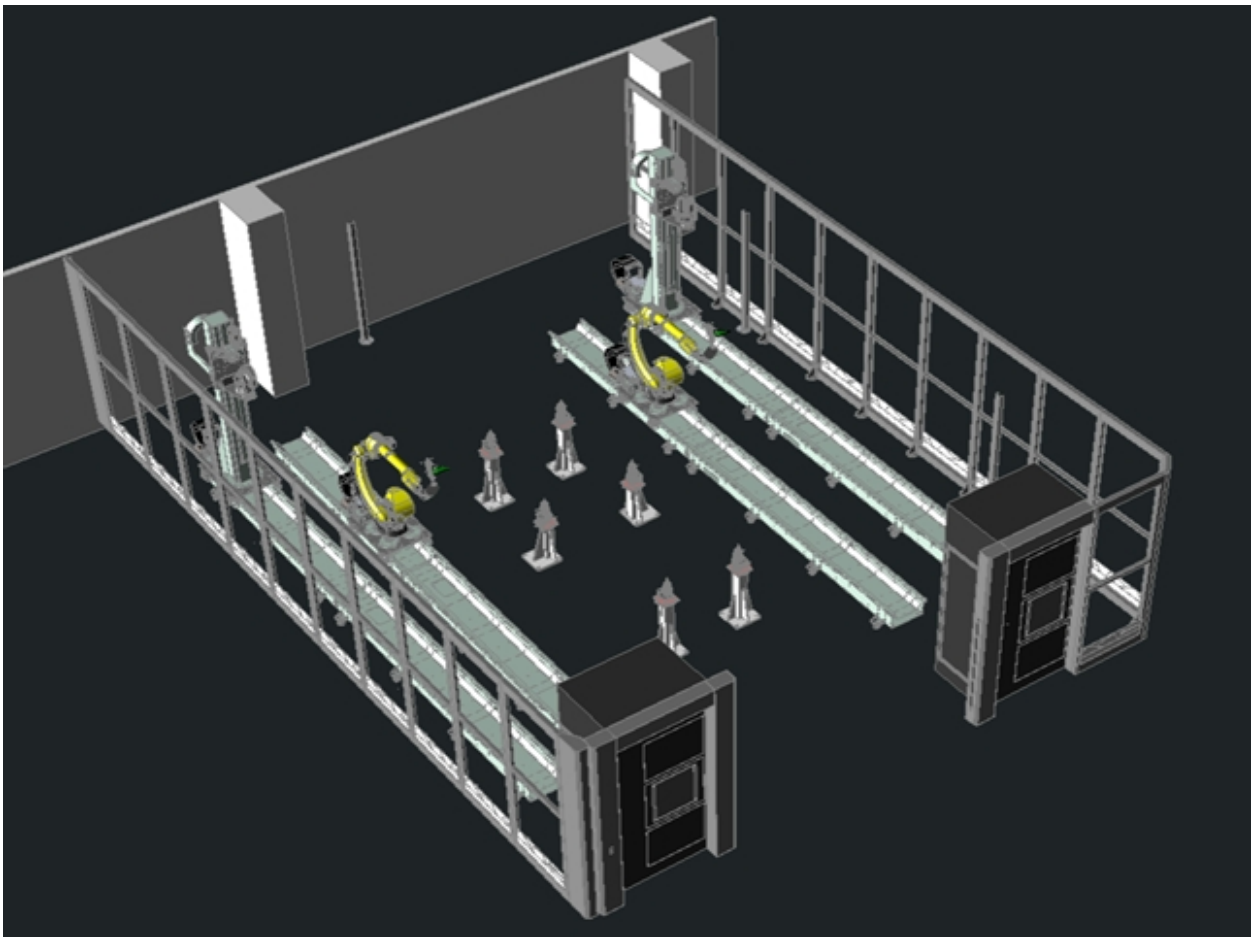
In this case whenever the Rotary Table moves, it is needed to measure the actual positions of the relative reflector group, and whenever the tracker moves, it is needed to measure the actual positions of the fixed reflector group.

The user can refer to Figure Programming workflow keeping in mind that the Rotary Table and tracker slider have to be considered independently within the compensation part.

## Dual robot cell with stationary part holder and dual movable trackers - Leica only

Considering a cell with the following configuration:

- 2 movable trackers over a slider
- 2 robots
- 1 group of fixed reflectors



A cell involving two robots and two trackers requires specific consideration about compensation, concurrent tracker usage by the two robots, collision avoidance prevention and parallel program execution capabilities to use the cell at its best capacity reducing at the same time the safety risks which are explained in the following sections. After the concepts are introduced separately, the complete workflow needed for a dual robot cell is explained in a dedicated section.

## Dual Tracker Compensation

For dual trackers compensation the same compensation principles apply. The only other difference is how to compensate both trackers to each other.

The idea is to measure as many reflectors as possible in their nominal positions with e.g., Tracker A. Tracker A might need to move to multiple positions to measure all reflectors.

After this step it is needed to measure the actual position of the reflectors with Tracker B, where we at least need three of the reflectors that were measured by tracker A.

As a general compensation rule, it is always necessary to “compensate” every change in the relationship between the part and the tracker (due to tracker or Rotary Table movement), to translate the taken measurement to the global coordinate system.

This means that after each tracker or Rotary Table movement a **Measure Actual Reflectors** command must be added for the tracker, we want to use in the measuring command. A **Measure Actual Reflectors** command by itself is meaningless without a **Measure Nominal Reflectors** command, so if it is needed to the move the tracker or the Rotary Table, it’s mandatory to add a “**Measure Nominal Reflectors** command at the beginning of the program.

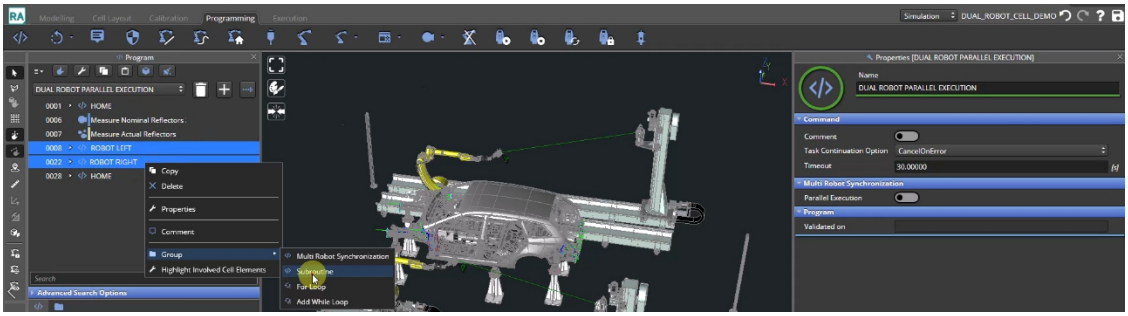
Therefore, if two trackers are available and both must be used, it’s needed to add at least a **Measure Nominal Reflectors** command for the first tracker and a **Measure Actual Reflectors** command for the second, even in case they did not move physically.

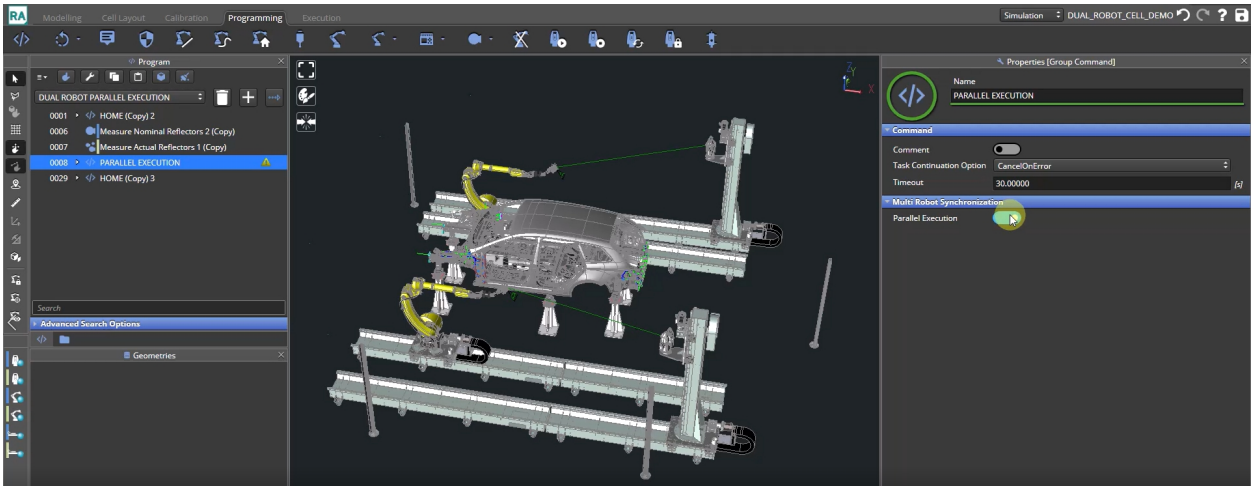
## Parallel programming

To execute commands in parallel user must group those commands by Subroutine commands and enable the **Parallel Execution** toggle switch.

First the user must create two separate robot subroutines (ROBOT LEFT and ROBOT RIGHT) were the two robots scan the features they need to measure. After that, the two subroutines can be grouped within a new subroutine by selecting them, right-clicking and selecting *Group* | *Subroutine*.

In this example the new subroutine is renamed “PARALLEL EXECUTION” inside its properties and is marked as parallel by enabling the **Parallel Execution** toggle switch. This means that its children (“ROBOT LEFT” and “ROBOT RIGHT” subroutines) will be executed in parallel.





## Tracker crossing

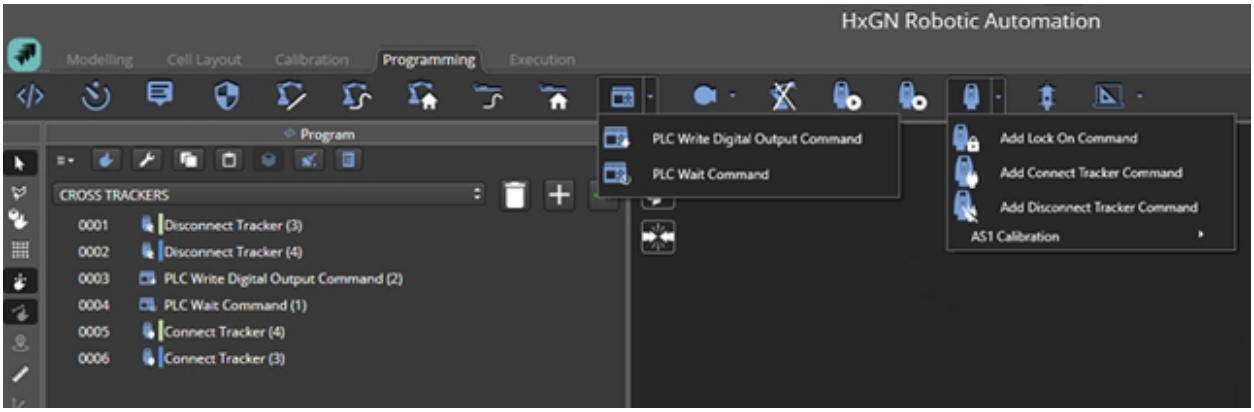
There might be situations where a robot cannot be used with the tracker available on the same side of the cell, e.g., Robot A and Tracker A. For this reason, it is necessary to use the tracker available on the other side of the cell and force it to lock on the scanner used for the robot in use, e.g., Robot A and Tracker B locked on Robot A scanner. The cell must be configured to use the Automation Switch Controller (ASC) and HxGN Robotic Automation allows to interact with the ASC using PLC commands as described in chapter UI Leica Commands

To prevent the tracker from being unable to connect because the other tracker is already connected to the same IP, a new workflow has been implemented which consists of adding a new connect / disconnect tracker commands described in chapter UI Leica Commands

Then the crossing workflow will be like:

1. Make both trackers **lock on a reflector**
2. **Disconnect Tracker** command for each tracker to make sure both trackers are available before to try any connection.
3. **Write PLC signal** configured with the desire value to perform the crossing at HW level.
4. **Wait PLC signal** to ensure the right value is set up in the HW.
5. **Connect Tracker** command for each tracker to enable connection once all configuration has been done.

In the picture below it is depicted a basic program showing how to cross the ASC signal using the new commands mentioned above.




Crossing – ASC

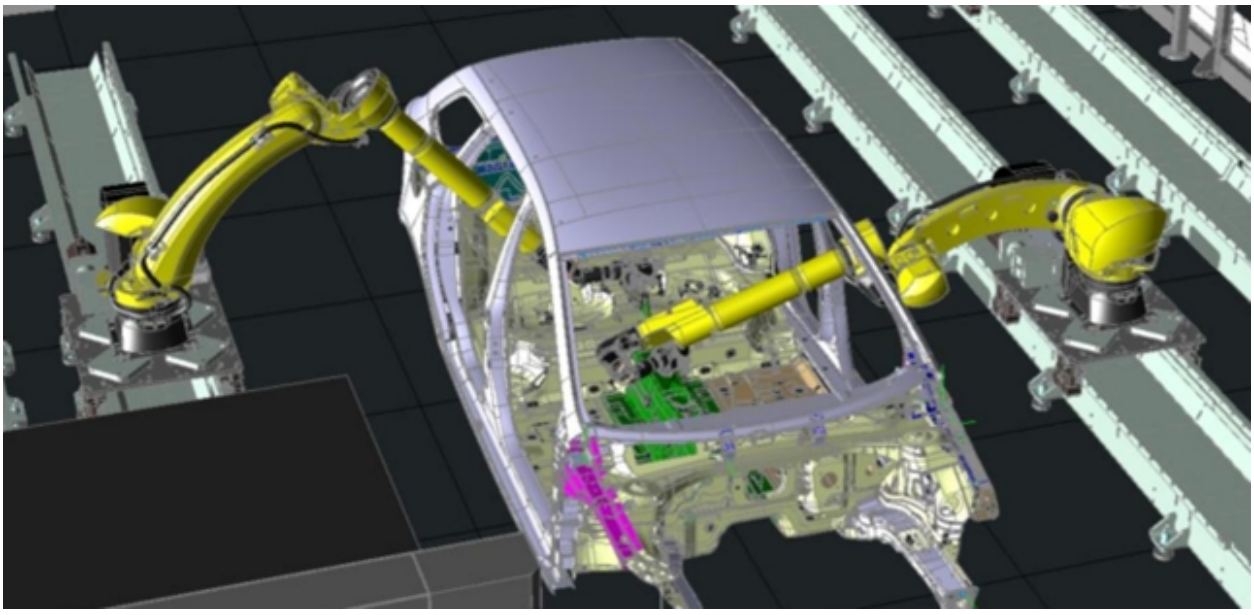
- Add Crossing flow at the start of the program to force desired consideration (not relay in theoretic status).

- Add Crossing flow at the end of the program to leave the cell in the usual configuration (to avoid issues because tracker is recognizing scanner as reflector instead of AS1).

### Concurrent resource

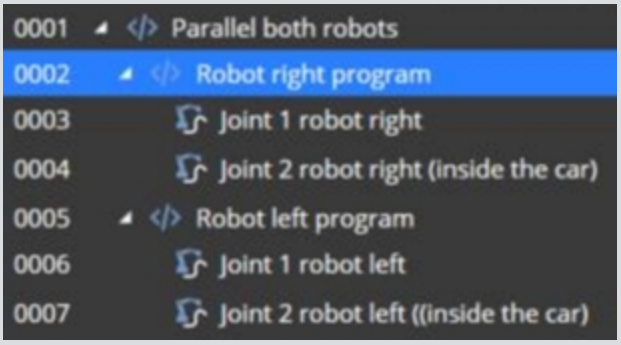
Although running robots in parallel saves overall measurement cycle time, there may be cases where it is needed to execute their movement in sequence, e.g., postpone one of them until the second has finished its task to prevent collisions.

There might be dangerous situations where both robots running in parallel are too close to each other and may collide, like in the picture below so it is paramount that when robot A is working in an interference zone, the other robot B waits for robot A to complete its task within the interference zone before entering it. For this reason, a special command is available as described in chapter Acquire / release Lock for Multi robot synchronization



Example: Lock resource to prevent collisions

Suppose we have program like the following:

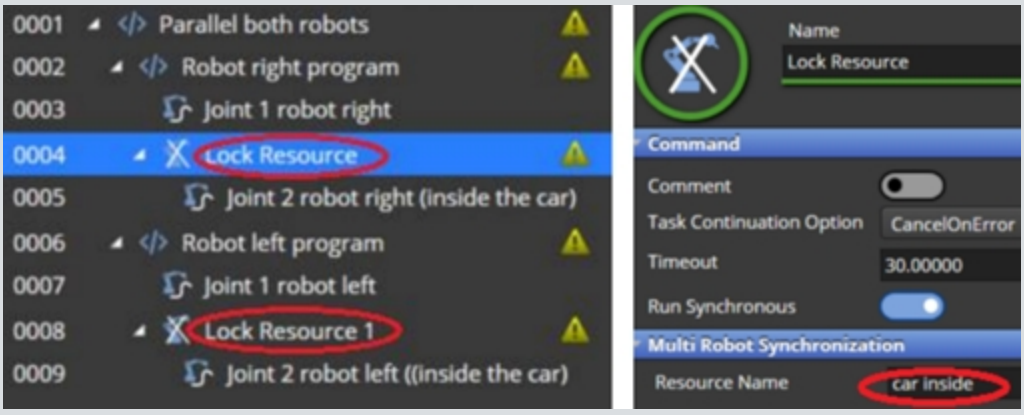



Here programs for both robots (Robot right program, Robot left program) must be executed in parallel, but “Joint 2” motions of both are happening inside the BIW and potentially colliding.

So, the potentially dangerous commands must be enclosed within Lock Resource commands and give them the same **Resource Name**.

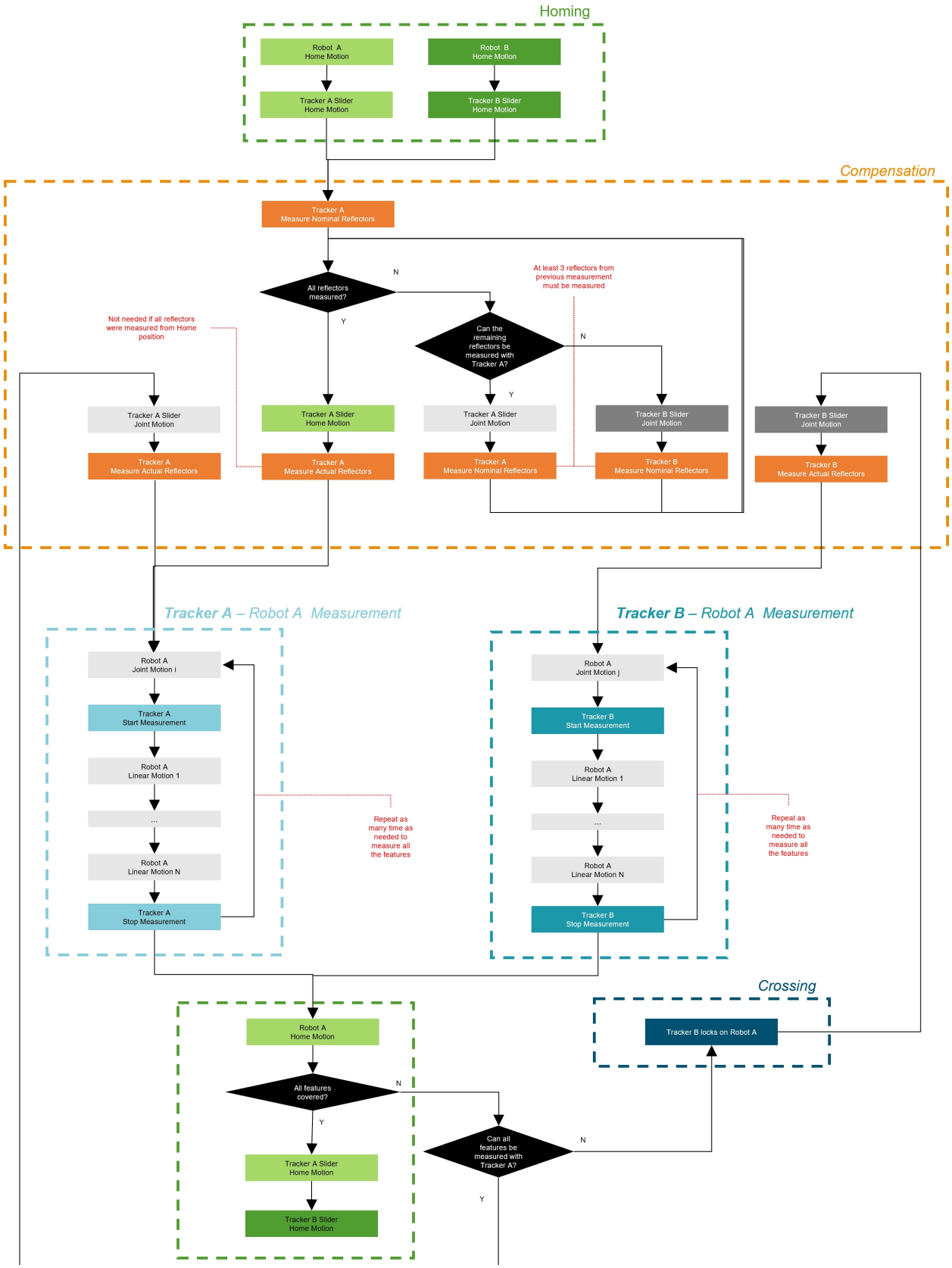
> Resource names for both commands must match exactly, e.g., “car inside” (as shown in picture below)

Now, the first robot reaching Lock Resource command will acquire the resource (“car inside”) and the second will wait until the first one will release the resource and only then continue with its execution..



Dual robot cell programming workflow using one robot and two trackers

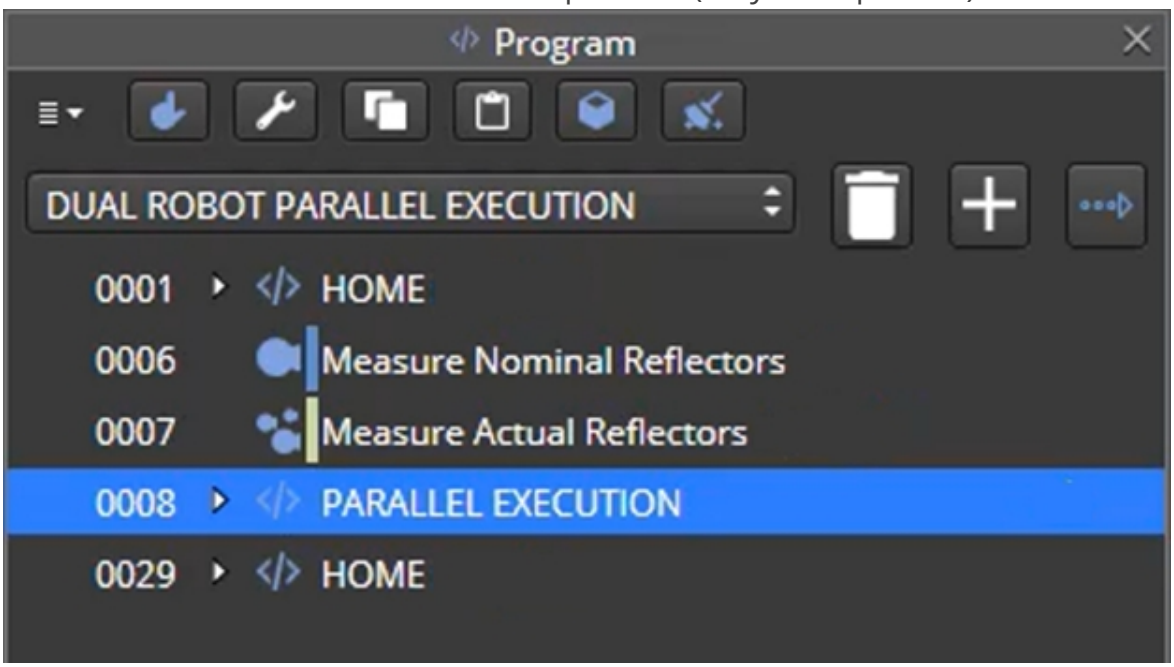
| | |
|---|---|
| Programming | 508 |


Homing
Robot A
Home Motion
Robot B
Home Motion
Tracker A Slider
Home Motion
Tracker B Slider
Home Motion
Compensation
Tracker A
Measure Nominal Reflectors
All reflectors
measured?
N
Y
At least 3 reflectors from
previous measurement
must be measured
Not needed if all reflectors
were measured from Home
position
Can the
remaining
reflectors be
measured with
Tracker A?
N
Y
Tracker A Slider
Joint Motion
Tracker A Slider
Home Motion
Tracker A Slider
Joint Motion
Tracker B Slider
Joint Motion
Tracker B Slider
Joint Motion
Tracker A
Measure Actual Reflectors
Tracker A
Measure Actual Reflectors
Tracker A
Measure Nominal Reflectors
Tracker B
Measure Nominal Reflectors
Tracker B
Measure Actual Reflectors
Tracker A – Robot A Measurement
Tracker B – Robot A Measurement
Robot A
Joint Motion i
Tracker A
Start Measurement
Robot A
Linear Motion 1
...
Robot A
Linear Motion N
Tracker A
Stop Measurement
Repeat as
many time as
needed to
measure all
the features
Robot A
Joint Motion j
Tracker B
Start Measurement
Robot A
Linear Motion 1
...
Robot A
Linear Motion N
Tracker B
Stop Measurement
Repeat as
many time as
needed to
measure all
the features
Crossing
Tracker B locks on Robot A
Robot A
Home Motion
All features
covered?
N
Y
Can all
features be
measured with
Tracker A?
N
Y
Tracker A Slider
Home Motion
Tracker B Slider
Home Motion

### Complete programming workflow

The general program workflow for a parallel execution program is described in Figure Parallel programming workflow and contain the following main steps:

1. Send all robots and sliders to “Home” position (may be in parallel)




2. Perform necessary compensation steps as explained in chapter Compensation
3. Create sub-routines as separate programs for the two Robot A / Tracker A and Robot B / Tracker B pairs in parallel as described in chapter Parallel_programming. This step may be repeated as many times as needed to measure all the features.
4. There might be situations when both robots running in parallel are only allowed to use the same tracker (the second is unreachable). A tracker can lock only on a face of the scanner mounted on the robot. E.g., when Tracker A is locked on a face of Robot A scanner, Robot B must wait for Robot A to complete its task, before Tracker A could lock on a face of Robot B scanner.

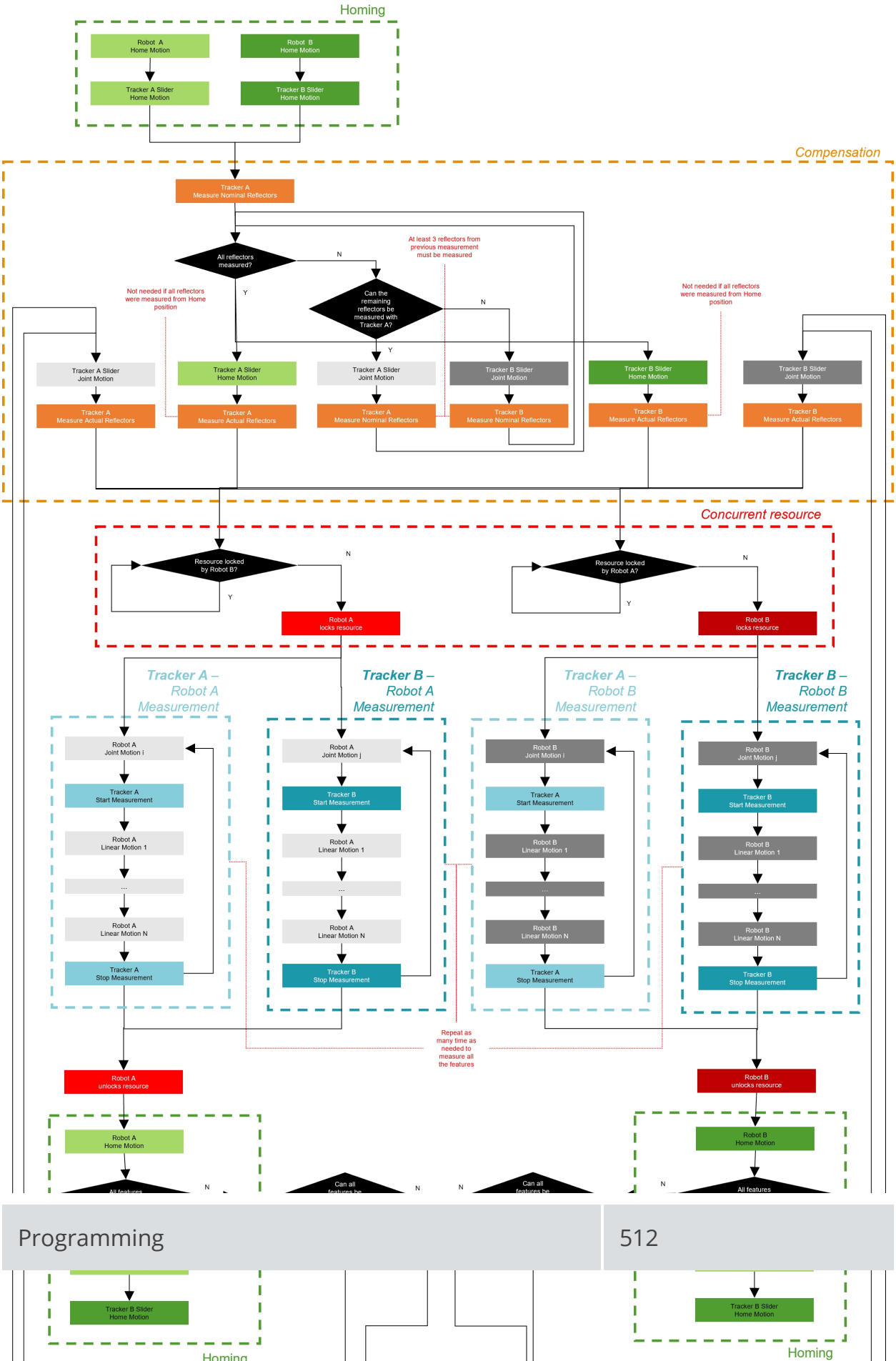
For this situation, the tracker crossing functionality is needed as described in chapter Tracker_crossing

The flowchart depicted in Figure Parallel programming workflow shows a workflow that includes only the measurement parts using Robot A together with either Tracker A or Tracker B.

A workflow containing both robot measurement routines have two additional measurement branches relative to Tracker B – Robot B and Tracker A – Robot B which are exactly the same as the one described in Robot A measurement branches

5. Whenever the measurement branches of Robot A and Robot B must inspect an interference zone, the concurrent resource usage functionality must be used as described in chapter Concurrent_resource
6. Send all robots and sliders to “Home” position (may be in parallel)

Complete dual robot cell programming workflow using two robots and two trackers


Homing
Robot A Home Motion
Robot B Home Motion
Tracker A Slider Home Motion
Tracker B Slider Home Motion
Compensation
Tracker A Measure Nominal Reflectors
All reflectors measured?
N
Y
At least 3 reflectors from previous measurement must be measured
Can the remaining reflectors be measured with Tracker A?
N
Y
Not needed if all reflectors were measured from Home position
Not needed if all reflectors were measured from Home position
Tracker A Slider Joint Motion
Tracker A Slider Home Motion
Tracker A Slider Joint Motion
Tracker B Slider Joint Motion
Tracker B Slider Home Motion
Tracker B Slider Joint Motion
Tracker A Measure Actual Reflectors
Tracker A Measure Actual Reflectors
Tracker A Measure Nominal Reflectors
Tracker B Measure Nominal Reflectors
Tracker B Measure Actual Reflectors
Tracker B Measure Actual Reflectors
Concurrent resource
Resource locked by Robot B?
N
Y
Resource locked by Robot A?
N
Y
Robot A locks resource
Robot B locks resource
Tracker A – Robot A Measurement
Tracker B – Robot A Measurement
Tracker A – Robot B Measurement
Tracker B – Robot B Measurement
Robot A Joint Motion i
Tracker A Start Measurement
Robot A Linear Motion 1
...
Robot A Linear Motion N
Tracker A Stop Measurement
Robot A Joint Motion j
Tracker B Start Measurement
Robot A Linear Motion 1
...
Robot A Linear Motion N
Tracker B Stop Measurement
Robot B Joint Motion i
Tracker A Start Measurement
Robot B Linear Motion 1
...
Robot B Linear Motion N
Tracker A Stop Measurement
Robot B Joint Motion j
Tracker B Start Measurement
Robot B Linear Motion 1
...
Robot B Linear Motion N
Tracker B Stop Measurement
Repeat as many time as needed to measure all the features
Robot A unlocks resource
Robot B unlocks resource
Robot A Home Motion
Robot B Home Motion
All features
N
Can all features be
N
N
Can all features be
All features
N
Tracker B Slider Home Motion
Homing
Tracker B Slider Home Motion
Homing

# Execution

Execution is the phase where a program that has been generated previously in Programming mode is run to perform a measurement task on a part.

Once a program is generated the user has the option to either run it in the Execution mode inside HxGN Robotic Automation by selecting the relative tab or run it outside of HxGN Robotic Automation within the Execution Manager, a web application specifically developed to running on a browser in kiosk mode for easy execution of programs by cell operators.

The user can run the program in both real and simulation environments. In real environment to execute a program, all the relative real cell components in use must be connected either to HxGN Robotic Automation or Execution Manager.

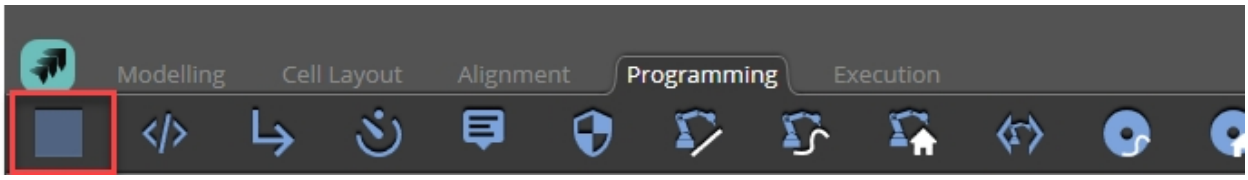
Before using a program within the Execution Manager, the user must validate the program and run it at least once in an online environment with HxGN Robotic Automation in Execution mode.

When using the “Move Machine” functionality in the real environment make sure all the other machines are in their correct position as it may cause collisions.

## Special case: Executing a safety motion in programming tab when connected to a real cell

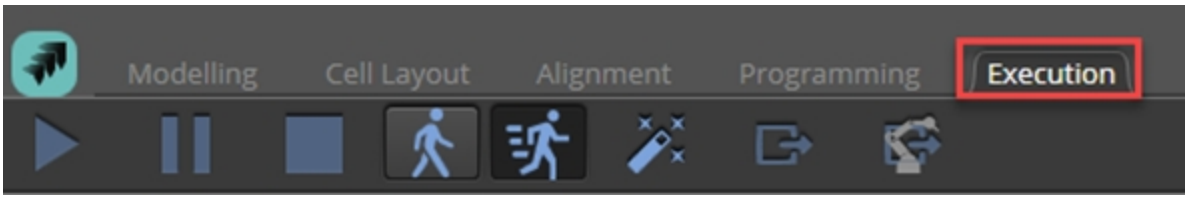
When connected to the real cell and executing a safety motion (e.g. by in the program tab clicking on a robot motion and clicking move to command), HxGN Robotic Automation will automatically enable an additional stop button in the programming tab.

This ensures that the user can order the software to stop this motion execution.

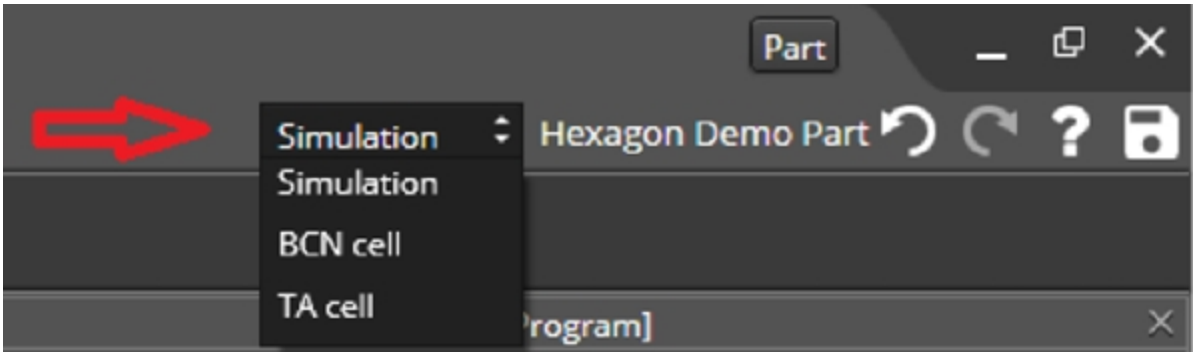



## Execution from HxGN Robotic Automation

The execution of a program in HxGN Robotic Automation is managed within the Execution tab.

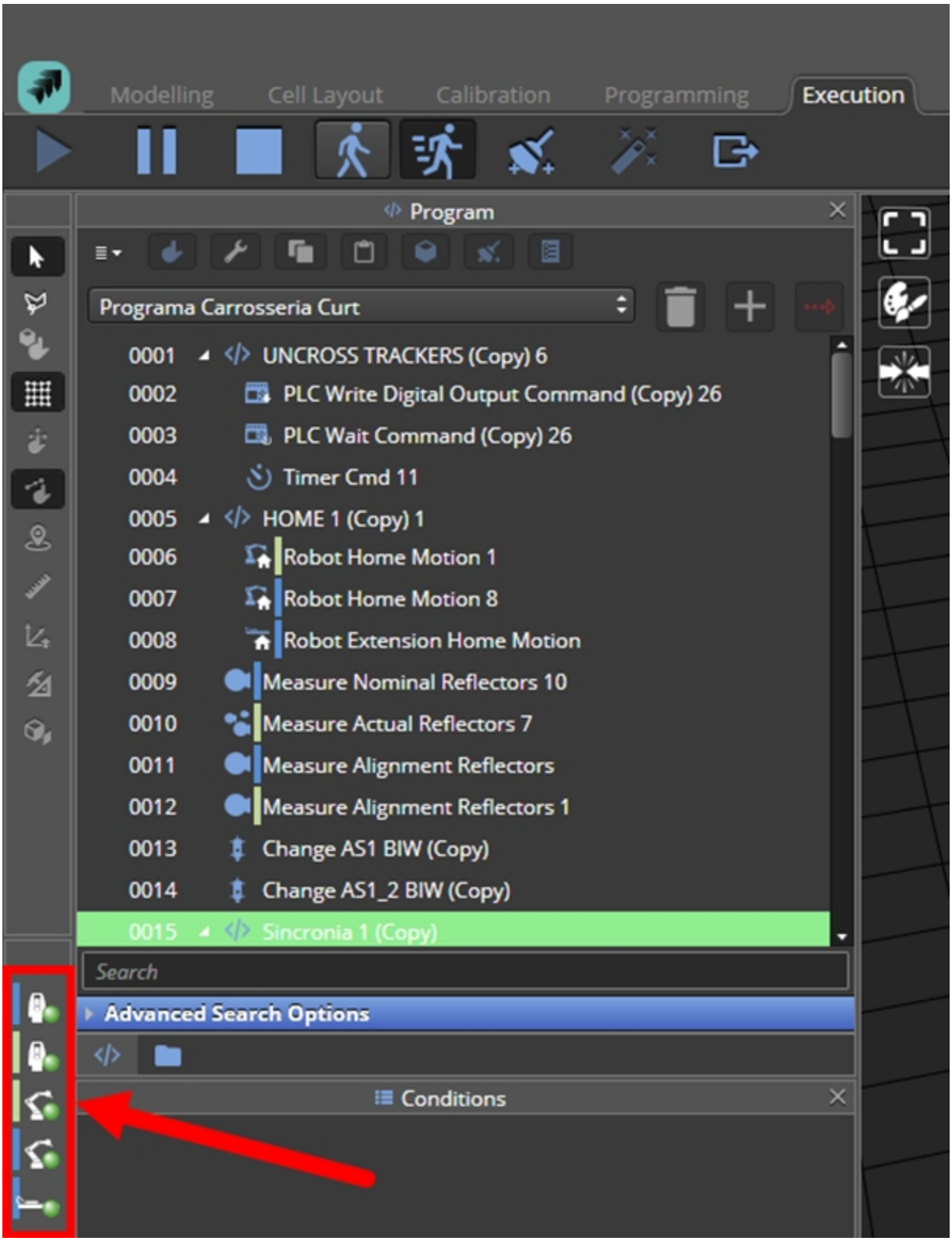



User can execute the program either in Simulation mode or in a Real environment i.e., connected to moving hardware in real. It is possible to select if using the simulation mode or to which real cell to connect to by selecting the environment in the top right part of the UI via a drop-down menu.




The configuration of cell name is done in the cell .json file creating a “Cell.json” file where “Cell” is the name of that cell model, e.g., in the example above “BCN cell.json” or “TA cell.json.”

Once the cell is connected to the real cell, the relative connection buttons will turn from blue to green.



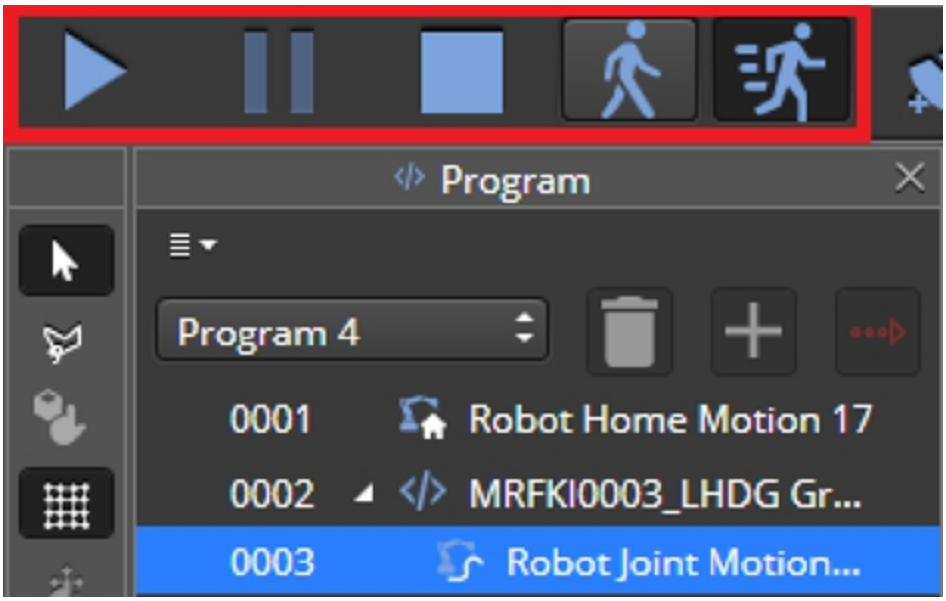

| | |
|---|---|
| Execution | 516 |

The user can choose between two modes of running the program:

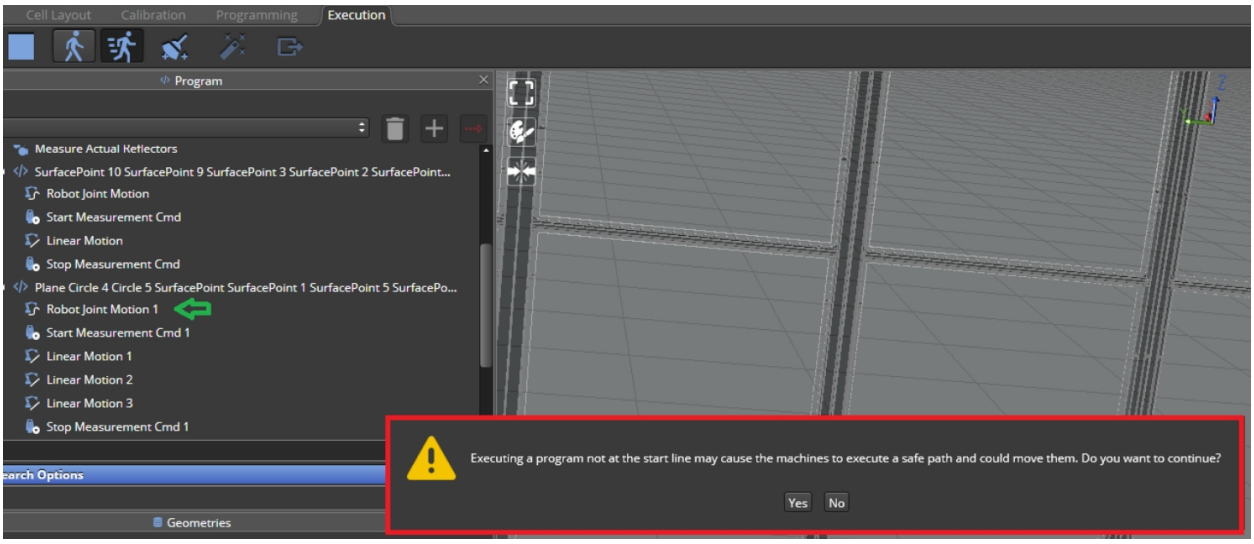
1. **Step Through** ( ): running the program step by step stopping after each of the commands and pressing the Start button to continue.
2. **Run**( ): running the program without any stops transitioning automatically between the commands.

To run certain commands of the program the user can either select a specific command and execute the program from there pressing the Start button ( ) or first click the **Start** button and then select the command to be executed first.

The user can at any moment decide to click Pause button ( ) to interrupt the program at the end of the current command or pressing Stop button ( ) to abort the program altogether.




When trying to execute a program not from the beginning, the following message will appear:

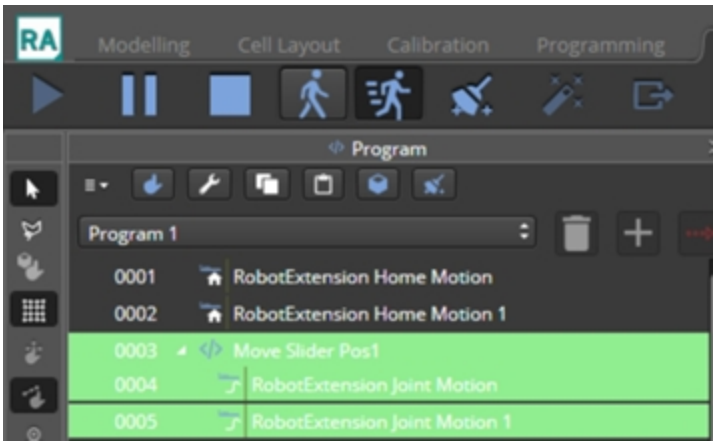


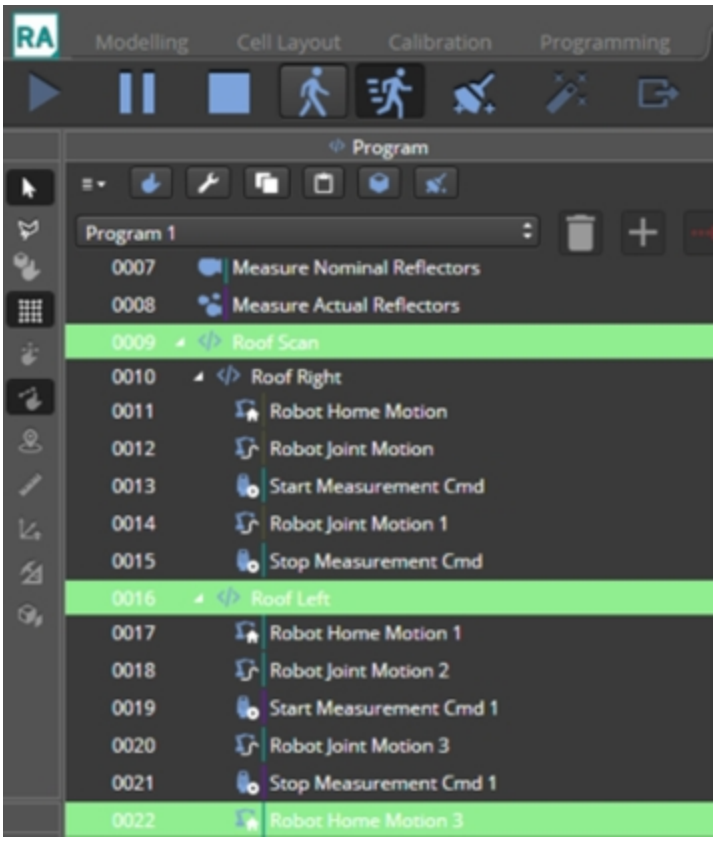

If the user clicks **Yes**, HxGN Robotic Automation will compute a new safe path from the current position to the selected command. If the user clicks **No**, the execution will be aborted.

When a program is executed, the command line being executed is highlighted in green.

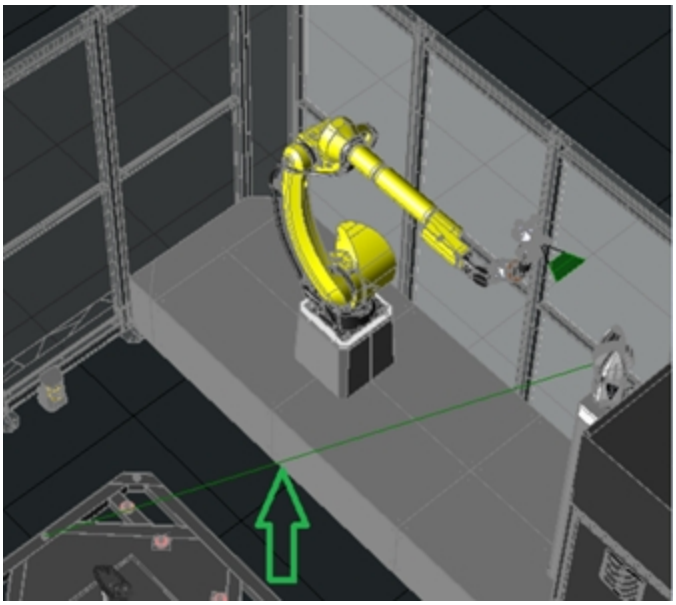
If a command is highlighted, it's parent will be highlighted as well.

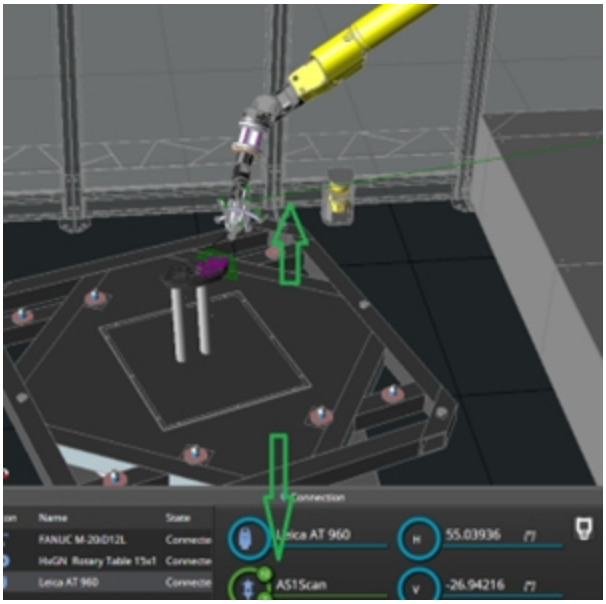
If the program has an error, the last active command will stay highlighted.





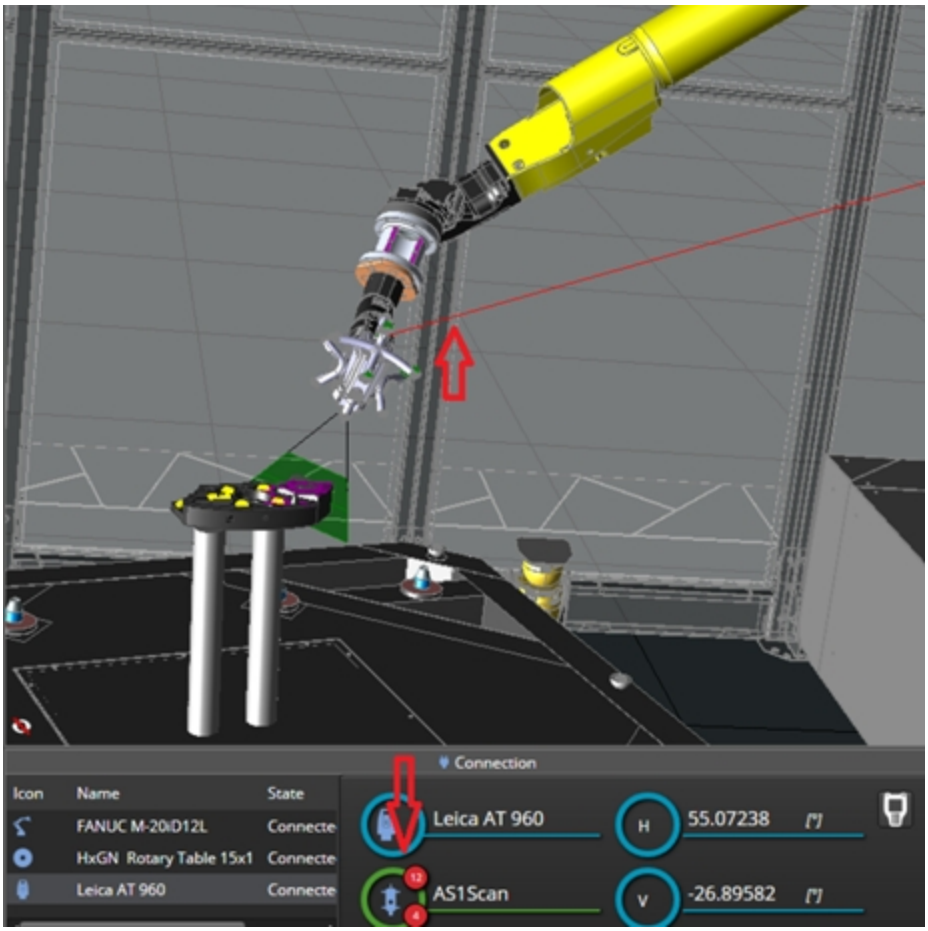

During program execution, the laser beam will be green when it is locked on a reflector, or a scanner face and it can get a 6DOF measurement.






Otherwise, the laser beam will be colored in red.

| | |
|---|---|
| Execution | 520 |




## Special case: Splitting a cell into 2 individual workstations

In case of for instance a PRESTO XL with 2 trackers, 2 robots and 2 scanners, it is possible to create 2 "mini" cells, holding each 1 tracker, 1 robot and 1 scanner.

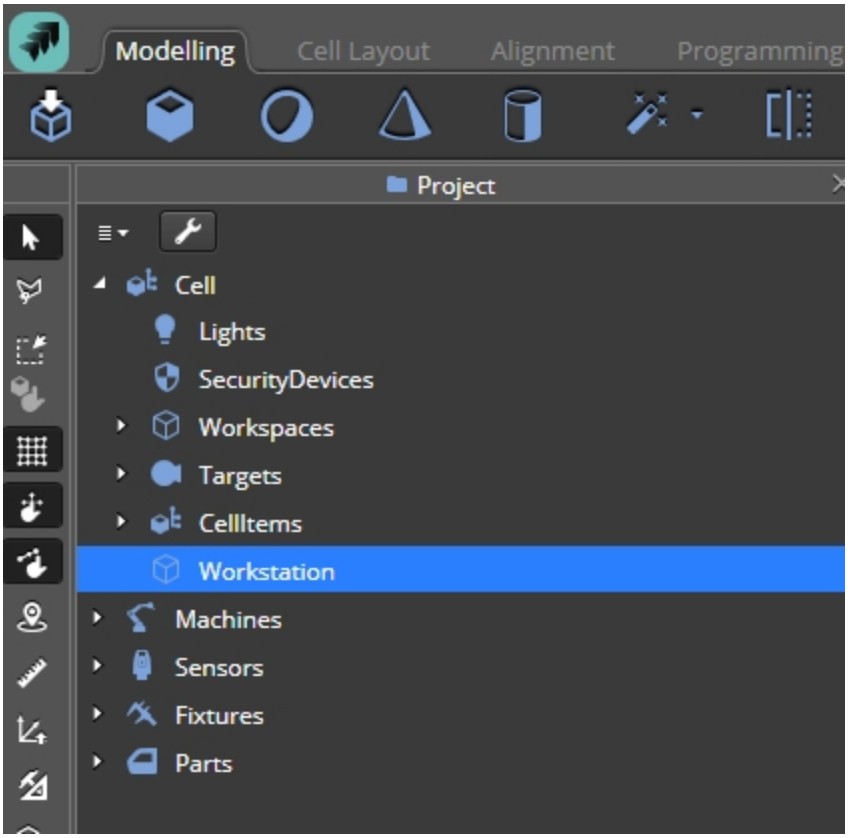
HxGN Robotic Automation will allow each of the "mini" cells to execute individually and at the same time!

The setup of these virtual cells is done in the cell json under the workstation area. See Cell JSON and the demo_cell.json example.

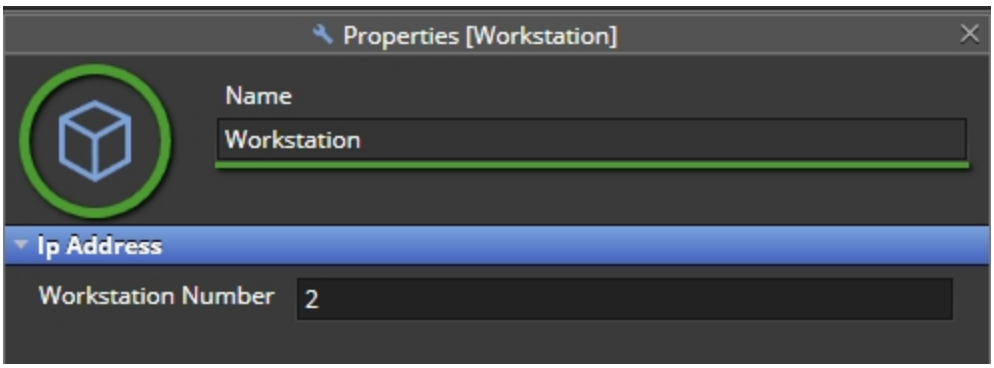
---

| | |
|---|---|
| Execution | 521 |

In HxGN Robotic Automation you can then select the modelling tab and the work-station.




In the properties you can then set which entry of the cell json you will use.




---

| | |
|---|---|
| Execution | 522 |

```
  "Workstations": [
    {
      "Id": "2ed723d5-672b-4b9c-b91a-91df92516f44",
      "Number": 2,
      "CellId": "2d208776-57a3-4e5a-aa18-ebebbcb73051",
      "TrackerIds": [
        "86e71621-7dea-42a8-8444-38ff1e6ae51e"
        //,
        //"e044521e-43ca-4ca9-96af-b3c6594bf4c8"
      ],
      "MachineIds": [
        "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
        "14a9baba-6410-44cc-a90a-fa1f73babbff",
        "dd0b8501-8043-4bd0-8864-ac262cc32e05"
      ]
    },
    {
      //LEFT
      "Id": "2ed723d5-672b-4b9c-b91a-91df92516f44",
      "Number": 20,
      "CellId": "2d208776-57a3-4e5a-aa18-ebebbcb73051",
      "TrackerIds": [
        "86e71621-7dea-42a8-8444-38ff1e6ae51e"
      ],
      "MachineIds": [
        "9aafa265-c6dd-4194-abc6-498fbb1bd1e9",
        "14a9baba-6410-44cc-a90a-fa1f73babbff",
      ]
    },
    {
      //RIGHT
      "Id": "2ed723d5-672b-4b9c-b91a-91df92516f44",
      "Number": 21,
      "CellId": "2d208776-57a3-4e5a-aa18-ebebbcb73051",
      "TrackerIds": [
        // "86e71621-7dea-42a8-8444-38ff1e6ae51e"
      ],
      "MachineIds": [
        "dd0b8501-8043-4bd0-8864-ac262cc32e05"
```

| | |
|---|---|
| Execution | 523 |

The program will need to be simulated, validated and run in the real cell like a normal program before it can then be exported to Execution Manager.

# Robot Program Export

## Configuring the application to show the export button.

1. Edit the *RAconfig.xml in C:\ProgramData\Hexagon\RA\Config* and make certain the following line states true:

   RA Config

   ```
   <ConfigSetting Name="IsShowExportRobotProgram" Value="True" DefaultValue="False"
   DisplayName="Flag whether to show export program to robot controller button"
   Description="Flag whether to show export program to robot controller button."
   TypeName="System.Boolean" />
   ```

2. Close the application and optional: Stop and Start Generic Host to let the new configuration apply.

## Exporting a program

For a program that you want to export, make certain the program simulation has been compiled and a clickable icon will appear in the execution tab to export the Robot Program.

Clicking the icon will show the following options:

- The controller type, aka FANUC or KUKA
- The controller: Select the relevant robot (relevant for dual robot programs)
- The export program button, allowing the program to be saved to the local machine.

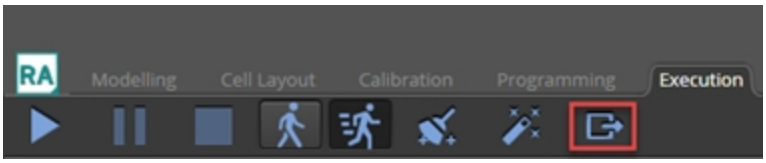
The exported robot program will contain all the Robot motions with its slider axes as robot motions, while all the other RA commands will appear as comments.

Exporting Robot Program doesn't support a program with parallel executing groups.

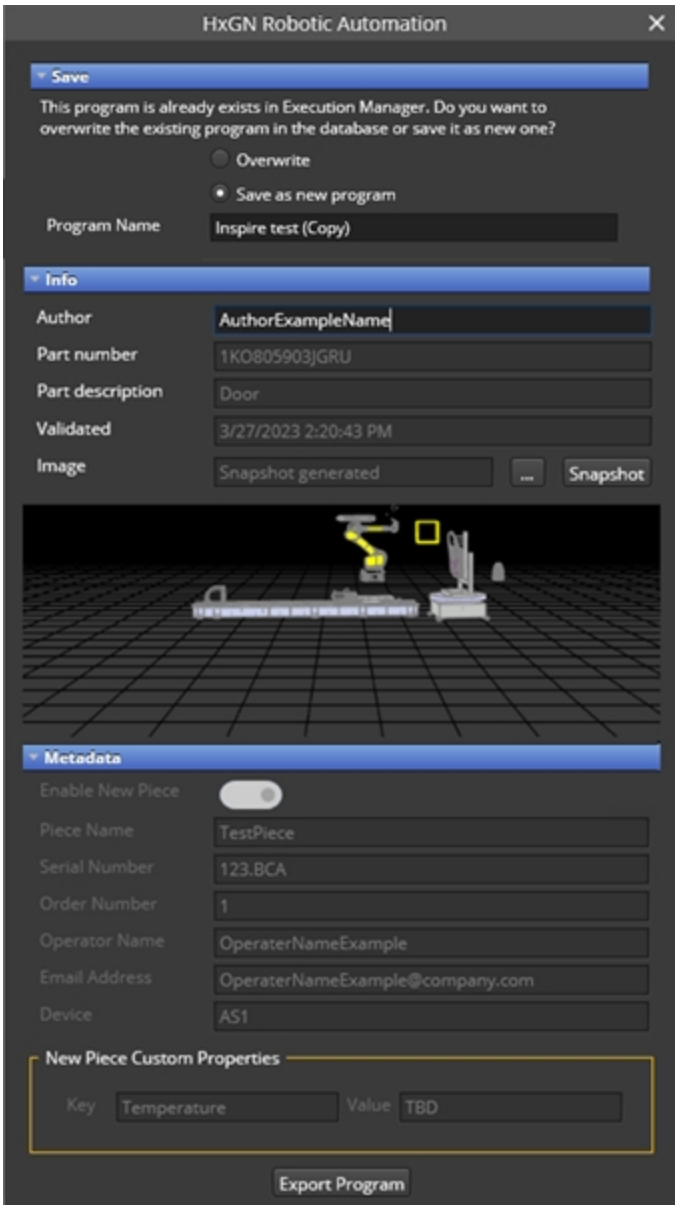
## Execution from Execution Manager (EM)

As explained at the beginning of chapter Validation and Simulation.htm, before a program can be run from Execution Manager (EM) it must be validated and executed from RA in the real cell represented by its digital twin where the program was generated offline and executed in simulation.

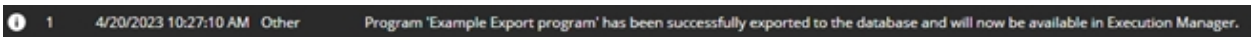
After this execution, the export button will be available.




Clicking on it, will open the export program pop-up. Filling in the required fields, will let the program to be exported correctly within the internal database, together with the embedded macros if using PolyWorks as metrology application.




After clicking on **Export Program**, a successful export confirmation message will appear in the status window.

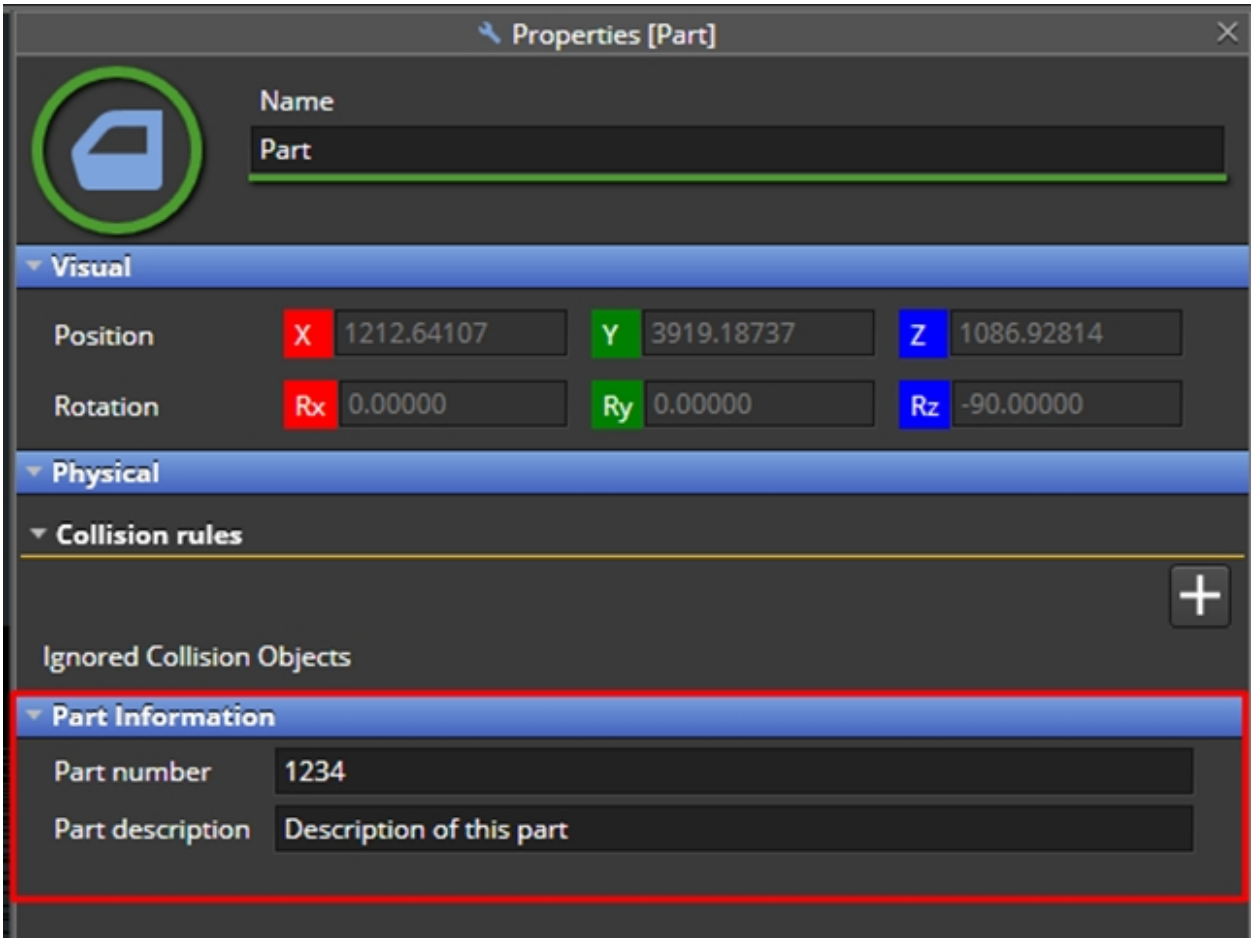



| | |
|---|---|
| | For the program to appear within the available programs in Execution Manager it |

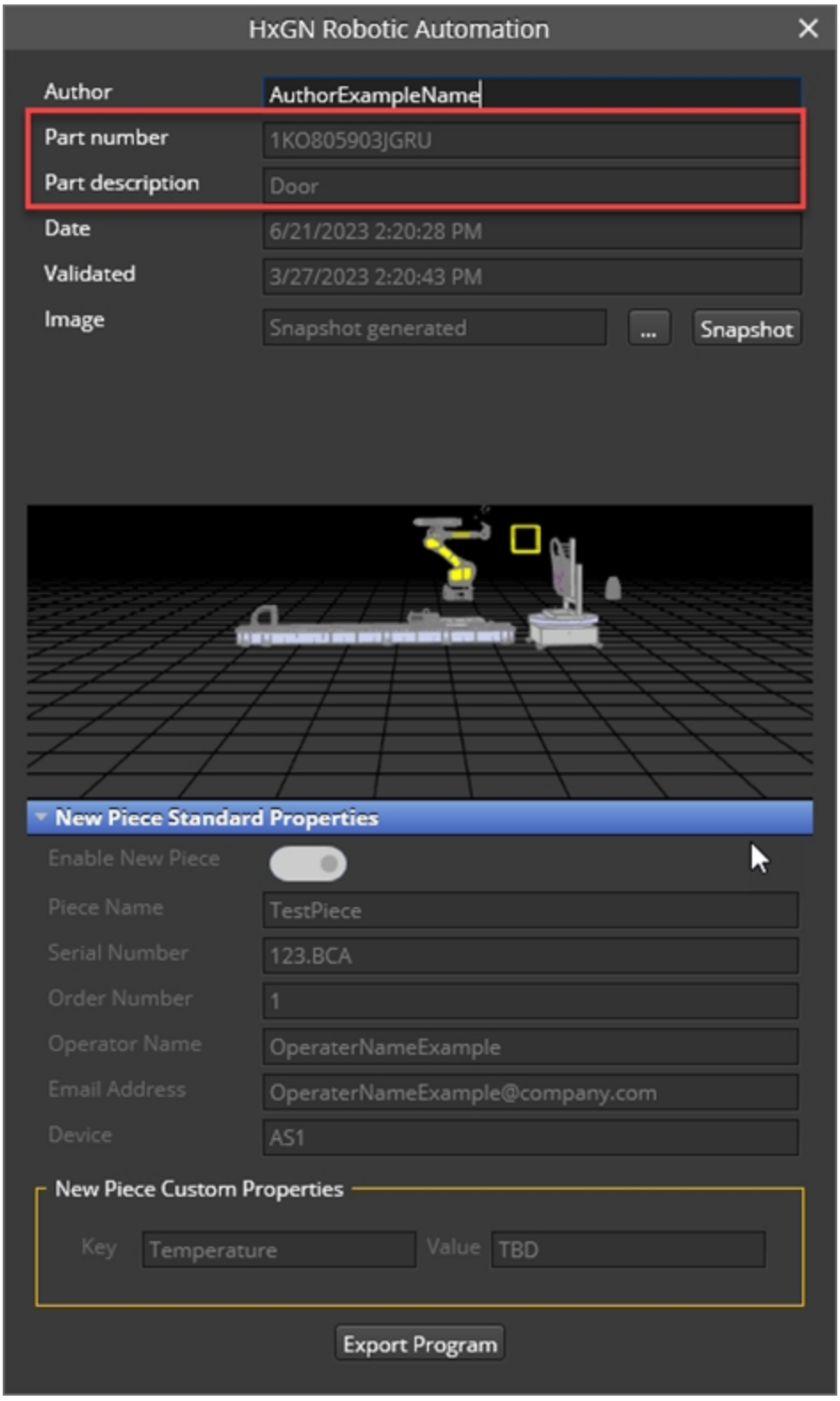
| | |
|---|---|
| | may be needed to refresh the browser page [F5]. |

## Part properties

New part number and description fields are available within Part Information fields in Properties:




When the export program dialog box opens, the fields are automatically filled with the respective values defined in the part properties. In case they are not defined in the properties, they can be filled in the program export dialog box, and they will be also updated and saved in the part properties.




| | |
|---|---|
| Execution | 528 |

Metadata (also called New Piece properties in PolyWorks) can be added to any program.

In case the user renames a program in HxGN Robotic Automation, validates it, runs it in the real cell and exports it to Execution Manager, then Execution Manager will overwrite the existing program name with the new one.

## SQLite

SQL Express of older versions (before 2025) has been replaced with SQLite which is integrated within the instattion. No user actions are required as the migration towards this new database is automatic.

## General User workflow

Execution Manager is a kiosk interface aimed to be used by cell operators to easily execute programs generated with HxGN Robotic Automation and create playlists composed of such programs.

Before start using the Execution Manager, ensure that is possible to communicate on port 80 and that HxGN.RA.ServerGenericHost service is up and running. If both conditions hold it is possible to navigate to *http://localhost/execmanager*.

A program can be:

- Added directly to the home page to be executed
- Added to a playlist in the playlist page

A playlist (having one or more programs) can be:

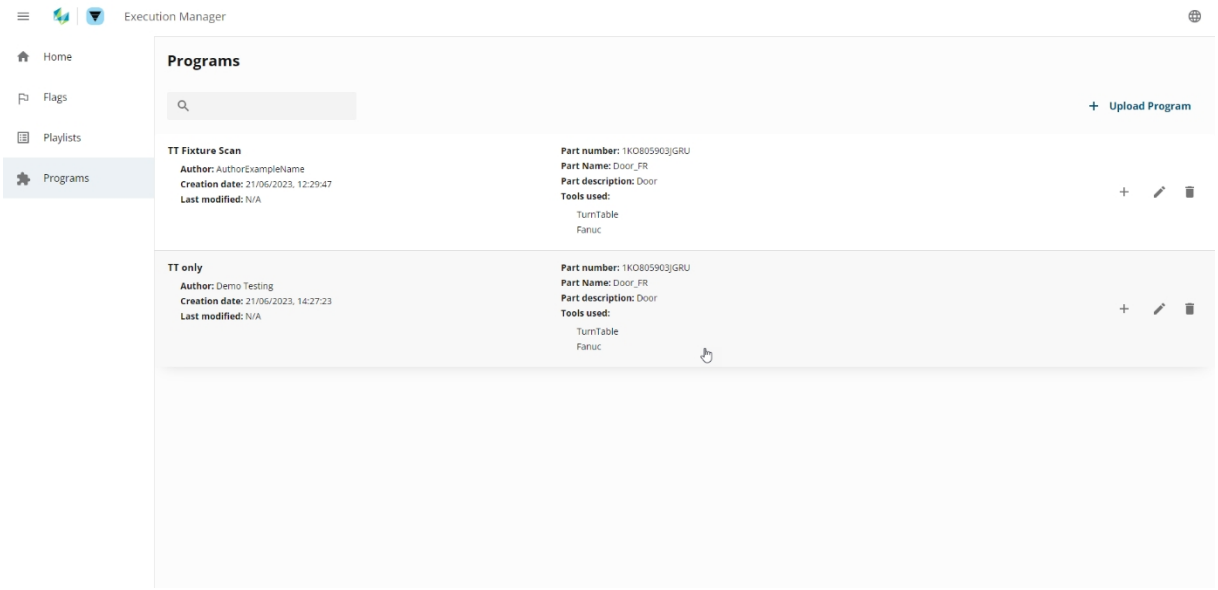
- Added directly to the home page to be executed
- Have flags added in the flags page

Programs and playlists can be executed either in manual or automatic mode:

- In manual mode for both programs and playlist, metadata and the number of repetitions can be changed by the user after pressing the start button before executing
- In automatic mode:
    - all program and playlist cards are greyed out
    - playlists are executed based on the PLC triggering the flags that have been set
    - metadata is populated from the PLC index that was defined for them

## Program page

This page is used to edit programs available in the internal database.




### Editing a program

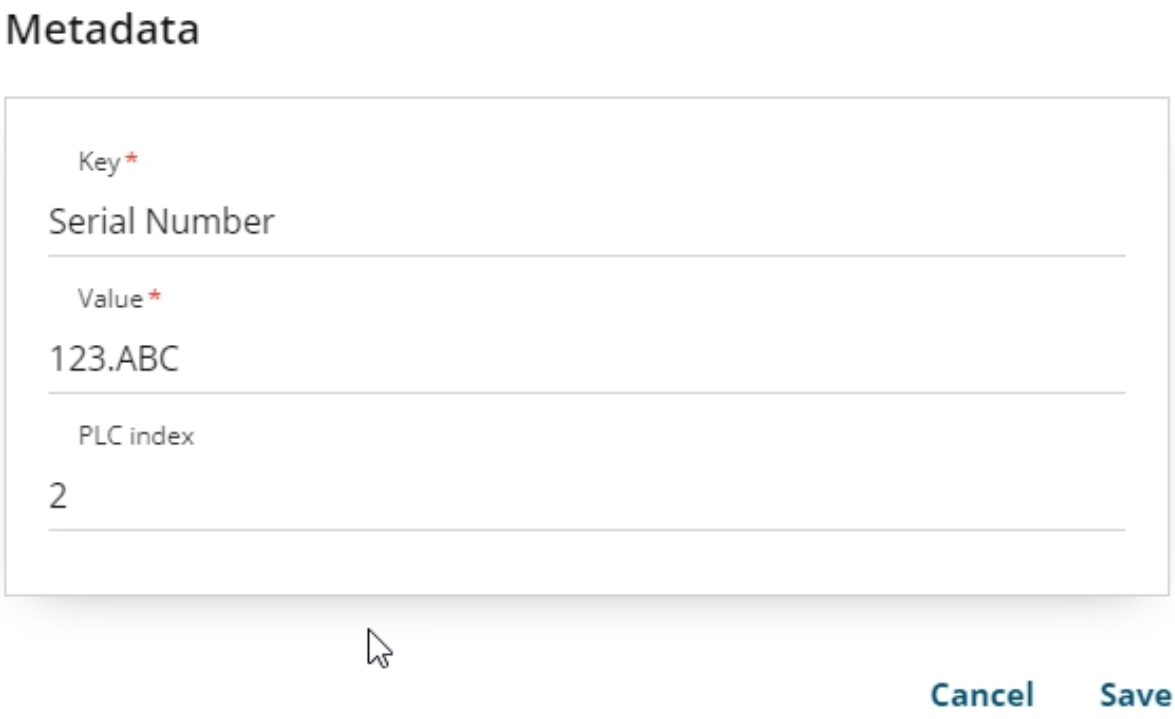
Click on the edit button to open properties page which allows the user to change the program metadata

---

In the program properties the user can edit the Metadata and assign a PLC Index for each field.

If a PLC Index is provided, Execution Manager on execution will search for the field in the PLC at that specified index and copy its contents to the Metadata Value.

In the following example Execution Manger will search in the PLC Index “2” for a value and replace the “123.ABC” value with the value provided by the PLC for the key “Serial Number”.

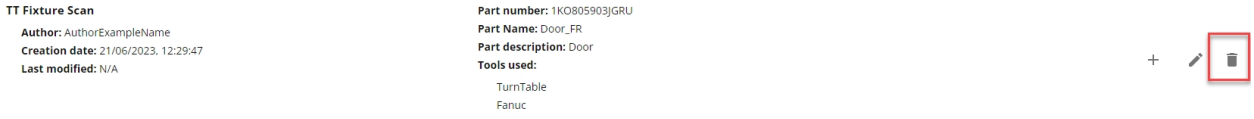



## Removing a program

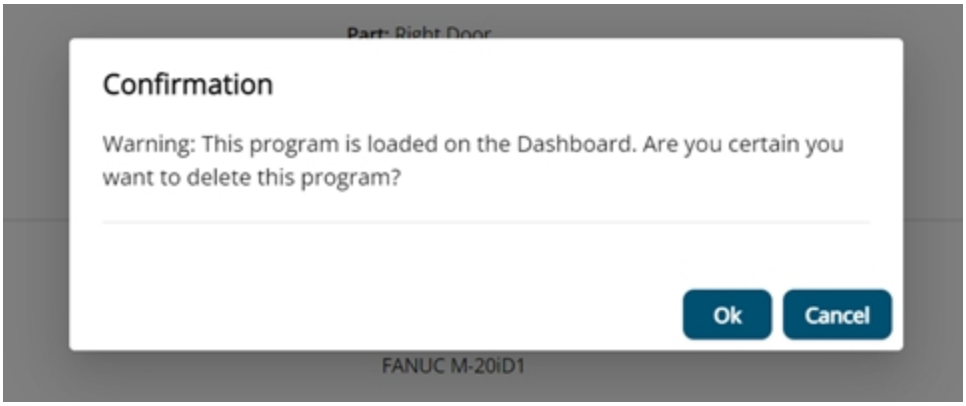
Click on the Delete button to remove a program.

---

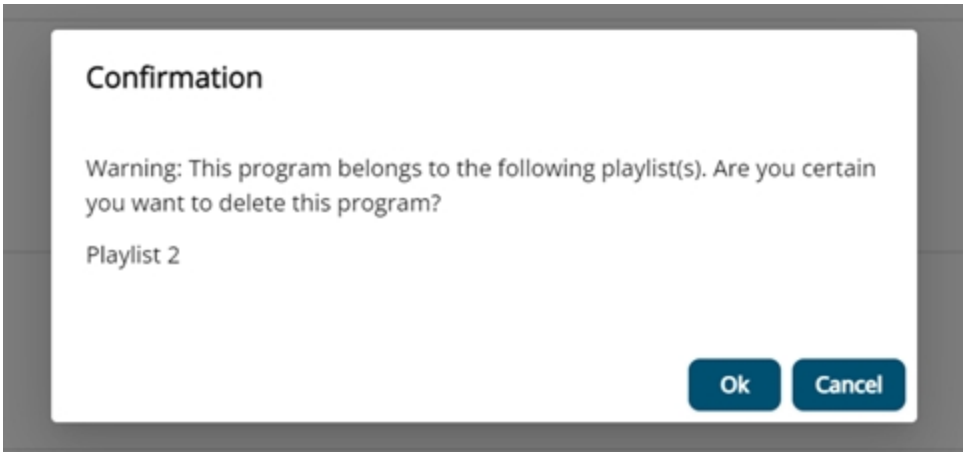
| | |
|---|---|
| Execution | 531 |




If the program is in the home page, but not added to a playlist you will see the following warning message.




If the program belongs to a playlist, you will see the following warning message.

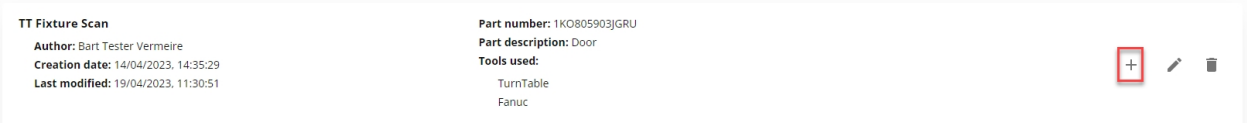



If the program (or playlist having this program) is running, you will see the following warning message.

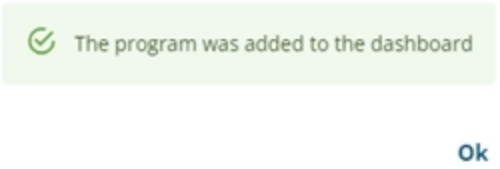
| | |
|---|---|
| Execution | 532 |

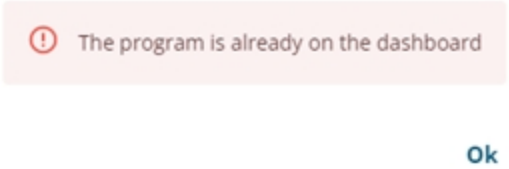
## Adding a program directly to the dashboard

Click on the + button to add the program directly to the home page.




A confirmation message will appear or an error in case it already exists on the home dashboard.




## Playlist page

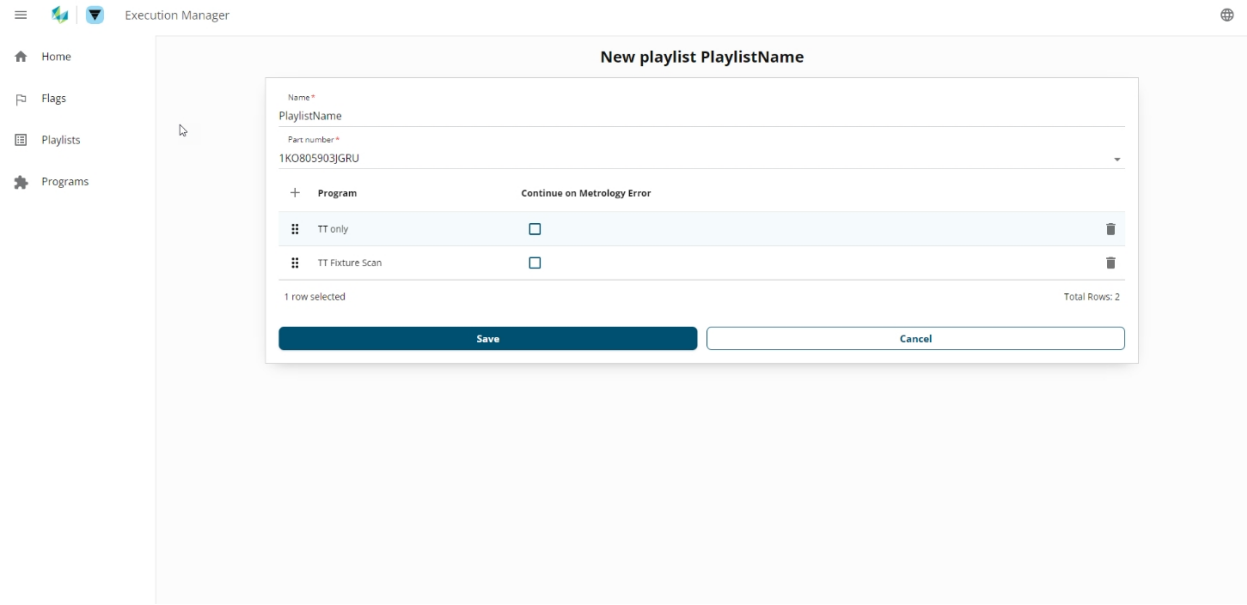
This page is used to create and edit playlists, i.e., sequences of programs to be executed either manually by the operator or automatically triggered by a PLC. How to set a flag to trigger the automatic execution of a playlist is explained in chapter Flags_page

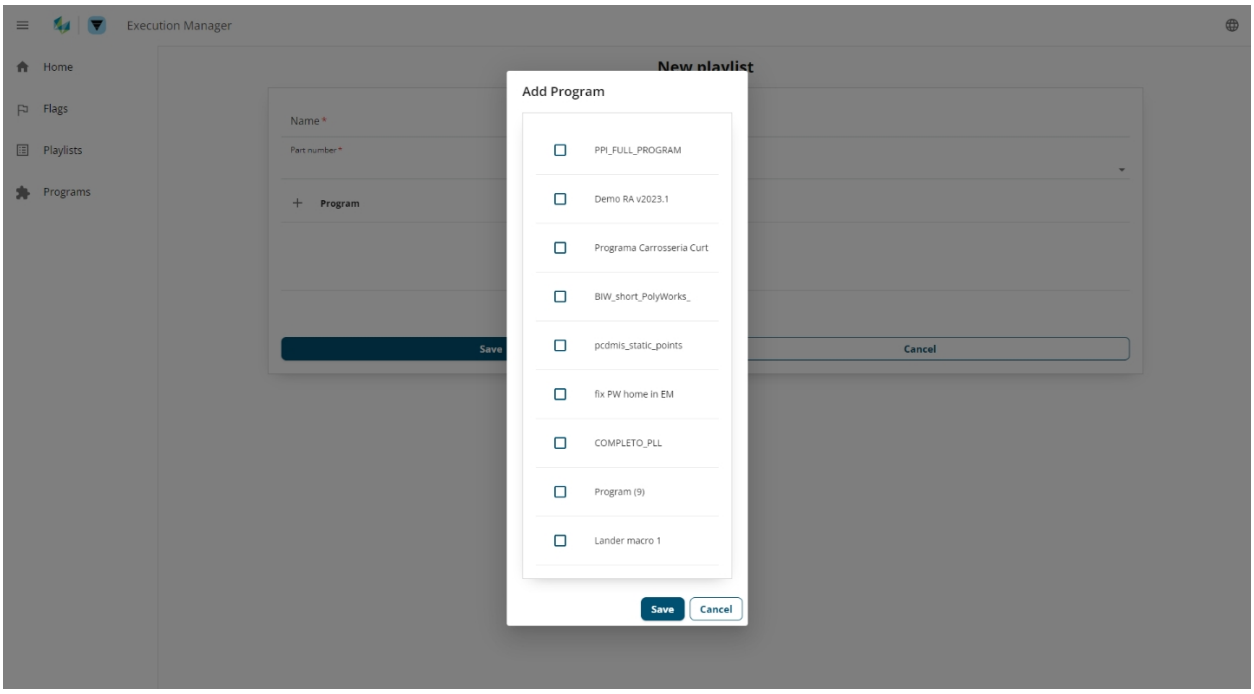
## Creating a playlist

Click on the **Add Playlist** button:

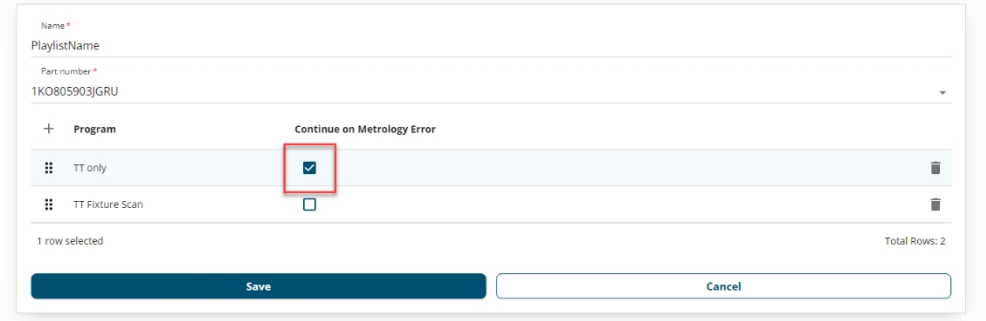
---

Fill in the new **PlaylistName**, select which **Part number** this playlist is associated to from the drop-down menu. This will show only the programs belonging to that part when the user clicks on the + button in the Programs section to add a new program to the playlist.



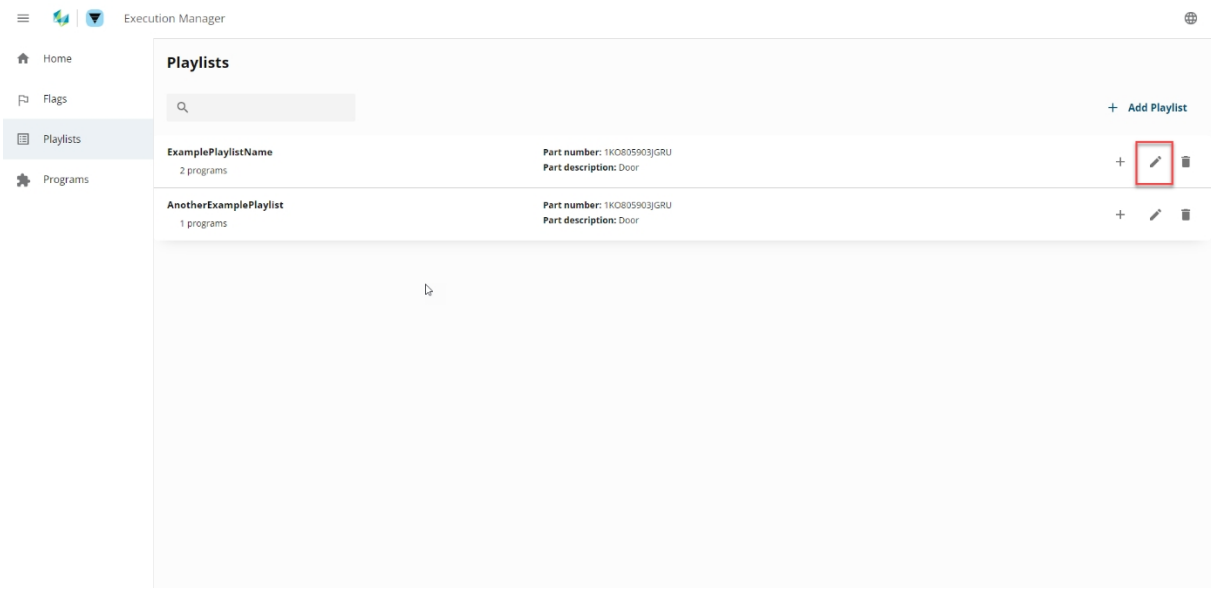



After you added a program to the playlist you can set **Continue On Metrology Error** checkbox for this program. This will cause the playlist either to continue to the next program in the playlist even if there is a Metrology error on the current program or to abort all further programs in the playlist.




## Editing a playlist

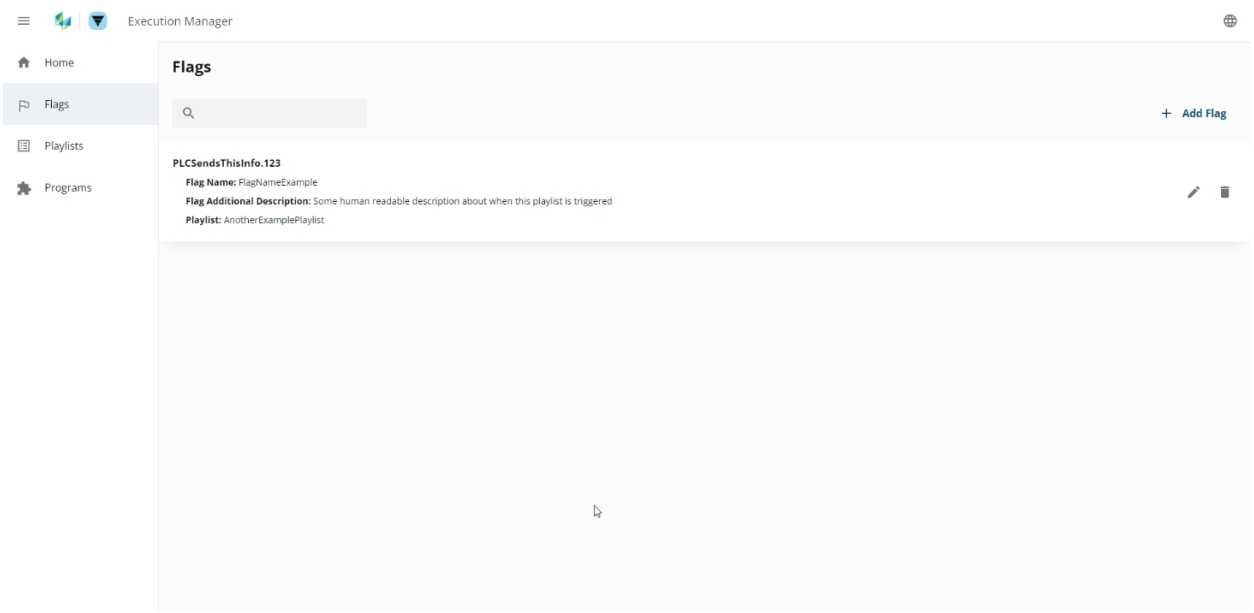
To edit playlist, click the **Edit** button on the playlist. The same playlist editing form as described in the program section will be shown.




## Flags page

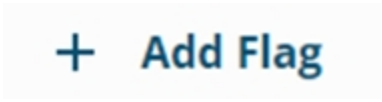
This page is used to create and edit flags, i.e., PLC values that are associated with a specific playlist execution.

When the Execution Manager is set to Automatic mode, it will wait for the PLC to communicate a flag code and execute the associated playlist. Click on **Flags** to navigate to the page showing the Flags registered in the EM:

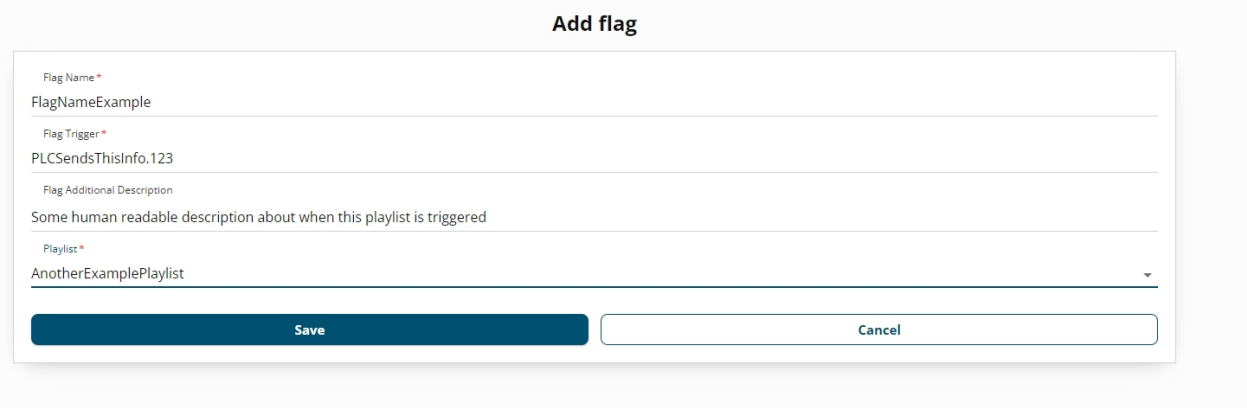



## Adding flags

To add new trigger, click on the **Add Flag** button:




Fill the **Flag Name**, **Flag Trigger**, **Flag Additional Description** and choose the **Playlist** this Flag must be connected to. The Flag Trigger is what the PLC will send to start the corresponding playlist execution.



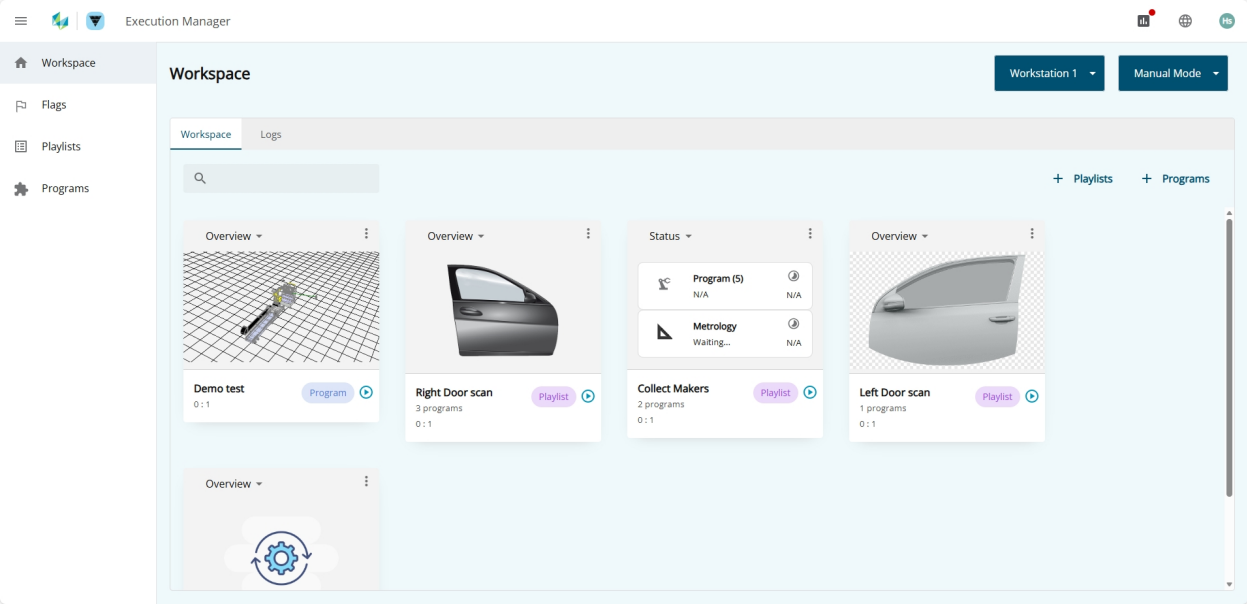

## Workspace (Dashboard) page

In the workspace page of the application, the user will be able to add either playlists or programs.

There are two separate buttons, one for adding playlists and another for adding programs to the home page.

Each of the buttons will open its own selection dialog, with all available playlists or programs.

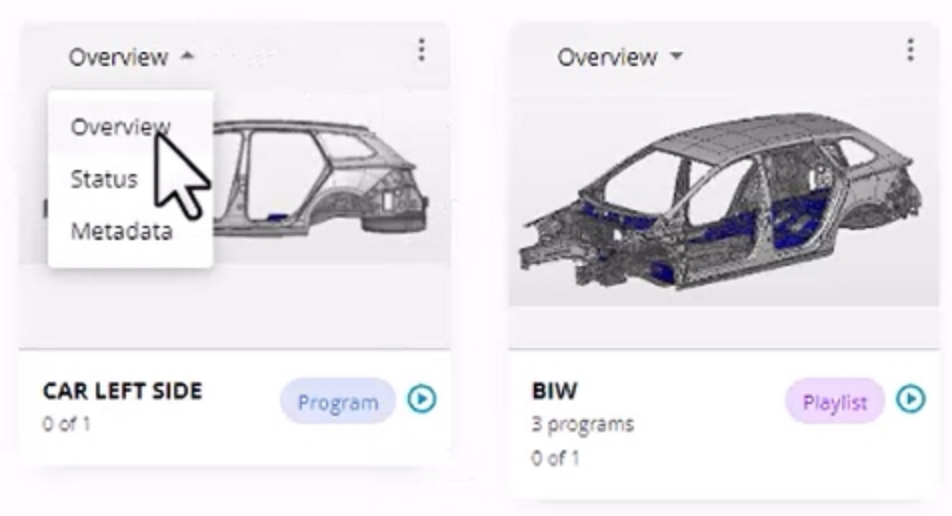
Select + **Playlists** or + **Programs** to open a list that can be easily filtered:




Both programs and playlists will appear as cards within the Workspace. The chip within the card allows the user to distinguish between a Playlist and a Program.

By clicking on the drop-down button on the top left of the card, the user can select the view between the three different views:

- **Overview** (default): showing the program/playlist image, the name of the program/playlist, and the number of repetitions. In case of a playlist card, also the number of programs contained inside the playlist is shown.



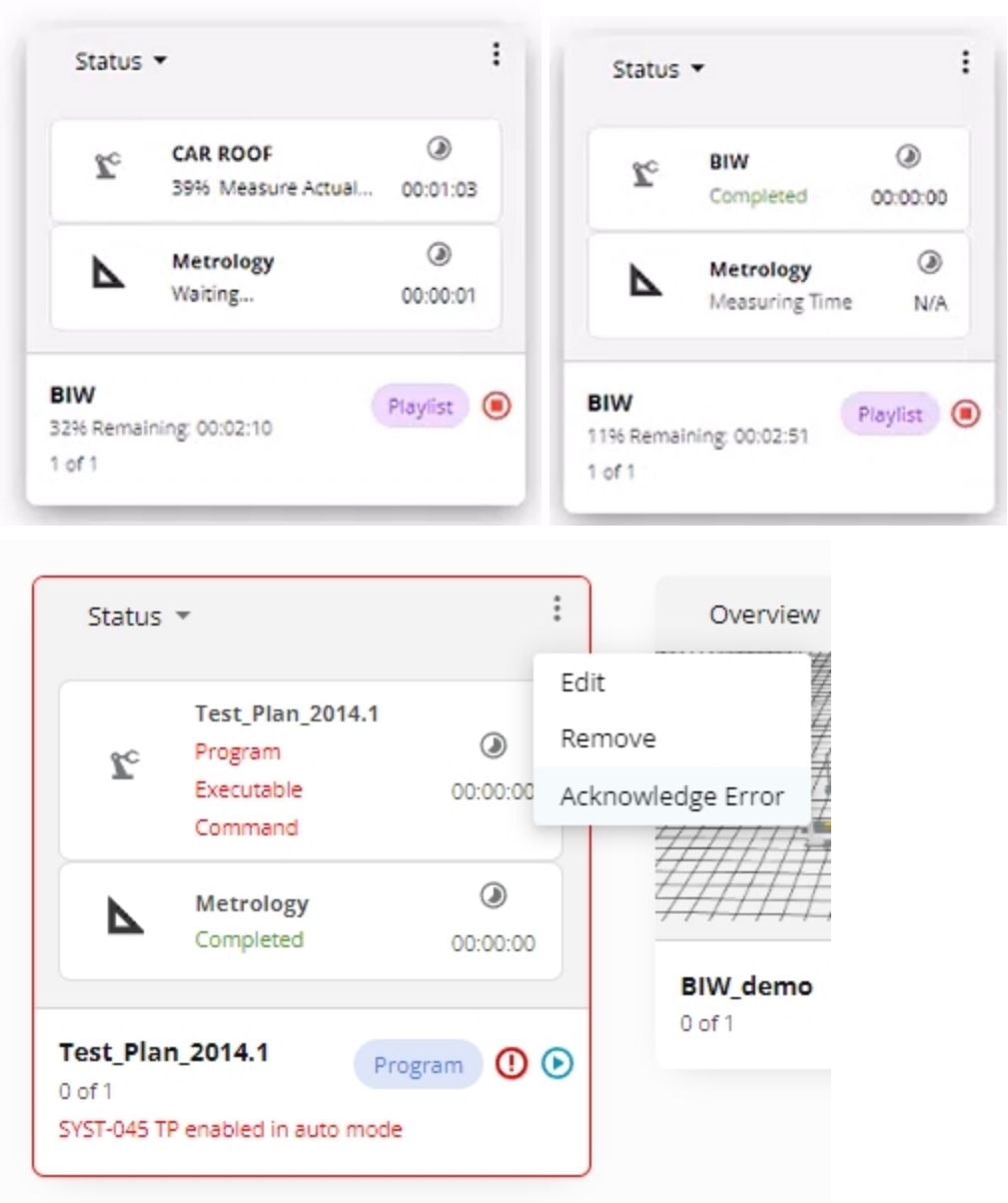

- **Status**: showing the execution progress (% of the total time) and the estimated remaining time for both the scanning program ( ) and the associated metrology routine ( ) that will evaluate the data received at the end of the scanning program. In the bottom part of the card, it will be also shown the overall execution progress and estimated remaining time considering both.

  In case the metrology routine is executed for the first time on the robotic cell, there cannot be an estimated time, so it will be calculated showing “Measuring Time” blinking message and stored in the memory for the next execution.

  In case of an error during the execution the card will be highlighted in red, and the error shown. It is possible to acknowledge it by clicking on the three dots and selecting **Acknowledge Error**.

---

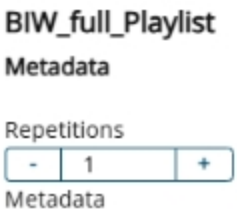
| | |
|---|---|
| Execution | 540 |

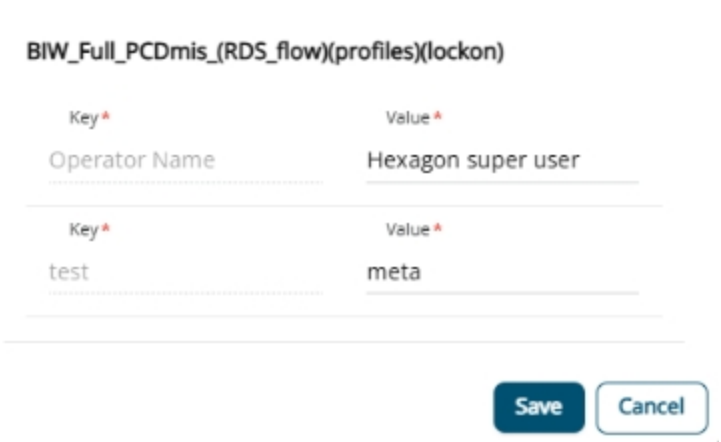



The execution times are stored within the program properties and can be reset by the user by clicking on the cross button.

| | |
|---|---|
| Execution | 542 |

**Metadata**: showing the program/playlist associated metadata exported from RA or postedited within EM program/playlist properties or before execution.




- 


- **Flags** (only for Playlists): showing the flag code that will trigger the execution from the PLC in automatic and semi automatic mode.

### Select the Workstation

In case of having a cell with 2 workstations we can open 2 Execution Managers (1 in each cell PC of PRESTO XL) and run 2 different programs or playlists at the same time, as long as the 2 workstations don't contain any overlapping cell elements (tracker, scanner, robots, etc...).

More information on where the Metrology results are saved in the following section:Metrology results with 2 workstations and PC's

## Set the Execution mode

In the home page, on the top right there is a possibility to select a mode in which a program or a playlist can be executed. There are three options: automatic, semi automatic, and manual.

- In automatic mode, the PLC will control the execution of the playlists by sending the relative Flag Code to Execution Manager.
- In semi automatic mode, the PLC will control the execution of the playlists by sending the relative Flag Code to Execution Manager. The difference wit the automatic mode, is that when a playlist is identified by the corresponding Flag Code, the user will see a pop-up showing the selected playlist asking to confirm or cancel the execution.
- In manual mode programs and playlists will run in the order defined by the user.

## Executing a playlist in automatic and semi automatic mode

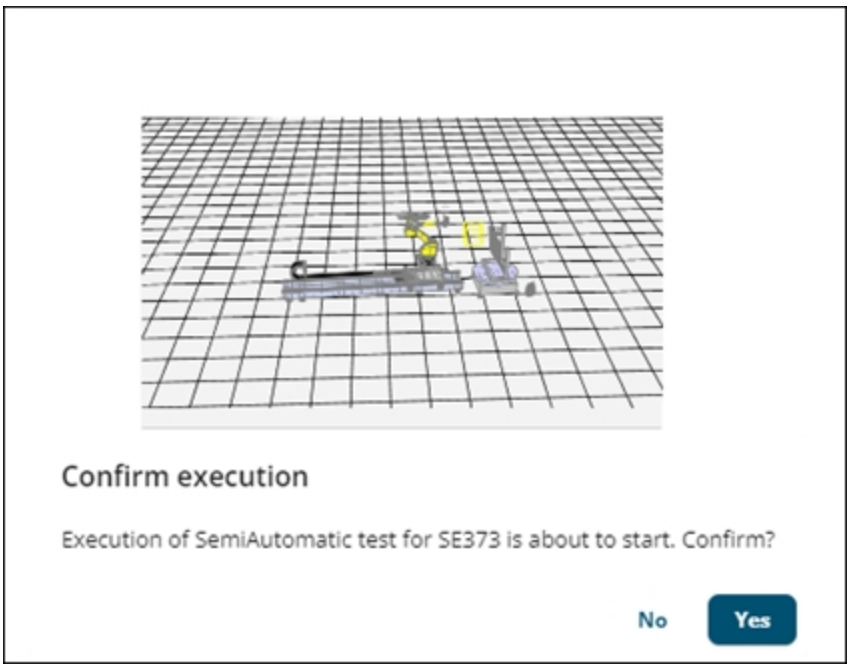
All the programs or playlists with no triggers will be greyed out and cannot be selected. The playlists with triggers are the ones Execution Manager will be watching.

Once a trigger is sent by the PLC, Execution Manager will load the playlist and confirm the part with the PLC before starting the program using the Metadata populated by the PLC, fetching it depending on the flag code.

The PLC needs to send the flag code to run the playlist associated with it. That is why in automatic mode programs and playlists without flag codes are greyed out, to let the user know that those programs or playlists cannot run automatically.

The flags can be visualized by using the drop-down menu in the Card view and selecting the Flags view.

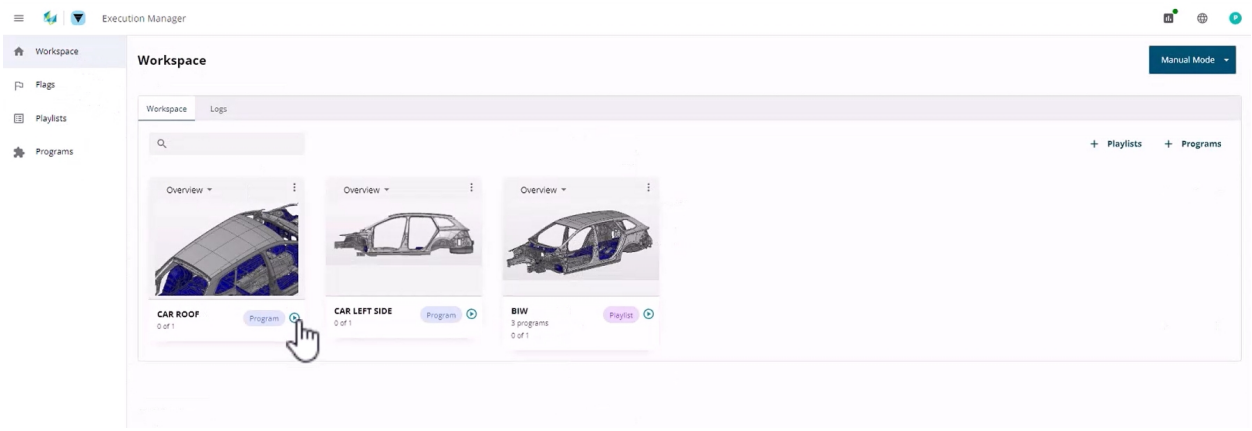
In case the semi automatic mode is selected, instead of starting immediately, the playlist execution will wait for the confirmation of the user through the corresponding pop-up menu, where it is also possible to cancel the execution.

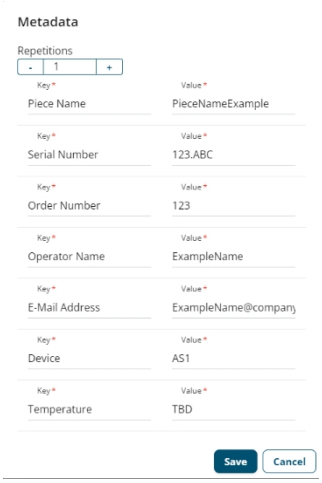



## Executing a program or a playlist in manual mode

In manual mode, after the user presses on the Execute button ( ), a new form will appear allowing to set the number of repetitions for the program. In manual mode the form will also have metadata that the user can edit before sending the execution.

| | |
|---|---|
| | The “operator” metadata is filled automatically using the username of the logged user. |

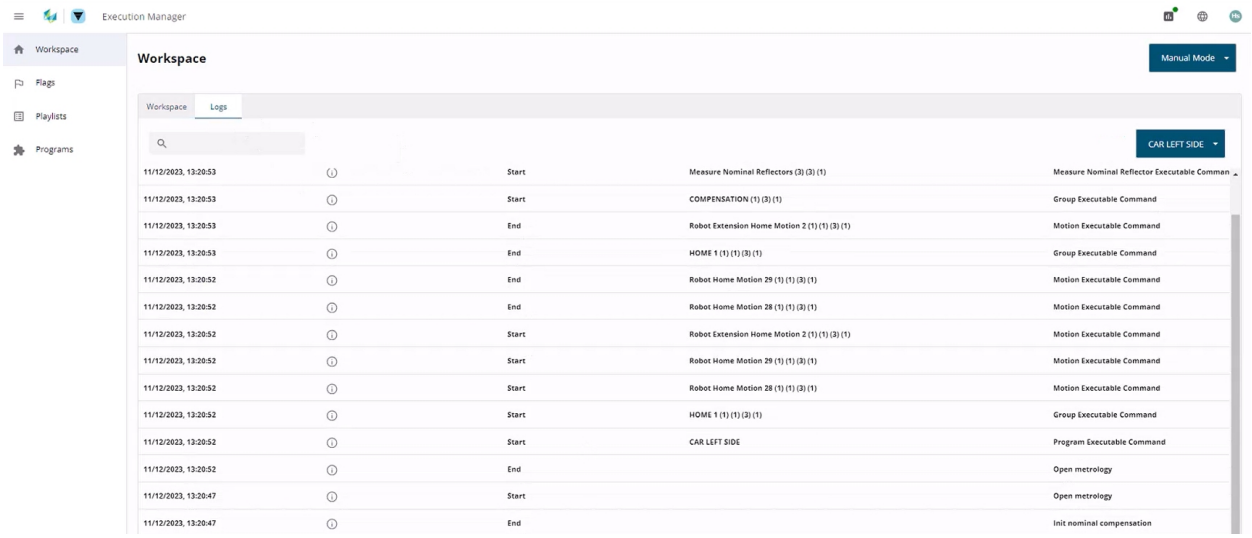



## Logs

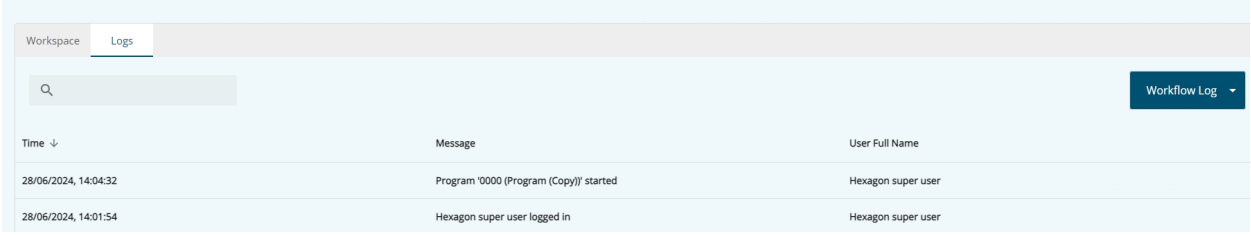
Execution Manager shows the program logs of the programs that have been executed.

Using the log drop-down button on the right, it is possible to filter the logs depending on the related program/playlist (e.g., “CAR LEFT SIDE” in the picture below), showing the history of their last execution .



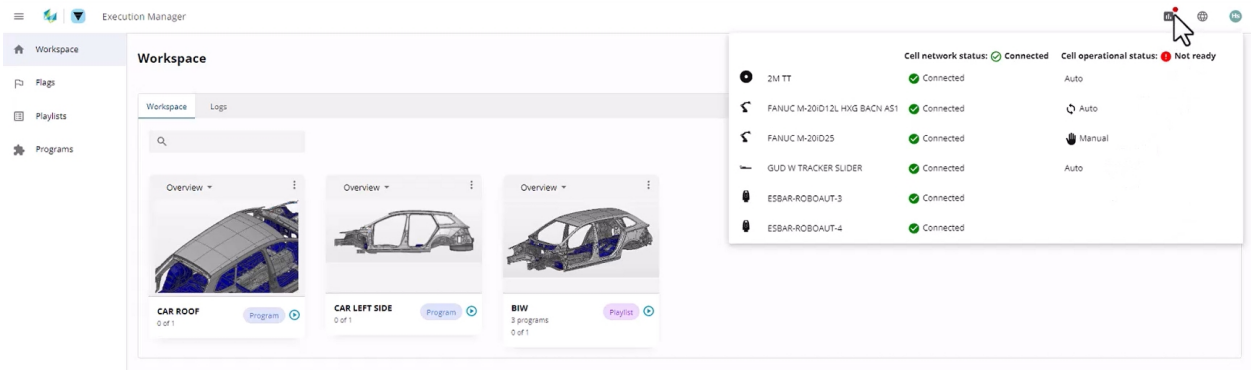

It is also possible to check the logs of the overall Execution Manager **workflow** by selecting Workflow Log from the drop-down menu. This will allow to see e.g., which programs/playlists started, ended, aborted/failed during the Execution Manager usage, keeping in memory the data of the last three months usage.

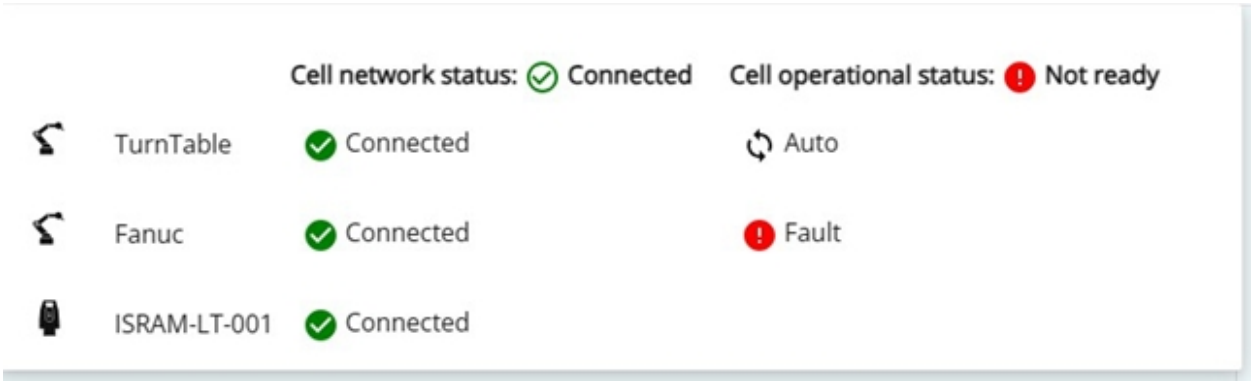
The workflow log will also show who logged in and who executed what program or playlist. Due to technical limitations, we cannot show who logged out as the application could be closed at any point without the proper logout procedure.



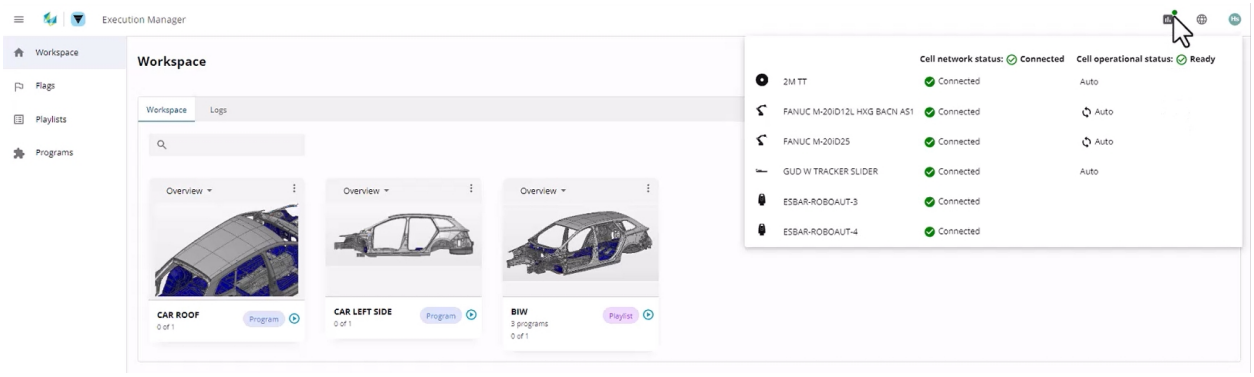

## Cell status

The cell status is shown on the top-right corner with a red dot in case not all the cell components are connected, and their operational status is not in auto mode (either because in Manual or Fault status).




If all cell components are connected and running in automatic mode, meaning that the Execution Manager is allowed to take control of the cell and execute programs or playlists, the status dot will be shown in green.



# Metrology Connection

Metrology features is a term used in metrology to refer to specific parts of a piece to be measured with a certain level of accuracy and are usually defined within an inspection plan and managed inside a metrology software. Features can be imported in HxGN Robotic Automation by establishing a connection with a metrology application. Alternatively, they can be generated inside HxGN Robotic Automation using the specific Feature Generation Mode.

## Adding Metrology to the Metrology drop down

The Metrology drop down is searching within a folder where any metrology application can add their dll's. By default, we support PolyWorks and a fixed interface with Inspire and PC-DMIS.

## Using Workstations and Metrology results

HxGN Robotic Automationand Execution Manager will save the results in the Metrology project according to the following table:

A program run from HxGN Robotic Automation will prioritize to open the Metrology application from that machine.

### PC-DMIS and Inspire

| HxGN Robotic Automation | PC-DMIS | Inspire |
|---|---|---|
| Run RA program for a workstation on PC 1 (primary) | Opens on PC 1 | Opens on PC 1 |

| HxGN Robotic Automation | PC-DMIS | Inspire |
|---|---|---|
| Run at the same time a 2nd RA Program on PC 1 (primary ) or PC 2<br>Different Workstation then above | Opens on PC 2 | Opens on PC 2 |

| HxGN Robotic Automation | PC-DMIS | Inspire |
|---|---|---|
| Run RA program for a workstation on PC 2 | Opens on PC 2 | Opens on PC 2 |
| Run at the same time a 2nd RA Program on PC 1 (primary) or PC 2<br>Different Workstation | Opens on PC 1 | Opens on PC 1 |

A program run from Execution Manager will prioritize to open the Metrology application on the primary Generic Host PC.

| Execution Manager | PC-DMIS | Inspire |
|---|---|---|
| Run Execution Manager program for a workstation on PC 1 (primary ) or PC 2 | Always opens on PC 1 (primary) | Always opens on PC 1 (primary) |
| Run at the same time a 2nd Execution Man- | Opens on PC 2 | Opens on PC 2 |

---

| Execution Manager | PC-DMIS | Inspire |
| --- | --- | --- |
| ager Program on PC 1 (primary) or PC 2 Different Workstation | | |

### PolyWorks

| | |
| --- | --- |
| | PolyWorks allows 2 instances on the same machine. |

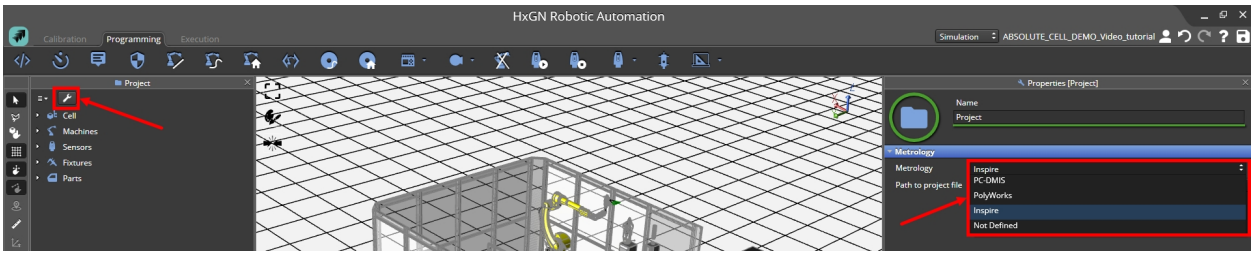
| Execution Manager | PolyWorks |
| --- | --- |
| Run Execution Manager program for a workstation on PC 1 (primary ) or PC 2 | Opens on PC 1 |
| Run at the same time a 2nd Execution Manager Program on PC 1 (primary) or PC 2 Different Workstation | Opens on PC 1 |

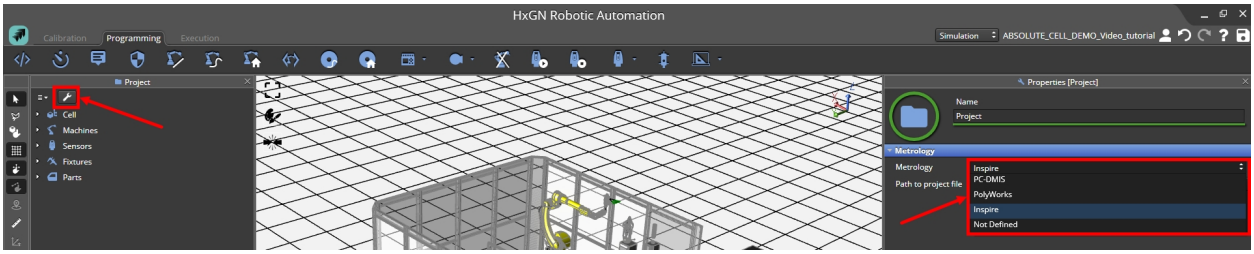
## Import features by establish the connection with a metrology

It is possible to define the metrology application used in the HxGN Robotic Automation project by clicking on Project Properties and selecting it using the relative **Metrology** drop-down menu.application

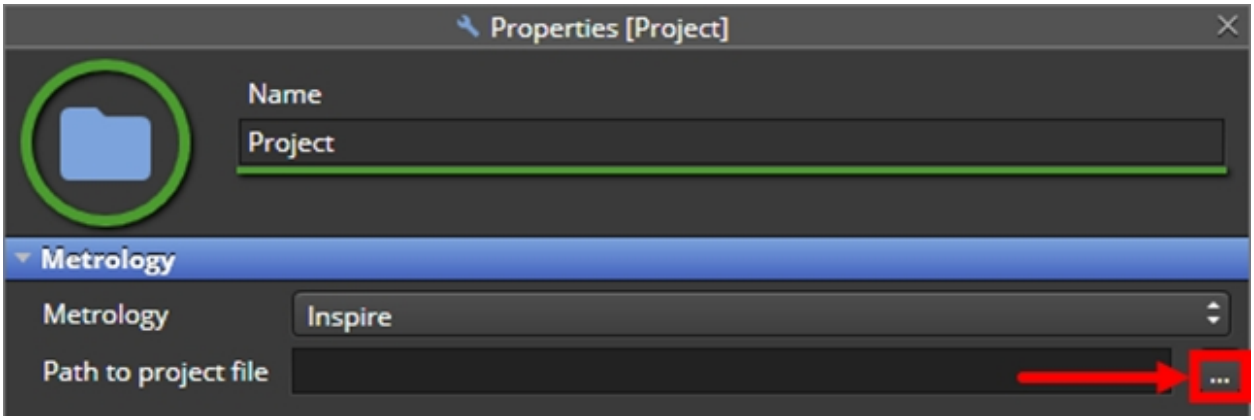
---

| | |
| --- | --- |
| Metrology Connection | 551 |




After selecting the correct metrology application, the user can pick the project to be used.




Currently, HxGN Robotic Automation supports as metrology applications:

- Inspire
- PC-DMIS
- PolyWorks

The related user workflows are explained in following chapters.

---

## Inspire

HxGN Robotic Automation application can support connection to Inspire metrology software to perform measurement analysis based on the data collected using a scanning device.

For more details on Inspire, please refer to:

- https://www.hexagonmi.com/products/metrology-software/inspire
- https://discoverinspire.com/download/InspireReferenceGuide.pdf

Inspire inspection software must have a valid Inspire license for scanning purposes.

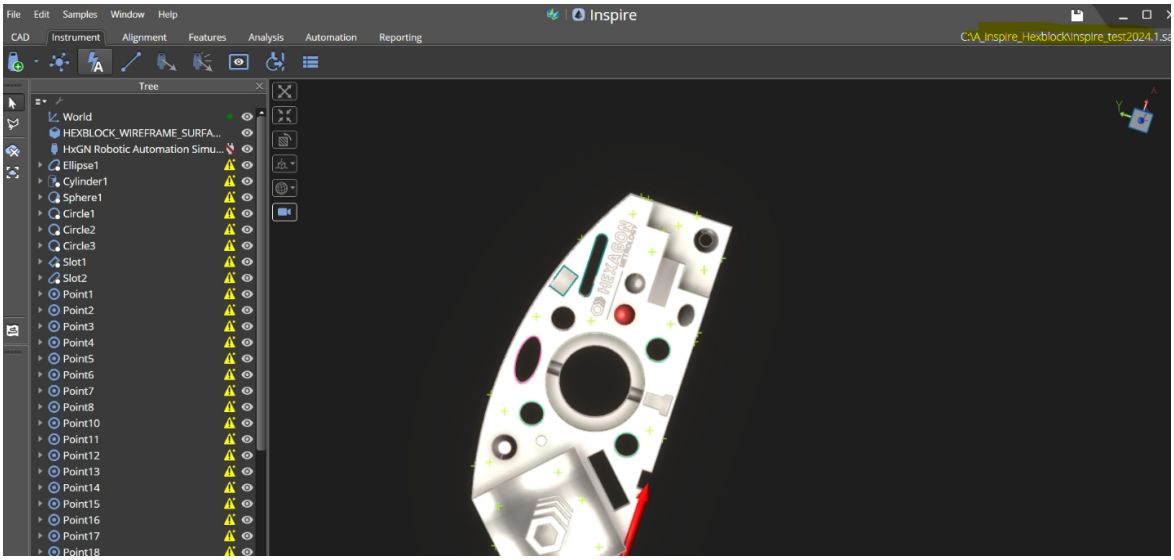
Obtain a license and the correct official software version release which supports the connection with HxGN Robotic Automation.

To configure the Inspire metrology software see chapter Inspire Configuration

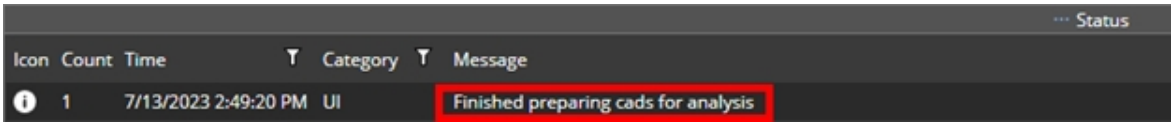
To set up the connection with Inspire and start using it together with HxGN Robotic Automation, follow the next steps:

1. Open Inspire Application official release (See chapter: Minimum Software Version)
2. Import CAD and prepare a project with a part and features.

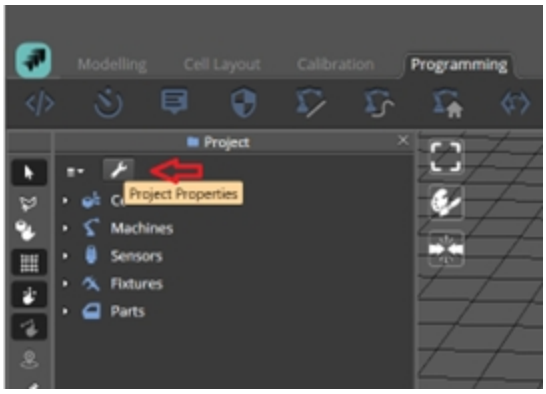
Save it in a specific folder on the PC. Exit Inspire application.

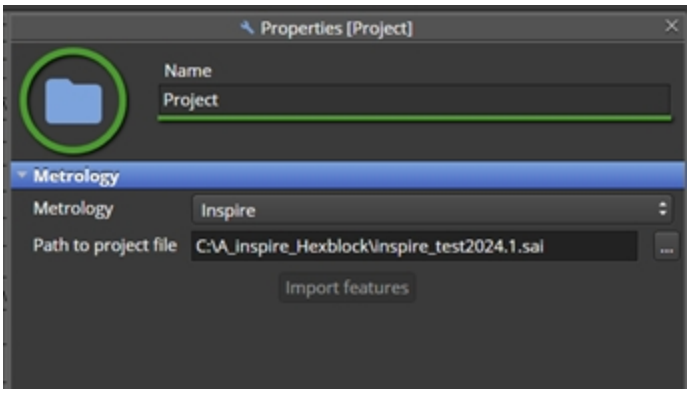
3. 


   Open HxGN Robotic Automation, load the cell project without the part. Import the Part CAD Model in HxGN Robotic Automation application. It must be the same CAD that was imported to Inspire. This step must be done manually. It is important to wait until the following message is displayed within the Status window to be able to interact with the CAD files.

4. 


5. Align the part in the cell and attach it to the fixture if needed..
6. Once part is placed on the HxGN Robotic Automation Cell Project and attached to the Fixture, go to Programming and open project properties, select Inspire from **Metrology** drop-down menu, then select the path to the metrology project (***.sai).

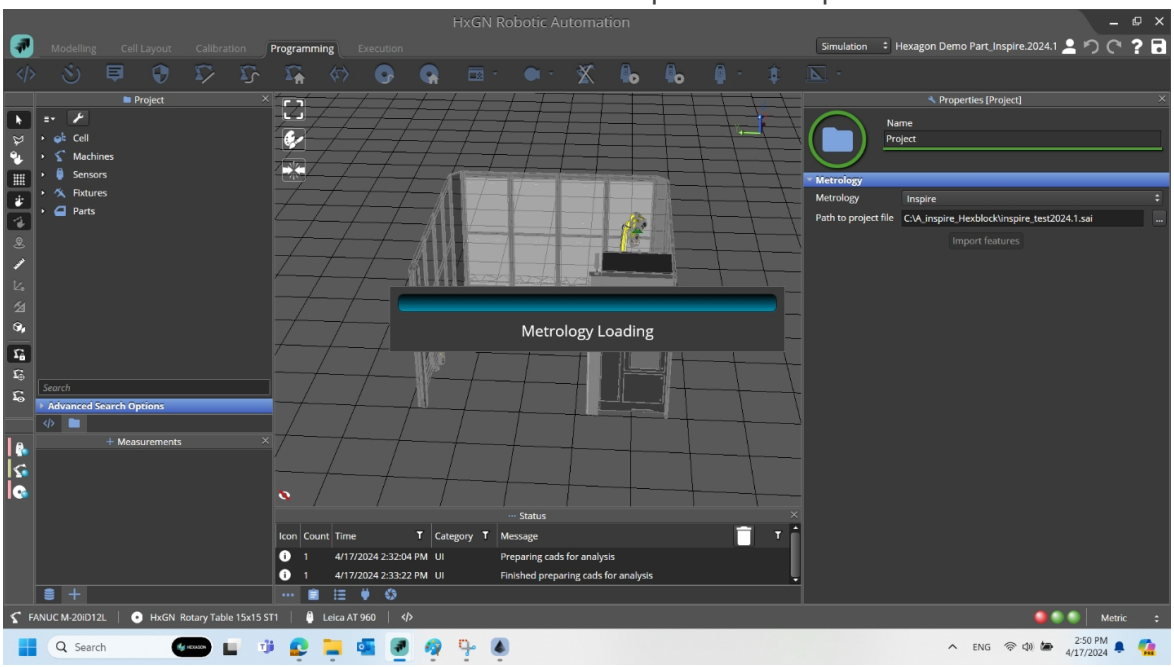


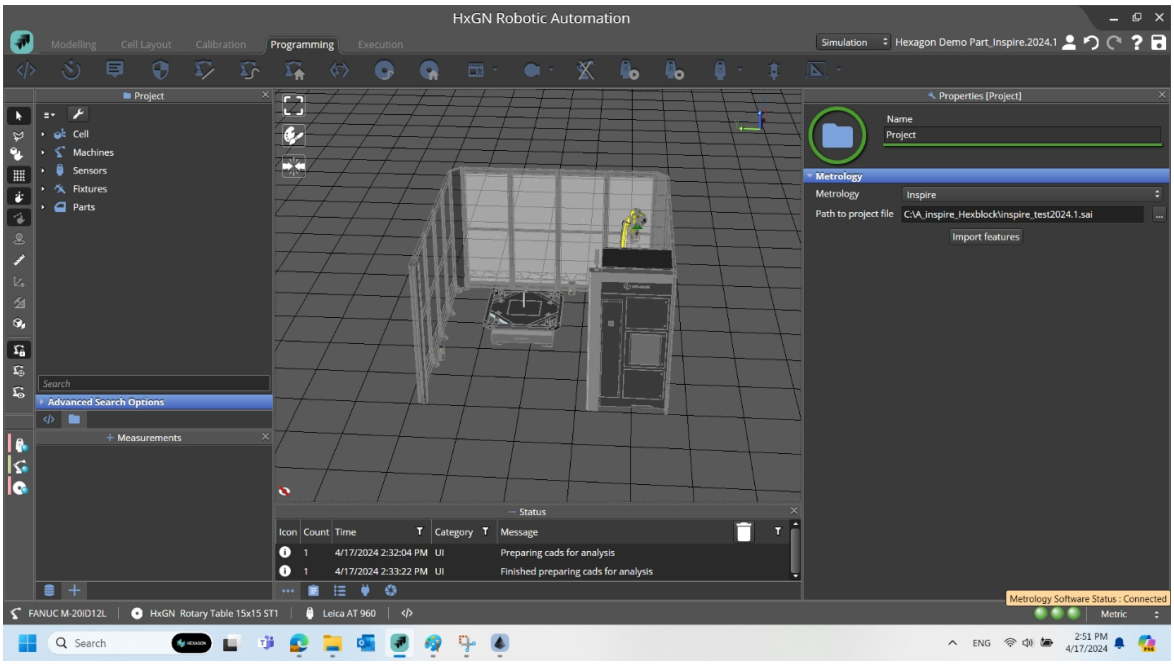

7. At this point, the **Import features** button is disabled, since HxGN Robotic Automation is not connected to inspire yet.

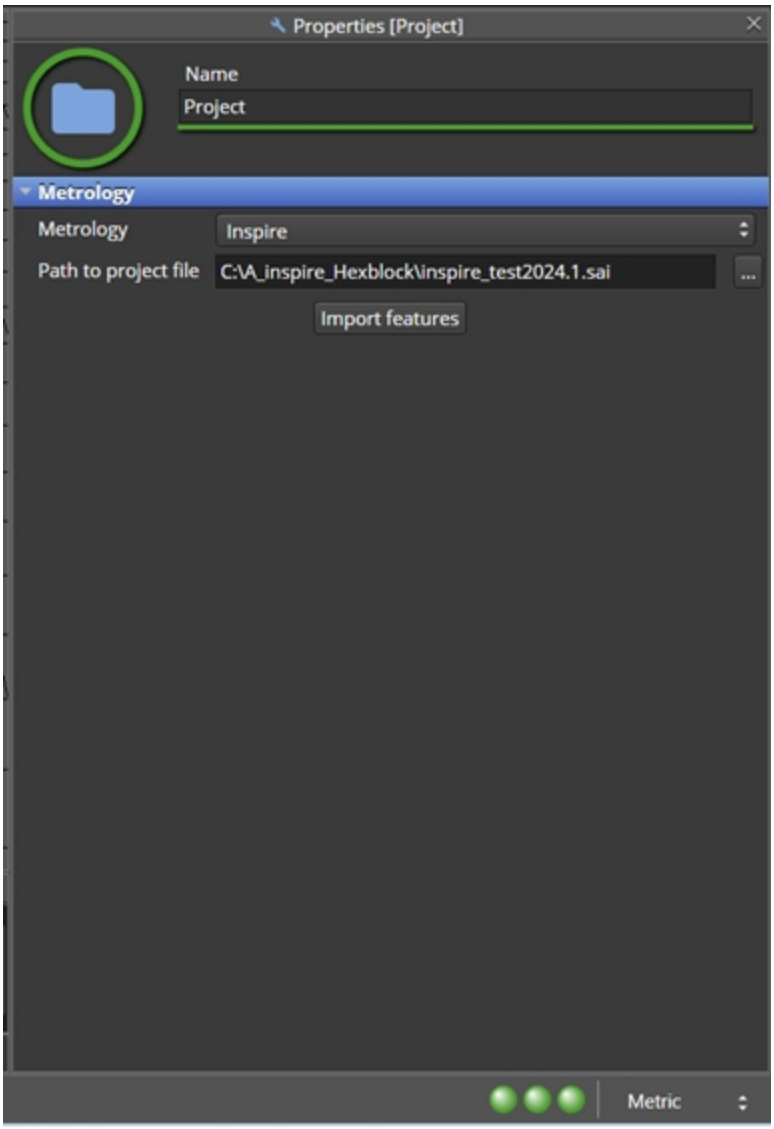
   Press on the metrology traffic light to open Inspire.

Automatically Inspire opens, the project is loaded in Inspire: the connection between HxGN Robotic Automation and Inspire is set up.

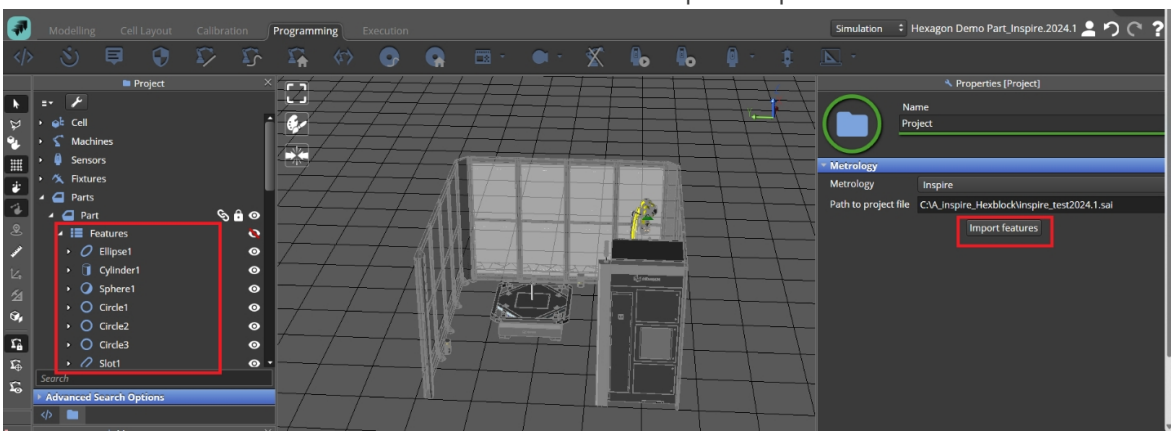



8. Once the connection to Inspire is set up, the Import feature button is enabled.


HxGN Robotic Automation
Modelling
Cell Layout
Calibration
Programming
Execution
Simulation
Hexagon Demo Part_Inspire.2024.1
Project
Cell
Machines
Sensors
Fixtures
Parts
Search
Advanced Search Options
Measurements
Properties [Project]
Name
Project
Metrology
Metrology
Inspire
Path to project file
C:\A_inspire_Hexblock\inspire_test2024.1.sai
Import features
Status
Icon
Count
Time
Category
Message
1
4/17/2024 2:32:04 PM
UI
Preparing cads for analysis
1
4/17/2024 2:33:22 PM
UI
Finished preparing cads for analysis
FANUC M-20iD12L
HxGN Rotary Table 15x15 ST1
Leica AT 960
Metrology Software Status : Connected
Metric
Search
ENG
2:51 PM
4/17/2024

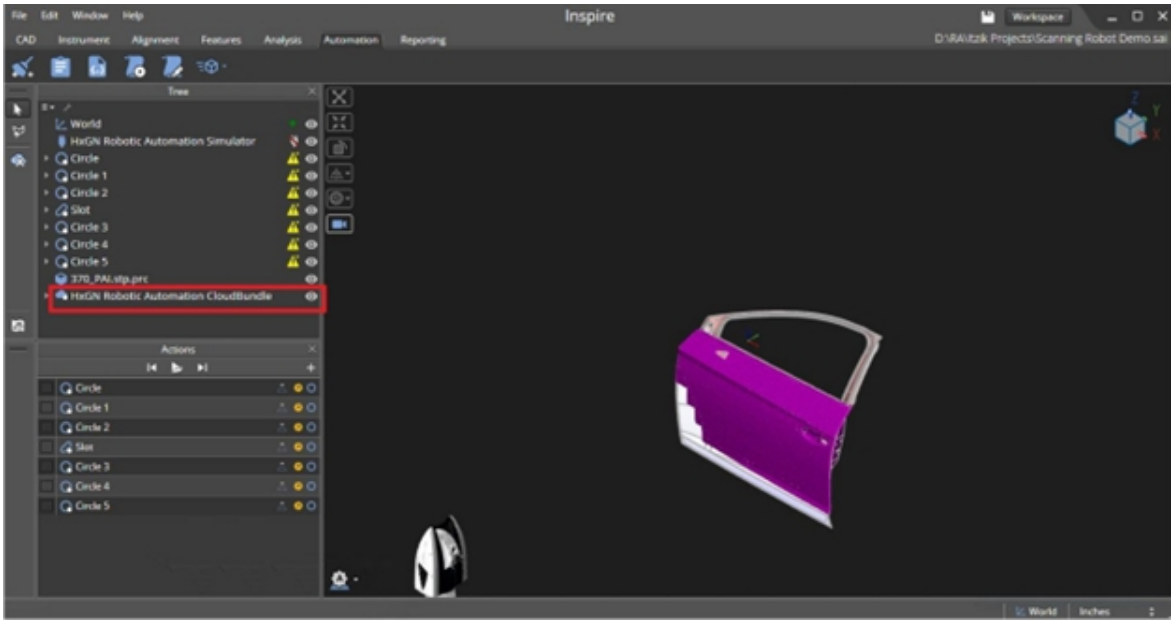

Properties [Project]
Name
Project
Metrology
Metrology
Inspire
Path to project file
C:\A_inspire_Hexblock\inspire_test2024.1.sai
...
Import features
Metric

Click on **Import features**. All featured from the project in inspire will be imported to HxGN Robotic Automation under *Parts* | *Part* | *Features*.

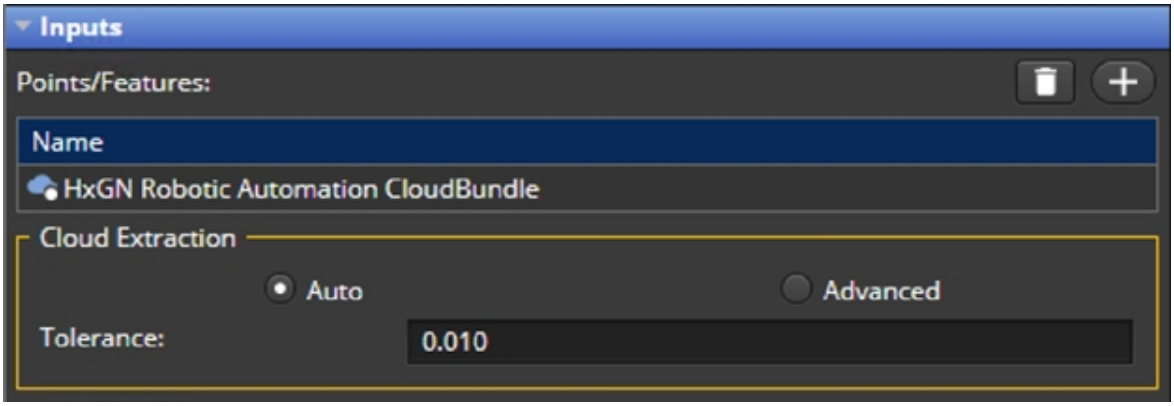
9. 


10. User may add new features in Inspire to be inspected. These will be added to HxGN Robotic Automation only if user repeat the import features again in HxGN Robotic Automation.
11. Once both programs have all the features needed for the part inspection, the user is ready to prepare the robot program in HxGN Robotic Automation.
12. When the robot program is executed from HxGN Robotic Automation, the COP data is sent to Inspire and stored in the **HxGN Robotic Automation**

**CloudBundle** object.




User must make sure that the properties of the features inserted from the CAD file in Inspire are set to use the HxGN Robotic Automation CloudBundle as the Input for the Feature Extraction.

13. 


| | |
|---|---|
| ⓘ | Features created in HxGN Robotic Automation are not sent to Inspire. Inspire is the primary application on the Inspection plan side and if the user wants to include features to ease the programming task but which |

| | |
|---|---|
| | are not relevant for metrology computation, these can be inserted in HxGN Robotic Automation and will not be added to the Inspire Project. |

## PC-DMIS

PC-DMIS must be licensed with the following minimum required options:

- ROY AUTOMATE
- CAD++ OFFLINE
- CMS106 Laser Interface

Recommended options depending on the application or CAD package in use by specific customer are:

- CSV-Converter
- Toolkit (ESF)
- SolidWorks
- Catia V5 or V6

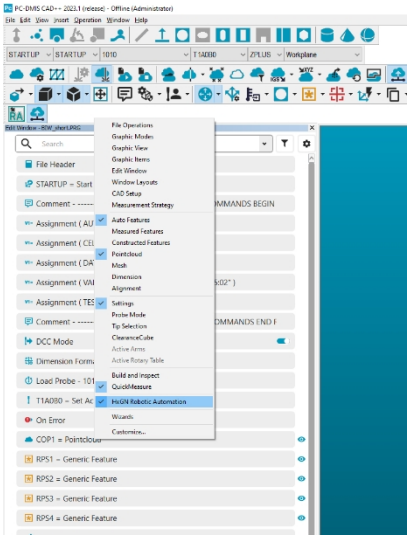
To configure the PC-DMIS metrology software see chapter PC-DMIS Configuration. For additional PC-DMIS configuration (e.g. suppress *laser_onError* messages, please see the PC-DMIS documentation.

The cell layout must be ready, aligned and the robot calibrated in the project. For more information about how to align the cell and calibrate the robot refer to chapter Alignment.htm.

To set up the connection with PC-DMIS and start using it together with HxGN Robotic Automation, follow the next steps:

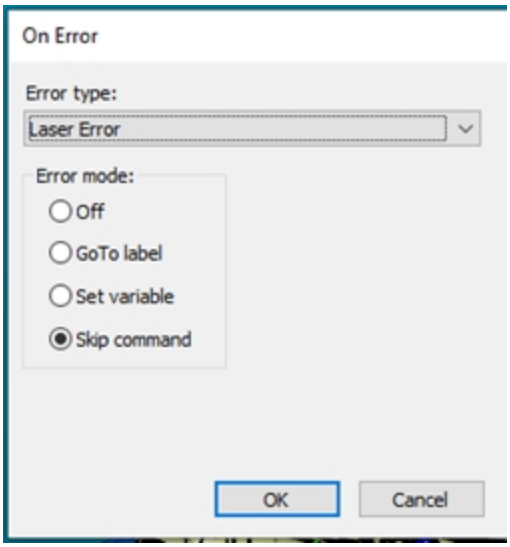
1. Open the PC-DMIS application

   Create/load the PC-DMIS program to be used with the features needed for the inspection plan. To see the HxGN Robotic Automation toolbar, go to View | Toolbars | HxGN Robotic Automation.

2. 


3. Type **ONERROR/Laser_error** in the program to skip all the error dialog boxes popping up in case of errors.

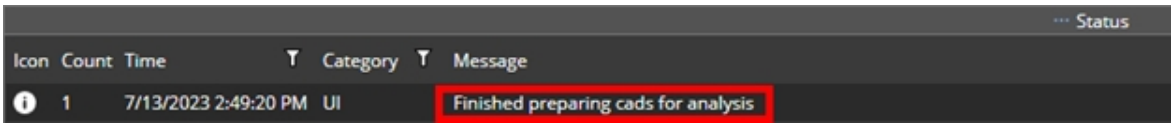
Optionally set **SKIP** in the program configuration to obtain a list of the features that have not been measured, after the program is executed.

4. 

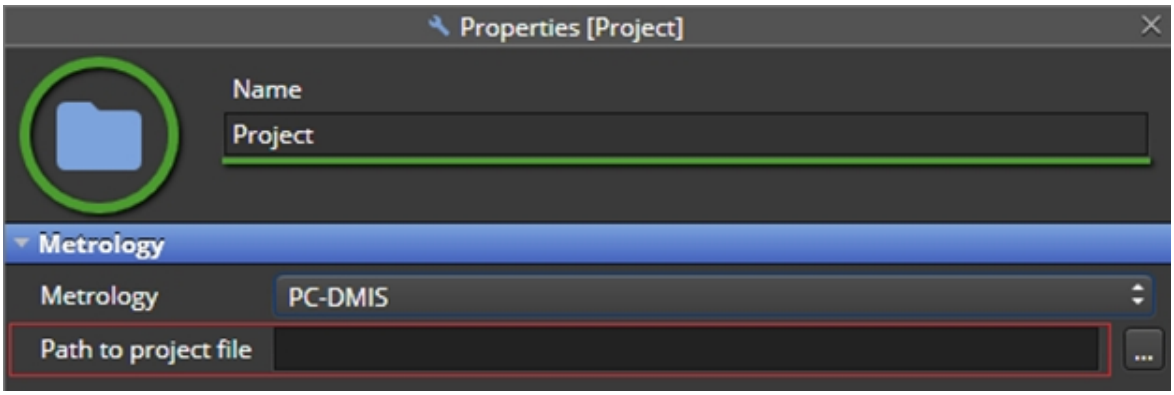

5. Create an empty point cloud.

6. Create a PC-DMIS measurement routine using a laser probe and make sure you have the proper PC-DMIS license. You can either manually create the measurement routine or import a CSV version with the CSV-Converter application.

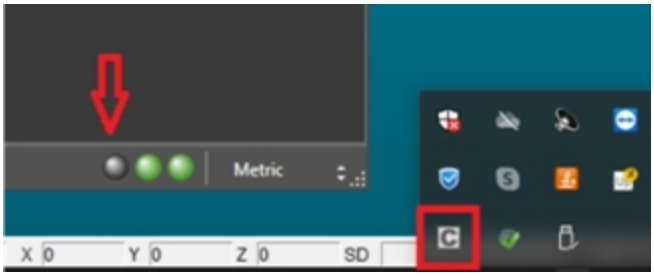
   Open HxGN Robotic Automation, load the cell project without the part. Import the Part CAD Model in HxGN Robotic Automation application. It must be the same CAD that was imported to Inspire. This step must be done manually. It is important to wait until the following message is displayed within the Status window to be able to interact with the CAD files.

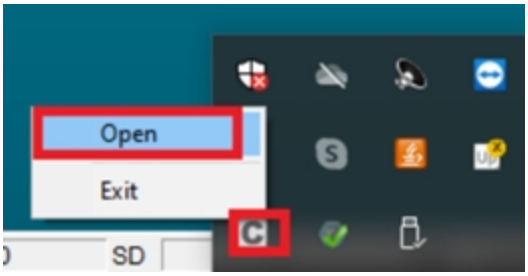
7. 

Within the project properties, define the path to the PC-DMIS project file to be used.

8. 


Click on the left traffic light related to Metrology Software to open the Connector. The icon is grey if there is no connection set up.

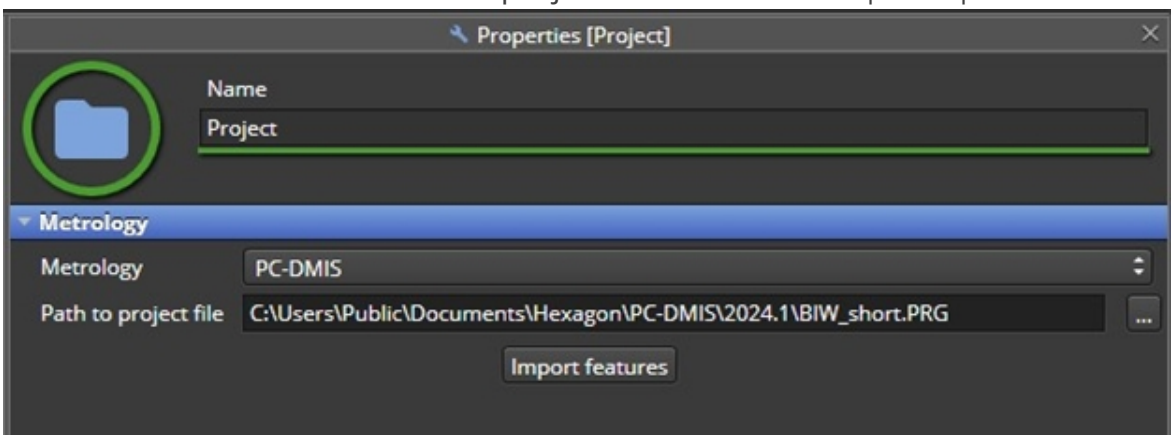
9. 

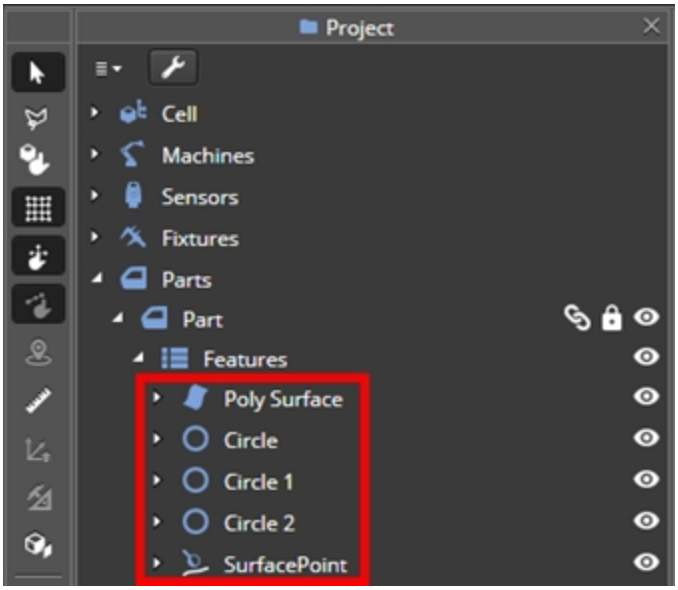




10. After clicking it, the color will change to yellow during the connection process and then turn to green when the connection is set up. If there is any error during the connection, the icon will turn red. It is possible to open the Connector window to check the status by clicking on the relative icon within the hidden icons in the taskbar. See PC-DMIS documentation for more details on how to

configure the connector application.

Once the connection is set up, from the project properties click on Import features button. The features available within the PC-DMIS project will be loaded inside HxGN Robotic Automation project tree under *Parts* | *Part* | *Features*.

11. 

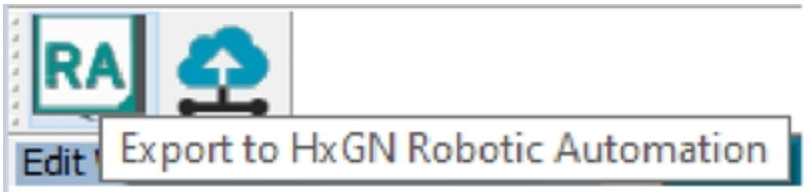




12. Once the program is ready to execute, from the HxGN Robotic Automation toolbar in PC-DMIS, click the TCP/IP Pointcloud Server receive data button ( ) to open the communication channel between PC-DMIS and HxGN Robotic Automation and send the COP from HxGN Robotic Automation to PC-DMIS.

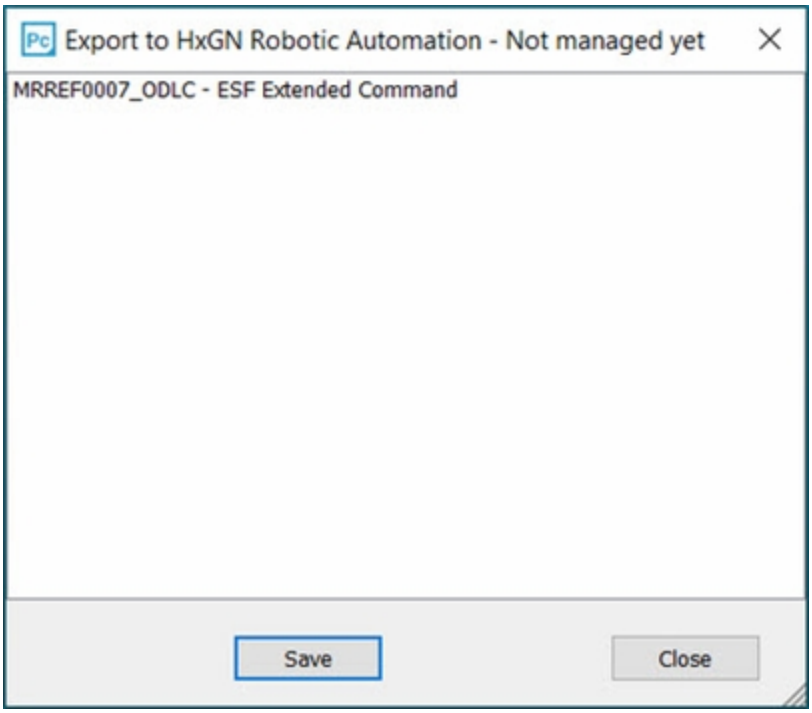
13. From HxGN Robotic Automation, create and execute the scanning program. PC-DMIS receives the scanned data from the HxGN Robotic Automation, and automatically executes the measurement routine. You can send scanned data to PC-DMIS to obtain measurement results, perform data analysis, and generate metrology reports.

Alternatively to steps 7) and 8) above it is possible to export features to HxGN Robotic Automation together with the CAD part from PC-DMIS application, following the next steps:

1. In PC-DMIS on the HxGN Robotic Automation toolbar, **click the Export to HxGN Robotic Automation** button or select the *File |Export | HxGN Robotic Automation* menu option.

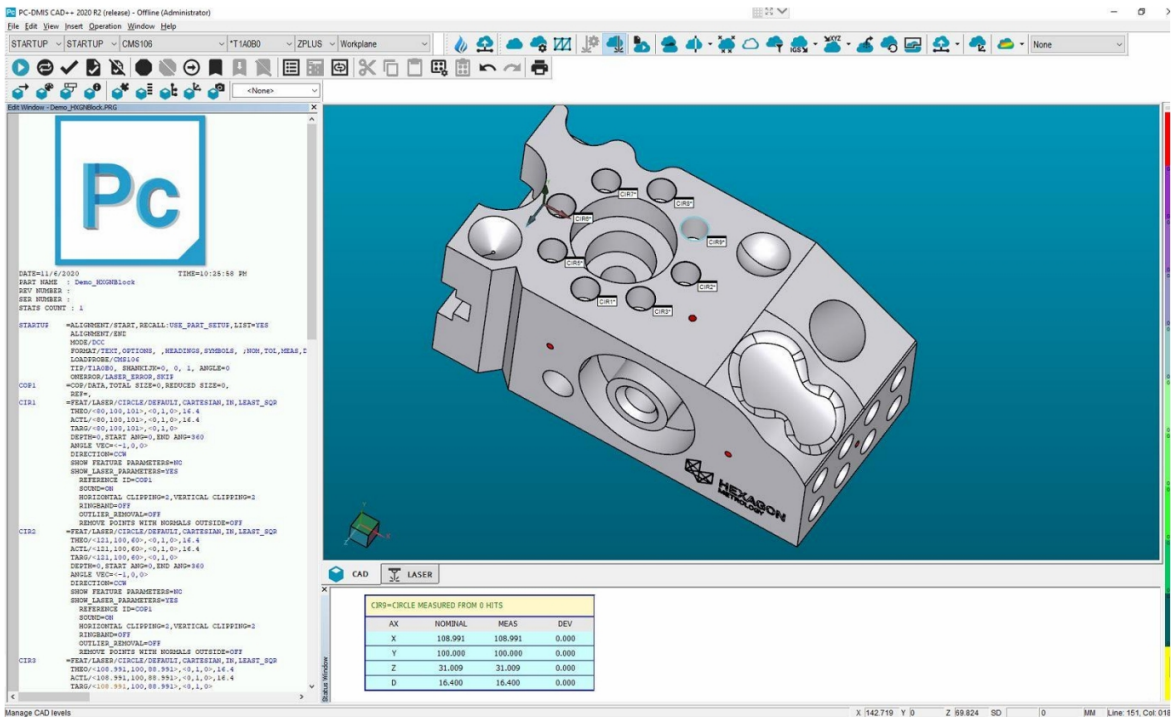


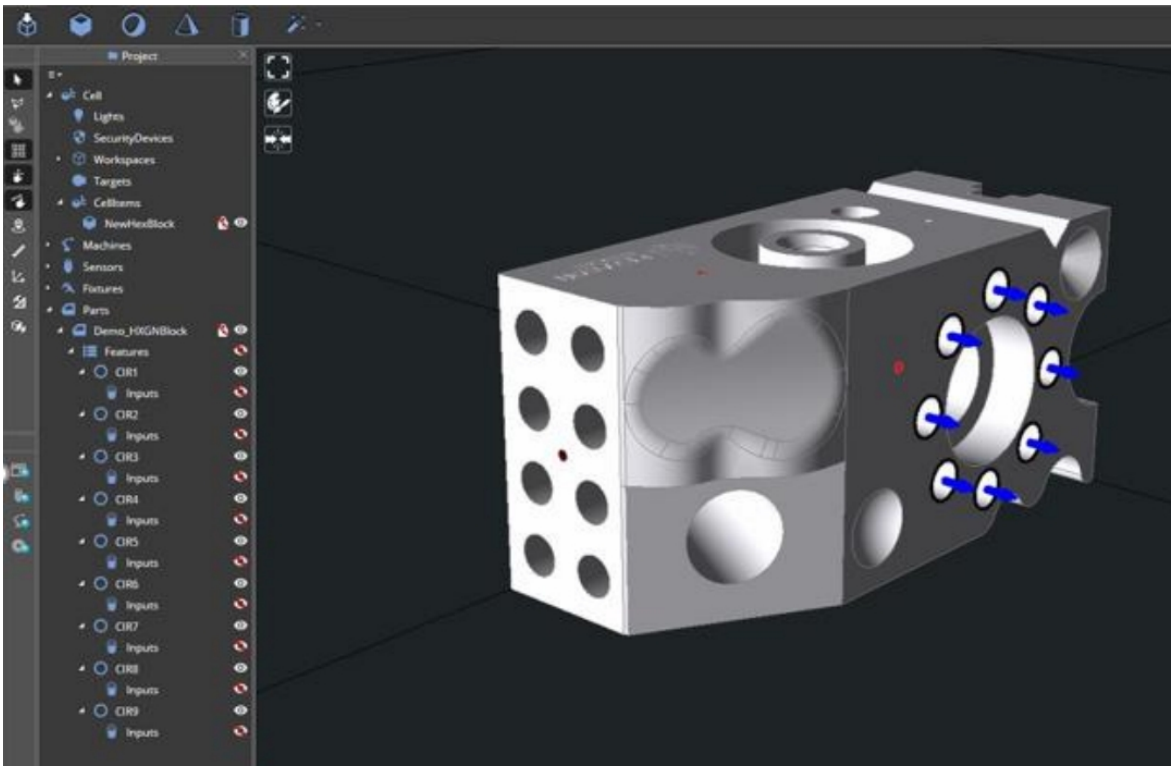

For features not yet supported, HxGN Robotic Automation displays the “Export to HxGN Robotic Automation - Not managed yet” dialog box to list them as shown in the following picture:



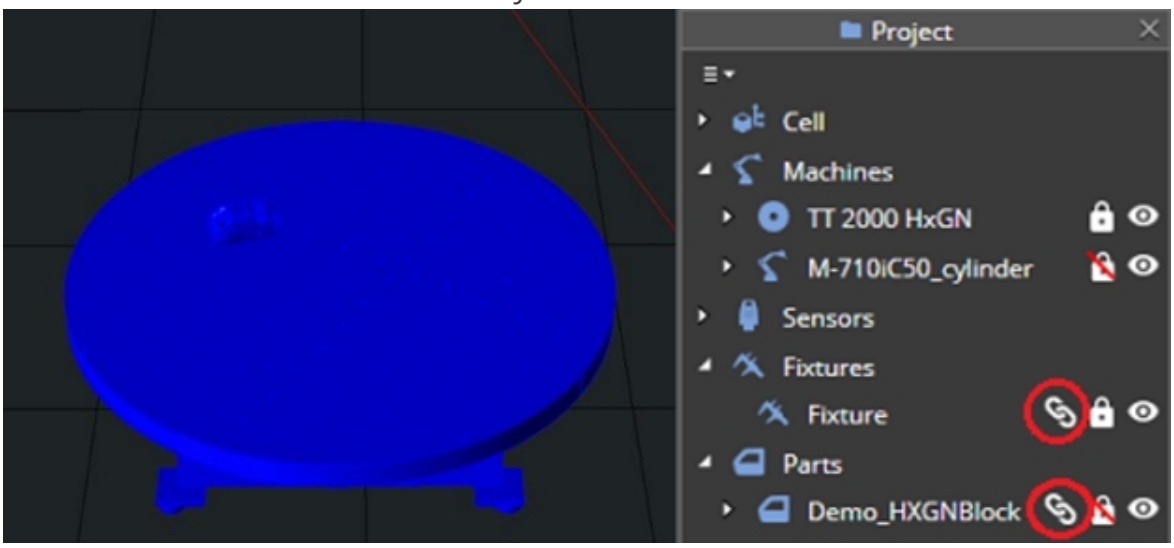

Click Save to save the report of unsupported features as a text file.

The part and the features will be sent to HxGN Robotic Automation application.


PC-DMIS CAD++ 2020 R2 (release) - Offline (Administrator)
CIR9=CIRCLE MEASURED FROM 0 HITS
AX NOMINAL MEAS DEV
X 108.991 108.991 0.000
Y 100.000 100.000 0.000
Z 31.009 31.009 0.000
D 16.400 16.400 0.000
CAD
LASER


Project
Cell
Lights
SecurityDevices
Workspaces
Targets
CellItems
NewHexBlock
Machines
Sensors
Fixtures
Parts
Demo_HXGNBlock
Features
CIR1
CIR2
CIR3
CIR4
CIR5
CIR6
CIR7
CIR8
CIR9
Inputs

If the part was exported from PC-DMIS, it must be aligned in the cell in HxGN Robotic Automation, e.g., if you are using a Rotary Table, assign the part to the fixture and the fixture to the Rotary Table.

2. 


## PC-DMIS Static point measurement

For alignment, Metrology applications use a RPS (Reference Point System) defined usually by specific alignment features (e.g., for a BIW by two circles and a slot) to align the part with respect to the RPS. However, for a BIW these alignment features are usually under the vehicle and not easy to scan.

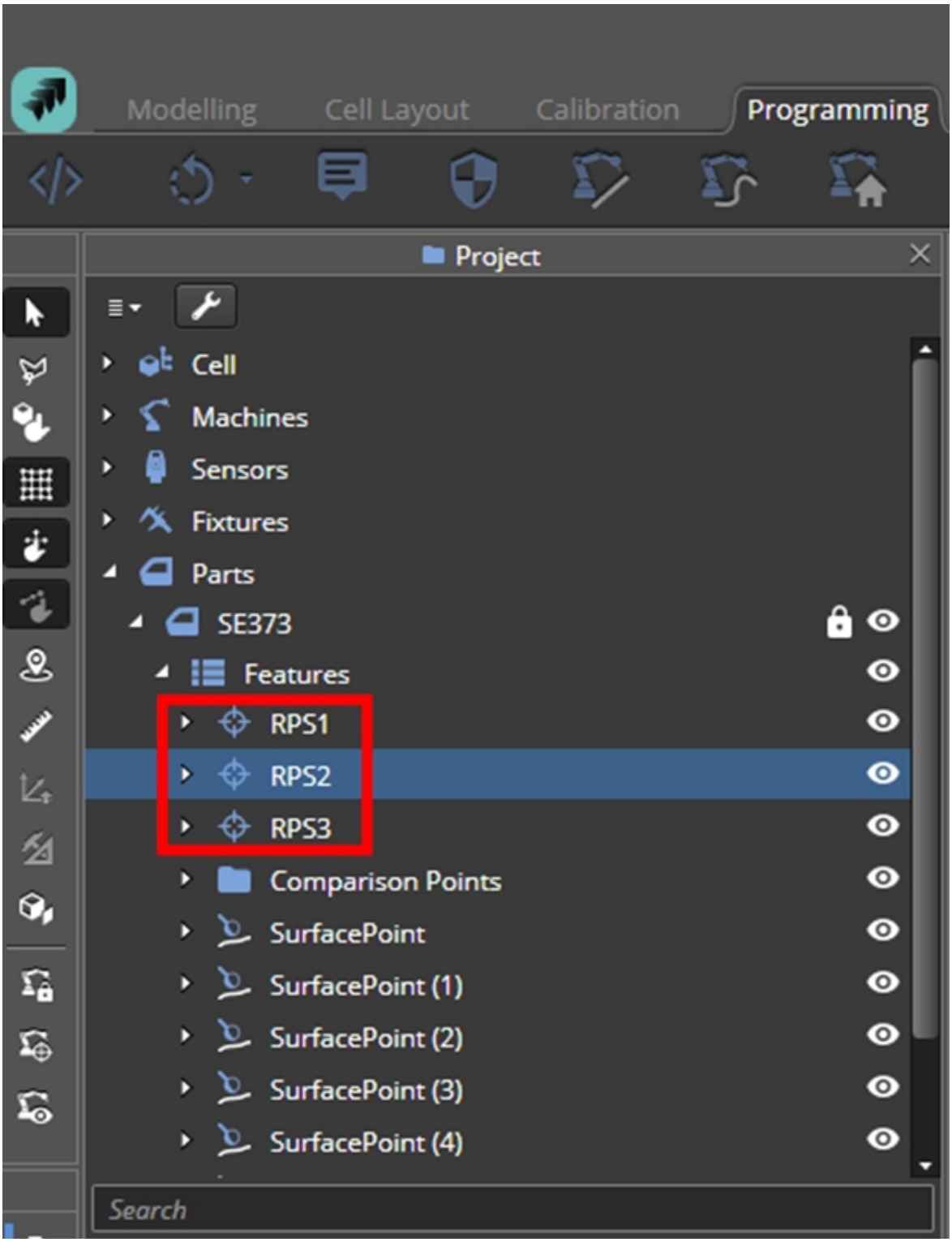
To solve this situation, in PC-DMIS metrology application the user can define some generic static point features, and these can be linked in HxGN Robotic Automation to reflectors.

The user can then create a program to measure these reflector positions and send the measurement to PC-DMIS that will be able to use them for alignment.

Below it is described the workflow to use static points for alignment using PC-DMIS metrology application.

1. Open the HxGN Robotic Automation application and connect to the PC-DMIS metrology software.
2. Using PC-DMIS define the static points as generic features that will be used for alignment and will be sent to HxGN Robotic Automation as specific features

defined by the icon.



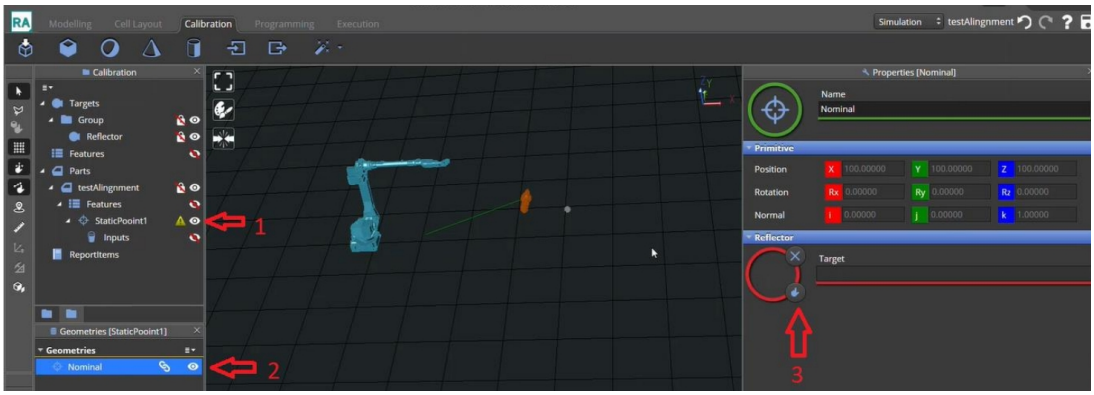

| | |
|---|---|
| Metrology Connection | 572 |

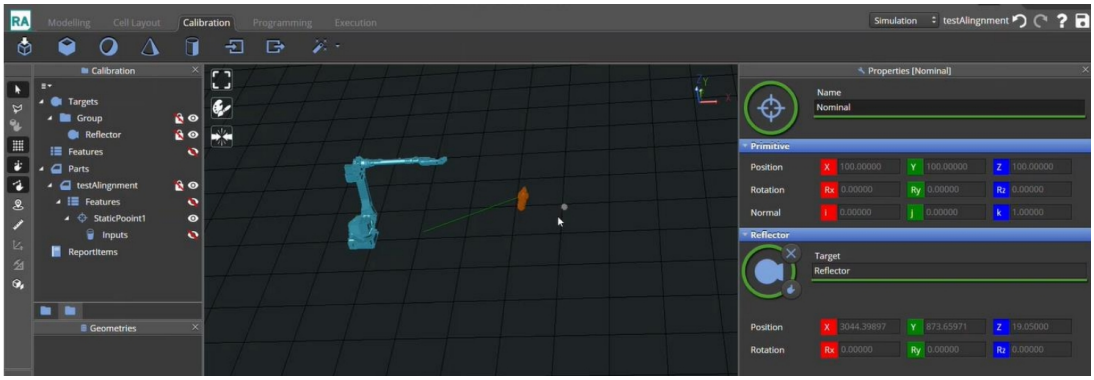
3. In HxGN Robotic Automation define a reflector using the Add Reflector Wizard

In HxGN Robotic Automation the newly added static points must be linked to a reflector. If a static point is not linked to a reflector, then a warning will appear beside the feature and validation will fail.

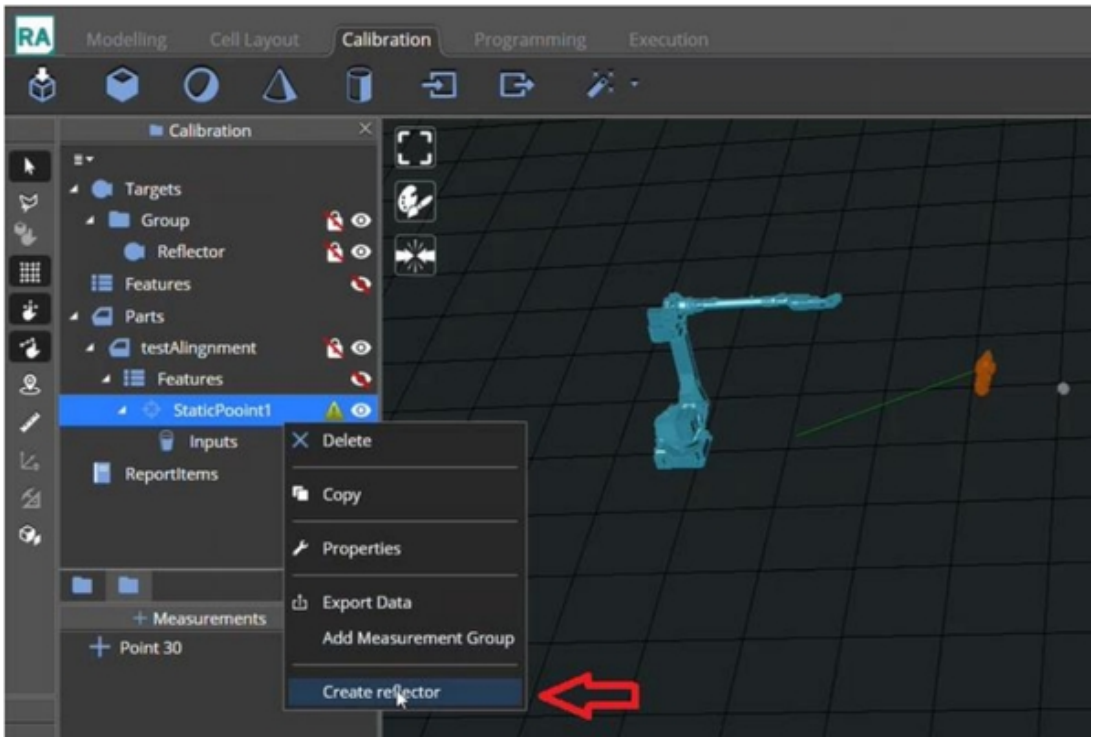
The static point feature is linked automatically to the reflector which is in the exact same coordinates as the feature.

4. If there is no reflector in the static point position, there are four options:
   a. Change the static point position through PC-DMIS to match the reflector position.
   b. Change the reflector position in HxGN Robotic Automation to match the static point position defined in PC-DMIS.
   c. Link the feature manually to a reflector by drag and drop in the project tree.

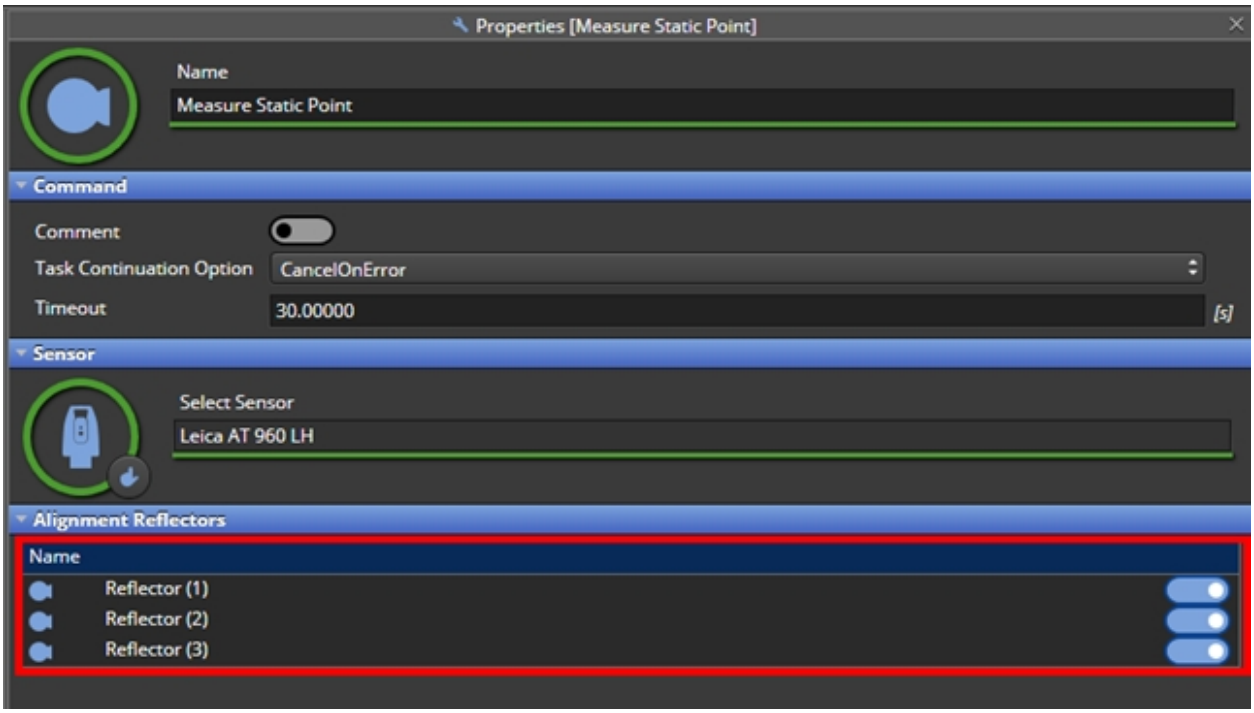

Modelling
Cell Layout
Calibration
Programming
Execution
Simulation
testAlingnment
Calibration
Targets
Group
Reflector
Features
Parts
testAlingnment
Features
StaticPooint1
Inputs
ReportItems
Geometries [StaticPooint1]
Geometries
Nominal
1
2
3
Properties [Nominal]
Name
Nominal
Primitive
Position
Rotation
Normal
Reflector
Target


Modelling
Cell Layout
Calibration
Programming
Execution
Simulation
testAlingnment
Calibration
Targets
Group
Reflector
Features
Parts
testAlingnment
Features
StaticPooint1
Inputs
ReportItems
Geometries
Properties [Nominal]
Name
Nominal
Primitive
Position
Rotation
Normal
Reflector
Target
Reflector
Position
Rotation

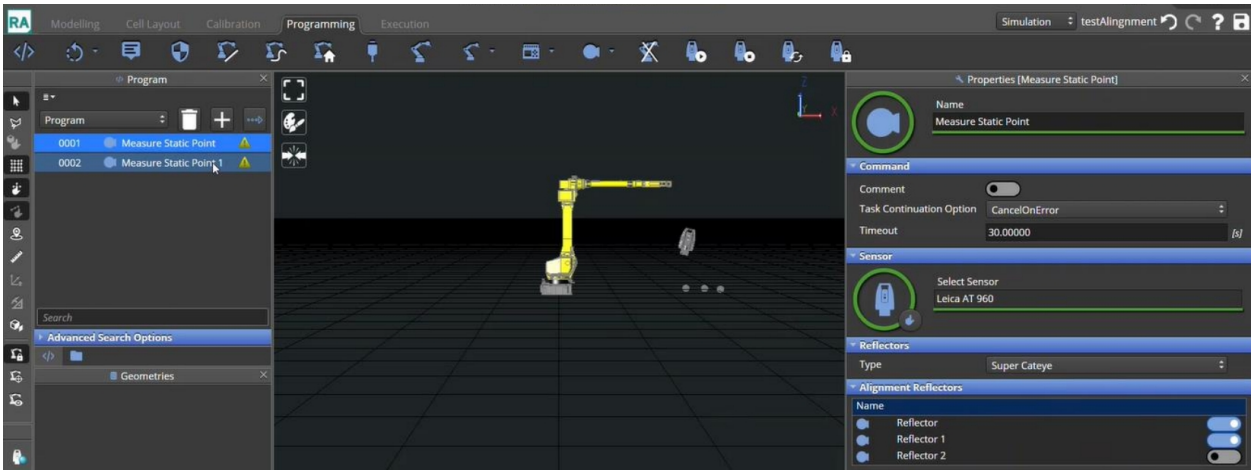
Create a reflector using the feature position by right clicking on it and select **Create reflector:**

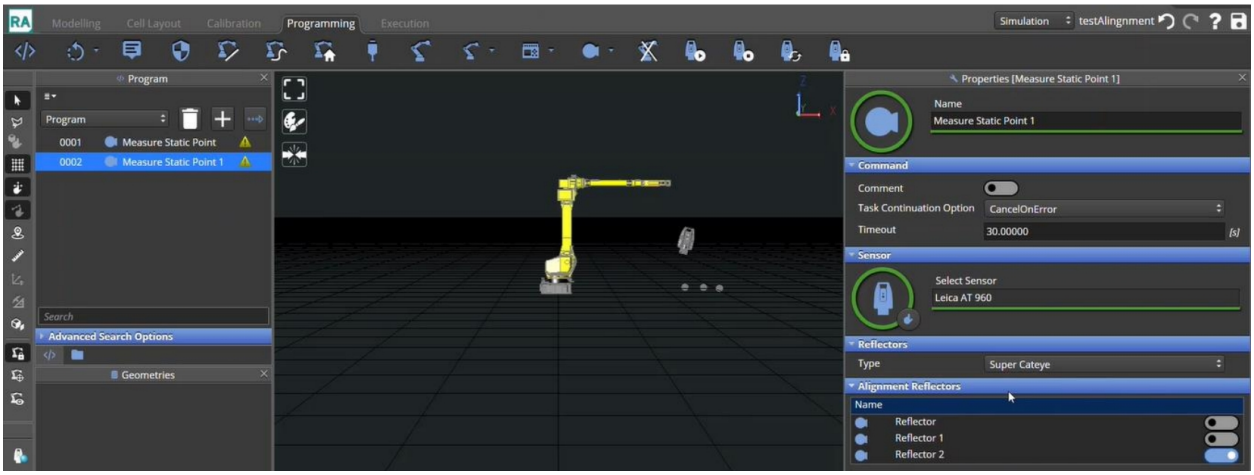
d. 


The **Measure PC-DMIS Static Points** command must be used to measure the static point positions, specifying in the properties which reflectors must be used. For the sake of accuracy, it is preferable to measure all the reflectors used for alignment with a single command.




During the validation of a program having **Measure PC-DMIS Static Points** commands, HxGN Robotic Automation will check if the command covers all the reflectors used for alignment, e.g., three of these reflectors can be covered with one command that measures all of them, or by two commands: the first one measuring two reflectors and the second one measuring the remaining reflector.






If less than three reflectors are measured validation will show a warning.

If more than three reflectors are measured (this will happen if the same reflector is measured multiple times) validation will also show a warning.

In case there are no static points the command **Measure PC-DMIS Static** Points cannot be validated.

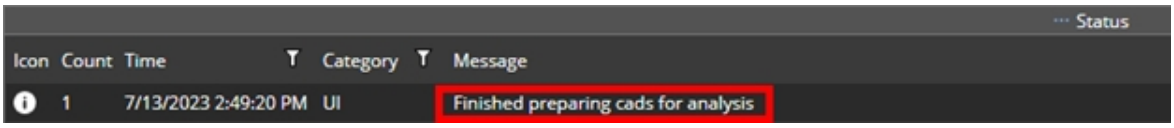
In case the state of the static points is changed (removed/added), validation will reset, and the program will need to be validated again.

When executing the program, the measured reflectors are sent to PC-DMIS for alignment calculation. In case of an issue, PC-DMIS can stop the program execution.

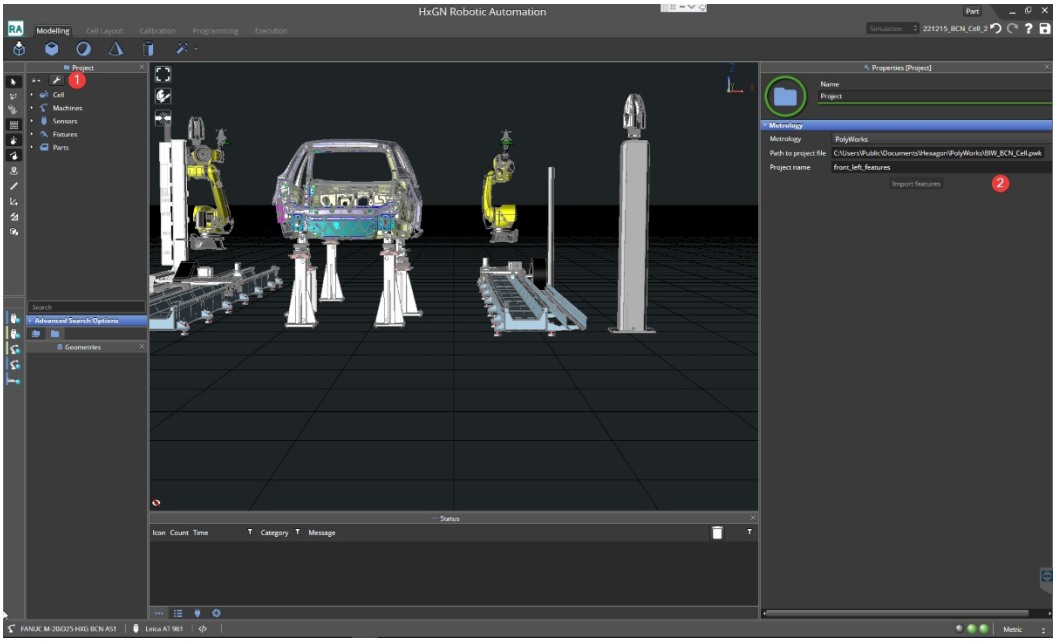
## PolyWorks

To set up the connection with PolyWorks and start using it together with HxGN Robotic Automation, follow the next steps:

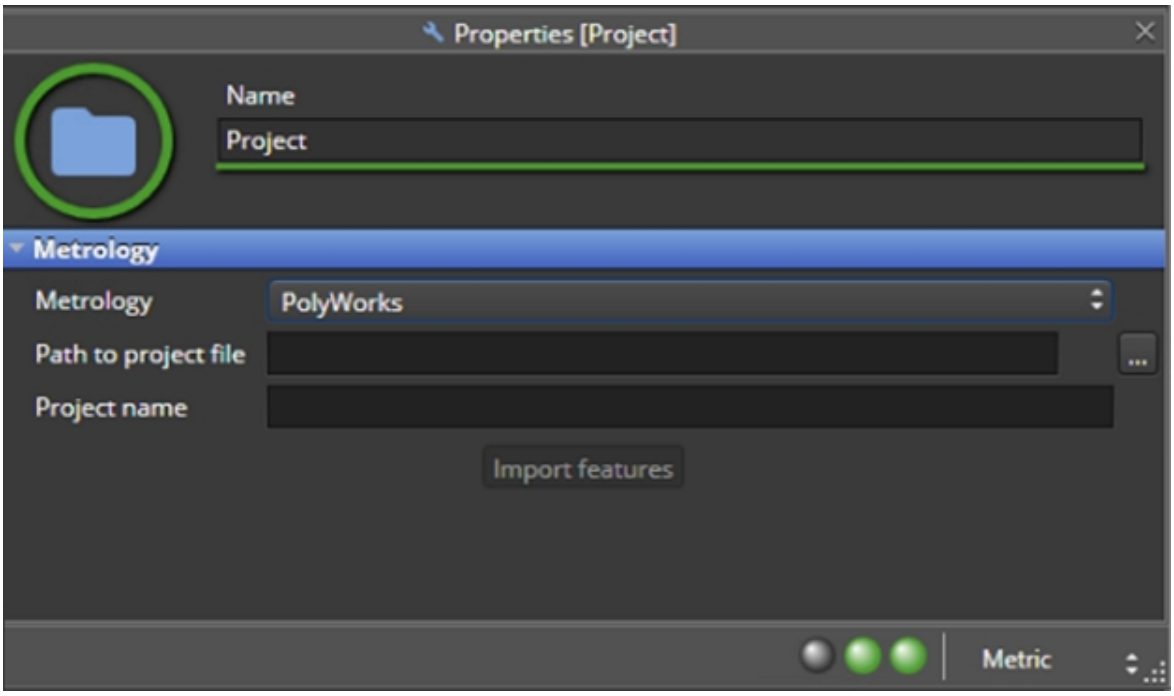
1. Create a new project in HxGN Robotic Automation, import the CAD manually into HxGN Robotic Automation (automatic CAD **import from PolyWorks** is not supported), and wait until the following message is displayed within the Status window.



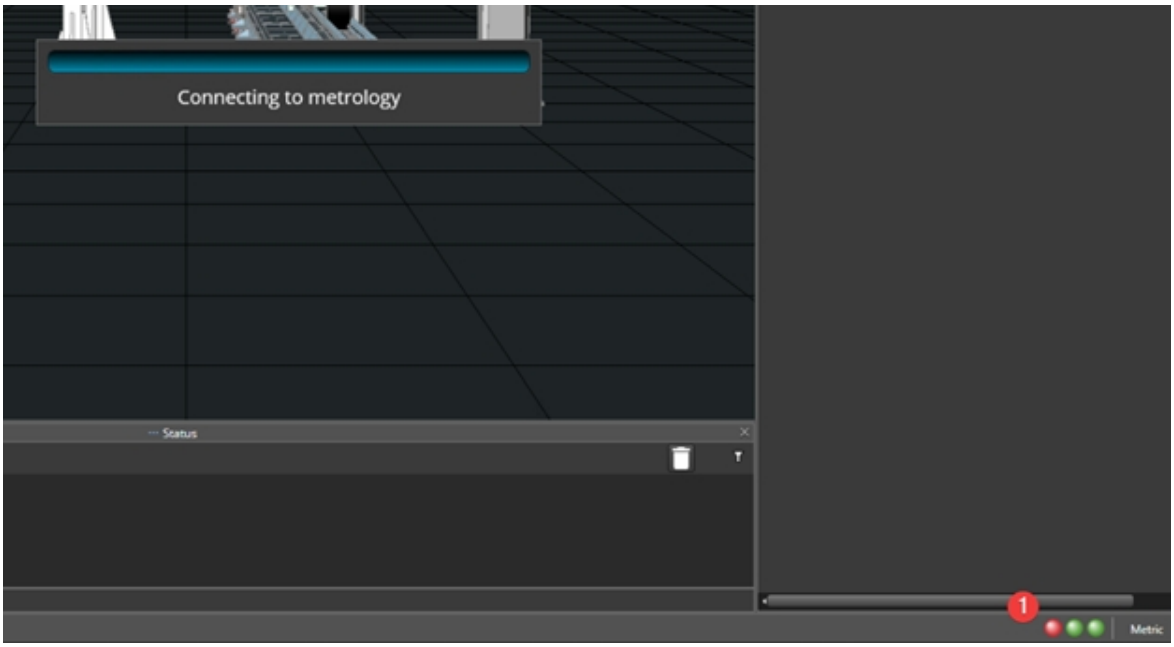

2. Under the project tree the user must click on the Project Properties icon to open the properties window on the right and select PolyWorks from the Metrology drop-down menu.
3. Set the **Path to project file** and PolyWorks Project name to be used.




Pay attention that PolyWorks structure has a Project file, and under it there is a Project name. Importing features capability is disabled when connection to metrology is disabled (Metrology Activation button is grey).

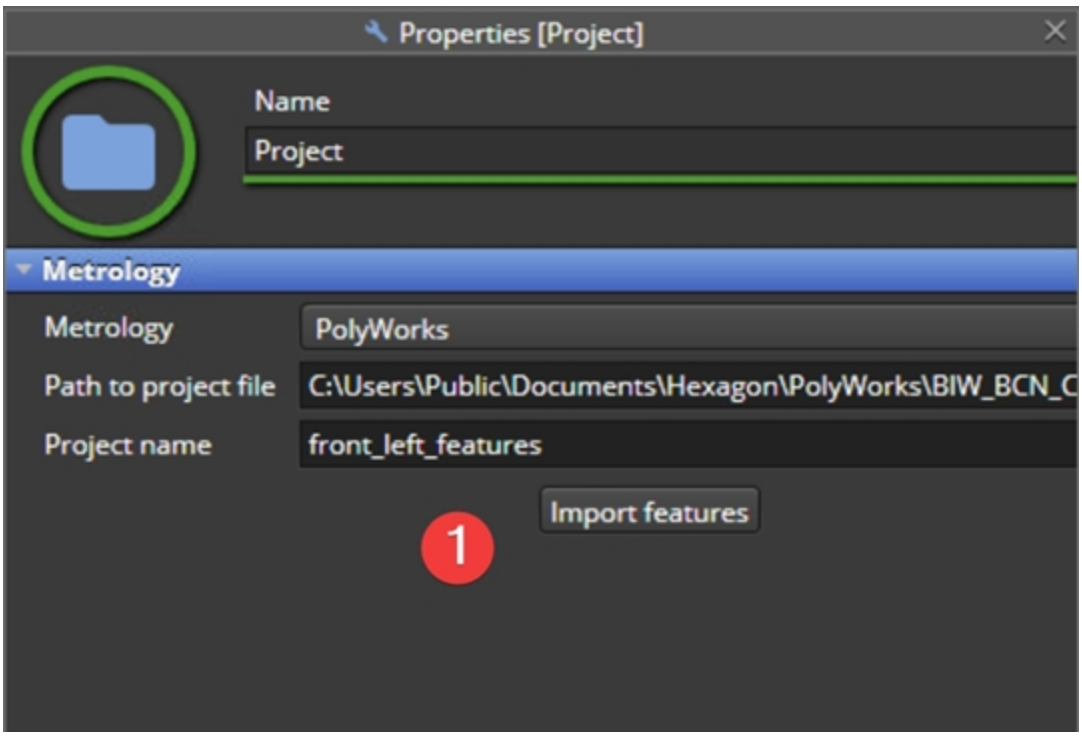



4. Connect to PolyWorks: the Metrology button will turn to green once the connection is set up.



5. Now the user can import the features in HxGN Robotic Automation, and it will be possible to start programming the robot path to scan them. A list of supported features can be found in chapter 0.

> ⚠ Ensure to use the English language in the PolyWorks project while creating the features, otherwise the import in HxGN Robotic Automation may fail.




6. Once the program has been written in HxGN Robotic Automation, the user can execute the program either on HxGN Robotic Automation or Execution Manager (see chapter Execution Manager).
    a. Executing in HxGN Robotic Automation:
        i. If the PolyWorks Inspector program is already opened, HxGN Robotic Automation will send the COP to this one.

ii. If the PolyWorks Inspector program is not opened, HxGN Robotic Automation will open PolyWorks Inspector program and send the COP to that instance.

When importing a project in HxGN Robotic Automation, all the Macro files that are used will need to be separately loaded from the preferred folder into the HxGN Robotic Automation project.

b. Executing in Execution Manager:

i. In case Generic Host is running as a console application, PolyWorks Inspector behaves as in HxGN Robotic Automation.

ii. In case Generic Host is running as a service, PolyWorks Inspector is hidden. Everything works in the background as a process without the visual application running.

### Programming PolyWorks Macros

The user can add an **Execute PolyWorks Macro** command ( ) to the program to run specific execution routines on PolyWorks. Once added to the program the user can specify the properties.

HxGN Robotic Automation offers pre-made macros (like Set Device Position to be used with reference points or custom macros, in which the user can add commands as needed.

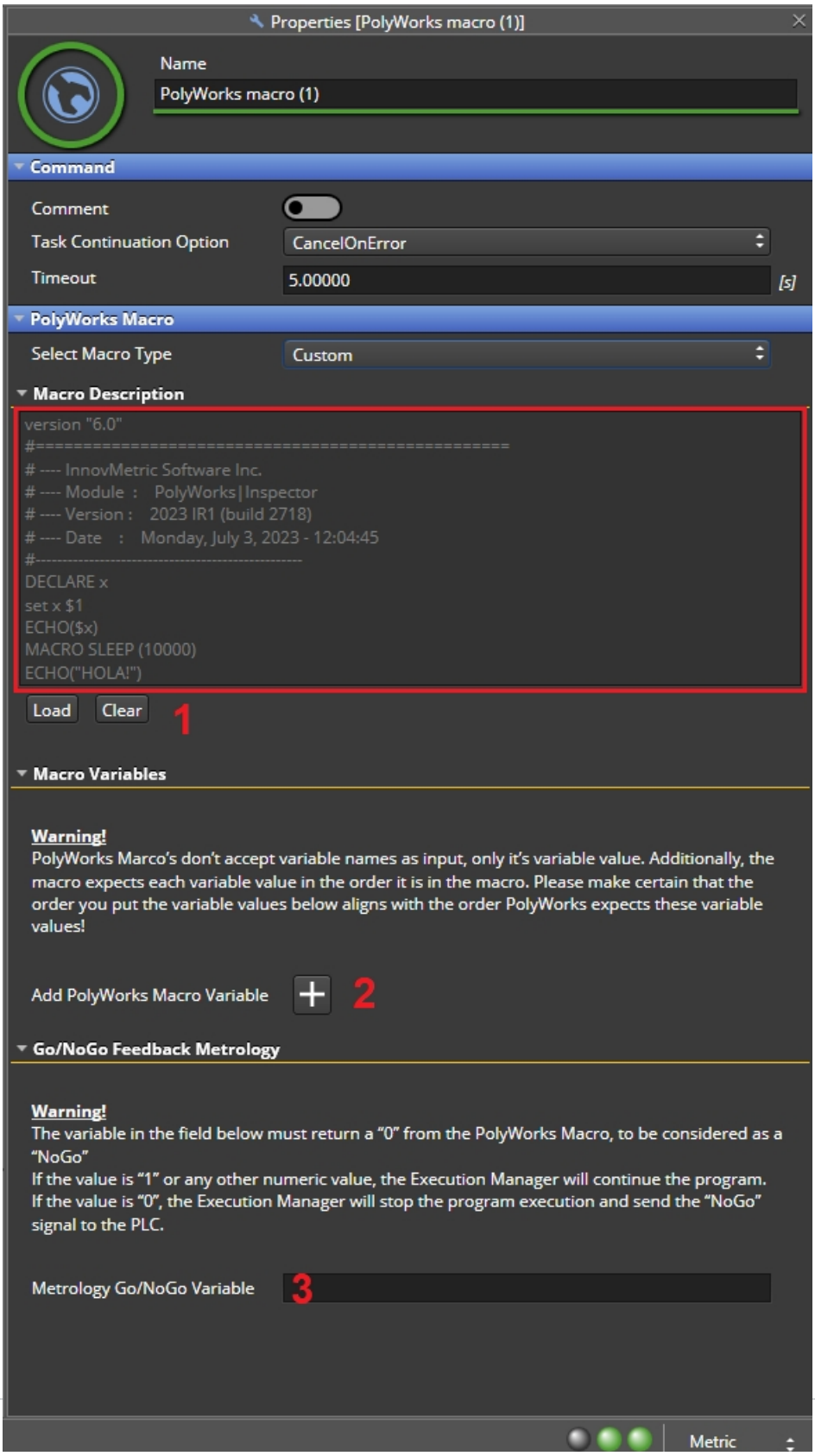
After selecting **Custom** within the Select Macro Type drop-down menu, the following properties will be shown:

1. Under *PolyWorksMacro* | *Macro Description* the user must load the PW Macro from the file system.

   The macro will appear in the box after successfully loading.

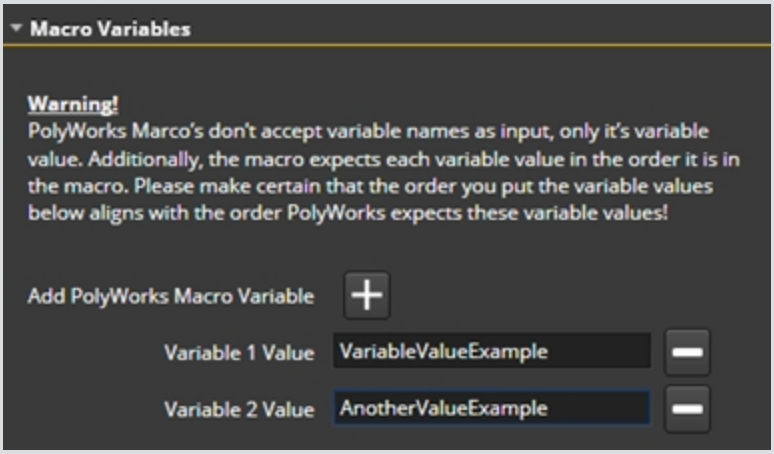
2. Under *PolyWorksMacro* | *Macro Variables* the user can set a specific variable value for the Macro. PolyWorks expects to receive these values according to the order they are listed: the first value will be evaluated for the first variable that is expecting a value, the second value will be evaluated for the second variable that is expecting a value, and so on.
3. Under *PolyWorksMacro* | *Go/NoGo* Feedback Metrology the user can set a Go/NoGo variable. Execution Manager will check the value of this variable as feedback from the PolyWorks Macro Command. Refer to chapter for further

information on the usage.


Properties [PolyWorks macro (1)]
Name
PolyWorks macro (1)
Command
Comment
Task Continuation Option
CancelOnError
Timeout
5.00000
[s]
PolyWorks Macro
Select Macro Type
Custom
Macro Description
version "6.0"
#==========
# ---- InnovMetric Software Inc.
# ---- Module : PolyWorks|Inspector
# ---- Version : 2023 IR1 (build 2718)
# ---- Date : Monday, July 3, 2023 - 12:04:45
#----------
DECLARE x
set x $1
ECHO($x)
MACRO SLEEP (10000)
ECHO("HOLA!")
Load
Clear
1
Macro Variables
Warning!
PolyWorks Marco's don't accept variable names as input, only it's variable value. Additionally, the macro expects each variable value in the order it is in the macro. Please make certain that the order you put the variable values below aligns with the order PolyWorks expects these variable values!
Add PolyWorks Macro Variable
2
Go/NoGo Feedback Metrology
Warning!
The variable in the field below must return a "0" from the PolyWorks Macro, to be considered as a "NoGo"
If the value is "1" or any other numeric value, the Execution Manager will continue the program.
If the value is "0", the Execution Manager will stop the program execution and send the "NoGo" signal to the PLC.
Metrology Go/NoGo Variable
3
Metric

Example: Using variable values

Two variable values have been added: *VariableValueExample* and *AnotheValueExample*.




In PolyWorks to call those values you would need to use $1 for the first and $2 for the second value.

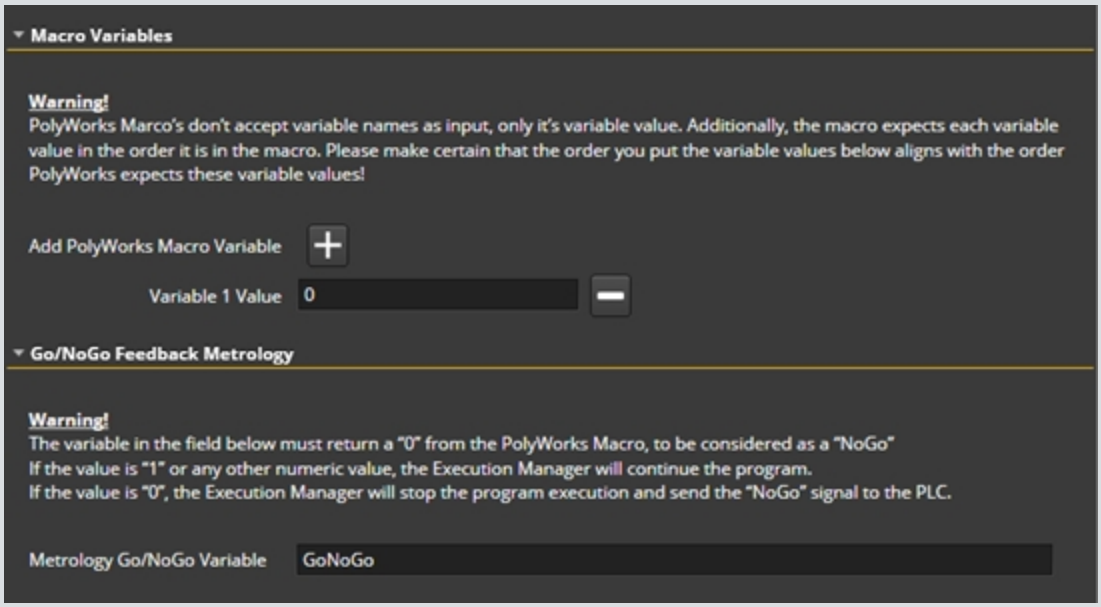
```
1  MACRO ECHO ( $1 )
2  MACRO ECHO ( $2 )
```

This will result in PolyWorks as:

```
# ----------------------------------
# Running macro script: test.pwmacro
VariableValueExample
AnotherValueExample
# End of macro script: test.pwmacro
# ----------------------------------
```

Example: Using Go/NoGo Feedback

Create a macro that will feedback failure (hardcoded in example).




In PolyWorks have the following Macro:

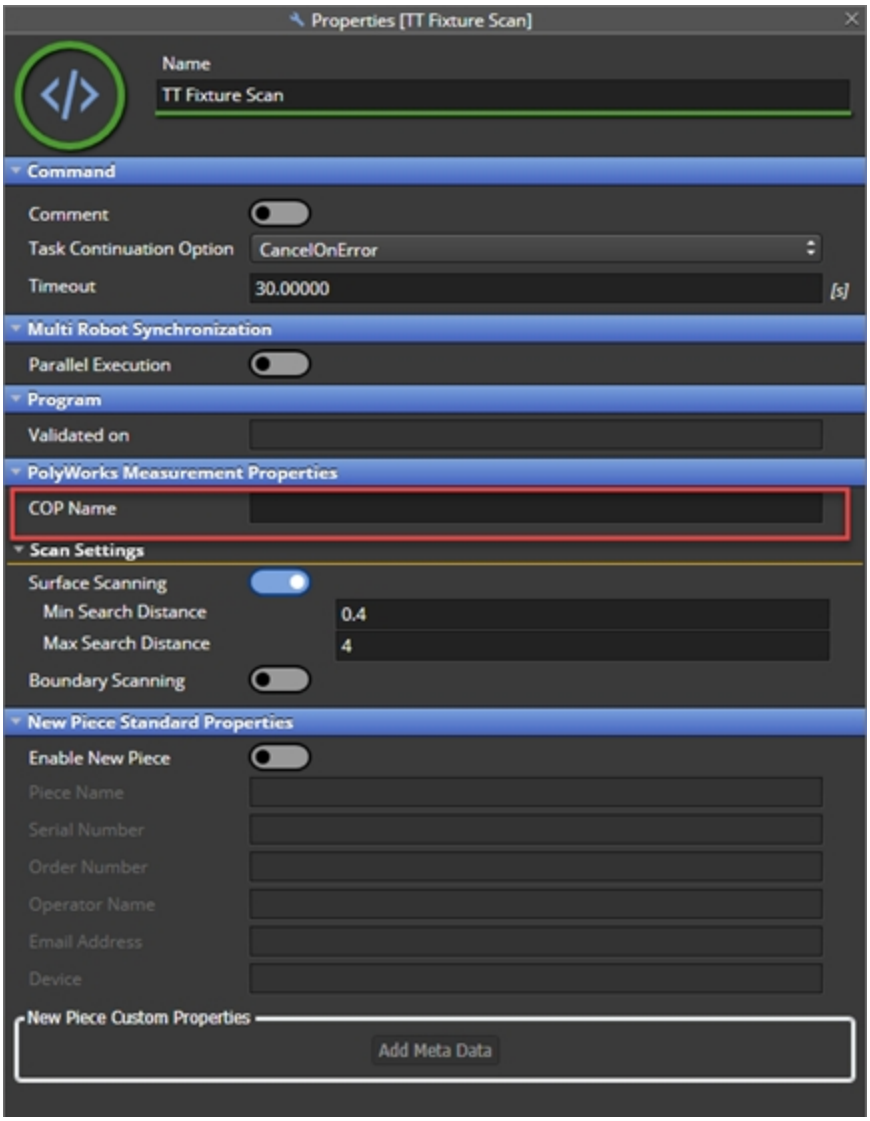
```
1  DECLARE GoNoGo
2
3  SET GoNoGo $1
```

In a real Go/NoGo scenario the value will be decided by PolyWorks based on the feature calculation and not as in this example based on the Variable 1 Value.

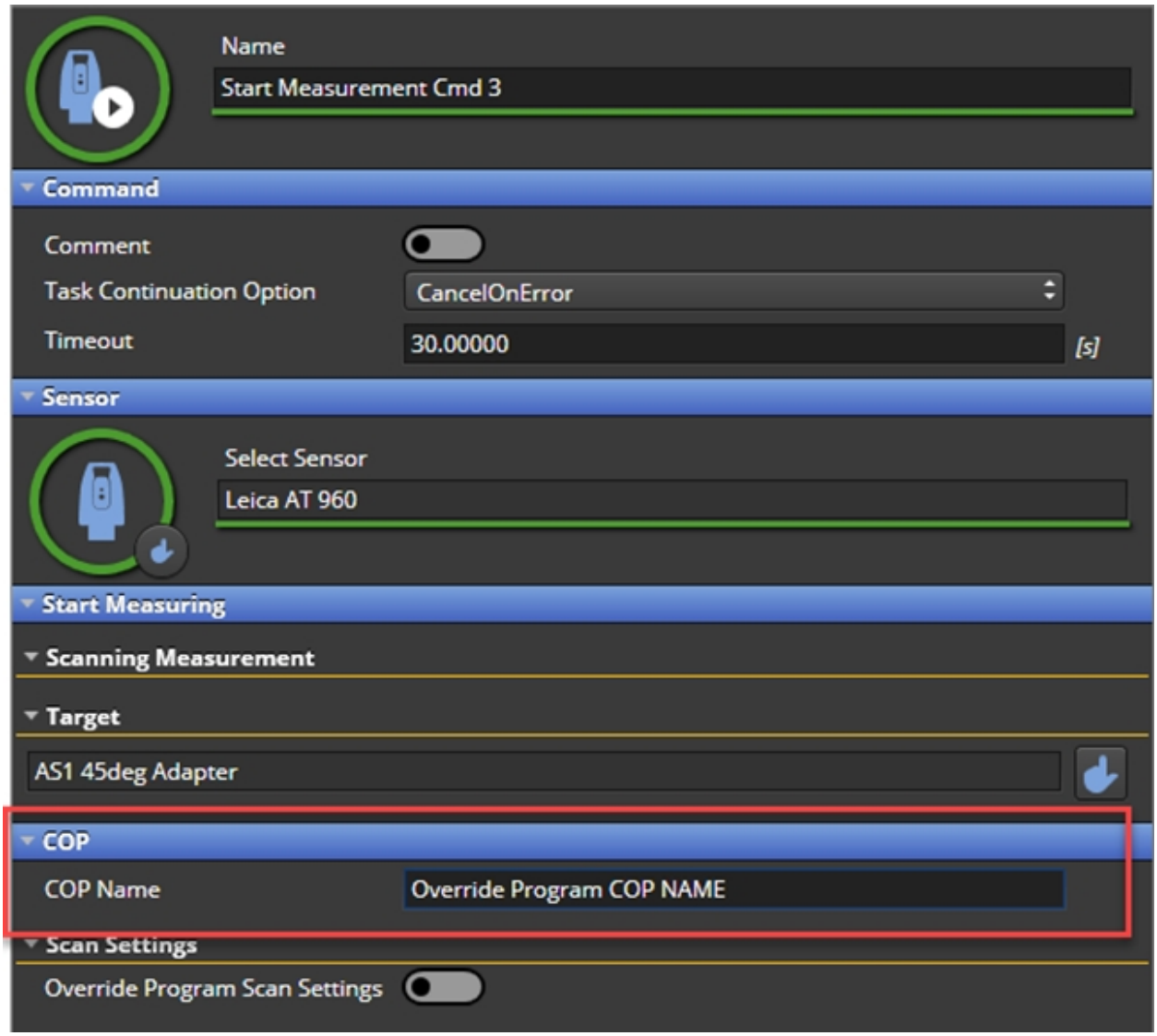
## Named COP

It is possible to define a global COP name to be used throughout the entire program execution or to define local COPs to be sent as separate data sets at every scan.

In the first case, the global COP name can be specified in the Program properties under COP | COP Name. If no name is specified, the COP will be named “Ra Cop.




In the second case, the local COP name can be specified in the Start Measurement command properties also under *COP* | *COP Name*.




If a Program Properties COP has been defined, this will be the name of the COP for the whole program, unless a different name is specified within another Start Measurement command property.

| | |
|---|---|
| | In case of multiple iterations of the same program for the same piece, please avoid having more than one space character in the COP name. This will cause the COP on next iterations of the program to create a new COP container in |

| | |
|---|---|
| | PolyWorks, rather than append to the existing one. E.g., if COP Name is: “Test 1 Container” (having two spaces) on the second run it will create a new container “Test 1 Container (1)” rather than append to the original one. |

The COP Name specified in the Start Measurement command properties will over-write the one defined under Program Properties. If nothing is specified as COP Name neither in the first nor in the second scenario, the resulting COP will be called “Ra Cop.”

Example: Program COP: Name_Example1

- Start measurement1 left blank
- Start measurement2: Named: Name_Example2
- Start measurement3 left blank
- Start measurement4: Named: Name_Example3
- Start measurement5: Named: Name Example2

Will results in:

- COP: Name_Example1 consists out of
  - Start measurement1
  - Start measurement3
- COP: Name_Example2 consists out of:
  - Start measurement2
  - Start measurement5
- COP: Name_Example3 consists out of:

| | |
|---|---|
| | • Start measurement4 |

## Scan Settings

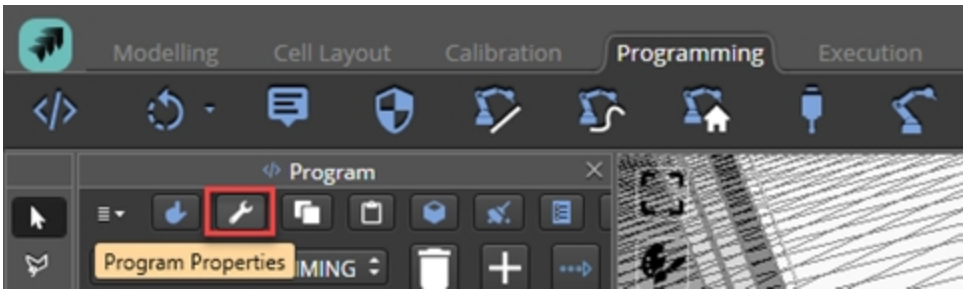
By default, a program created under a PolyWorks enabled project has the Surface Scan enabled with the following default parameters:

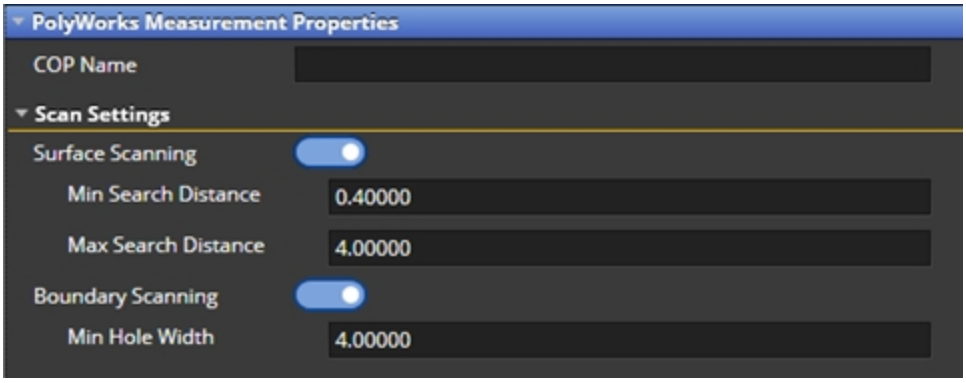
- Min. Search Distance: 0.40 mm
- Max. Search Distance: 4.0 mm

The boundary scan when enabled will have the following default parameter:

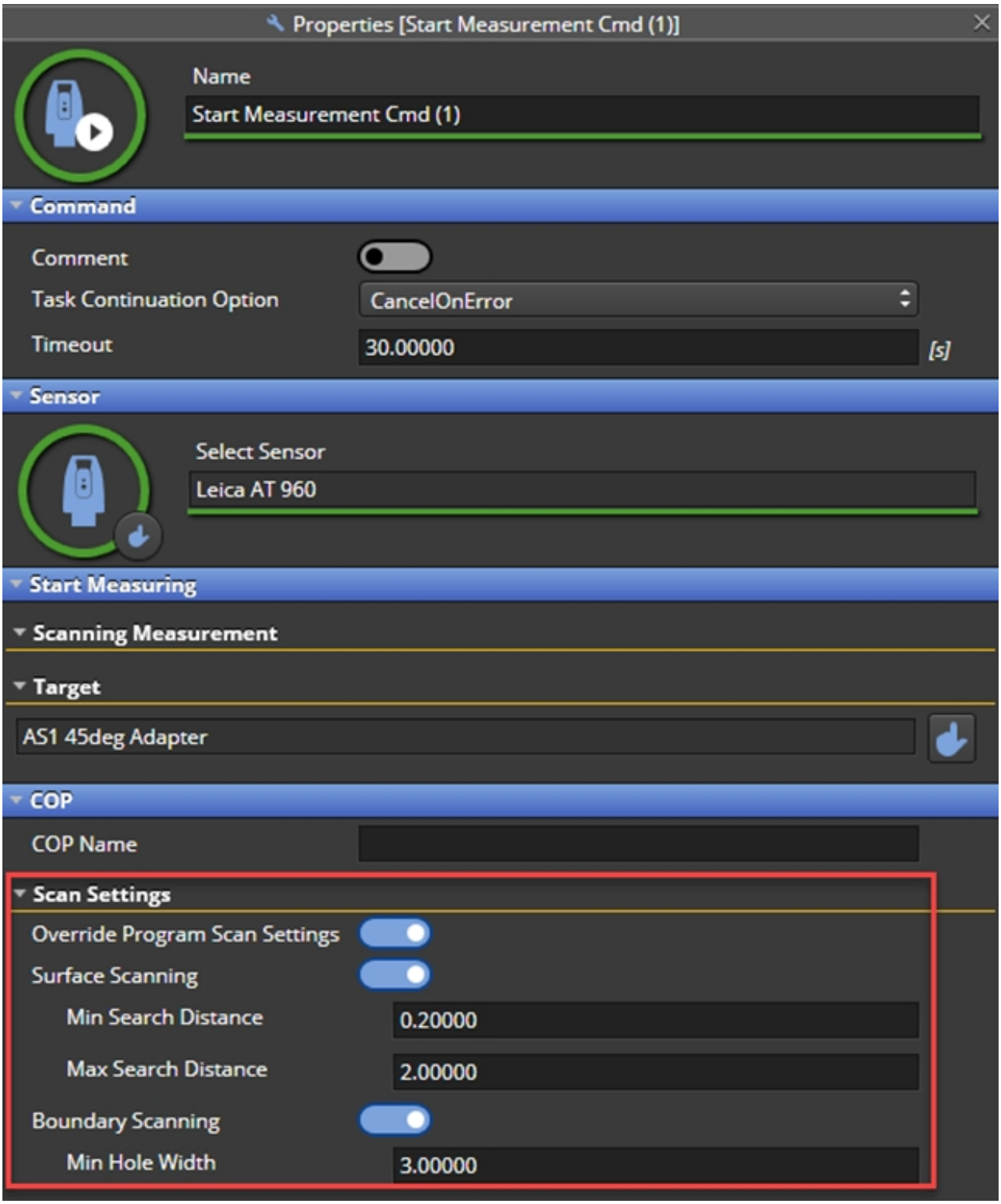
- Min. Hole Width: 4.0 mm

The settings can be changed for the whole program by going to the **Program Properties** of a PolyWorks enabled project.



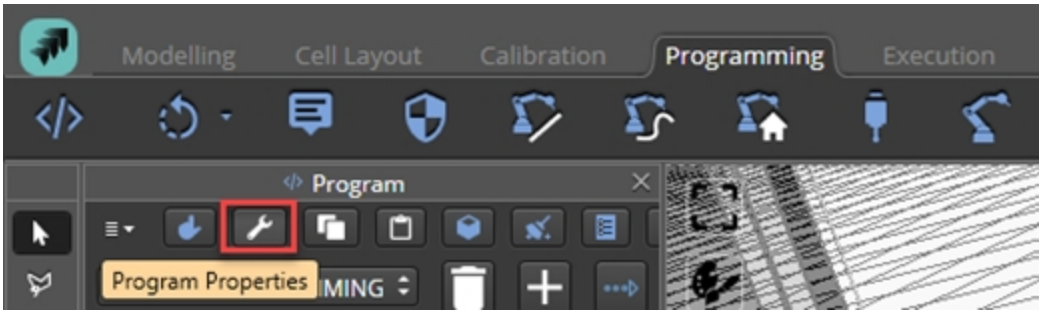



Each individual start measurement command can overwrite the Scan Settings between that start measurement command and the next stop measurement command.

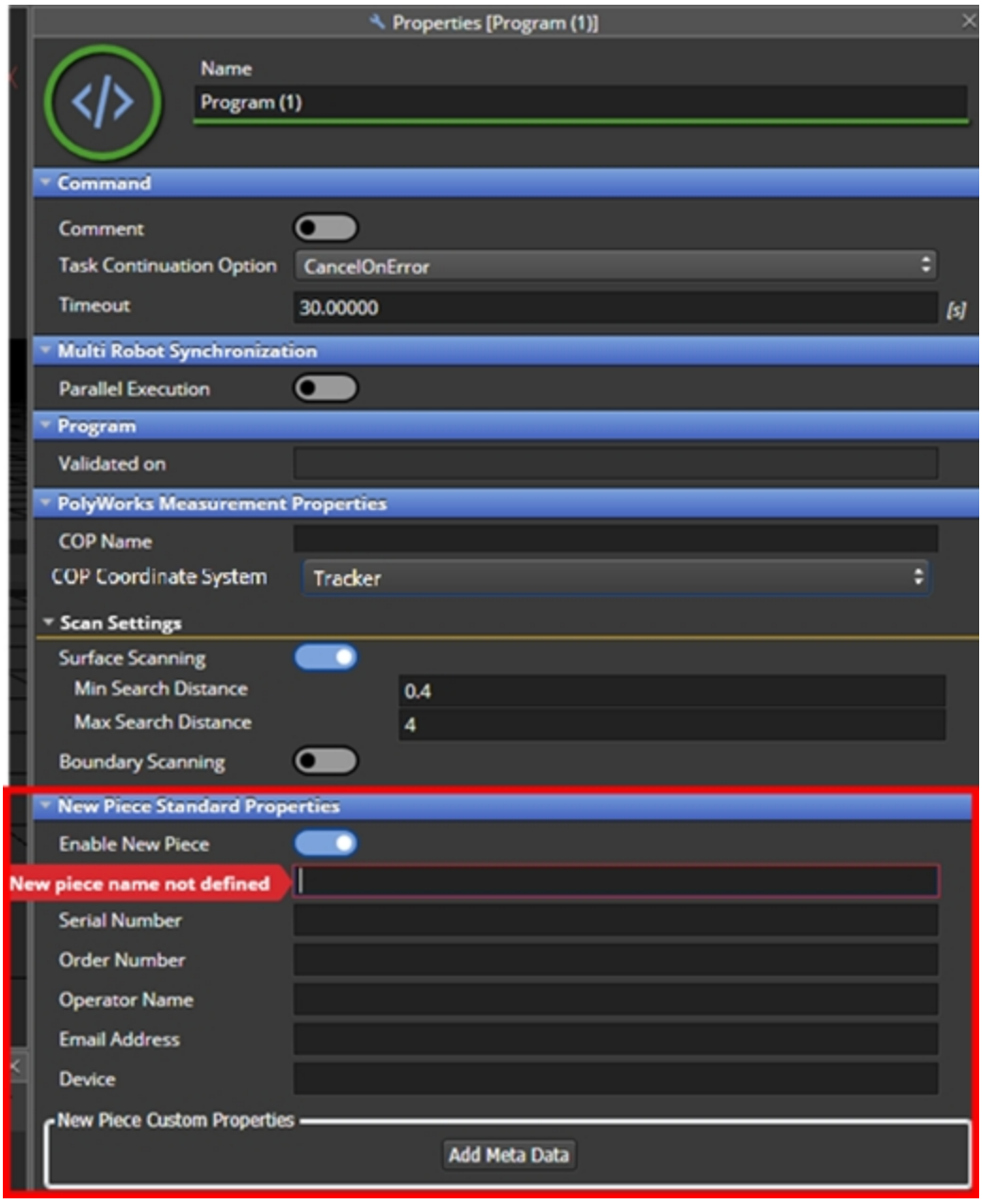

Properties [Start Measurement Cmd (1)]
Name
Start Measurement Cmd (1)
Command
Comment
Task Continuation Option
CancelOnError
Timeout
30.00000
[s]
Sensor
Select Sensor
Leica AT 960
Start Measuring
Scanning Measurement
Target
AS1 45deg Adapter
COP
COP Name
Scan Settings
Override Program Scan Settings
Surface Scanning
Min Search Distance
0.20000
Max Search Distance
2.00000
Boundary Scanning
Min Hole Width
3.00000

## Setting metadata

To enable the new piece creation in PolyWorks, go to the program properties and select "**Enable New Piece**."




The new piece is not enabled by default to allow the user to run various programs for only one piece in PolyWorks. Once enabled, the Piece Name must be specified.oints



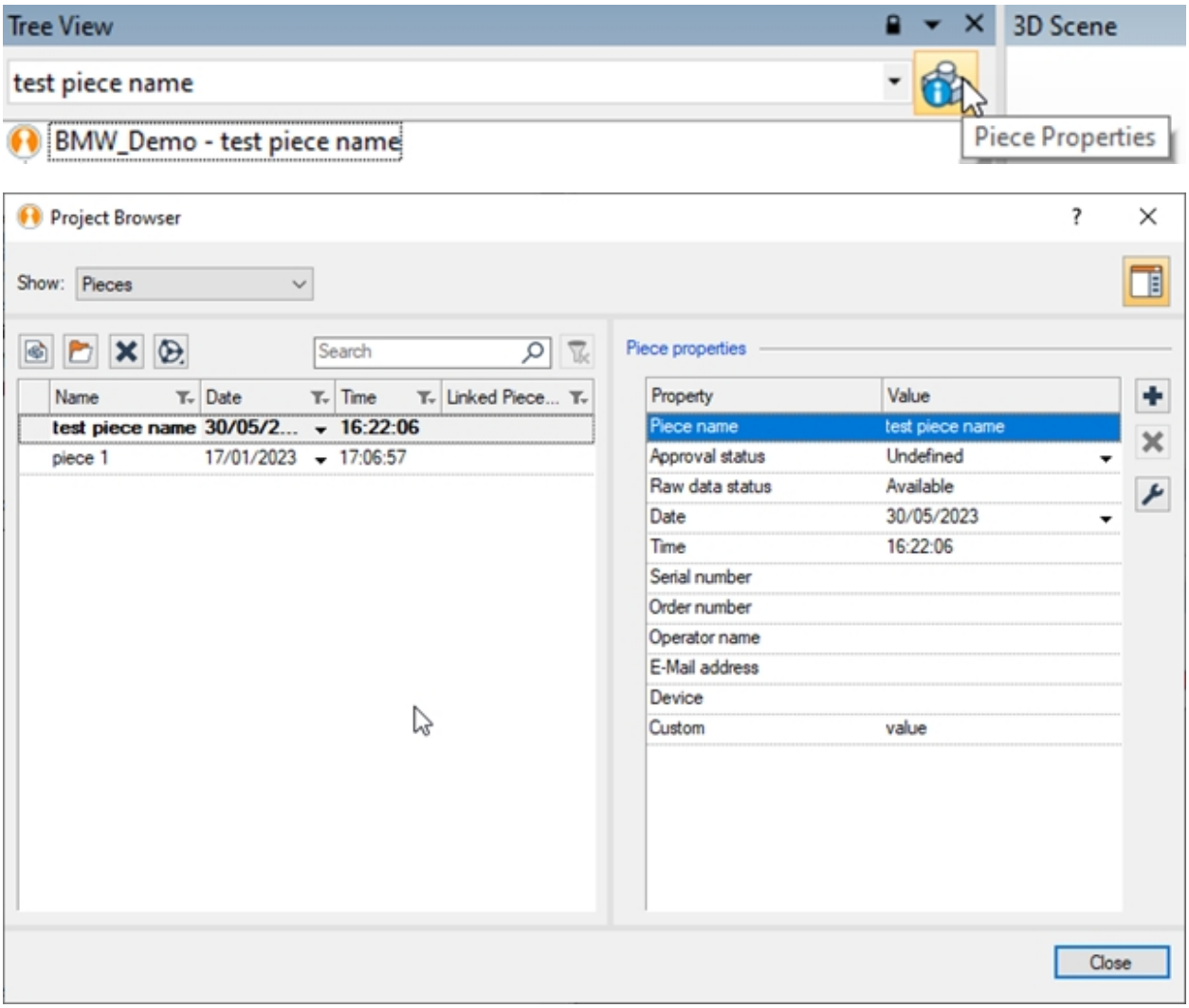

The new piece is not enabled by default to allow the user to run various programs for only one piece in PolyWorks. Once enabled, the Piece Name must be specified.

HxGN Robotic Automation will already provide some fields that are default piece properties in PolyWorks, while also offering a possibility to add custom Meta Data.

---

When creating a new piece, scan data is saved automatically within PolyWorks. In case the program is not configured to create a new piece, this data will have to be saved manually or through a custom macro.

In PolyWorks, metadata can be visualized by clicking on the Piece Properties.



## Reference points

When executing a program with reference points and than a program with nominals and actuals that point to the same PolyWorks project, it is mandatory to reset the device position to the default, to prevent the point cloud not being aligned correctly in the nominals and actuals program.

Reference points are used in PolyWorks to compensate the cloud of point as it is done using internal HxGN Robotic Automation compensation.

The functionality is similar to PC-DMIS Static points, used for part alignment work-flow with PC-DMIS (see static points). Reference points workflow with PolyWorks consists of HxGN Robotic Automation sending reflector positions to PolyWorks so it can either use them to align the various COPs received during the scans or to check part alignment.

When using reference points, COP will appear in a wrong place in RA because data compensation is managed in PolyWorks side.

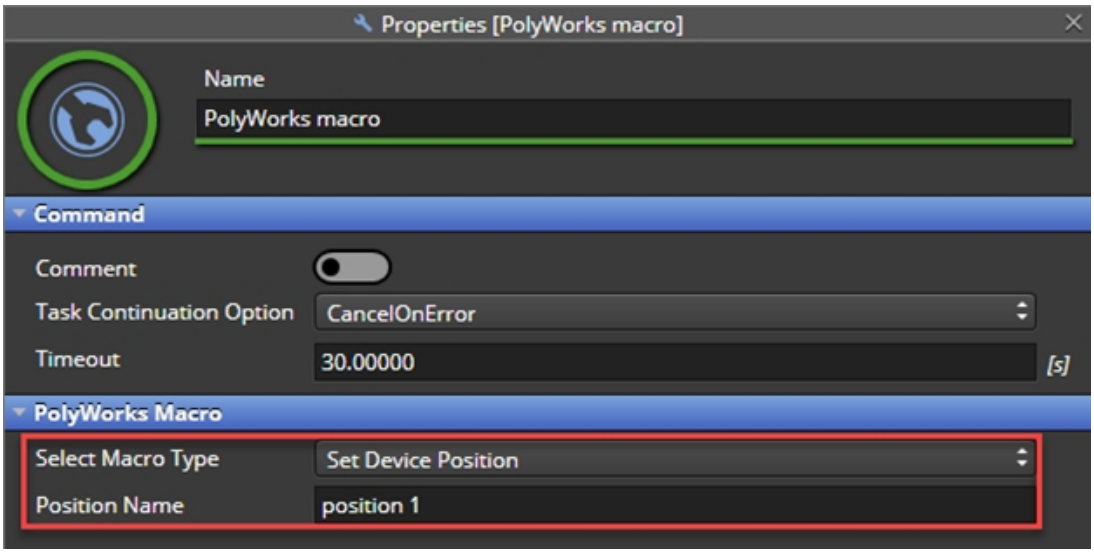
To set the reference points, it is first necessary to:

1. Create device positions (if they do not already exist) in the PolyWorks project to be used either using Macro programming directly in PolyWorks or using an Execute Macro command within the HxGN Robotic Automation program.
2. Set the active device position by using the predefined Set Device Position macro available in HxGN Robotic Automation. This predefined macro allows the software to automatically ensure the COP name is not re-used in various positions:

a. add an **Execute PolyWorks Macro** command

b. select **Set Device Position** as Macro Type from the relative drop-down menu

c. set the device position name (case sensitive) as defined in the relative PolyWorks project (e.g., “position 1”)




The workflow using Reference points within an HxGN Robotic Automation program is shown in the following two examples below.

Example: Rotary table moving to different scan positions

1. Rotary Table and Robots move to home positions
2. Create the positions in PolyWorks through a macro in case they don’t exist yet in the project.
3. Set Device Position Macro
4. Measure the reflectors
5. Measure a COP

6. Rotate the Rotary Table
7. Set Device Position Macro
8. Measure the reflectors
9. Measure a COP
10. Repeat steps 6) to 9) to add more rotary table scan positions

Example: Tracker slider moving to different scan positions

1. Tracker slider and Robots move to home positions
2. Create the positions in PolyWorks through a macro in case they don't exist yet in the project.
3. Set Device Position Macro
4. Measure the reflectors
5. Measure a COP
6. Move the tracker slider
7. Set Device Position Macro
8. Measure the reflectors
9. Measure a COP
10. Repeat steps 6) to 9) to add more tracker slider scan positions

The order of either doing first the COP measurement (step 5) or the reflector measurement (step 4) is not important.

Due to a PolyWorks limitation it is required that the COP names cannot be re-used in various positions. E.g.:

Position1

- Start measurement (COP NAME: BIW_LEFT)
- Start measurement (COP NAME: BIW_RIGHT)

Position2

- Start measurement (COP NAME: BIW_LEFT) → **Error!**

PolyWorks does not support multiple positions at the same time. This means reference points cannot be used for a dual robot, dual tracker cell where it is needed to have two parallel scans at the same time. Using HxGN Robotic Automation compensation (Nominal vs Actual reflector positions, see chapter Compensation) is recommended for this use case.

## List of supported features

The following table shows if the different metrology features supported by HxGN Robotic Automation can be imported from the different metrology applications to which it is interfaced with. It also shows a quick reference on which features are supported by the path planning algorithm based on best practices for obtaining the

best results using laser scanners (please refer to Hexagon Metrology Devices - AS1 Best Practice and Hexagon Metrology Devices - T-Scan5 Best Practice).

| Feature type | Import-able from PC-DMIS as | Importable from Inspire as | Importable from PolyWorks as | Sup-ported by path plan-ning |
|---|---|---|---|---|
| Circle | | | | |
| Cone | | | | |
| Cylinder | | | | |
| Edge Point | Edge point (BK) High point (BH) | Edge point (BK) High point (BH) | TrimmedEdgeCom-parisonPoint HemmedEdgeCom-parisonPoint | See (1) |

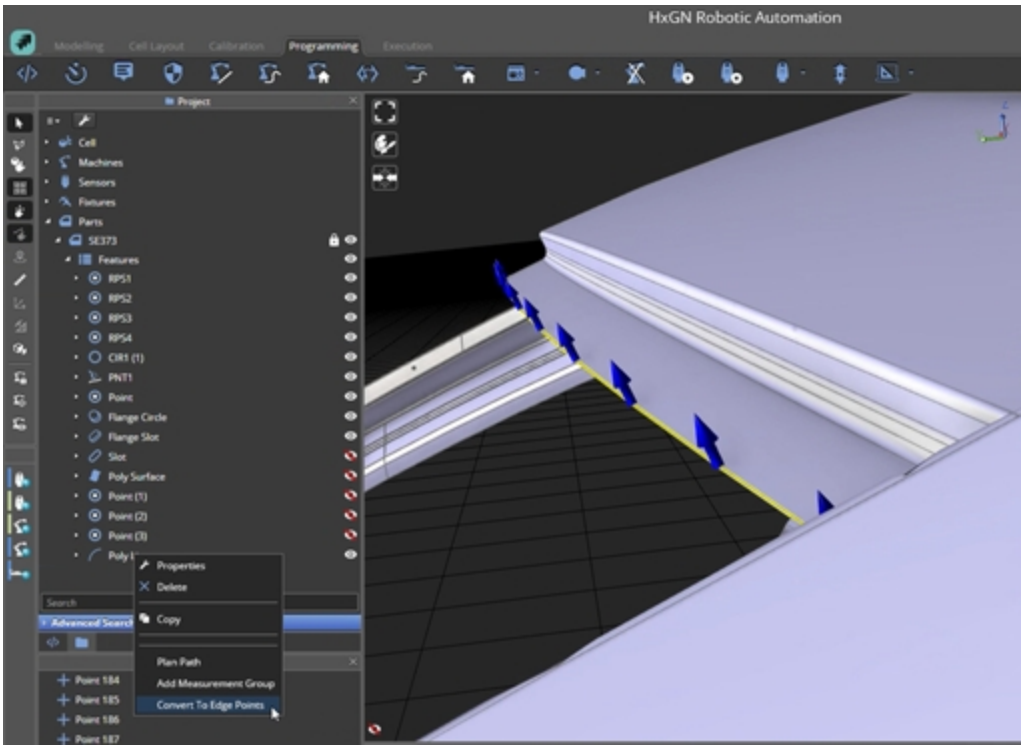
| Feature type | Import-able from PC-DMIS as | Importable from Inspire as | Importable from PolyWorks as | Sup-ported by path plan-ning |
|---|---|---|---|---|
| Ellipse | ✗ | ✓ | ✓ | ✓ |
| Flanged Circle | ✓ See (2) | ✗ | ✗ | ✗ |
| Flanged Slot | ✗ See (2) | ✗ | ✗ | ✗ |
| Hexagon | ✓ | ✗ | ✗ See (3) | ✗ |
| Line | ✗ | ✓ | ✓ | ✗ |

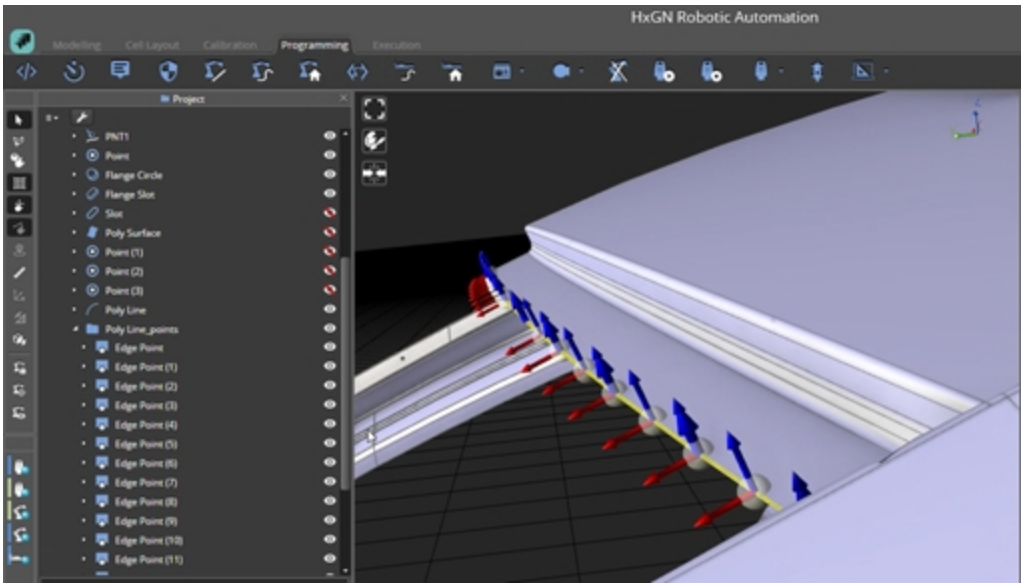
| Feature type | Import-able from PC-DMIS as | Importable from Inspire as | Importable from PolyWorks as | Sup-ported by path plan-ning |
|---|---|---|---|---|
| Plane | | | See (4) | |
| Poly-sur-face | | CADFeatur-e (Surface) | See (5) | |
| Polyline | | | | See (6,7) |
| Slot | | | Slot, Polygon (Rectangle) (3) | See (7) |

| Feature type | Importable from PC-DMIS as | Importable from Inspire as | Importable from PolyWorks as | Supported by path planning |
|---|---|---|---|---|
| Sphere | ✓ | ✓ | ✓ | ✓ |
| Surface point | ✓<br>Surface pont | ✓<br>Point (w/ or w/o normal) | ✓<br>SurfaceComparisonPoint ComparisonPoint | ✓ |

1. Scanned only along the edge direction, not the internal surface direction.
2. Imported as cylinders. Requires VW features enabled.
3. PolyWorks considers all polygons equal (hexagons, octagons, etc.).
4. Planes are defined by point + vector in PolyWorks and list of vertexes of a polygon in Robotic Automation so Plane is imported in the project tree but cannot be visualized.
5. PolyWorks treats it as a mesh.
6. The user needs to deconstruct the polyline into a set of points so that each is scanned along the direction of its corresponding line before the path planning algorithm can be used on it (see below figure).

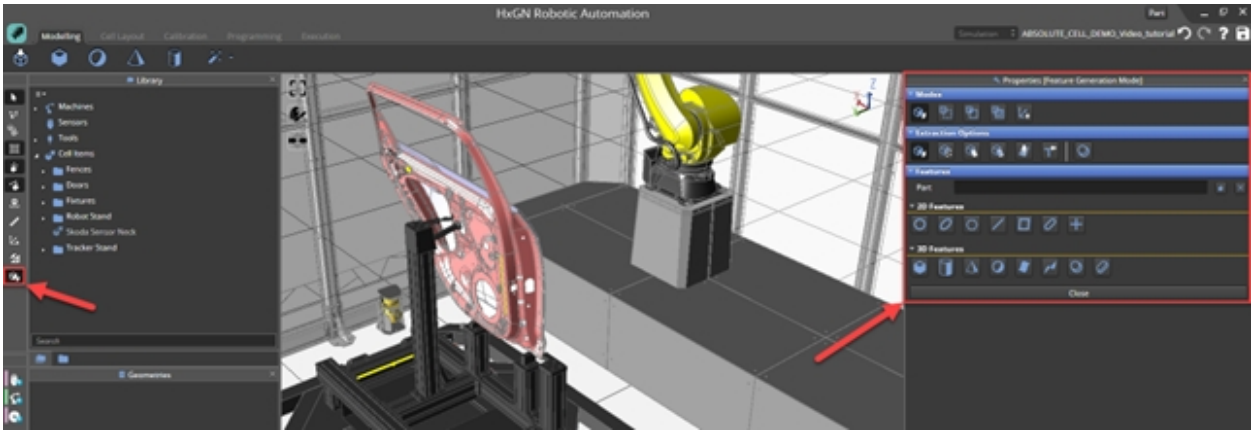
7. These features have a specific direction in which they are scanned (e.g., a slot is scanned diagonally and not following the major axis direction, a polyline is scanned along each of the lines constituting it).

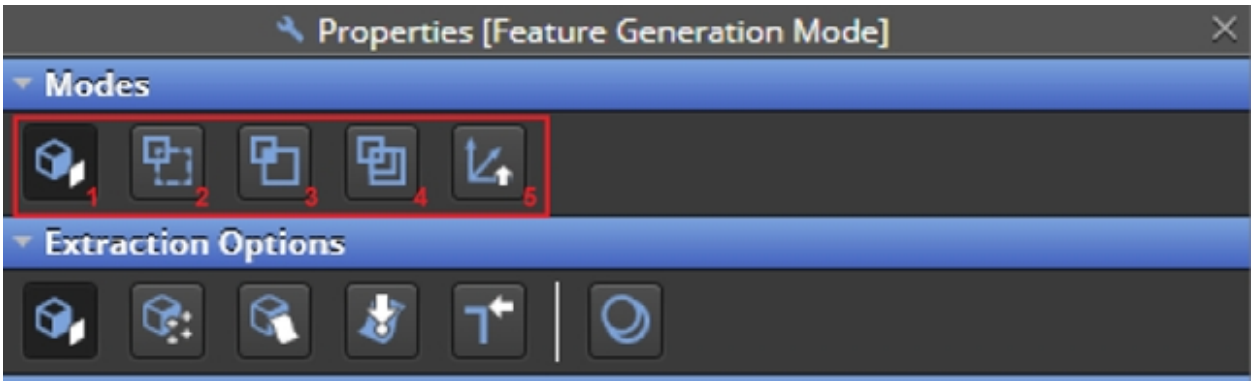





## Feature Generation Mode

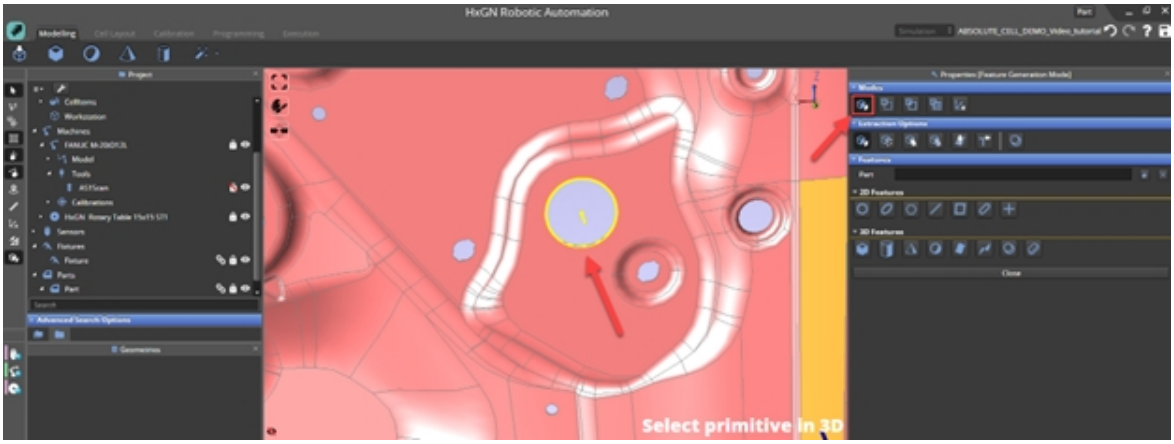
The user can generate features from HxGN Robotic Automation using the feature generation mode. After clicking on the Feature Generation Mode button on the special modes' toolbar, in the Properties window it will be possible to select among different options defining how to create features.

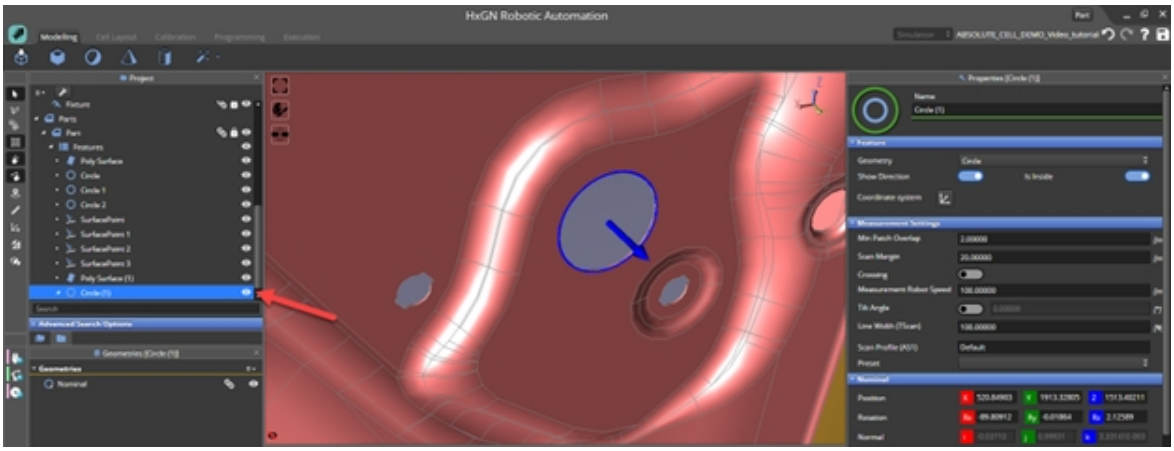





The extraction modes are:

1. **Extract from CAD**: the user needs to select the features by clicking on the CAD of the part in the 3D View as shown in the image below. After selecting the feature (e.g., a circle), the feature will be added to the Project Tree.






The feature can be extracted from the 3D view with different options:

a. Feature

b. Feature From Points

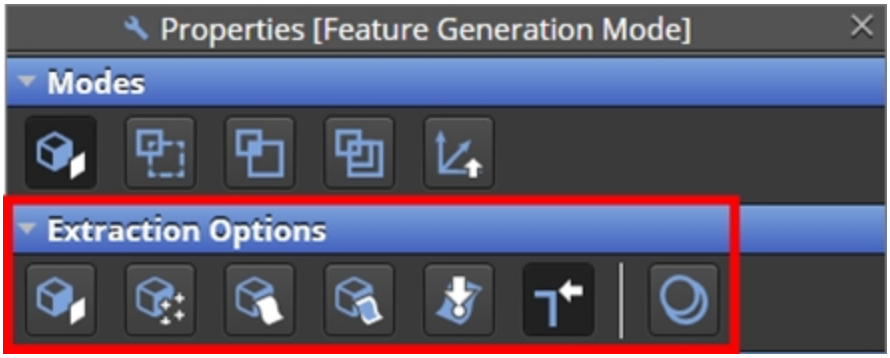
c. Surface

d. Edge

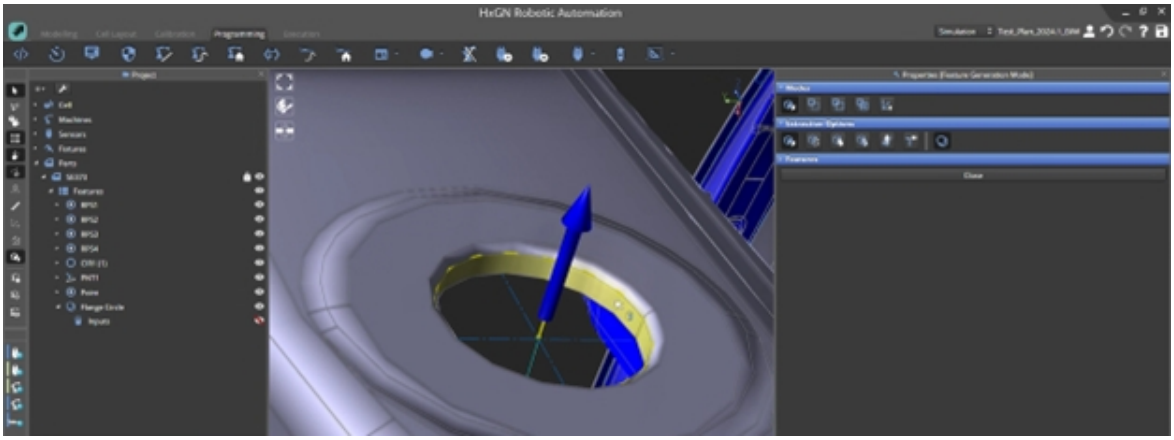
e. Surface Point

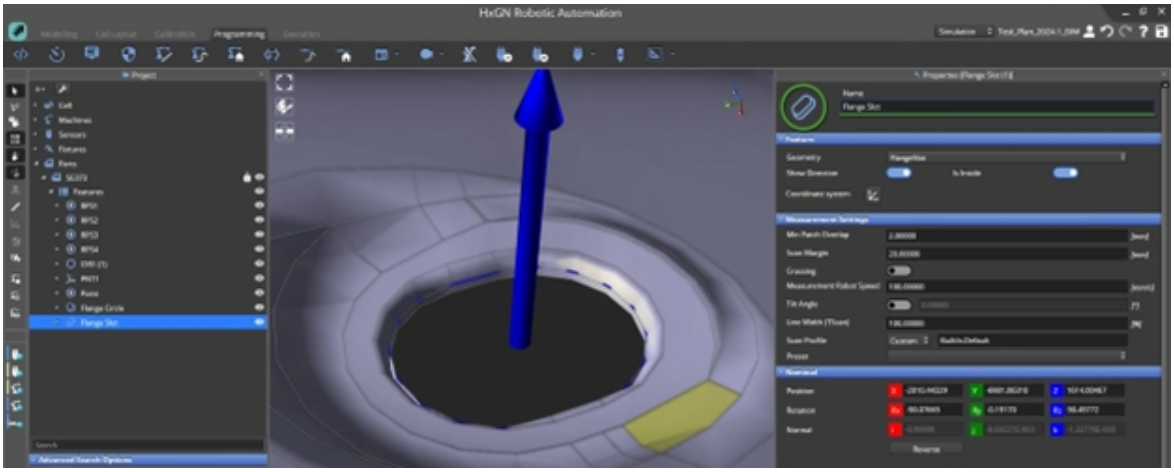
f. Edge Point

g. Flanged Feature Mode



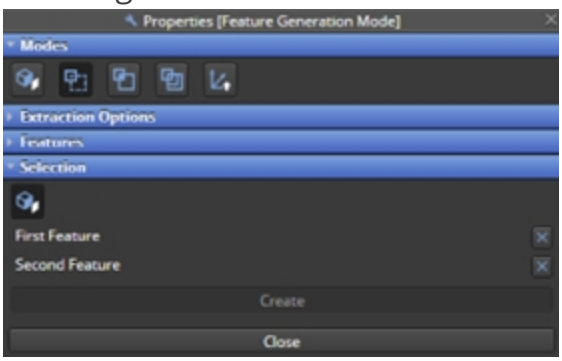
The examples below show how a flange circle and a flange slot can be extracted from the CAD.



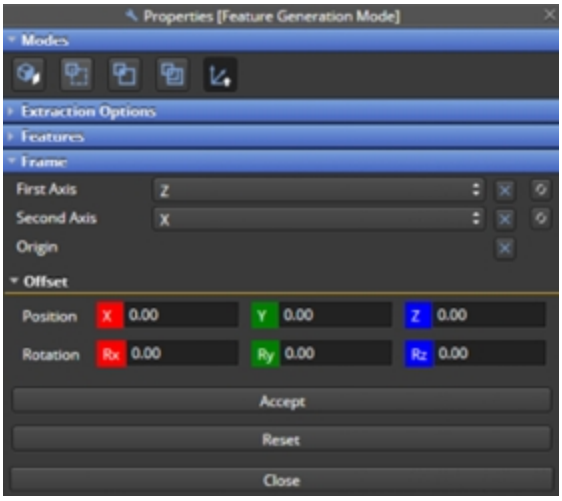




2. **Construct**: the user needs to select the features to do the construction as shown in the image below.
3. **Intersect**: the user needs to select the features to do the intersection as shown in the image below.

**Project**: the user needs to select the features to do the projection as shown in the image below.

4. 


**Frame**: the user needs to select axis and origin and the frame offset.

5. 

# Troubleshooting

The HxGN Robotic Automation logs are located in the following directory:

*C:\Users\YourUser\AppData\Roaming\Hexagon\RA\Logs*

The Generic Host logs are located in the following directory:

*C:\ProgramData\Hexagon\HxGN.RA.GenericHost\Logs*

# Frequently asked questions

| Question | Answer |
|---|---|
| Which Robots have a direct Interface with HxGN Robotics Automation? | FANUC robots using R-30ia controllers and KUKA robots using KR C4 controllers. |
| Which Scanners have a direct Interface with HxGN Robotics Automation? | HxGN Robotics Automation supports the Leica T-SCAN, AS1 and ZG Scanners.<br>More HW components will be integrated in future versions. |
| Which PLC's have a direct Interface with HxGN Robotics Automation? | HxGN Robotics Automation supports SIEMENS PLC Interface through OPC UA Protocol. |
| Which Metrology SW has direct Interface with HxGN Robotics Automation? | HxGN Robotics Automation interfaces with HxGN PC-DMIS and Inspire Metrology SW. It also works with InnovMetric PolyWorks.<br>More SW Packages from HxGN and 3rd party will be supported in future versions. |
| Which Robot post-processors exist in HxGN Robotics Automation? | None yet. |

| Question | Answer |
| --- | --- |
| What is a postprocessor? | A postprocessor is a unique "translator" specific to a robot controller which can convert simulated robot poses into a specific brand language that can be then loaded to the respective robot controller. |
| What can be done on different Robot types with no direct Interface? | If a postprocessor exists, the respective robot program can be generated and loaded to the robot controller. |
| What is Robot Simulation in HxGN Robotics Automation good for? | HxGN Robotics Automation Simulates the robot path generated for path inspection, taking into consideration the robot volume, collisions, tool/scanner occlusion, tracker LOS (Line of Sight), sensor data simulation.<br>All this delivers a realistic simulation of the part inspection process that can be validated offline, even before building the actual cell. |
| How to get a demo SW? | After a training session a demo license is delivered to HxGN employees. In case a demo license is needed for a customer, please raise a ticket on Jira |
| We have a customer with AT901 T-Mac trackers that has asked to use HxGN Robotics Auto- | HxGN Robotics Automation is designed to work best with a T-Mac and AT960 using the LMF interface to connect to the Leica Absolute Trackers. |

| Question | Answer |
| --- | --- |
| mation. Any chance that we will see an update that supports the AT901? | |
| How is the security granted on the software? | A cloud license is used to protect the software. That means you can run HxGN Robotics Automation on as many PCs as you want. All that is needed is to deactivate a license on one PC and reactivate it on another one. The preferred way to use it is over the internet using online access, but we also supply a way to activate the license offline. |
| What is the recommended training time? | HxGN Robotics Automation can typically be trained in 3 days. Prerequisites for this training are AT9x0 and basic robot knowledge. Basic Metrology SW Knowledge is recommended (PC-DMIS, Inspire or PolyWorks). |
| Is it possible to imagine a TCP link between HxGN Robotics Automation and other robots different than FANUC or KUKA? | A direct link to other Robots can be develop on request and will be available in upcoming versions. |
| Is it possible to edit robot programs in HxGN Robot- | No, currently it is not possible to import Robot programs to the HxGN Robotics Automation SW. Still, programs |

| Question | Answer |
| --- | --- |
| ics Automation that were built on the robot teach pendant? | can be executed Step by Step, and Robot Motions can be recorded in HxGN Robotics Automation SW. |
| Is Automatic path planning based on CAD features supported? | Yes, HxGN Robotics Automation supports path planning based on features and on CAD entities as surfaces and edge lines. |
| Does HxGN Robotics Automation have a 'results viewer,' or should this be done in the metrology software you choose? | No, Measurements Results should be reviewed in the Metrology Software. |
| Is it possible for HxGN Robotics Automation also to simulate the handling of components, such as fixtures and workpieces? | No, we are considering that functionalities for future releases. |
| Cable Modelling and Simulation, dress pack movement? Will it be simulated? | No, HxGN Robotics Automation does not include cable modelling, so dress pack movement is not supported. |
| Are there any strategies | Yes, user can define in HxGN Robotics Automation dif- |

| Question | Answer |
| --- | --- |
| that can be used to scan distinct types of features, such as flanged holes and slots and so on? | ferent presets or Scan Strategies for the distinctive features to be measured. |
| Is it planned to integrate custom communication [example VASS / VKRC] | Yes, this can be managed with the customer as a custom development project. |
| Does HxGN Robotics Automation directly control the scanner settings which can change with the features to be measured? | Yes, HxGN Robotics Automation controls the scanner settings, and these can be adjusted in HxGN Robotics Automation based on the feature type and feature scanning strategy. |

# Known Issues

| Description | Workaround | JIRA item |
|---|---|---|
| No pointcloud when using continuous motions with ZG sensor | Remove the continuous motions for the program with a ZG sensor | https://hexagon.atlassian.net/browse/BF-15974 |
| ATS800 is not working fully offline and online | The main remaining issues will be fixed in an upcoming patch. | |
| Planes imported from PC-DMIS can sometimes show an incorrect rotation | Manually adjust the plane primitive in HxGN Robotic Automationso it matches PC-DMIS nominal definition | https://hexagon.atlassian.net/browse/BF-15978 |
| Loading the project gives a collision warning on the tracker | Due to HxGN Robotic Automationnot controlling the tracker rotation directly, it can cause potential collisions.<br>To avoid this a minimum clearance | Clearance Settings |

| Description | Workaround | JIRA item |
| --- | --- | --- |
| that didn't appear in previous version | has been added to the Leica trackers. | |
| PC-DMIS will give an "failed to start Automation" error in the connector | Manually stop the PCDLNR.exe process in the task manager or restart Generic Host or use PC-DMIS 2024.1 SP2 | https://hexagon.atlassian.net/browse/PCD-282138 |
| Uninstalling HxGN Robotic Automation doesn't remove all the files | Manually go to: C:\Program Files\Hexagon\RA and remove the files. | https://hexagon.atlassian.net/browse/BF-15924 |
| App Manager language drop down doesn't work | No workaround. The function has been implemented, but the translations are not integrated yet. | |
| Measurement with T-probe takes sometimes 5 | This is a known limitation and is aimed to be fixed in the linked bug | https://hexagon.atlassian.net/browse/BF-15812 |

| | |
| --- | --- |
| Known Issues | 617 |

| Description | Workaround | JIRA item |
| --- | --- | --- |
| seconds | | |
| After changing the limits of HxGN Robotic Automation they are not working. | HxGN Robotic Automationneeds to be restarted for the limits to be updated. | https://hexagon.atlassian.net/browse/BF-15915 |
| SCRT reflectors added to the cell are not always orientated with the normal of the item they are placed on | Manually adjust the SCRT reflectors orientation after having placed them with the add reflector wizard | https://hexagon.atlassian.net/browse/BF-15822 |
| Metrology Service is "Not Defined" or part not defined in the project | This can occur if starting HxGN Robotic Automationimmediately after starting Generic Host. Please close HxGN Robotic Automation and when starting Generic Host wait 1 or 2 minutes for all service to be loaded before start- | https://hexagon.atlassian.net/browse/BF-15930 |

| Description | Workaround | JIRA item |
| --- | --- | --- |
| | ing HxGN Robotic Automation. | |
| PolyWorks Measurement properties are showing when using a ZG scanner | These properties are specifically for the Leica scanner point cloud and have no functionality. The linked bug is there to remove these when selecting a ZG scanner. | https://hexagon.atlassian.net/browse/BF-15913 |
| License expire warning window is hidden when starting RA | Check in the task bar the corresponding warning window and close it from there. | https://hexagonmi.atlassian.net/browse/BF-13760 |
| Forgotten root Hexagon Password | 1. In the windows start button type %appadata% and press enter<br>2. Go to Roaming/Hexagon/HxGN.RA.GenericHost/LocalDb<br>3. Delete the “RA Identity” | |
| Generic Host can not con- | **Option 1**:<br>1. Stop RA (and GH) | |

| Description | Workaround | JIRA item |
| --- | --- | --- |
| nect due to the IP and port are occupied | 2. To find the process using port 80 → run cmd and execute this command →"netstat -ano \| find "80".<br>3. Find which process is using port 80 and if it's not needed stop it.<br>4. Start Generic Host<br>5. Start RA → everything should be working as expected.<br>**Option 2** (just in case port 80 has to be used by another proccess/app):<br>1. Stop RA (and GH)<br>2. Change ApiPort from 80 to 5000 in Basic window in App Manager<br>3. Change UserManagementURL value from http://-localhost/usermanager to *http://-localhost/usermanager:5000 in "C:\ProgramData\Hexagon\RA\Config\RAconfig.xml*<br>4. Start Generic Host | |

| | |
| --- | --- |
| Known Issues | 620 |

| Description | Workaround | JIRA item |
| --- | --- | --- |
| | 5. Start RA → everything should be working as expected.<br><br>In this case, URLs for execution manager and user manager will be:<br><br>• *http://-localhost:5000/execmanager/*<br>• *http://-localhost:5000/usermanager/* | |
| Point cloud is not managed properly on Inspire when it comes from a parallel execution in Robotic Automation. | N/A<br><br>Parallel execution is not currently supported when using Inspire as metrology software . | https://hexagonmi.atlassian.net/browse/BF-14035 |
| Point cloud is not managed properly on Inspire when it comes from a parallel exe- | N/A<br><br>Parallel execution is not currently supported when using Inspire as metrology software. | https://hexagonmi.atlassian.net/browse/BF-14036 |

| Description | Workaround | JIRA item |
| --- | --- | --- |
| cution in Execution Manager. | | |
| Sometimes, Robotic Automation crashes when plugin/unplug in a second monitor. | If there is need to connect/disconnect a second monitor, make sure to save and close Robotic Automation before to do it. It will prevent Robotic Automation to crash or the open project to be corrupted. | https://hexagonmi.atlassian.net/browse/BF-10052 |
| Initialize Tracker button does not work. | N/A | https://hexagonmi.atlassian.net/browse/BF-12866 |
| Turn off Tracker button gets Tracker stuck. | N/A | https://hexagonmi.atlassian.net/browse/BF-12967 |
| Symptom: Due to a caching exception, we are not | Make certain the application is closed and go to the folder: *C:\ProgramData\Hexagon\HxGN.RA.GenericHost\CollisionStructures* | |

| Description | Workaround | JIRA item |
| --- | --- | --- |
| able to generate the Point Cloud in the simulation and might fail validation. | Delete the folder with the same name of the project having issues | |
| RDS Error: Requirements not met | Make certain you are using RDS 6.6<br>There is a feature in RDS 6.5 that can cause an issue: In case the lock gets lost and the user tries send a new RDS command (before an internal RDS timeout has been reached, this error message can occur: **Requirements not met** | |
| Not able to add, modify or delete planning presets | Make certain read, write and modify permission are given to the current user in “C:\ProgramData\Hexagon\RA\Config“ | |