

Week 4 – ABB Robot Teaching Offline Programming

Advanced Robotic Systems – MANU2453

Dr Ehsan Asadi, School of Engineering
RMIT University, Victoria, Australia
Email: ehsan.asadi@rmit.edu.au

Content

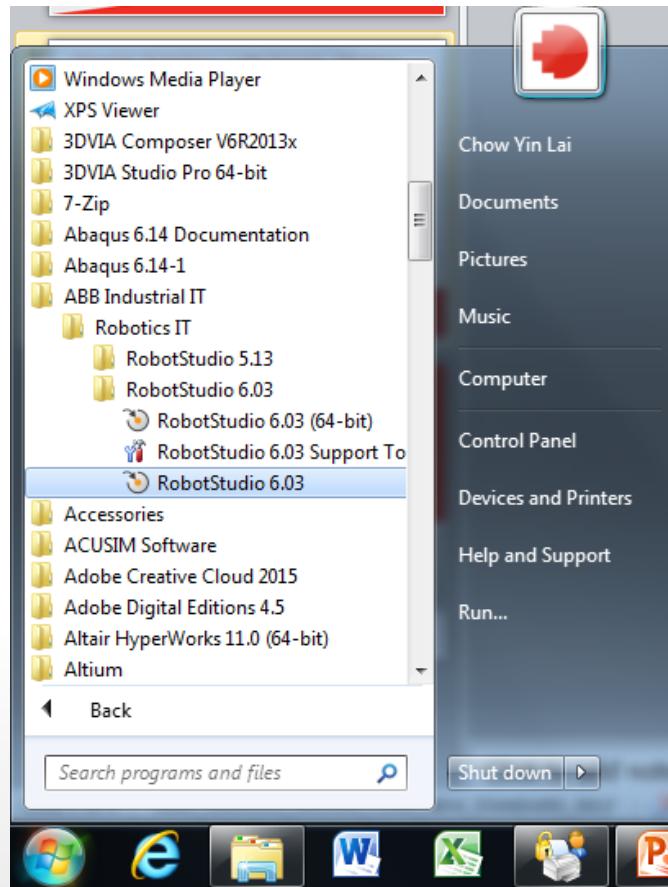
- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

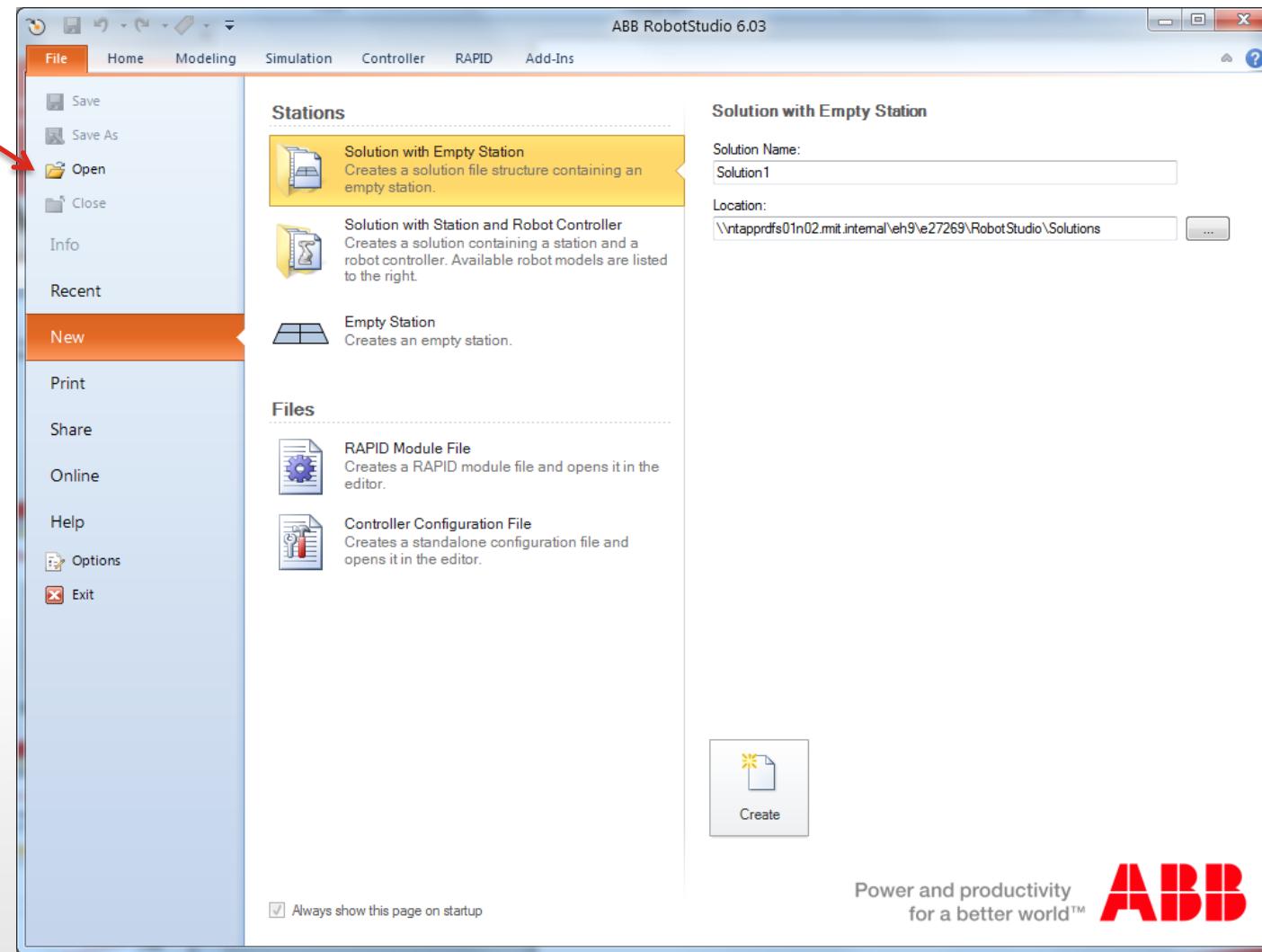
Open Solution from Previous Tutorial

- Start RobotStudio 6.03



Open Solution from Previous Tutorial

- In the next window, 1. Select “Open”

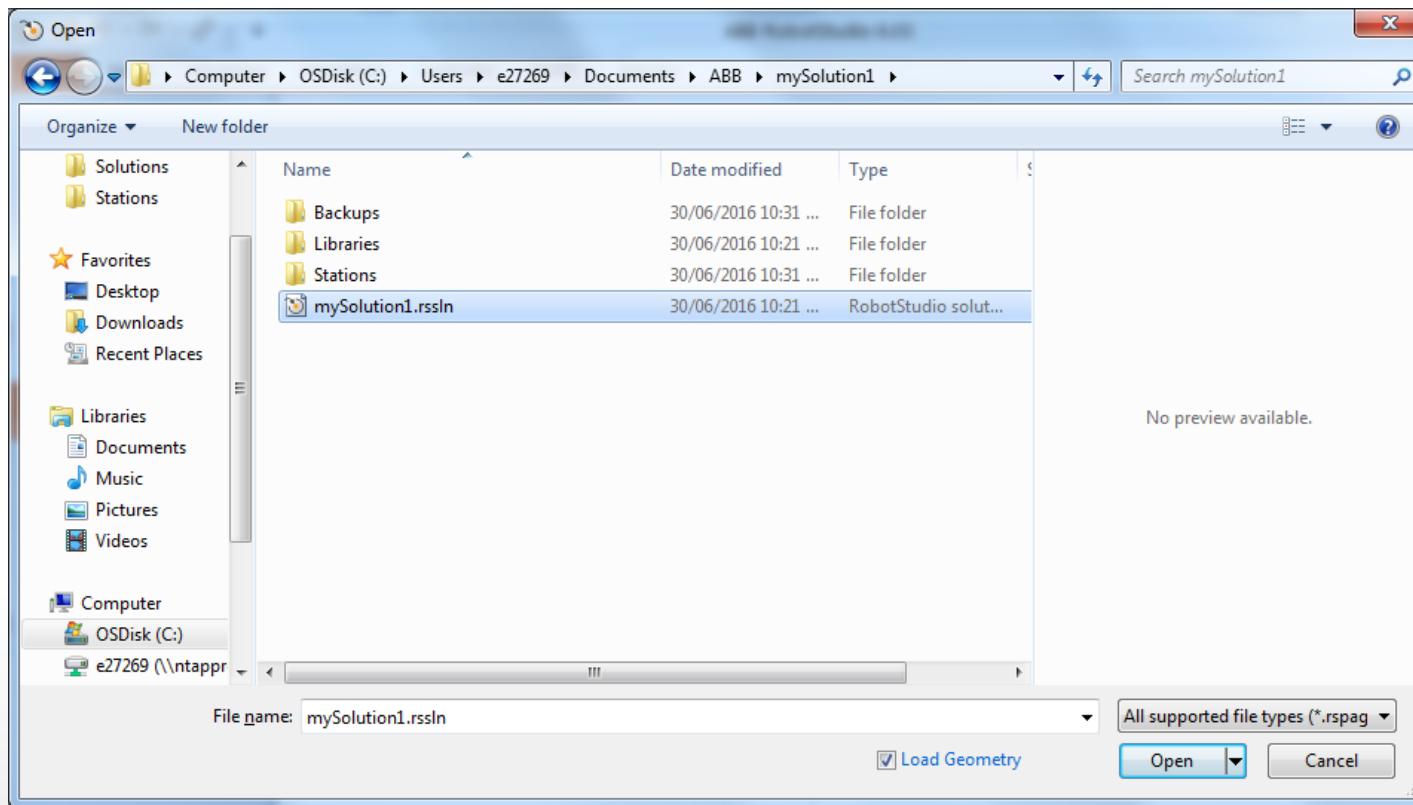


Power and productivity
for a better world™



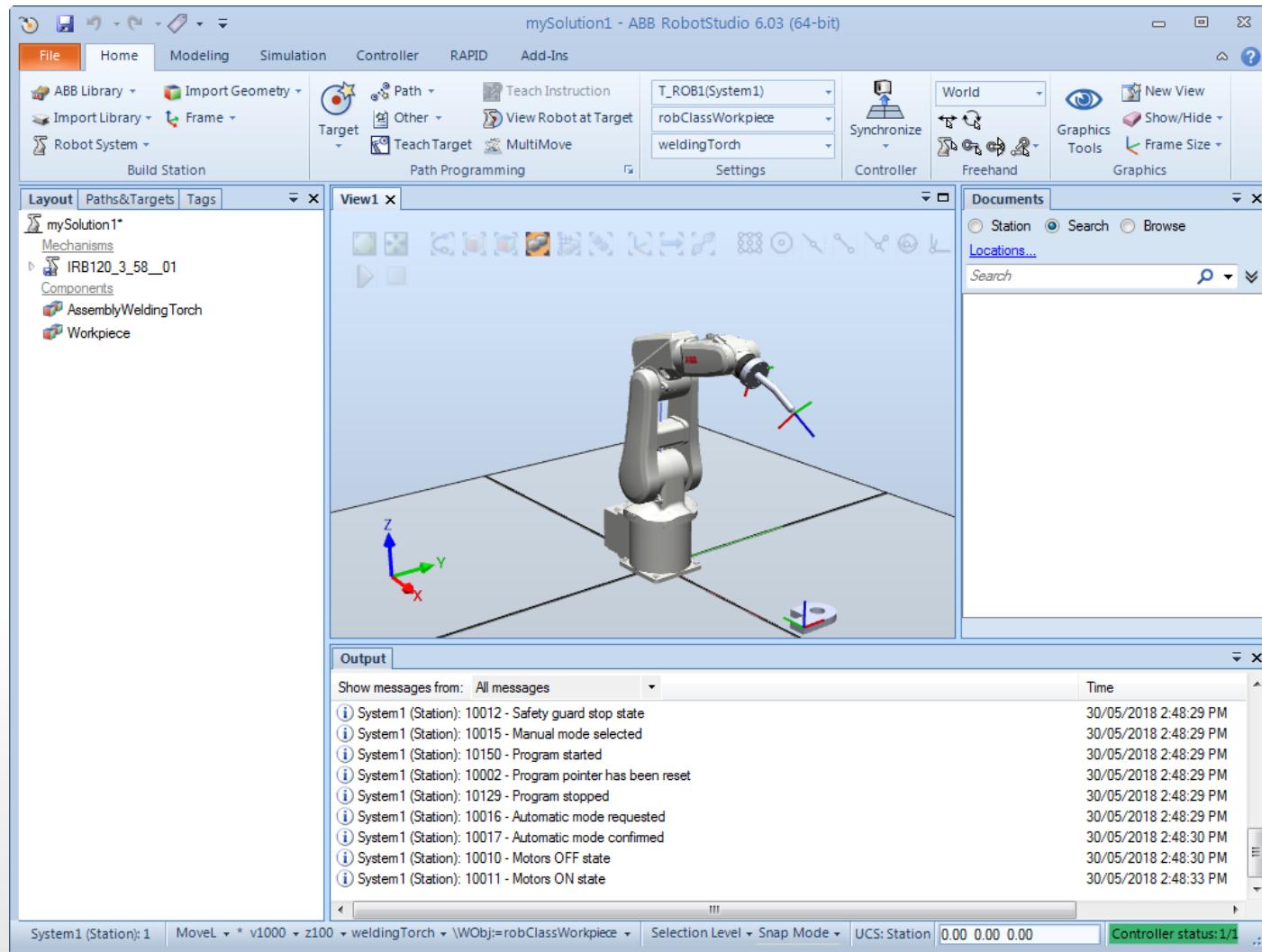
Open Solution from Previous Tutorial

- Look for the previously-created solution (.rssln file), then click Open.



Open Solution from Previous Tutorial

- The station will be loaded. Notice that the controller is also started.

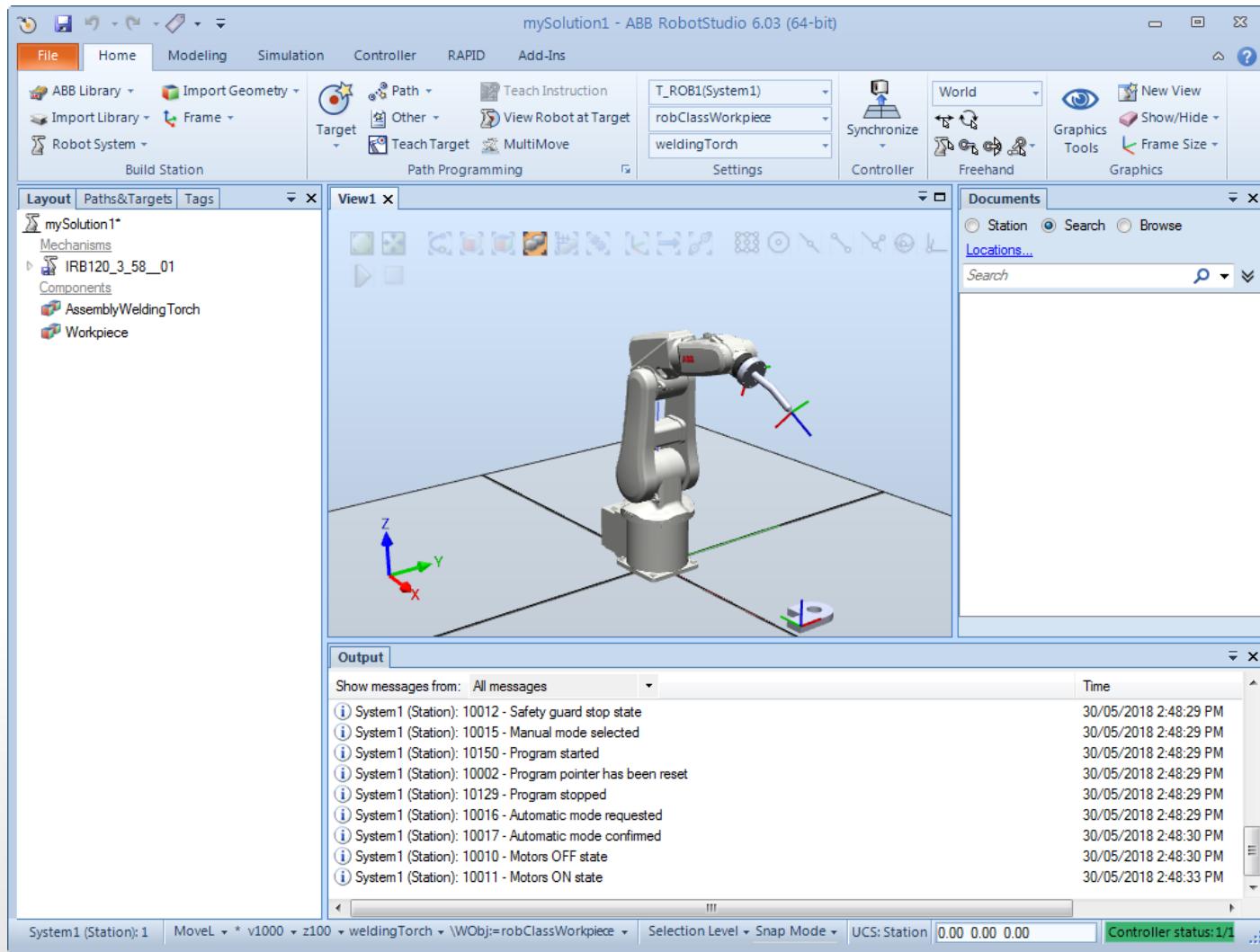


Content

- Open Solution from Previous Tutorial
- **Create Targets**
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

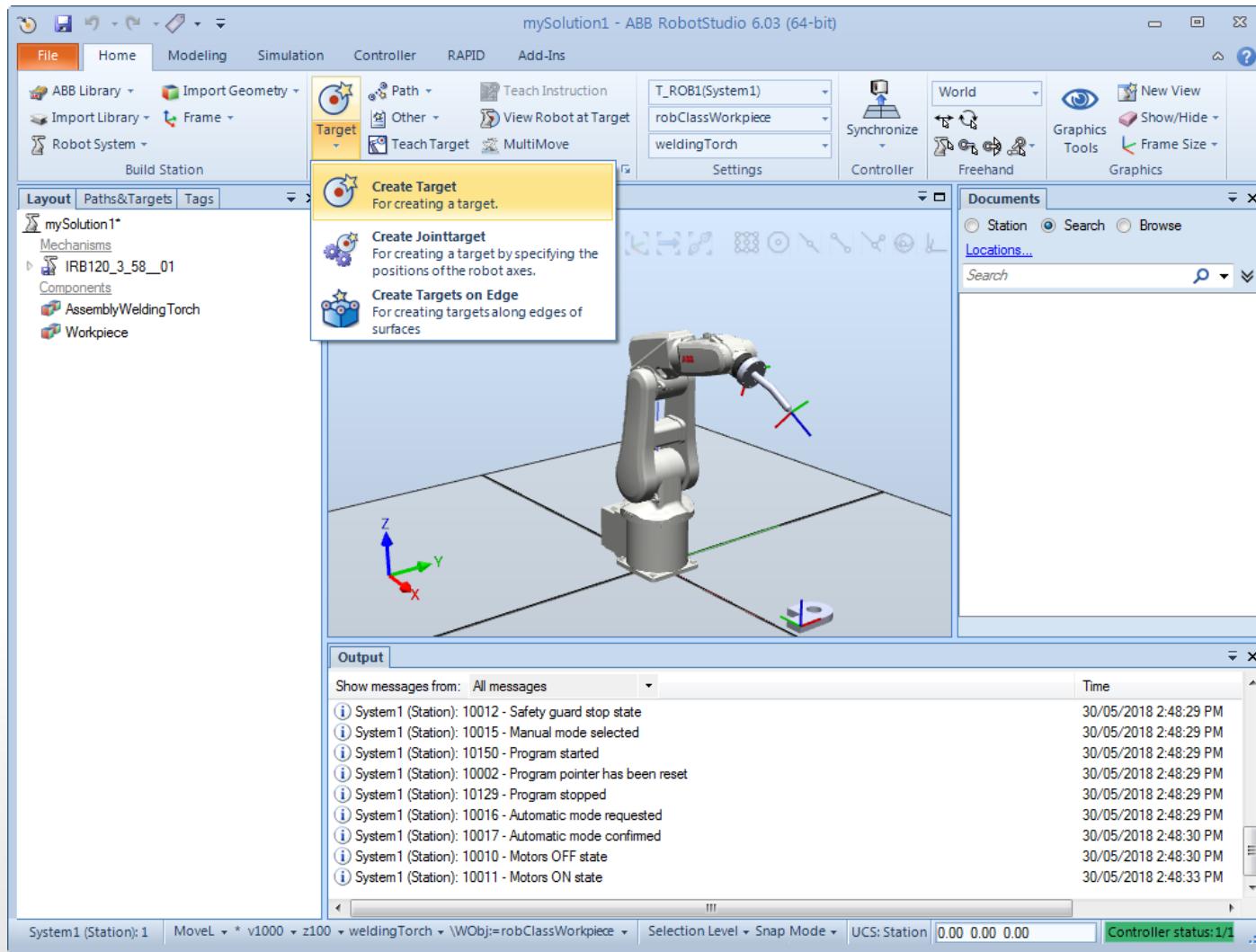
Create Targets

- Now, we will create target points with reference to workobject frame.



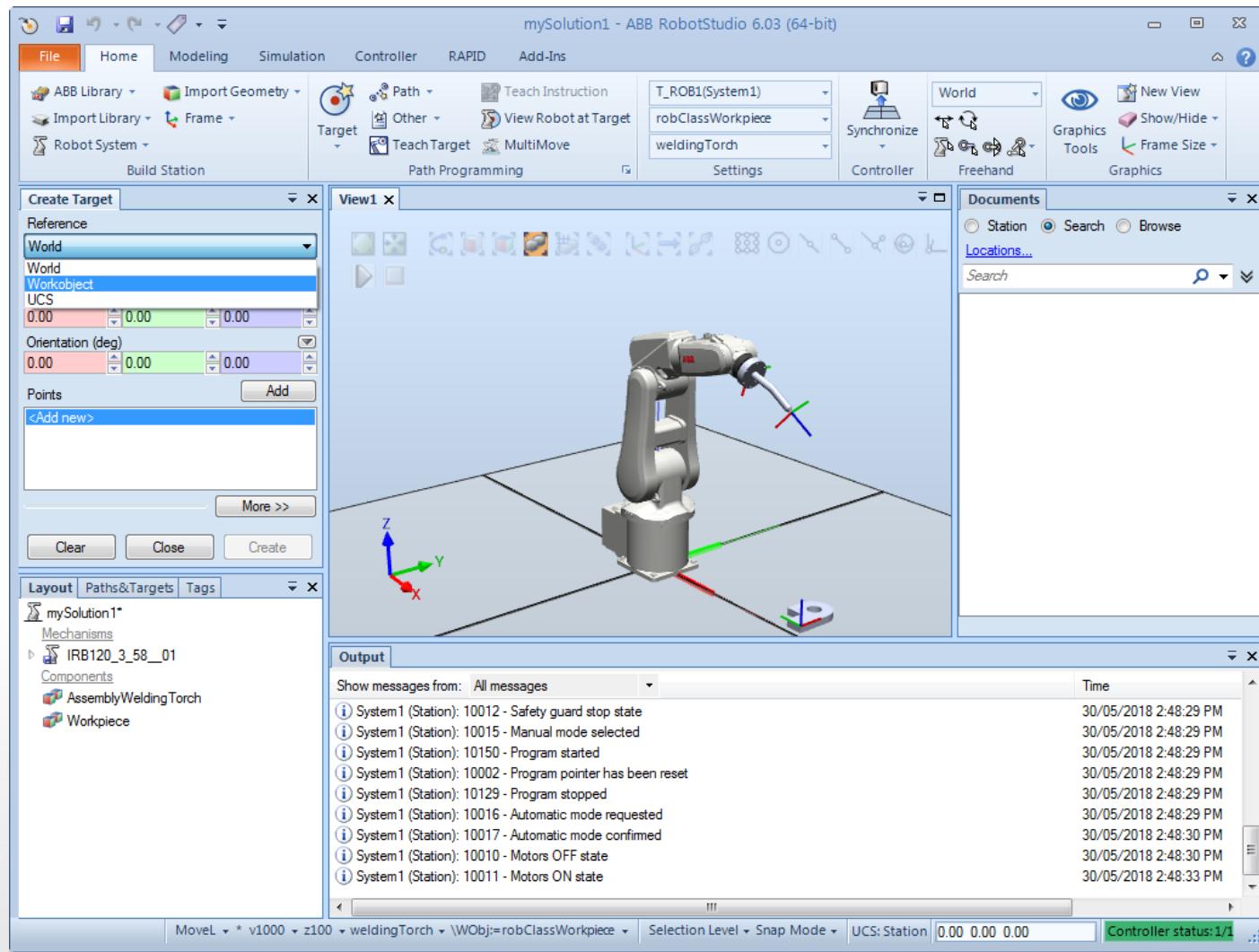
Create Targets

- Go to **create target** function.



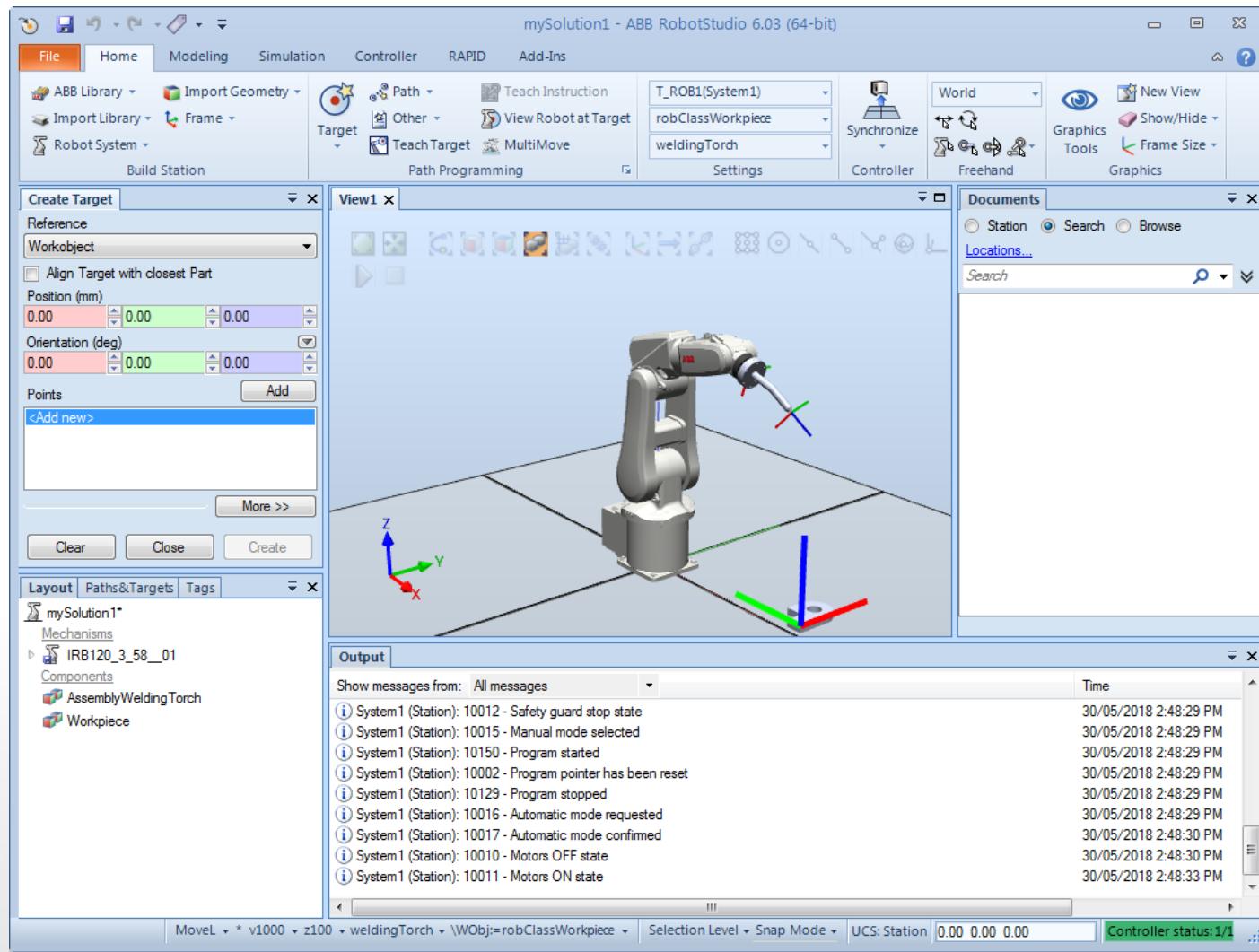
Create Targets

- At the left window, choose “Workobject” as Reference.



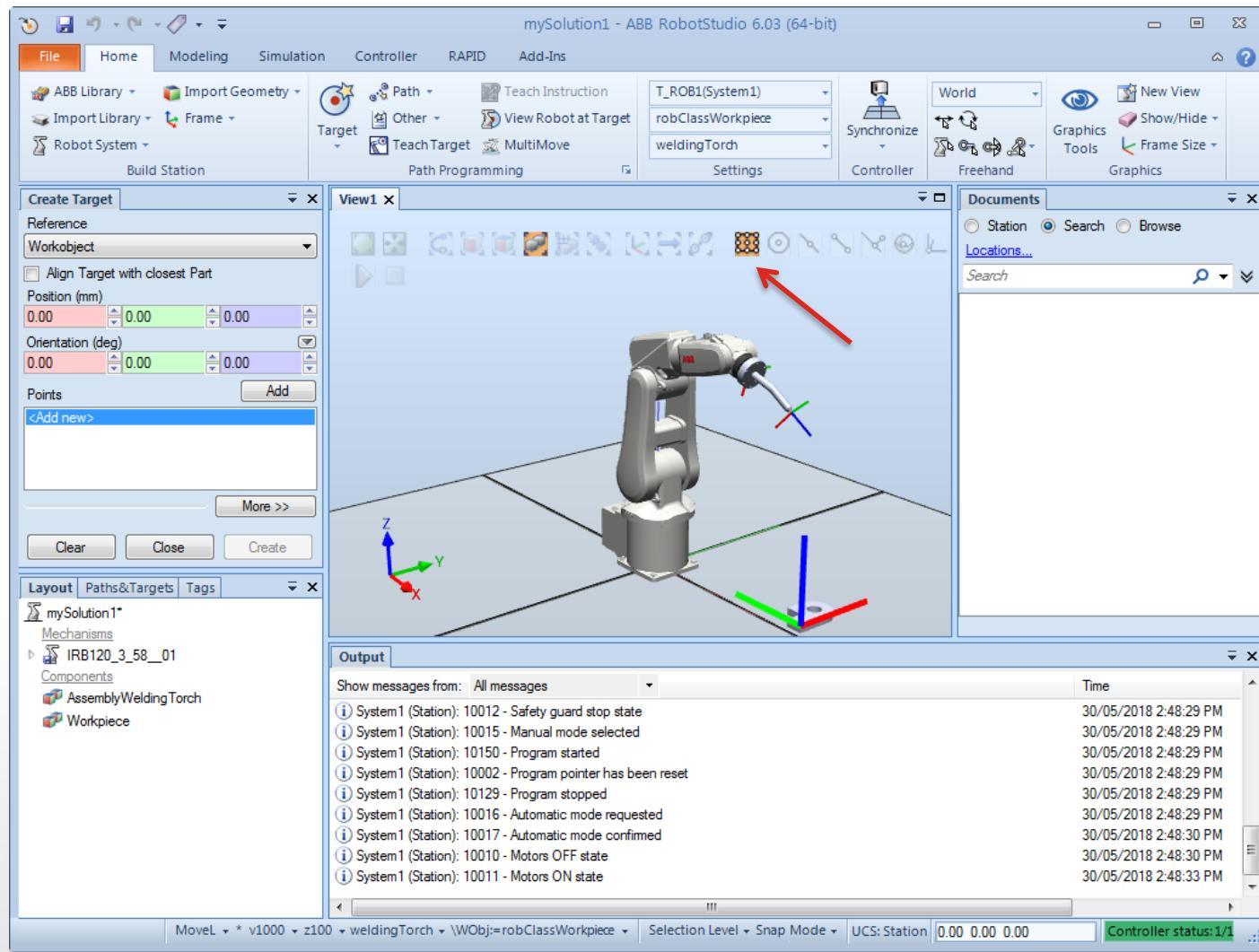
Create Targets

- The **workobject**, i.e. **robClassWorkpiece**, is highlighted.



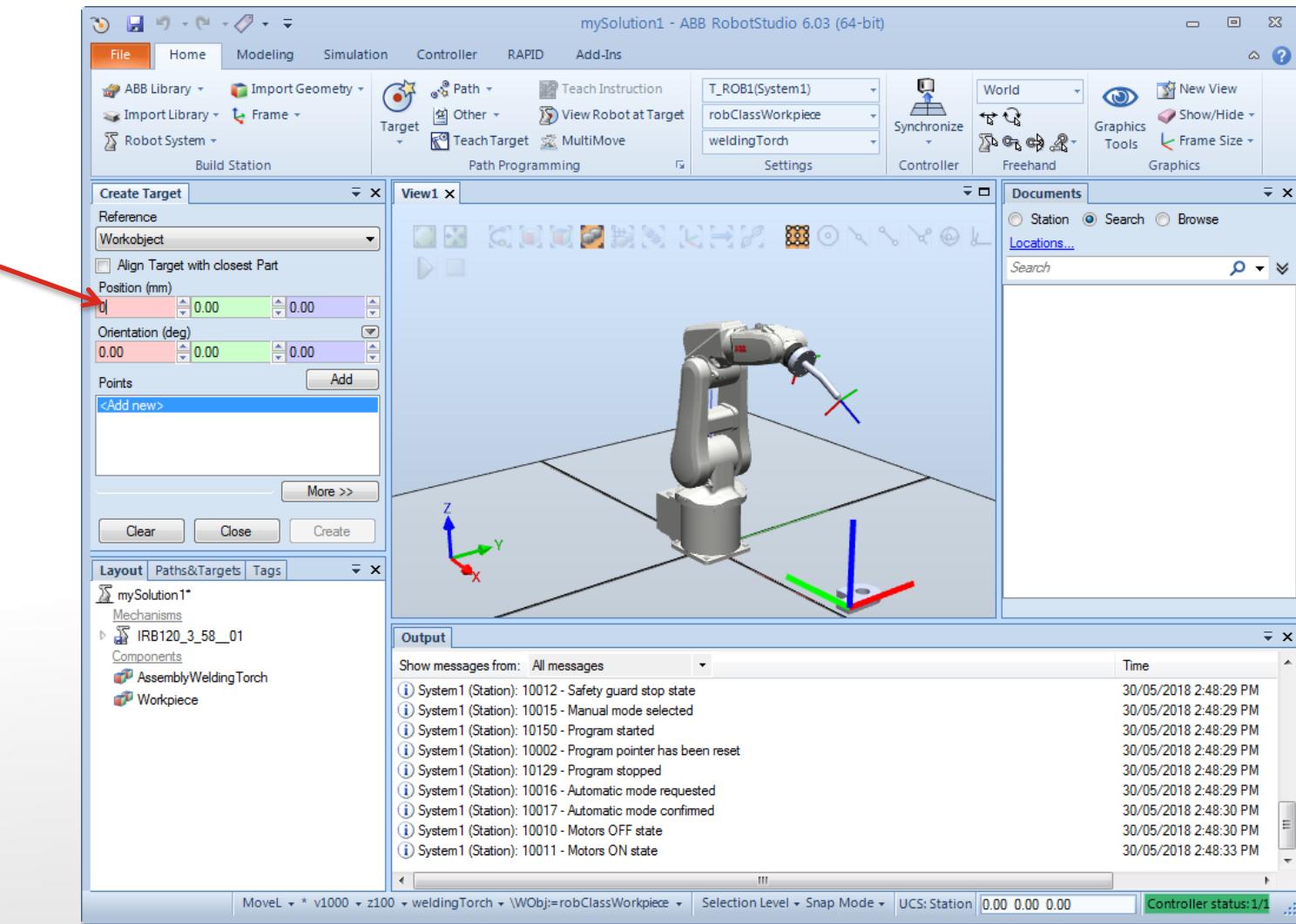
Create Targets

- Change snap mode to object.



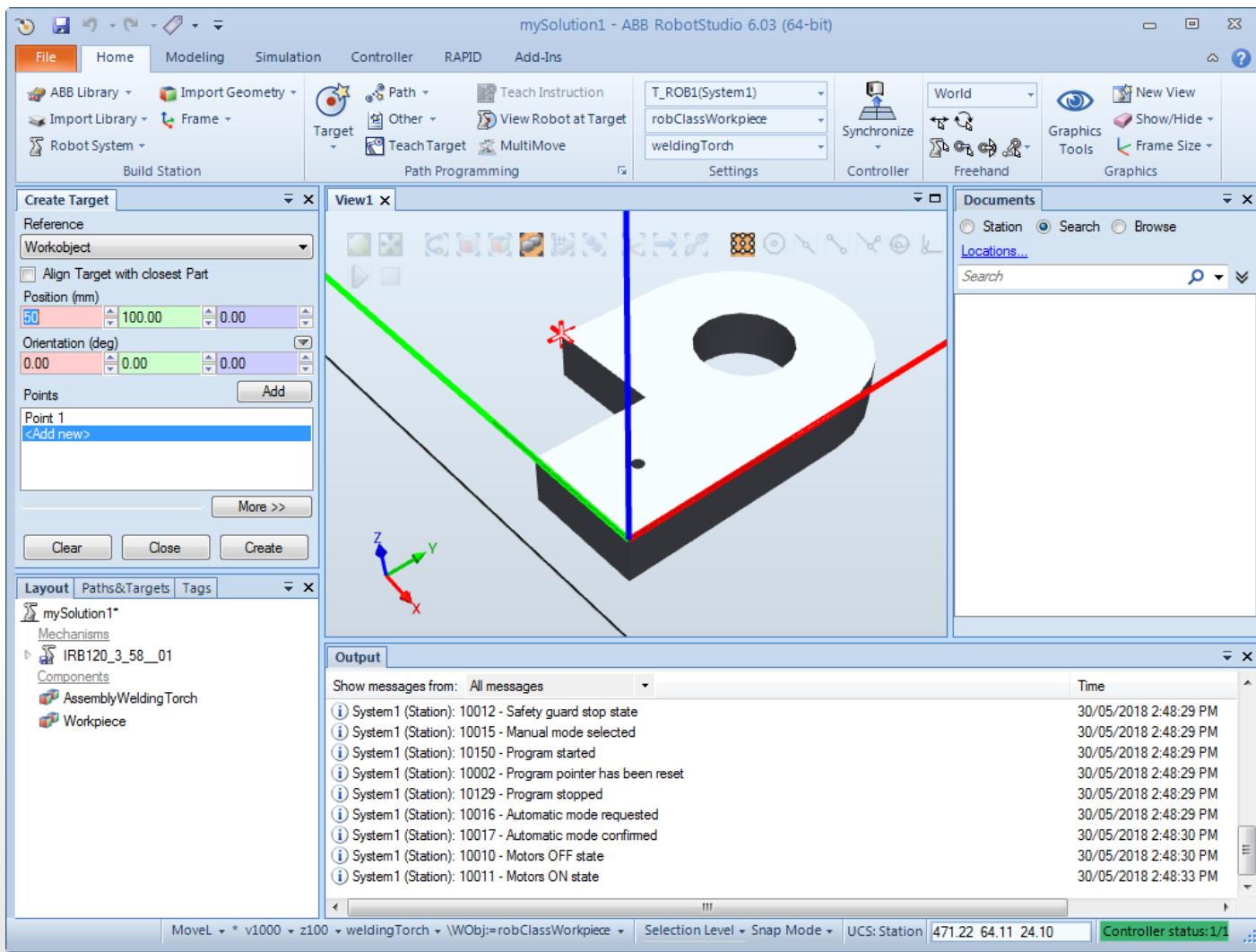
Create Targets

- Click in the **input window** on the left.



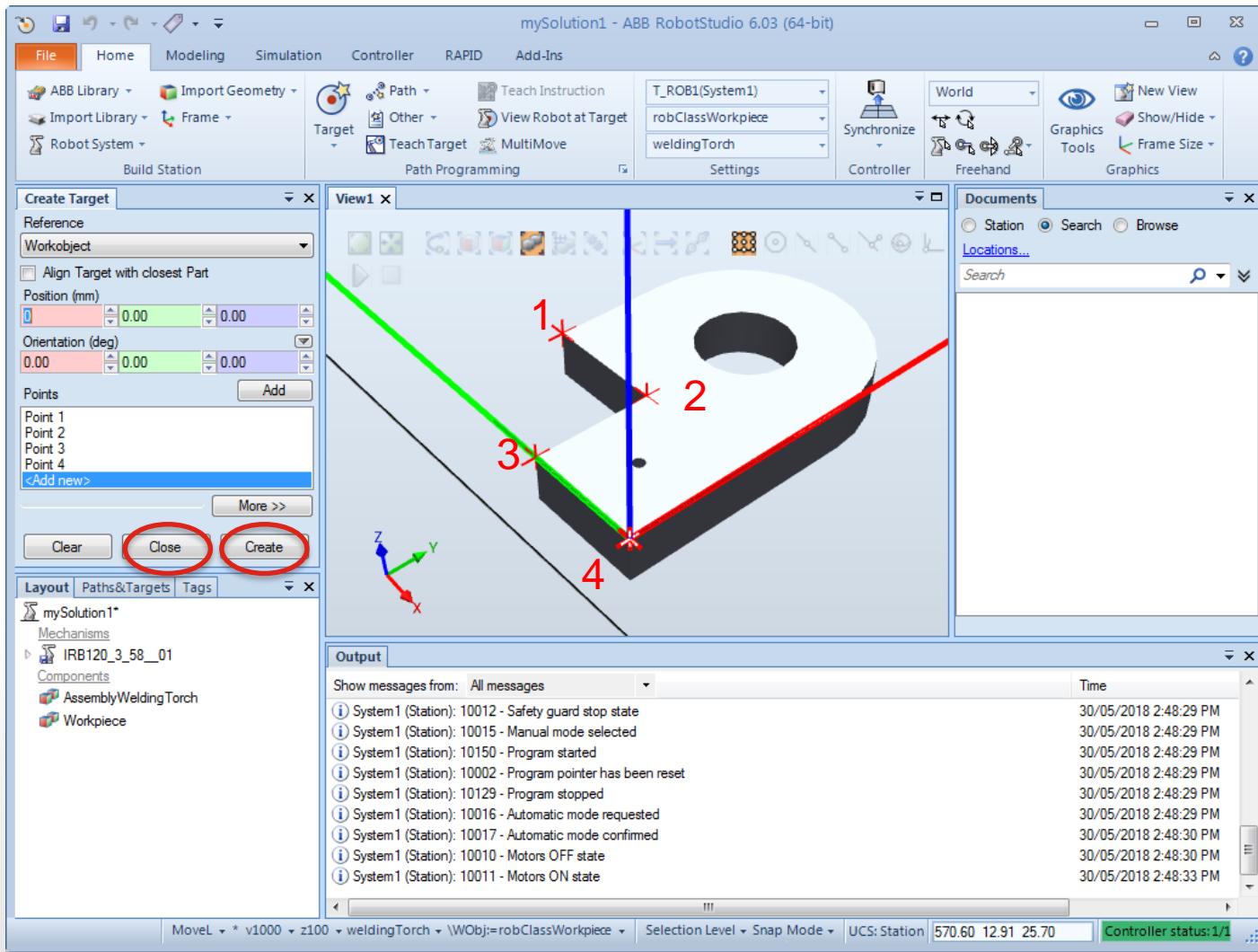
Create Targets

- Click the **first target point** on the workpiece. Notice that the values on the left have changed.



Create Targets

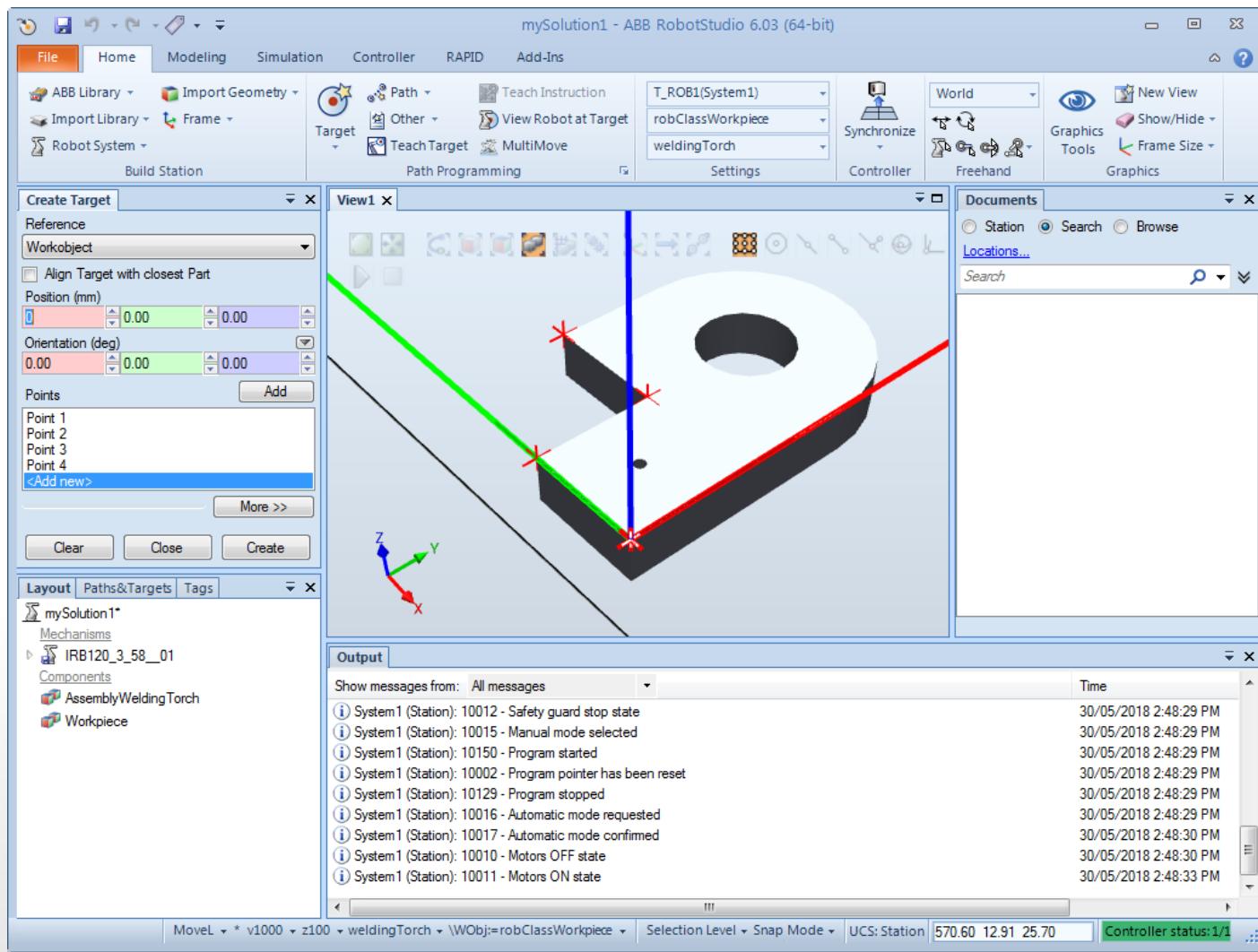
- Click on the 2nd, 3rd and 4th points, and then click “create” → close.



Create Targets

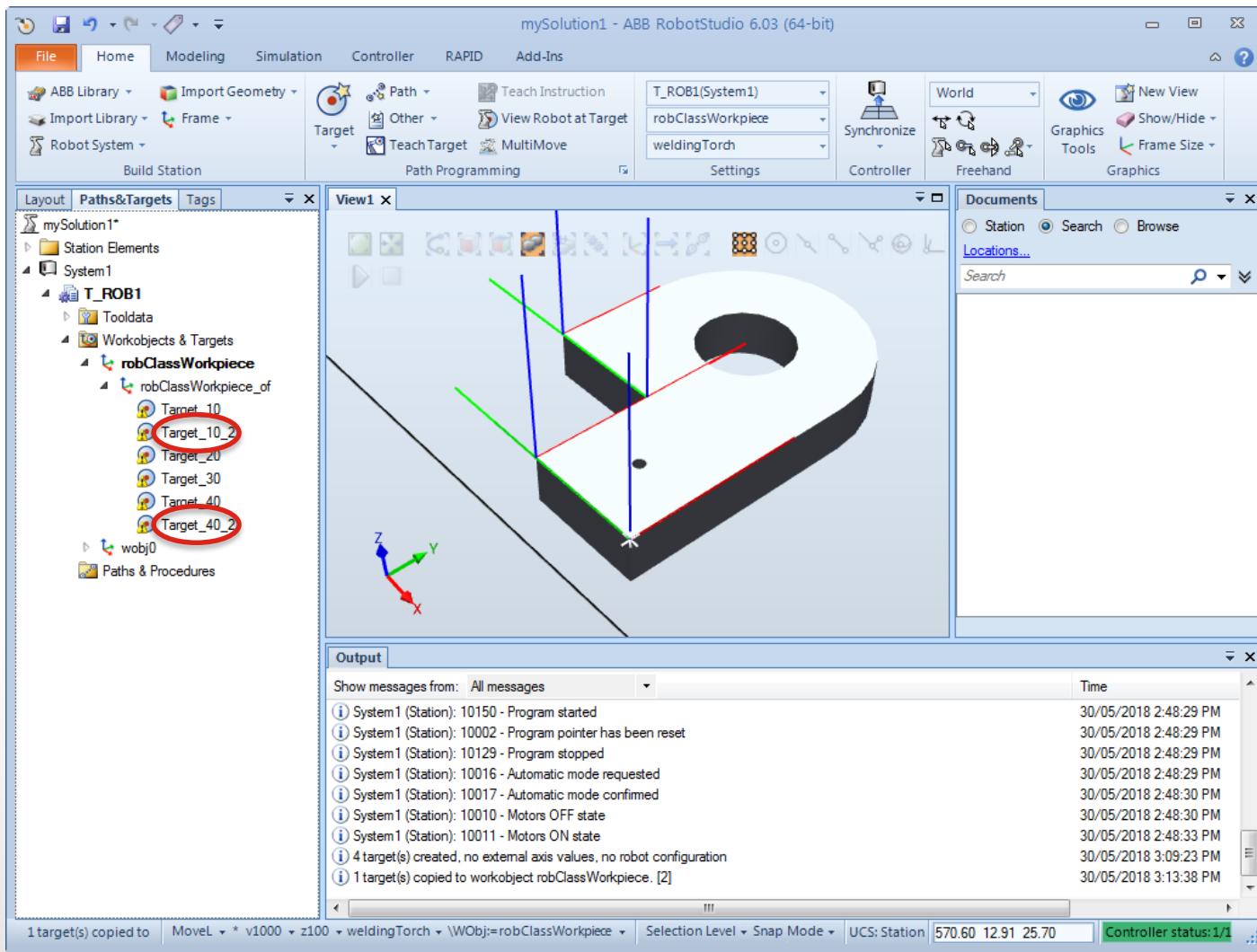
- We also need the starting point and the lift-off point. They are vertically

above
Point 1
and 4.



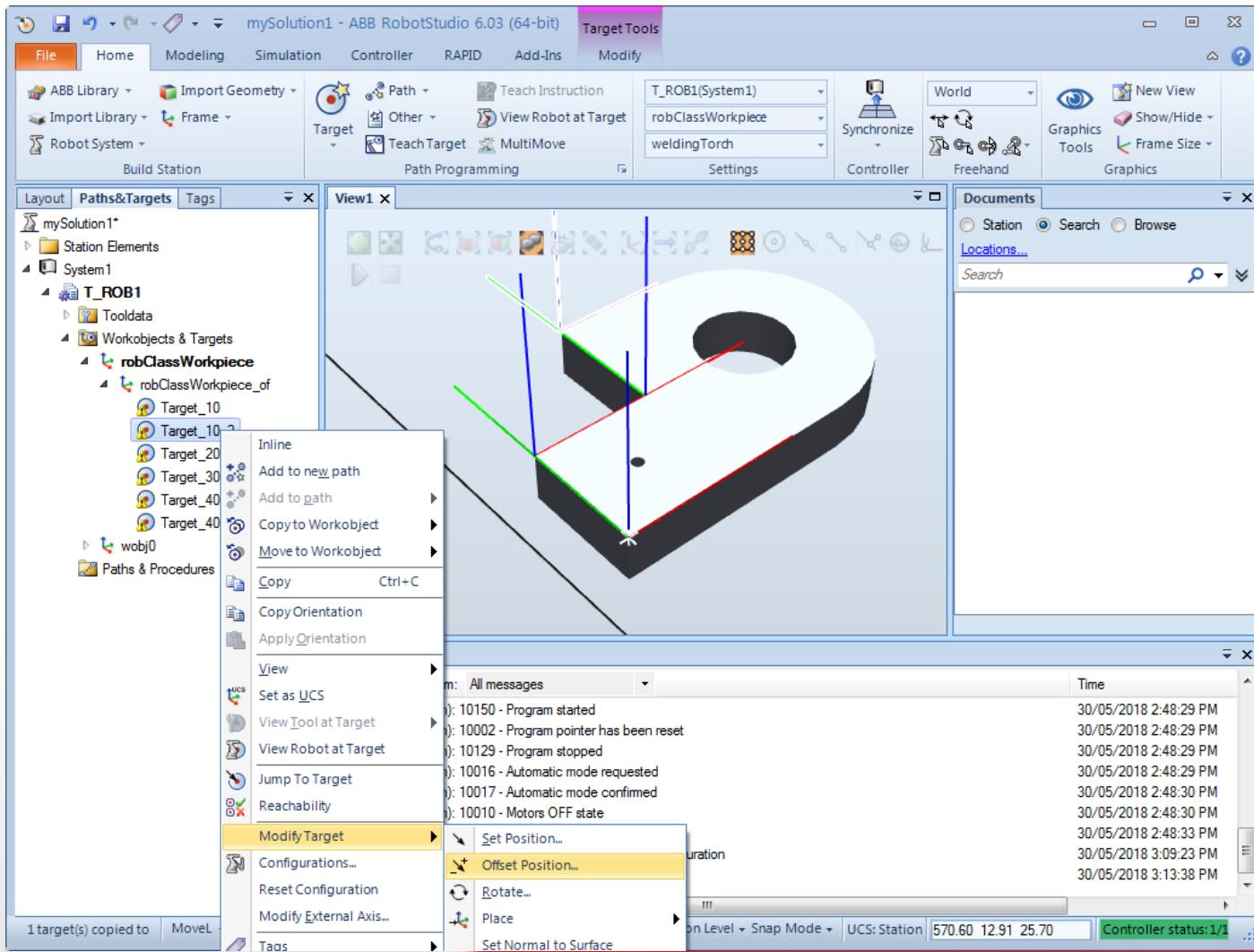
Create Targets

- Copy Target_10 and Target_40, and paste back under robClassWorkpiece.



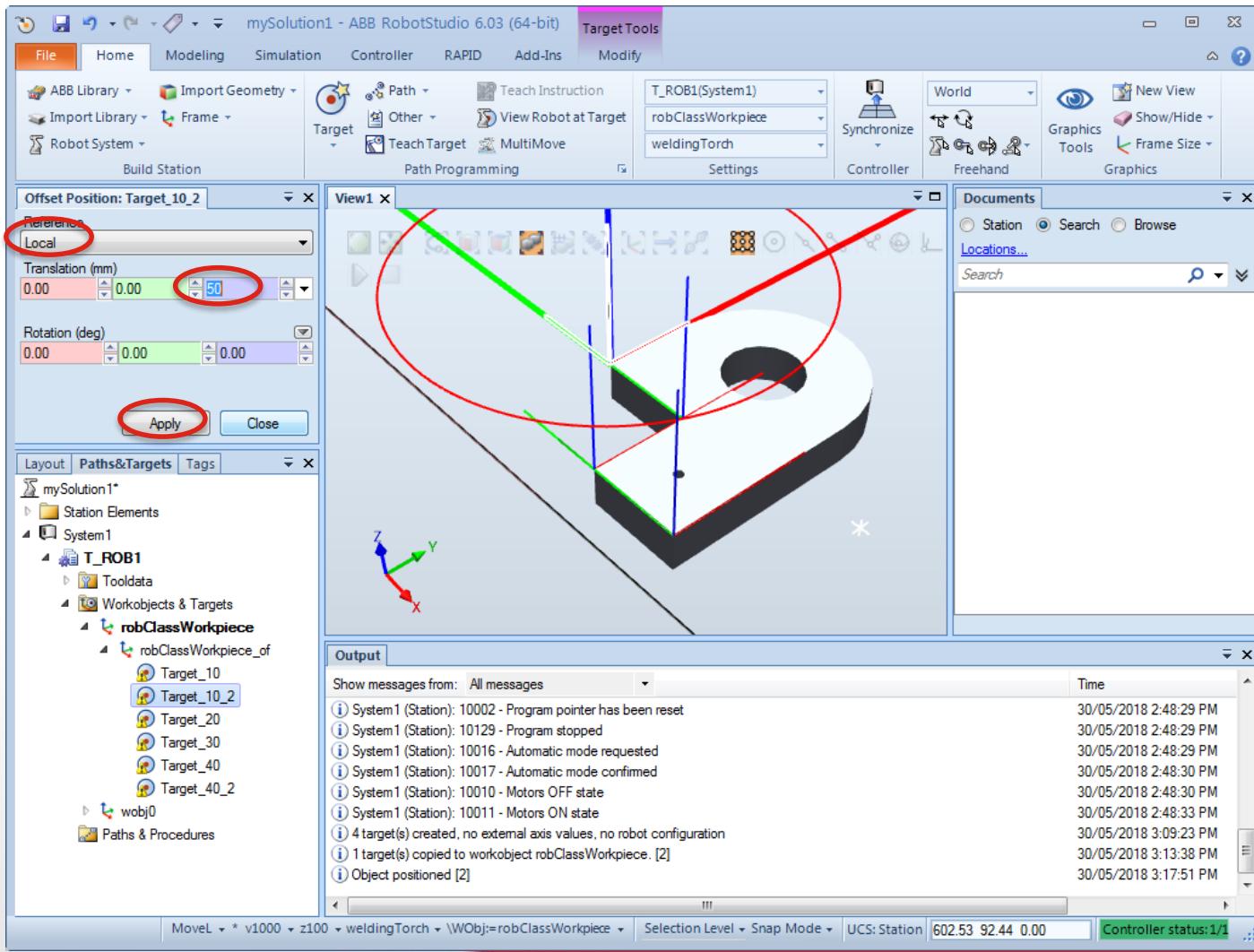
Create Targets

- Select Target_10_2 and then click “Offset Position” as shown.



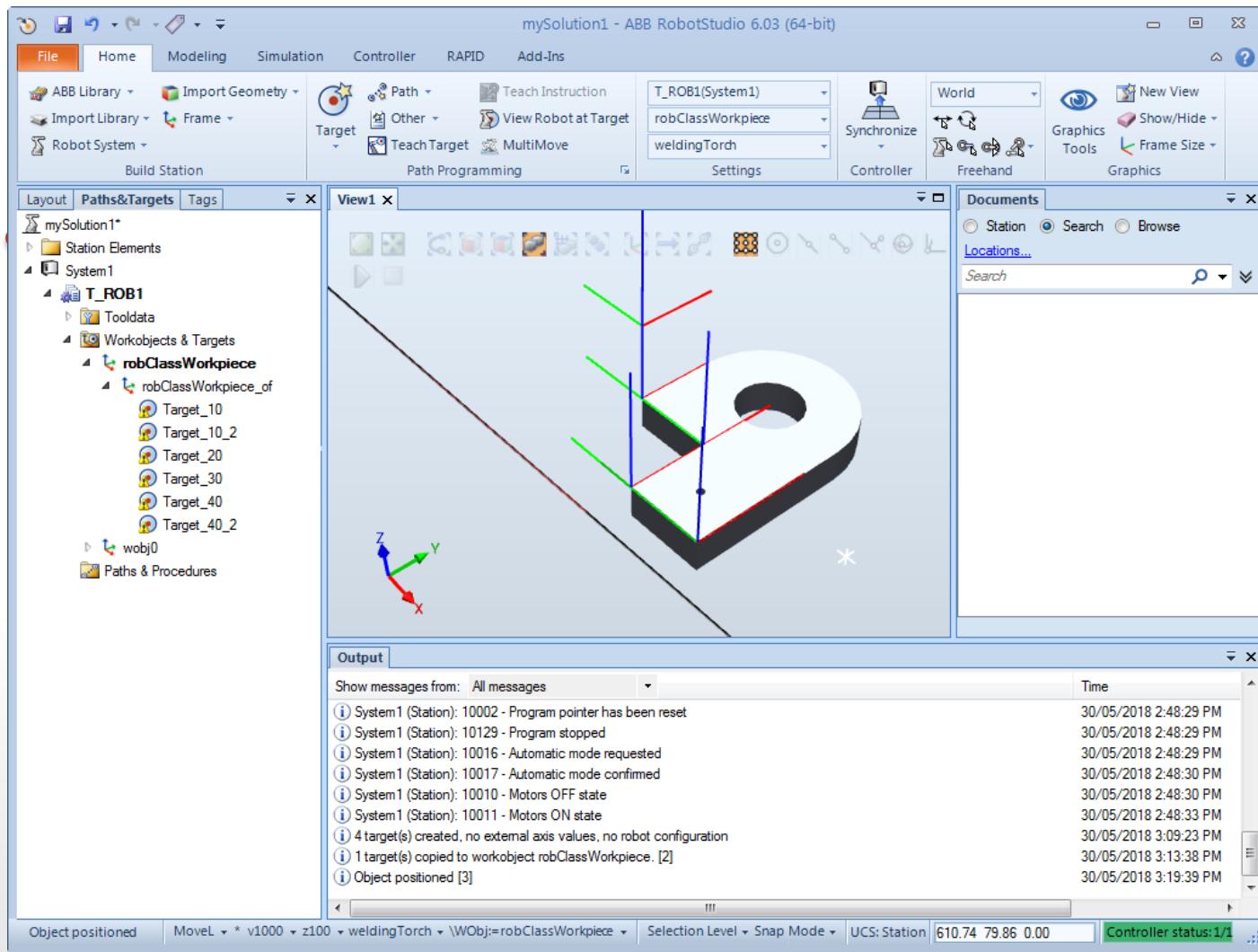
Create Targets

- Choose “Local” as reference, and key in value for offset. Finally click “Apply”.



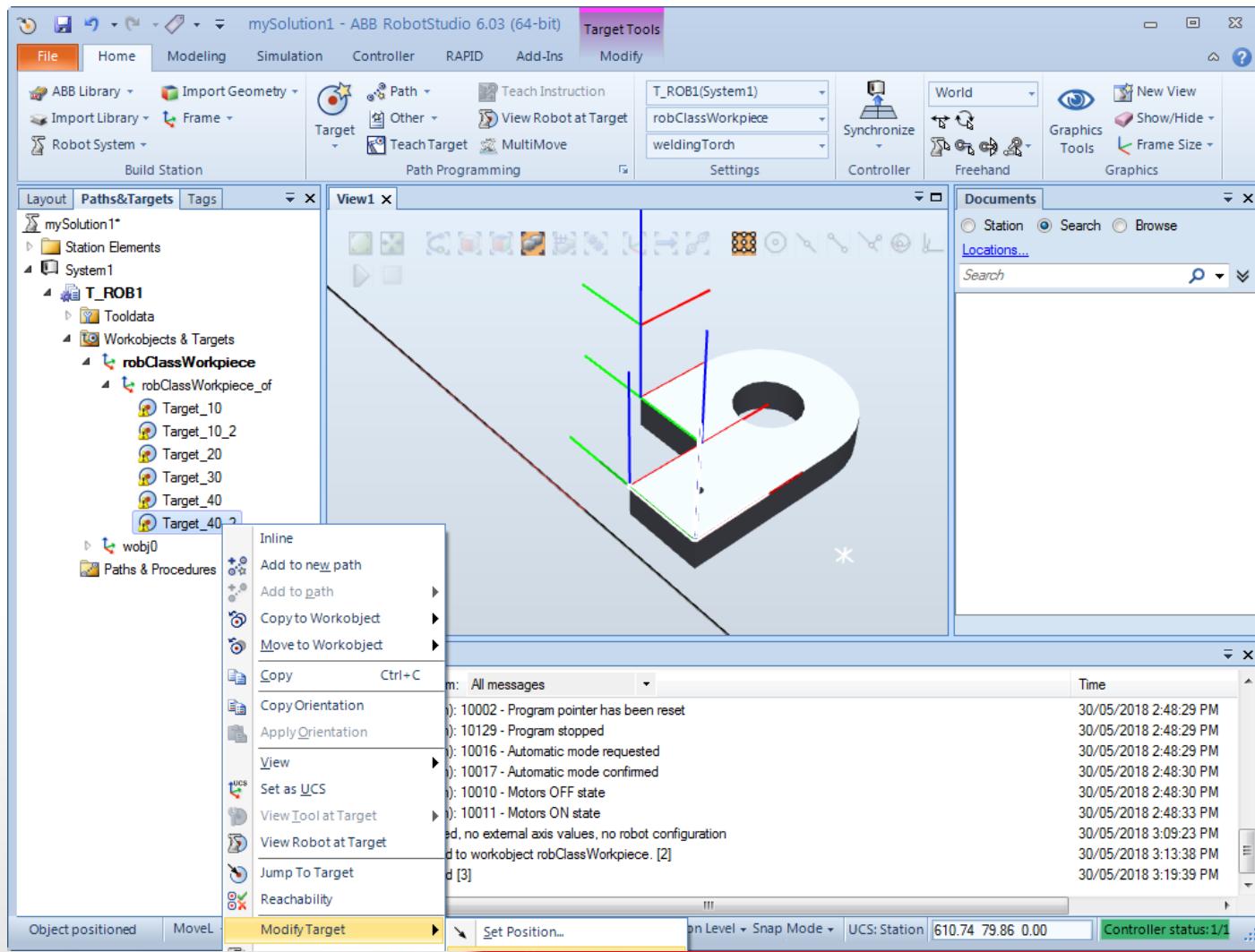
Create Targets

- Target_10_2 is now shifted above Target_10.



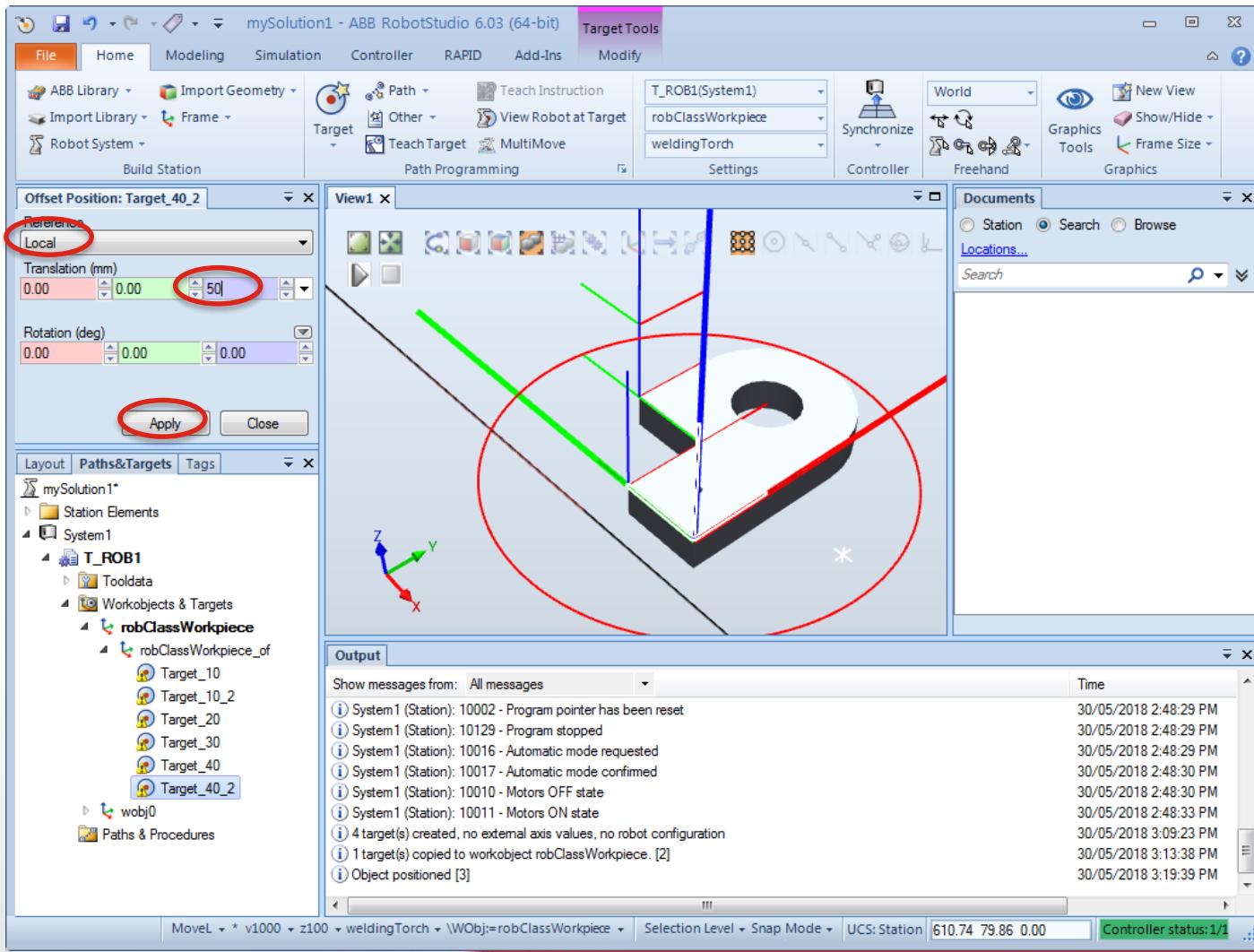
Create Targets

- Select Target_40_2 and then click “Offset Position” as shown.



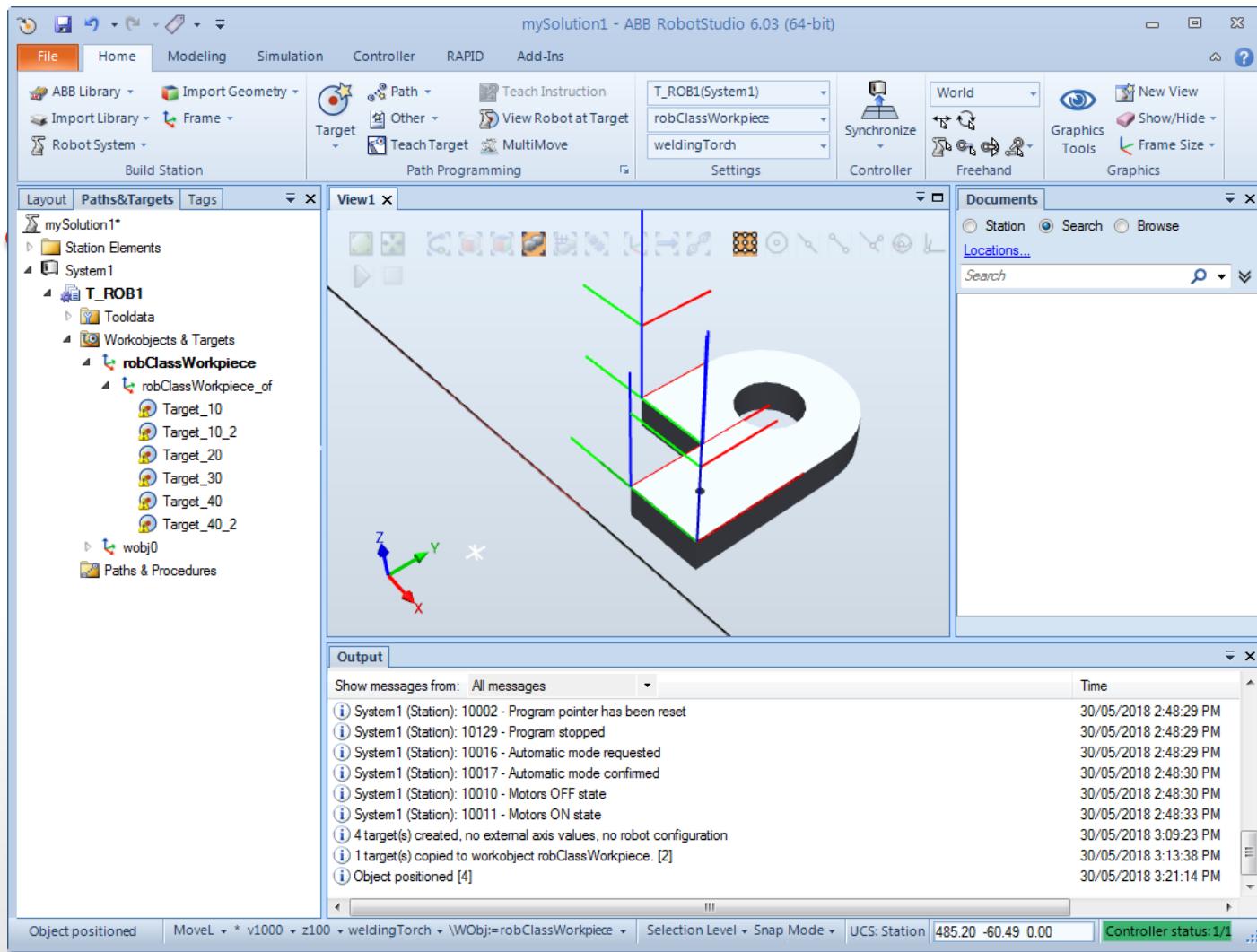
Create Targets

- Choose “Local” as reference, and key in value for offset. Finally click “Apply”.



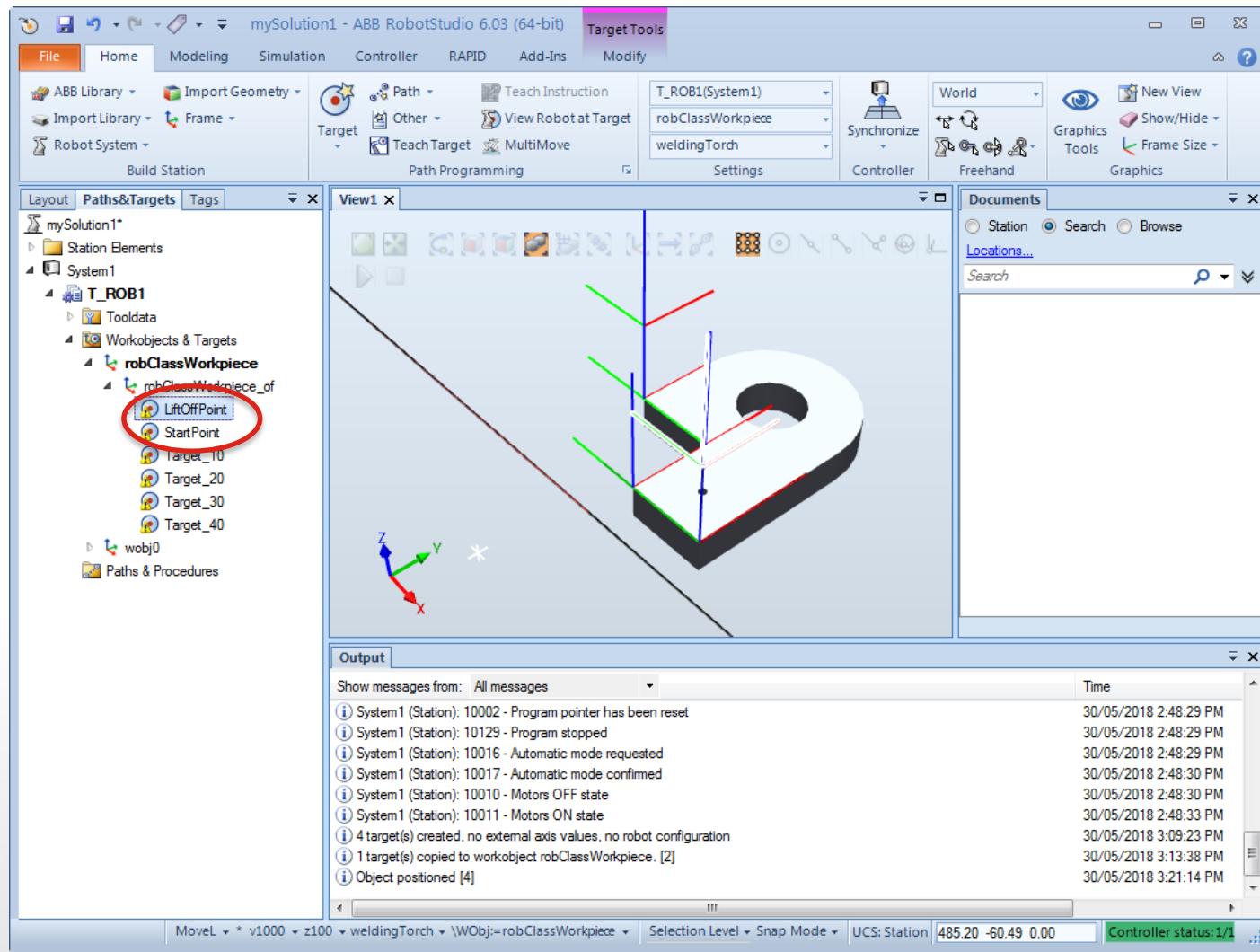
Create Targets

- Target_40_2 is now **shifted** above Target_40.



Create Targets

- Rename Target_10_2 & Target_40_2 as **StartPoint** & **LiftOffPoint** respectively.

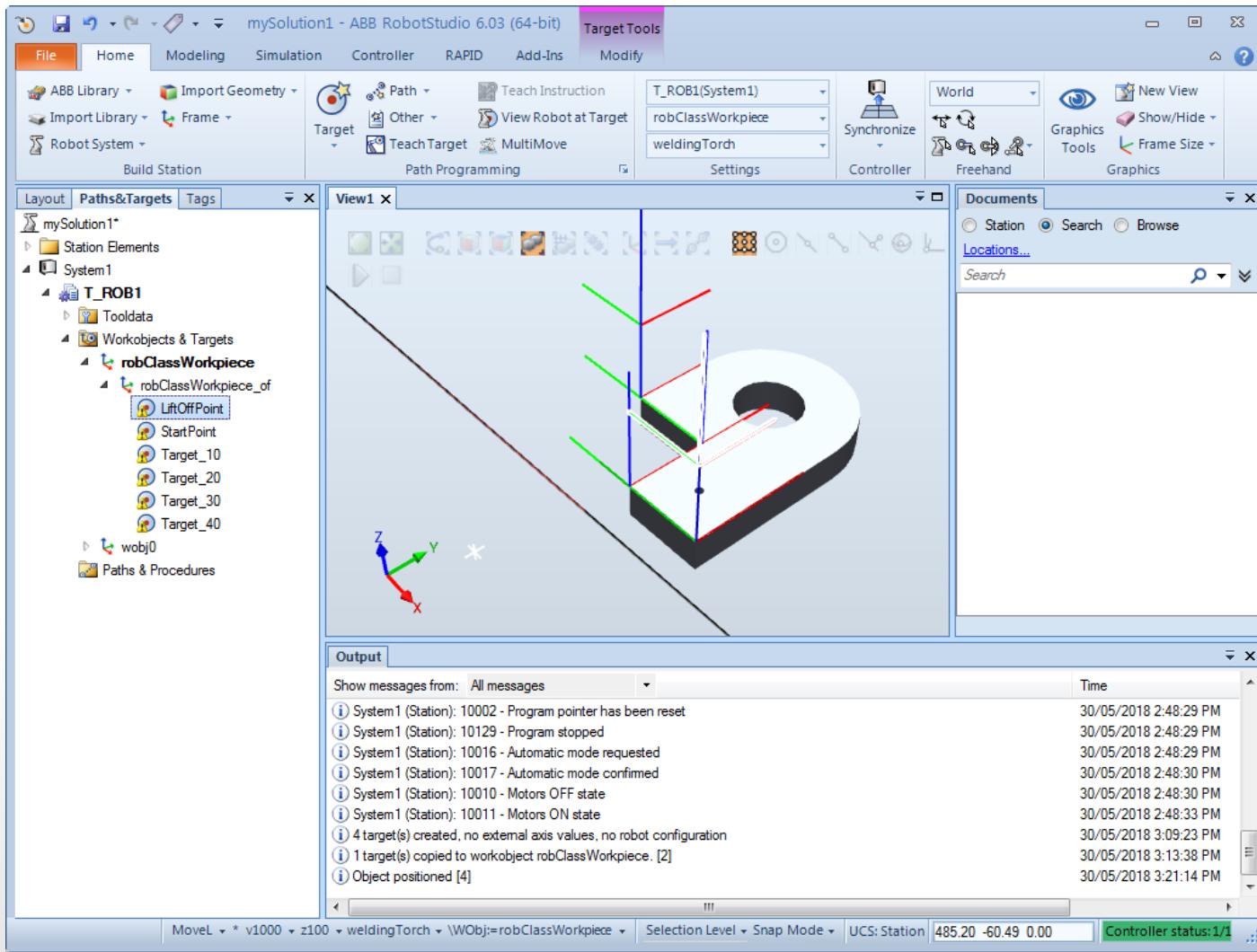


Content

- Open Solution from Previous Tutorial
- Create Targets
- **Reorient Targets**
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

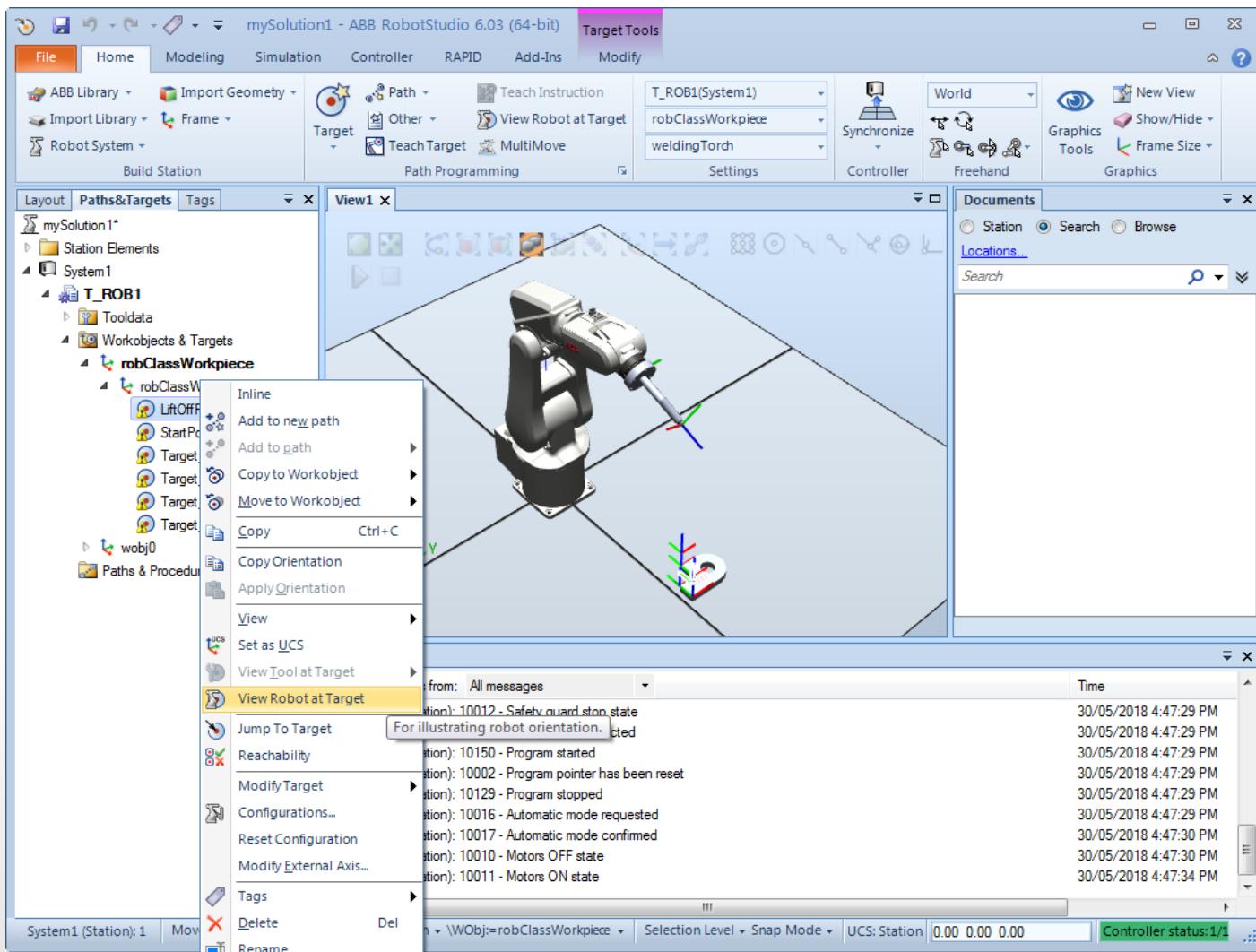
View Robot at Target

- To make sure points are **reachable**, we can view the robots at the targets.



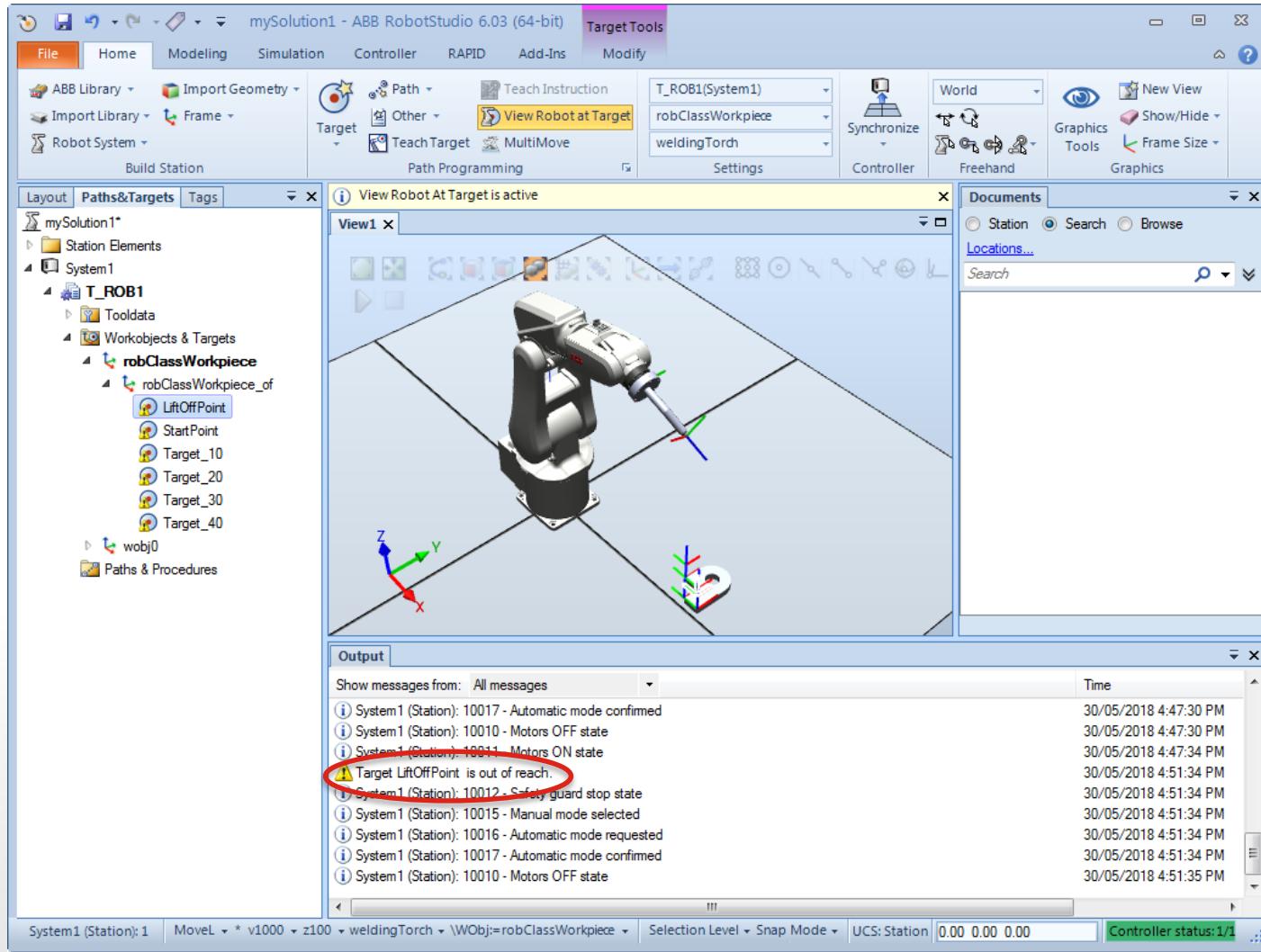
View Robot at Target

- Under “Path & Target” browser, right click on any target → **View Robot at Target**



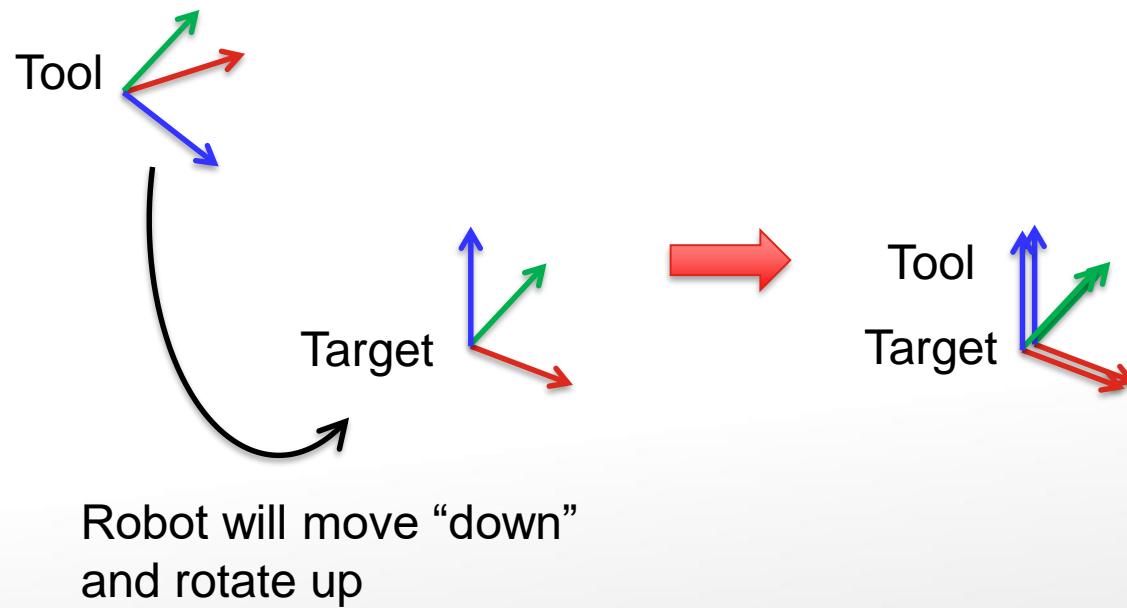
View Robot at Target

- There is an error message stating that the target is **out of reach**. But why?



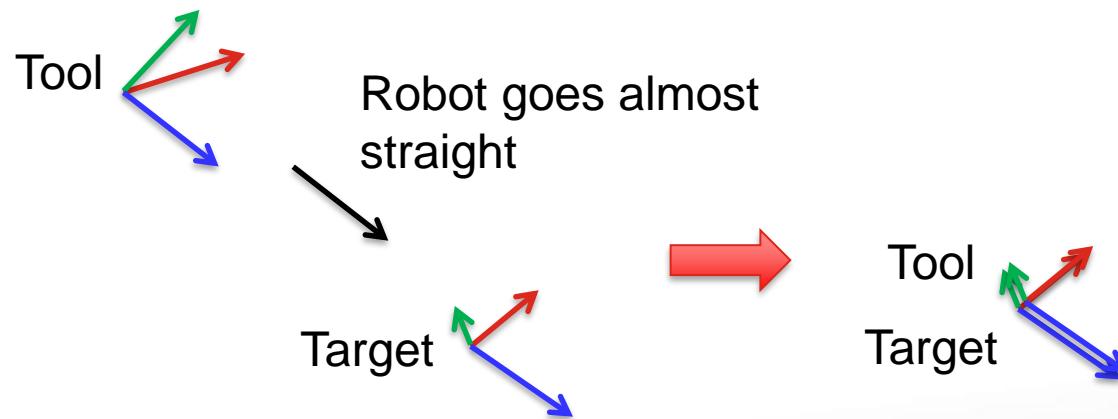
Reorient Target

- The reason is that the robot will try to move in a way such that the **tool frame** will **match the target point frames**.



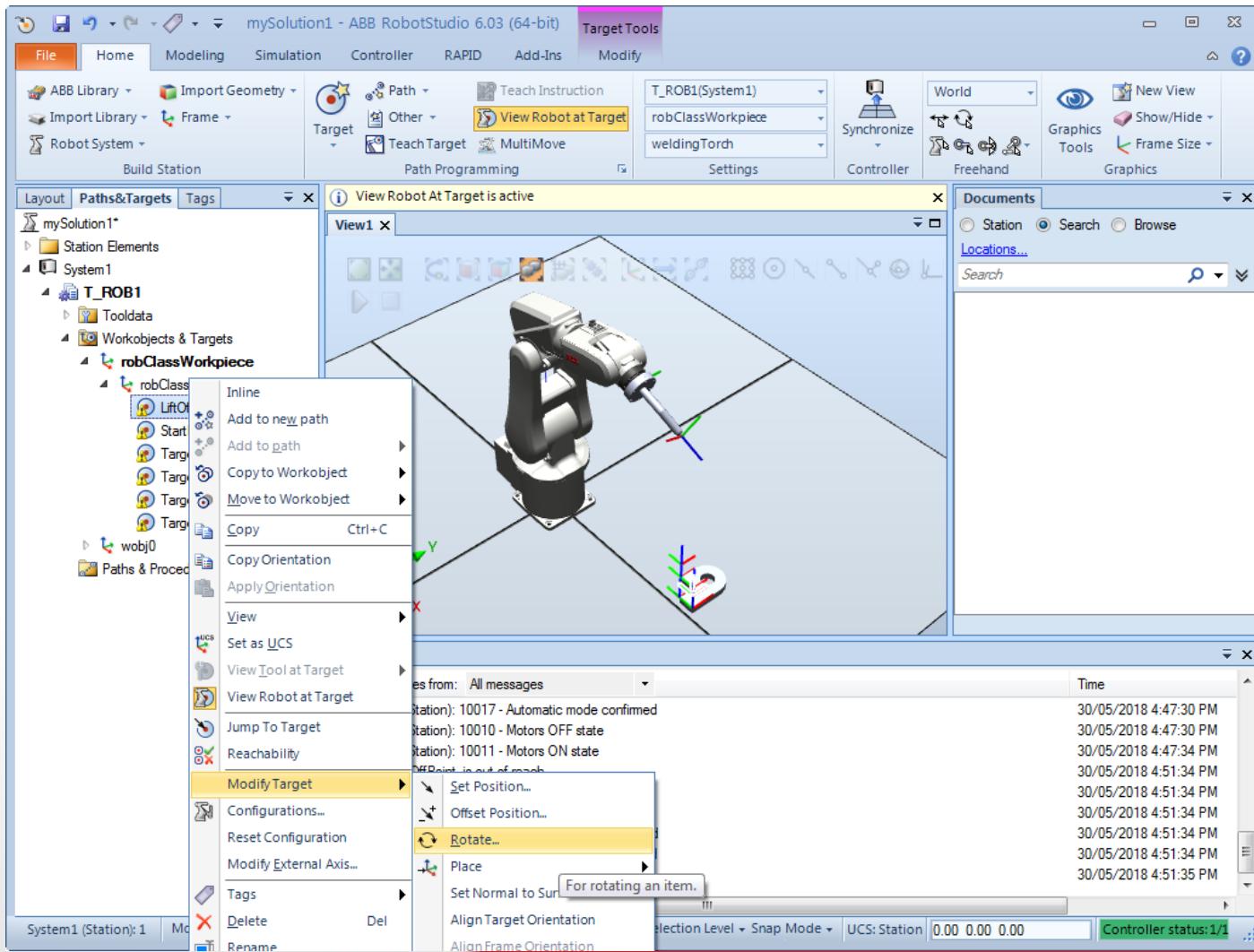
Reorient Target

- If we can **reorient the target point**, then the robot has higher likelihood of reaching the point.



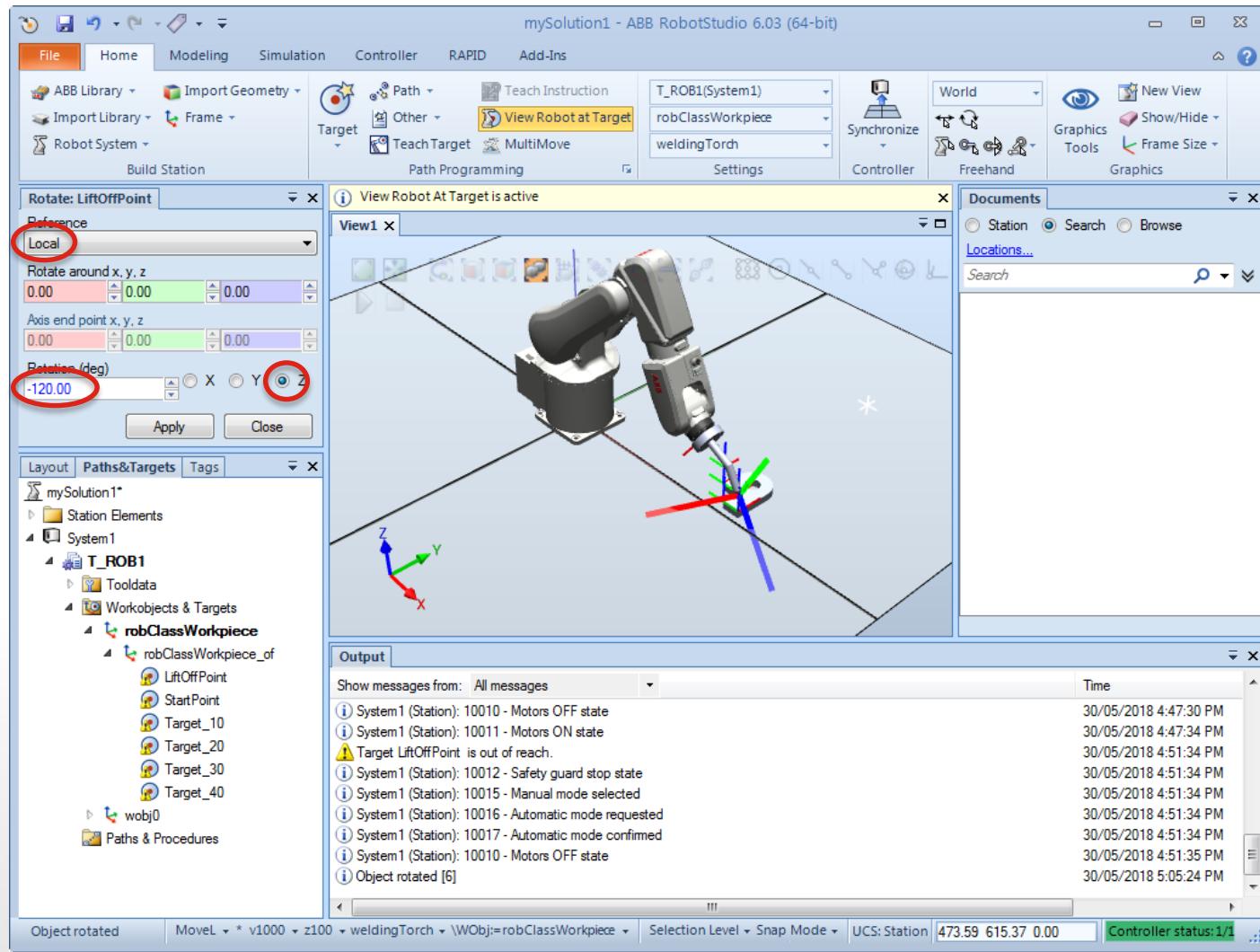
Reorient Target

- To reorient target, right click any target and choose modify target → Rotate.



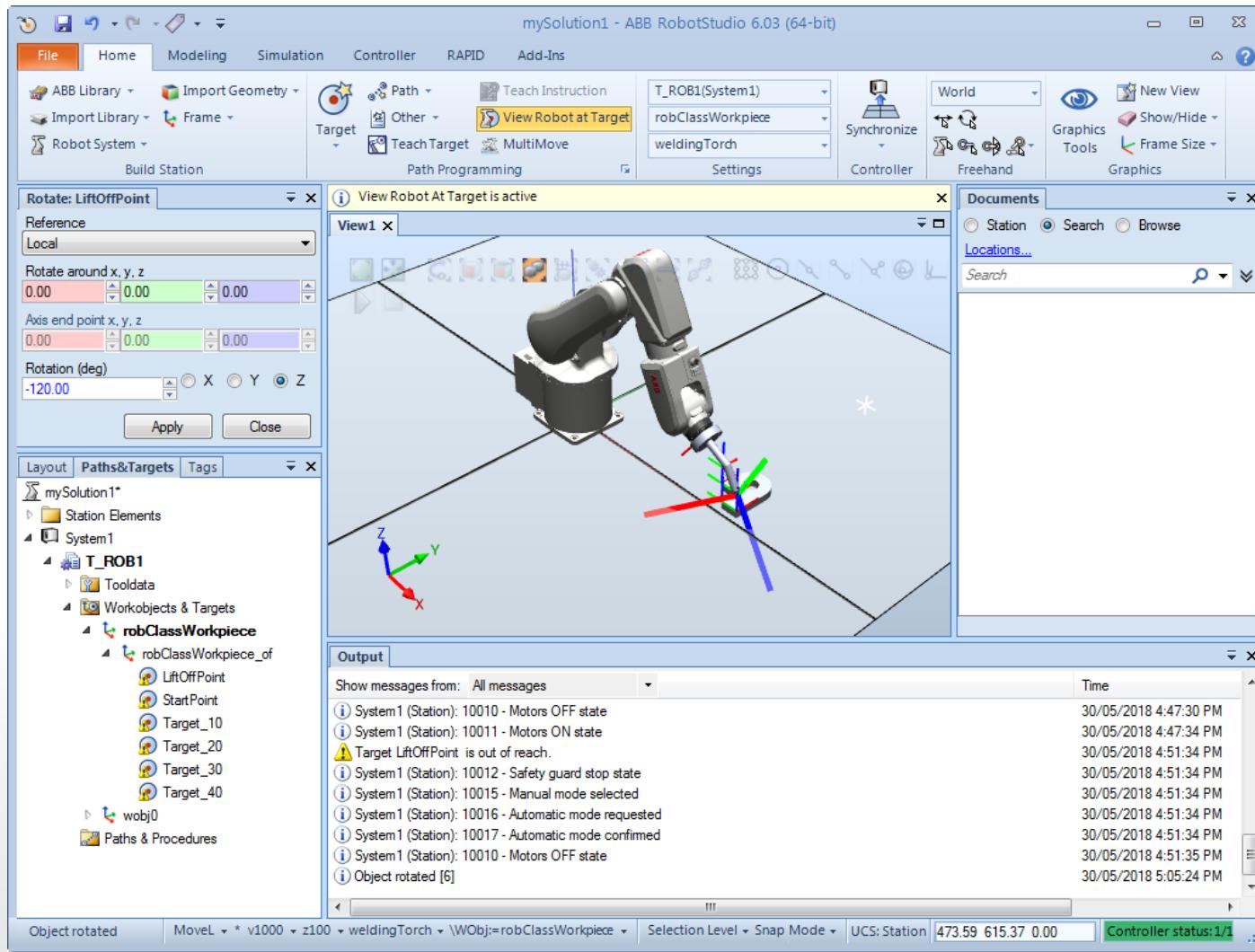
Reorient Target

- Rotate locally along x-axis: 160deg, then z-axis: -120deg



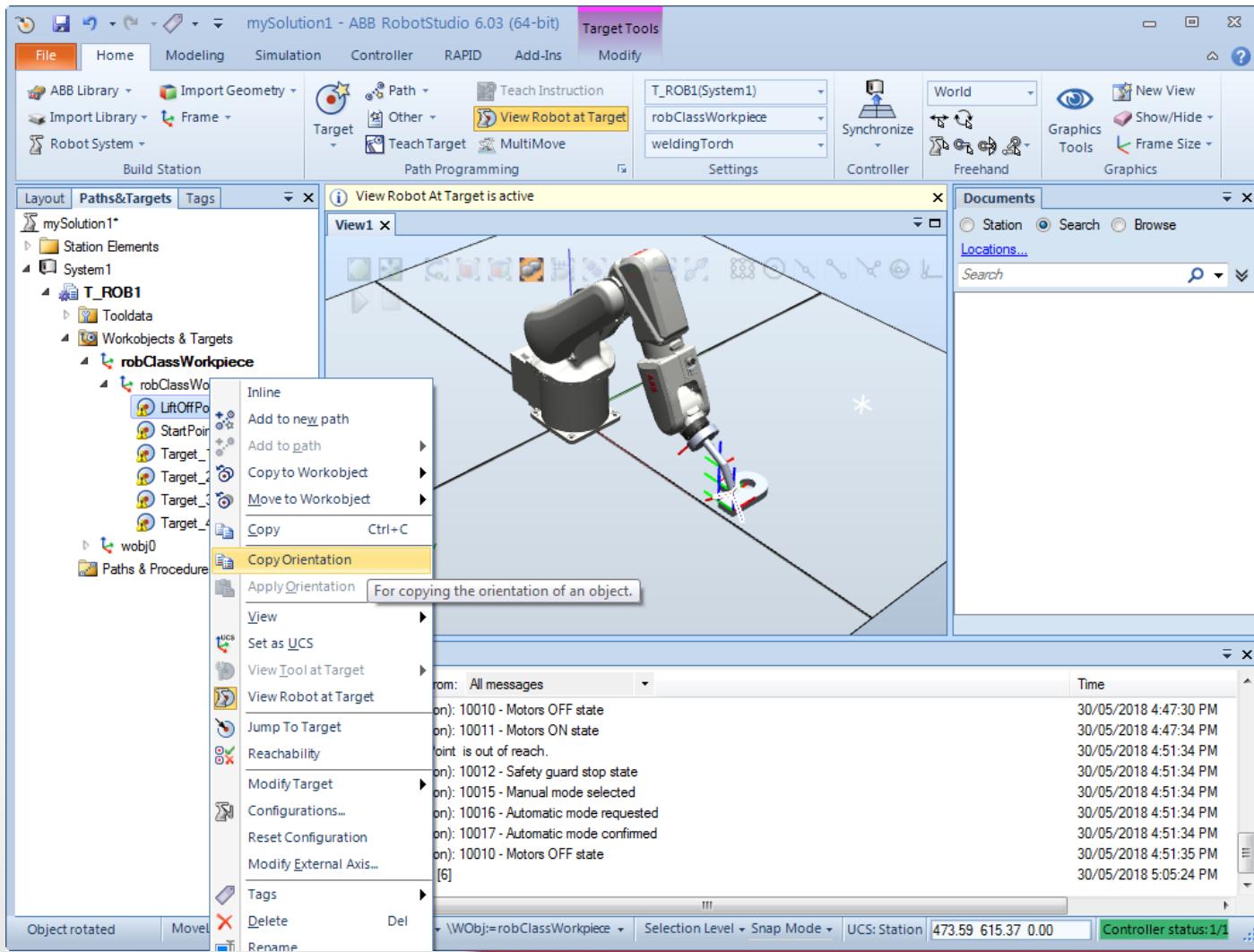
Reorient Target

- We now see that the robot can reach the target point.



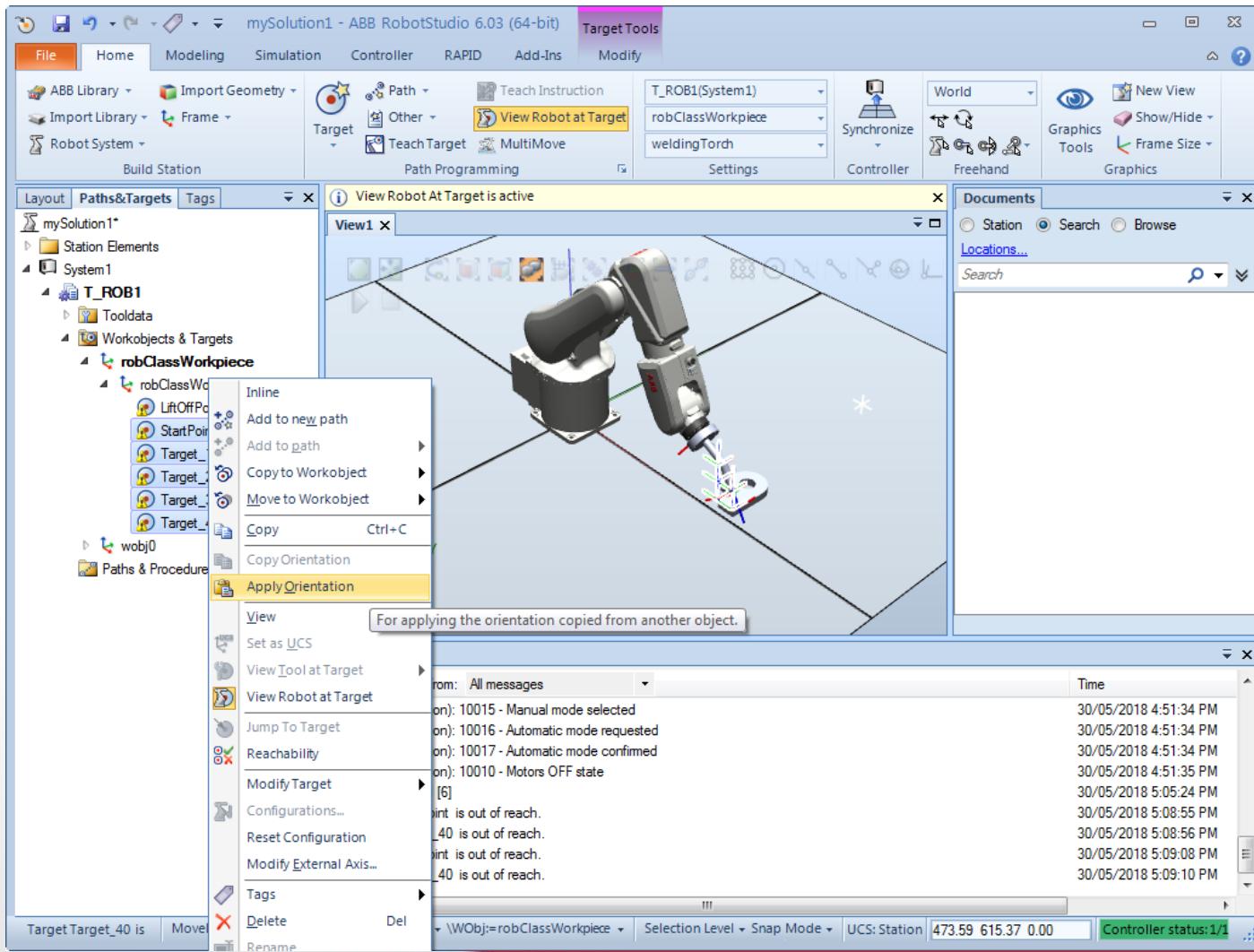
Reorient Target

- Right click on the reoriented target point → Copy orientation.



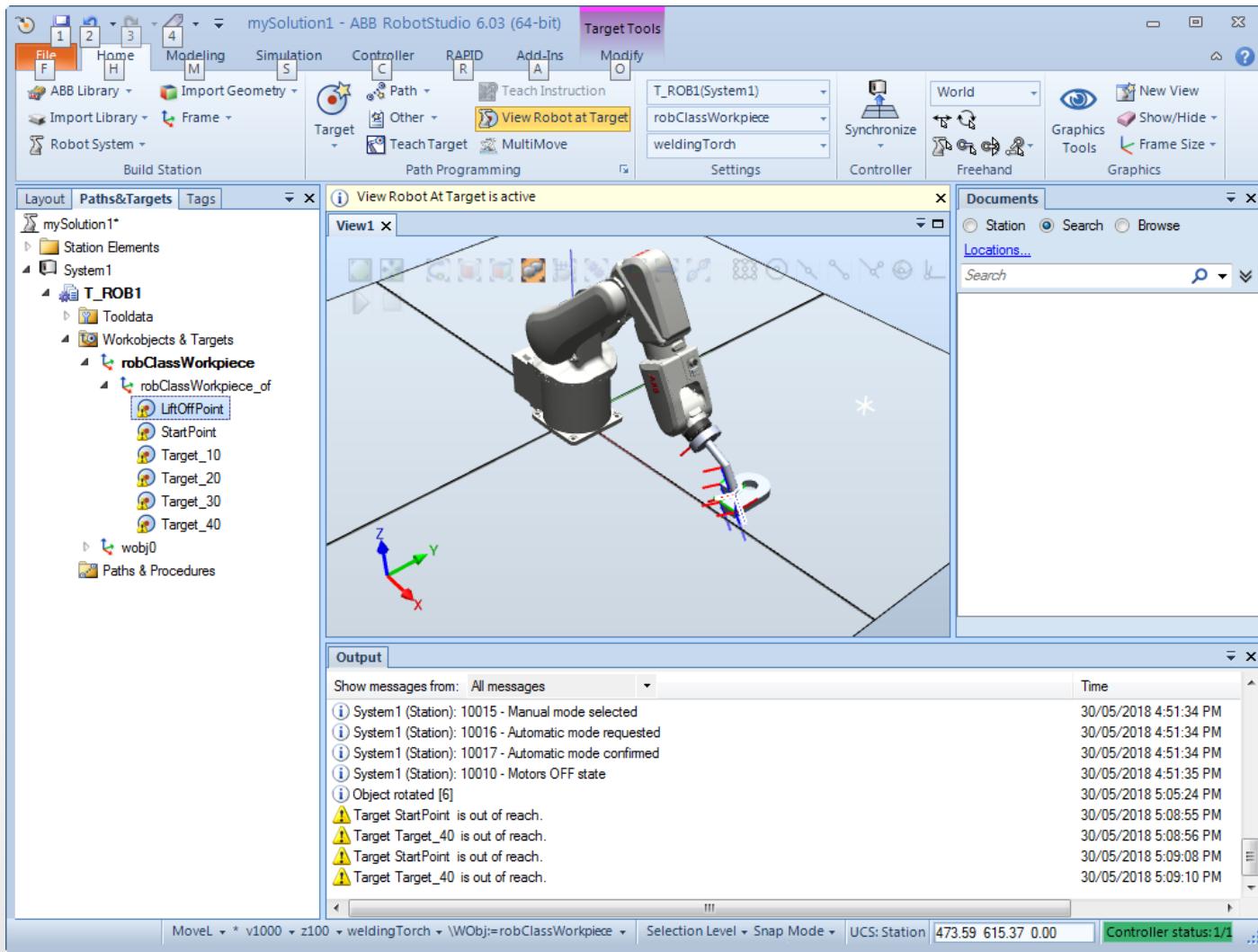
Reorient Target

- Multi-select all other points → Apply orientation.



Reorient Target

- Now all the target points have the same orientations.



Content

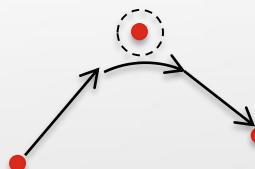
- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- **Set Instruction Template**
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

Set Instruction Template

- We already have the **target points**.
- Next, we would like to **create a path** to connect the target points, for the **robot to follow**.
 - A path in RobotStudio is a **holder for motion instructions**.
- But before we do that, we should set the **instruction template**.
 - The default motion type (Joint or Linear).
 - The default speed.
 - The default zone
 - Zone specifies how accurate the points should be reached.
 - High accuracy requirement also means more time consumed.
 - Trade off.



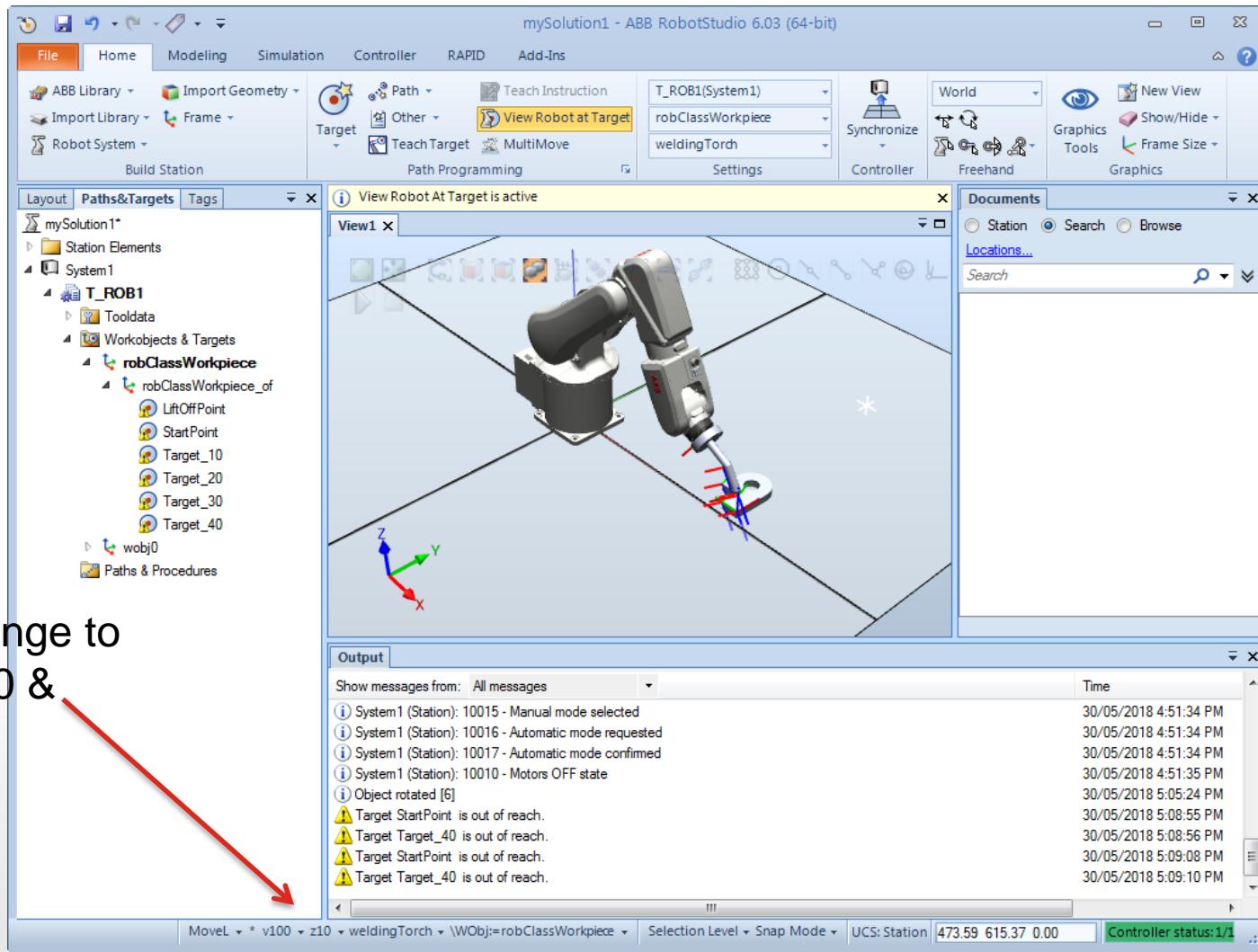
Less accurate
but faster



More accurate
but slower

Set Instruction Template

- At the bottom of RobotStudio window, you can change the default values.

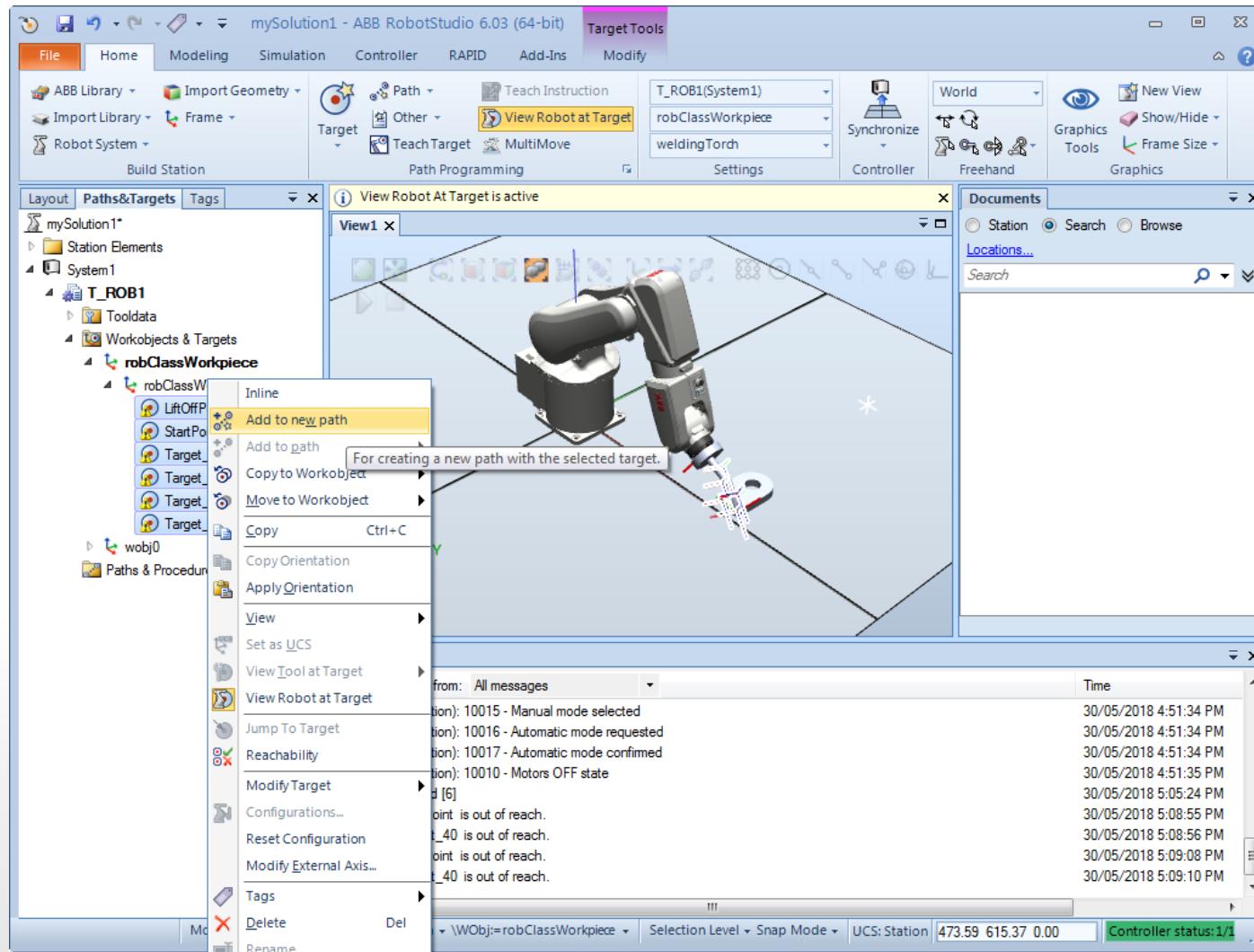


Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- **Add Instruction to New Path**
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

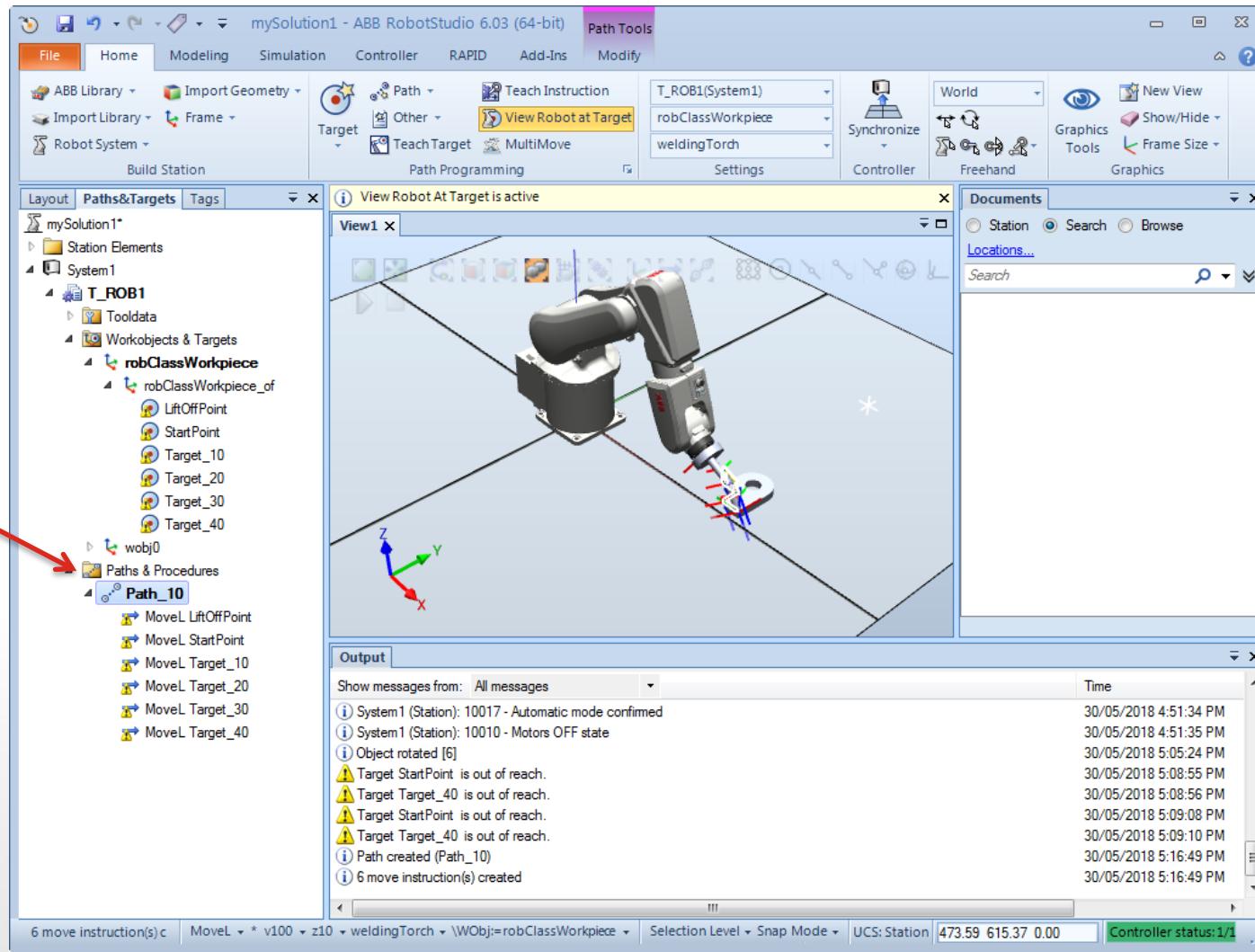
Add Instruction to New Path

- Multi-select all targets. Right click and choose “Add to New Path”.



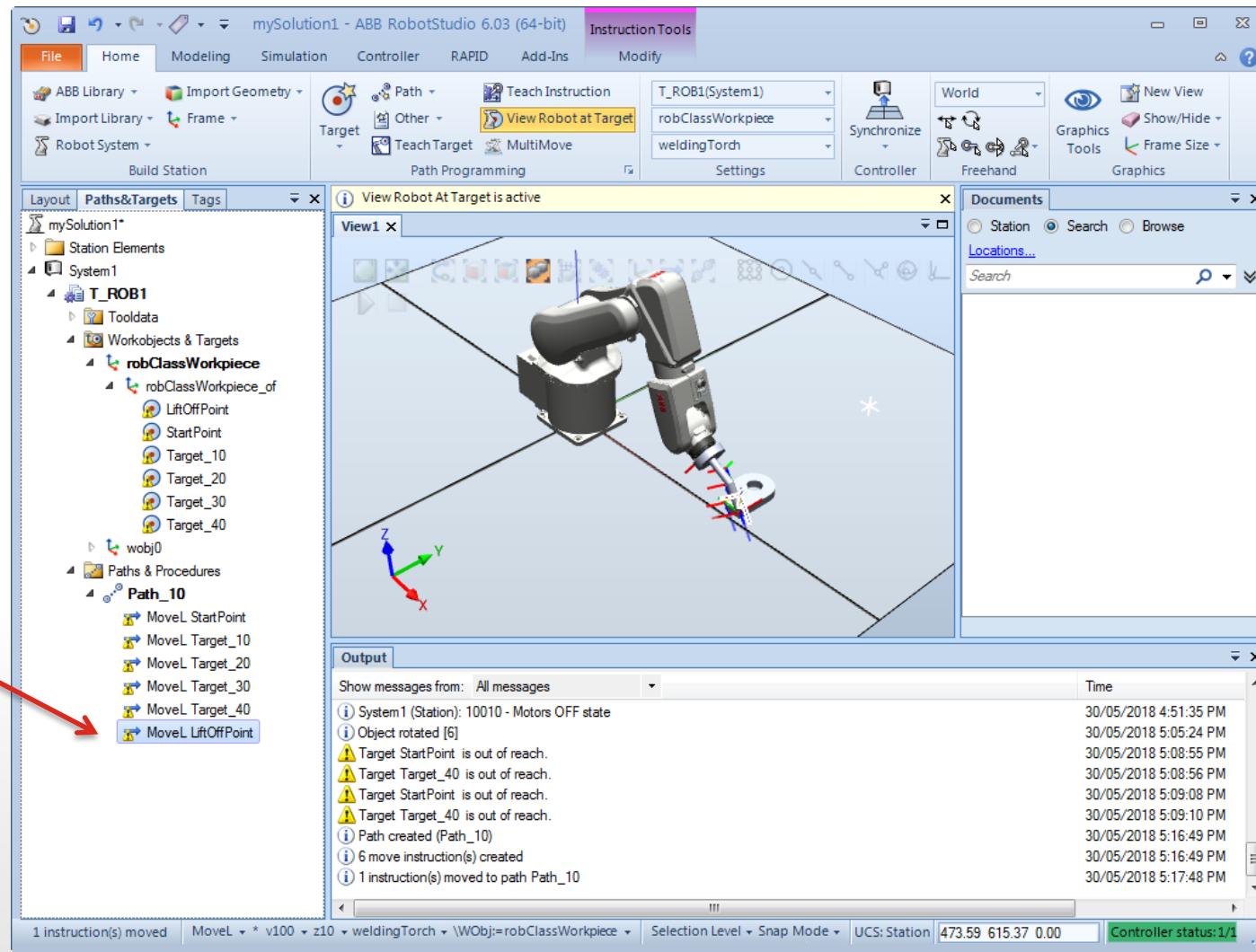
Add Instruction to New Path

- A path is created with instruction set from the instruction template.



Add Instruction to New Path

- Drag MoveL LiftOffPoint to the last position.

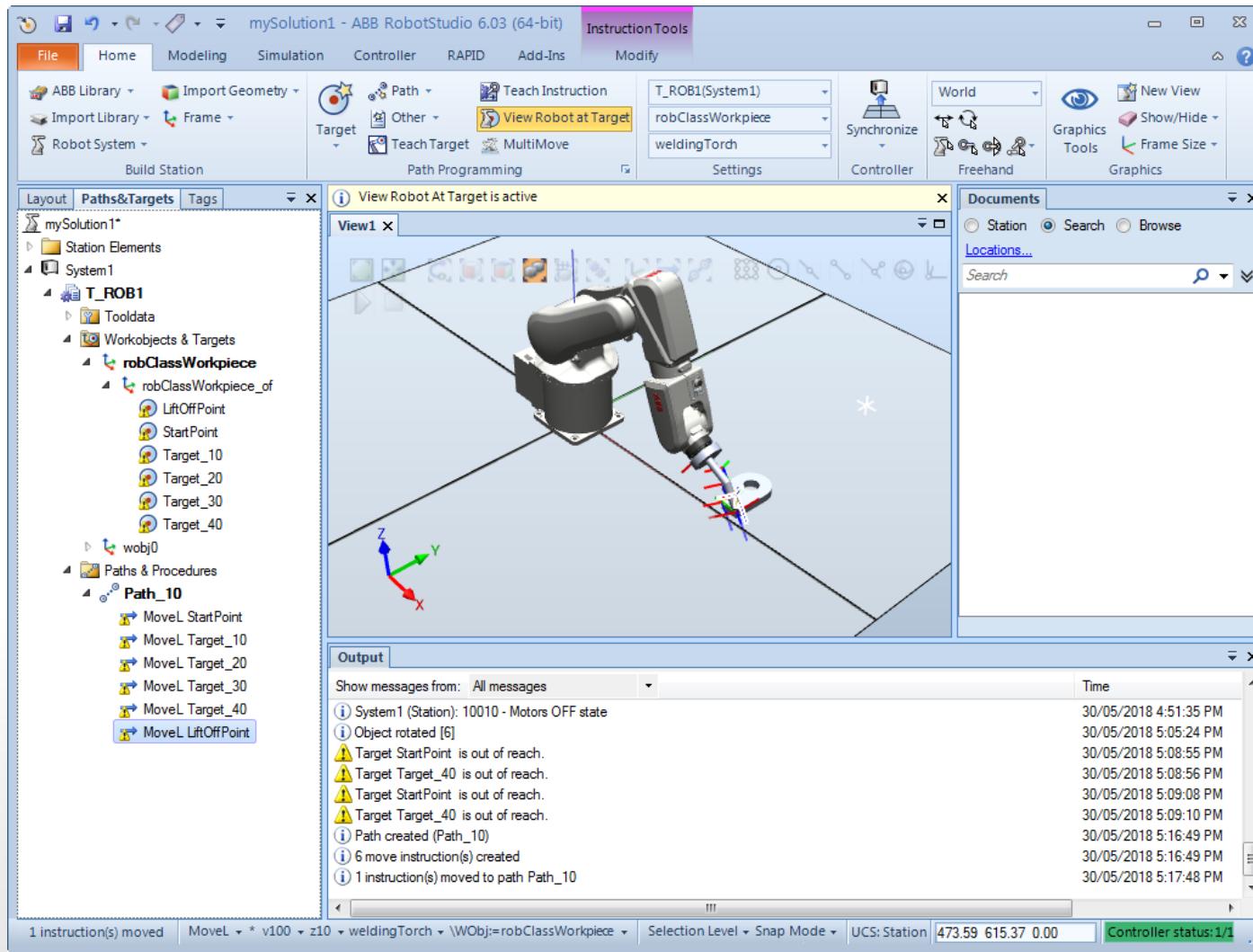


Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- **Modify Instruction**
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

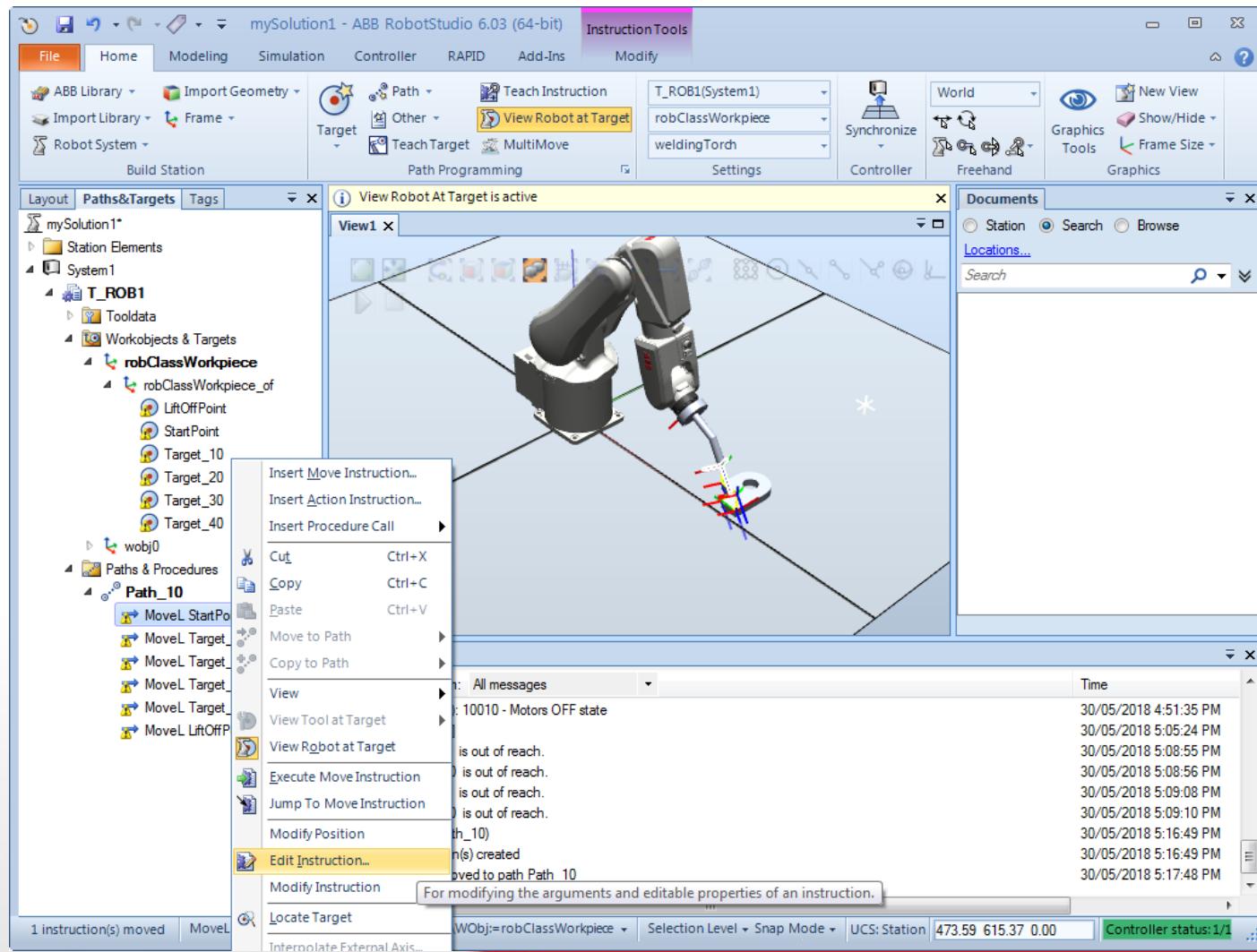
Modify Instruction

- It is common to have joint motion to first target point, instead of linear.



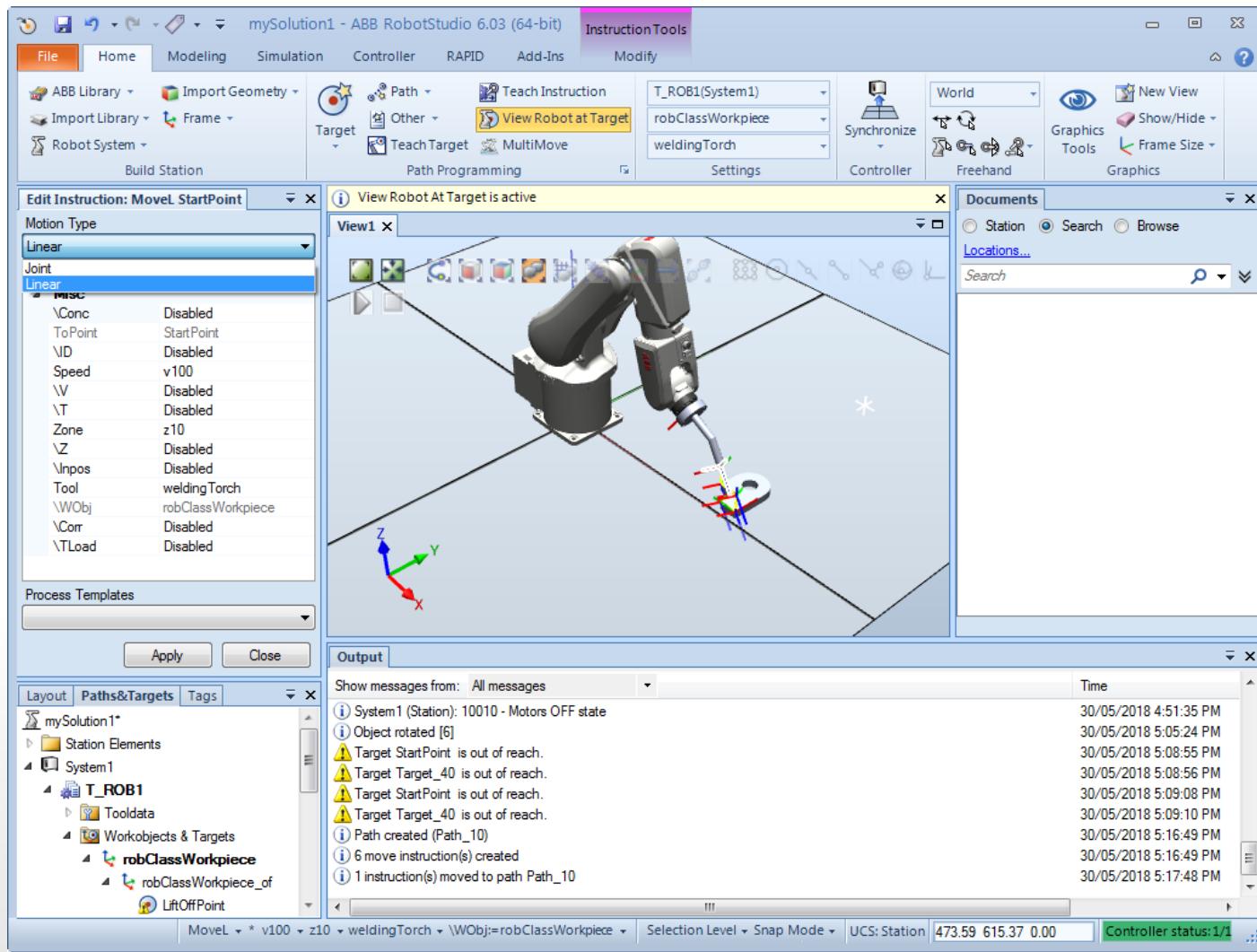
Modify Instruction

- Right click 1st instruction → Edit Instruction.



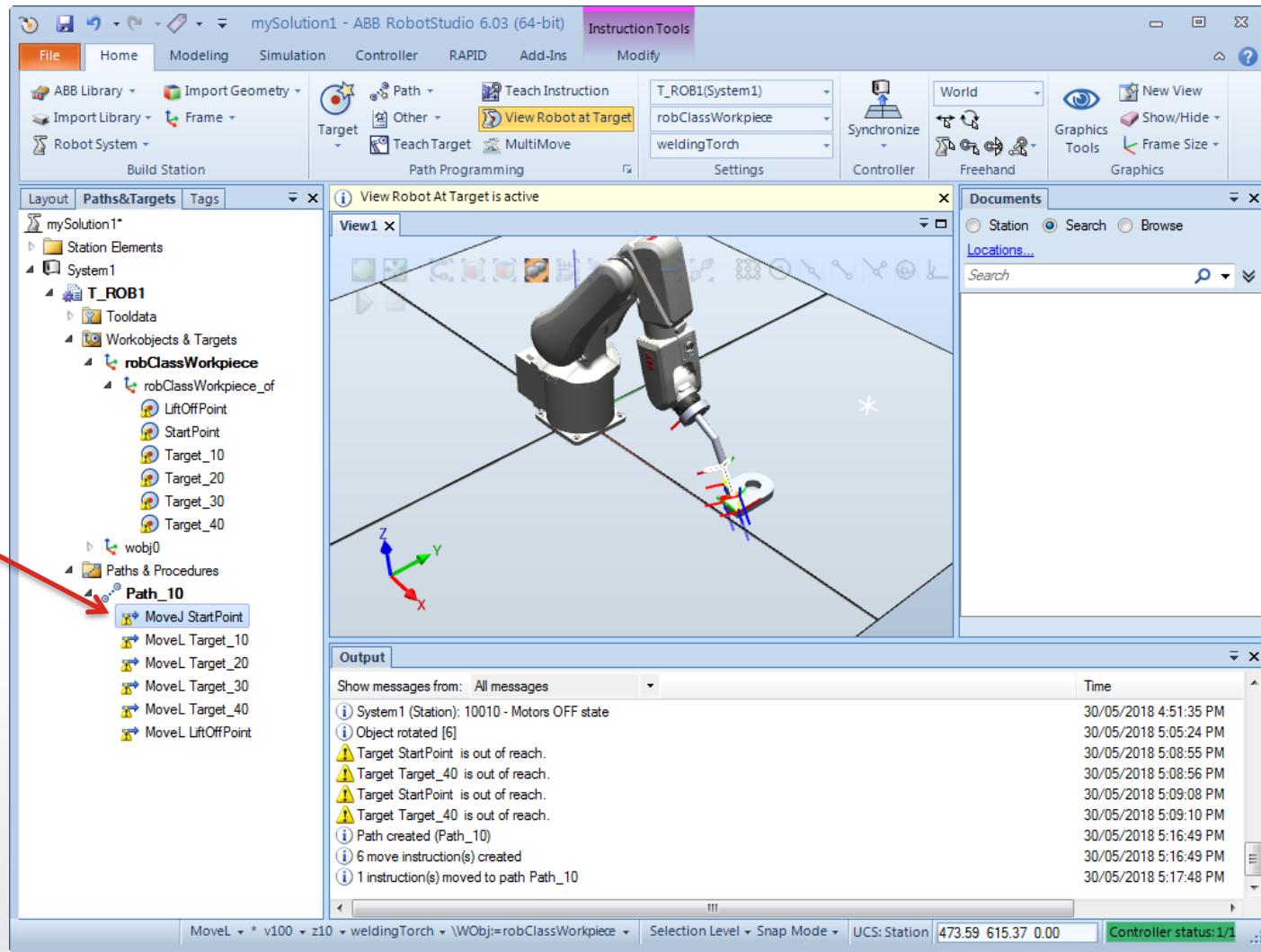
Modify Instruction

- Under the motion type, choose “Joint” → Apply → Close.



Modify Instruction

- The motion type is now changed to MoveJ.



Content

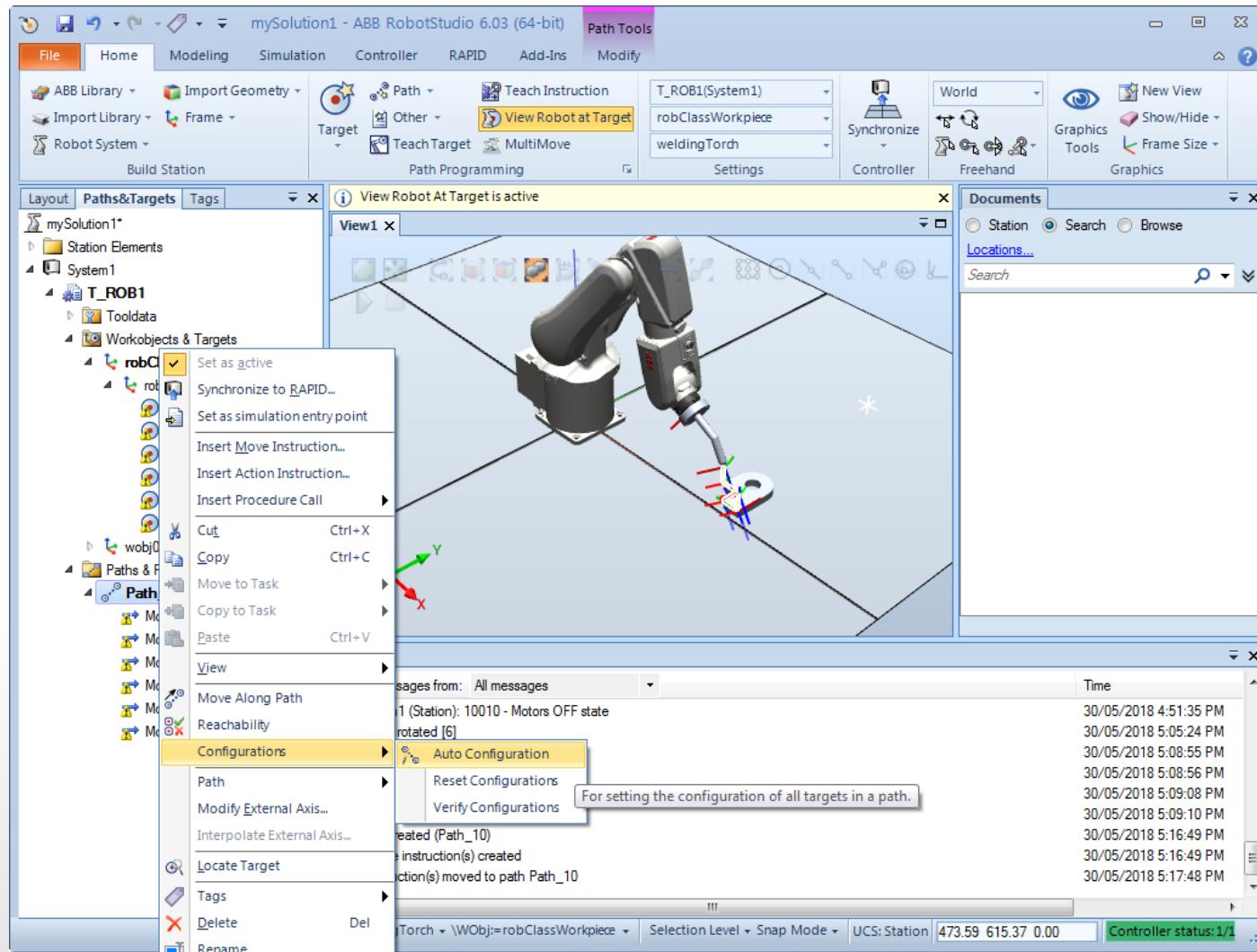
- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

Auto Configuration

- We now need to **configure the robot** to reach all the target points.
 - Configure means to find the solutions (elbow-up, elbow-down, joints 4/5/6 flipped etc.).
 - We should choose a **good starting configuration**, with most joints close to zero degree.

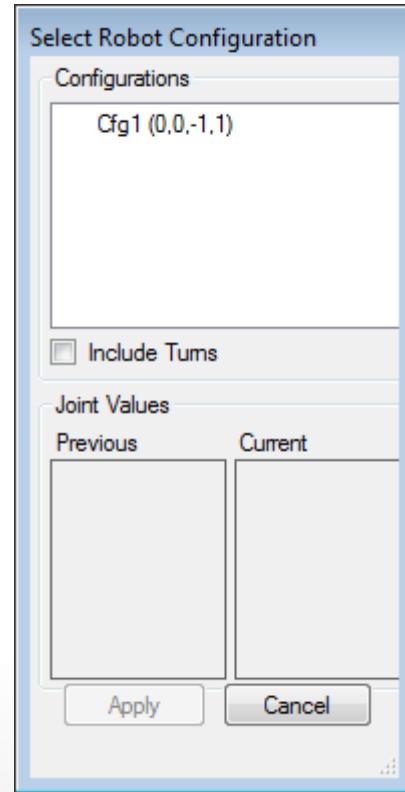
Auto Configuration

- Right click Path_10 → Configuration → Auto configuration.



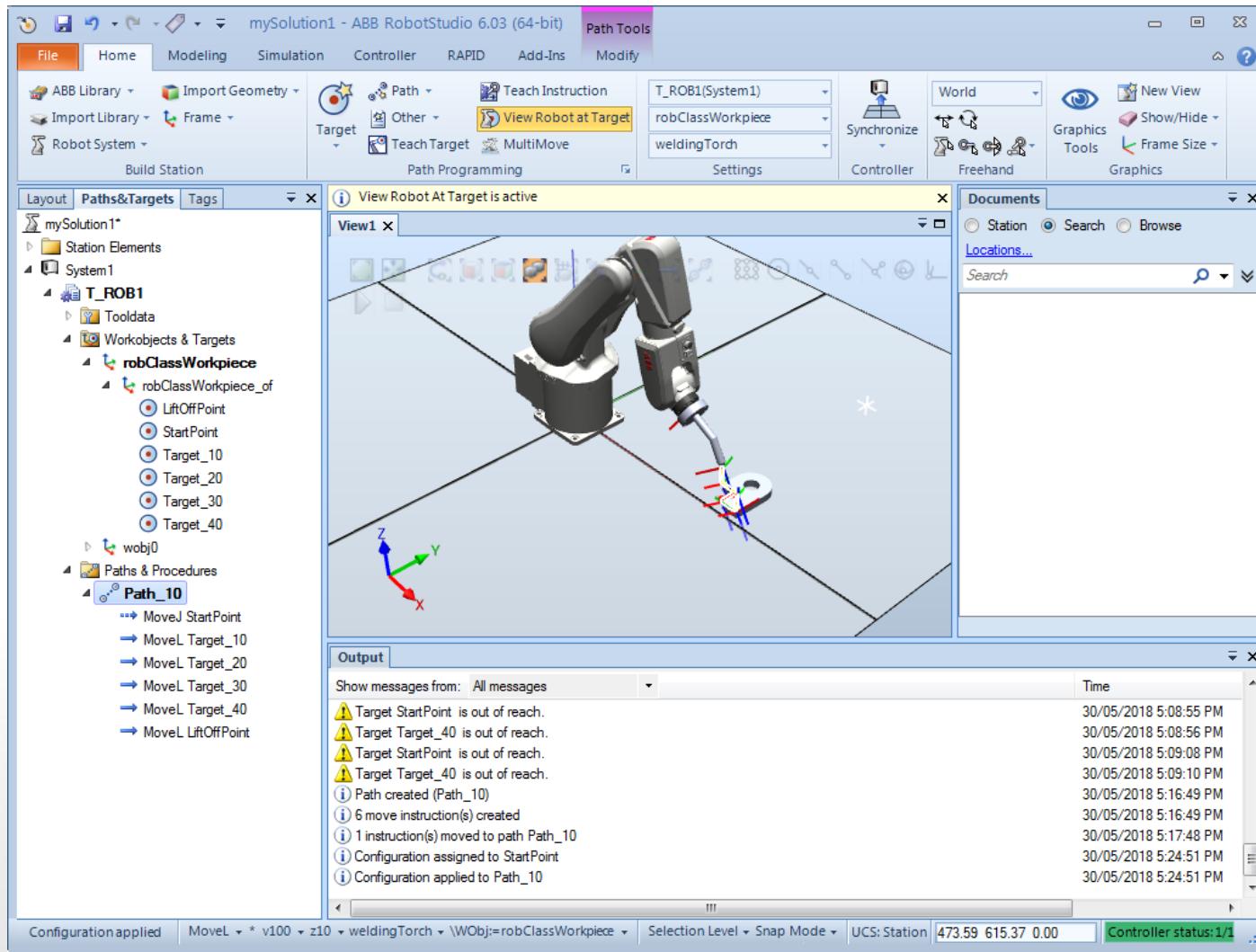
Auto Configuration

- In this case, only one configuration is found.
 - Click on Cfg1 and then hit **Apply**.
- Sometimes, there may be more options.
 - Click on those solutions and see the illustration.
 - Choose the best one based on your judgement.



Auto Configuration

- Now, we see that the robot passes through all the points.

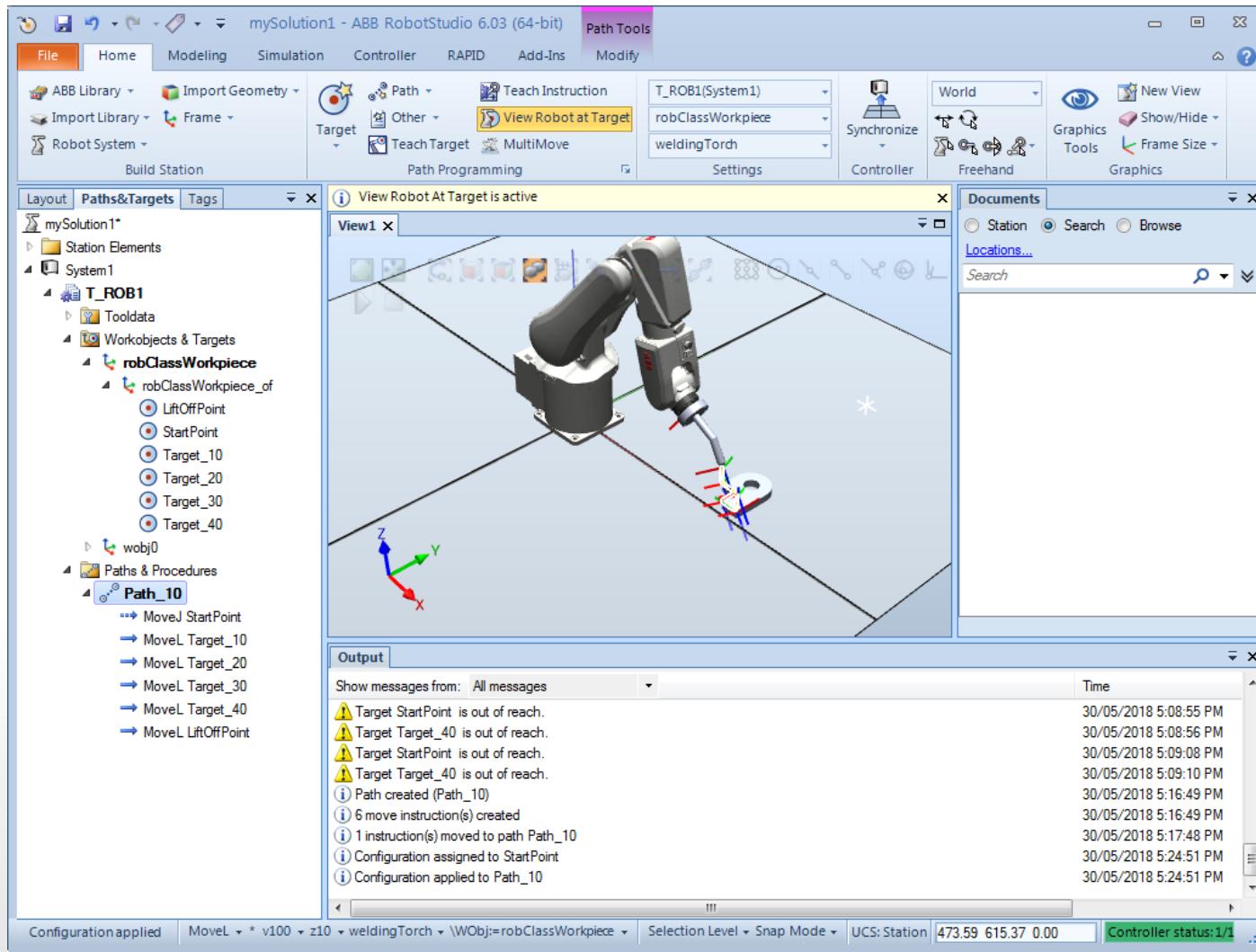


Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- **Create Main Path**
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

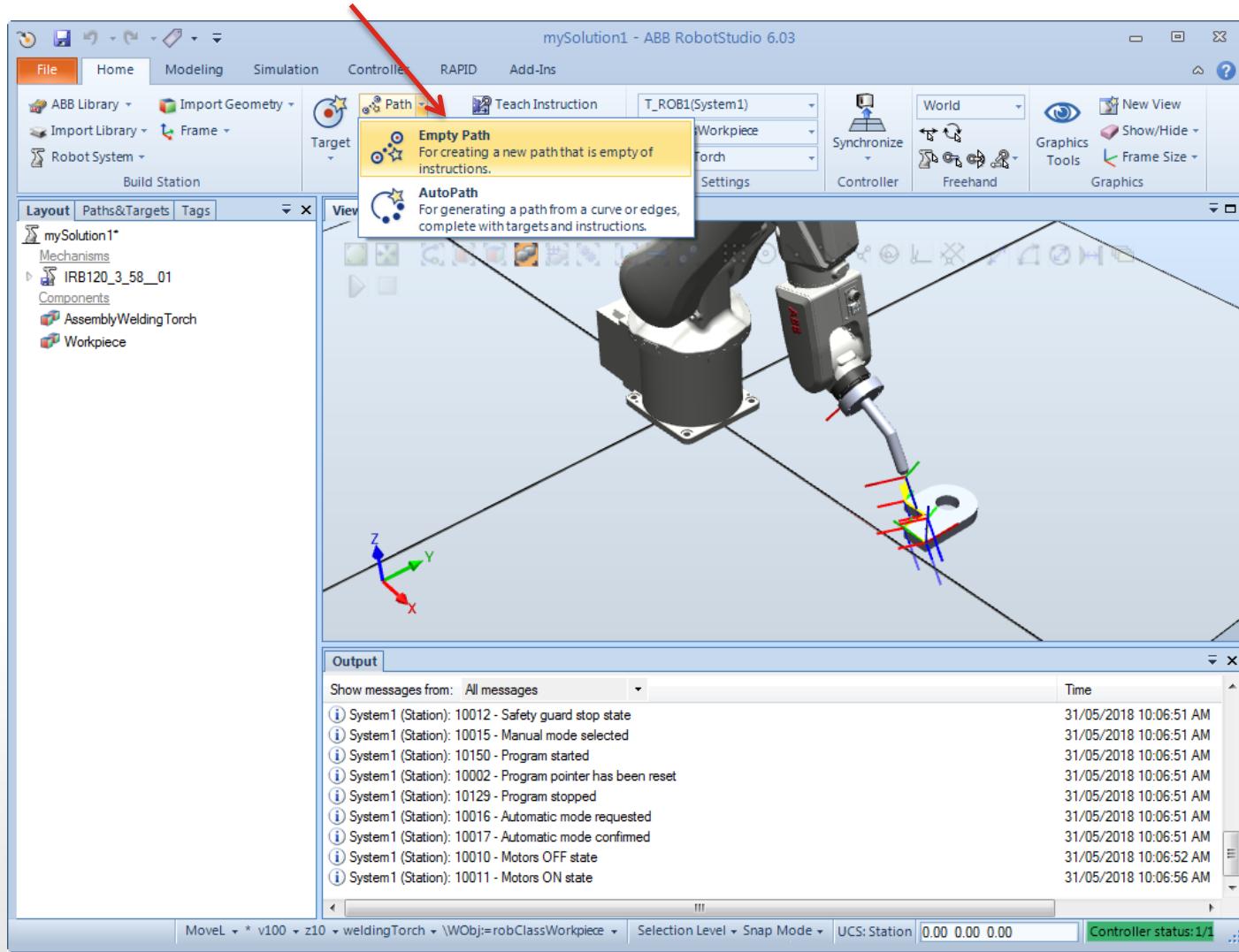
Create Main Path

- “Main” is the entry point when running a robot program.



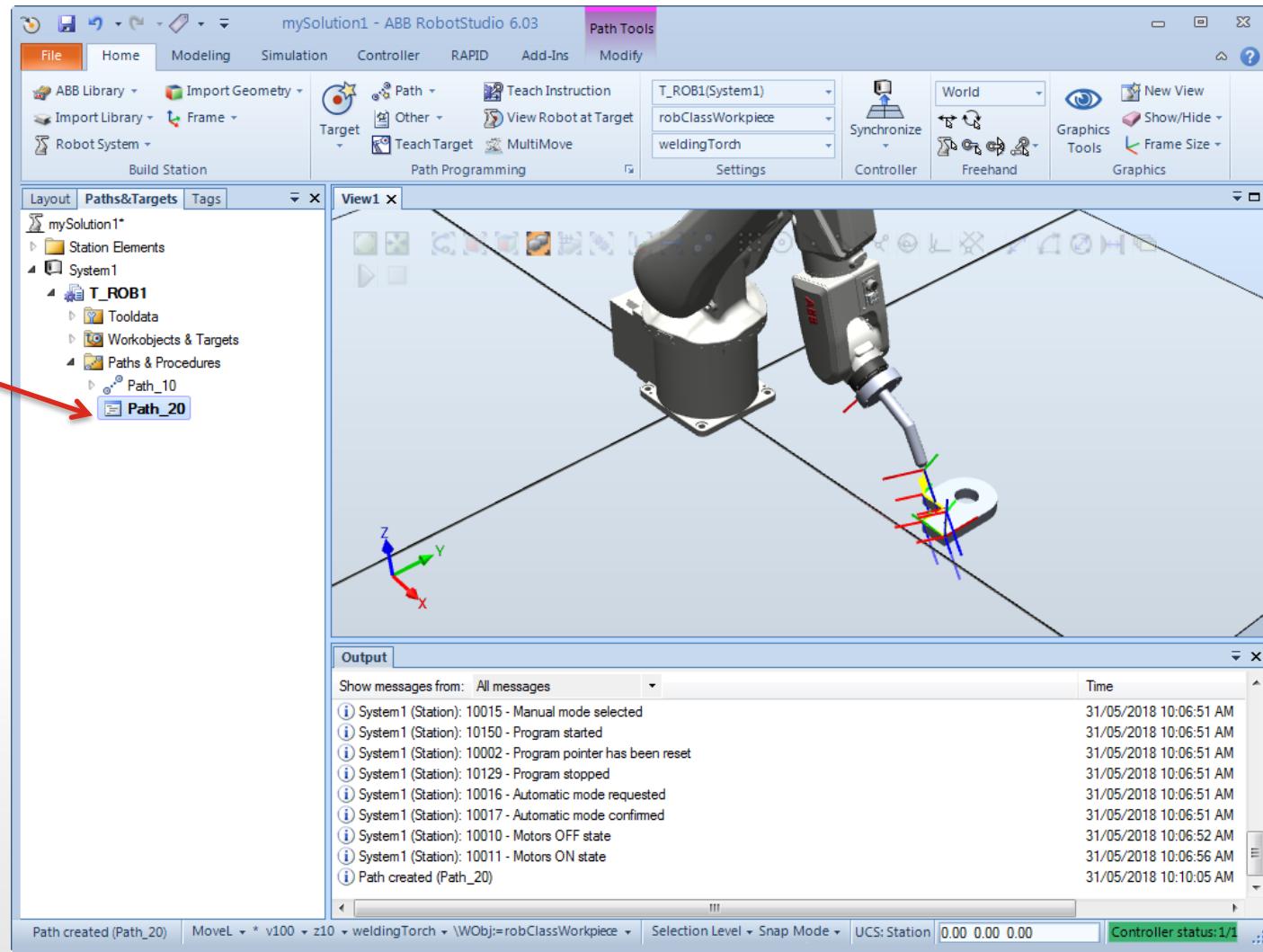
Create Main Path

- Under “Path” → Create a new **empty path**.



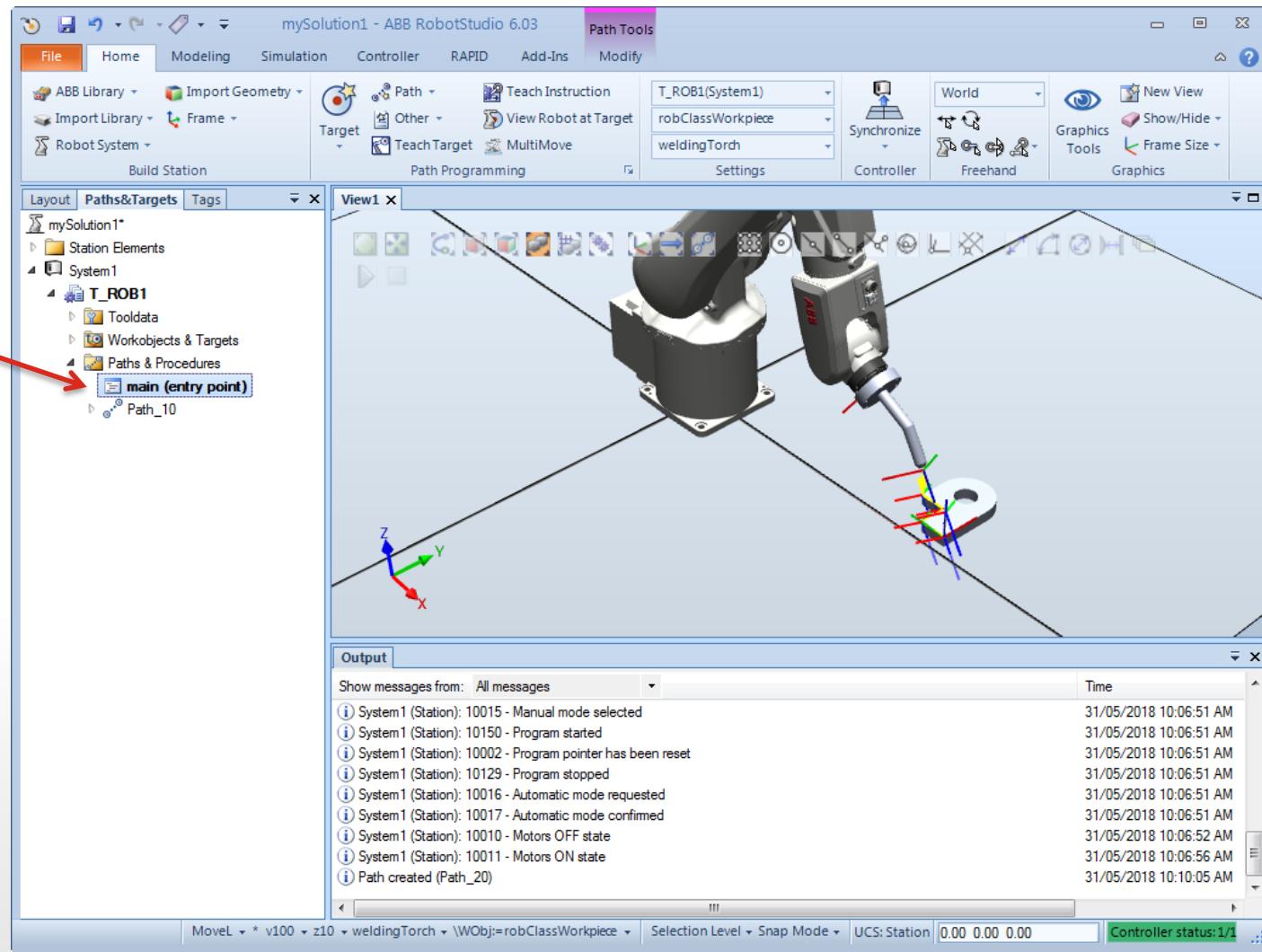
Create Main Path

- A “Path_20” is created.



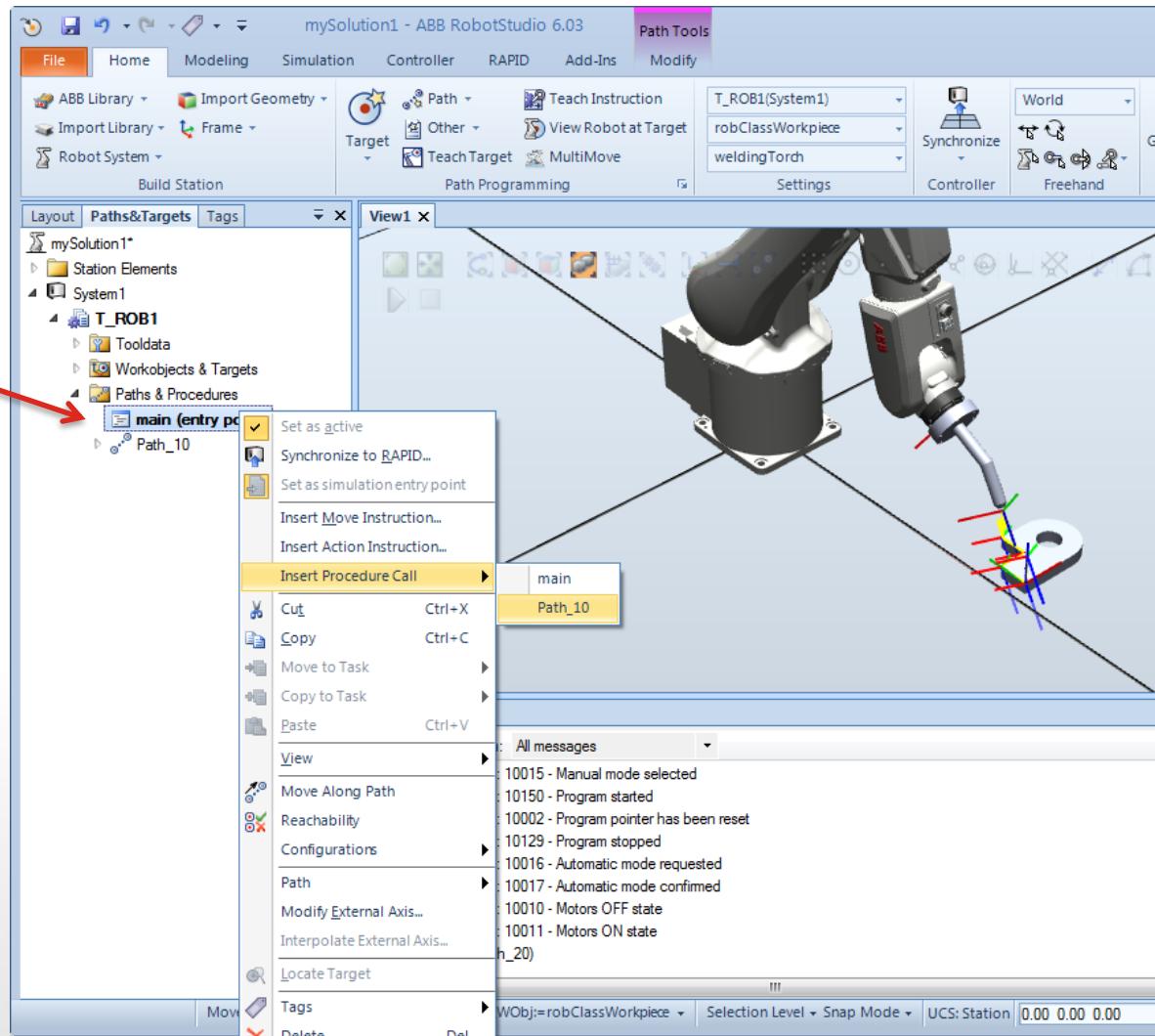
Create Main Path

- Rename this to “main”.



Insert Procedure Call

- Right click main → Insert procedure call → choose Path_10

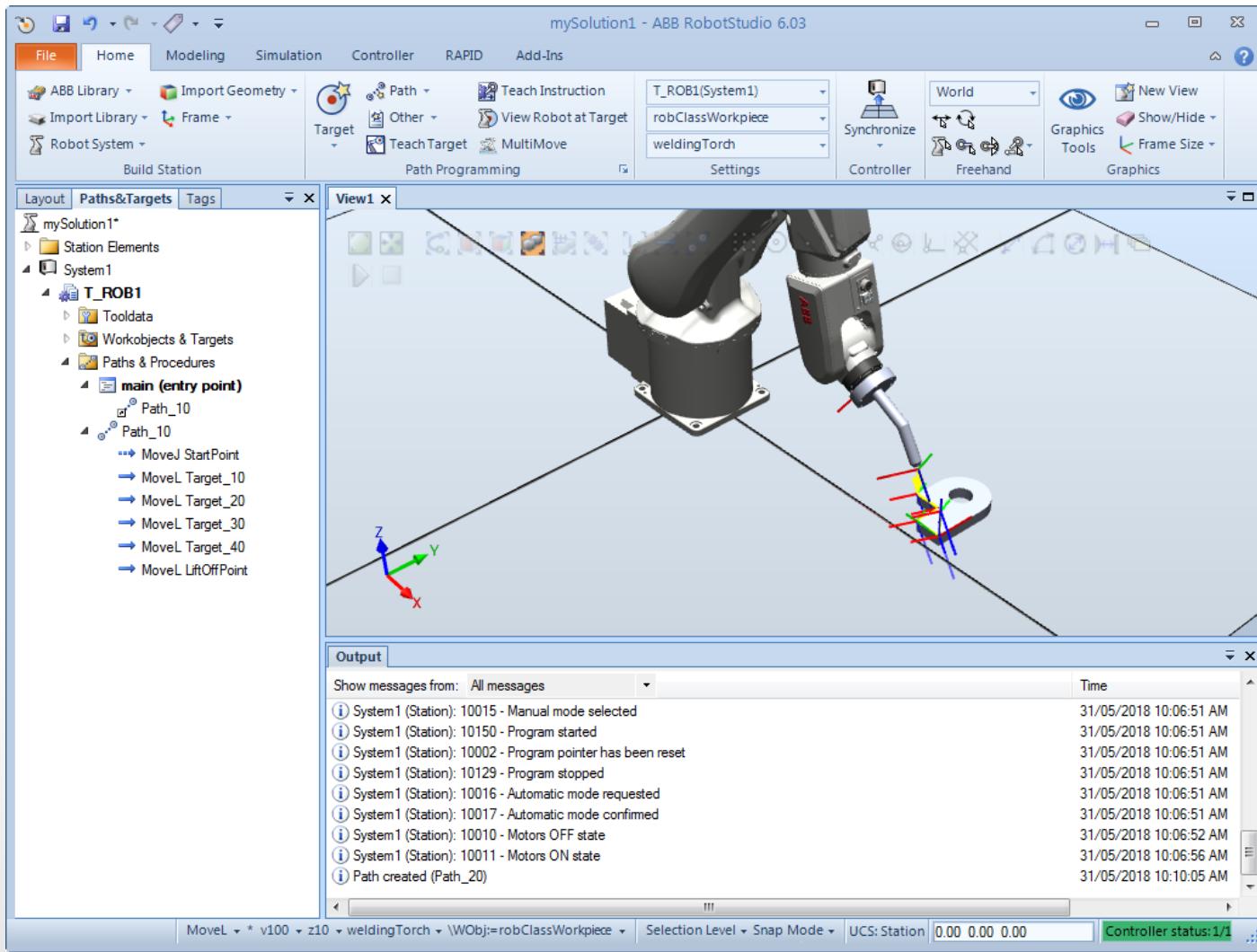


Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- **Synchronize to Rapid**
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

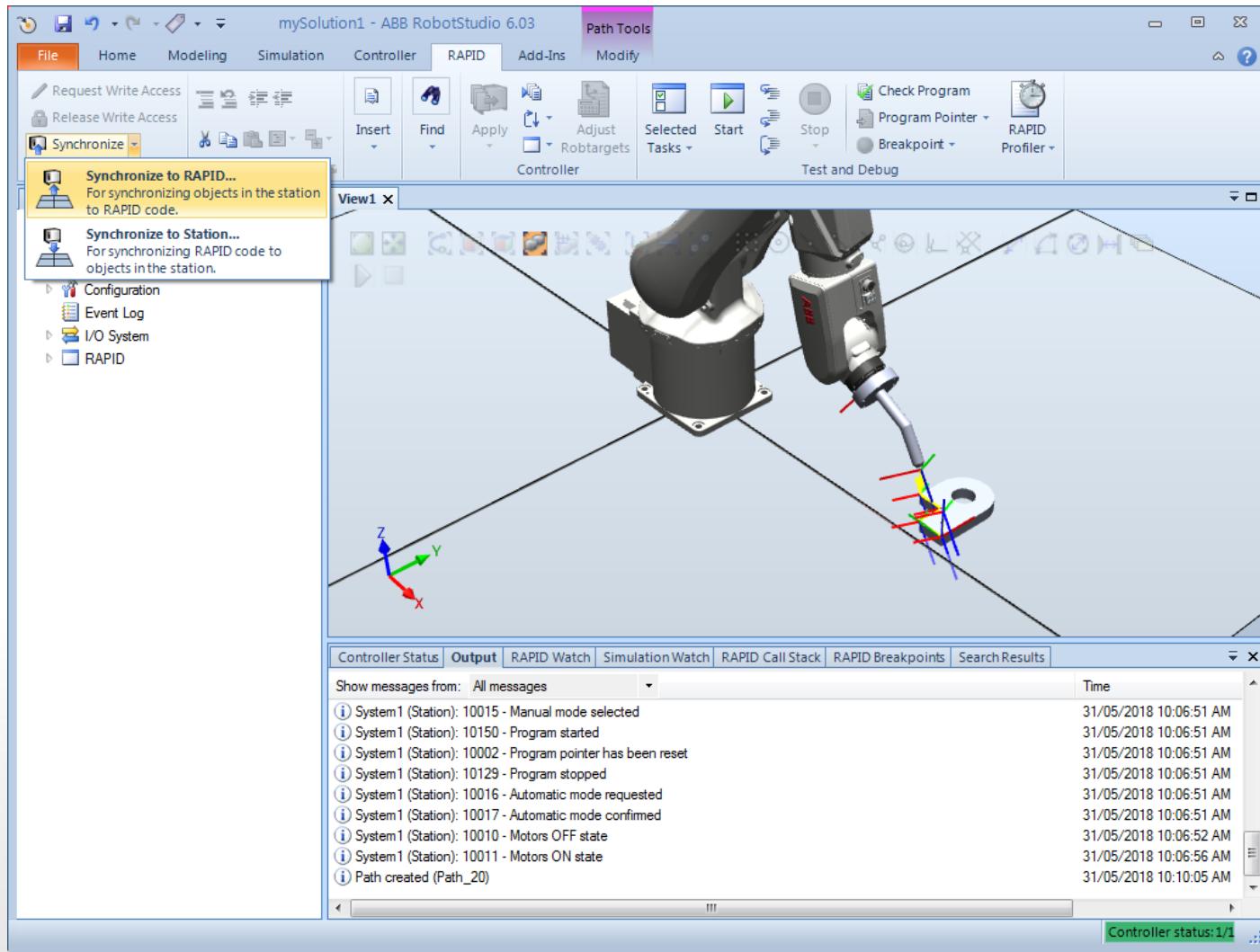
Synchronize to Rapid

- We now synchronize to Rapid. This will create a Rapid code for the robot.



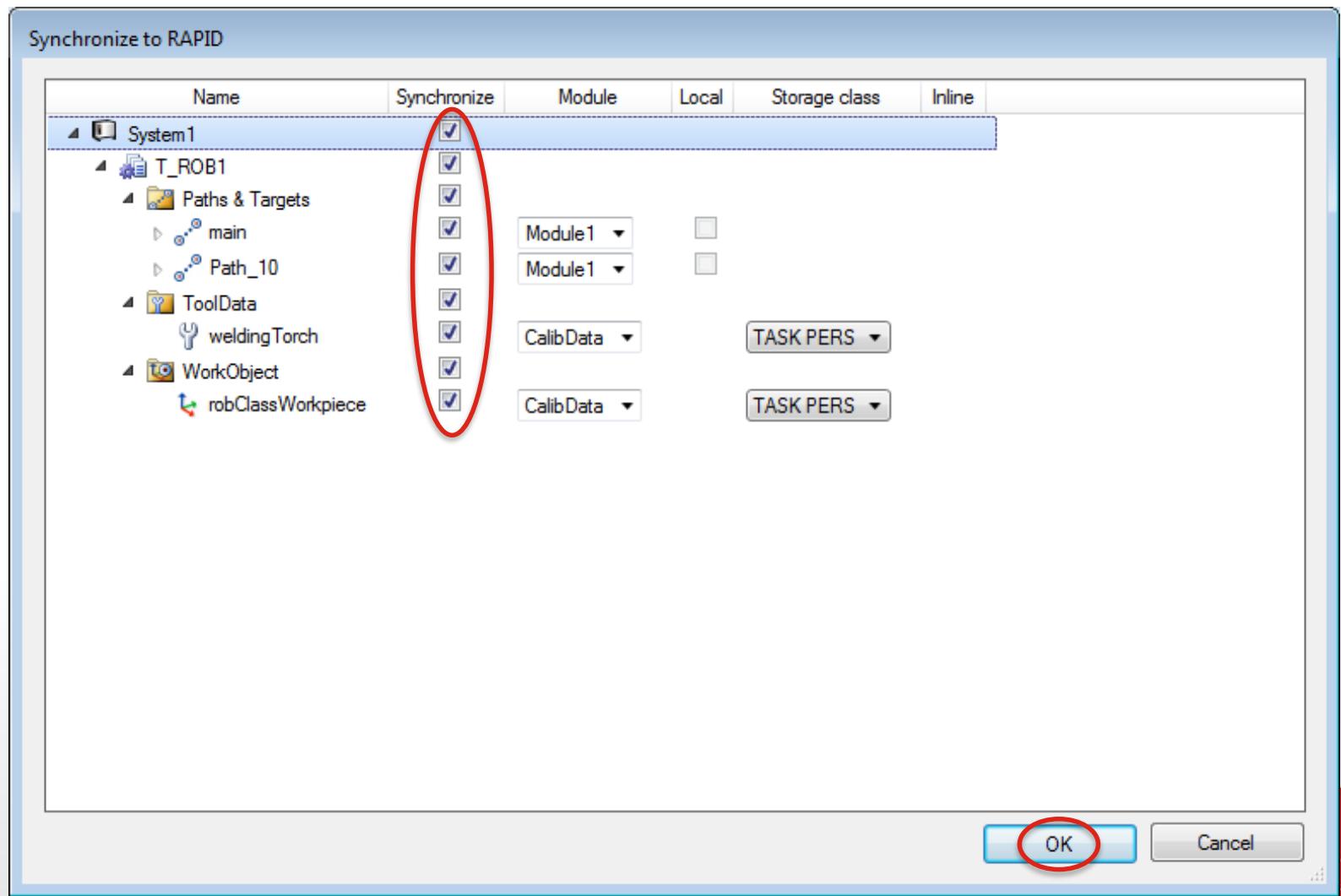
Synchronize to Rapid

- Under “Rapid” tab, click Synchronize → Synchronize to Rapid



Synchronize to Rapid

- In the following window, select **ALL boxes** and then click OK.

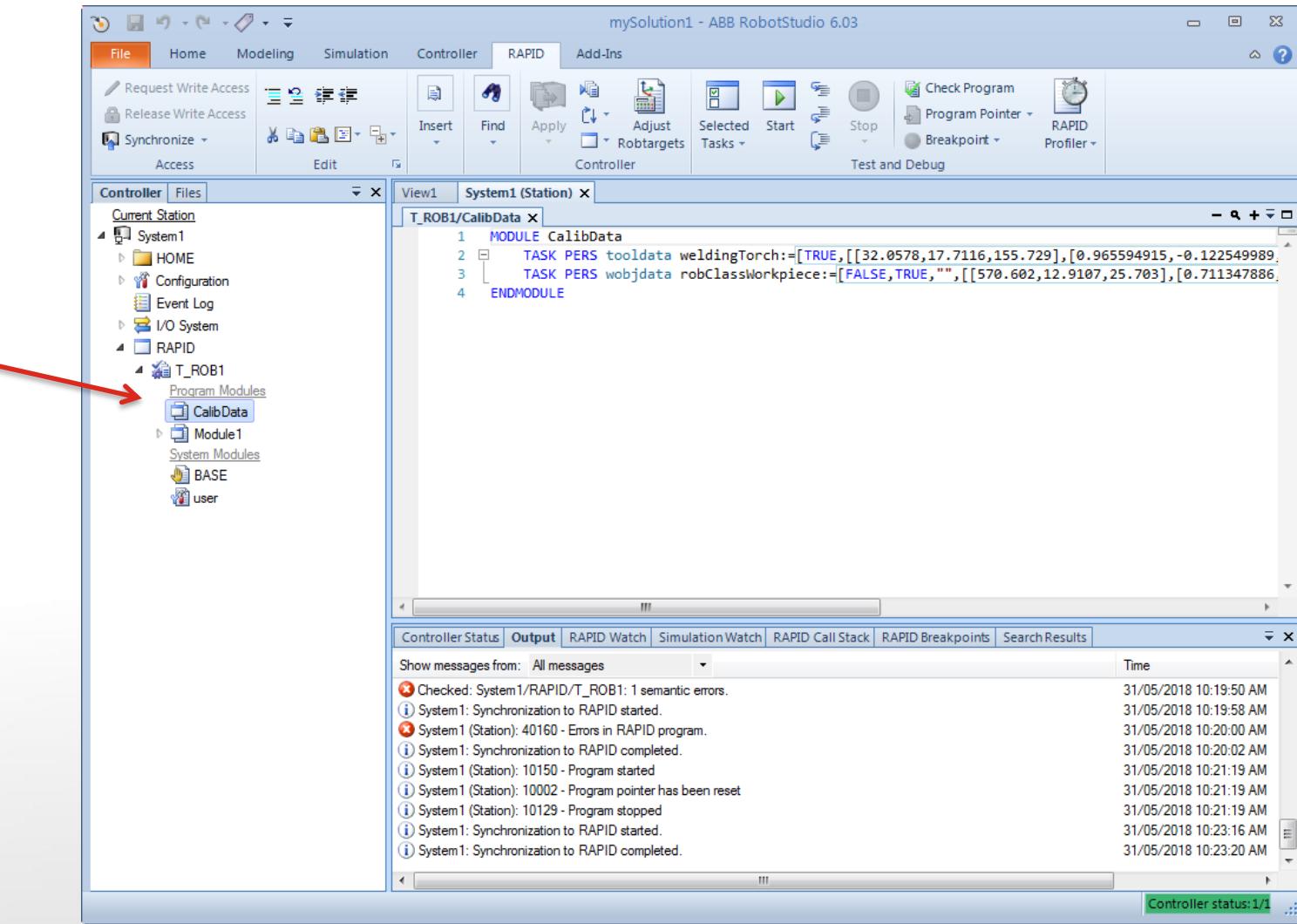


Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- **Show Module in Rapid**
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

Show Module in Rapid

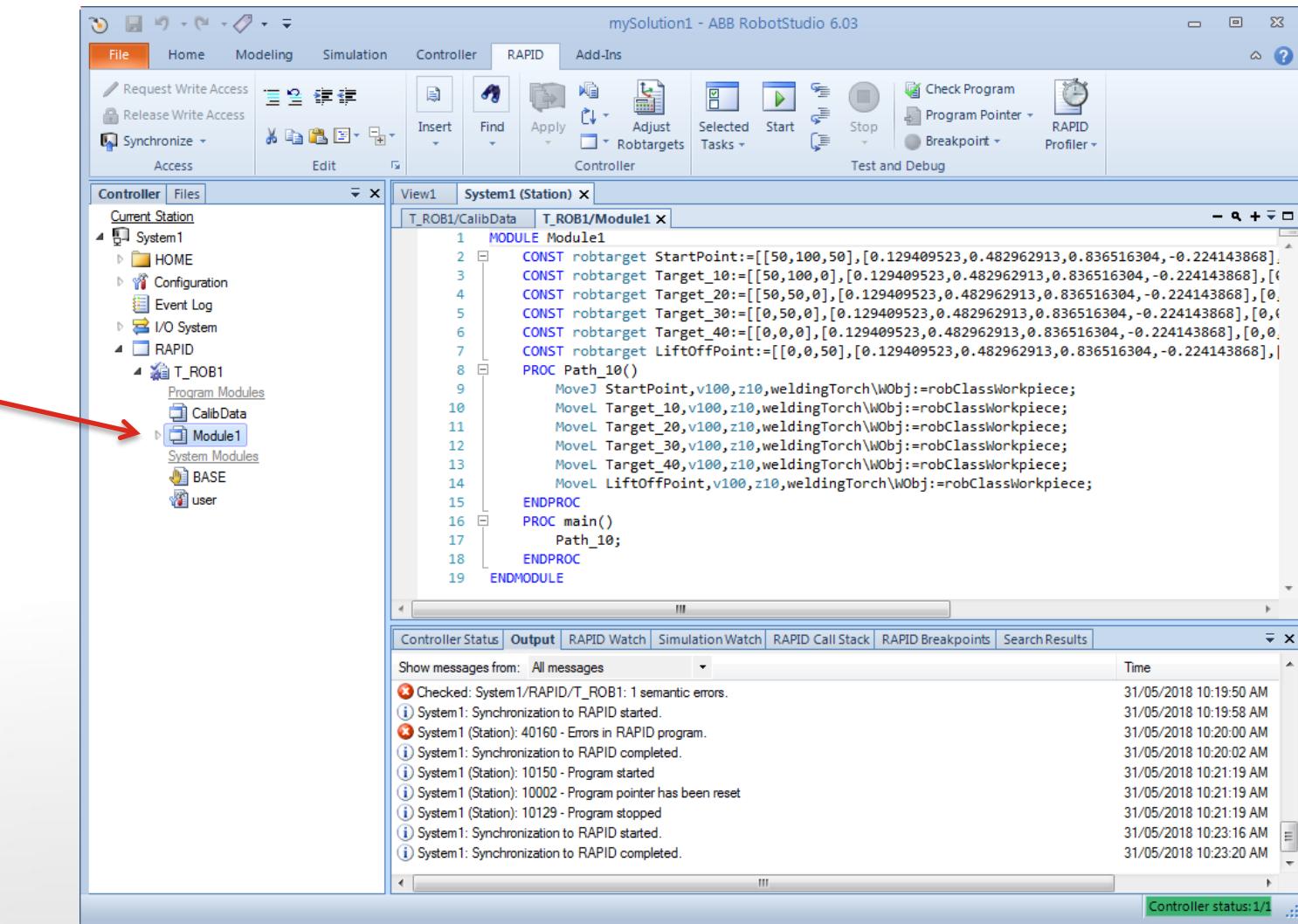
- Under “Rapid” tab, expand the tree, and double click **CalibData**.



These are the tool and workpiece calibration data.

Show Module in Rapid

- Under “Rapid” tab, expand the tree, and double click **Module 1**.



The screenshot shows the ABB RobotStudio 6.03 interface with the 'RAPID' tab selected. The left sidebar displays a tree structure for 'Controller' and 'Files'. Under 'T_ROB1', 'Program Modules' contains 'CalibData' and 'Module1'. A red arrow points to 'Module1'. The main workspace shows the RAPID code for 'Module1' with line numbers 1 through 19. The code defines a module with various constants for target points and a path procedure. Below the code, the 'Output' tab of the status bar shows a list of messages from the RAPID program.

```
mySolution1 - ABB RobotStudio 6.03
File Home Modeling Simulation Controller RAPID Add-Ins
Request Write Access Release Write Access Synchronize
Access Edit
Controller Files View1 System1 (Station) x
T_ROB1/CalibData T_ROB1/Module1 x
1 MODULE Module1
2 CONST robtarget StartPoint:=[[50,100,50],[0.129409523,0.482962913,0.836516304,-0.224143868].
3 CONST robtarget Target_10:=[[50,100,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
4 CONST robtarget Target_20:=[[50,50,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
5 CONST robtarget Target_30:=[[0,50,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
6 CONST robtarget Target_40:=[[0,0,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
7 CONST robtarget LiftOffPoint:=[[0,0,50],[0.129409523,0.482962913,0.836516304,-0.224143868],|]
8 PROC Path_10()
9     MoveJ StartPoint,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
10    MoveL Target_10,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
11    MoveL Target_20,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
12    MoveL Target_30,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
13    MoveL Target_40,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
14    MoveL LiftOffPoint,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
15 ENDPROC
16 PROC main()
17     Path_10;
18 ENDPROC
19 ENDMODULE
```

Controller Status Output RAPID Watch Simulation Watch RAPID Call Stack RAPID Breakpoints Search Results

Show messages from: All messages Time

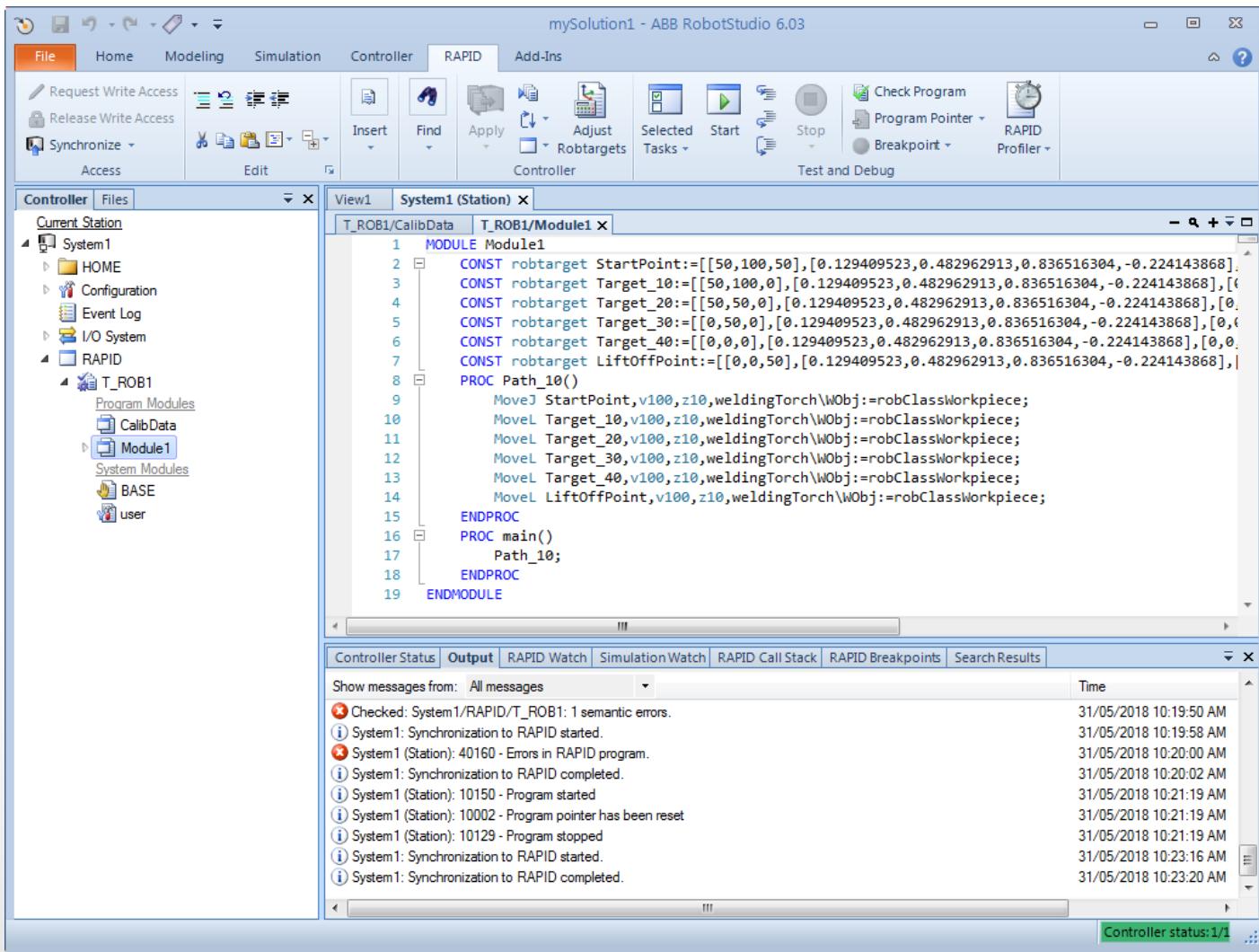
- Checked: System1/RAPID/T_ROB1: 1 semantic errors. 31/05/2018 10:19:50 AM
- System1: Synchronization to RAPID started. 31/05/2018 10:19:58 AM
- System1 (Station): 40160 - Errors in RAPID program. 31/05/2018 10:20:00 AM
- System1: Synchronization to RAPID completed. 31/05/2018 10:20:02 AM
- System1 (Station): 10150 - Program started 31/05/2018 10:21:19 AM
- System1 (Station): 10002 - Program pointer has been reset 31/05/2018 10:21:19 AM
- System1 (Station): 10129 - Program stopped 31/05/2018 10:21:19 AM
- System1: Synchronization to RAPID started. 31/05/2018 10:23:16 AM
- System1: Synchronization to RAPID completed. 31/05/2018 10:23:20 AM

Controller status: 1/1

This contains the target points, path and main program.

Show Module in Rapid

- These Rapid codes can be copied and uploaded to the real robot! Will be shown later.



The screenshot shows the ABB RobotStudio 6.03 interface with the title bar "mySolution1 - ABB RobotStudio 6.03". The menu bar includes File, Home, Modeling, Simulation, Controller, RAPID, and Add-Ins. The RAPID tab is selected. The left sidebar shows the "Controller" tab and a tree view of the system structure under "T_ROB1": System1, HOME, Configuration, Event Log, I/O System, RAPID, T_ROB1, Program Modules, Calib Data, Module1, System Modules, BASE, and user. The main workspace displays RAPID code for a module named "Module1". The code defines several constants for target points and a procedure "Path_10" which moves the welding torch through various targets. The output window at the bottom shows messages from the RAPID program, including semantic errors and synchronization status.

```
1 MODULE Module1
2 CONST robtarget StartPoint:=[[50,100,50],[0.129409523,0.482962913,0.836516304,-0.224143868].
3 CONST robtarget Target_10:=[[50,100,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
4 CONST robtarget Target_20:=[[50,50,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
5 CONST robtarget Target_30:=[[0,50,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
6 CONST robtarget Target_40:=[[0,0,0],[0.129409523,0.482962913,0.836516304,-0.224143868],[0,
7 CONST robtarget LiftOffPoint:=[[0,0,50],[0.129409523,0.482962913,0.836516304,-0.224143868],
8 PROC Path_10()
9     MoveJ StartPoint,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
10    MoveL Target_10,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
11    MoveL Target_20,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
12    MoveL Target_30,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
13    MoveL Target_40,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
14    MoveL LiftOffPoint,v100,z10,weldingTorch\WObj:=robClassWorkpiece;
15 ENDPROC
16 PROC main()
17     Path_10;
18 ENDPROC
19 ENDMODULE
```

Show messages from: All messages

- Checked: System1/RAPID/T_ROB1: 1 semantic errors.
- System1: Synchronization to RAPID started.
- System1 (Station): 40160 - Errors in RAPID program.
- System1: Synchronization to RAPID completed.
- System1 (Station): 10150 - Program started
- System1 (Station): 10002 - Program pointer has been reset
- System1 (Station): 10129 - Program stopped
- System1: Synchronization to RAPID started.
- System1: Synchronization to RAPID completed.

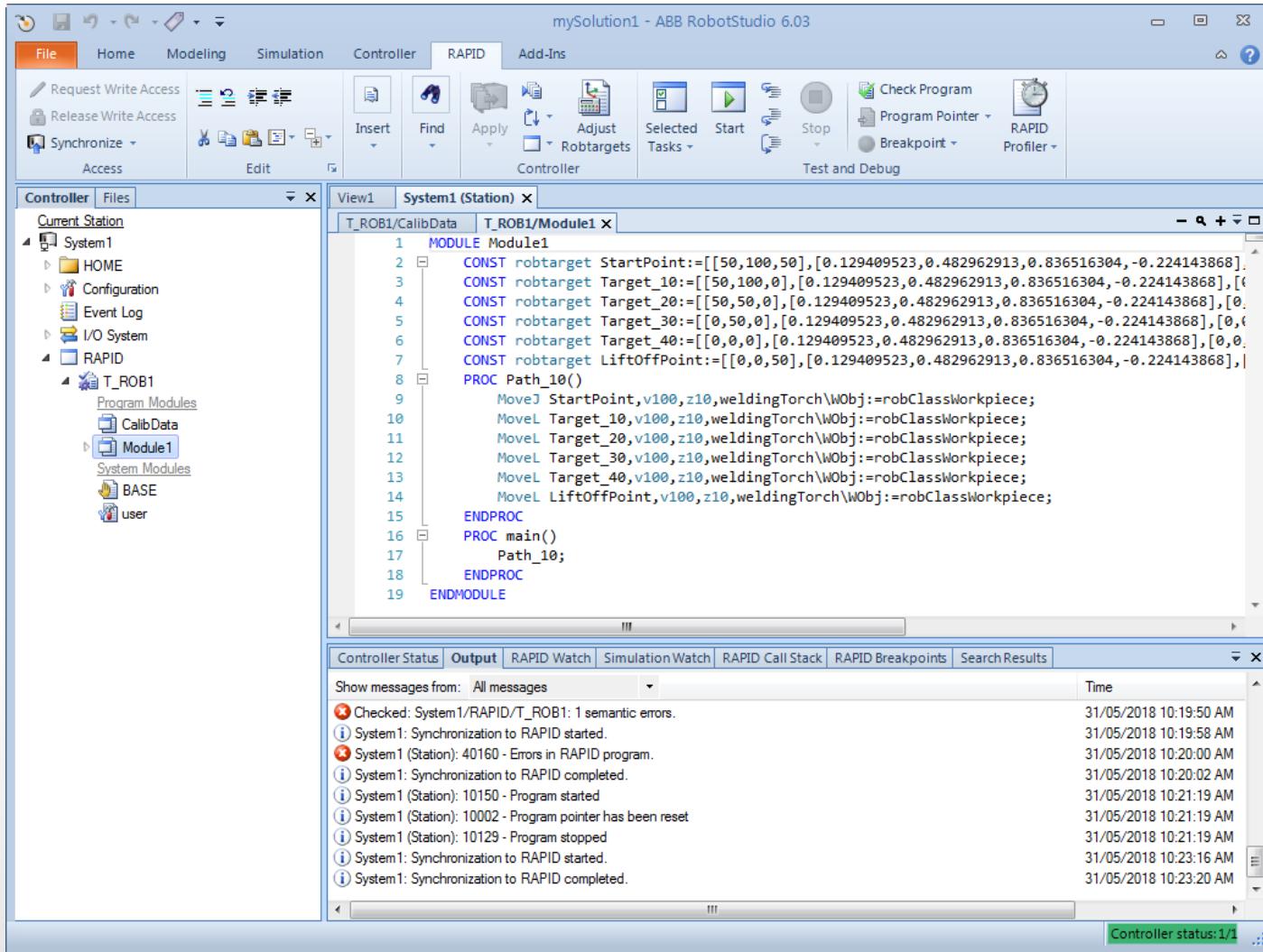
Controller status: 1/1

Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- **Play Simulation**
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

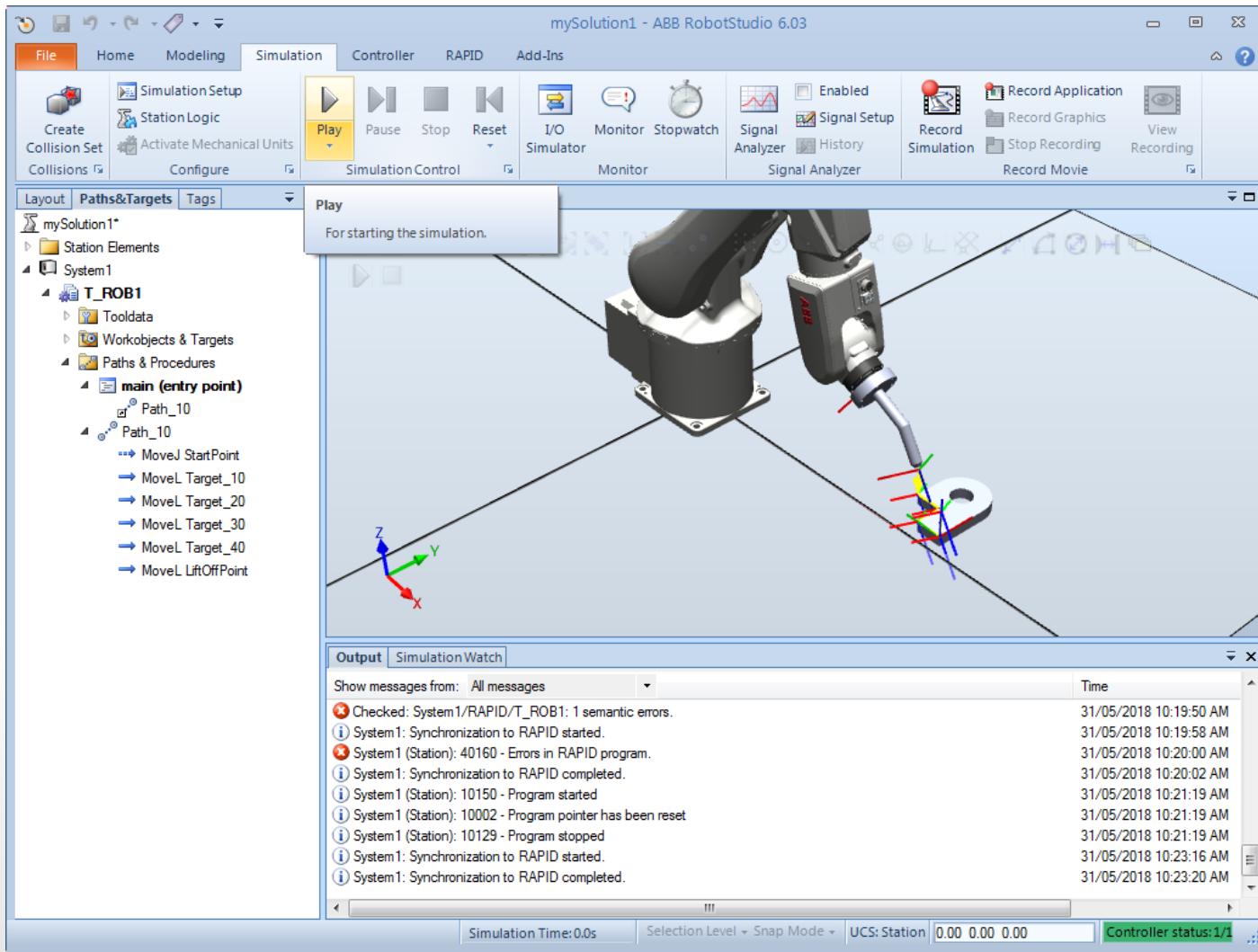
Play Simulation

- We can also **simulate** the code in RobotStudio.



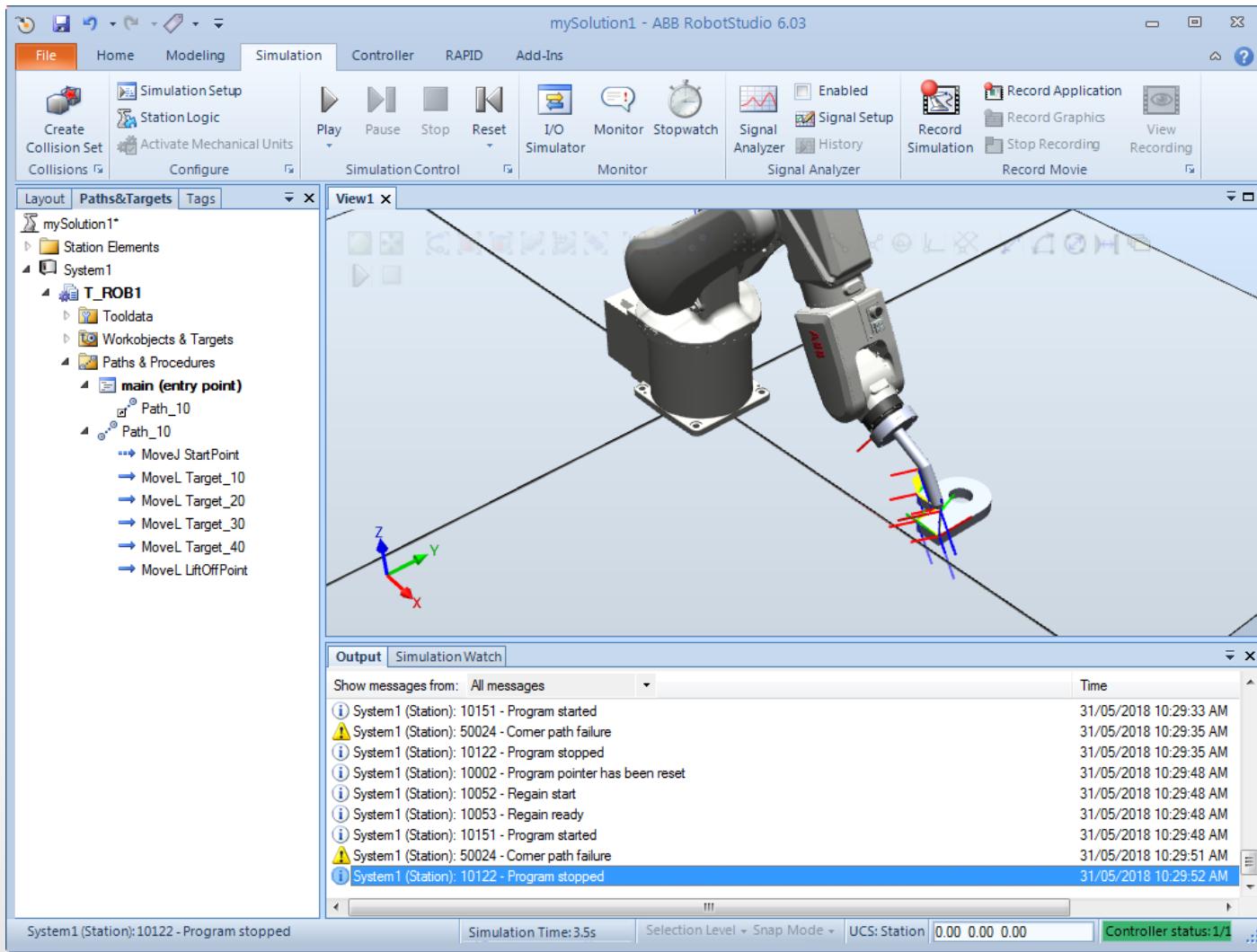
Play Simulation

- Under “Simulation” tab, press the “Play” button.



Play Simulation

- We see the simulation running **successfully**.

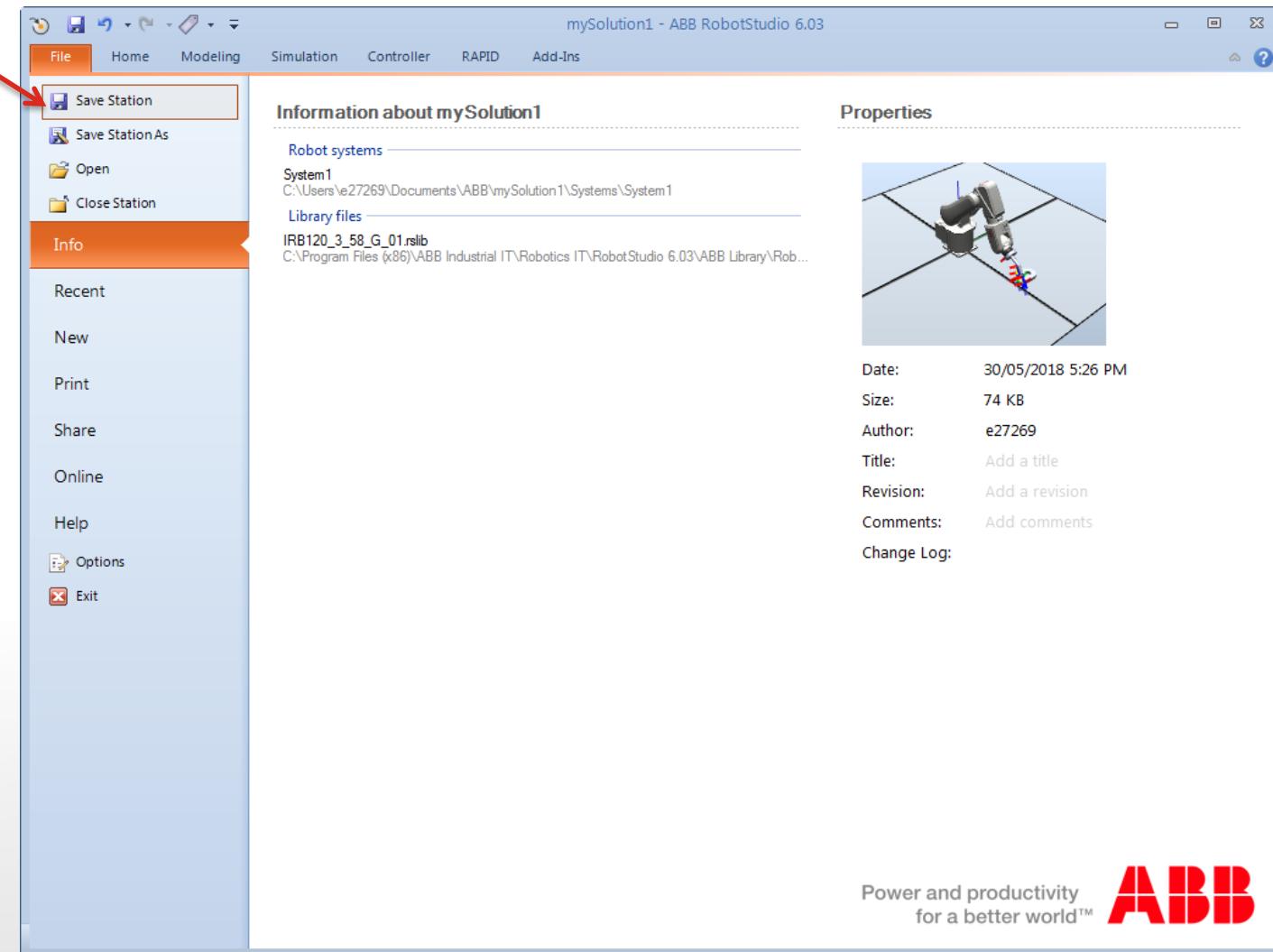


Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- **Save Station**
- Saving the Rapid File
- Pack and Go / Unpack and Work

Save Station

- Finally, save the station.



Power and productivity
for a better world™



Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

Save the Rapid File

- You can copy the created Rapid files as **mod** files.

```
CalibData.mod - Notepad
File Edit Format View Help
MODULE CalibData
    TASK PERS tooldata weldingTorch:=[TRUE, [[32.0578, :,
    TASK PERS wobjdata robclassworkpiece:=[FALSE, TRUE
ENDMODULE
```

```
Module2.txt - Notepad
File Edit Format View Help
MODULE Module2
    CONST robtarget StartPoint:=[[50,100,50],[0.12940
    CONST robtarget Target_10:=[[50,100,0],[0.1294095
    CONST robtarget Target_20:=[[50,50,0],[0.12940952
    CONST robtarget Target_30:=[[0,50,0],[0.129409523
    CONST robtarget Target_40:=[[0,0,0],[0.129409523,'
    CONST robtarget LiftoffPoint:=[[0,0,50],[0.129409
    PROC Path_10()
        MoveJ StartPoint,v100,z10,weldingTorch\wobj:=r
        MoveL Target_10,v100,z10,weldingTorch\wobj:=r
        MoveL Target_20,v100,z10,weldingTorch\wobj:=r
        MoveL Target_30,v100,z10,weldingTorch\wobj:=r
        MoveL Target_40,v100,z10,weldingTorch\wobj:=r
        MoveL LiftoffPoint,v100,z10,weldingTorch\wobj
    ENDPROC
    PROC main()
        Path_10;
    ENDPROC
ENDMODULE
```

- Copy the files into a USB drive.

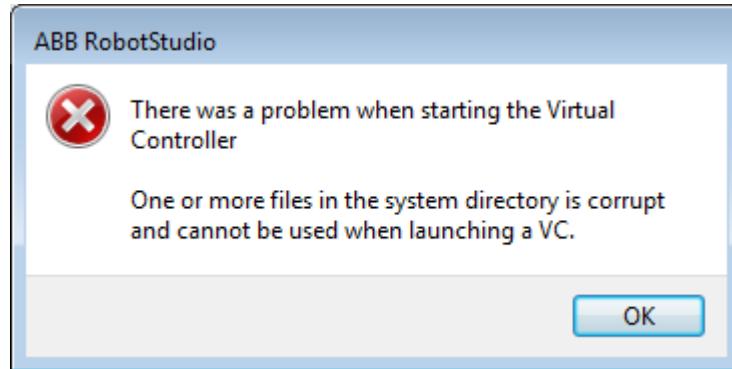
Note: Change to **module2** because there was already module1 on robot. This prevents conflict!

Content

- Open Solution from Previous Tutorial
- Create Targets
- Reorient Targets
- Set Instruction Template
- Add Instruction to New Path
- Modify Instruction
- Auto Configuration
- Create Main Path
- Synchronize to Rapid
- Show Module in Rapid
- Play Simulation
- Save Station
- Saving the Rapid File
- Pack and Go / Unpack and Work

Error when opening - Workaround

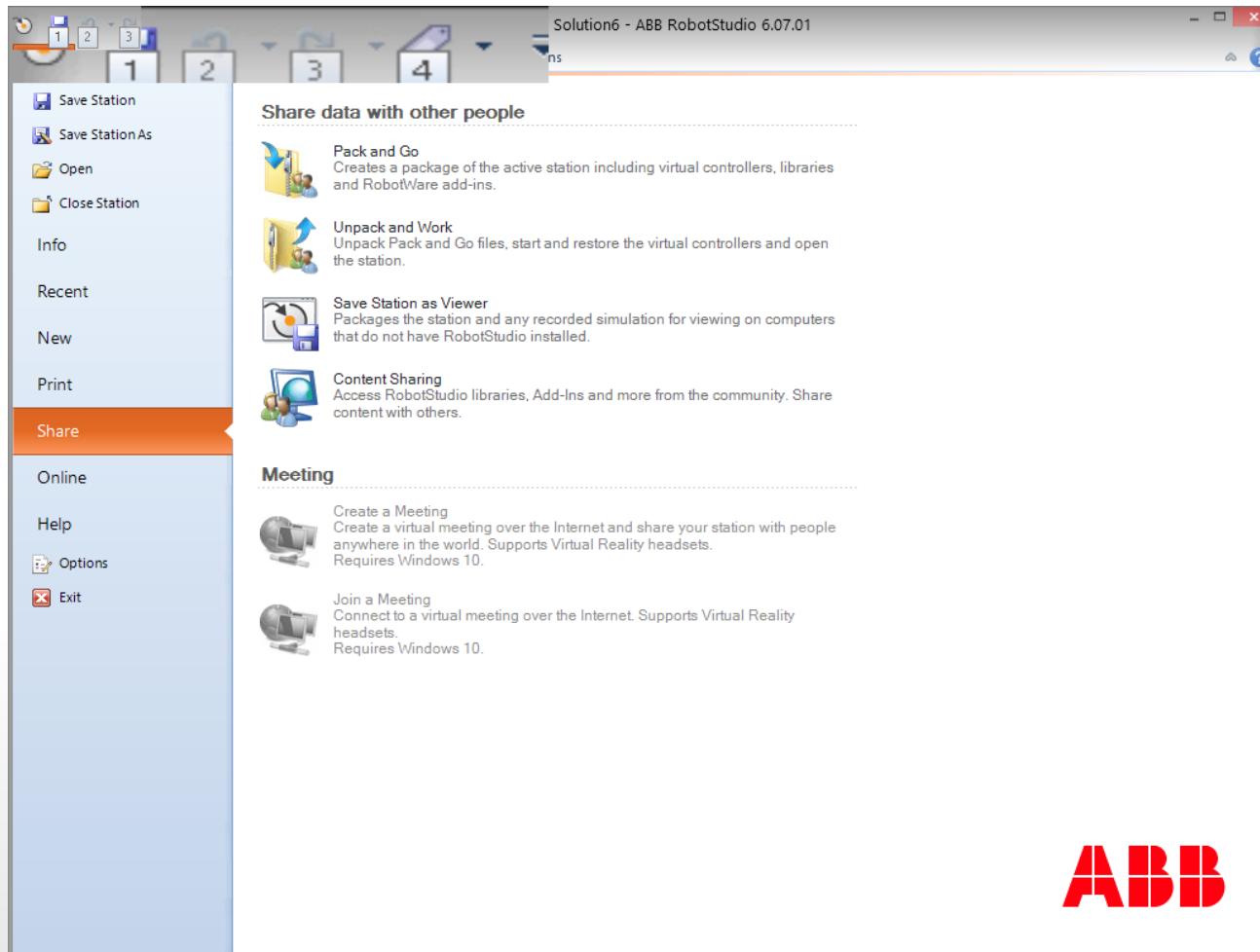
- Sometimes when re-opening solutions, there might be error message as follows:



- Workaround:
 - Robot System → New System → Any System.
 - Close the station.
 - Reload the original / saved solution again.

Pack and Go

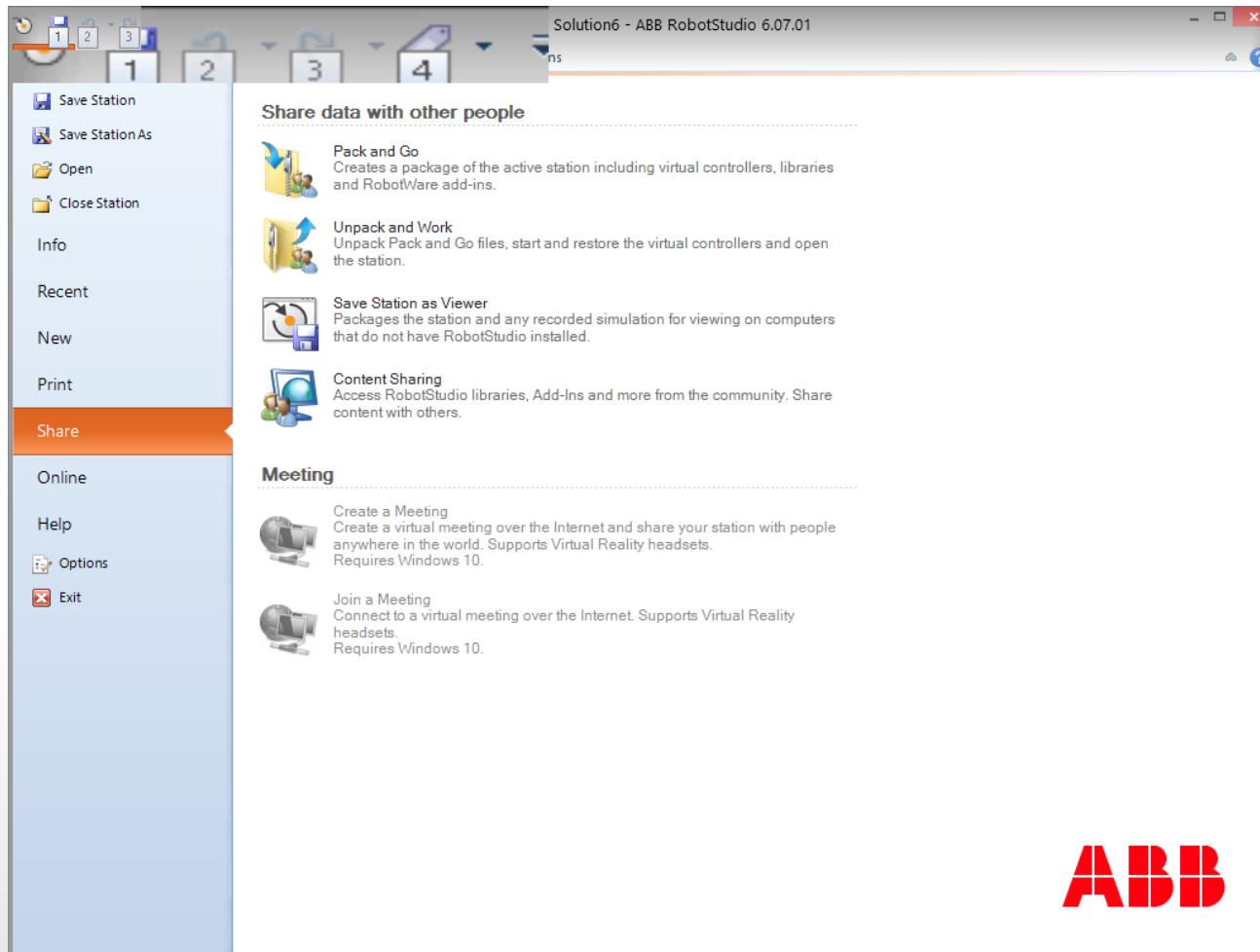
- A better way to save a file for use on another PC later is to use Share → Pack and Go.



ABB

Unpack and Work

- To open the file, choose **Share** → **Unpack and Work**



ABB

Thank you!

Have a good evening.

