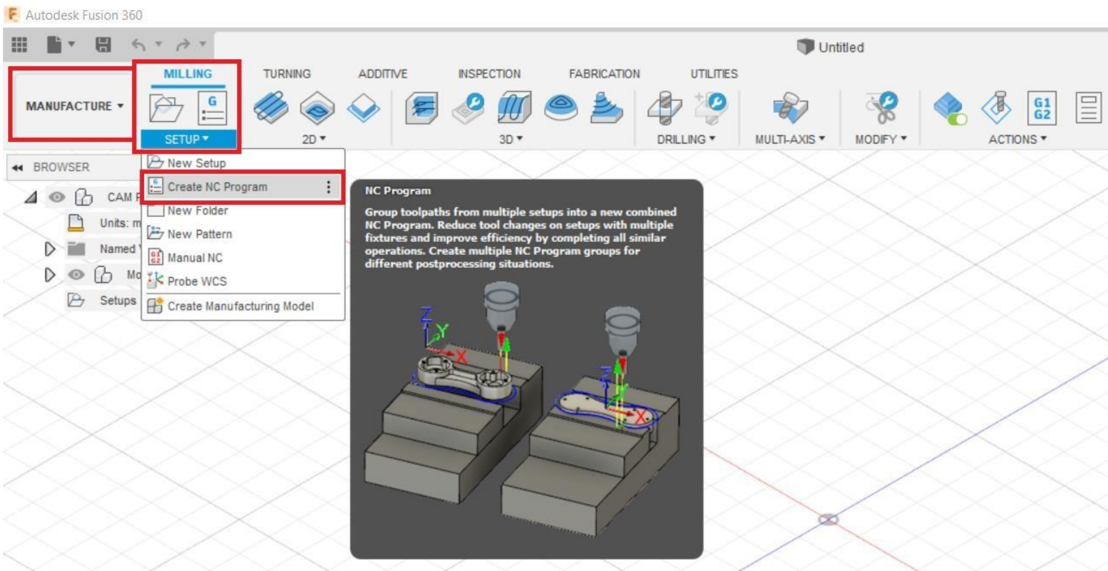
POST-PROCESSOR MANUAL

Contents

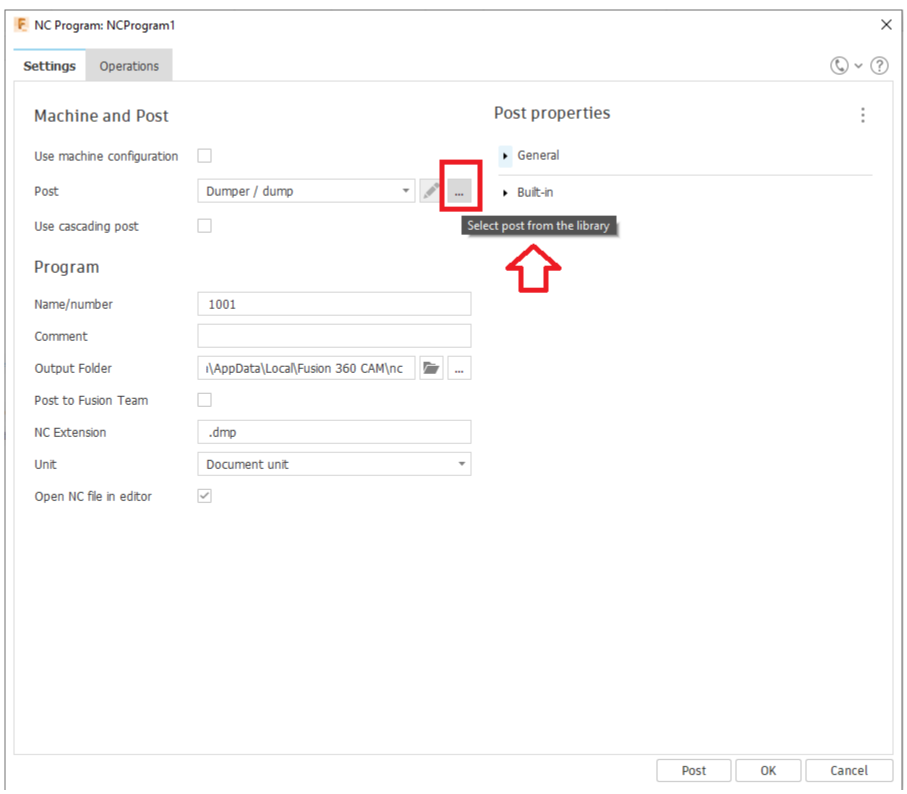
AUTODESK

1. Install Post-processor

Open Autodesk Fusion 360 and switch to the Manufacture Workspace. Expand the Setup menu in the Ribbon to create an NC program.

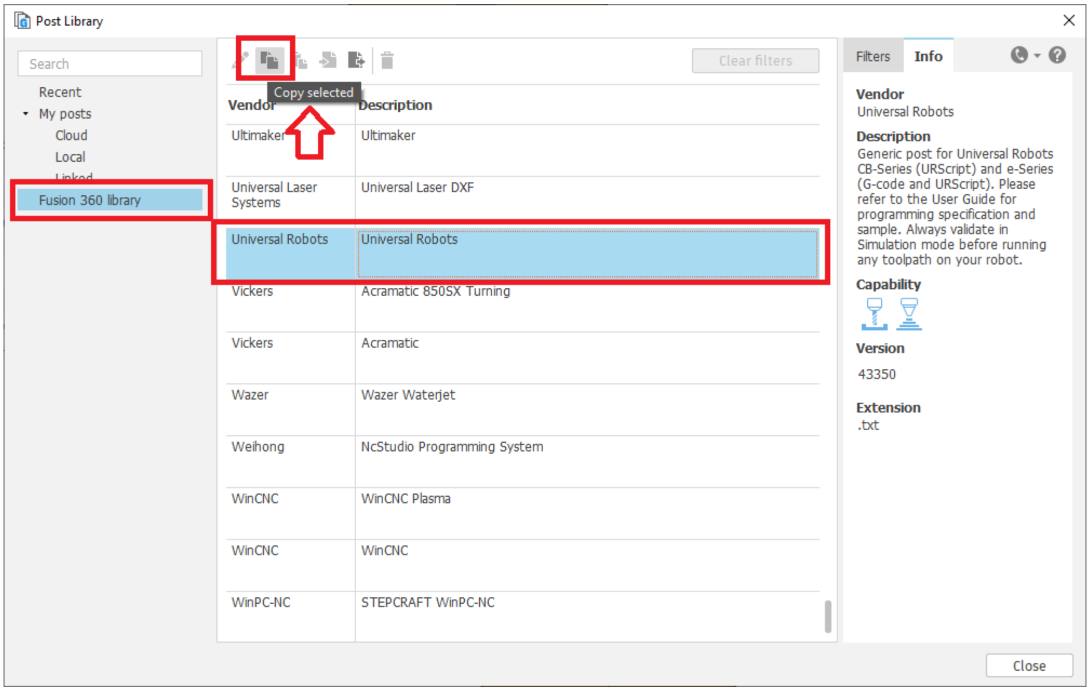


Follow the link in the NC Program window to access the Autodesk HSM Post Library.

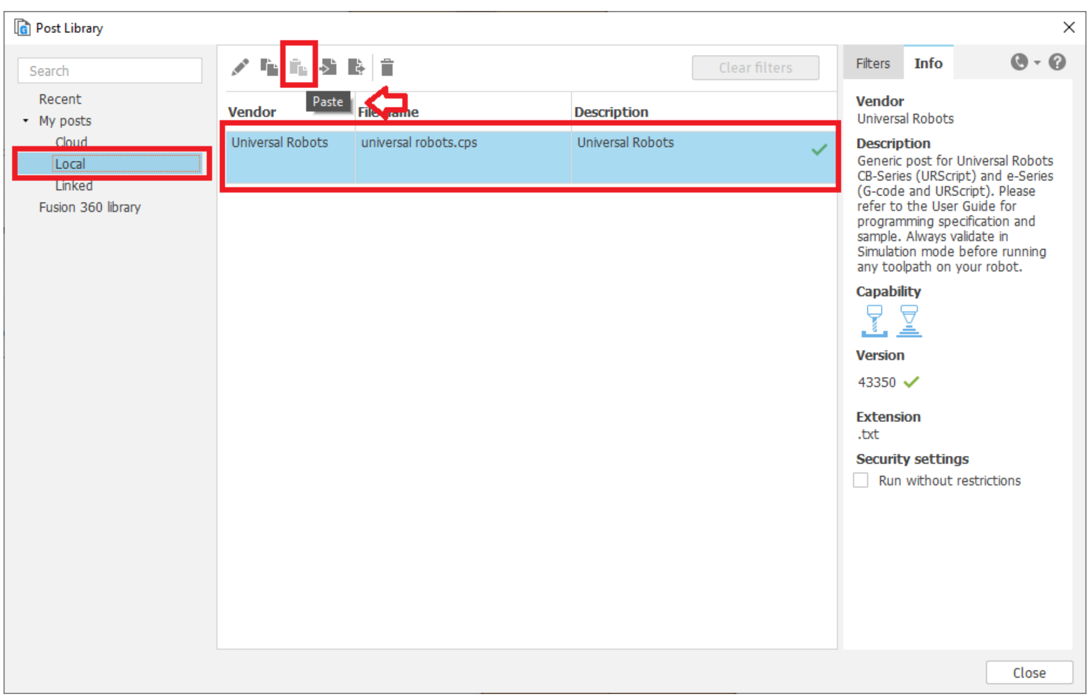


AUTODESK

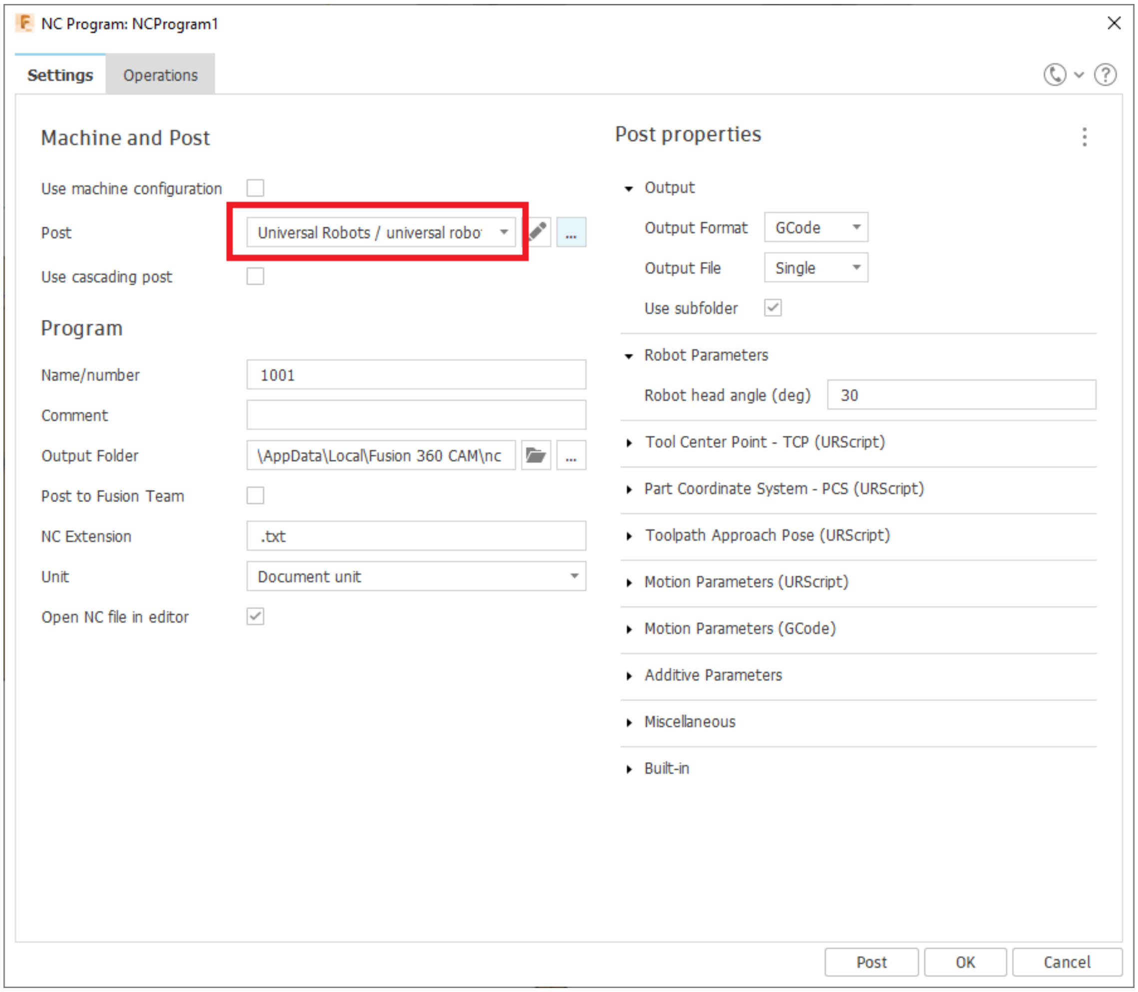
In the Post Library, search for “Universal Robots” post processor and copy the available version.



Paste the post processor configuration file into your Local folder.



Now you are ready to use this post-processor to generate toolpath files for your UR robots.



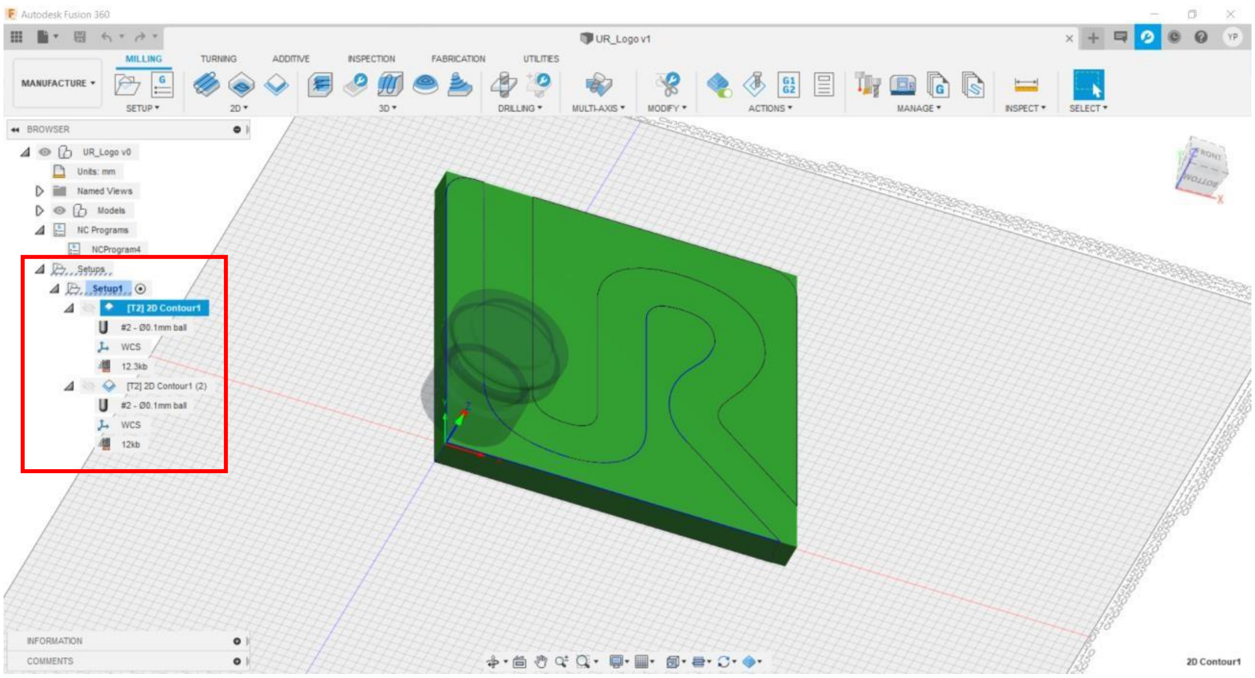
2. Configure and Post Process Toolpath

This section explains how to configure post settings in Fusion 360 to generate toolpath files for Universal Robots.

2.1 Create Toolpath

Follow the normal steps in Fusion 360 to generate one or more toolpaths in your setup.

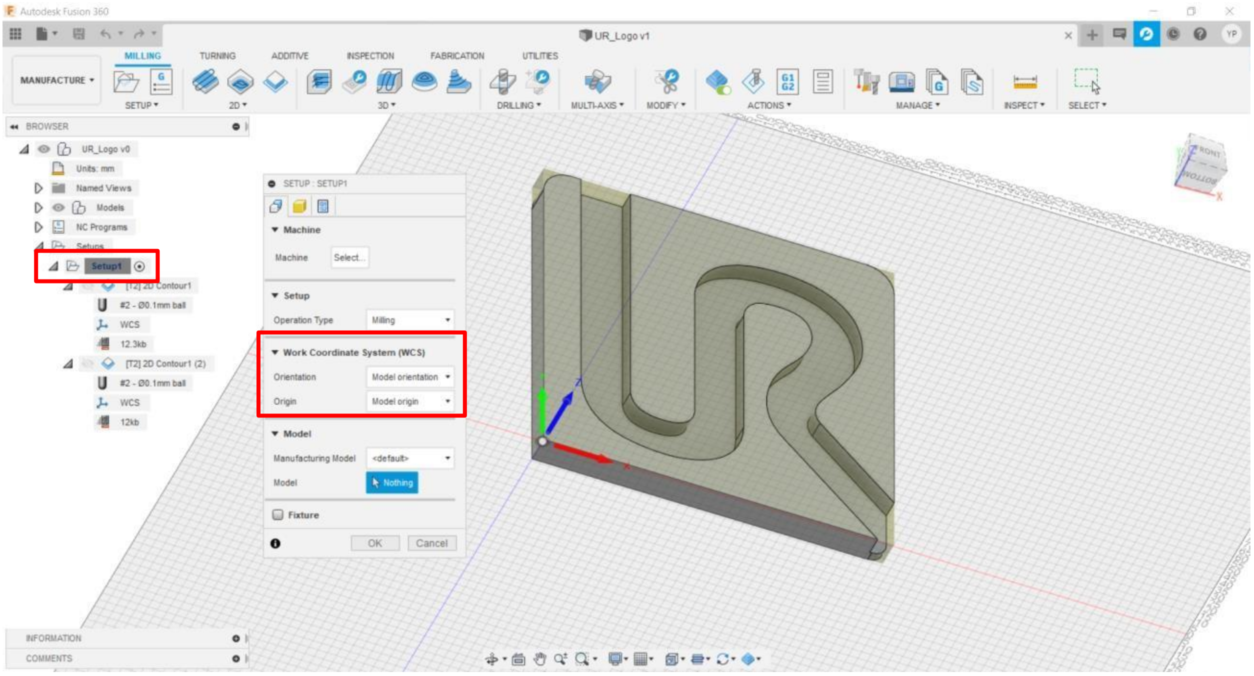
If you need instructions on how to create toolpaths in Fusion 360, please check out the online product documentation for Manufacture Workspace.



2.2 Set up World Coordinate System

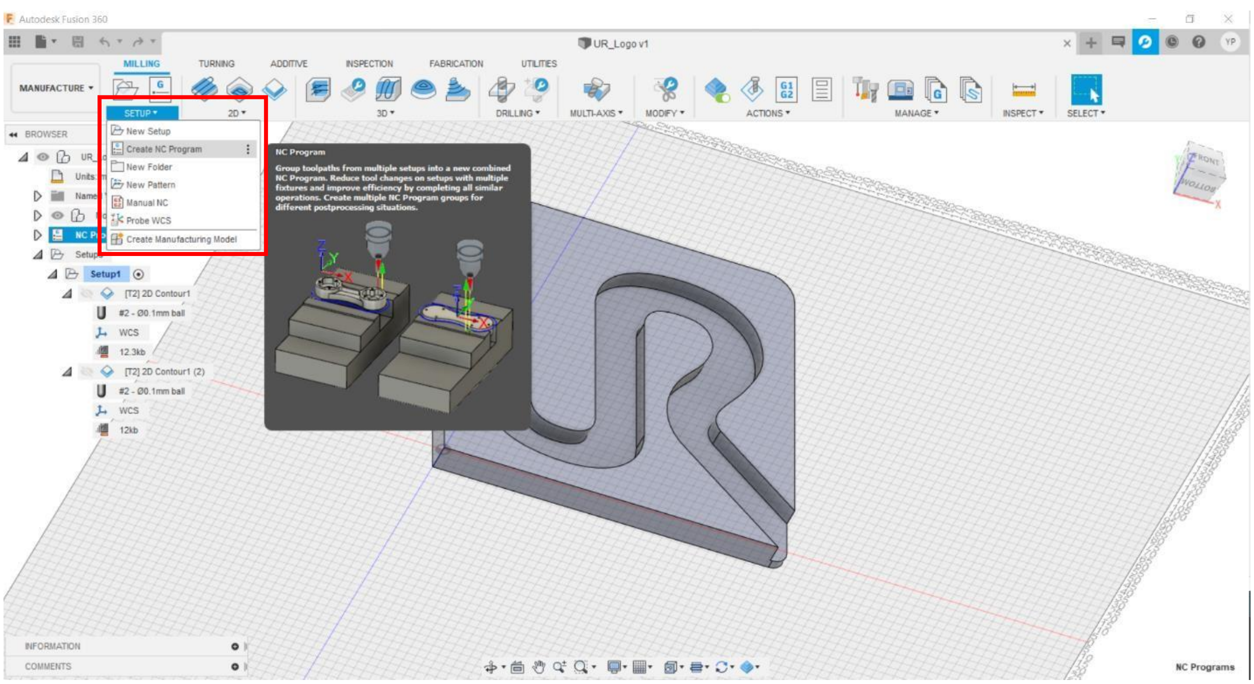
A coordinate system or reference frame fixed to the part is required to define a toolpath relative to the part. In Polyscope, users can define such a coordinate system known as Part Coordinate System (PCS). To run a Fusion 360 toolpath successfully on a UR robot, users must ensure the PCS in the robot program matches the World Coordinate System (WCS) in Fusion 360.

Use the WCS setup menu to replicate the position and orientation of the PCS on the part:

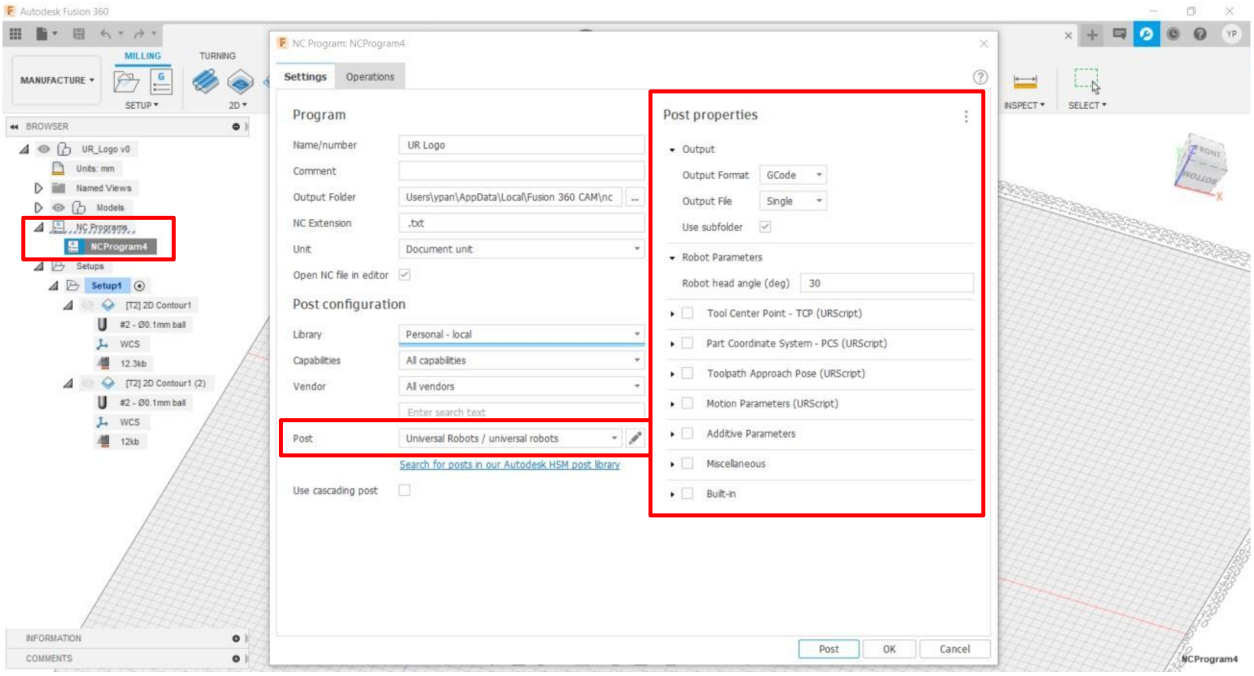


2.3 Create NC Program

It is recommended to configure the post properties in an NC program. Expand the Setup menu in the Ribbon to create a new NC program.



When a new NC program is created, the post properties can be found under Settings.



2.4 Post Processing: General Parameters

Output Format

GCode generates **.nc** files, and URScript generates **.script** files. The URScript format is compatible with both CB-Series and e-Series, whereas the G-code format is only compatible with the e-Series robots running the Remote TCP & Toolpath URCap. The output format determines which workflow to use on the UR teach pendant.

When generating a toolpath for e-Series robots, we highly recommend to use the G-code output format to take advantage of the unique features in the Remote TCP & Toolpath URCap:

- The toolpath can be executed with respect to a Remote TCP. The URCap will convert the toolpath into the Remote TCP frame automatically.
- In the Regular TCP mode, the robot is allowed to spin the end-effector freely around the Z-axis of the TCP. This feature helps avoid common issues such as singularity and joint limit.

Please refer to the appendix on how to activate the Remote TCP & Toolpath URCap on e-Series robots.

Output File

It is possible to write all the toolpaths in one **Single** file or split them into **Multiple** files, one for each individual toolpath operation.

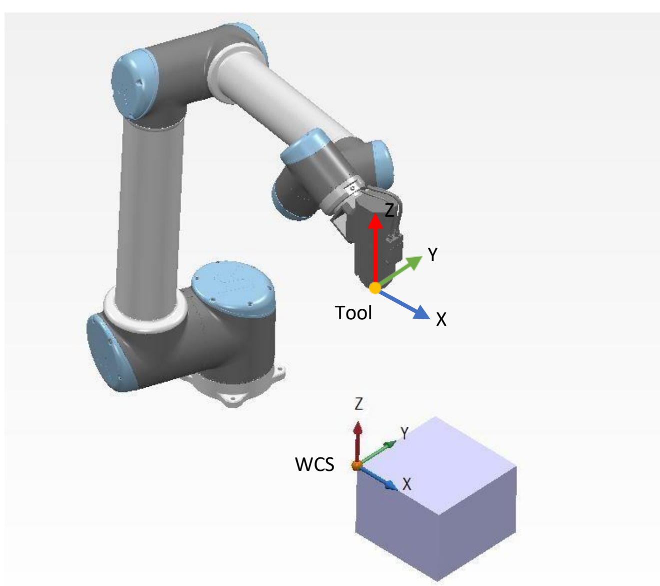
Use Subfolder

Default option is to write out all data in a subfolder named after the NC program.

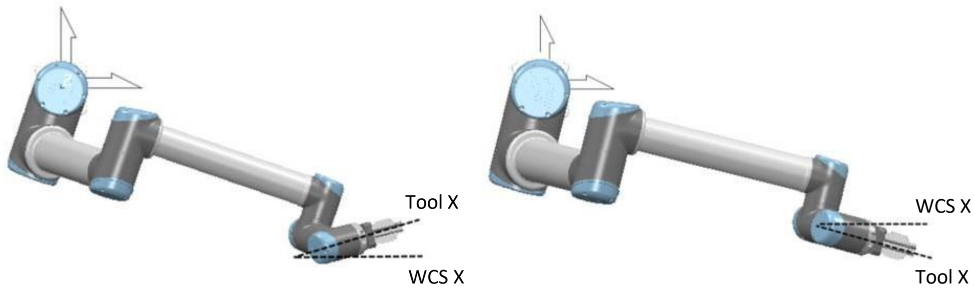
Robot Head Angle

This option allows the user to enter an angle of rotation around the Z-axis of tool. This angle is defined relative to the X-axis of the WCS on the part. This angle will be kept throughout the entire toolpath to avoid singularity.

In the illustration below, the robot head angle is set to zero, and thus the X-axis of the tool and the X-axis of the WCS are parallel to each other.



Below you can see examples of both +15 degree (left) and -15 degree (right) head angles relative to the X-axis of the WCS. Please note these are top views, and the Z-axis of the tool is pointing out of the page.

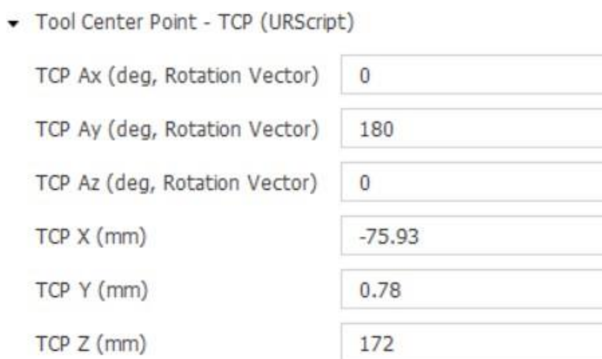


2.5 Post Processing: URScript Parameters

Below is a list of properties applicable to the URScript format only.

Tool Center Point (TCP)

Here you can define the TCP to be written in URScript files.



Tool Center Point - TCP (URScript)

TCP Ax (deg, Rotation Vector)	0
TCP Ay (deg, Rotation Vector)	180
TCP Az (deg, Rotation Vector)	0
TCP X (mm)	-75.93
TCP Y (mm)	0.78
TCP Z (mm)	172

The URScript file will contain the following command line.

```
# Set TCP
set_tcp(p[-0.075930,0.000780,0.172000,0,3.141592653589793,0])
```

If all these values are set to 0 (as by default), the TCP command line will be disabled as a comment.

```
# Set TCP
# set_tcp(p[0.000000,0.000000,0.000000,0.000000000,0.000000000,0.000000000])
```

Part Coordinate System (PCS)

Here you can define the Part Coordinate System (PCS) to be written in URScript files.

▾ Part Coordinate System - PCS (URScript)

PCS Ax (deg, Rotation Vector)	0
PCS Ay (deg, Rotation Vector)	0
PCS Az (deg, Rotation Vector)	180
PCS X (mm)	-243.34
PCS Y (mm)	86.05
PCS Z (mm)	-10.62

The URScript file will contain the following command line.

```
# Set Part Coordinate System
global Ref_frame = p[-0.243340,0.086050,-0.010620,0,0,3.141592653589793]
```

If all these values are set to 0 (as by default), the PCS command line will be disabled as a comment.

```
# Set Part Coordinate System
# global Ref_frame = p[0.000000,0.000000,0.000000,0,0,0]
```

Toolpath Approach Pose

Here you can define the Toolpath Approach Pose in joint angles (degrees) to be written in URScript files.

▾ Toolpath Approach Pose (URScript)

Base (deg)	196.77
Shoulder (deg)	-91.76
Elbow (deg)	-55.92
Wrist 1 (deg)	-124.49
Wrist 2 (deg)	90.59
Wrist 3 (deg)	263.28

The URScript file will contain the following command line.

```
# Toolpath Approach Pose
movej([3.43428437,-1.60151412,-0.97598812,-2.17276039,1.58109377,4.59510285],a=acc_1,v=vel_1,r=rad_smooth_1)
```

If all these values are set 0 (as by default), the Toolpath Approach Pose command line will be disabled as a comment.

```
# Toolpath Approach Pose
# movej([0.00000000,0.00000000,0.00000000,0.00000000,0.00000000,0.00000000],a=acc_1,v=vel_1,r=rad_smooth_1)
```

Motion Parameters (URScript)

Here you can define four motion parameters.

Motion Parameters (URScript)

Robot acceleration (mm/s^2)	1000
Robot movej tool speed (mm/s)	200
Robot payload (kg)	0
Robot radius smoothing (mm)	0.005

In the URScript file, **Robot acceleration**, **Robot movej tool speed** and **Robot radius smoothing** are defined as named variables.

```
# Motion Settings
acc_1 = 1 # Robot acceleration
vel_1 = 0.2 # Robot movej tool speed
fed3_1 = 0.0388 # Finish
fed9_1 = 0.0388 # Plunge
fed10_1 = 0.0166667 # High Feed-rapid
rad_smooth_1 = 0.000005 # Robot radius smoothing
```

The payload is defined as follows.

```
# Set Payload
set_payload(0.38)
```

If this value is set to 0 (as by default), the Payload command line will be disabled as a comment:

```
# Set Payload
# set_payload(0)
```

Motion Parameters (GCode)

Here you can define two motion parameters for GCode only.

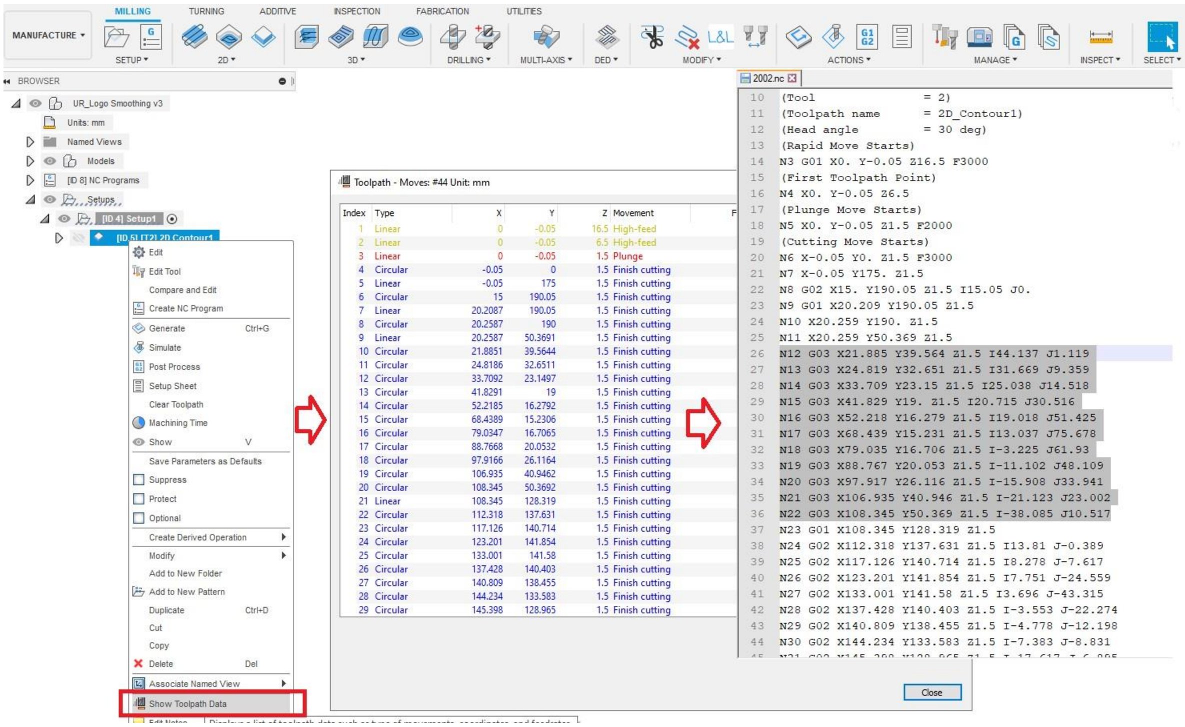
Motion Parameters (GCode)

Automatic toolpath splitting	☐
Use circular moves	☑

The **Automatic toolpath splitting** is related to Addittive toolpaths only. It allows to split toolpaths at each 'end-effector' command call. It could be necessary for old URCap versions.

For any further information on Additive please refer to section 2.6 below.

Use circular moves writes G02/G03 functions whether the toolpath has been calculated having circular movements:



Feedrates

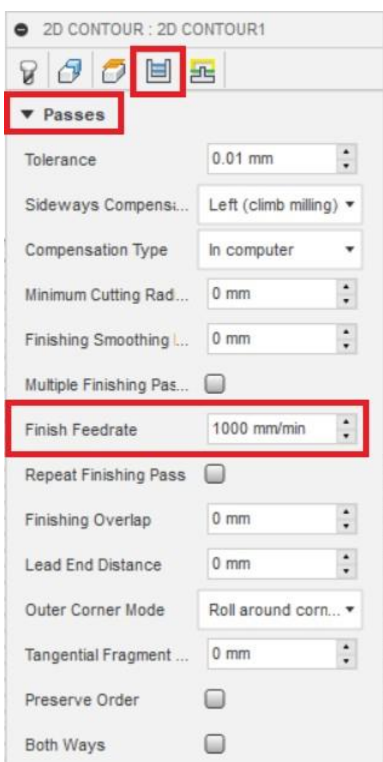
The feedrates defined in each operation will directly affect the output. They appear as named variables in URScript files as shown in the example below. You can modify these values directly from the teach pendant instead of going back to Fusion 360 to generate a new URScript file.

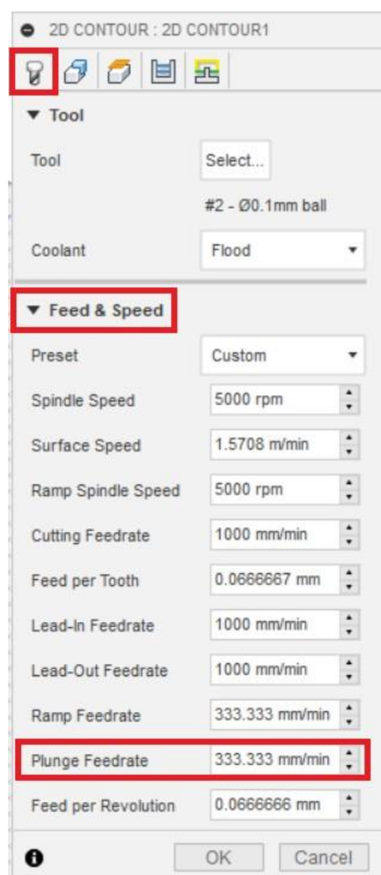
```
# Motion Settings
acc_1 = 1 # Robot acceleration
vel_1 = 0.2 # Robot movej tool speed
fed3_1 = 0.0388 # Finish
fed9_1 = 0.0388 # Plunge
fed10_1 = 0.0166667 # High Feed-rapid
rad_smooth_1 = 0.000005 # Robot radius smoothing
```

The table below presents a list of feedrates that may appear in URScript files based on which operations are used in Fusion 360.

Fusion 360 GUI		URScript Output
Feedrate Name	**Location**	**Comment**
Lead-in Feedrate	Tool > Feed & Speed	Lead-in
Lead-out Feedrate	Tool > Feed & Speed	Lead-out
Ramp Feedrate	Tool > Feed & Speed	Ramp
Cutting Feedrate or Links or Extension	Tool > Feed & Speed	Cutting
Plunge Feedrate	Tool > Feed & Speed	Plunge
Finish Feedrate	Passes > Passes	Finish
Reduced Feedrate	Passes > Feed Optimization	Reduced
High Feedrate or Rapid Feedrate	Linking > Linking	High Feed-rapid

In the example above, *fed3_1* represents the Finish feedrate, as indicated by the comment next to it. It corresponds to the Finish Feedrate setting in the Passes section under the Passes tab. Likewise, *fed9_1* represents the Plunge Feedrate in the Feed & Speed section under the Tools tab.





2.6 Post Processing: Additive Parameters

This option is for non-subtractive operations when the end-effector needs to be turned on/off.

Commands to control end-effectors are non-standard and thus must be customized.

The post-processor simply writes these lines as comments:

```
==> END EFFECTOR ON: DEFINE YOUR CODE HERE IN THE POST
```

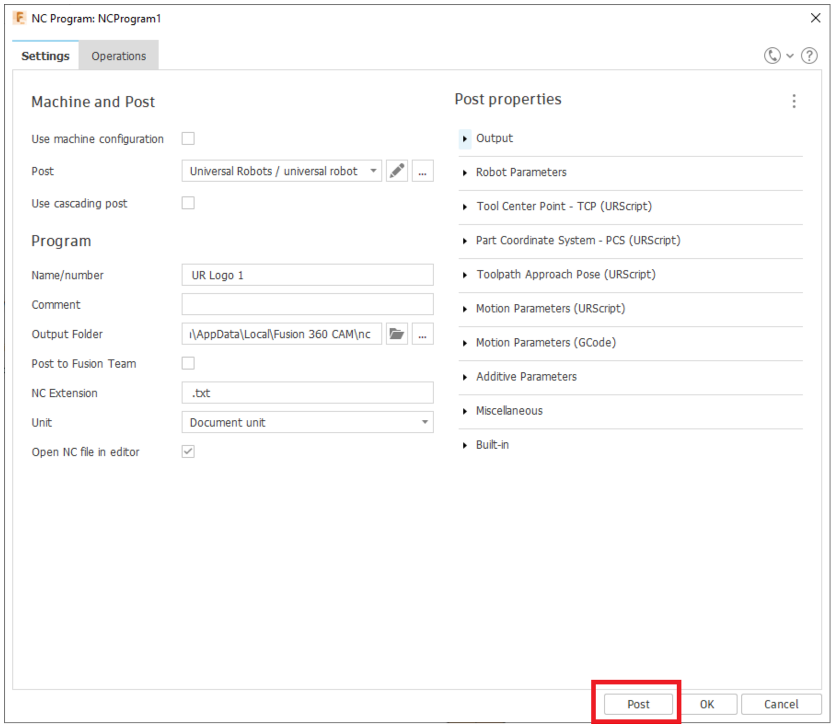
and

```
==> END EFFECTOR OFF: DEFINE YOUR CODE HERE IN THE POST
```

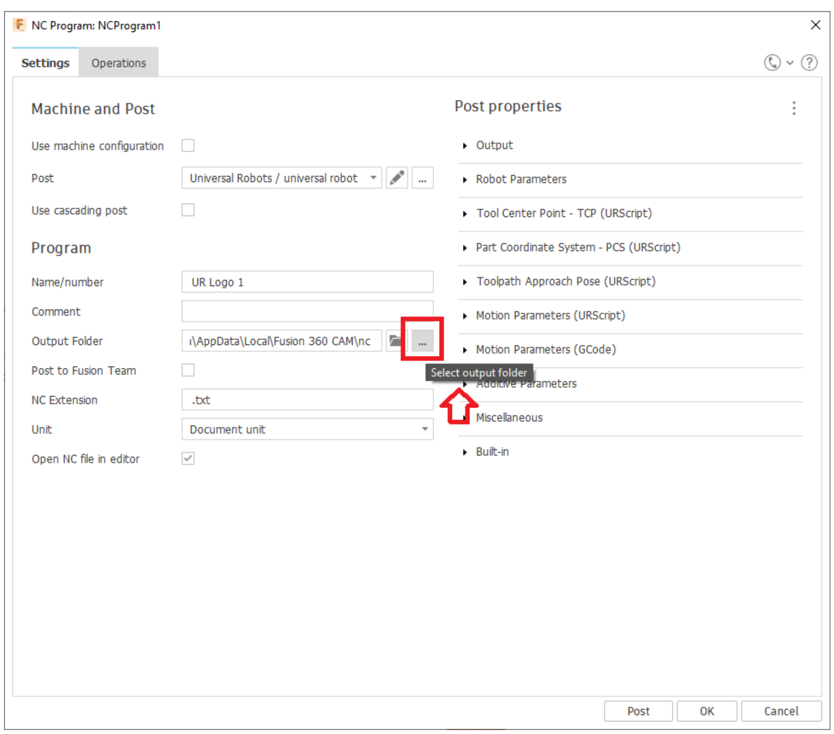
You can replace these lines with your custom code for end-effector control.

2.7 Posting

When you finish configuring the post properties in the NC program, click Post to generate the output toolpath files.



Please remember to select your proper local output folder.



A dummy file with standard information is always created after posting regardless of the property settings. It contains the name of the directory where you can find your NC files. The dummy file is named after the Program Name configured in the NC Program. Here is an example:

```
This is a dummy file.

Your program files are located here:
C:\Users\<Your_Username>\AppData\Local\Fusion 360 CAM\nc\1001
```

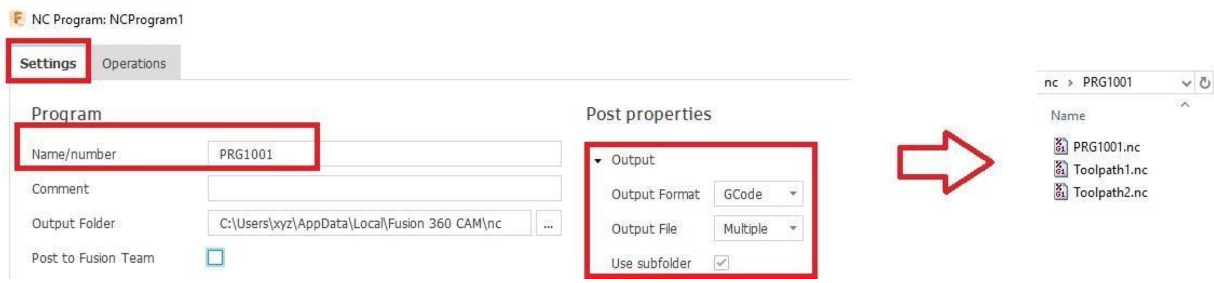
2.8 Sample Output in G-code

```
%
(G-code output for Universal Robots' Remote TCP & Toolpath URCap)
(Generated by AUTODESK Fusion 360 CAM 2.0.9305)
(Post version: 42973)
(Creation date: Friday, October 30, 2020 16:43:06)
N1 G90
N2 G21
(Spindle Speed      = 5000 RPM)
(Tool               = 2)
(Toolpath name      = 2D_Contour1)
(Head angle         = 30 deg)
N3 G00 X0. Y-0.05 Z16.5 F100
(First Toolpath Point)
N4 X0. Y-0.05 Z6.5
(Plunge Move Starts)
N5 G01 X0. Y-0.05 Z2.5 F333.333
N6 X0. Y-0.05 Z1.5
(Cutting Move Starts)
N7 X-0.05 Y0. Z1.5 F1000
N8 X-0.05 Y175. Z1.5
N9 G17 G02 X15. Y190.05 I15.05 J0.
…

…

N884 X47.347 Y48.503 Z1.5
N885 X47.214 Y49.039 Z1.5
N886 X47.214 Y49.041 Z1.5
N887 X47.095 Y49.595 Z1.5
N888 X47.095 Y49.596 Z1.5
N889 X46.991 Y50.169 Z1.5
N890 X46.991 Y50.17 Z1.5
N891 X46.902 Y50.76 Z1.5
N892 X46.901 Y50.768 Z1.5
N893 X46.901 Y190. Z1.5
(Rapid Move Starts)
N894 G00 X46.901 Y190. Z16.5 F100
N895 M30
%
```

If you choose to output multiple files, you will find one file containing the Multiple File List and the tools information. It is named after the Program Name in the NC Program.



The PRG1001.nc file contains the following content:

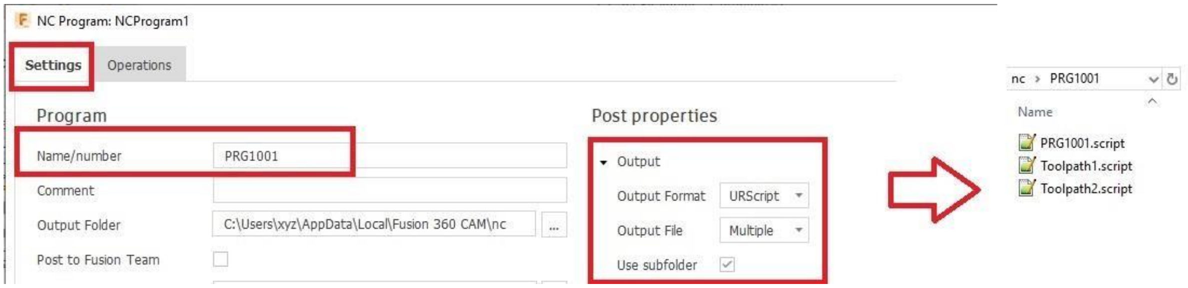
```
(Multiple Programs list)
(TOOL 1)
(Toolpath1)
(TOOL 2)
(Toolpath2)
```

2.9 Sample Output in URScript

```
#
# Generated by AUTODESK Fusion 360 CAM 2.0.9305
# Post version: 42973
# Creation date: Friday, October 30, 2020 15:34:26
#
# Toolpath Name = 2D_Contour1
# Head angle    = 30 deg
#
# Set TCP
set_tcp(p[-0.075930,0.000780,0.172000,0,3.141592653589793,0])
#
# Set Part Coordinate System
global Ref_frame = p[-0.243340,0.086050,-0.010620,0,0,3.141592653589793]
#
# Set Payload
set_payload(0.38)
#
# Spindle Speed      = 5000 RPM
# Tool               = 2
# Motion Settings
acc_1 = 1 # Robot acceleration
vel_1 = 0.2 # Robot movej tool speed
fed3_1 = 0.0388 # Finish
fed9_1 = 0.0388 # Plunge
fed10_1 = 0.0166667 # High Feed-rapid
rad_smooth_1 = 0.000005 # Robot radius smoothing
```

```
#
# Toolpath Approach Pose
movej([3.43428437,-1.60151412,-0.97598812,-
2.17276039,1.58109377,4.59510285],a=acc_1,v=vel_1,r=rad_smooth_1)
# First Toolpath Point
movel(pose_trans(Ref_frame,p[0.000000,-
0.000050,0.016500,0.000000000,0.000000000,0.523598776]),a=acc_1,v=fed10_1,r=rad_smoot
h_1)
movel(pose_trans(Ref_frame,p[0.000000,-
0.000050,0.006500,0.000000000,0.000000000,0.523598776]),a=acc_1,v=fed10_1,r=rad_smoot
h_1)
# Plunge Move Starts
movel(pose_trans(Ref_frame,p[0.000000,-
0.000050,0.002500,0.000000000,0.000000000,0.523598776]),a=acc_1,v=fed9_1,r=rad_smooth
_1)
movel(pose_trans(Ref_frame,p[0.000000,-
0.000050,0.001500,0.000000000,0.000000000,0.523598776]),a=acc_1,v=fed9_1,r=rad_smooth
_1)
# Cutting Move Starts
movel(pose_trans(Ref_frame,p[-
0.000050,0.000000,0.001500,0.000000000,0.000000000,0.523598776]),a=acc_1,v=fed3_1,r=r
ad_smooth_1)
movel(pose_trans(Ref_frame,p[-
0.000050,0.175000,0.001500,0.000000000,0.000000000,0.523598776]),a=acc_1,v=fed3_1,r=r
ad_smooth_1)
movel(pose_trans(Ref_frame,p[0.000327,0.178349,0.001500,0.000000000,0.000000000,0.523
598776]),a=acc_1,v=fed3_1,r=rad_smooth_1)
…
…
movel(pose_trans(Ref_frame,p[0.046902,0.050760,0.001500,0.000000000,0.000000000,0.523
598776]),a=acc_2,v=fed3_2,r=rad_smooth_2)
movel(pose_trans(Ref_frame,p[0.046901,0.050768,0.001500,0.000000000,0.000000000,0.523
598776]),a=acc_2,v=fed3_2,r=rad_smooth_2)
movel(pose_trans(Ref_frame,p[0.046901,0.190000,0.001500,0.000000000,0.000000000,0.523
598776]),a=acc_2,v=fed3_2,r=rad_smooth_2)
# Rapid Move Starts
movel(pose_trans(Ref_frame,p[0.046901,0.190000,0.016500,0.000000000,0.000000000,0.523
598776]),a=acc_2,v=fed10_2,r=rad_smooth_2)
# Last Toolpath Point
end
```

If you choose to output multiple files, you will find one file containing the Multiple File List and the tools information. It is named after the Program Name in the NC Program.



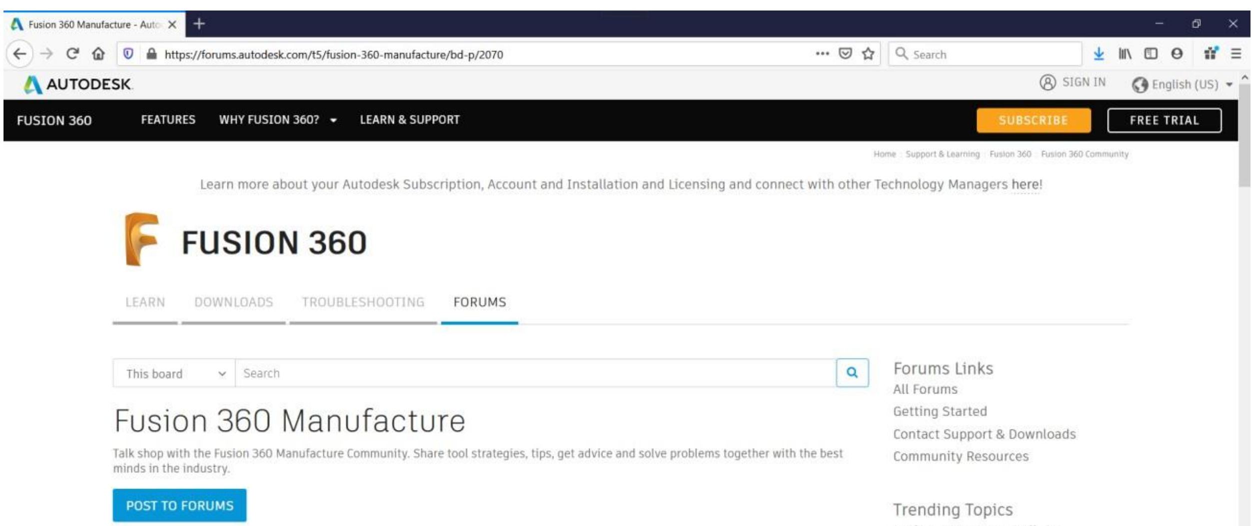
The PRG1001.script file contains the following content:

```
# Multiple Programs list
# TOOL 1
# Toolpath1
# TOOL 2
# Toolpath2
```

2.10 General Information

For more information or request help, please visit the [Autodesk Forum](#).

Select *Fusion 360* and then *Fusion 360 Manufacturing or HSM Post Processor Forum.*



If the Fusion 360 CAM session is running in inches, the output file for URScript format will still be written in mm. G-code format allows the use of both units.

IMPORTANT: Please note that program and toolpath names should not contain any symbol or special character.

3. Import Toolpath into Polyscope

The following items are required for the examples.

- A UR e-Series robot securely mounted on a table
    - A UR3e robot is used in the examples
- A dispensing head
    - Or any other pointed end-effector that fits on your robot
- A UR logo pattern printed on a Letter size paper
- A toolpath file corresponding to the UR logo pattern
    - The toolpath file is in the URScript (.script) or G-code (.nc) format

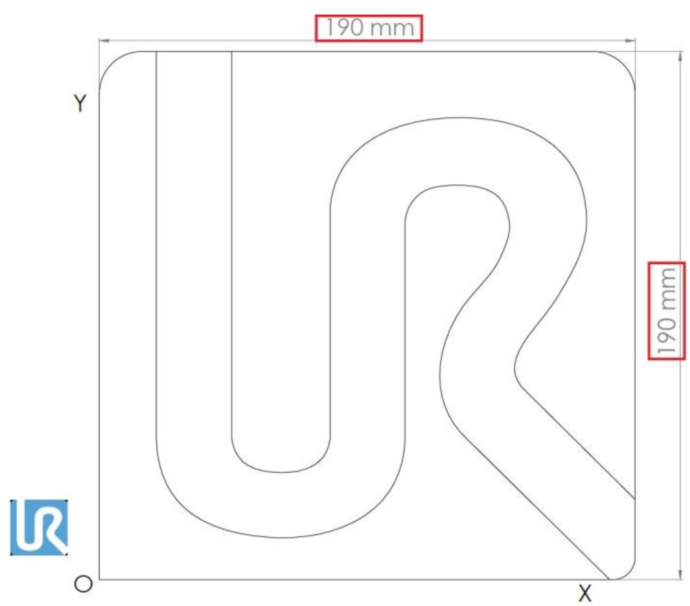
Please learn how to do the following before moving on. Refer to the Polyscope user manual from Universal Robots for detailed instructions.

- Configure a TCP
- Create and edit a Plane feature
- Set a MoveJ waypoint

3.1 Hardware Setup

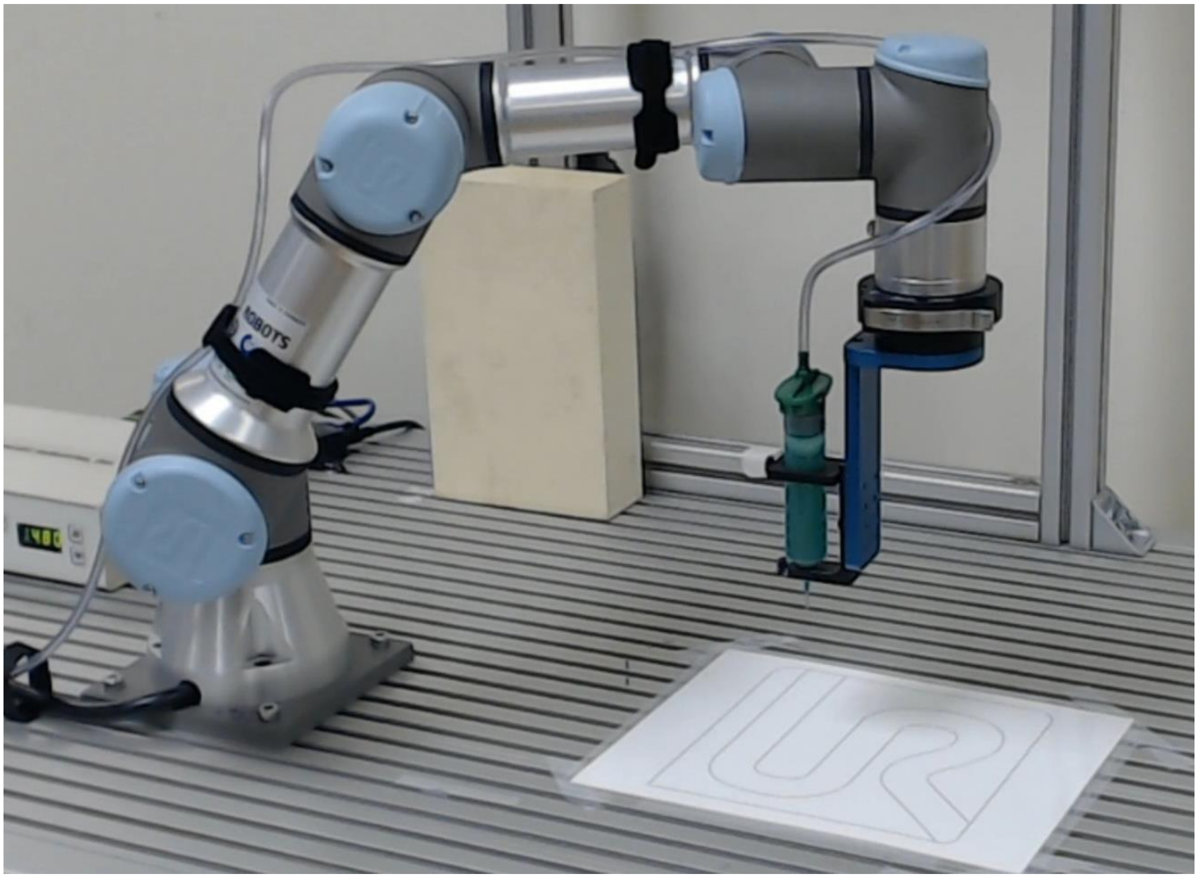
1. Mount your e-Series robot to a table securely.
    - Leave enough space for a Letter size paper next to the robot.

2. Print out the UR logo on a Letter size paper to its *actual* size.
    - Double check the major dimensions of the printed pattern match the marks.



3. Tape the UR logo sheet next to the robot base.
    - Make sure the whole sheet is within the reach of the robot.
    - In this example, the origin of the UR logo pattern is located at (-270.0, 86.05, -10.62) in the robot base frame.

4. Attach the dispensing head or your own pointed end-effector to the robot tool flange. The setup should look like this.

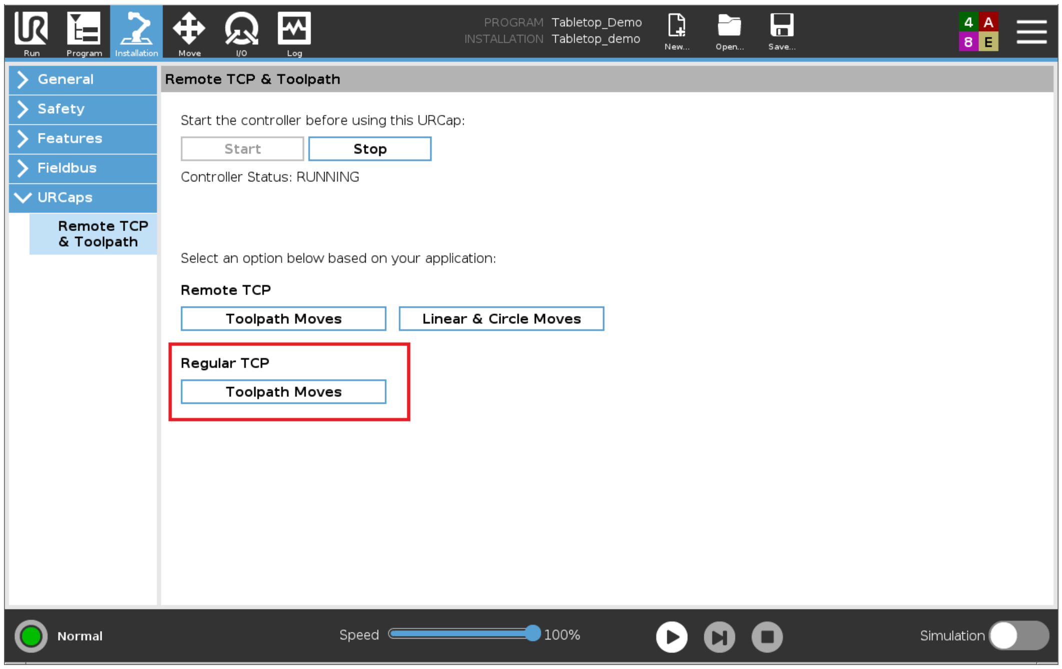


3.2 Import Toolpath in G-code Format

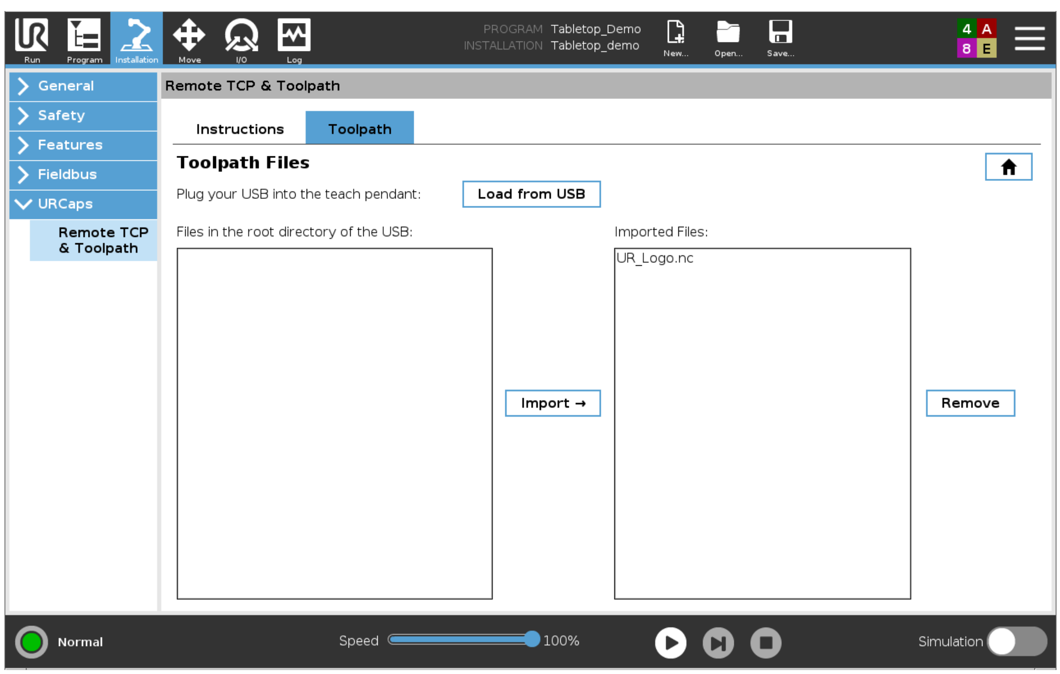
1. Only e-Series robots with the Remote TCP & Toolpath URCap can accept G-code toolpath files. Please make sure you have upgraded your e-Series robot to Polyscope 5.6 or above and activated the Remote TCP & Toolpath URCap.

   Please refer to the appendix on how to activate the Remote TCP & Toolpath URCap on e-Series robots.

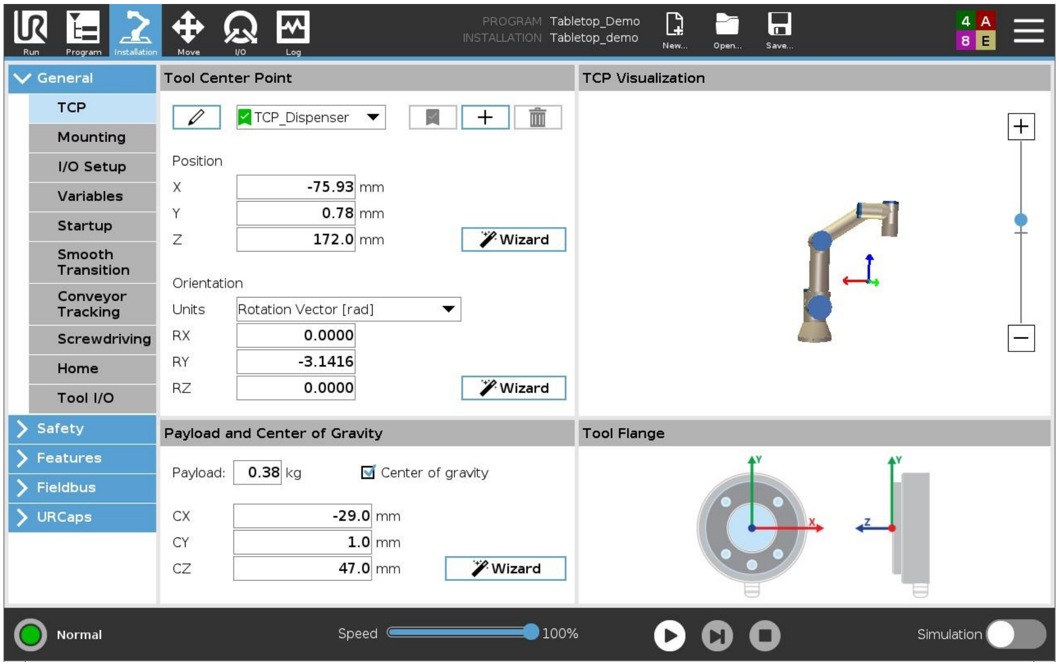
2. Select *Regular TCP Toolpath Moves* in the Remote TCP & Toolpath URCap and go through the embedded instructions to understand the workflow.



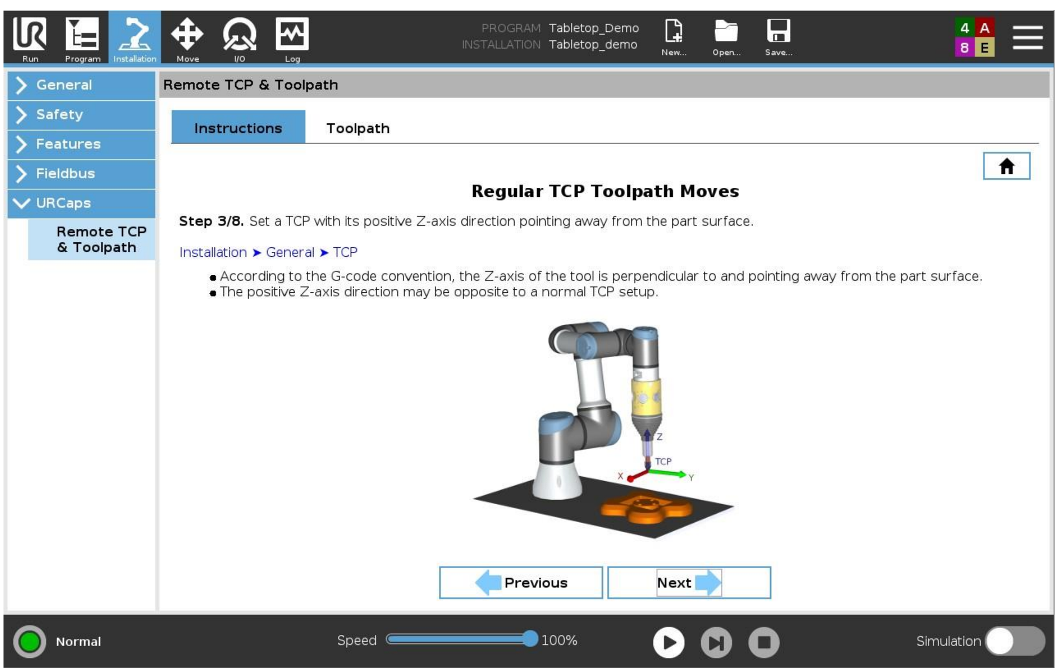
3. Save the G-code toolpath file named *UR_Logo.nc* to the root directory of your USB drive and import it into Polyscope.



4. Rename the active TCP to *TCP_Dispenser* and configure it using the wizard in Polyscope.

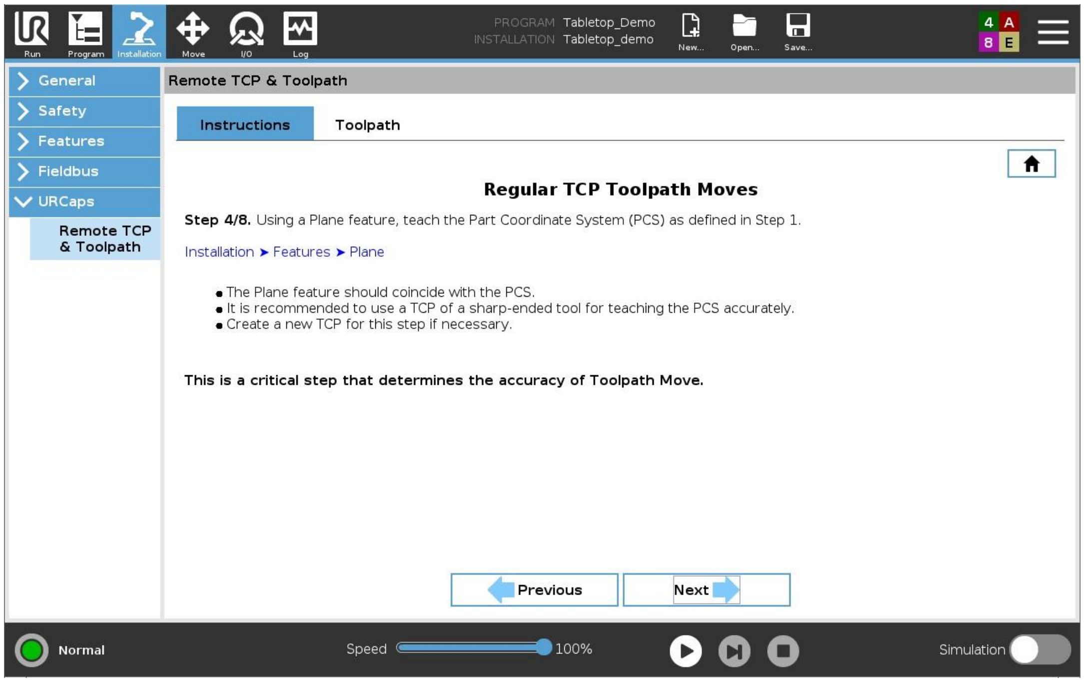


   a. Follow Step 3/8 in the embedded instructions. Make sure the Z-axis of the tool is perpendicular to and pointing away from the part surface.
   b. In this example, the Z-axis of the tool is pointing directly into the tool flange, so we set Ry to be -180 degrees (-3.1416 rad).

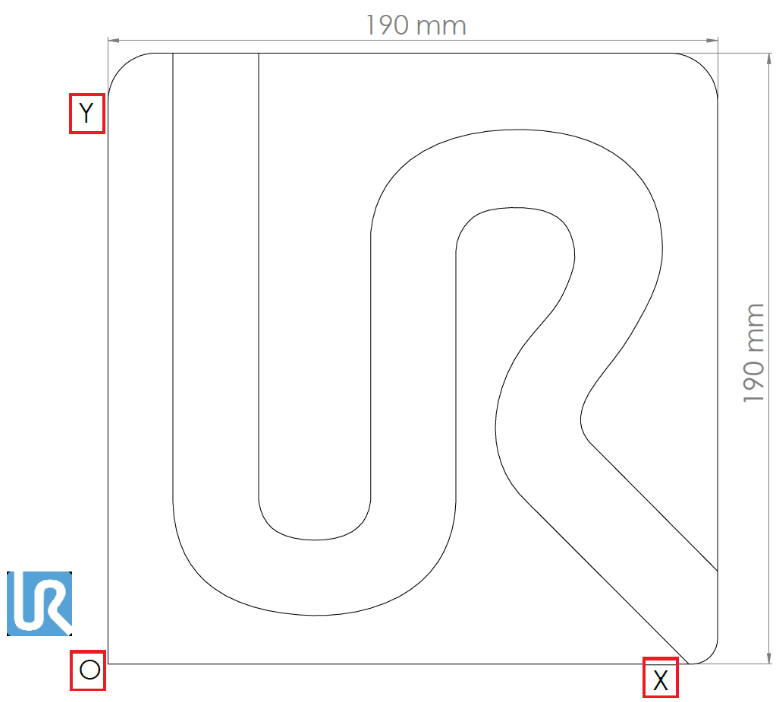


AUTODESK

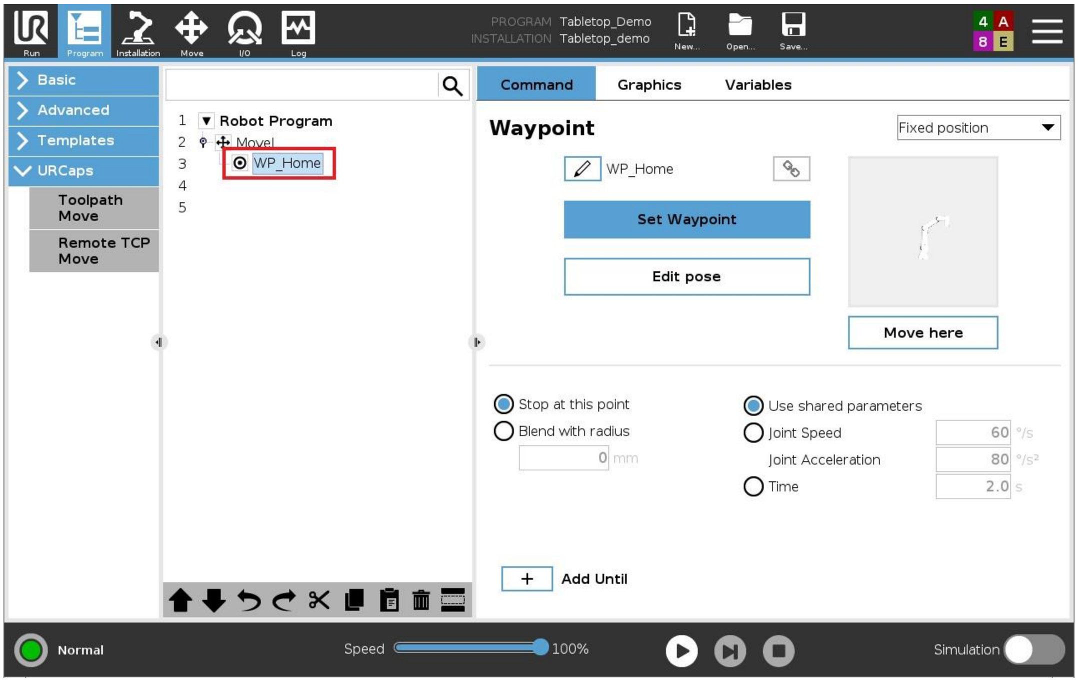
5. Create a Plane feature named *PCS_Logo*.
   a. Following Step 4/8 in the embedded instructions.



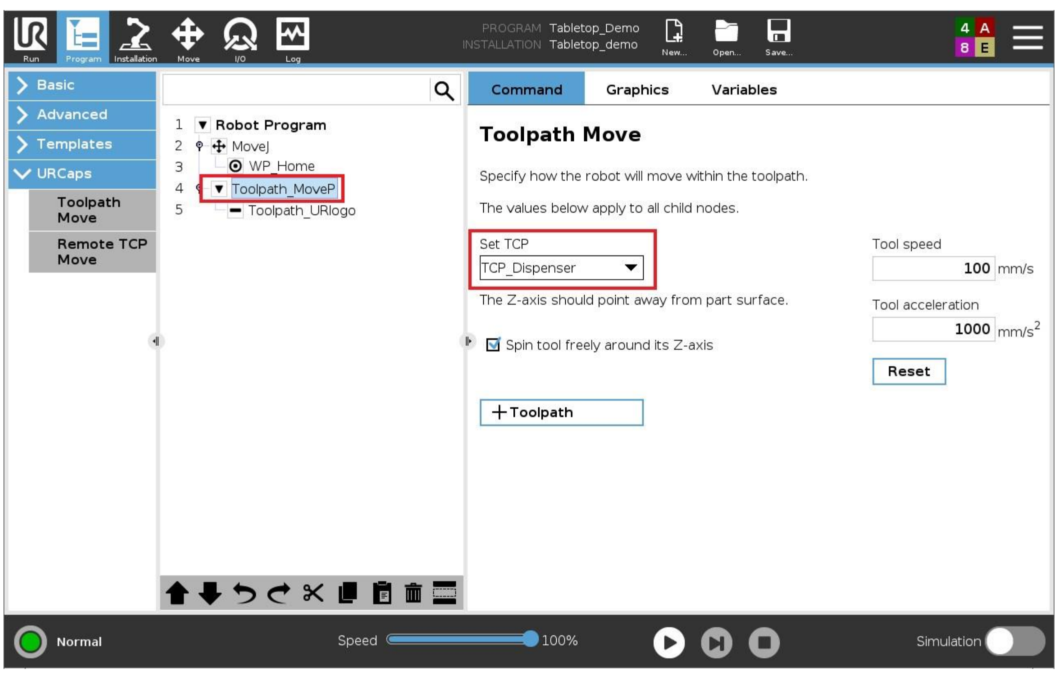
   b. The origin of the PCS is located at the bottom left corner of the UR logo. The X- and Y-axes are along the two edges.

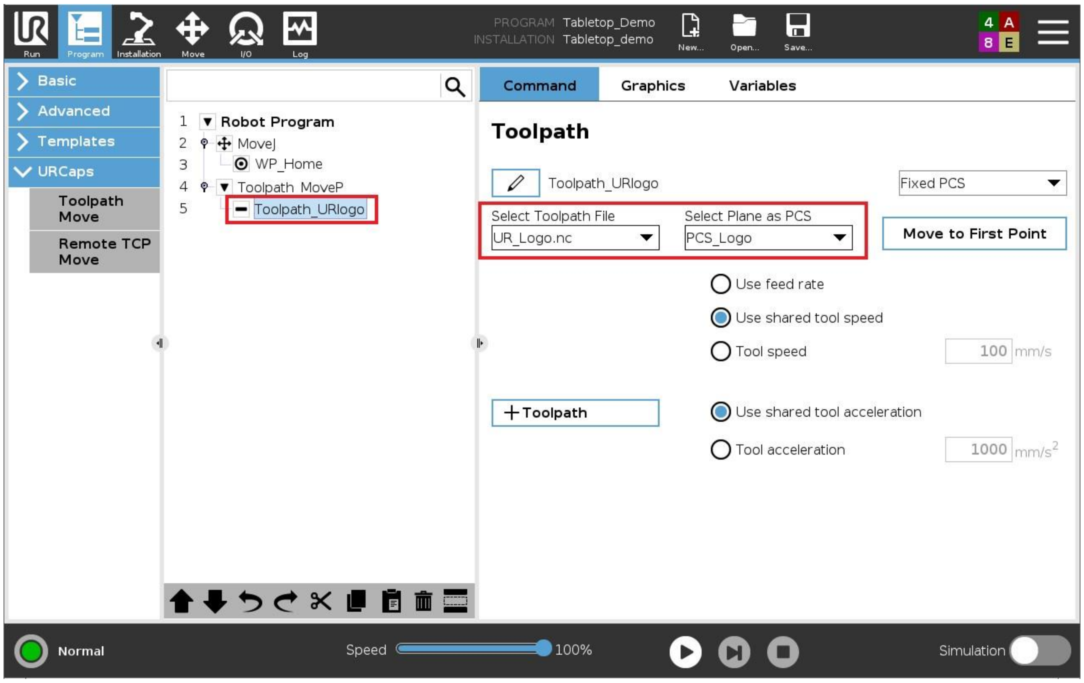


6. Go to Program. Create a MoveJ waypoint above the UR logo sheet. Rename the waypoint to *WP_Home*.



7. Click *Toolpath Move* under URCaps to create a Toolpath_MoveP node in the Program Tree. Rename its child node to *Toolpath_URlogo*. Select the TCP, Toolpath File, and PCS from the dropdown.





8. Run the program at a low speed to verify the motion is expected.

Reference Files:

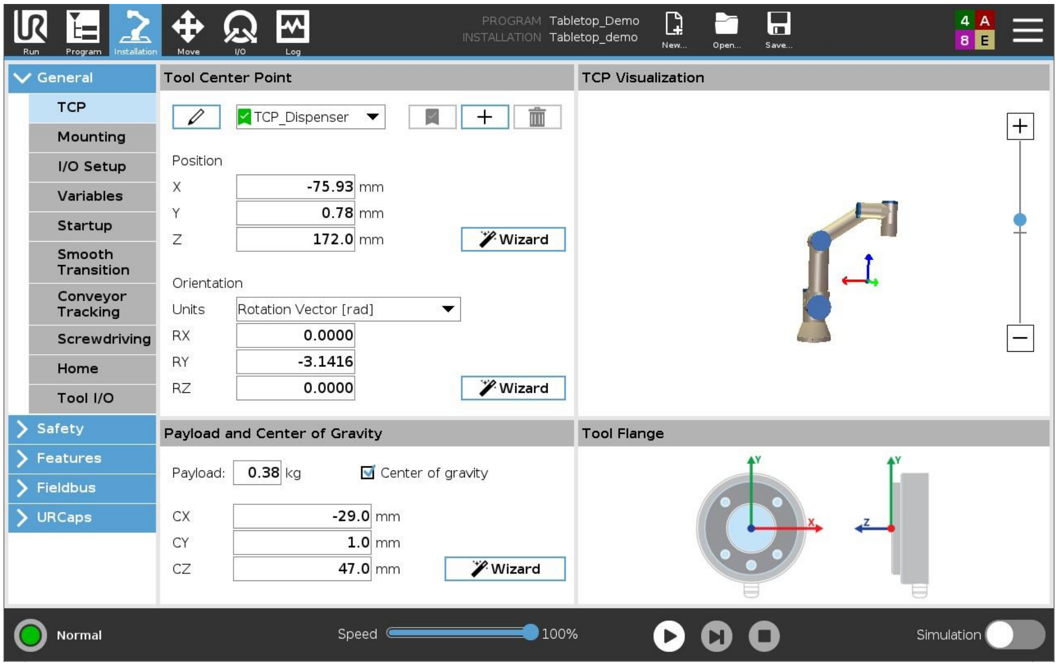
- Fusion 360 File: UR_Logo.f3d
- UR Logo Pattern: UR_Logo.PDF
- G-code Toolpath File: UR_Logo.nc
- Program File: Tabletop_Demo_GCode.urp
- Installation File: Tabletop_Demo_GCode.installation

3.3 Import Toolpath in URScript Format

A toolpath file in the URScript format can be loaded on either e-Series or CB-Series robot. Since the workflow is almost identical, we will continue using a UR3e robot in the following example.

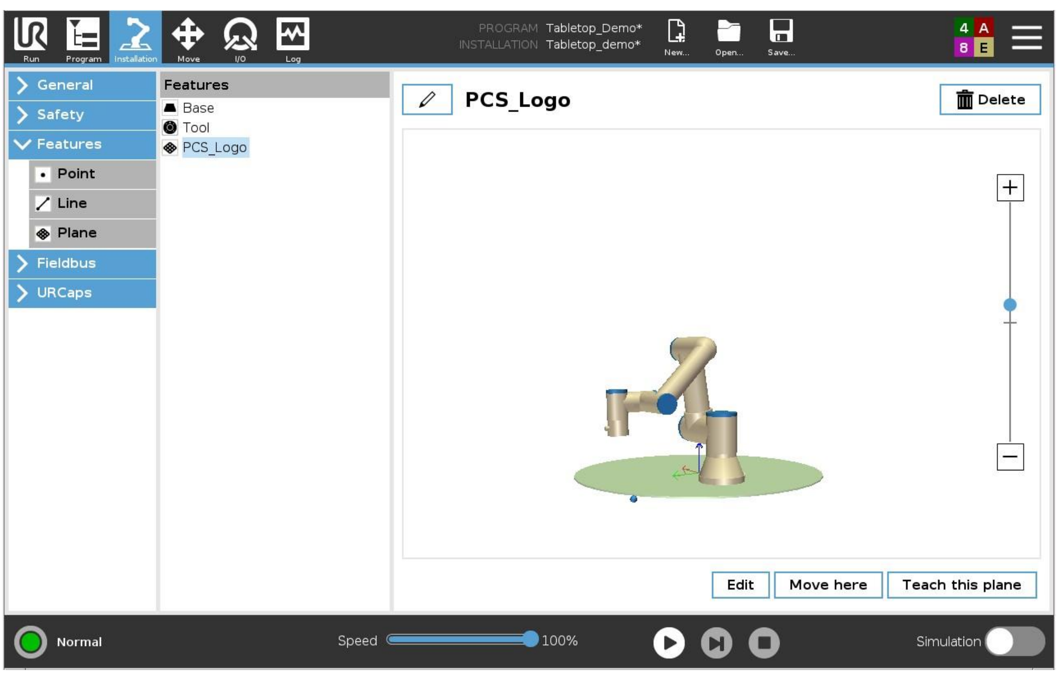
A toolpath file requires you to configure the Tool Center Point (TCP), payload, Part Coordinate System (PCS), and Toolpath Approach Pose. They can be defined either as variables in the URScript file or directly in the Program and Installation files.

1. Go to Installation. Rename the active TCP to *TCP_Dispenser* and configure it using the wizard in Polyscope. Also configure the payload and center of gravity using the corresponding wizard.

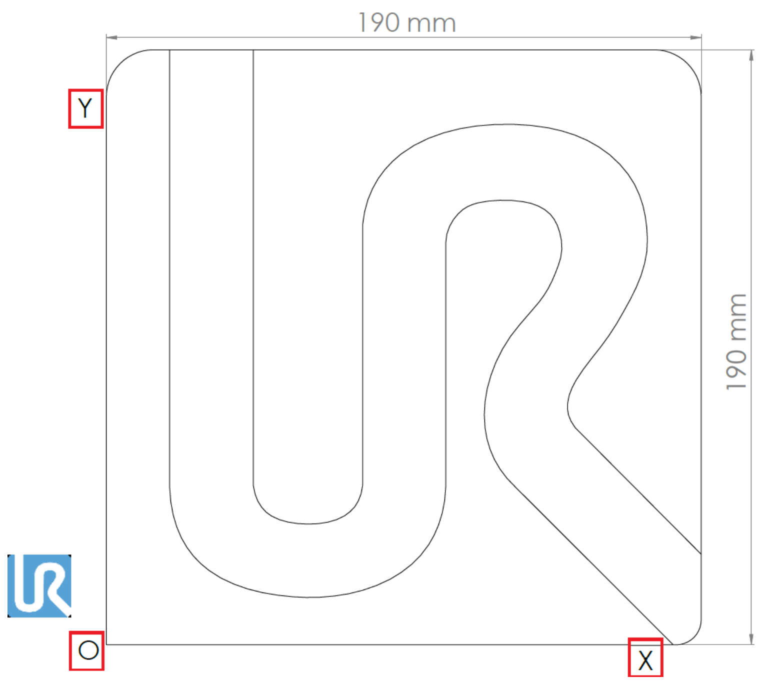


a. Make sure the Z-axis of the tool is perpendicular to and pointing away from the part surface.
b. In this example, the Z-axis of the tool is pointing directly into the tool flange, so we set Ry to be -180 degrees (-3.1416 rad).

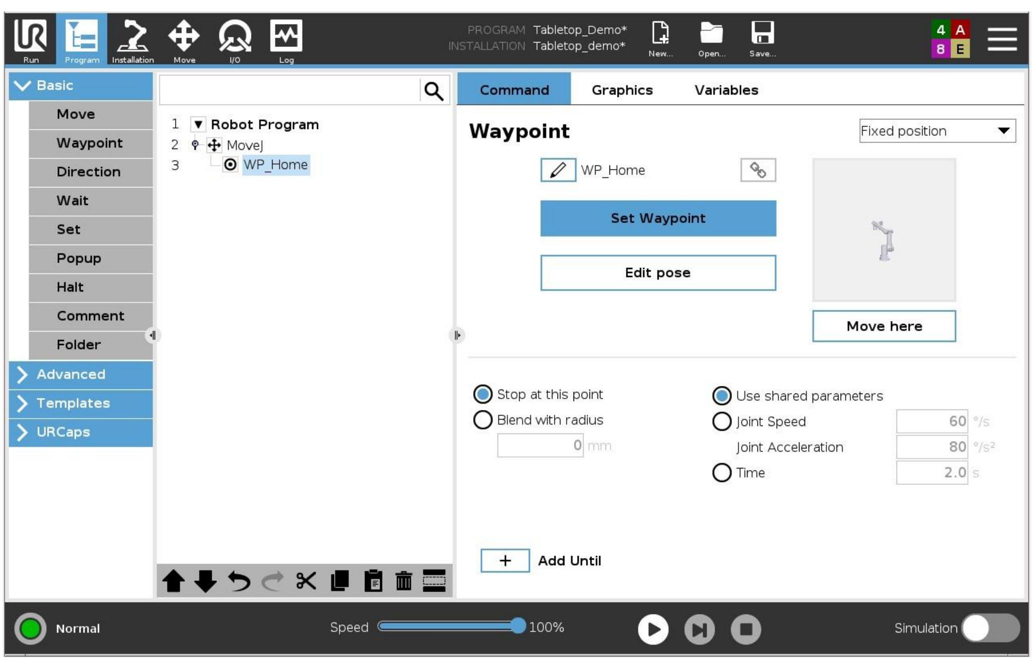
2. Create a Plane feature named *PCS_Logo*.



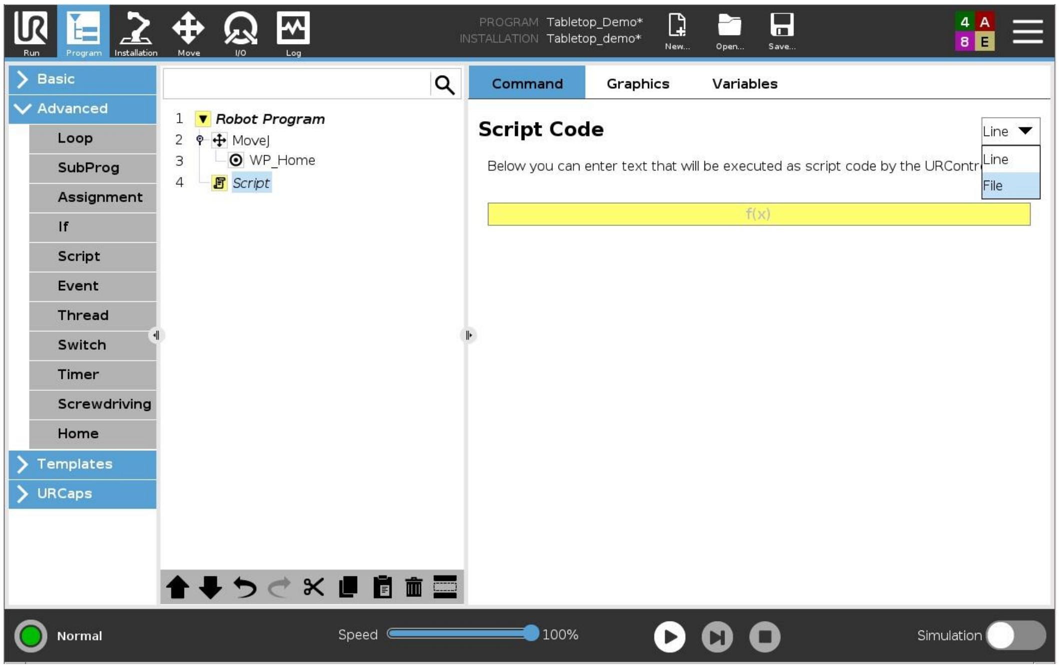
a. The origin of the PCS is located at the bottom left corner of the UR logo. The X- and Y-axes are along the two edges.



3. Go to Program. Create a MoveJ waypoint above the UR logo sheet. Rename the waypoint to *WP_Home*.

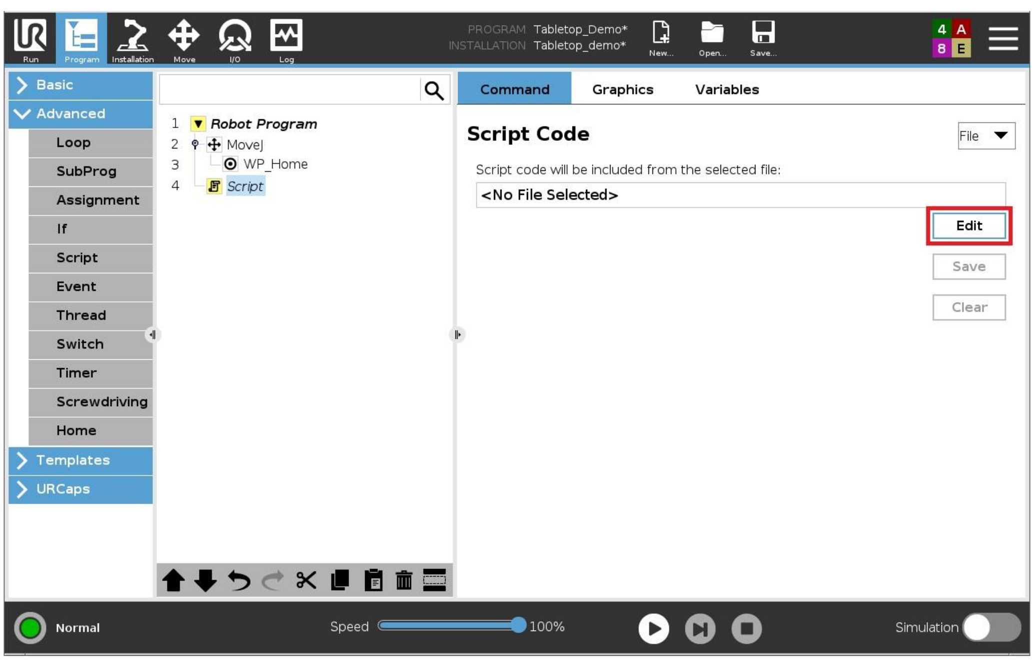


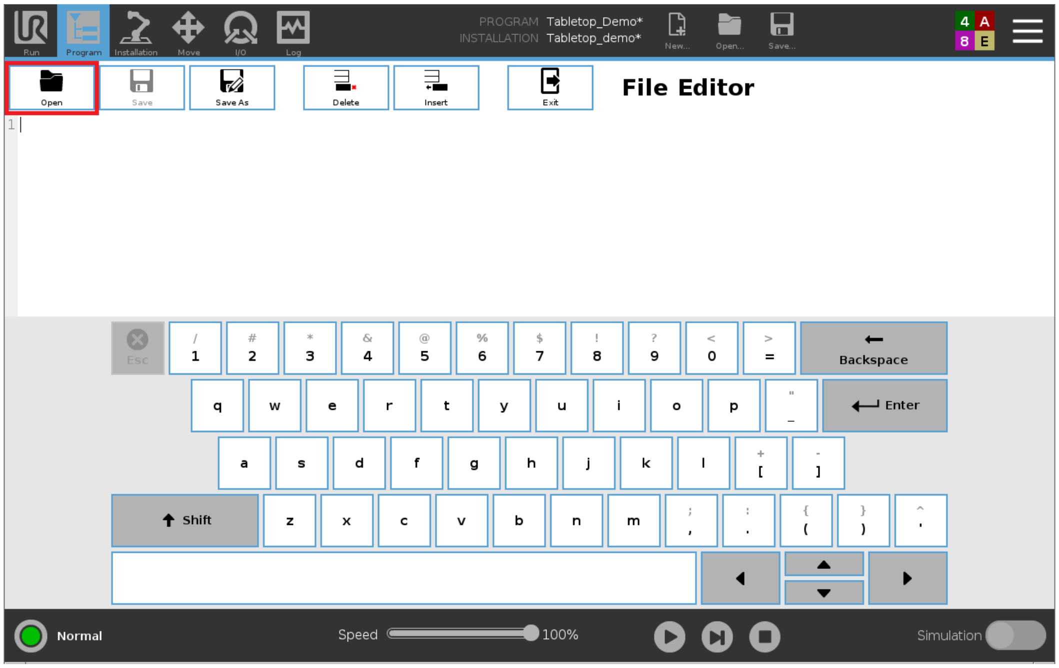
4. Insert a Script node under MoveJ and select File from the dropdown.

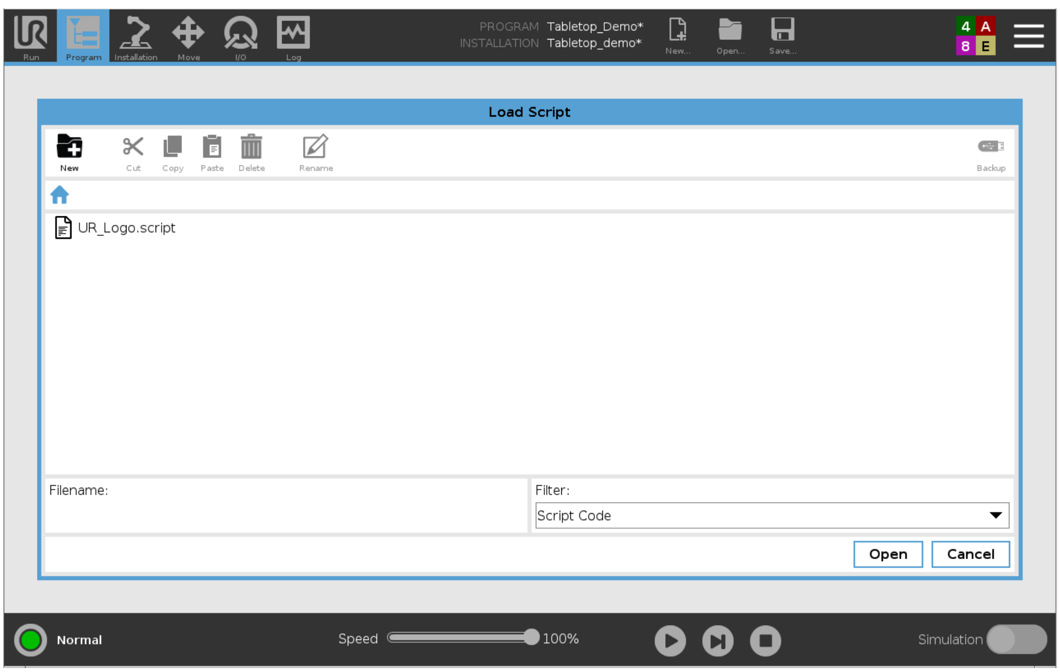


5. Save the URScript toolpath file named *UR_Logo.script* to your USB drive. Plug your USB drive to the USB port on the UR teach pendant.

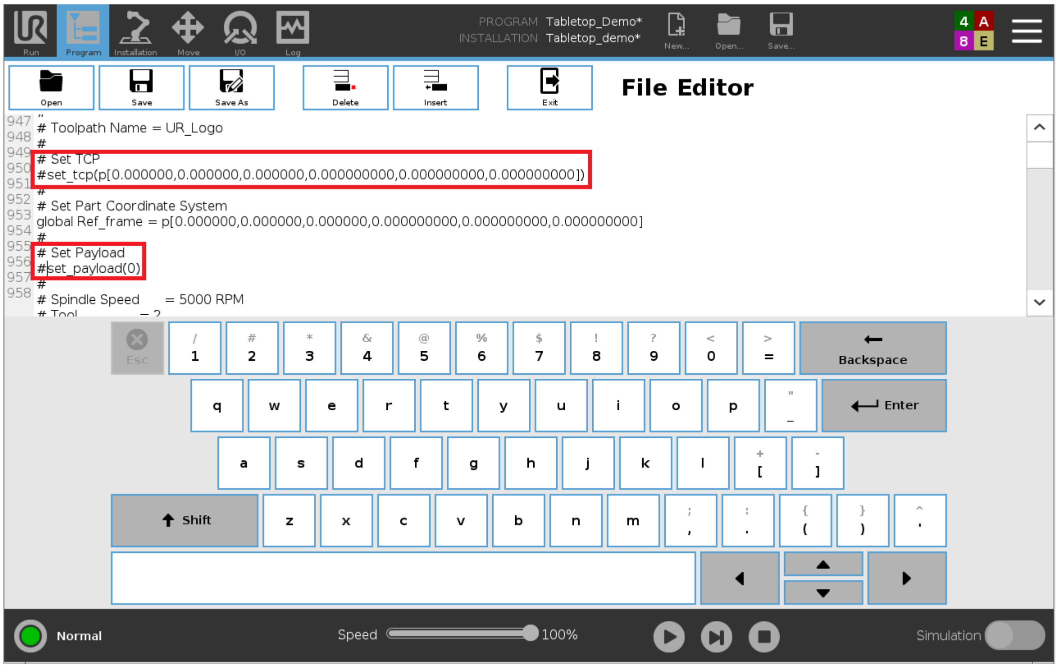
6. Tap Edit to open the File Editor. Tap Open to load the toolpath file from your USB drive. Make sure you save the file before exiting.



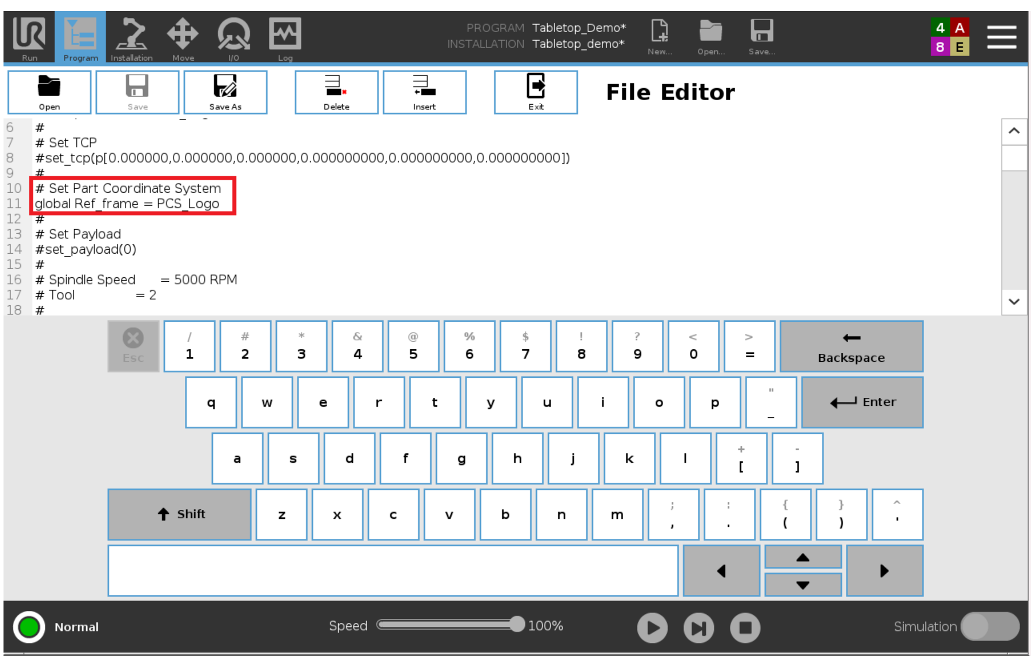




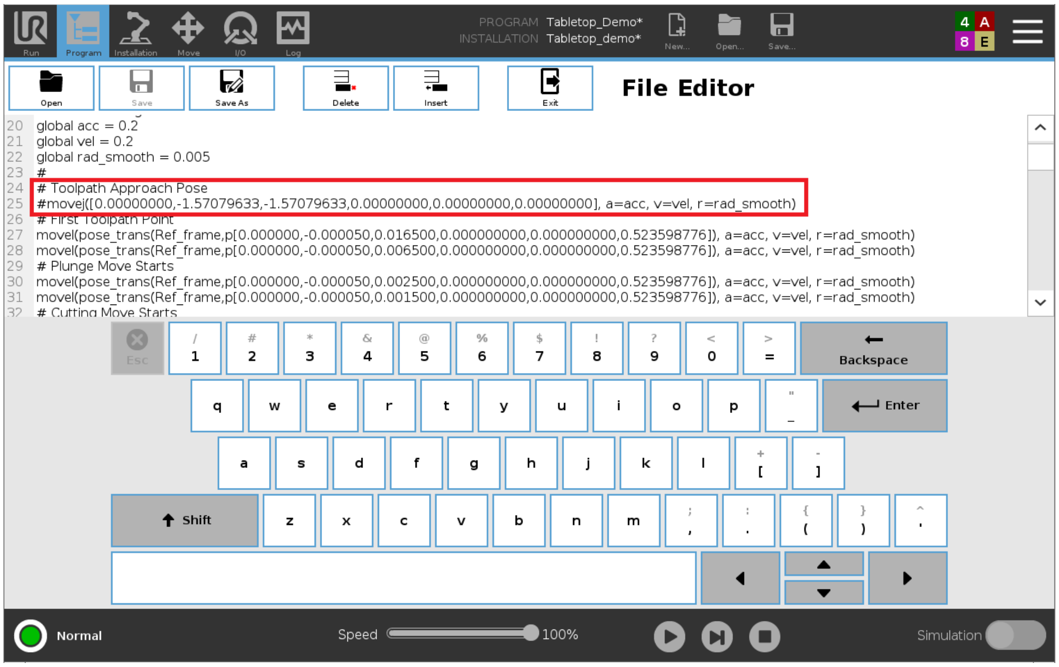
7. If you prefer to use the TCP, payload, and center of gravity defined in the Installation, please make sure the corresponding lines in the URScript file are commented out. If you do not comment them out, then these settings will be updated while the robot executes the Script node.



8. The Part Coordinate System (PCS) is defined in the URScript file by default. You can leave the script as is if the values you specified in the NC Program in Fusion 360 are correct. Alternatively, it can be linked to the Plane feature that you created in Step 4.
   a. Set *Ref_frame* to be *PCS_Logo*, the name of the Plane feature. Please note it is case sensitive.



9. The Toolpath Approach Pose is also defined in the URScript file by default. You can leave the script as is if the values you specified in the NC Program in Fusion 360 are correct. Alternatively, you can comment out this line, and the MoveJ waypoint above the Script node will be treated as the Toolpath Approach Pose.



10. Run the program at a low speed to verify the motion is expected.

Reference Files:

- Fusion 360 File: UR_Logo.f3d
- UR Logo Pattern: UR_Logo.PDF
- URScript Toolpath File: UR_Logo.script
- Program File: Tabletop_Demo_Script.urp
- Installation File: Tabletop_Demo_Script.installation

Appendix. Remote TCP & Toolpath URCap Activation

The Remote TCP & Toolpath URCap is provided to e-Series users at no additional cost. Please follow the instructions below to activate it on your e-Series robot.

- Upgrade your e-Series robot to Polyscope 5.6 or above. Please download the installer from [Universal Robots Support site](#).
- Follow the instructions to register the robot and activate the embedded Remote TCP & Toolpath URCap. (Hamburger Menu -> Settings -> System -> URCaps)
    - Step 1. Sign in at www.universal-robots.com/activate
    - Step 2. Download the registration file to your USB drive
    - Step 3. Plug your USB drive into the teach pendant and tap Activate to load the registration file
    - Step 4. Restart your robot to complete the activation process