IM467178-L

Learn the Rules...then Break Them: Fusion 360 for Industrial Robot Arms

Stefanie Pender
Autodesk

Learning Objectives

+ Quick introduction to Industrial Robots
+ Set-up manufacturing space in Fusion 360 to post-process native robot code for the application of milling
+ Test and validate posted robot code from Fusion 360 in a virtual robot simulator
+ Identify various coordinate systems of the industrial robot arm and calibrate tool data and work coordinate system
+ Demonstrate customer example utilizing workflow
+ Introduction to the Autodesk Technology Centers and access to Industrial Robot Arms

Description

Fusion 360 software now offers a simple workflow for programming industrial robot arms. Anyone familiar with post-processing machine code from Fusion 360 can now create robot code in the same way. During this lab, users will use the manufacturing space from Fusion 360 to generate robot native code that can drive industrial robot arms. First, we will show a generic robot workflow within Fusion 360 that will post with the ABB post processor. Then, we will cover exporting code into a virtual robot simulator, ABB RobotStudio, for validation. We'll then pivot to recordings at Autodesk Technology Centers, explaining calibration of various coordinate systems localizing the robot arm and robot operation for milling applications. We'll introduce other applications that participants can use with this workflow, such as welding and additive. Finally, we will share various case studies from Autodesk Technology Center residents and customers utilizing Fusion 360 for industrial robots.

Speaker

Stefanie Pender is a Senior Shop Supervisor at the Autodesk Technology Center Boston, a research and development workspace where Autodesk invites startups, colleges and universities, and industry experts to explore ways to advance the building industry. The center focuses on industrialized construction, digital fabrication, automation and robotics in construction, and other ideas that are transforming the built world. Stefanie supports residents utilizing robotics at the technology center in applications that range from milling, welding, 3D printing, cutting, and material handling. She is a certified ABB Robot Programmer for material handling.

THIS HANDS-ON LAB DOES NOT SUBSTITUTE IN-PERSON EQUIPMENT TRAINING.

IMPORTANT SAFETY INFORMATION

Before using an industrial robot make sure to follow the applicable safety instructions and guidelines.

Get informed on general safety hazards surrounding the use of industrial robots and prevent injuries by controlling associated risks.

+ The ABB posts provided are only compatible with Fusion 360. There are many factors that can cause a post to not be compatible with your particular setup. When you use a new post make sure to test it carefully as the post might not match your particular robot configuration and requirements.
+ Simulation must be performed using a Virtual Controller in proprietary software such as ABB RobotStudio

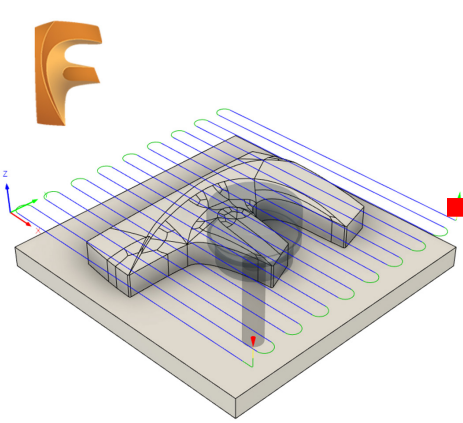
FOLLOW THE STANDARD! PLEASE CONSULT THE AMERICAN NATIONAL STANDARDS INSTITUTE ANSI/RIA R15.06-2012 AMERICAN NATIONAL STANDARD FOR INDUSTRIAL ROBOTS AND ROBOT SYSTEMS- SAFETY REQUIREMENTS BEFORE OPERATING ANY INDUSTRIAL ROBOT

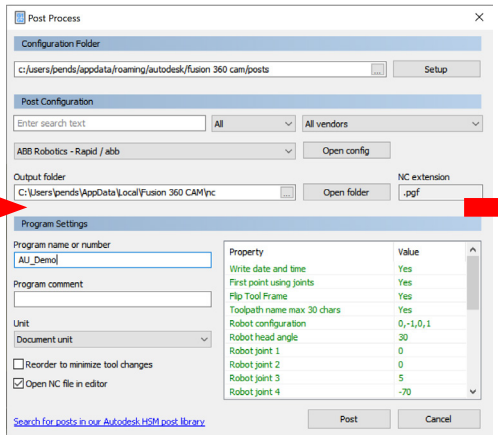
This is an image of our robot cell.
For milling applications we utilize polycarbonate hard guarding and have interlocked doors. No one is allowed to enter the cell during a milling operation. If anyone tries to enter the cell during a milling operation, the spindle automatically turns off.

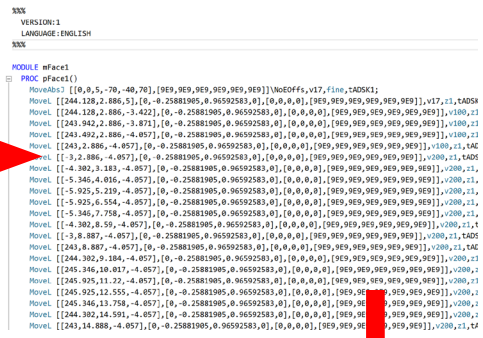
DO NOT mill without the appropriate safety systems in place.

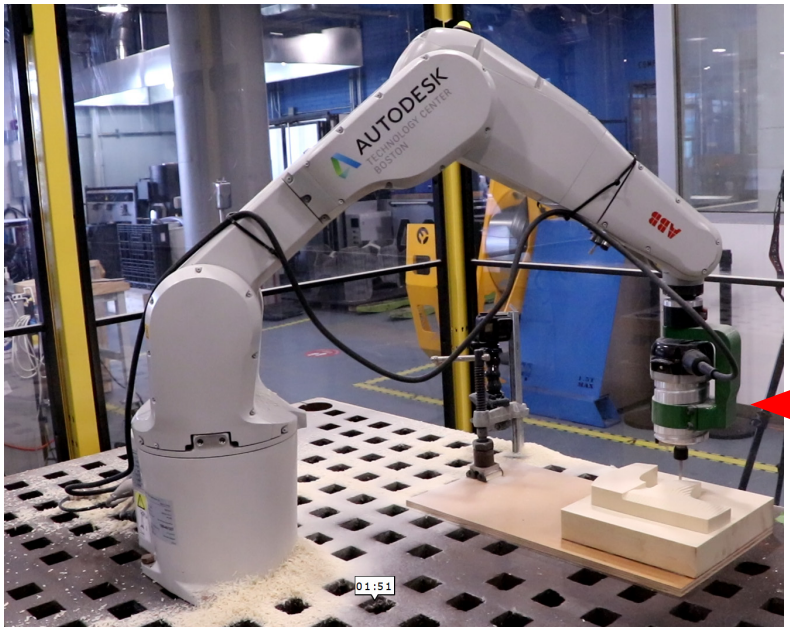
Using this handout:

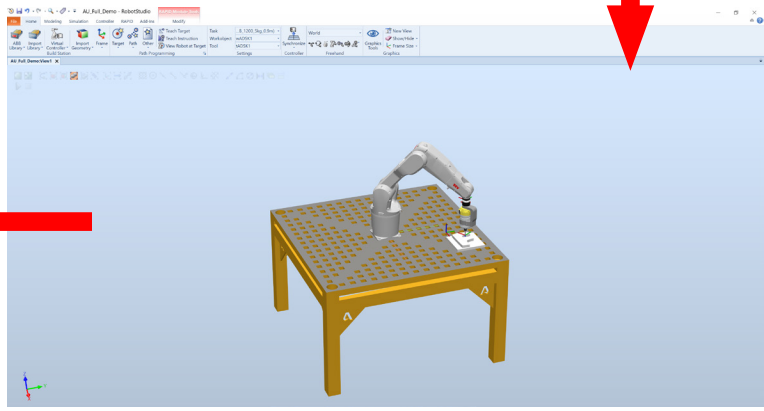
+ We will setup toolpaths in Fusion 360
+ Post-process those toolpaths
+ Validate toolpaths and simulate an industrial robot in ABB RobotStudio
+ Set up coordinate systems on an ABB Robot
+ Mill using an ABB Industrial Robot.











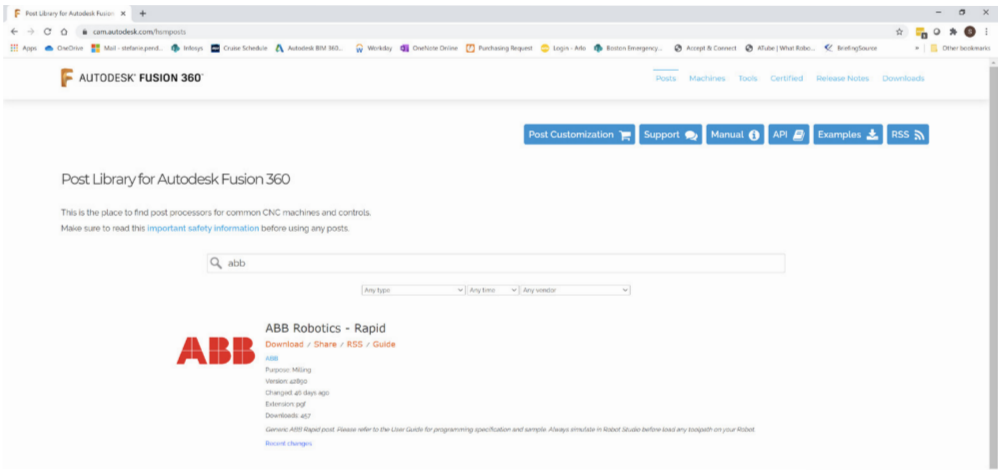
Equipment used:

+ ABB IRB 1200 (5 kg/0.9 m reach) with IRC5 controller
+ Dewalt hand router with custom mount
+ End mill
+ Polyurethane foam

DOWNLOADING/INSTALLING POST INTO FUSION 360

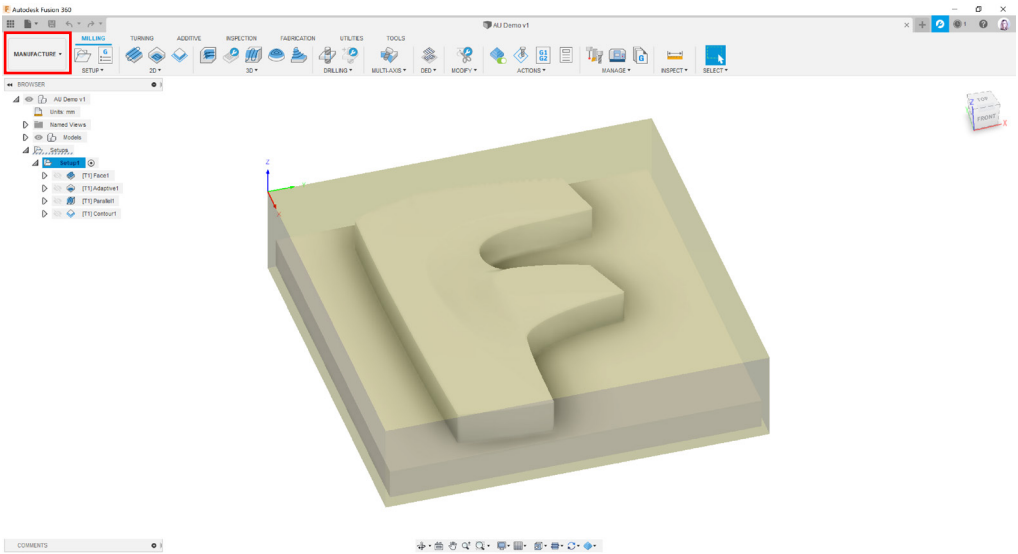
Go to: *https://cam.autodesk.com/hsmposts*

1. Search for **ABB**.
2. Download latest **ABB post**.

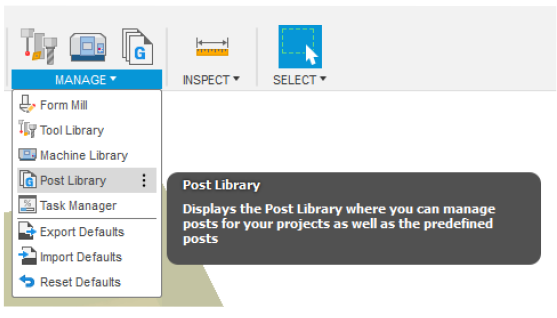


Fusion 360: Open file *AU Demo*

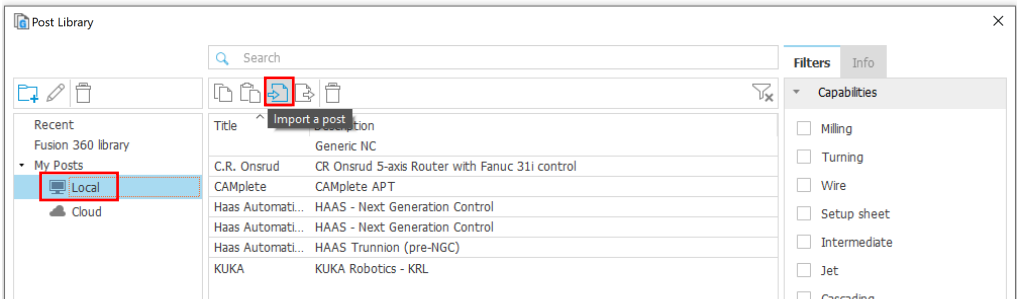
1. Under **File** in the toolbar, choose **Open**.
2. Make sure you are in the **Manufacture Workspace**.



3. Under **Manage** → Go to **Post Library**



DOWNLOADING/INSTALLING POST INTO FUSION 360



4. Select **Local** → Go to **Import a post**
5. Select **abb.cps** from **Downloads** folder
6. ABB post should now be in the Local folder.

Title	Description
	Generic NC
ABB	ABB Robotics - Rapid
C.R. Onsrud	CR Onsrud 5-axis Router with Fanuc 31i control
CAMplete	CAMplete APT
Haas Automati...	HAAS - Next Generation Control
Haas Automati...	HAAS - Next Generation Control
Haas Automati...	HAAS Trunnion (pre-NGC)
KUKA	KUKA Robotics - KRL

To post-process for Kuka:

1. Go to: *https://cam.autodesk.com/hsmposts*
2. Search for Kuka or Universal Robots and download appropriate post file.
3. Follow the steps 3-5 above and import the appropriate post into your local post folder.
4. There are also **Post-Processor Guides** offered on the Autodesk post site**.**

ABB Robotics - Rapid

Download / Share / RSS / Guide

ABB

Purpose: Milling
Version: 42890
Changed: 46 days ago
Extension: pgf
Downloads: 365

Generic ABB Rapid post. Please refer to the User Guide for programming specification and sample. Always simulate in Robot Studio before load any toolpath on your Robot.

KUKA Robotics - KRL

Download / Share / RSS / Guide

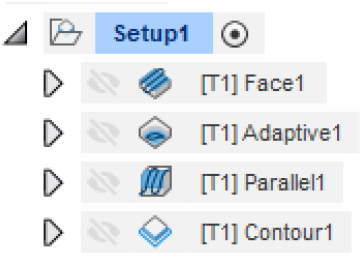
KUKA

Purpose: Milling
Version: 42890
Changed: 47 days ago
Extension: src
Downloads: 185

Generic KUKA KRL post. Please refer to the User Guide for programming specification and sample. Always validate with KUKA.Sim before load any toolpath on your Robot.

PREPARING TOOLPATHS FOR POST PROCESSING

The prepared Fusion 360 file *AU Demo* has all of the appropriate toolpaths for this training.

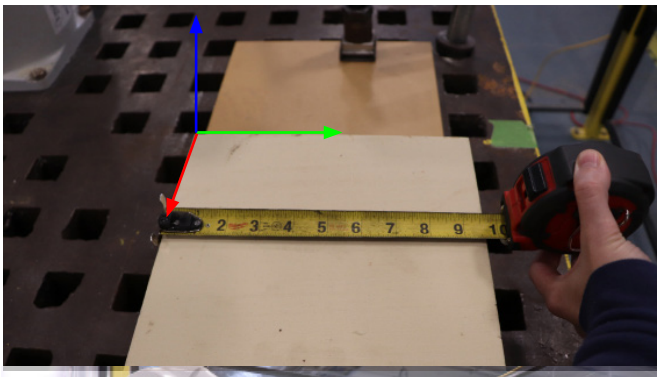


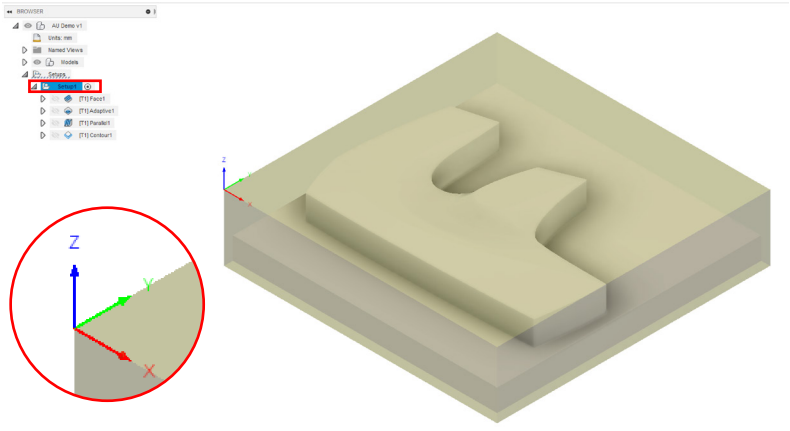
1. **Face**: prepares raw stock for machining, removes all material from top of stock to top of the model
2. **Adaptive**: a roughing strategy that uses an optimized toolpath. Objective of this toolpath is to remove bulk amount of excess material from workpiece in every pass.
3. **Parallel**: widely used finishing strategy. Objective of this toolpath is to improve surface finish, dimensional accuracy and tolerance.
4. **Contour**: To finish off sides of profile.

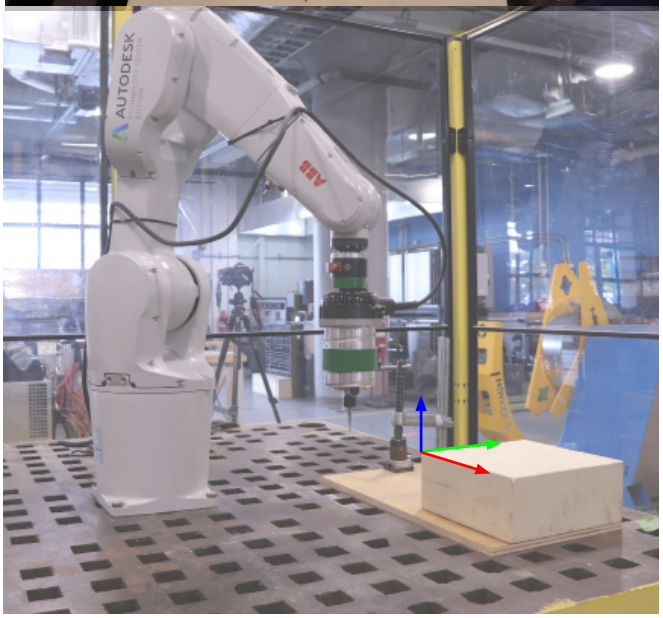
With Fusion 360 - with minimum input and by using a lot of the default settings - a toolpath will be created. This tutorial will not cover toolpath optimization, but will focus on using mostly default settings. There are many other tutorials that cover toolpath optimization. We will not be covering this in our lab.

Measure your stock and update Setup window.

1. Position your stock in a similar orientation to the image on the right. Familiarize yourself with the coordinate system and make sure it matches the coordinate system in Fusion 360.
2. Measure your stock in X, Y and Z. You'll need these measurements for the next page.



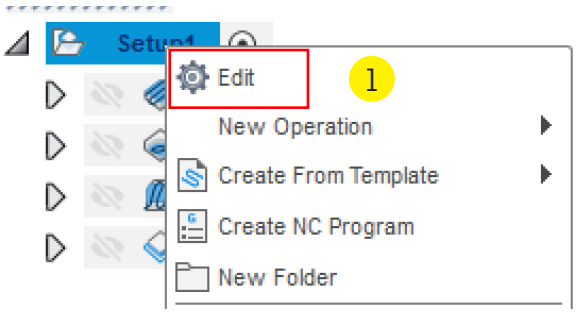


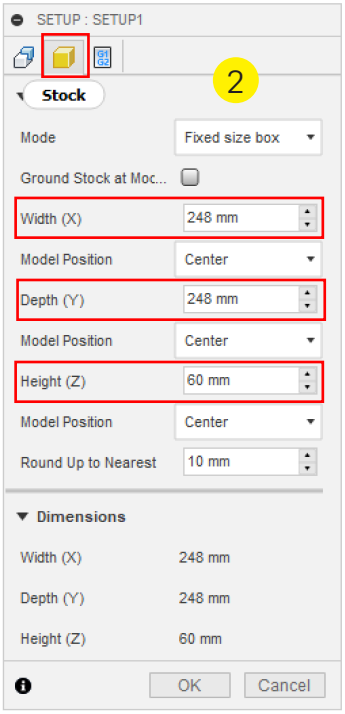


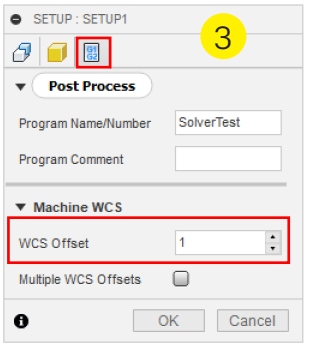
PREPARING TOOLPATHS FOR POST PROCESSING: UPDATE STOCK MEASUREMENT

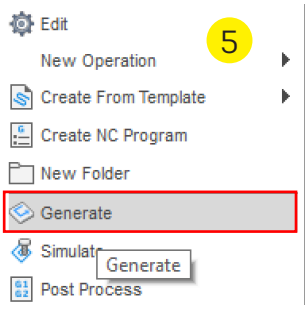
In Fusion 360:

1. Right click **Setup1** → Select **Edit**
2. Select **Stock** tab. Update **Width (X), Depth(Y)** and **Height (Z)** with your stock measurements from the previous page.
3. Select **Post Process** tab. Make sure **WCS Offset** is assigned a number. In this example we are using **1**.
   - This is your Workobject definition
   - This example will create a Workobject **wADSK1** in the code
   - **This number cannot be zero, otherwise you will get an error when you post-process.** *Error: Workobject data has not been specified. Define it as WCS value, editing current Setup.*
4. Toolpath edits will cause the toolpaths to be out of date.
5. Right click **Setup1** → Select **Generate** to update toolpaths with edited settings.





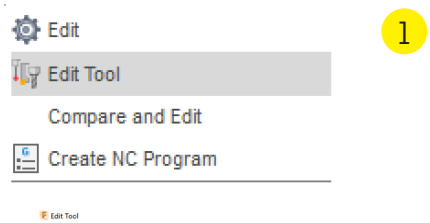


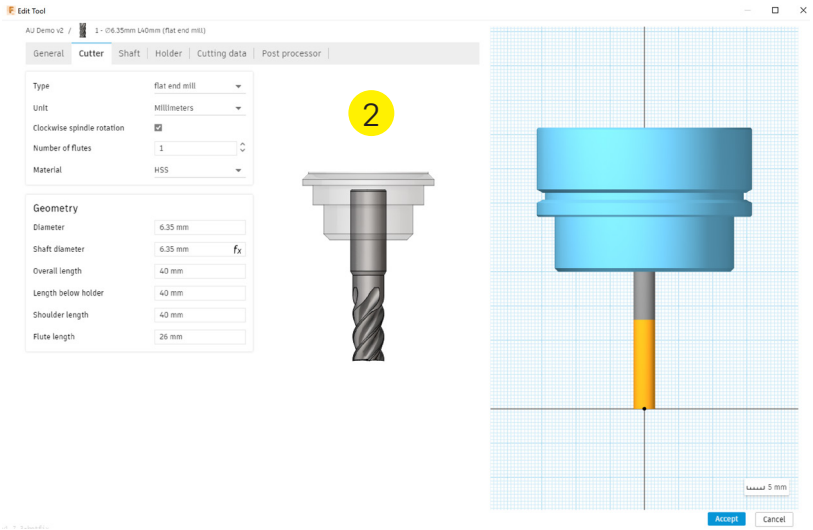


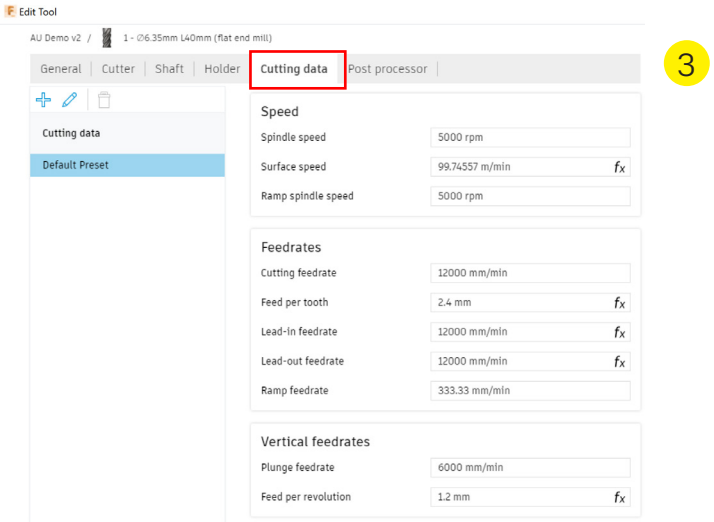
PREPARING TOOLPATHS FOR POST PROCESSING: UPDATE TOOL

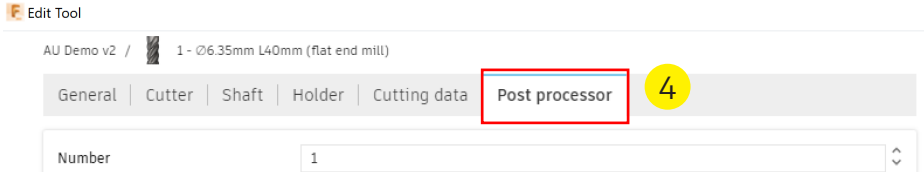
Update your tool

1. Right click **Face1**→ Select **Edit Tool**
2. Update your tool measurements here.
3. Go to **Cutting data** tab → update your feeds and speeds
   - In Fusion → Feed & Speed is in mm/min
   - In RAPID Code (posted robot code)→ Speed is in mm/sec
   - Check your feeds & speeds and adjust accordingly.
   - In this example setting **Cutting feedrate** to **12000** mm/min will create speed data that is **v200** (200 mm/sec)
4. Go to **Post processor** tab → assign tool number. In this example it is **1**.
   - Your tool number will be represented in the tool data. For example tool number 1 will cause the tool data to be named **tADSK1**





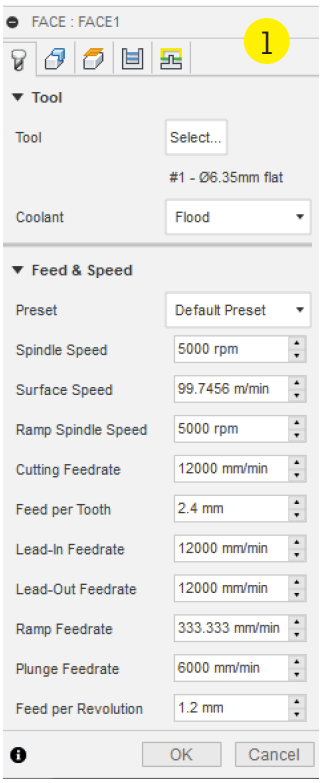


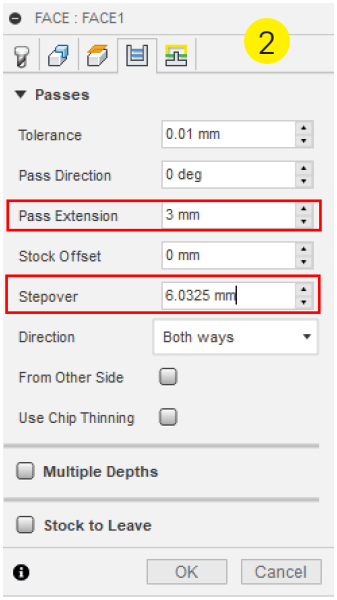


PREPARING TOOLPATHS FOR POST PROCESSING: ADJUST TOOLPATH SETTINGS

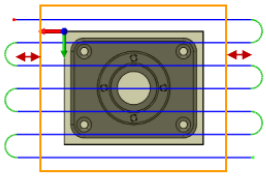
Adjust toolpath settings: Face

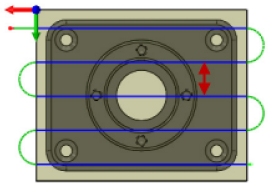
1. Right click **Face1**→ Select **Edit**
   - Check **Tool** and **Feeds & Speeds**
   - Can change tool with **Select...**
2. Select the **Passes** tab.
3. Check the **Pass Extension** - distance to extend the passes beyond the machining boundary
   - *in this example it is set to 3 mm*
4. Check the **Stepover** - horizontal stepover between passes
   - *default value is used in this example.*
5. Select **OK**





Pass Extension

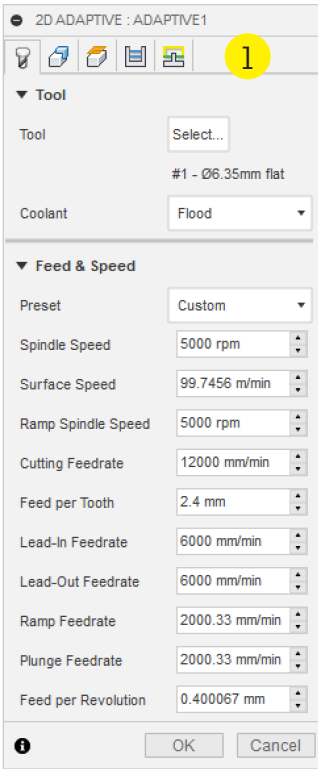


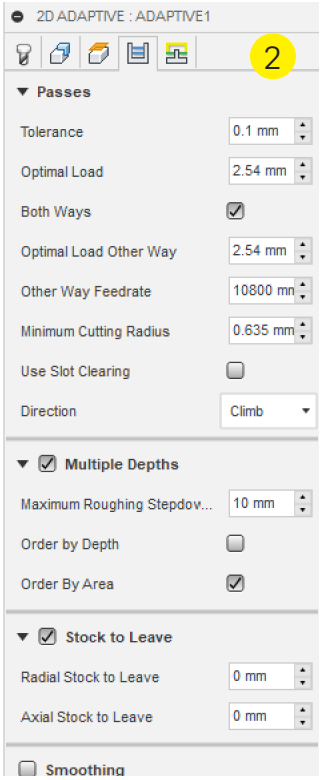


Stepover

Adjust toolpath settings: Adaptive

1. Right click **Adaptive1**→ Select **Edit**
   - Check **Tool** and **Feeds & Speeds**
   - Can change tool with **Select...**
2. Select the **Passes** tab.
3. To optimize the toolpath for faster milling:
   - Select **Both Ways** - this operation will use both climb and conventional milling (increasing the speed of the toolpath)
   - Select **Order by Area**
4. Select **OK**



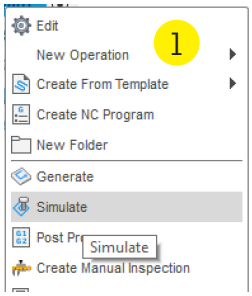


Adjust other toolpath settings

1. Right click **Parallel1** → Select **Edit**
   - Check **Tool** and **Feeds & Speeds**
   - Can change tool with **Select...**
   - Select **OK**
2. Repeat with **Contour1**

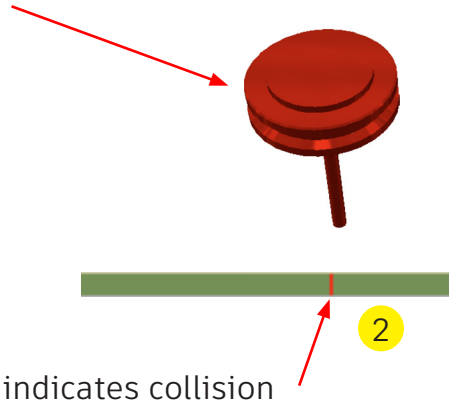
PREPARING TOOLPATHS FOR POST PROCESSING: SIMULATE

Simulate toolpath

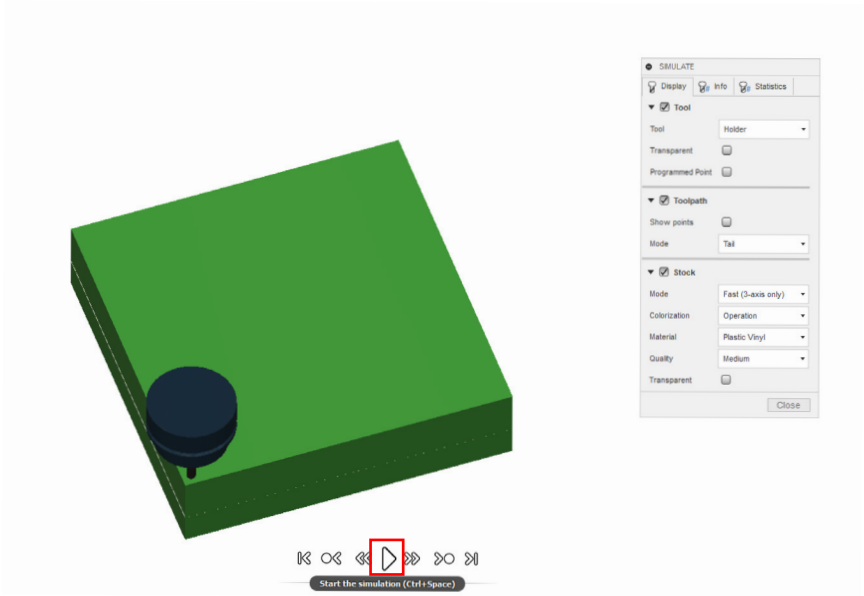


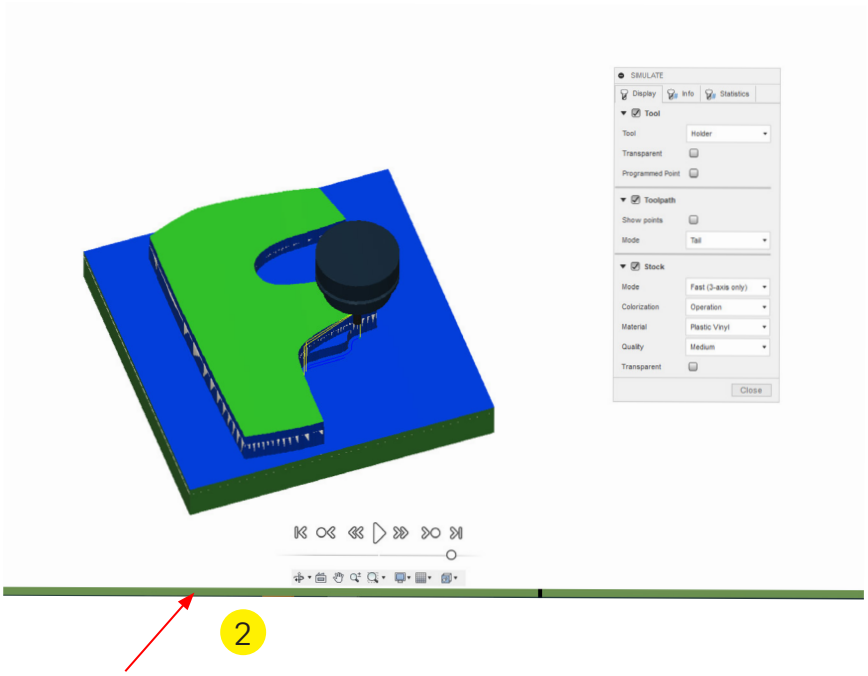
1. Right click **Setup1**→ Select **Simulate**
   - Press **Play**
   - Look for the tool or the tool holder to turn red. This will indicate a collision.
2. Any red in the timeline at the bottom of the screen will also indicate a collision.
   - *the timeline here is all green*

Tool indicates collision

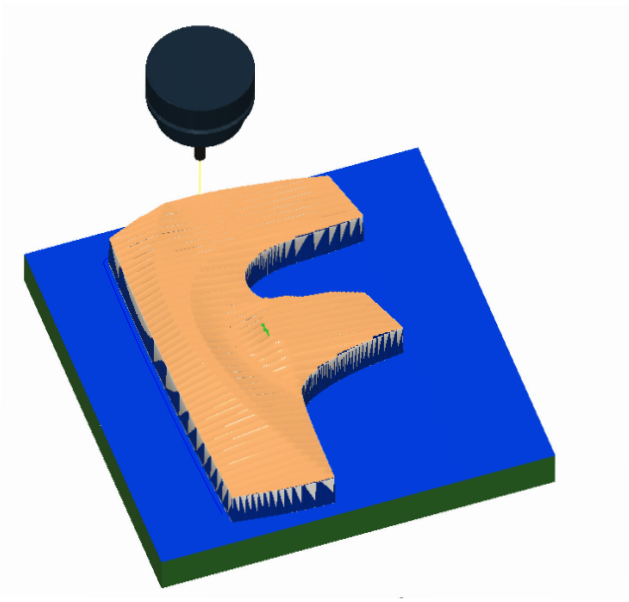


Timeline indicates collision

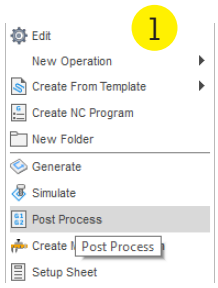




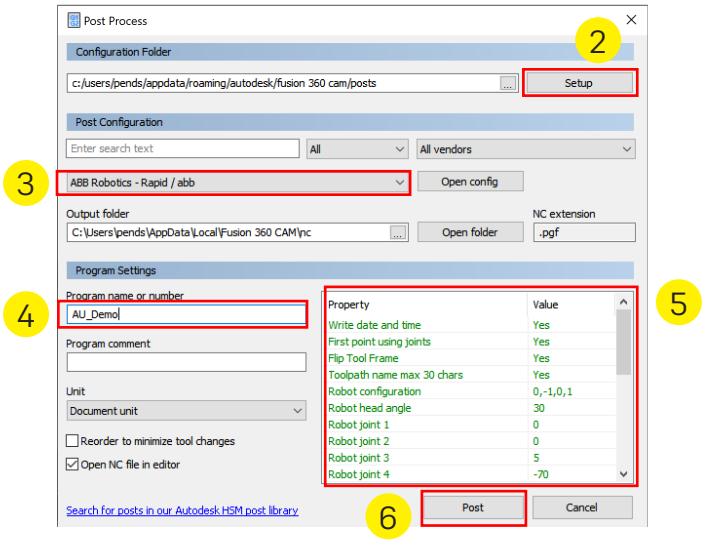
The timeline shows all green, therefore no collisions.

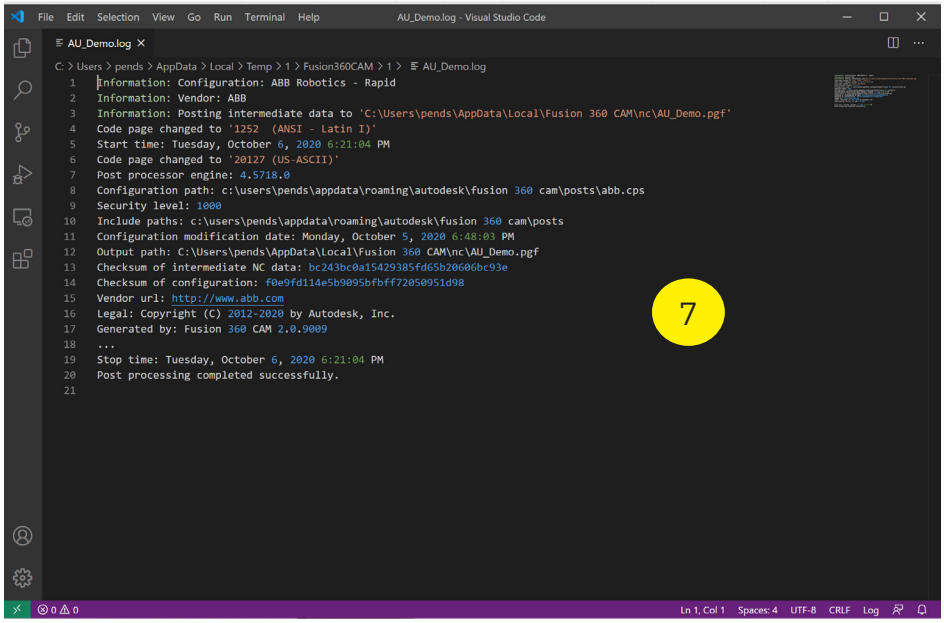


POST PROCESS IN FUSION 360



1. Right click **Setup1**→ Select **Post Process**
2. Post Process Window will open.
   - Select **Setup** → **Select Personal Post Library** *(earlier in the handout we covered steps to place post in this folder)*
3. Select **ABB Robotics post** here
4. Give your program a name here.
   - In this example **AU_Demo**
5. Leave these values with their default values for now. We will explain these settings on the next page.
6. Select **Post**
7. *Post processing completed successfully* will be displayed.



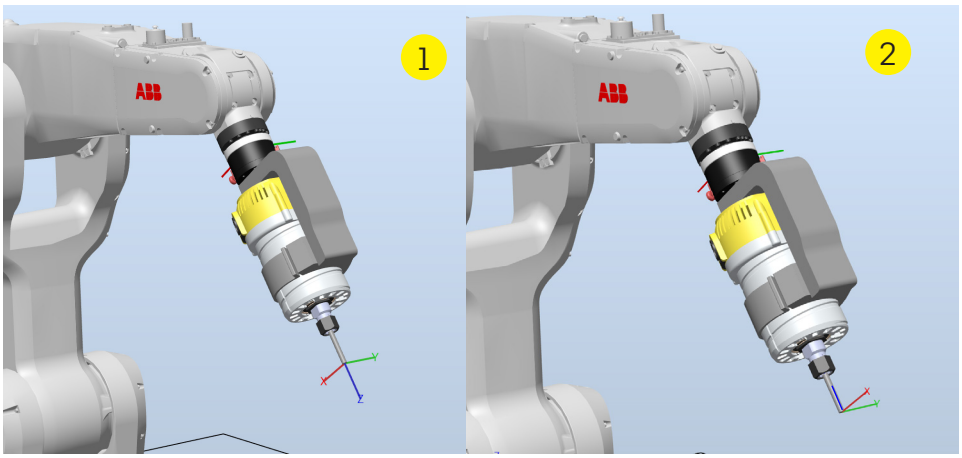


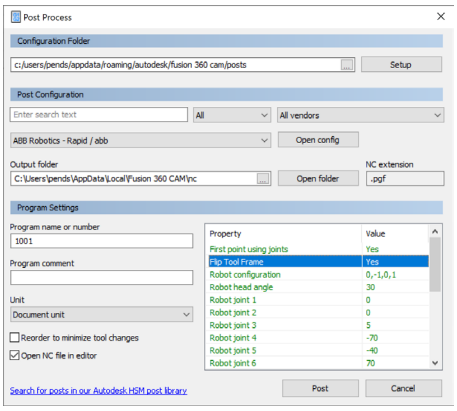
POST-PROCESSOR SETTINGS

TOOL ORIENTATION

1. You can choose if tool orientation has the same Z+ direction as Fusion CAM **(Flip Tool Frame → No)**
2. Or as ABB robot default **(Flip Tool Frame → Yes).** *This is the recommended tool orientation. See #1*

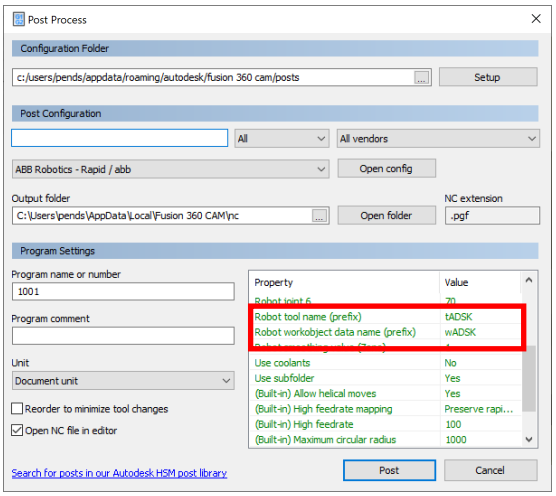
We can manage this condition using *Flip Tool Frame* **option:**





TOOL AND WORKOBJECT PREFIX

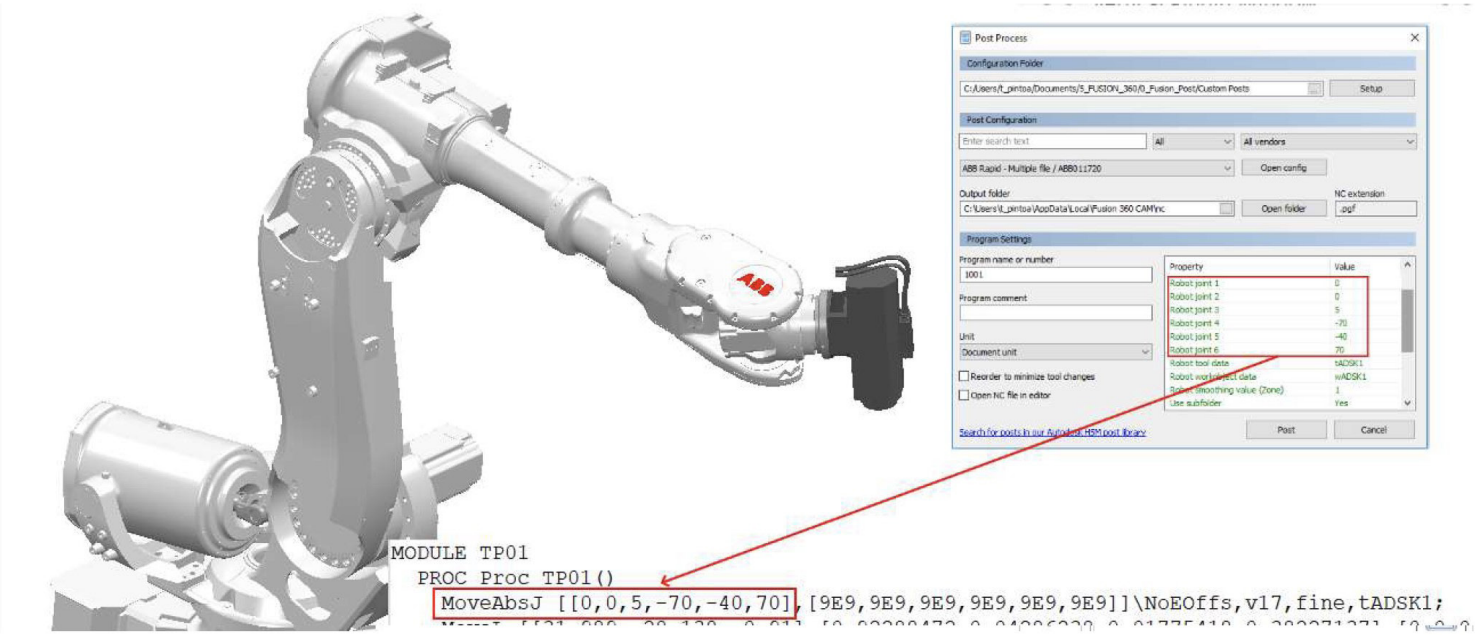
+ Both tool name and WorkObject are written with a prefix set by two specific parameters.
+ Default is tADSK# and wADSK#
+ You can give your Tool and WorkObject any name, but for purposes of this training we will stick to the ADSK convention. We recommend using the naming convention of starting your tool with t*Name* and workobject with w*Name* to keep organized.



POST-PROCESSOR SETTINGS

JOINT VALUE USED FOR INITIAL POSITION

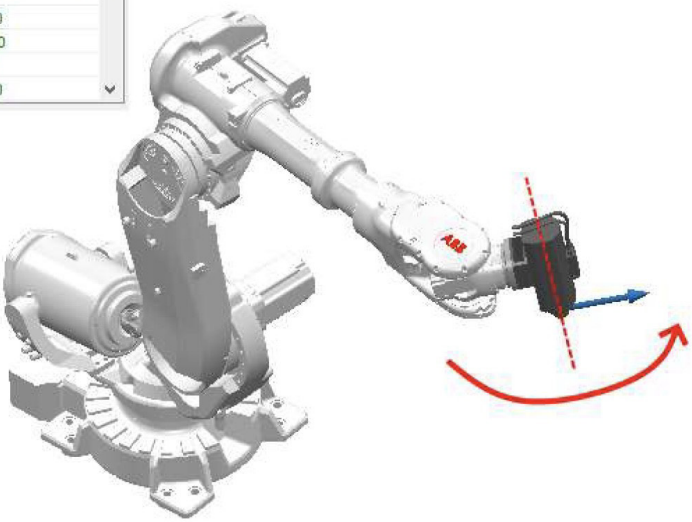
+ There is a parameter for each axis position in degrees.
+ This means that first point is a joint move using joint position and it is not a cartesian position.
+ They will be used to avoid the part and keep the robot posture free from singularities



ROBOT HEAD ANGLE

+ Robot head angle parameter describes the rotation value around tool axis.
+ The angle will be kept throughout the toolpath to avoid singularity. To learn more about singularities, see page 22 in the Appendix of this manual.

Property	Value
Robot path smoothing (APO.CDIS) (mm)	2
Robot 'Status' configuration	'B110'
TOOL number ($TOOL)	1
Robot 'Turn' configuration	'B110010'
Robot head angle	45
Robot joint 1	5
Robot joint 2	-80
Robot joint 3	100
Robot joint 4	75
Robot joint 5	-60



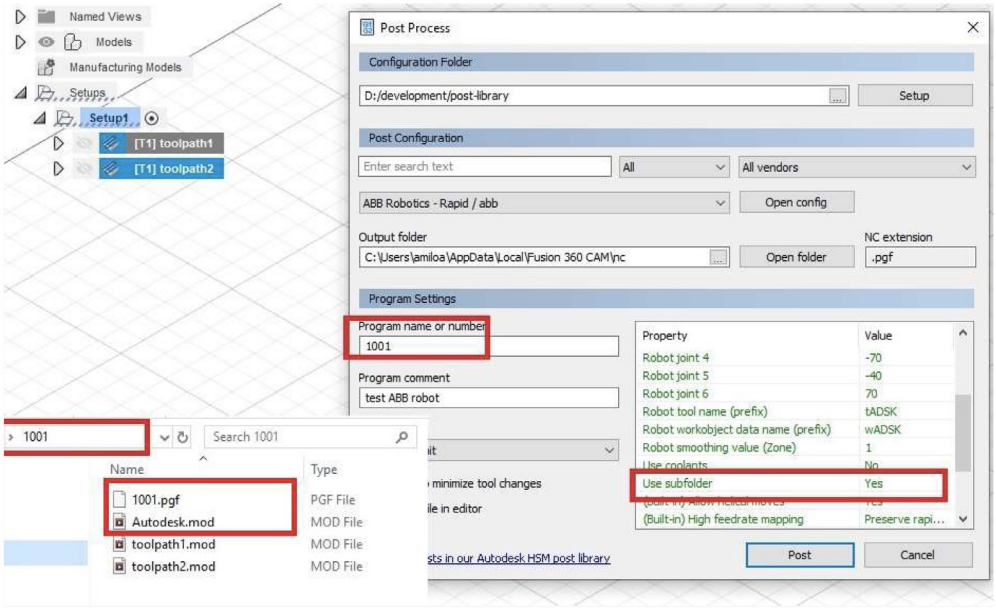
POST-PROCESSOR SETTINGS

PLEASE REMEMBER:

1. The units must be **mm** for ABB industrial robots
2. Change the name of the toolpaths so they **do not** begin with a number: example **2D Contour**. *Otherwise, a syntax error may occur.*

OUTPUT FILES FORMAT

+ ABB postprocessor creates Autodesk.mod as main program, *toolpath_name.mod* as subprograms and *program_number.pgf* as program file
+ You can choose if output files are written in defined output folder or in an additional subfolder (named as program number).



Example Output files format - from **AU_Demo**

Local › Fusion 360 CAM › nc › AU_Demo

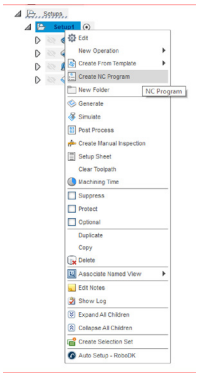
Search AU_Demo

Name	Date modified	Type	Size
AU_Demo.pgf	10/6/2020 2:21 PM	PGF File	1 KB
mAdaptive1.mod	10/6/2020 2:21 PM	MOD File	3,716 KB
mAutodesk.mod	10/6/2020 2:21 PM	MOD File	1 KB
mContour1.mod	10/6/2020 2:21 PM	MOD File	164 KB
mFace1.mod	10/6/2020 2:21 PM	MOD File	41 KB
mParallel1.mod	10/6/2020 2:21 PM	MOD File	325 KB

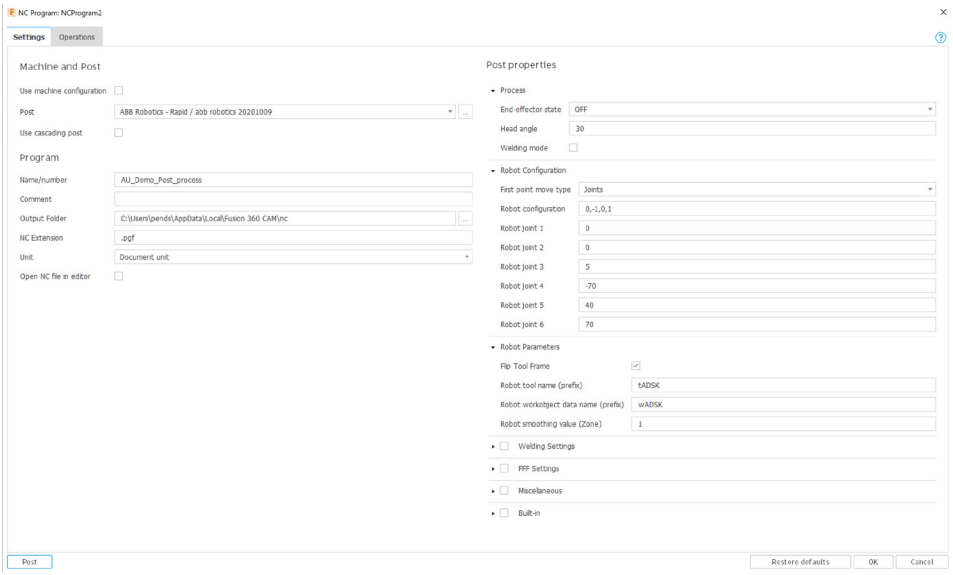
NEW NC POST PROCESS DIALOG WINDOW

New NC Post Process Dialog Window is similar to the other post process window

1. Right click Setup1
2. Create NC program



- The new interface offers options for Welding and FFF (Fused Filament Fabrication)
- Welding creates additional program data: **Seamdata**, **Weavedata**, and **welddata**
  - Welding data is required for welding applications
- FFF settings
  - Signals for communicating with extruder, example extruder on/ extruder off can be utilized here.
- Can support signals for communicating to end effector → on ξ off



Check: *https://cam.autodesk.com/hsmposts* for updates and more supporting documentation on using these settings.

ABB Robotics - Rapid
Download / Share / RSS / Guide
ABB
Purpose: Milling
Version: 42890
Changed: 46 days ago
Extension: pgf
Downloads: 365

VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: ROBOT SIMULATOR

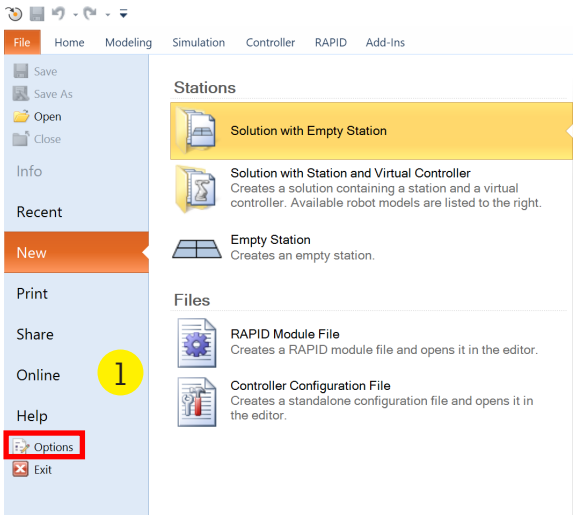
ABB RobotStudio is an engineering tool for configuring and programming ABB robots, both physical robots on the shop floor and virtual robots in a PC. Use this application for modeling, offline programming, and simulation of robot cells. Its advanced modeling and simulation features help in visualizing multi robot control, safety features, 3D vision, and robot supervision. RobotStudio's built-in programming environment allows online and offline programming of robot controllers. In online mode, it is connected to a real controller and in offline mode, it is connected to a virtual controller that emulates a real controller in a PC.

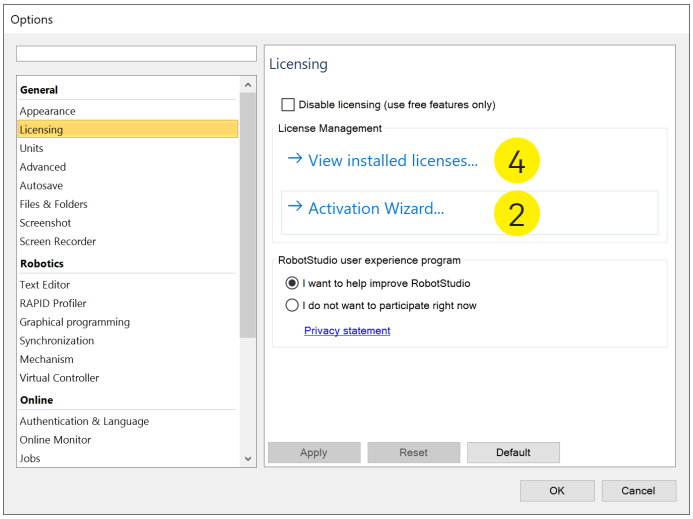
RobotStudio is built on the ABB Virtual Controller, an exact copy of the real software that runs your robots in production. This allows very realistic simulations to be performed, using real robot programs and configuration files identical to those used on the shop floor.

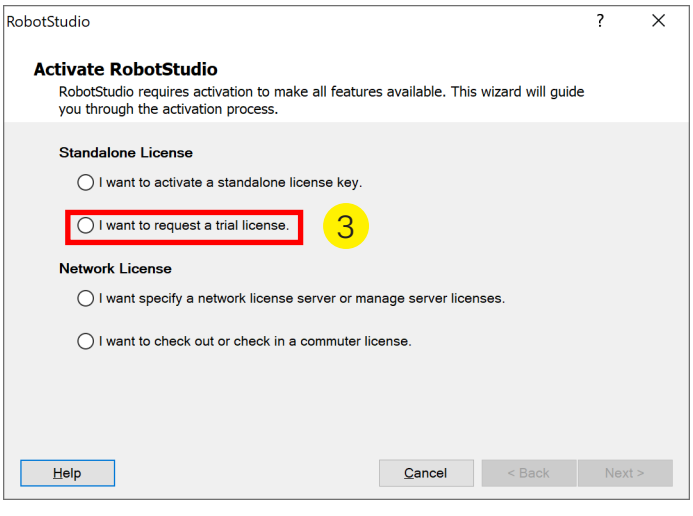
Download here: https://new.abb.com/products/robotics/robotstudio/downloads

+ Premium features must be activated to use the features in this manual.

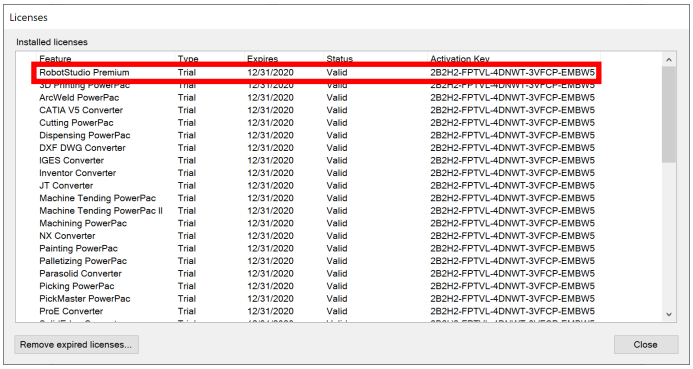
1. **File → Options**
2. **Activation Wizard...**
3. **I want to request a trial license**, follow steps to request trial license.
4. **View Installed Licenses**
5. **RobotStudio Premium**







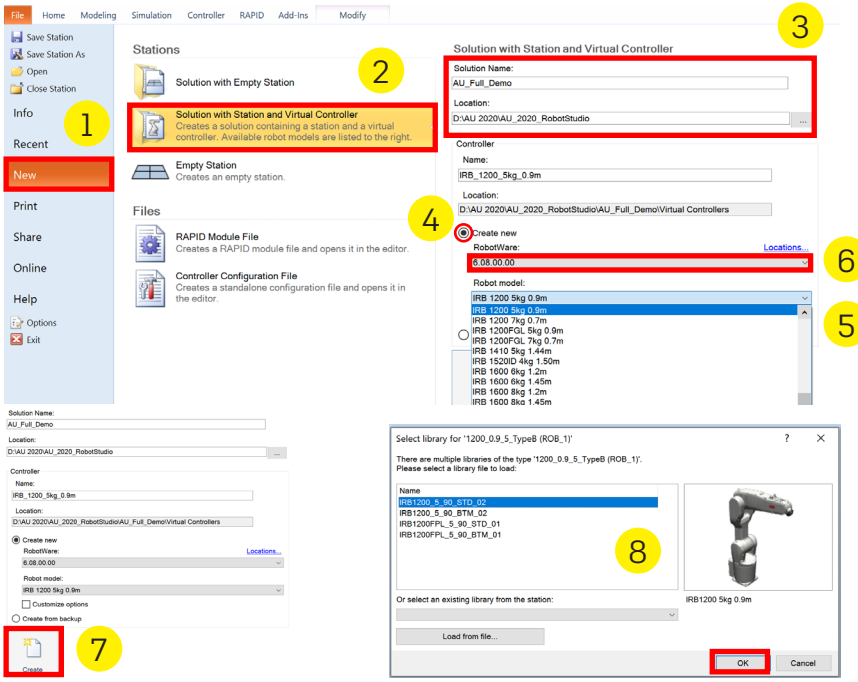
5



VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: ROBOT SIMULATOR

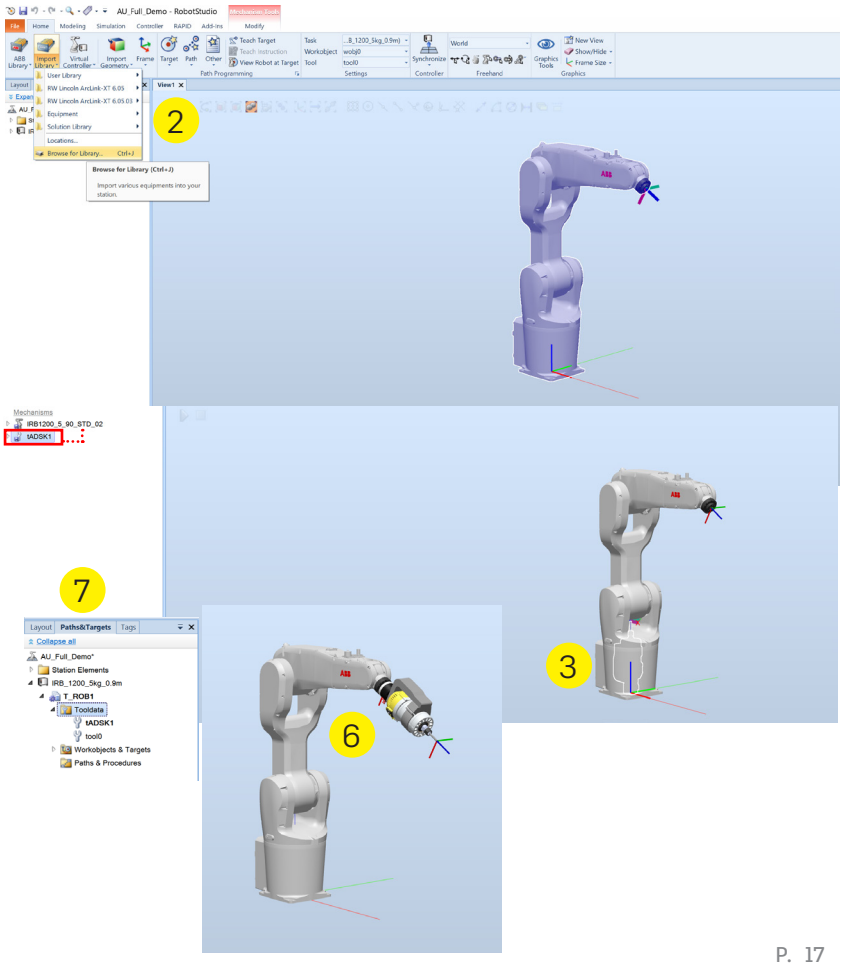
Create a new File in RobotStudio

1. **File → New**
2. **Solution with Station and Virtual Controller**
3. Name your Solution and identify the file location. ex. *AU Full Demo*
4. Select **Create new**
5. Select **IRB 1200 5kg 0.9m**
6. For this demo use **RobotWare 6.08**
7. Select **Create**
8. When **Select Library** window pops up:
   - Select **IRB 1200_5_90_STD_02**
   - Select **OK**



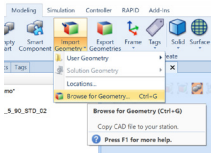
Setting up Robot Cell

1. A robot should appear similar to the image in the left.
2. To Import Tool:
   - **Home** tab → **Import Library**
   - Select **Browse for Library**
   - Navigate to **Data Set Folder** and Select **tADSK1.rslib**
   - Select **Open**
3. A silhouette of the tool will appear at the base of the robot
4. Select the **Layout** tab
5. Select **tADSK1** and drag it onto the robot name (in this example *IRB 1200_5_90_STD_02*)
6. Select **Yes**: *Do you want to update the position of 'tADSK1'?*
   - The tool will attach to joint 6 of the robot arm
7. Select **Path & Targets**
   - Expand **Tooldata**
   - **tADSK1** will be listed

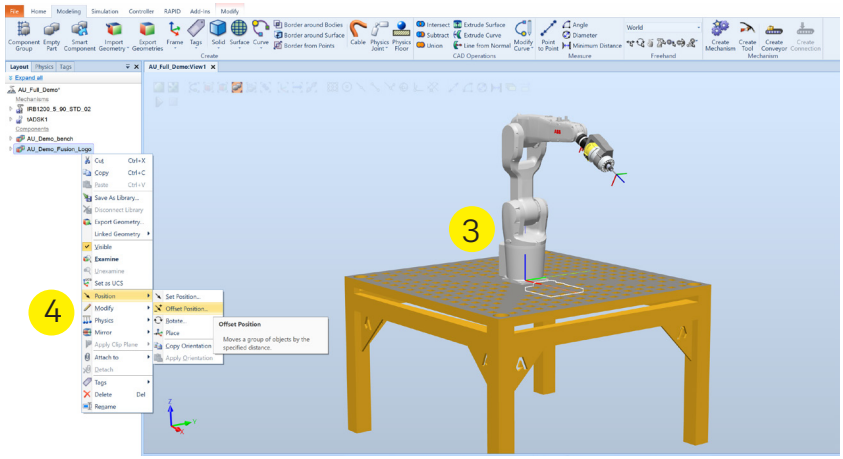


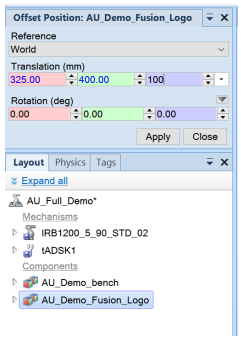
VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: ROBOT SIMULATOR

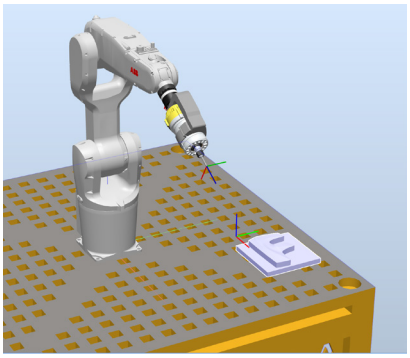
Add Bench and reference model:



1. **Modeling** tab → **Import Geometry**
   - **Browse for Geometry**
   - Navigate to Data Set Folder
   - Select *AU_Demo_bench.step*
   - Select **Open**
2. **Modeling** tab → **Import Geometry**
   - **Browse for Geometry**
   - Navigate to Data Set Folder
   - Select *AU_Demo_Fusion_Logo.iges*
   - Select **Open**
3. A silhouette of the model will appear at the base of the robot
4. Right-click *AU_Demo_Fusion_Logo*
   - Select **Position** → **Offset Position**
   - Type values in for X: **325,** Y: **400** and Z: **100**
   - Select **Apply**
5. The Fusion logo model will shift from base of robot.

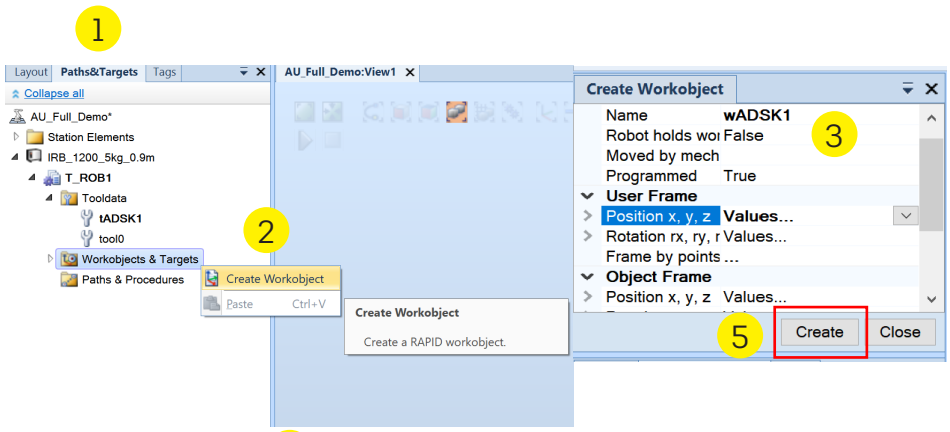


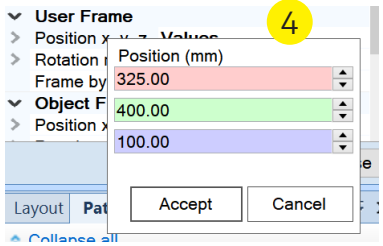


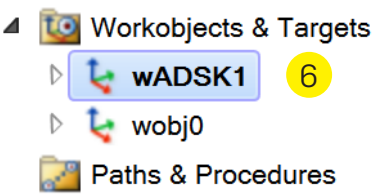


Add Work Object

1. **Home** tab → **Paths & Targets**
2. Right click **Workobject & Targets** → **Create Workobject**
3. **Create Workobject** → Change name to *wADSK1*
4. Under **User Frame** → Select small arrow next to Position x,y,z→ Change position values:
   - Type values in for X: **325,** Y: **400** and Z: **100**
   - Select **Accept**
5. Select **Create**
6. **wADSK1** will now be listed under **Workobjects & Targets**

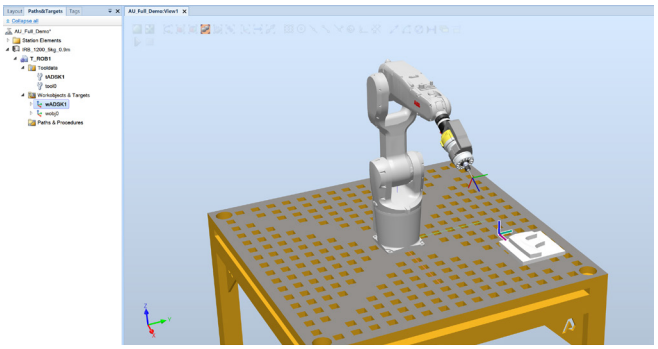






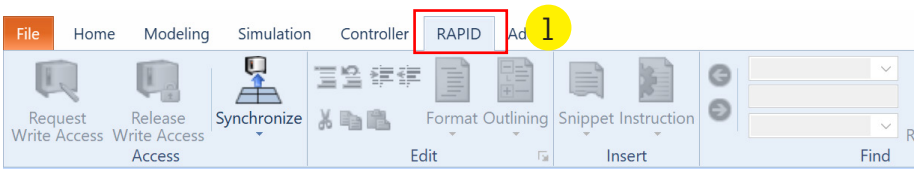
VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: LOADING CODE

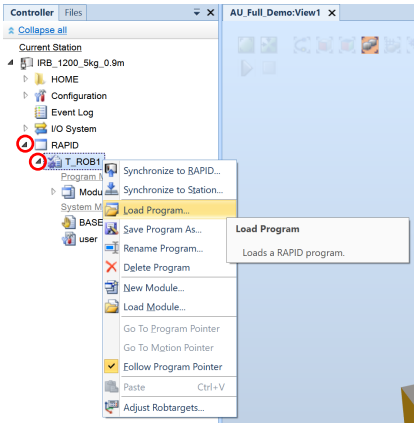
The file should look like this→

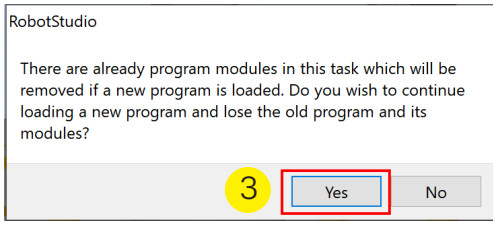


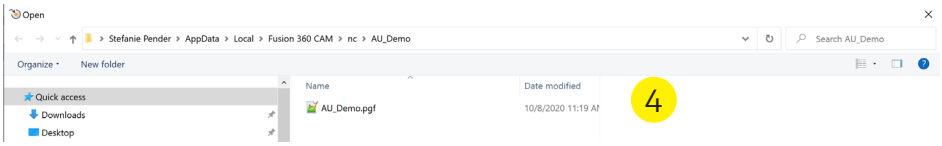
Opening RAPID Code from Fusion :

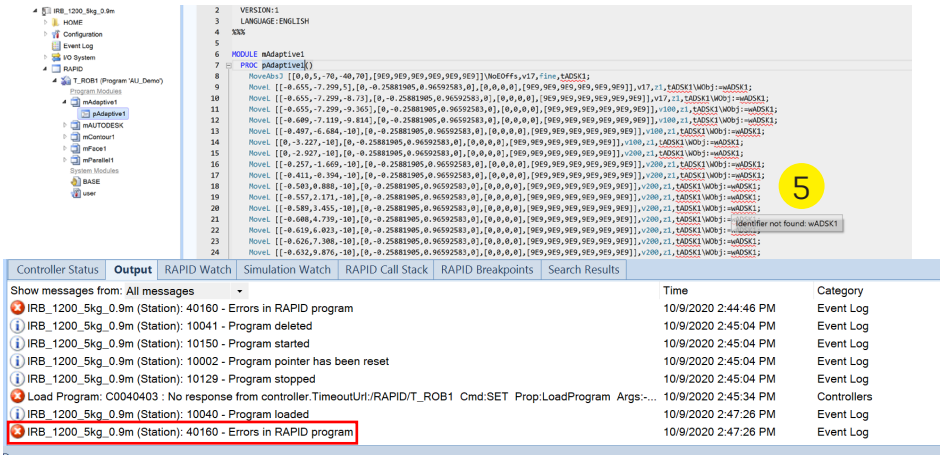
1. Select **Rapid tab** in the toolbar.
2. Under **Controller** Tab
   - Expand **RAPID**
   - Expand **T_ROB1**
   - Right click **T_ROB1** → **LOAD PROGRAM…**
3. A warning will pop up: select **Yes**
4. Select the posted .pgf file located in your nc folder.
   - Example location: *C:\Users\pends\AppData\Local\Fusion 360 CAM\nc*
   - Example: Select *AU_demo.pgf*
   - RobotStudio may take a while to load, be patient
5. In the **RAPID** tab: Double click the toolpath mAdaptive1. The syntax errors are underlined in red: tADSK1 and wADSK1
   - There are syntax errors because Fusion **does not** create tool data or work object data in the code.
   - The output window will also show *Errors in RAPID Program*
6. Your tool data and work object are defined in the Station, however that data has not been pushed to RAPID Code yet. You must **Synchronize to RAPID** from Menu Ribbon. That will eliminate the red underlined errors.

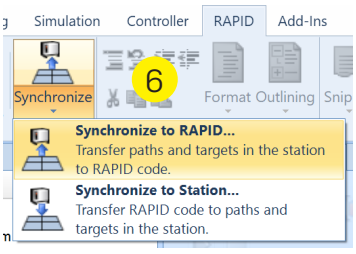








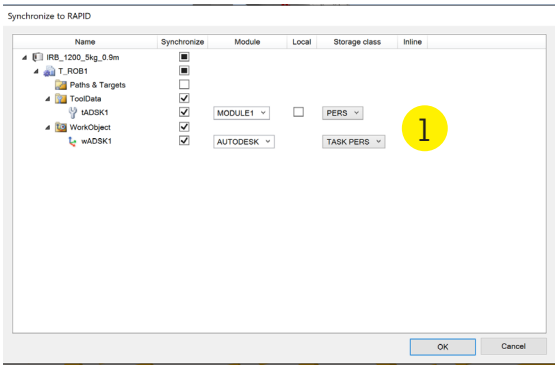


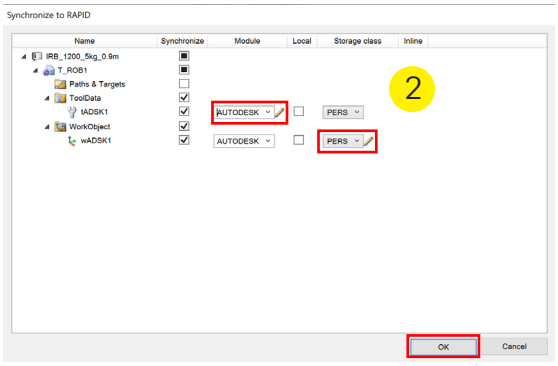


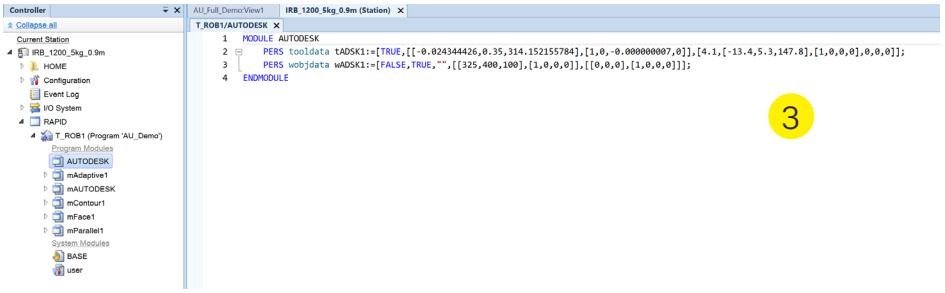
VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: LOADING CODE

Synchronize to RAPID :

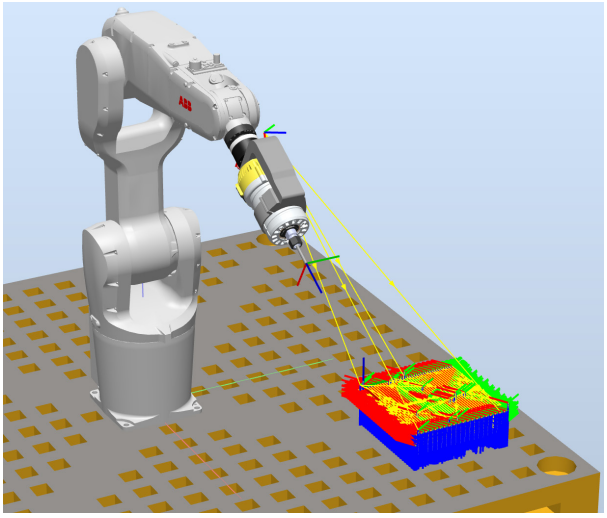
1. Select **Tooldata and WorkObject** data
2. Make the selections:
   - **Module** → change to **Autodesk**
   - **Storage class** → select **PERS**
   - Select **OK**
3. Return to the **RAPID** tab
   - Double-click **Autodesk**
   - **tooldata** and **wobjdata** will be defined
   - Double-click mAdaptive1 → Syntax errors will be corrected.
4. Select **Check Program**
   - Select this to check the syntactic correctness of the modules
   - In the **Output** window→ **No Errors**
5. Under the **RAPID** tab → **Synchronize** → **Synchronize to Station**
6. Return to the **Home** tab
   - Station will now be updated with the toolpaths

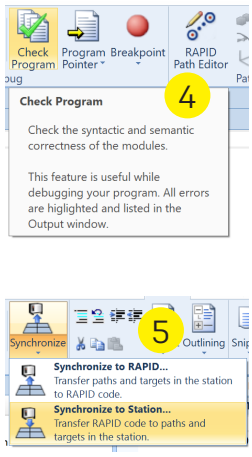


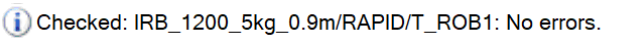


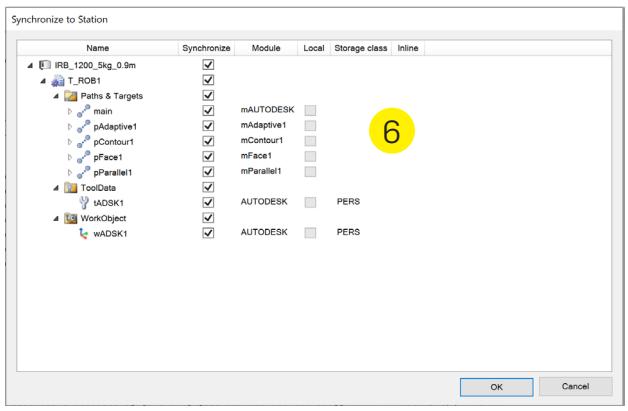



```
6  MODULE mAdaptive1
7  PROC pAdaptive1()
8    MoveAbsJ [[0,0,5,-70,-40,70],[9E9,9E9,9E9,9E9,9E9,9E9]]\NoEOffs,v17,fine,tADSK1;
9    MoveL [[-0.655,-7.299,5],[0,-0.25881905,0.96592583,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v17,z1,tADSK1\WObj:=wADSK1;
10   MoveL [[-0.655,-7.299,-8.73],[0,-0.25881905,0.96592583,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v17,z1,tADSK1\WObj:=wADSK1;
11   MoveL [[-0.655,-7.299,-9.365],[0,-0.25881905,0.96592583,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v100,z1,tADSK1\WObj:=wADSK1;
12   MoveL [[-0.609,-7.119,-9.814],[0,-0.25881905,0.96592583,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v100,z1,tADSK1\WObj:=wADSK1;
13   MoveL [[-0.497,-6.684,-10],[0,-0.25881905,0.96592583,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v100,z1,tADSK1\WObj:=wADSK1;
14   MoveL [[0,-3.227,-10],[0,-0.25881905,0.96592583,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v100,z1,tADSK1\WObj:=wADSK1;
15   MoveL [[0,-2.927,-10],[0,-0.25881905,0.96592583,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]],v200,z1,tADSK1\WObj:=wADSK1;
```





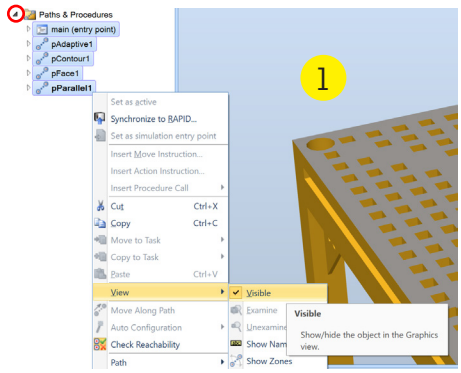


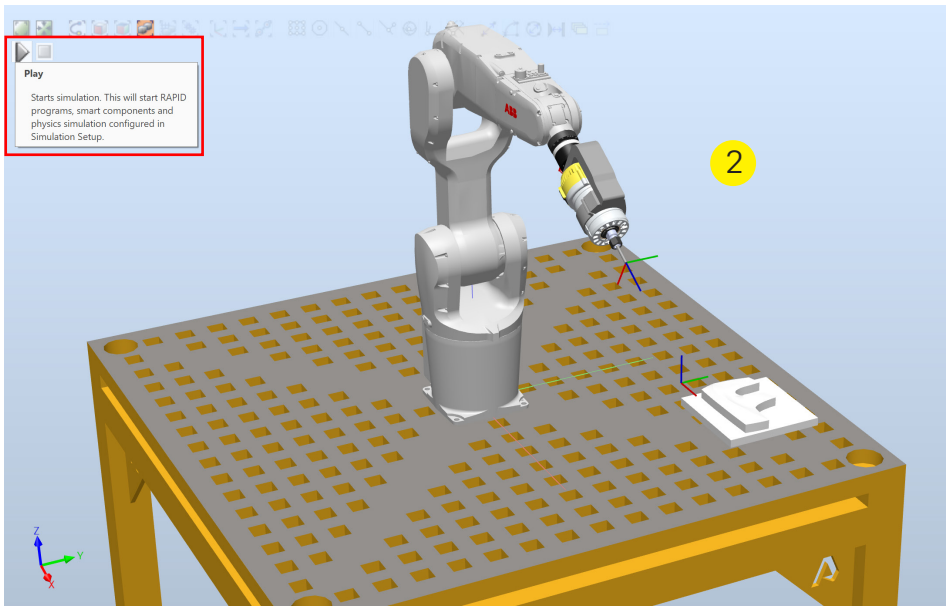


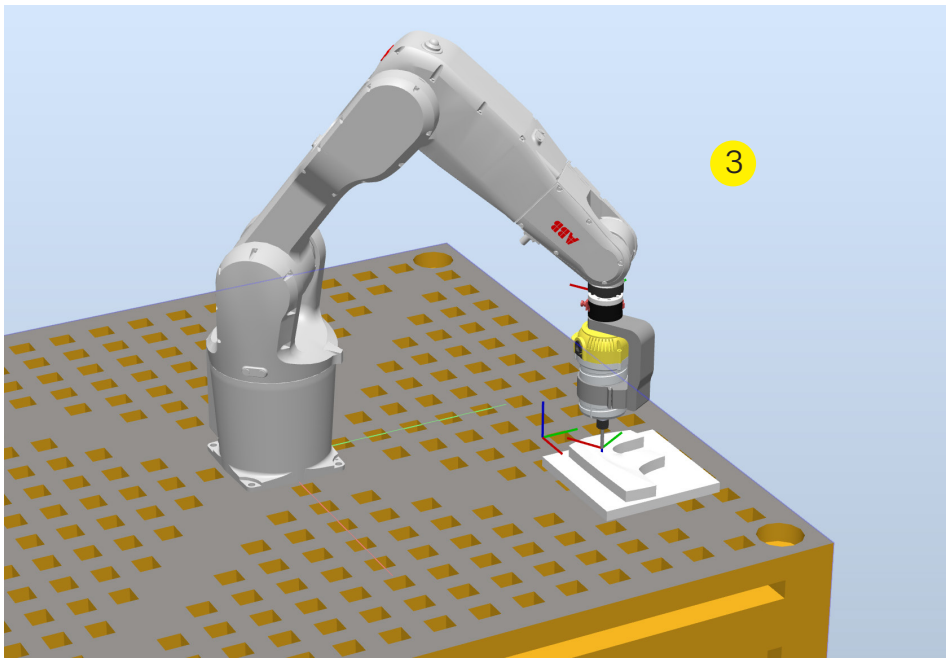
VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: SIMULATION

In the Home Tab:

1. Expand **Paths & Procedures**
   - Select all the toolpaths (Shift + click)
   - Right-click → Select **View** → Select **Visible**
   - Robot targets will be hidden → it will be easier to view toolpaths
2. Select **Play**
   - Simulation will start. This will start RAPID programs.
   - Observe the toolpaths and look for any errors or unexpected movements



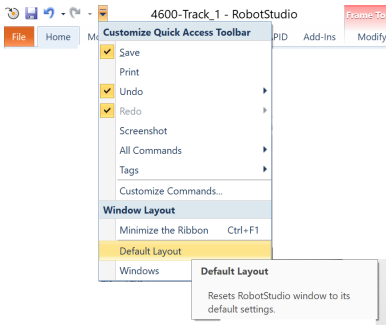




ALWAYS SIMULATE IN ROBOTSTUDIO BEFORE RUNNING PROGRAMS ON ANY ROBOT

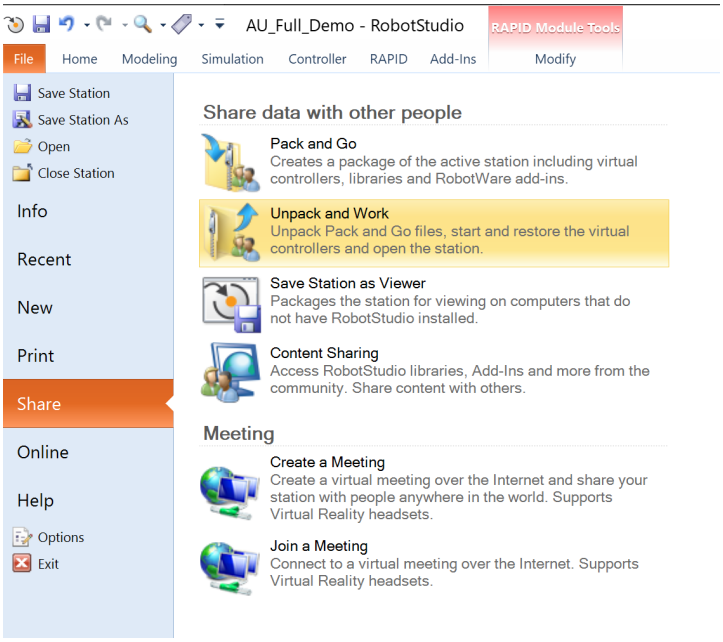
VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: TIPS AND TRICKS

TIPS & TRICKS



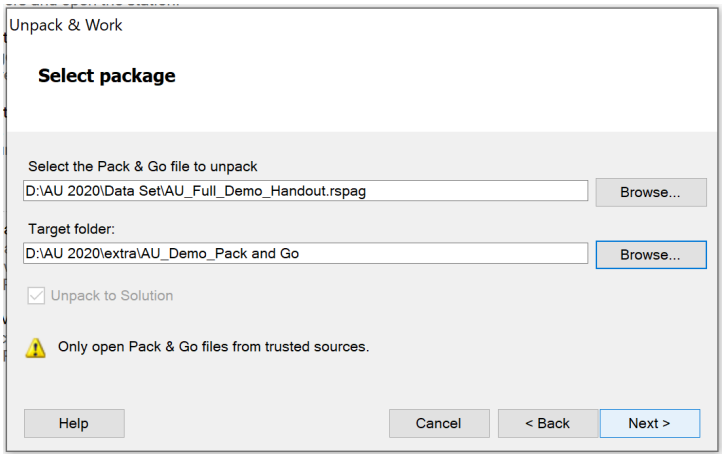
1. To orbit in RobotStudio → Select *Shift + Ctrl + Left Mouse Button*
2. If you are unable to locate a window, return to your default layout.

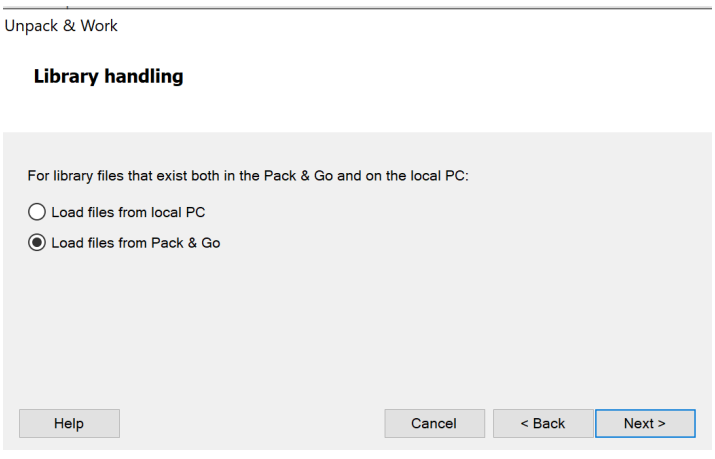
USE PACK AND GO FILES



1. In the Data Set there are **.rspag** files → use these files to troubleshoot the demo
2. Go to **File → Share → Unpack and Work**
3. Navigate to Pack & Go file in Data Set and set Target Folder as an empty folder locally on your PC.
4. Load files from Pack & Go
5. Select Next
6. The virtual station will load

Pack & Go files are the best way to share RobotStudio files for troubleshooting.

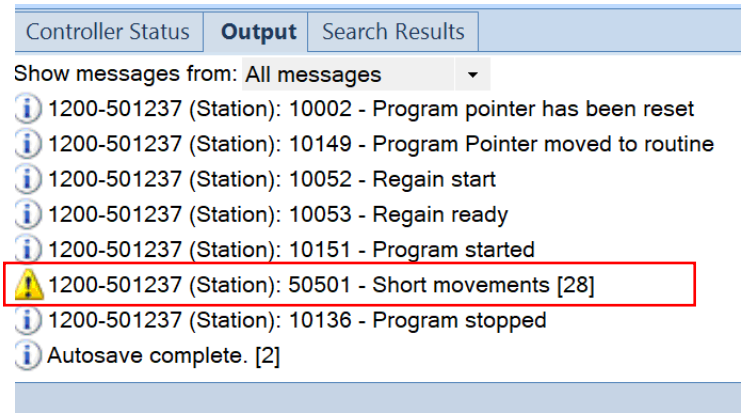


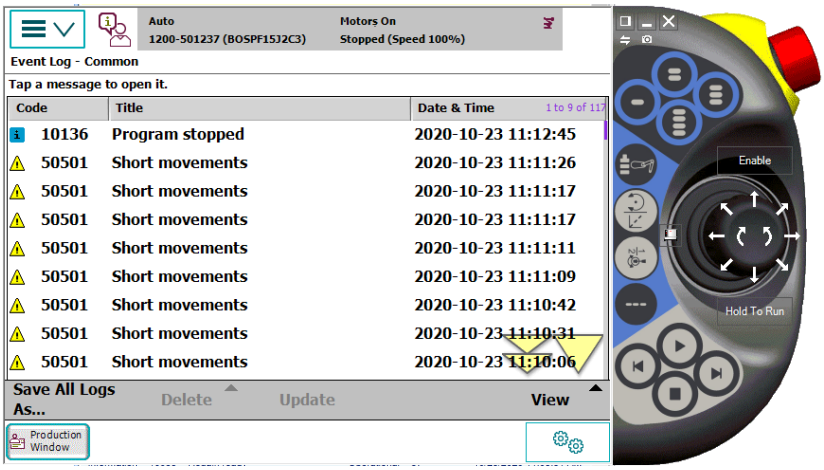


More ABB RobotStudio tutorials can be found here: *https://new.abb.com/products/robotics/tr/robotstudio/tutorials*

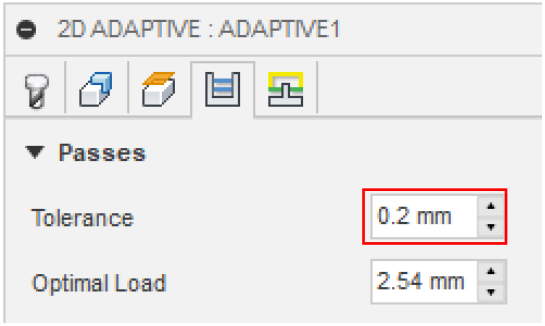
VALIDATING TOOLPATHS IN ABB ROBOTSTUDIO: TIPS AND TRICKS

Adjusting the tolerance of the toolpaths in Fusion 360 should correct errors if the output window or teach pendant event log is indicating the *short movements error* during simulation in RobotStudio





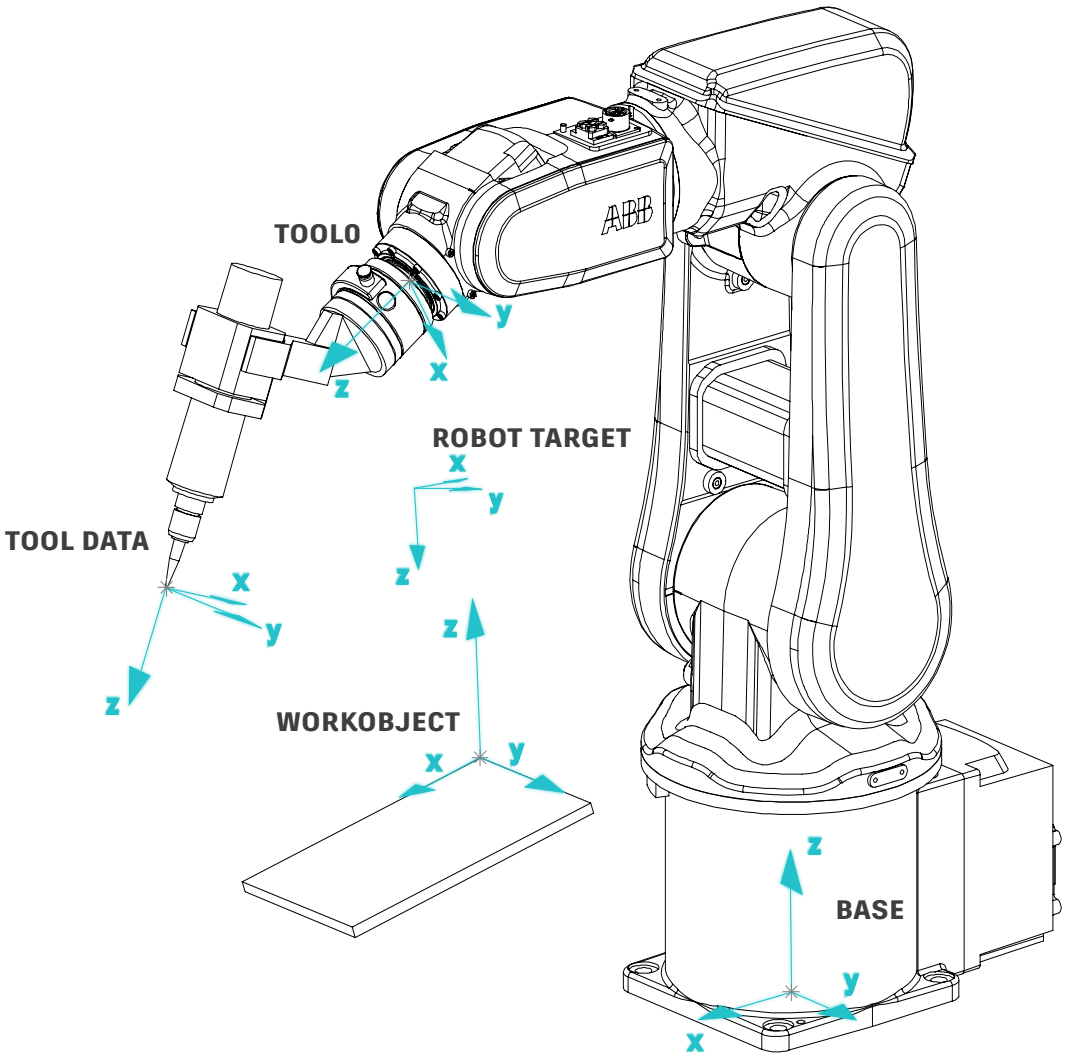
It is tempting to always use very tight tolerances, but there are trade-offs including longer calculation times, large robot code files, and very short line moves. Short line moves, coupled with high feedrates cause the *short movements* error.



Reducing the tolerance (increasing the value in the window) will decrease the number of robot targets in your robot code. This should eliminate the *short movements error.*

CALIBRATING COORDINATE SYSTEMS ON THE ROBOT

Tool Data and WorkObject Data created in RobotStudio will not match your physical robot cell. You will have to use probing techniques to generate this data.



COORDINATE SYSTEMS

+ **Base -** the center of the base of the robot.
+ **Tool** - a user defined coordinate system, usually fixed to End of Arm tooling.
  - The origin of this coordinate system is the **Tool Center Point** or **TCP.**
  - **`tool0`** is the default and is the center of the mounting flange on link 6.
+ **Work Object** - a user defined coordinate system fixed to the work piece.
  - **wobj0** is the default and is coincident with the Base Coordinate system.
+ **robtarget** - target **pose** (position and orientation) used with move instructions.

Jogging in **Linear** and **Reorient** Motion Modes are movements in Cartesian space.

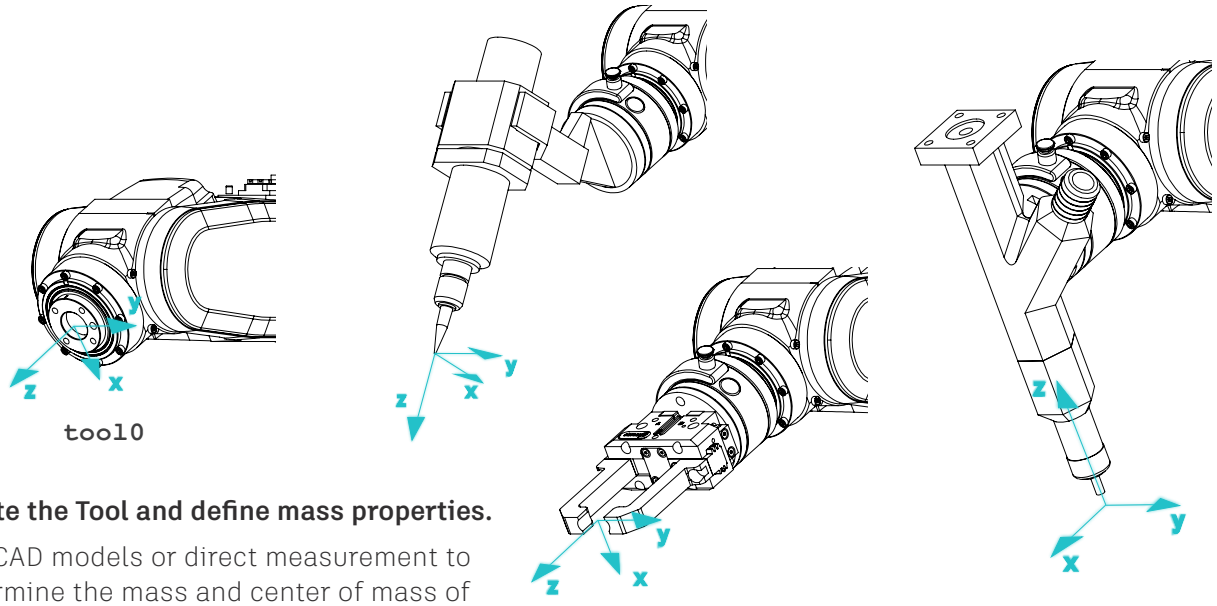
CALIBRATING COORDINATE SYSTEMS ON THE ROBOT: TOOL DATA

TOOL CALIBRATION

For the controller to calculate the joint torques necessary to move the end-effector at a specified speed, the end-effector must be calibrated. To calibrate a new tool, define its mass properties and coordinate system.

Mass properties are mass, center of mass, axes of moment, and moments of inertia. Typically, axes of moment and moments are not required.

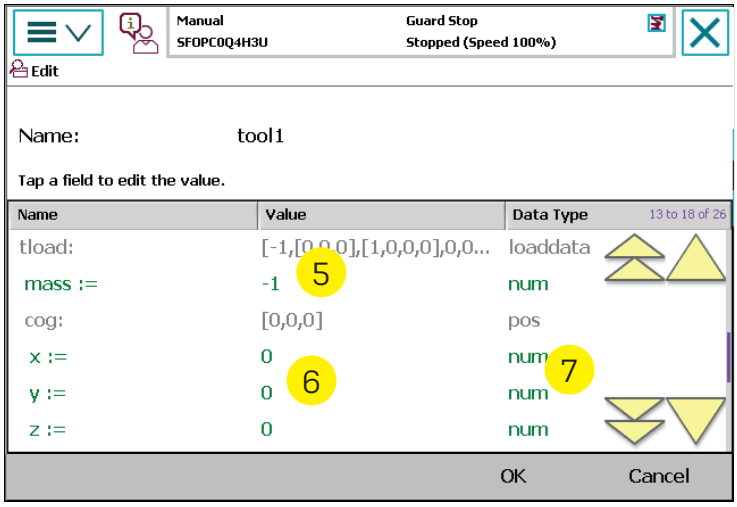
The tool coordinate system is defined by an origin (TCP) and an orientation, relative to Tool0. Choose a meaningful TCP for your end-effector - for example, the tip of an end mill.



Create the Tool and define mass properties.

Use CAD models or direct measurement to determine the mass and center of mass of your EAT.

1. In the Program Data window double-tap **tooldata.**
2. Tap **New...**
3. Enter the name of your tool.
4. Tap **Initial Values.**
5. Scroll down to **mass:=** and input the end effector's mass in Kg.
6. Next to the green **x:=**, **y:=**, and **z:=** values following **mass:=**, specify the center of mass in relation to Tool0.
7. Tap **OK** to create the tool.



PLEASE SEE IRC5 OPERATORS MANUAL FOR MORE INFORMATION ON RUNNING SERVICE ROUTINES.

This process is also automated within the **LoadIdentify service routine** on teach pendant

Go to **Program Editor** → **Debug** → **Call Routine** → **LoadIdentify**

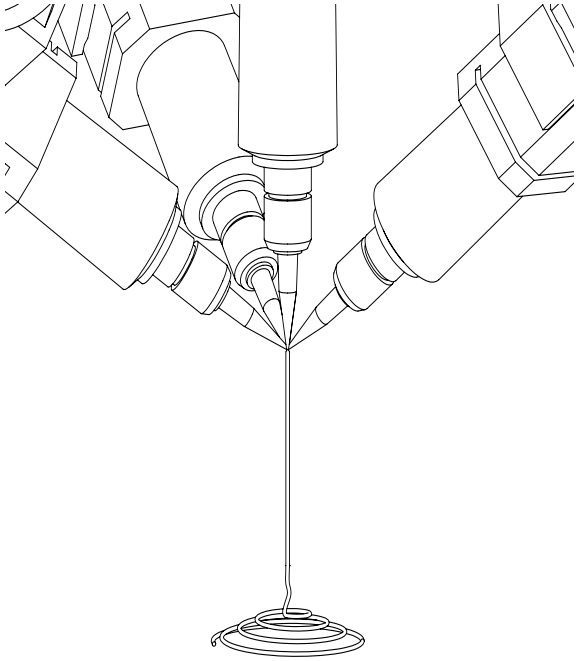
The routine will undergo **Load Identify service routine** and automatically calculate mass and center of gravity (cog).

CALIBRATING COORDINATE SYSTEMS ON THE ROBOT: TOOL CENTER POINT (TCP)

Defining the TCP

1. In the Program Data window double-tap **tooldata.**
2. Tap once on the tool to select it.
3. From the **Edit** menu, choose **Define**.
4. Use the Method **TCP (default orient.)**
5. Jog the robot carefully so the TCP touches a fixed, recognizable location.
6. Select **Point 1** and tap **Modify Position.**
7. Jog the robot to a new orientation, with the TCP touching the same, fixed point.
8. Select **Point 2** and tap **Modify Position**.
9. Repeat steps 6 and 7 for **Point 3** and **Point 4**, using a new orientation each time.
10. Tap **OK**.

The mean error should be <0.5mm and the max error should be <1.0mm.

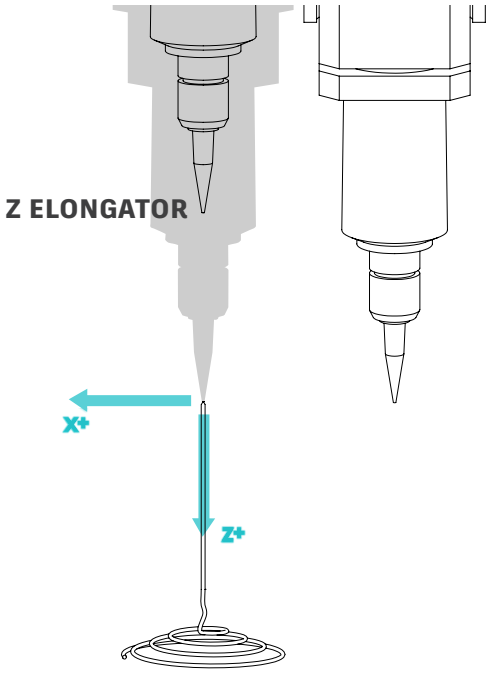


Defining the Tool Orientation

To set an orientation different from the default `tool0`, it is recommended to directly input the orientation quaternion when creating the tool.

Alternatively, follow the above steps for defining the TCP, with the following exceptions:

+ In step 4, use the Method **TCP & Z** or **TCP & Z, X** :
  - ▸ Using TCP & Z will not change the y-axis direction.
  - ▸ Using TCP & Z, X will change all axes.
+ After step 9, for the **Z Elongator** jog the TCP in the desired -Z direction of the tool, with point 4 as the starting point. Do not change the tool orientation.
+ If setting an **X Elongator,** jog the TCP in the desired -X direction, with point 4 as the starting point.

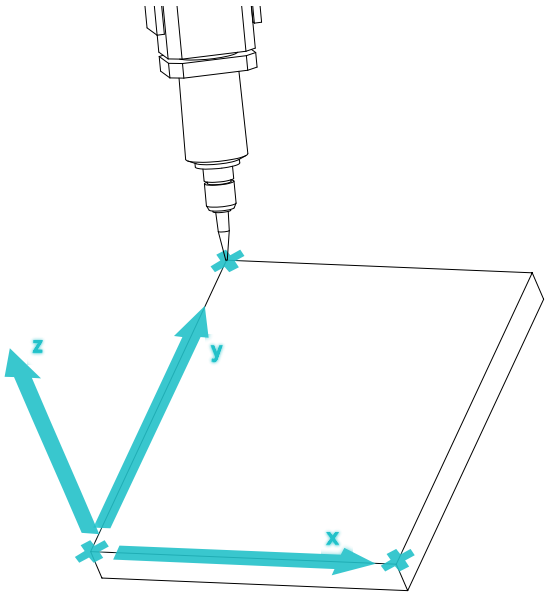


DO NOT MODIFY `tool0`

CALIBRATING COORDINATE SYSTEMS ON THE ROBOT: WORK OBJECT

WORK OBJECT

The work object reference frame is used to affix a reference frame to the object the robot is interacting with. If this object is moved relative to the robot, the program can be adapted quickly by updating the work object's coordinate system.

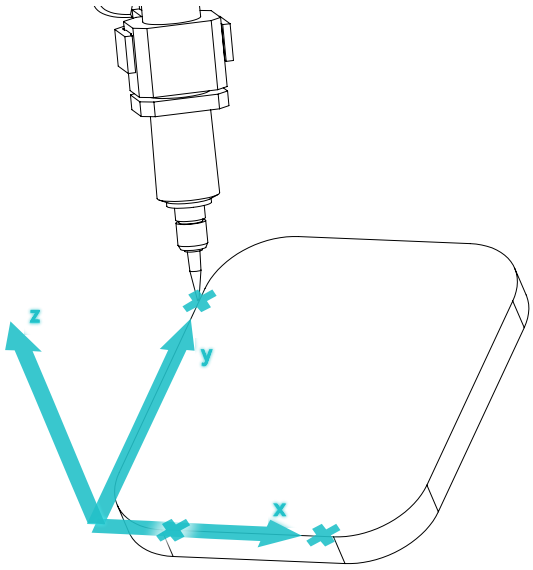


A work object consists of two coordinate systems - an object coordinate system nested under a user coordinate system. In most cases, it is necessary to only define one.

The work object is created by choosing three points on the object to establish the X-Y plane. Pick identifiable features on your work object.

Create a Work Object.

1. In the **Program Data** window, double-tap **workobj**.
2. Tap **New...**.
3. Enter the name of your work object.
4. Change other fields if needed.
5. Tap **OK** to create the work object.
6. Select the new work object.
7. From the **Edit** menu, select **Define.**
8. Under **Object method,** select **3 points**.
9. Jog the robot to a point along the x-axis.
10. Select **Object Point X 1** and tap **Modify position**.
11. Jog the robot to a second point along the x-axis, in the positive direction.
12. Select **Object Point X 2** and tap **Modify position**.
13. Jog the robot to a point along the y-axis.
14. Select **Object Point Y 1** and tap **Modify position**.
15. Tap **OK.**



A large distance between X1 and X2 is preferable for precision.

DO NOT MODIFY `wobj0`

CREATE BACKUP FROM ROBOT AND OPEN IN ROBOTSTUDIO FOR PROBED DATA

Using actual tooldata and workobject data - Create a backup from your robot

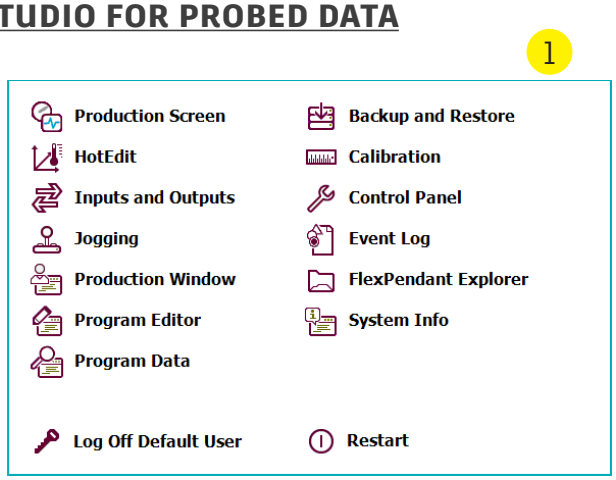
1. Create backup from robot controller on teach pendant
2. Open backup in RobotStudio

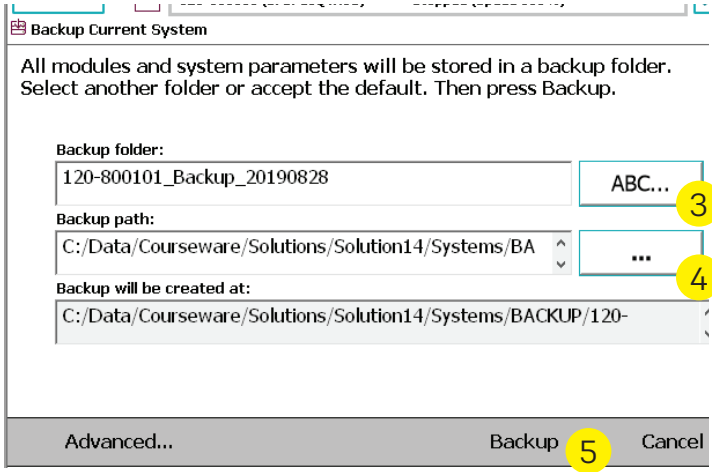
BACKUP FROM TEACH PENDANT

A backup saves the current state of the robot, including:

+ Files and folders stored under the Home directory
+ System parameters
+ Tasks, programs, and modules

Backup whenever making significant changes to your program or parameters.





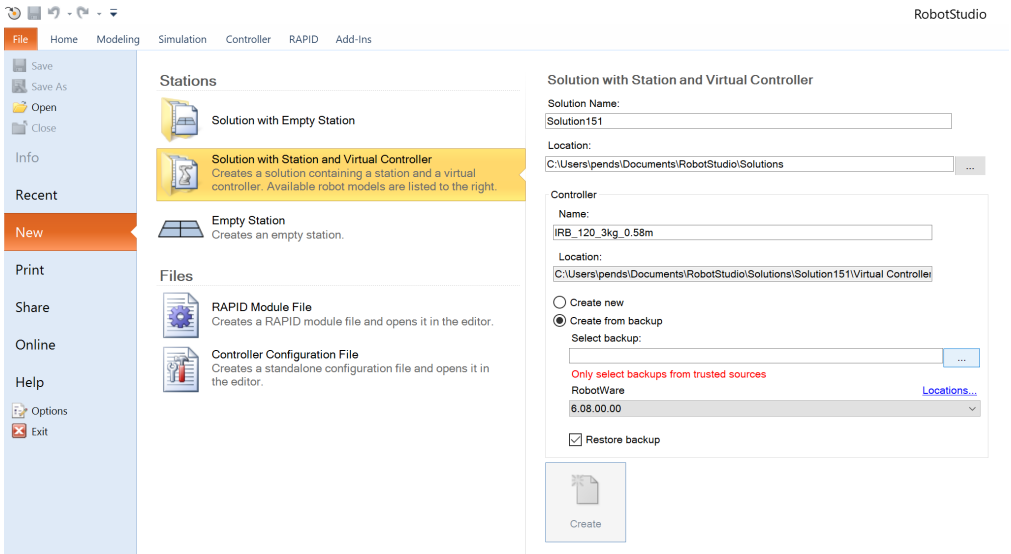
To Perform a Backup (on teach pendant):

1. From **Main Menu**, choose **Backup and Restore**.
2. Choose **Backup Current System.**
3. Change the name of the backup.
4. Set the storage location to your USB.
5. Click **Backup**.

RobotStudio Create from Backup

A digital copy of your robot system can be created from your backup in RobotStudio:

1. Under **File** in the toolbar, choose **New**.
2. Under **Stations**, choose **Solution with Station and Robot Controller**.
3. Input name and save location.
4. Click radial button **Create from Backup** and select the backup location.

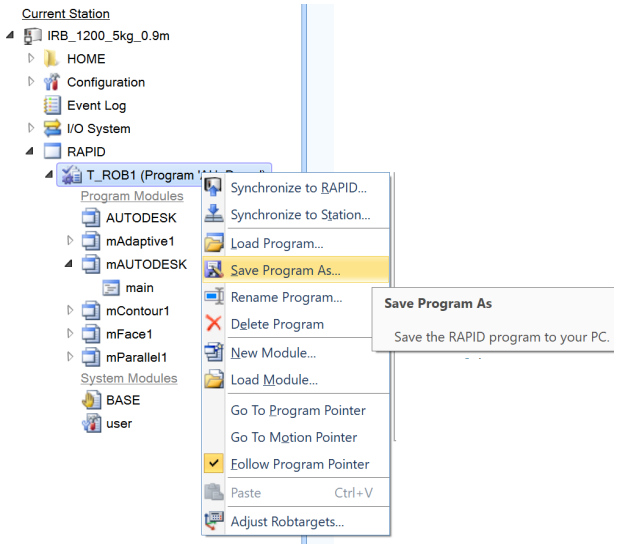


Continue steps to load program in RobotStudio from instructions earlier in the handout.

LOAD PROGRAM ON ACTUAL ROBOT USING USB OR ROBOTSTUDIO

Save program in RobotStudio

1. In **RAPID** tab
   - right click **T_ROB1**
   - Select **Save Program As…**
2. Save the program on a USB

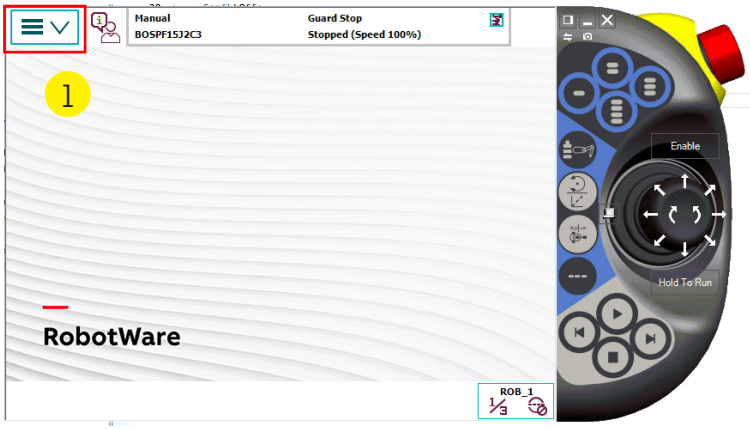


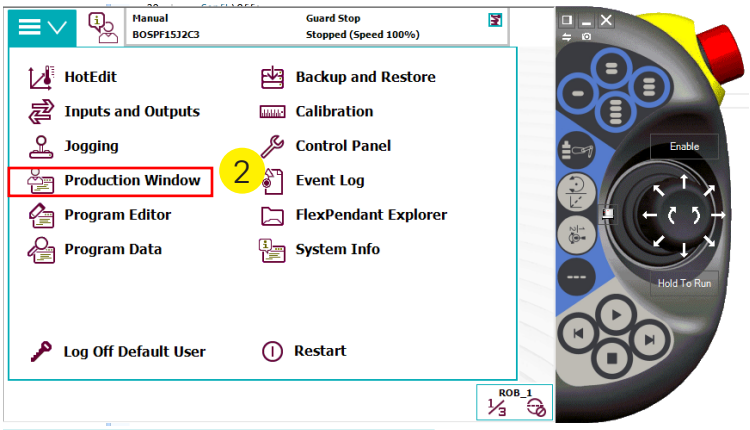
On teach pendant

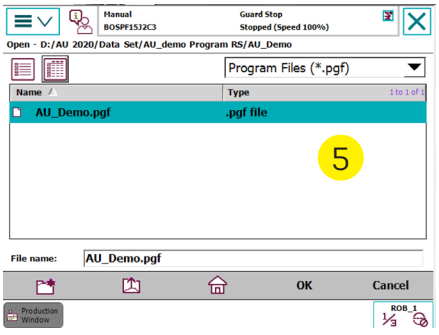
1. Select menu button in upper left corner of GUI
2. Select Production Window
3. Select **Load Program**
4. A warning will appear → Select **Yes**
5. Navigate to the .pgf file and select **OK**

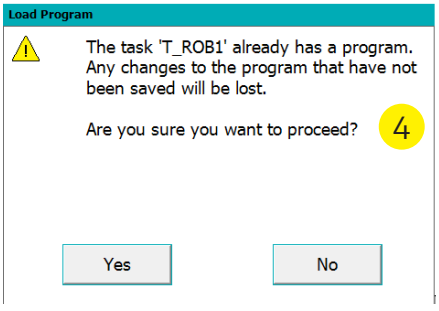
Loading programs on RobotStudio

1. Connect ethernet cable via service port to laptop with RobotStudio
2. Under **Controller** tab
   - **Add Controller → One Click connect…**
   - The robot will appear in the explorer menu
3. Select **Request Write Access** under **Controller tab**
4. Select **Grant** on the teach pendant
5. Load program using RobotStudio and then select **Apply** under **RAPID** Tab





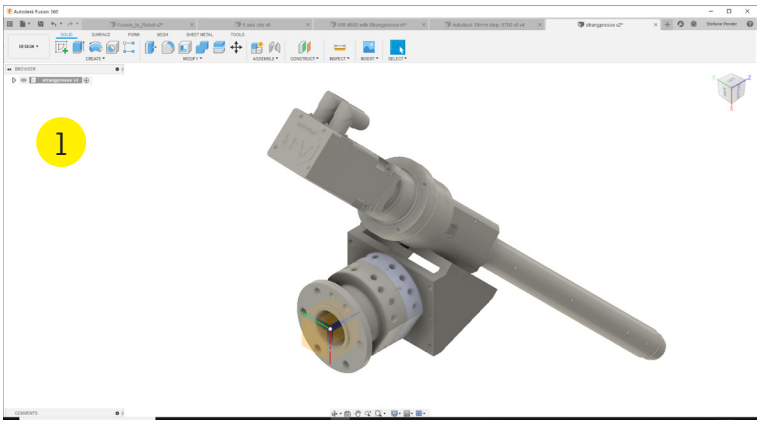


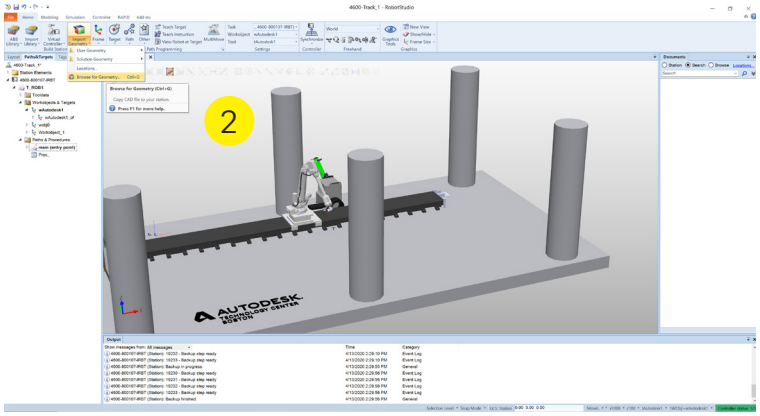


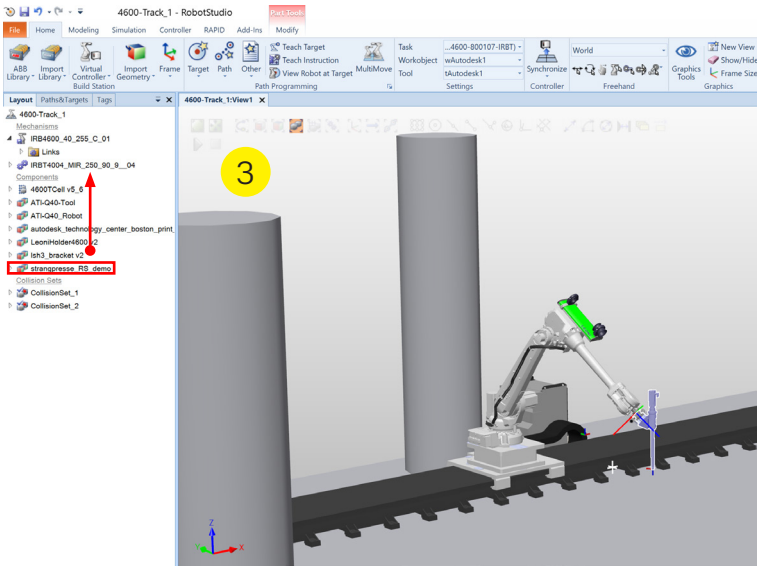
IMPORTING YOUR OWN TOOLS INTO ROBOTSTUDIO

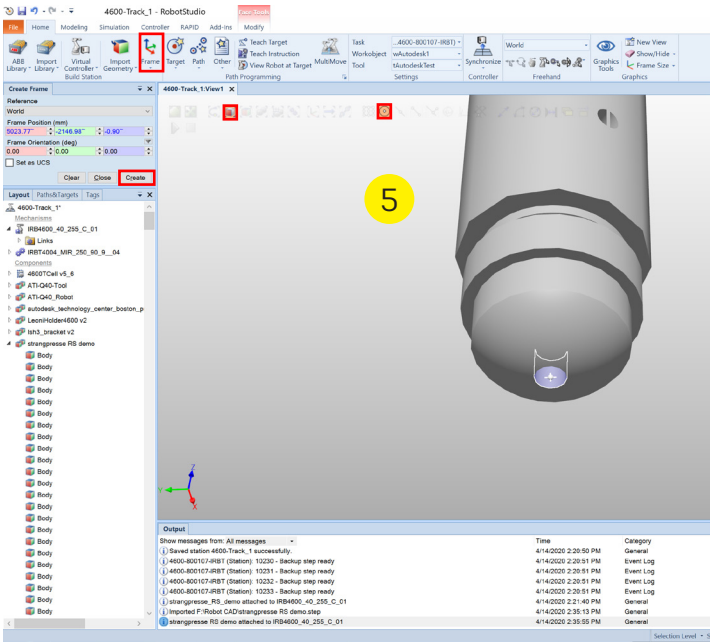
IMPORTING USER TOOL IN RS:

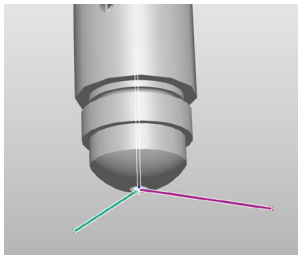
1. Position your tool in *Fusion* at the origin in an orientation matching the extruder.
   - Export your tool in .step or. iges file format
2. In *RobotStudio*:
   - Go to **Import Geometry**
   - Select **Browse for Geometry**
   - Select your .step or .iges file format.
3. Drag Tool onto robot
   - Select **Yes**: *Do you want to update the position of 'Tool'?*
4. To create tool data for the tool you may:
   - Probe actual tool and enter tool data (most accurate and recommended)
   - Select point on tool CAD and create tool data (only for virtual simulation). See step 5.
   - See Shop Staff for this step
5. Select point on CAD for Tool Center Point (TCP) tool data.
   - Select **Frame→Create Frame**. Select **Surface Selection** and **Snap Center**. Select the point where your TCP is located and select **Create** to make a frame.





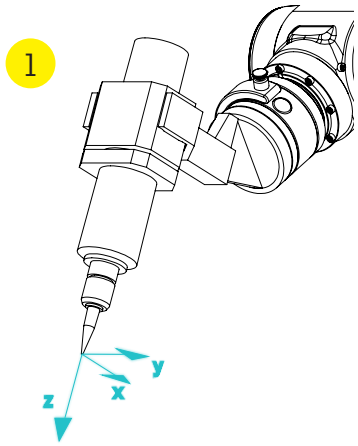


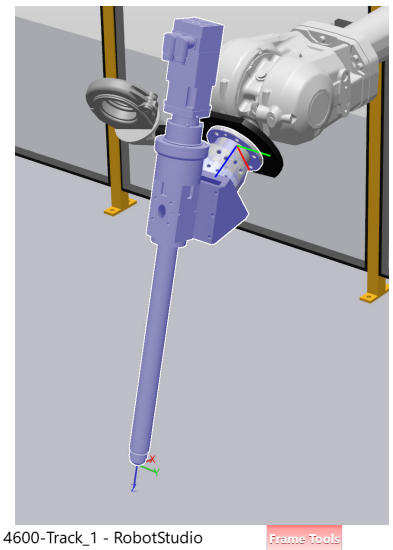


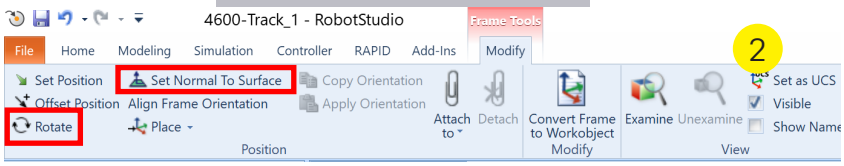


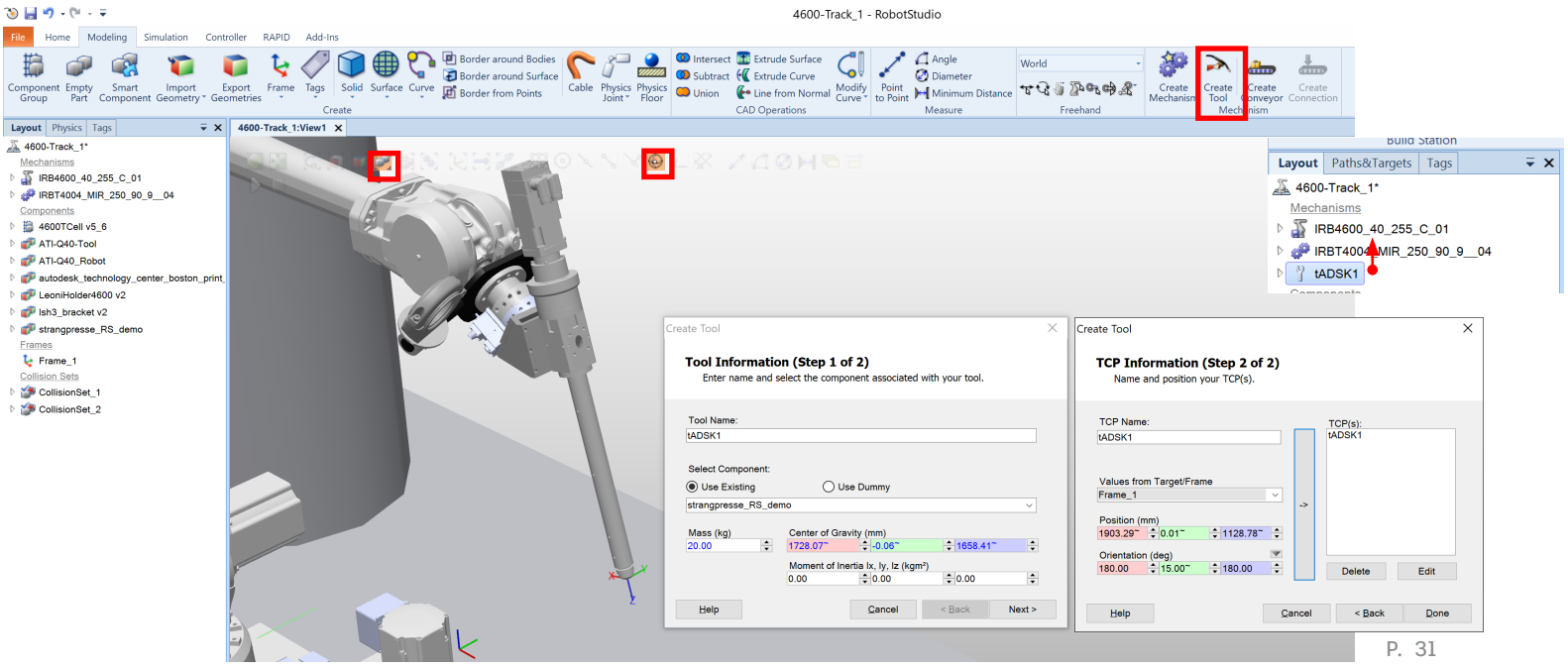
TCP ORIENTATION

1. The tool orientation should look like this image→
2. In *RobotStudio*:
   - Double click the Frame
   - **Frame Tools** will appear in the Menu ribbon
   - Use **Rotate** and **Set Normal to Surface** to reorient the frame
3. Make this Frame *Tool Data*
   - Select **Modeling** tab → Select **Create Tool**
   - **Tool Information (Step 1 of 2)** Rename tool *tADSK1* (or whatever tool #)
   - Select **Use Existing** and scroll down to your tool CAD
   - Enter estimated weight
   - For **Center of Gravity** → Use **Snap Part** Selection and **Snap Gravity** (red boxes below). If there is an issue entering values, make sure you click in the X entry value box of Center of Value before you select the part.
   - Select **Next**
   - **Tool Information (Step 2 of 2)**→ Select the Frame you just created in the drop down menu. In the example, *Frame_1*. Select the Arrow **->**. *tADSK1* will be listed under *TCP(s)*
   - Select **Done**
   - Drag and drop **tADSK1** onto your robot
   - Select **No**: *Do you want to update the position of 'tADSK1'?*



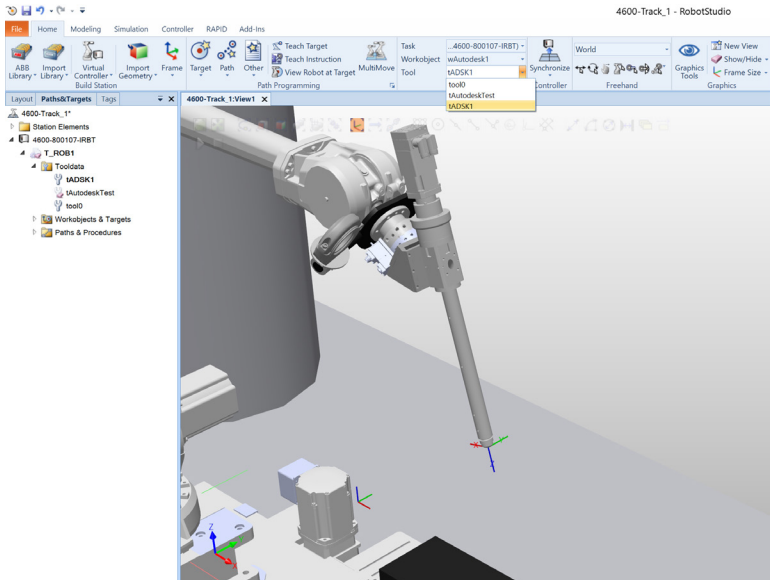




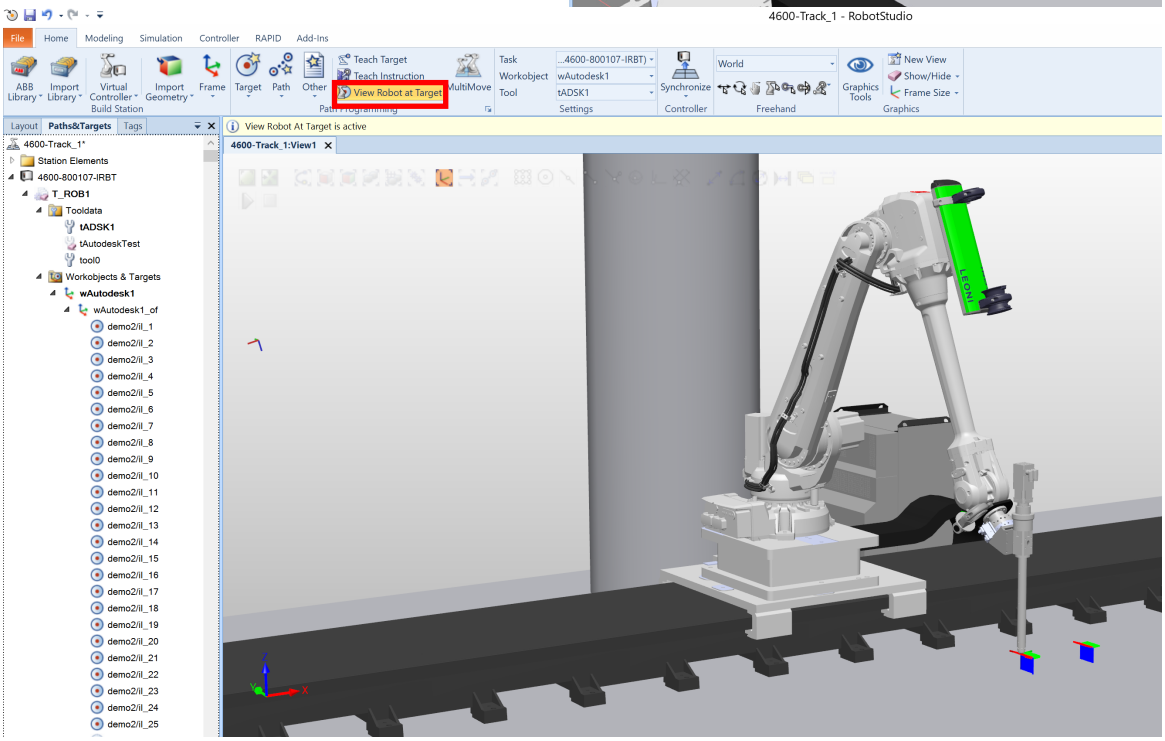


TOOL SELECTION

1. Select **Paths & Targets**
   - Make sure under *Tool*, **tADSK1** is selected.
2. You can view **Robot at Target**
   - Select **View Robot At Target**
   - Expand Targets→ Double click *Workobjects & Targets → wAutodesk1*
   - As you select each target the robot will adjust its position to reach the target.



4600-Track_1 - RobotStudio



IMPORTING WORKOBJECT CAD

1. You can import geometry and create workobjects using the steps above.
2. Create a Frame by using the Snap Tools.
   - Right Click the Frame→ *Convert Frame to Workobject*