How to build a station
in RoboDK

In-Position Technologies
www.iptech1.com

+1-855-692-7772
info@robodk.com
www.robodk.com

RoboDK
ROBOT DEVELOPMENT KIT

Contents

Getting Started

This document shows an example of how to prepare a basic offline station in RoboDK for robot offline programming. In this example, a UR robot is simulated and programmed for a robot painting application.

All the settings of a RoboDK project are saved as a RoboDK station (RDK file). A RoboDK station contains all robots, tools, reference frames, objects and other parameters. The RoboDK station is stored in one file (RDK extension). It is not required to keep a separate copy of the robot files, tools and objects as they are saved as one RDK file.

The videos available in the examples section of RoboDK website provide an overview on how to use RoboDK for different applications: https://www.robodk.com/examples.
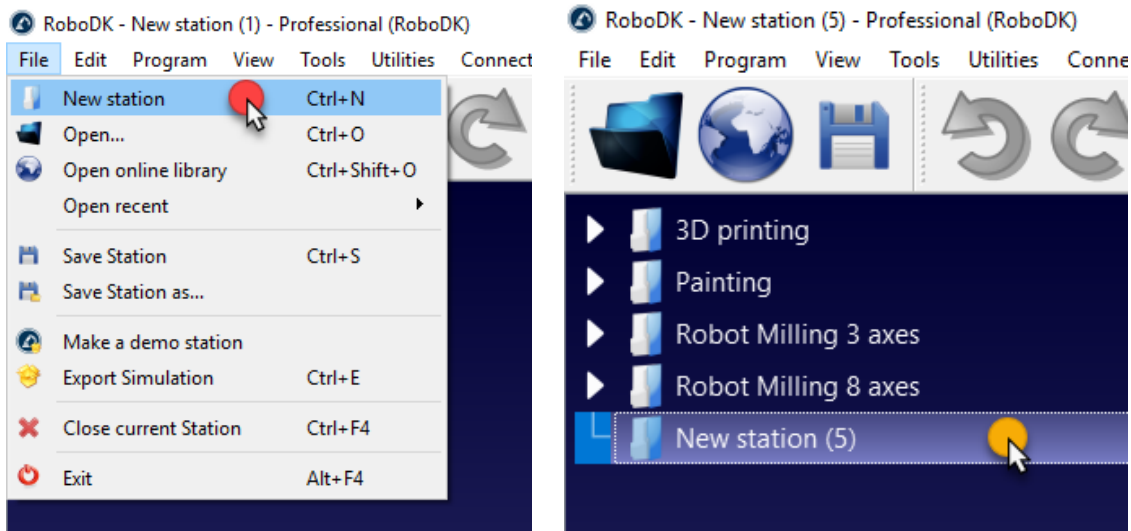
New project

Follow these steps to create a new RoboDK project (RDK station):

1. Download and install RoboDK from the website: https://robodk.com/download

2. Double click the shortcut on the Desktop
3. If other stations are open:

    select **File→** **New Station (Ctrl+N)** to start a new project

Multiple RoboDK projects can be open at the same time. Double clicking the Station icon in the tree will activate and display that project.

Select a robot

New robots can be added from a local drive or from the online library:

1. Select **File→ ⊕ Open online library** (Ctrl+Shift+O). A new nested window will appear showing the online library

   It is also possible to select the corresponding button in the toolbar. ⊕

2. Use the filters to find your robot by brand, payload, ...
   In this example, we will use a UR10 robot (10 kg payload robot and 1.3 m reach).
3. Select **Download.** The robot should automatically appear in the station in a few seconds.
4. The online library can be closed once the robot is loaded

Tip: Selecting **reset filter** in the online library will remove any filter that was used.

Tip: Alternatively, it is also possible to download the robot files (.robot extension) separately, from the website: http://robodk.com/library and open them in RoboDK by drag & dropping the file to the main window or by selecting **File→Open**.

Note: Every time a new robot is loaded in RoboDK, a new reference frame is added representing the robot base frame.

Note: Loading robots from the online library will store them in the local library. The default location of this folder is: C:/RoboDK/Library/.
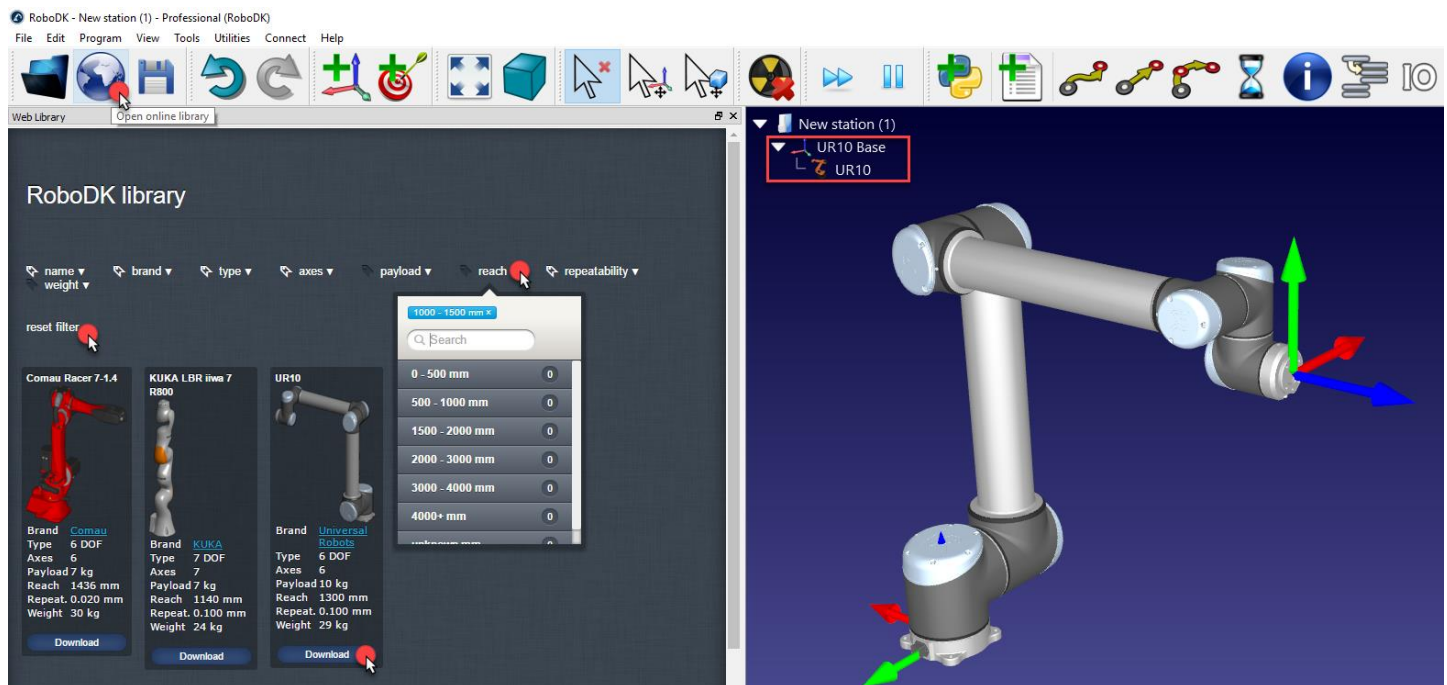
Add a Reference Frame

A Reference frame allows placing objects with respect to a robot or with respect to other objects in the 3D space (including position and orientation).

Note: More information about reference frames in RoboDK in the <u>reference frames</u> section.

To add a new reference frame:

1. Select **Program→** ![icon] **Add Reference Frame**
   Alternatively, select the equivalent button in the toolbar
2. Double click the reference frame (on the tree or on the 3D geometry on the main screen) to enter the coordinates shown in the image (X,Y,Z position and Euler angles for the orientation). The mouse wheel can be used on top of each case to quickly update the position of the reference frame on the main screen.
   The following colors are used by default:

   - X coordinate → Red
   - Y coordinate → Green
   - Z coordinate → Blue
   - 1st Euler rotation → Cyan
   - 2nd Euler rotation → Magenta
   - 3rd Euler rotation → Yellow

Tip: Select Tools→Options→Display→Display XYZ axis letters to see the Coordinate system axes.

3. Select **View→Make reference frames bigger (+)** to increase the size of the reference frames
4. Select **View→Make reference frames smaller (-)** to decrease the size of the reference frames
5. Select **View→Show/Hide text on screen (/)** to show or hide the text on the screen
6. Optionally, rename any reference frame or object in the tree by selecting **F2**
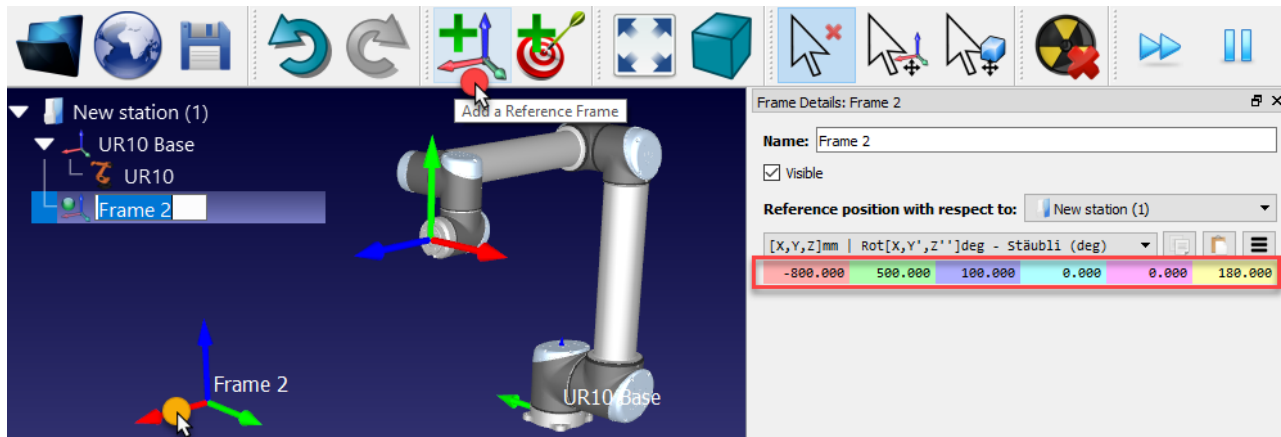
If more than one reference frame is used, it is possible to drag and drop them inside the Station Tree to match the dependency that exists in the real setup. For example, the reference Frame 2 may be placed with respect to the robot base reference. In this case, if the UR10 Base reference is moved, the Frame 2 is also moved with it. It is important to take this into account if other robots or reference frames are used. The next image shows the difference in dependency.

Tip: Drag and drop reference frames (or any other items) in the tree holding the mouse right click to reorder the items in the Station Tree (before other items, after or as nested items).

Even if the dependency is different, it is still possible to enter or retrieve the coordinates of any reference frame with respect to any other reference frame, as shown in the next image. Most robot controllers require the coordinates of the reference frame with respect to the robot base frame.
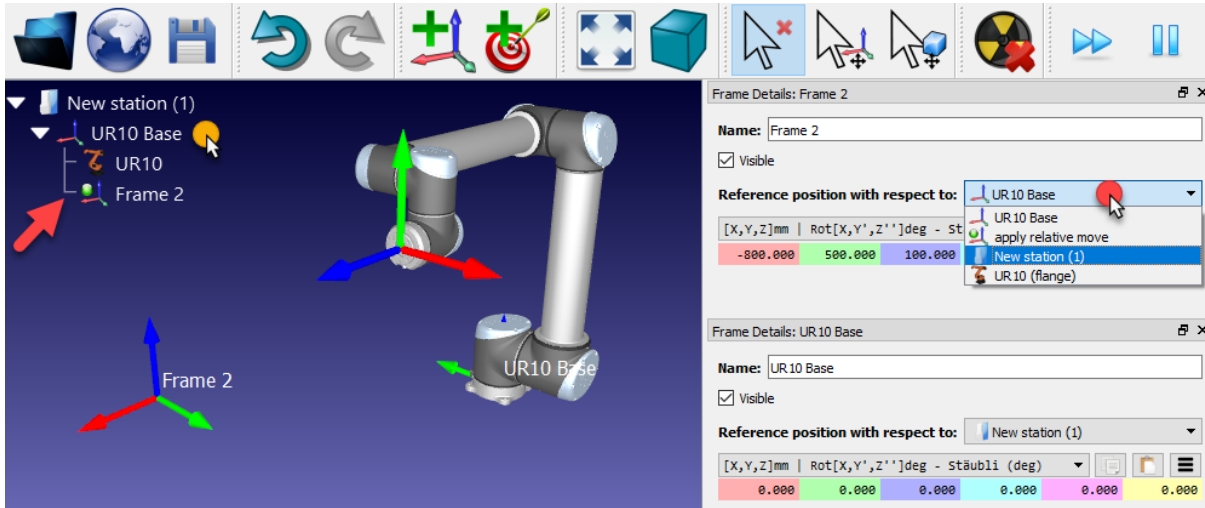
Reference frames can also be moved in the main screen by holding the **Alt** key, or selecting the corresponding button in the toolbar ![icon]. Then, drag the reference with the mouse on the screen. As the reference is being moved, the corresponding coordinate values will be updated.
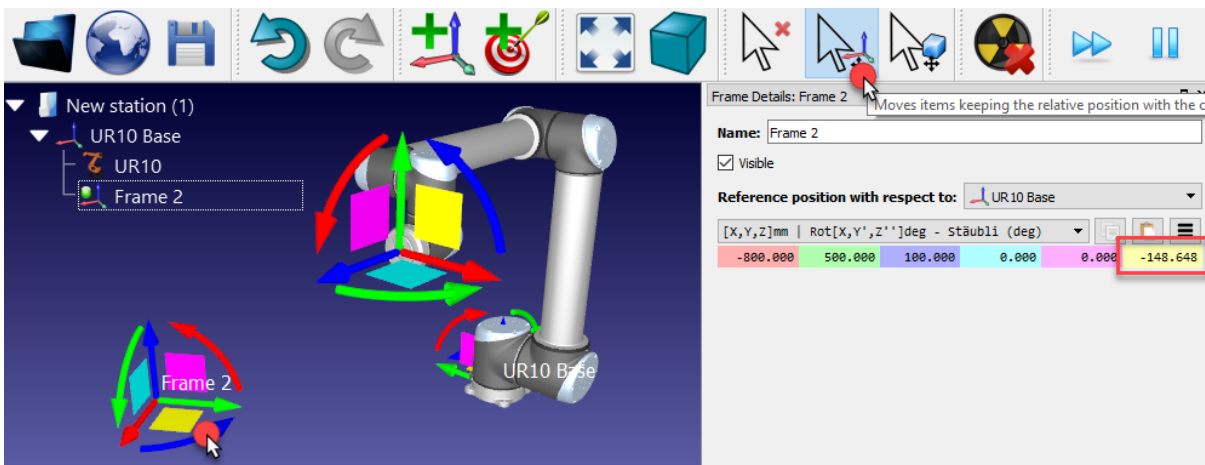

Import 3D objects

RoboDK supports most standard 3D formats such as STL, STEP (or STP) and IGES (or IGS) formats. Other formats such as WRML, 3DS or OBJ are also supported (STEP and IGES are not supported on Mac and Linux versions). To load a new 3D file:

1. Select **File→** ![icon] **Open**
2. Select the object **Object Inspection** available in RoboDK's default library:
   C:/RoboDK/Library/**Object Inspection**.
3. Alternatively, drag & drop files into RoboDK's main window to import them automatically
4. Drag & drop the object to the reference frame **Frame 2** (inside the station tree)

Tip: Drag & drop the object by holding the right click to reorder items inside the tree.

Important: It is important to place the object nested in the reference frame to teach the targets with respect to the object. Then, if the reference frame is moved the targets are moved together with the object.
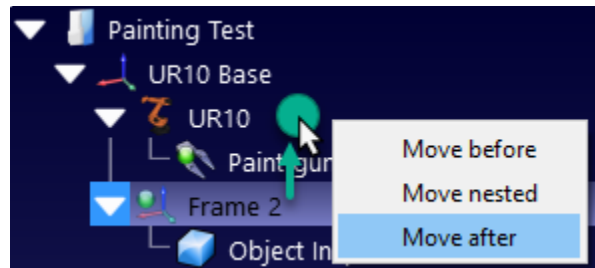
Tip: For large STEP or IGES files it is possible to accelerate import speed by selecting:
Tools→Options→CAD→Fast import settings. More information available in the STEP/IGES section.

Tip: Although objects can be moved with respect to the reference frame they are attached to (by double clicking the object and entering the coordinates), it is recommended to maintain them always with respect to the given reference frame and move the reference frame instead. This allows matching the object reference properly by just updating the reference frame from/to the robot controller.
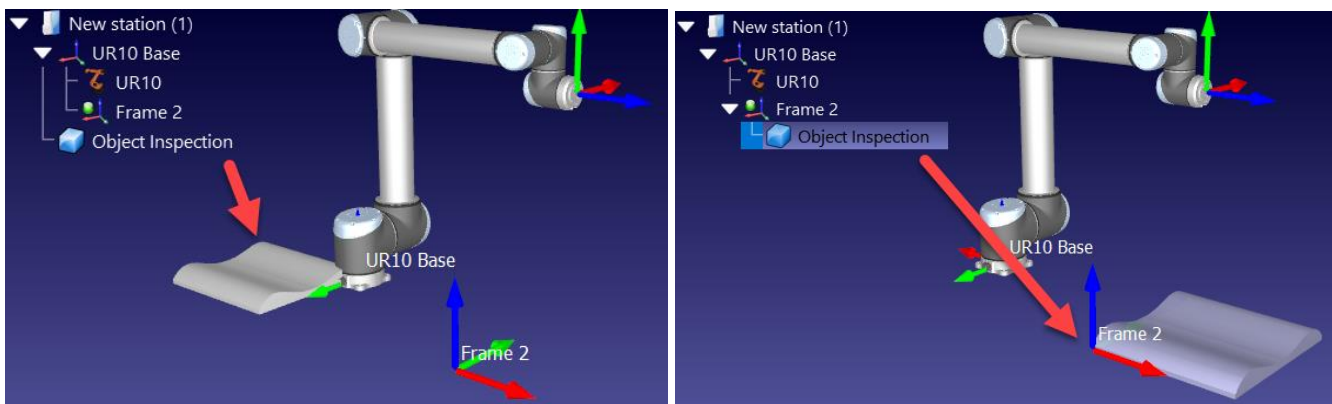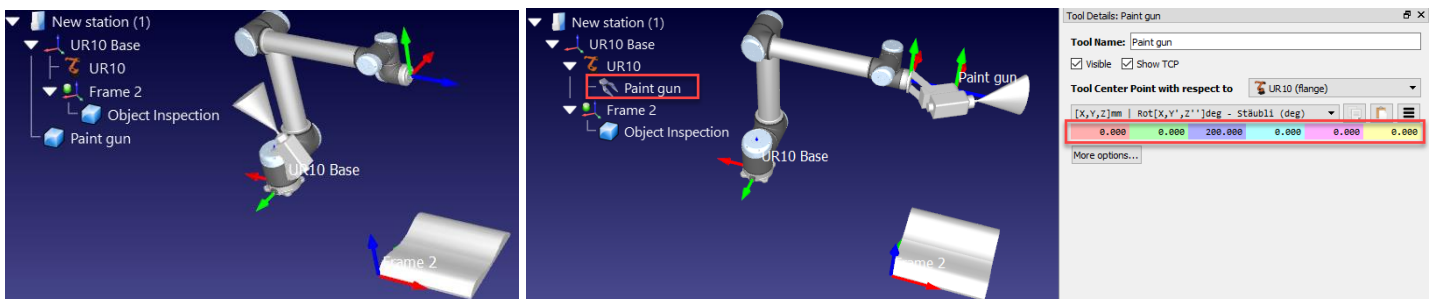

Create a TCP

New tools can be created in RoboDK from previously loaded 3D geometry:

1. Select **File→ Open** (as described in the previous section)
2. Select the Paint gun.stl file to add it as an object (it will be added at the robot base frame)
3. Drag & drop the object to the robot item inside the station tree as shown in the next image

New tools can be loaded or saved as a .tool format.

By default, RoboDK will define the TCP at the position [X,Y,Z]=[0,0,200] mm. This can be changed by entering the coordinates manually and/or by moving the TCP holding the ALT+Shift key as shown in the next image:

1. Hold **ALT+Shift** or select the highlighted button from the toolbar
2. Select the light blue plane (XZ plane of the TCP) and drag the TCP approximately towards the surface of the spray gun, as shown in the next image

3. Select the Green rounded arrow (rotation around the Y axis) to make the Z axis point outwards
4. Once an estimate of the coordinates is obtained it is possible to touch up these values manually by double clicking the Paint gun object. The mouse wheel can be used on top of each case to quickly update the position on the main screen.

Note: These are estimate values according to the 3D drawings. If the definition of the TCP is calibrated on the robot it is possible to import it by pasting the coordinates in that box.
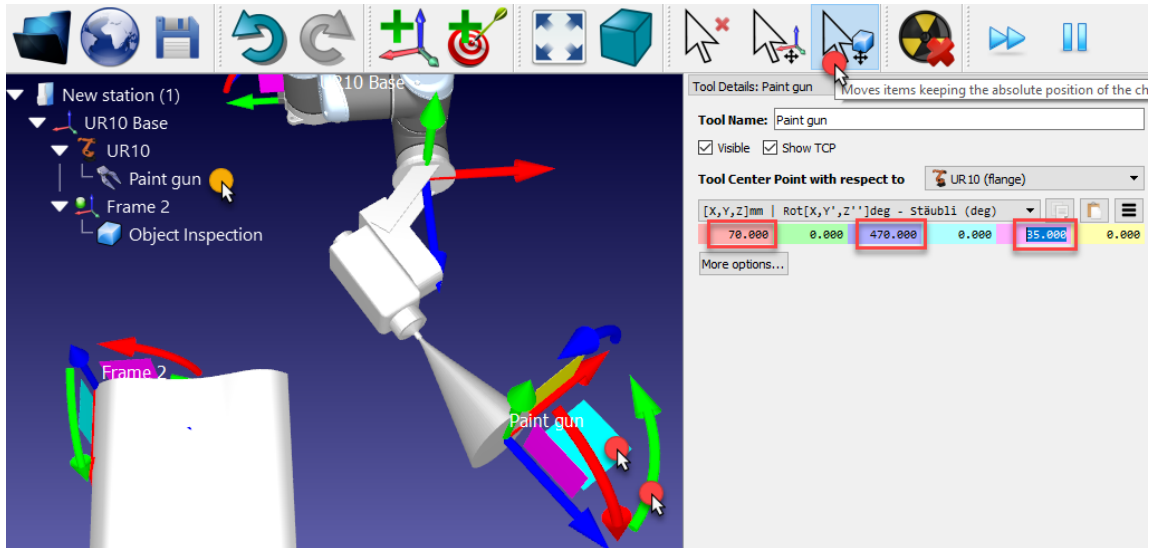
At this point, the station can be saved:

1. Select **File→** 💾 **Save Station** (Ctrl+S)
2. Save the file as Paint Test.rdk. The Window title and the Station name will be updated

Note: The following 2 subsections are optional and show hypothetical situations that require defining the TCP in different ways. The modifications in these subsections will not be used in the following steps of this example. To continue with this example, continue with the section showing how to create targets.

Set a relative TCP

It is possible to place a second TCP with respect to the first one, for example, to define a given standoff or to place a cutter with respect to a spindle TCP.

These steps assume that a second TCP must be added with respect to the first one at 150 mm along the Z axis:

1. Right click the robot
2. Select 🔧 **Add Tool (TCP)**. A new item called Tool 2 will appear.
3. Double click this new TCP
4. Select **Tool Center Point with respect to →Paint gun**
5. Enter the coordinate Z to 150 mm and set the other translations and rotations to 0.
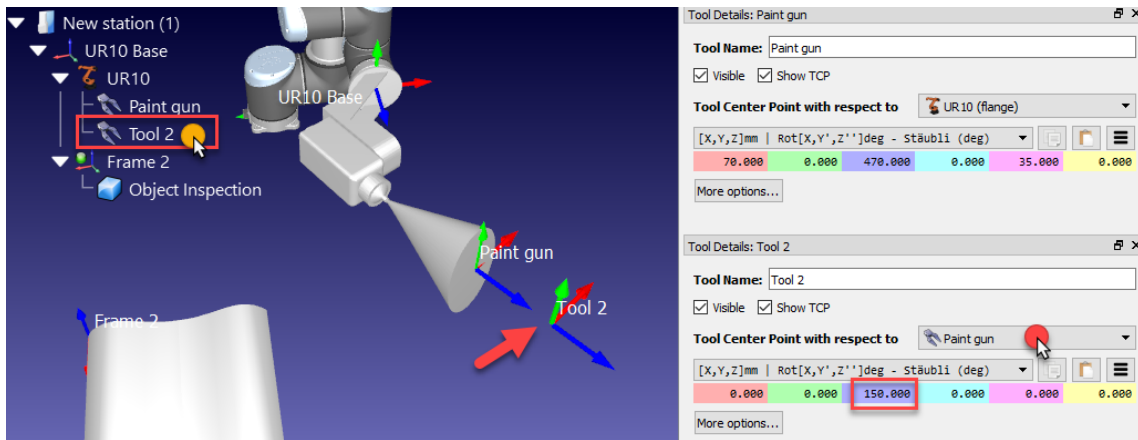
This new TCP relative to the previously defined TCP will be removed in the next sections of this demonstration.

Move the Tool geometry

The geometry of the tool might not be aligned properly with respect to the robot flange (adaptor reference frame) when it is loaded in RoboDK. The following steps assume that a mistake was made and the tool was mounted 180 degrees around the Z axis, so the following correction should be applied:

1. Select More options in the Tool details window
2. Enter 180 in the yellow case (Z rotation) of the paint gun geometry, as shown in the following image. The mouse wheel can be used on top of each case to quickly update the position on the main screen.
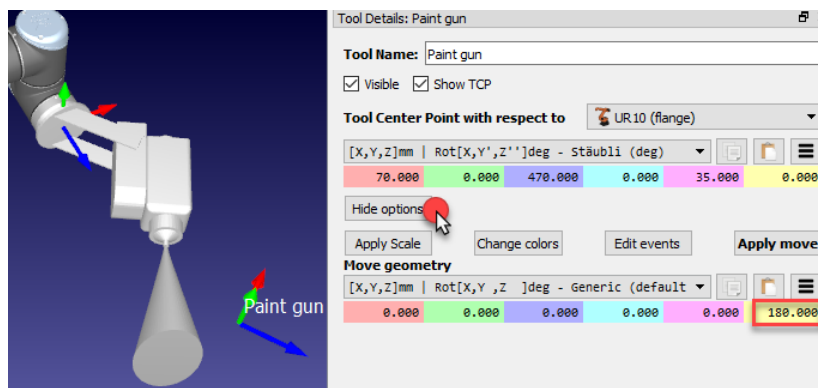
Tip: If the TCP was defined before this operation, the TCP won't match the geometry anymore. To update the position of TCP to the right location we could follow the steps described at the beginning of this section (by holding the Alt key or entering the values manually). Alternatively, we could rotate the TCP around the robot flange by selecting the **Custom Pose** representation method (as shown in the next image) and insert the Z rotation pre-multiplying the TCP definition (at the beginning of the TCP line definition).
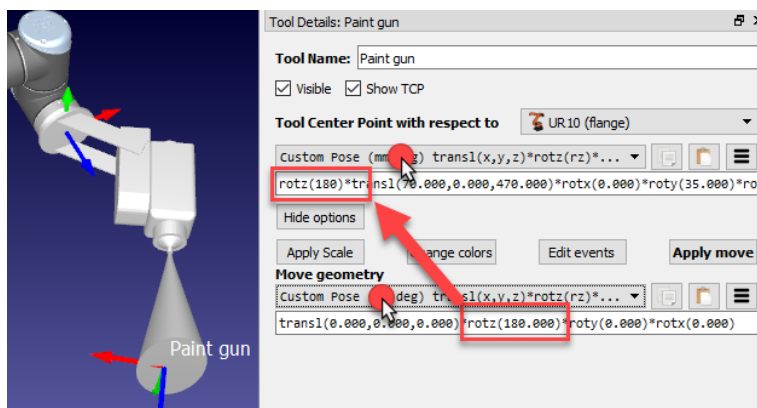
Note: More information about how to provide transformations and poses available in the reference frames section.

Create Targets

Robot positions are recorded as Targets. Follow these steps to create two targets as a new home target and approach target respectively:

1. Double click the robot to show the robot panel
2. Select Paint gun as the Tool Frame. Once a tool or a reference frame becomes active it will show a green dot.
3. Select Frame 2 as the Reference Frame
4. Hold the **Alt** key and move the robot by dragging it through the TCP or the robot flange to a safe position, free of collisions with any objects. Alternatively, move the coordinates of the Tool Frame (TCP) with respect to the reference Frame.
5. Use the **Other configurations** section to switch between different robot configurations and make sure that none of the robot axes are close to the axis limits.

Tip: In general, it is better if the first target of a program has the joint axes as centered as possible (the joint sliders are as centered as possible, as shown in the following image). This makes sure that the robot won't reach the axis limits as it follows linear moves along the program. This is possible by changing the robot configuration.
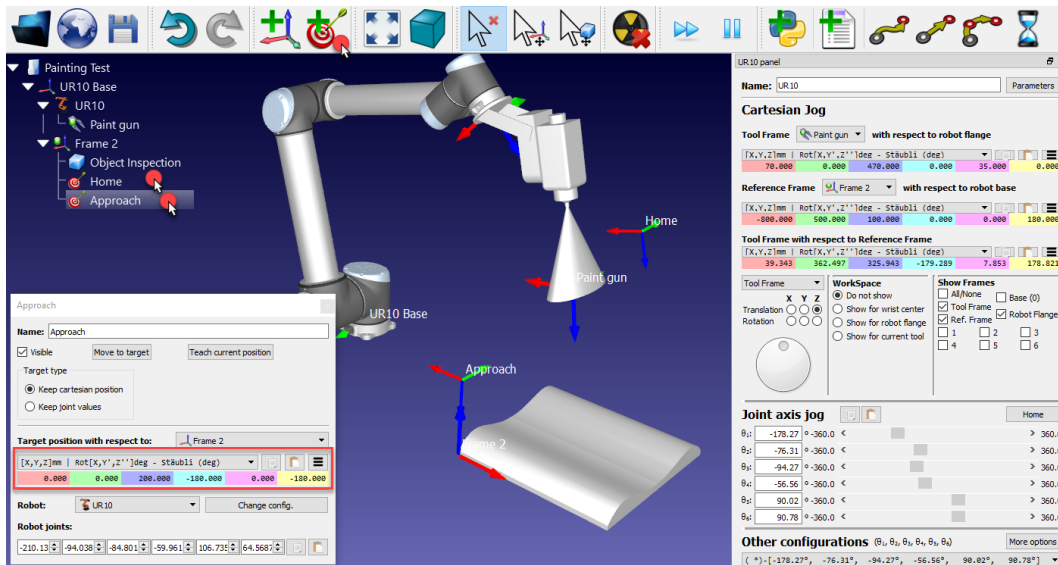
6. Select **Program→ Teach Target** (Ctrl+T), or the corresponding button in the toolbar (as shown in the image). The target will be placed as a dependency of the active reference frame and will automatically remember the current robot position (cartesian and joints axes).
   In this example, the robot joint coordinates used for the first target are: [-150, -75, -90, -60, 70, 110]
   deg. These values can be copied from this text and pasted in the **Joint axis jog** of the robot panel using the corresponding button.

Note: Selecting another reference frame will place the new target in that reference frame instead.
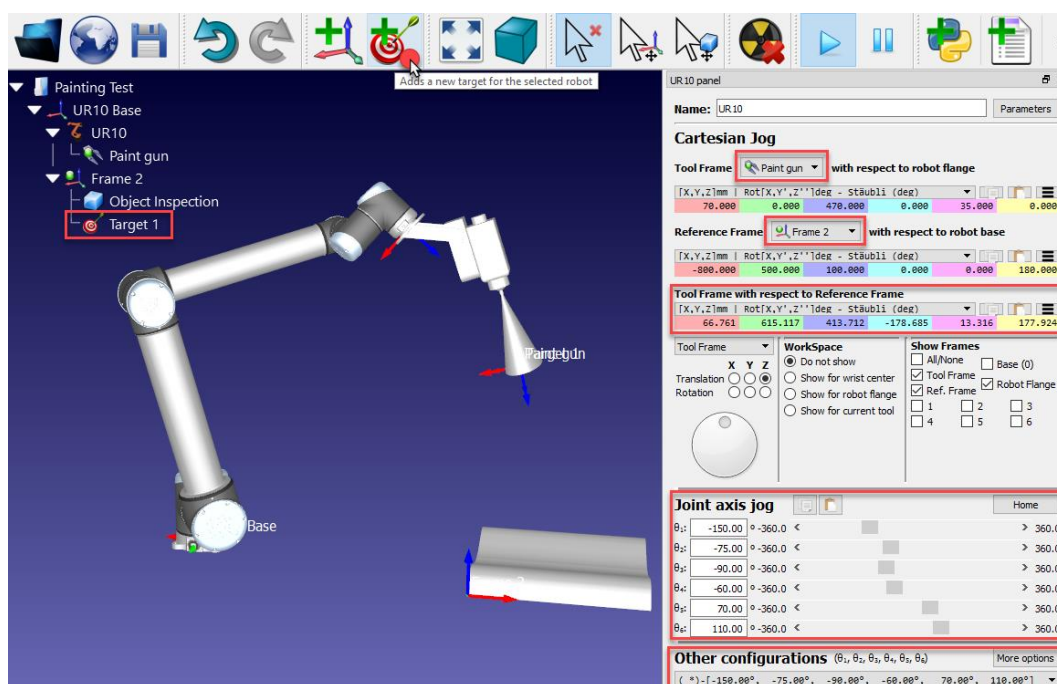
7. Rename the first target as **Home** by pressing **F2**. Alternatively, select **Tools→Rename item**.
8. Move the robot closer to one edge of the part (by dragging the tool using the Alt key, entering coordinates or jogging the axis manually)
   In this example we used the following robot joint coordinates [0,0,200,180,0,180] deg.
9. Select **Program→ Teach Target** (Ctrl+T) or the appropriate button in the toolbar to create a new target
10. Rename the target to **Approach** as shown in step 7

11. Select the Home target and the Approach target alternatively to see the robot moving between the two targets
12. Right click the target and select **Teach Current Position** (Alt+double click) if a different position needs to be recorded for one of the targets
13. Right click the target and select ⚒ **Target Options…** (F3) to open the target options window shown in the next image

If required, provide different values to define the targets.

Add an Approach Program

Follow these steps to create a program that moves from the **Home** target to the **Approach** target:

1. Select **Program→**📄 **Add Program** from the menu or the corresponding button in the toolbar (as shown in the next image)
2. Rename the program to **ApproachMove**
3. Select the **Home** target
4. Select **Program→**🔴 **Move Joint Instruction** (or the corresponding button in the toolbar)
Two instructions will be added automatically to tell the robot what tool frame and reference frames we are using

Note: If no target is selected, a new target will be created at the same location of the robot.

5. Select the **Approach** target
6. Select **Program→**🔴 **Move Joint Instruction** again

Double click the **ApproachMove** program and it will execute the program simulation. The simulation bar and an estimated cycle time will be displayed.
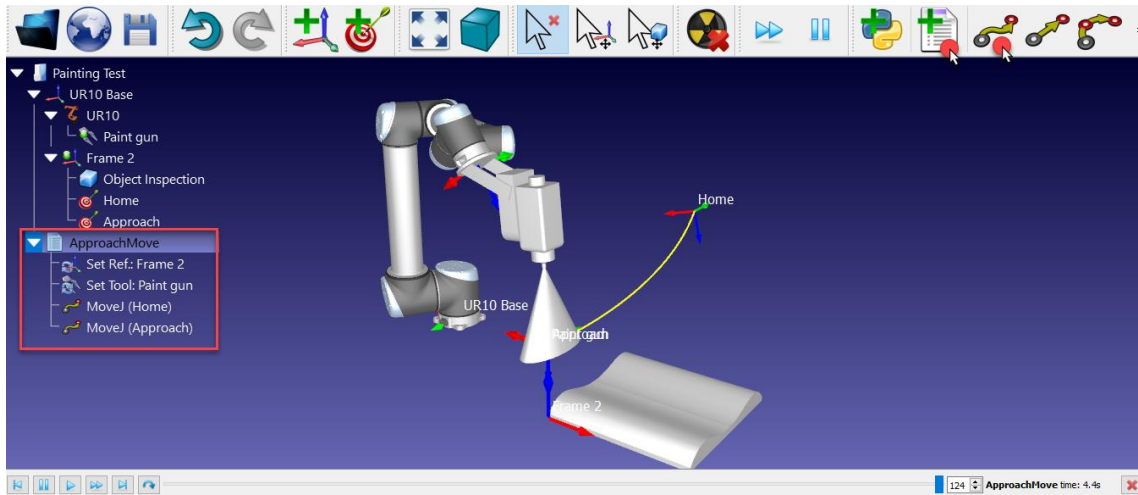
Note: More information about how to add robot programs and instructions is available in the robot program section.

Create Targets on Surface

The Create Targets on Surface feature, is useful for applications such as painting or inspection:

1. Select **Program→ Teach Target(s) on Surface** (Ctrl+Shift+T)
2. Move the mouse cursor over the part to see a preview of what the robot looks like when it reaches the part
3. Select a few points on the object (left click). Each mouse left click will define a new target keeping the Z axis of the TCP normal to the surface (perpendicular to the surface).
4. If necessary, adjust the orientation around the Z axis by moving the wheel on the left panel or pressing the left/right keys.
5. Hold **Alt** to move an existing target
6. Hold **Alt+Shift** to move an existing target while keeping it on the surface
7. Select Esc key or right click on the screen and select **Done** to exit the Create Targets on Surface mode
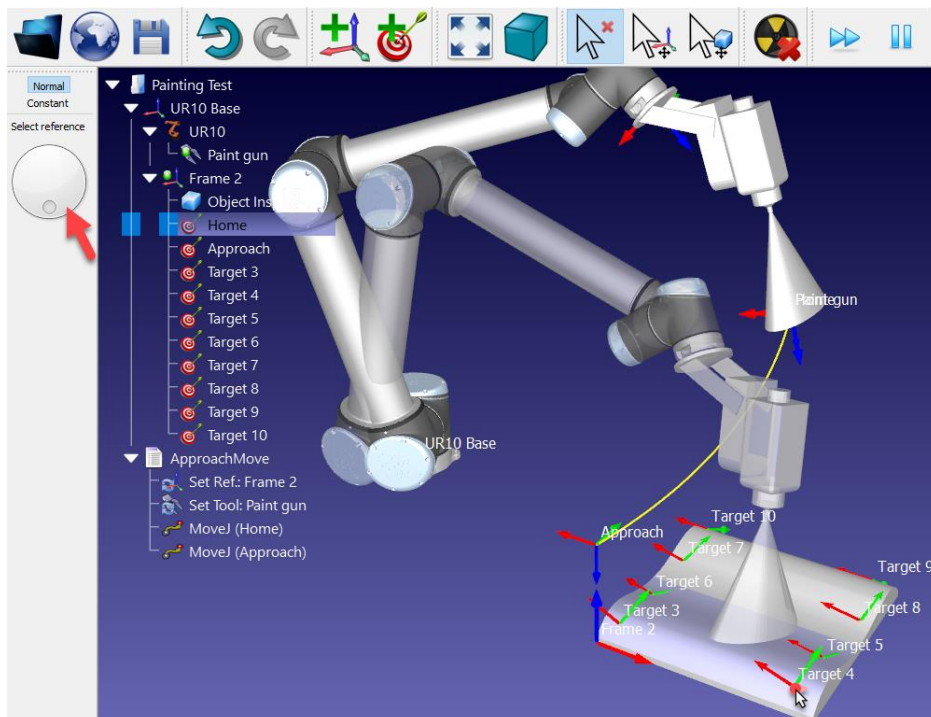

Once the targets have been created, generate a program following these steps:

1. Select all the targets created on the surface and right click

Tip: Hold the Ctrl key to select multiple targets. Alternatively, select the Target 3, hold shift, then select Target 10 to select all targets between Target 3 and Target 10.

2. Select **Rename group** from the pop up menu
3. Enter **Top Paint**. All selected targets will be renamed and numbered.
4. Right click on the targets again and select **Create Program**. A new program will be generated. The first movement will be a joint move and following movements will be linear.
5. Select **F2** to rename the program to **PaintTop**
6. Double click the **PaintTop** program to see the simulation moving along the targets
7. If required, reorder the movements by dragging the move instructions inside the program
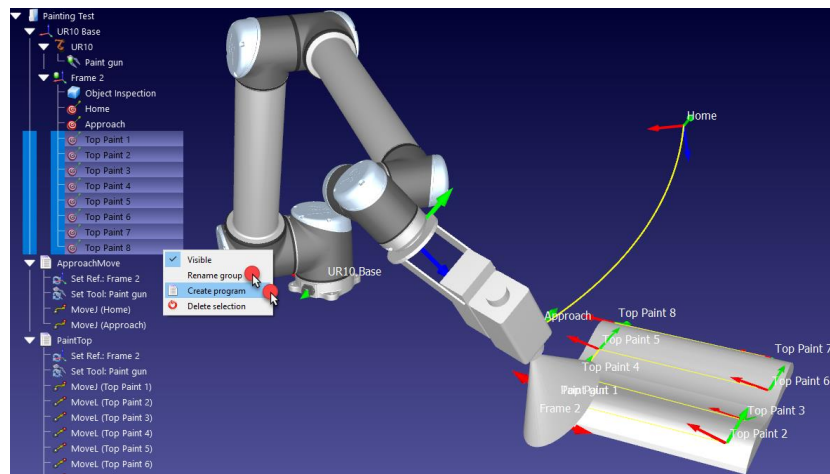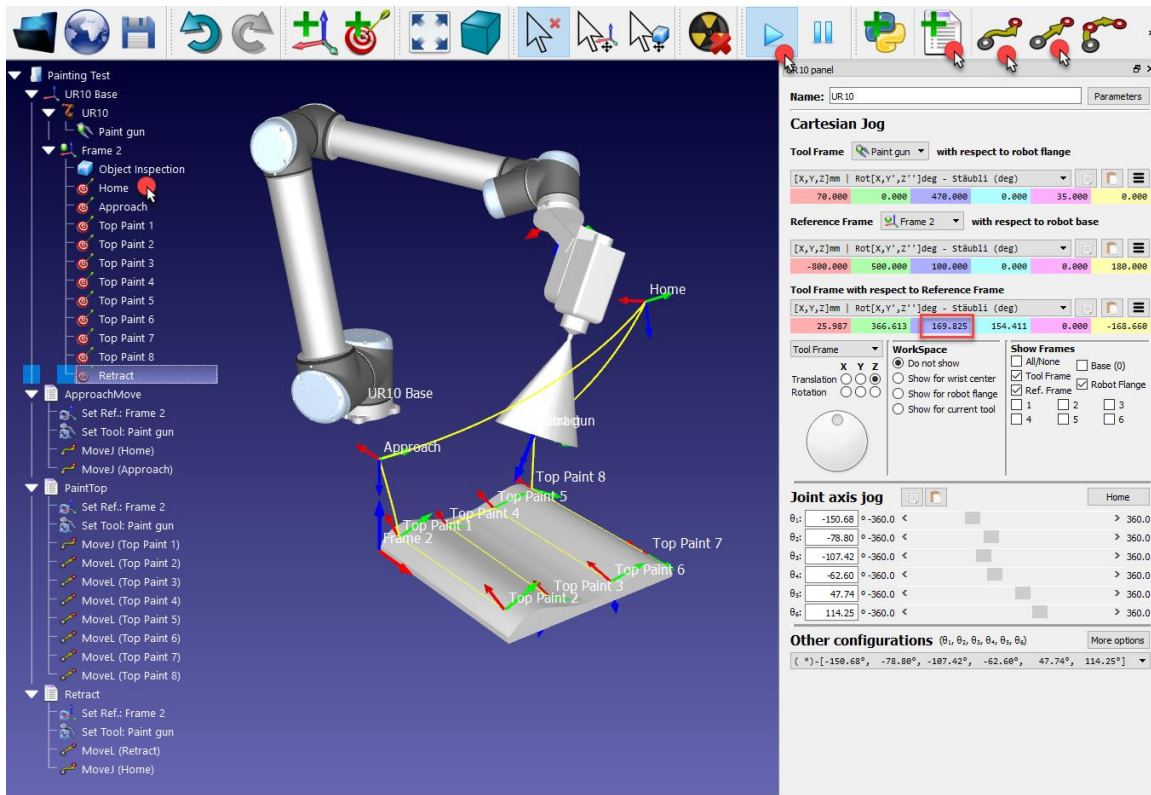

Add a Retract Program

Similar to the previous operations:

1. With the robot placed at the last target, move the robot upwards by increasing the Z coordinate of the TCP with respect to the reference frame in the robot panel (highlighted case in the next image)
2. Select **Program→ Add Program**, or the appropriate button in the toolbar.
3. Select **Program→ Move Linear Instruction**, or the appropriate button in the toolbar . Rename it to **Retract** by pressing **F2** key.
4. Select the **Home** target
5. Select **Program→ Move Joint Instruction**. A new move instruction will be added, linked to the Home target.

Simulate each program individually by double clicking it. The simulation can be accelerated by holding the Spacebar key or selecting the Fast Simulation button ▶▶/▶.

Main Program

A main robot program can execute the approach, paint and retract programs sequentially:

1. Select **Program→** 📄 **Add Program**

2. Select **Program→** 🗂️ **Program Call Instruction**
3. Enter the name **ApproachMove** or select Select program to automatically select it
4. Select **OK**
5. Repeat the previous steps for **PaintTop** and **Retract** as shown in the next image

Double clicking the Main Program will run the complete simulation. Right click the Main Program and select **Loop** to make it simulate in a loop.

Tip: Multiple lines can be typed when providing the program name as a program call. Each line will be converted to a new program call instruction.
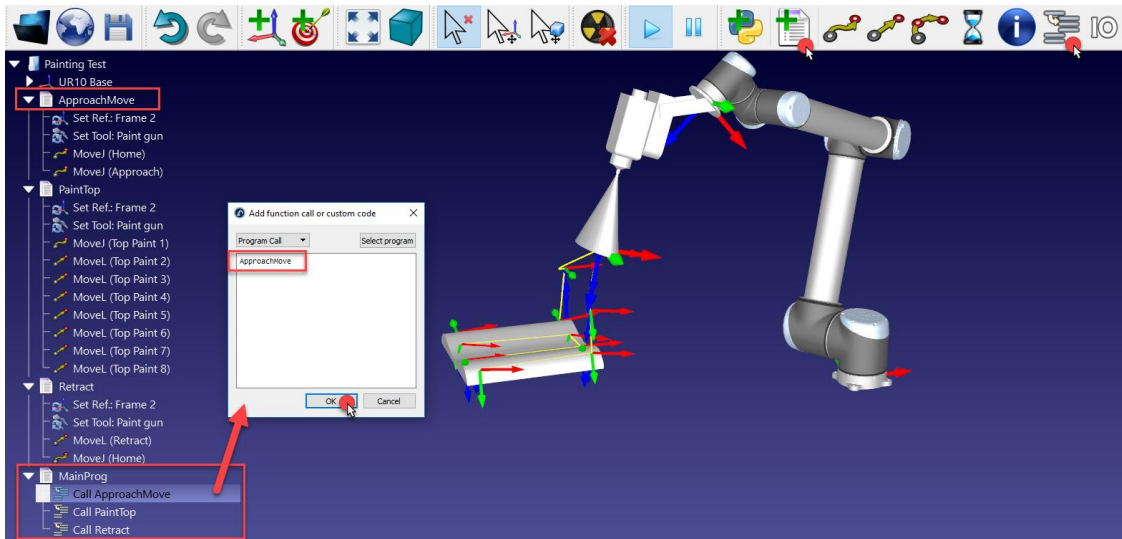
If the reference frame **Frame 2** is moved (for example, by holding Alt key and dragging the X/red axis of the reference frame), the object and targets will follow. If the targets become not reachable, a small warning sign will be displayed on the target icon as shown in the next image.

Tip: Select **View→Show/hide robot workspaces (*)** to display the robot workspace.
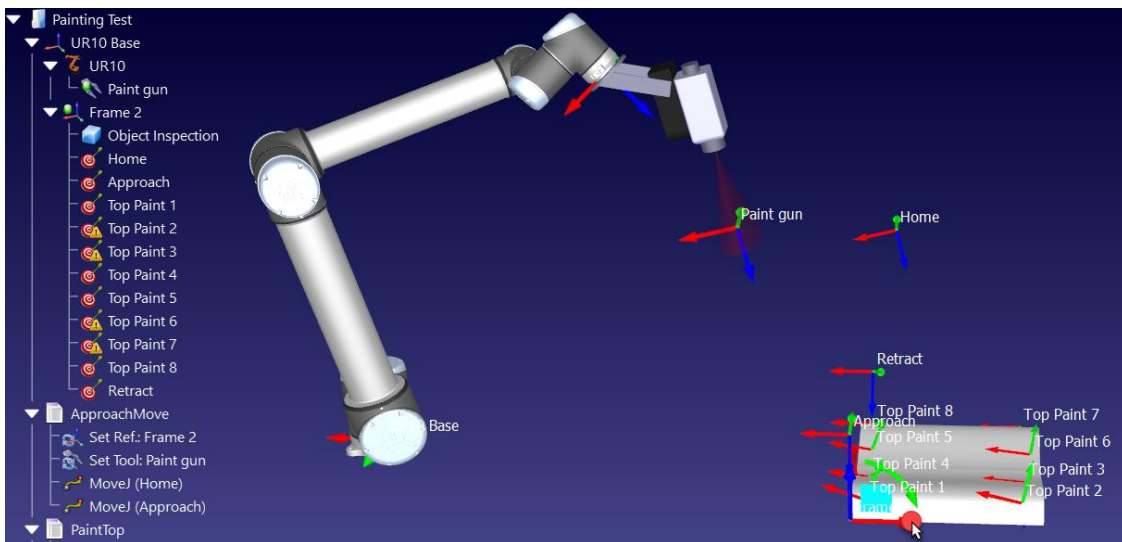
Export simulation

It is possible to export simulations as a 3D HTML or 3D PDF as well as distribute your station as a simplified version of RoboDK (RoboDK Viewer).

Note: Exporting as 3D PDF is not available on Mac and Linux versions,

3D HTML and 3D PDF documents can be generated following these steps:

1. Right click the Main Program
2. Select **Export Simulation…**
3. Select **Start**. The program will start and the simulation will record until the program is completed.
4. Save the file. After the file is saved it will be automatically opened to preview it.

HTML and PDF simulations are saved as one single file. Compressing an HTML file will remarkably reduce its size. PDF simulations must be opened with Acrobat Reader. Exporting PDF format is available for Windows platforms only.
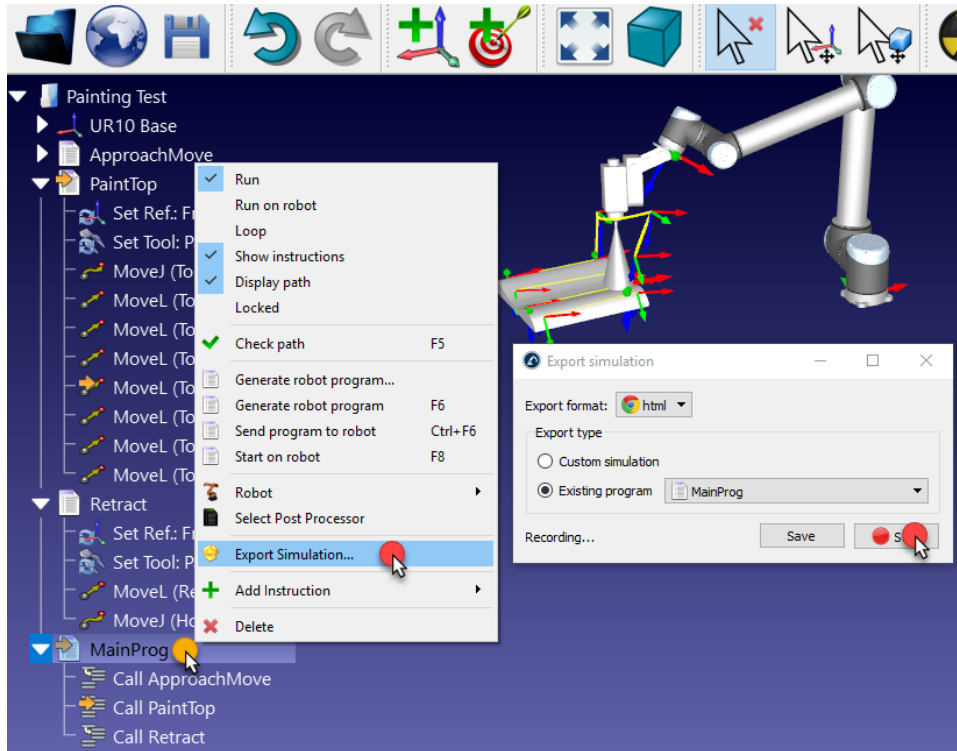
The following links are examples of 3D HTML simulations:

Getting Started with RoboDK

- o https://robodk.com/simulations/UR-Paint.html
- o https://robodk.com/simulations/Welding-3-Fanuc-Robots.html
- o https://robodk.com/simulations/Robot-Drawing.html

And the following links are examples of 3D PDF simulations:

- o https://robodk.com/simulations/UR-Paint.pdf
- o https://robodk.com/simulations/Welding-3-Fanuc-Robots.pdf
- o https://robodk.com/simulations/Milling-with-External-Axes.pdf


Another way of exporting 3D simulations is by generating a self-extracting EXE file or a ZIP file with a reduced version of RoboDK (available for Windows version only), as shown in the next image.

Note: This option is not available for Mac and Linux versions,

To generate this file:

1. Select **File→ Make a demo station**
2. Select the preferred output (ZIP or EXE format)
3. Select **Generate file**

After a few seconds, it will generate the desired file (20 MB) which is a simplified version of RoboDK. The station will automatically be loaded on start.
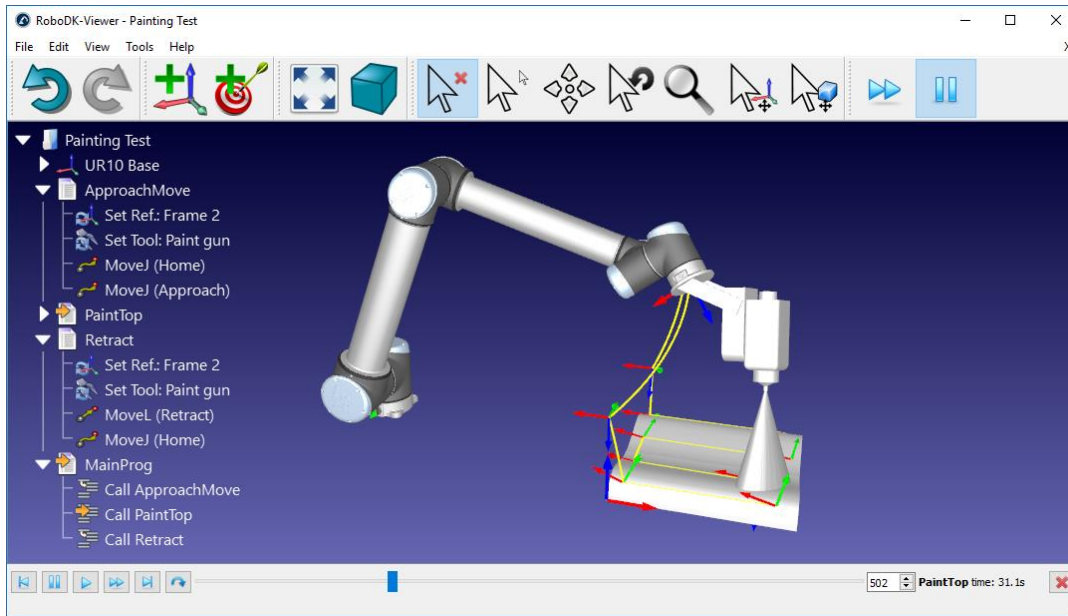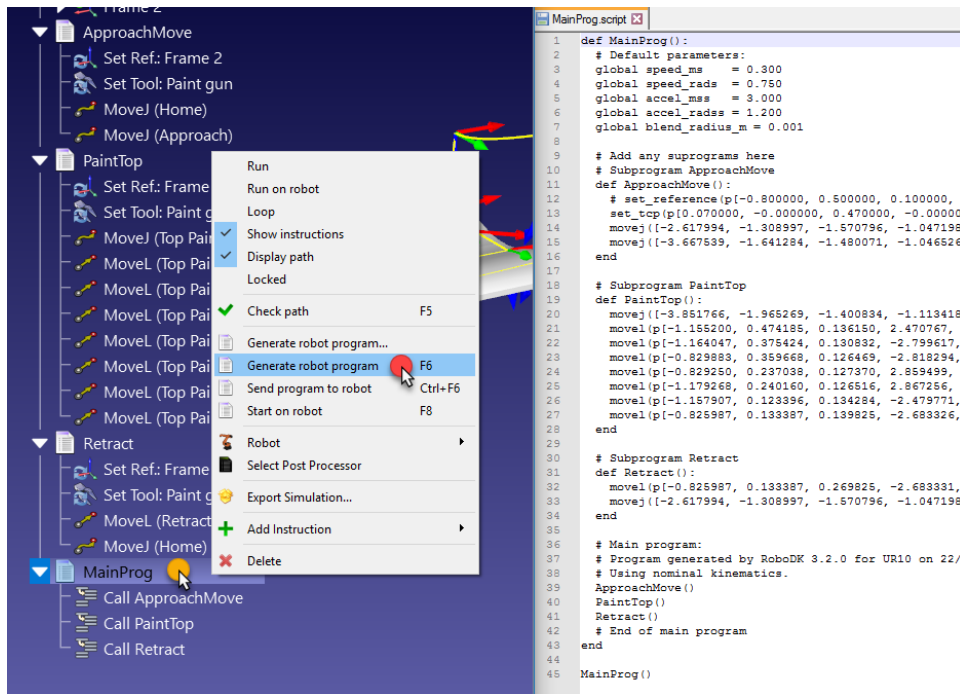
Generate Robot Program

It is possible to export any program individually or the main program including the subprograms:

1. Right click a program (**MainProg** for example)
2. Select **Generate robot program (F6)**
    Alternatively, select **Generate robot program…** to specify the location to save the file
3. The SCRIPT program for UR robot will be displayed in a text editor

This file is the result of generating the program offline. The file can be sent to the robot controller to run the same movements that were simulated in RoboDK.

Tip: If we are properly connected to the robot we can also select one of the following options in the same menu:

o   Select **Send program to robot (Ctrl+F6)** to send the program through FTP (Offline Programming)
o   Select **Start on robot (F8)**: to send the program to the robot and execute it (Offline Programming)
o   Check the option **Run on robot** to run the program step by step each time we run the program (Online Programming). This allows executing the program on the robot as it is simulated at the same time. Robot drivers are required for Online Programming.

Tip: Offline Programming with RoboDK involves using a post processor that defines the syntax used to generate each program. RoboDK provides many post processors to support different brands. More than one post processors might be available for a specific robot controller.

It is possible to change the post processor for UR robots and customize the way a program is generated:

1. Right click a program or a robot
2. Select ⬛ **Select Post Processor**
3. Select **Universal Robots**

Note: Post Processors that have a black icon can be modified by selecting **Program→Add/Edit Post Processor**.

Generate the program again. In this example, the default post processor uses joint values to define each linear move and the second post processor uses cartesian coordinates to define each linear move.

Using Macros

The RoboDK API allows customizing the simulation as much as desired. By default, RoboDK installs Python and a set of sample macros that allows a higher level of customization. This includes simulating a paint gun, programming robots through Python, simulating 2D cameras, converting SVG files to robot programs, automatically setting a TCP given a standoff, simulating discrete events and much more. Other programming languages can also be used, such as C# or Matlab. For more information visit: https://robodk.com/doc/PythonAPI/intro.html

In this example, we will add an existing sample macro that will simulate the behavior of the paint gun. We will also recolor the spray volume with a transparent color (by selecting **Tools→Change color tool - Shift+T**) or load the existing model with appropriate colors (available from the local library as **paint_gun.tool** or the online library, note that the Set Tool instruction might need to be updated to link to the new tool).

1. Select **File→** 🖼 **Open** to open a new Python macro (py file)
2. Navigate to C:/RoboDK/Library/Macros/ to see some sample macros
3. Select **SprayOn**
4. Select **Open.** A new Python object 🐍 will be added. This macro allows simulating particle deposition modeling the spray volume.
5. Double click the **SprayOn** macro to test it

6. Select **On** to activate it
7. Hold Alt key, drag the robot flange and move the robot along the surface with the Paint gun.
You should see the trace of the paint gun. The color and transparency should change depending on how close or far the TCP is from the surface.
Select Esc once to clear the simulated paint.
8. Double click the same **SprayOn** program and select **Off** to turn the particle simulation Off.
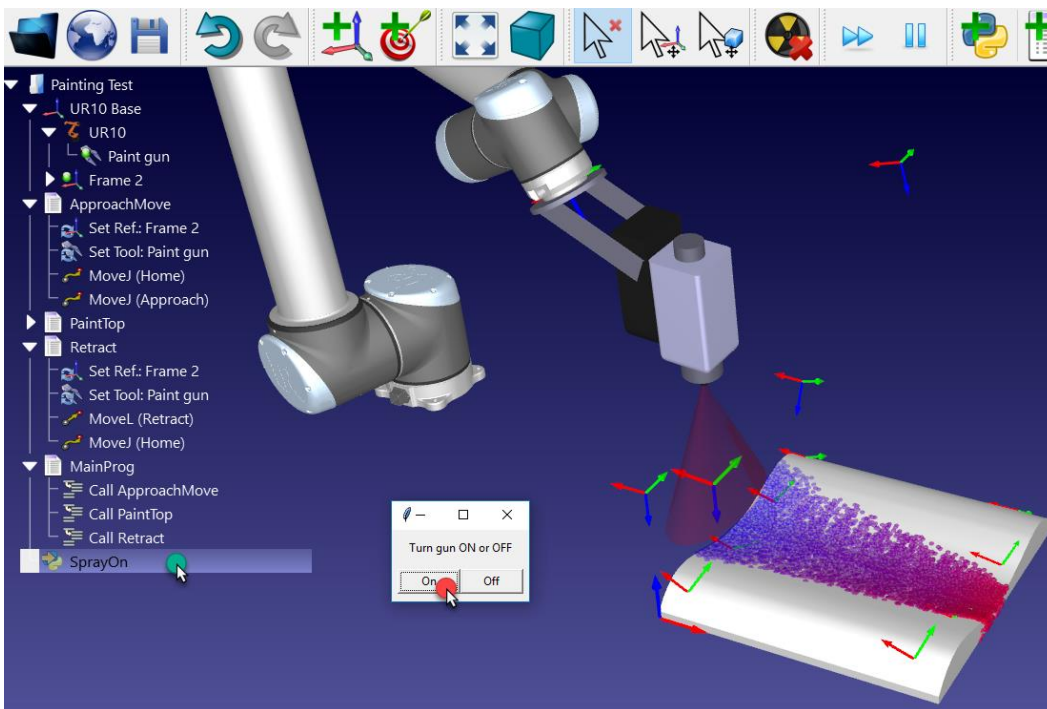

To better understand what happens behind the scenes, it is possible to view or edit the Python code the following way:

1. Right click **SprayOn**
2. Select **Edit Python Script**

A new window (text editor) will appear showing the code that models the spray behavior and how Python is integrated with RoboDK.

To take the spray simulation into account in the main program we can follow these steps:

1. Select the instruction **Call ApproachMove**
2. Select **Program→ Program call Instruction**, a new instruction will be added after the first program call and a new window will pop up
3. Enter **SprayOn(1)**
4. Select OK
5. Repeat the same operation after the **PaintTop** program setting **SprayOn(0)**, as shown in the following image.
If necessary, reorder the instructions by drag & dropping them within the program.
6. Run the **MainProg** program. After two iterations, the result should look like as shown in the image (simulating at normal speed).
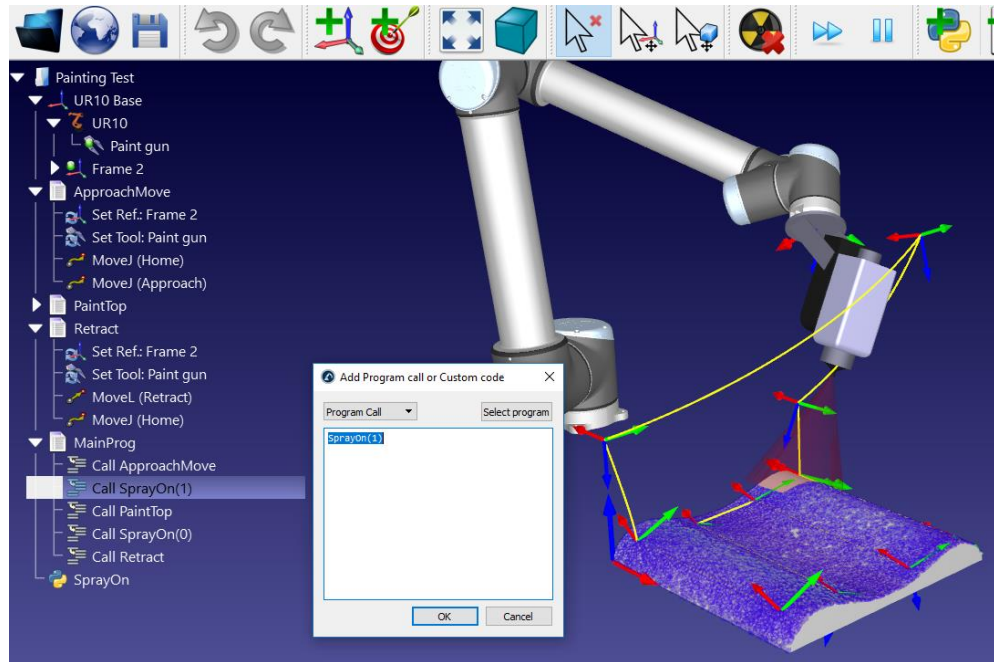
It is also possible to add new macros:

1. Select **Program→** 🐍 **Add Python Program**
2. Right click the new program and select **Edit Python Script**

Next: RoboDK supports setting the robot speed within the program, setting digital outputs, waiting for digital inputs, displaying messages, etc. These instructions are available under the **Program** menu.