Practice Project 1: Pick and Place

A simple pick and place operation in RoboDK involves programming a robot to pick up an object from one location and place it in another using a simulation environment. RoboDK provides an intuitive platform to model, simulate, and generate robot programs without needing direct access to the physical robot. In this operation, users typically define the robot, tool (end effector), and workpiece within the virtual workspace, then create targets for the "pick" and "place" positions. By linking these targets in a sequence, RoboDK allows visualization and verification of the robot's motion paths, ensuring collision-free and efficient operation. This forms the foundation for more advanced automation tasks such as sorting, assembly, or packaging.

Setting the environment

To perform the basic pick and place operation, we will set up the environment with these components.

- ABB CRB 1300-7/1.4 manipulator
- Cube 50mm
- Table 2000x1200x800mm
- End effectors: RobotiQ 2F-85 Gripper (Closed) and RobotiQ 2F-85 Gripper (Open)

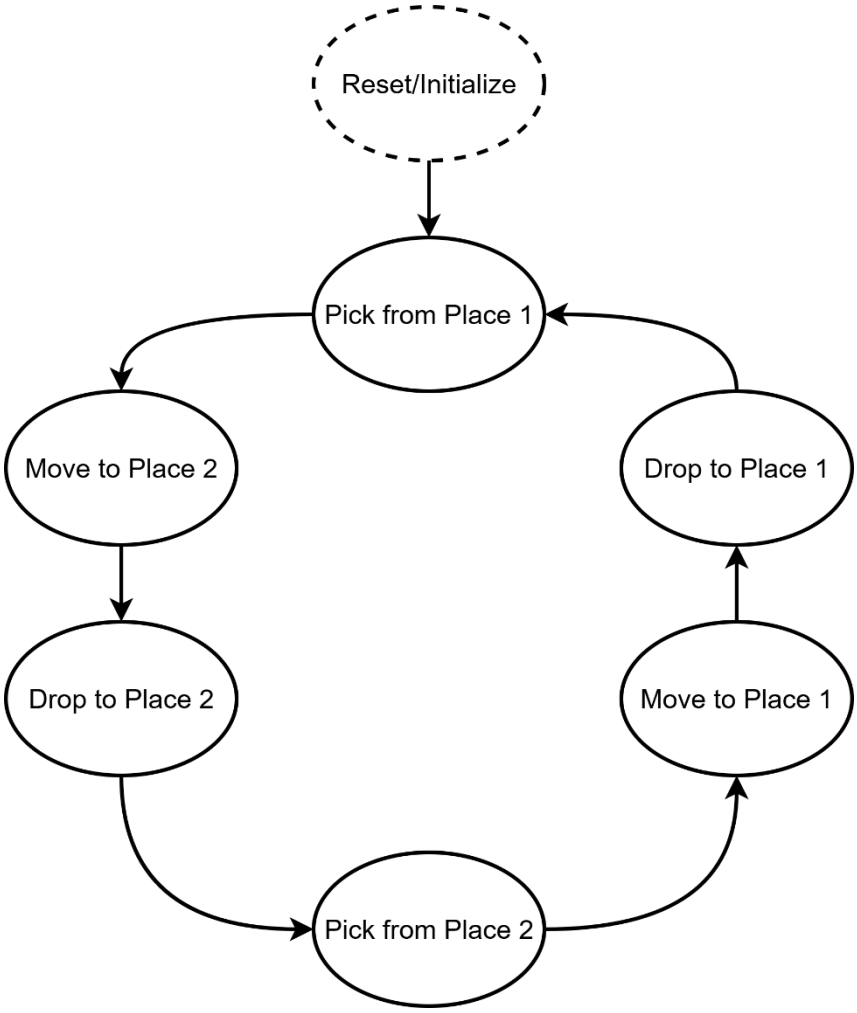
Note

Both the grippers can be imported to the scene, whereas the visibility of the gripper in closed configuration is turned off.

Once all the components are imported to the workstation, we have to position them according to our desire.

Defining the Task

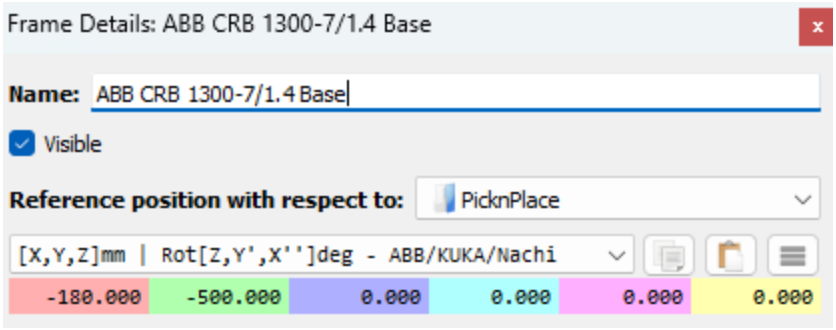
The task is to perform a simple pick and place operation. It is considered a circular operation, so a pick point becomes a drop point in a repeated cycle. The task is to pick an object from place 1, make a transition to place 2, drop the object there and redo this process by considering place 2 as a pickup point. To reset all the items to their home position, a reset function is also introduced.



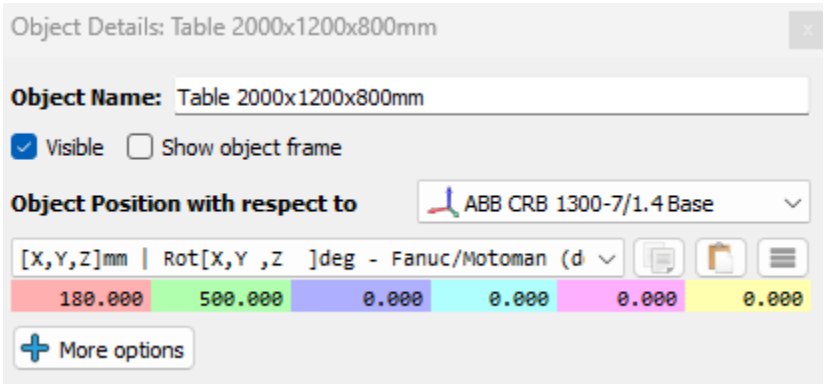
Defining Coordinate Frames and Association

The first action to perform is to attach the gripper in open configuration to the robot manipulator. For this purpose, simply drag it and make a child of the manipulator. This will make sure that the gripper is introduced in a proper configuration meaning that its z-axis should be pointing outwards. Place the manipulator on top of the table in such a manner that the object of interest will have sufficient space to be moved around.

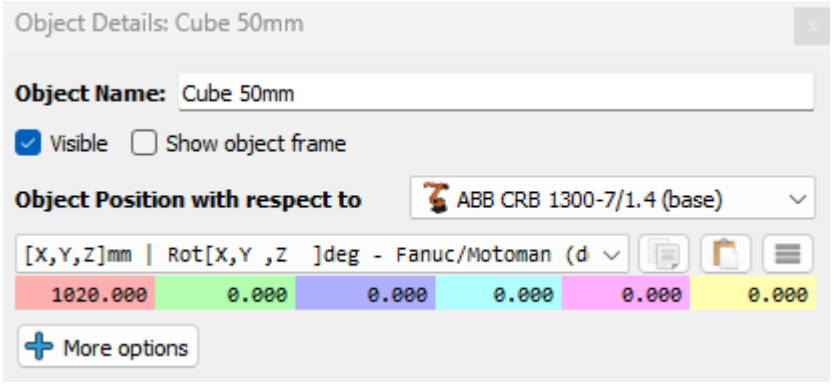
In our case, robot's base was shifted to



There is a virtual frame attached to the table and that reference frame is positioned with respect to base reference frame at



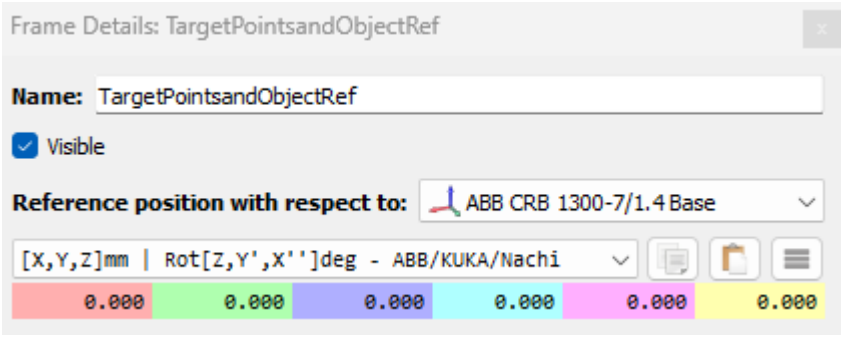
Introduce the box in the scene and place it on the top of table, in front of the manipulator. This should be considered as its first pickup point.



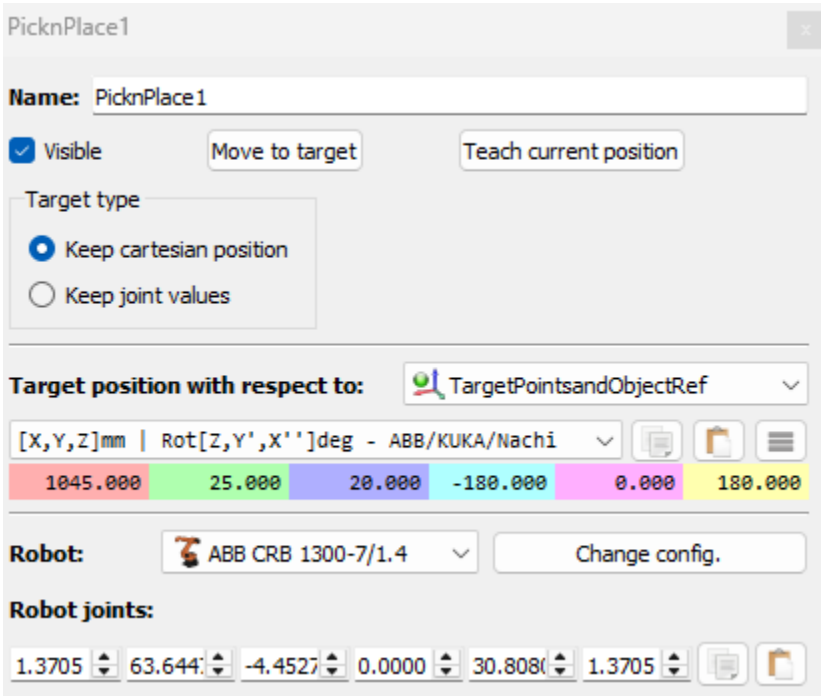
Define Target Points

In total four target points are defined to perform this operation: pick, place, approach/retract from the pick point, and approach/retract from the place point. In addition, a home target point can be defined to reset the robot to its home position, renamed as ‘Home’.

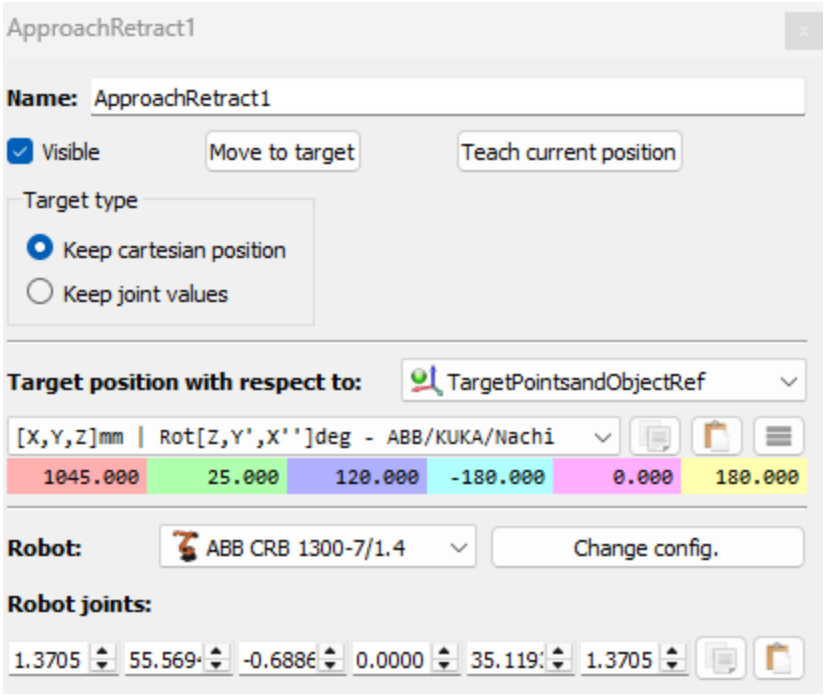
The target points are defined with respect to the local reference frame; this approach is particularly helpful when the reset operation is performed. A reference frame ‘TargetPointsandObjectRef’ is defined at the base of the robot.



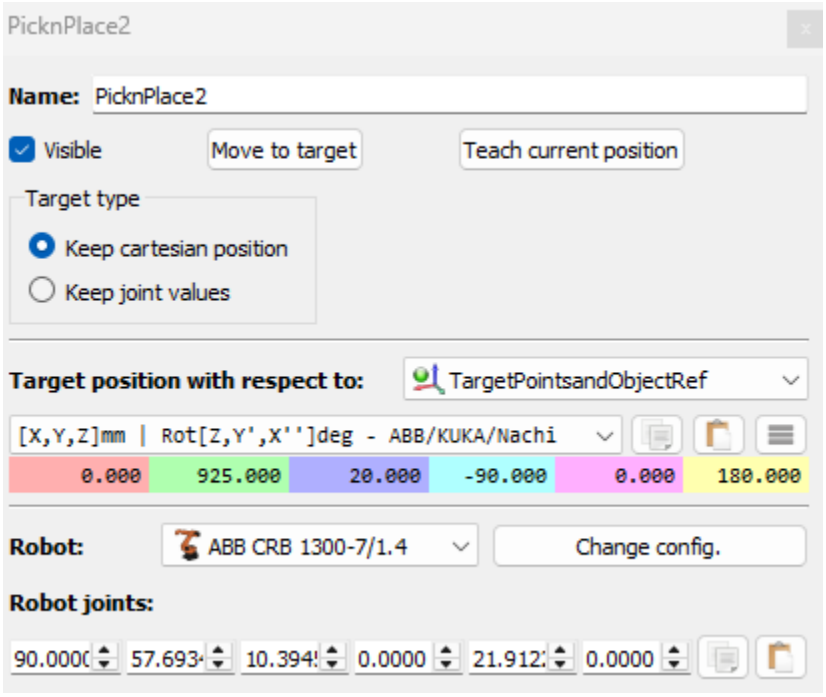
The first pick/place target is defined at the initial location of the cube and rename it as ‘PicknPlace1’.



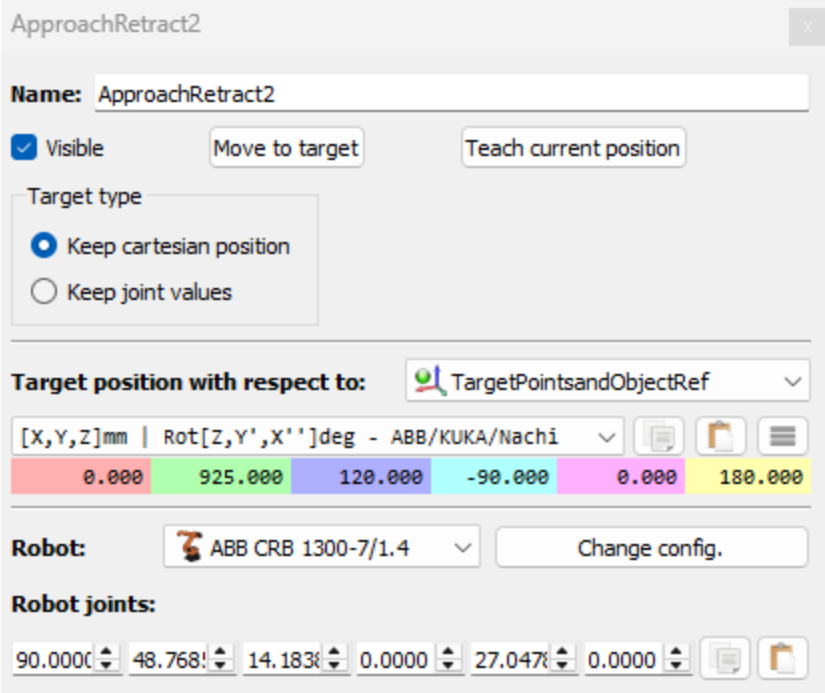
The first approach/retract target is introduced just above the initial location of the cube.



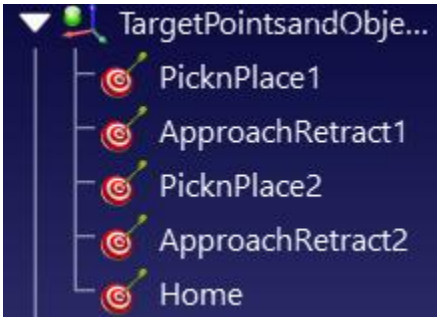
The second pick/place target is renamed as ‘PicknPlace1’ and is defined as



As previously, the approach/retract point for the second point is placed at some offset along z-axis.



The final collection of all target points will look like



Program the robot

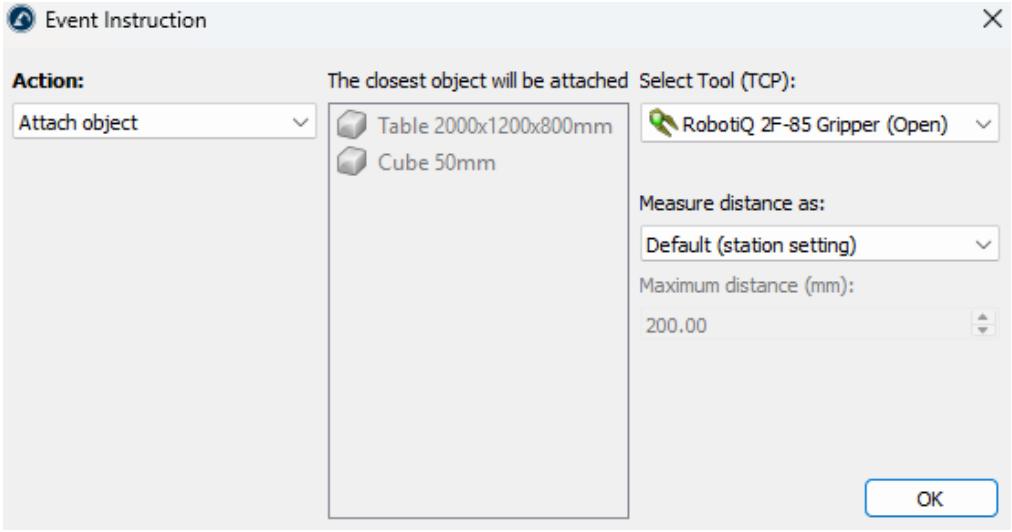
In addition to reset/initialize program, six sub-programs will be developed with two routines, each to perform a sub-task as listed earlier.

Routines

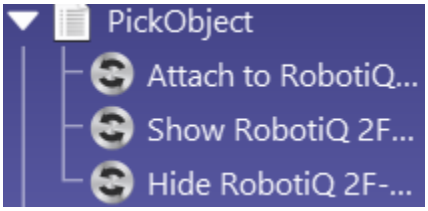
Three routines are defined which will be used by the sub-programs with the purpose of attach/detach the object, reset the space, and switch between both end-effectors.

1. Pick Object

The first task is to attach the desired object i.e. the cube to the gripper.

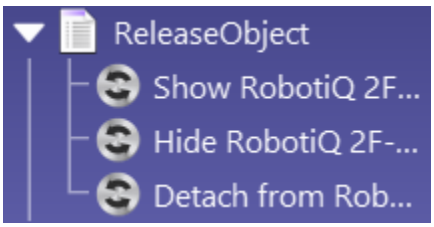


Secondly, turn the visibility on for the closed configuration of the gripper and off for the open configuration.



2. Release Object

The second routine is exactly the opposite to the first one in which the object will be detached from the manipulator, open configuration of the end-effector is enabled, and closed configuration is disabled.

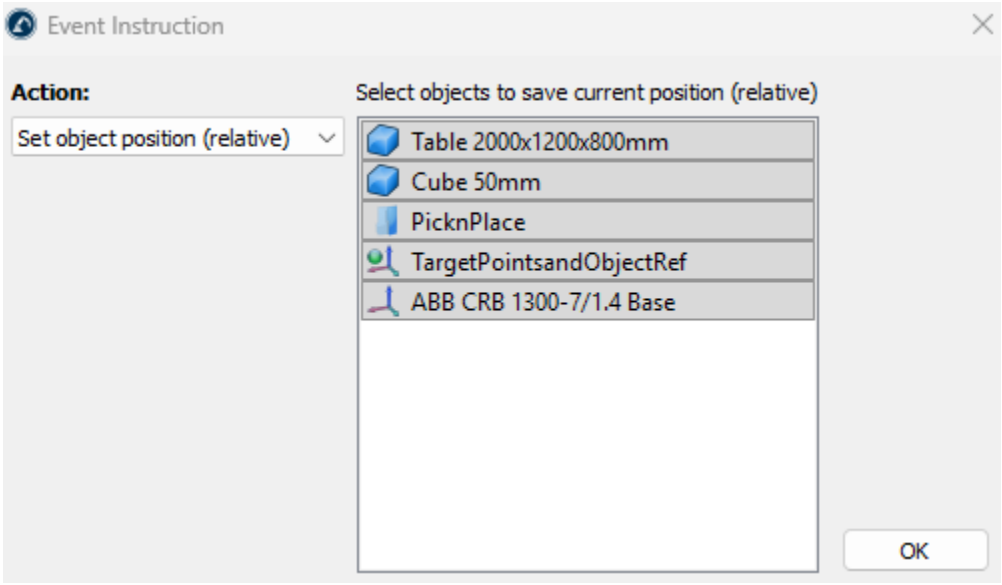


3. Reset

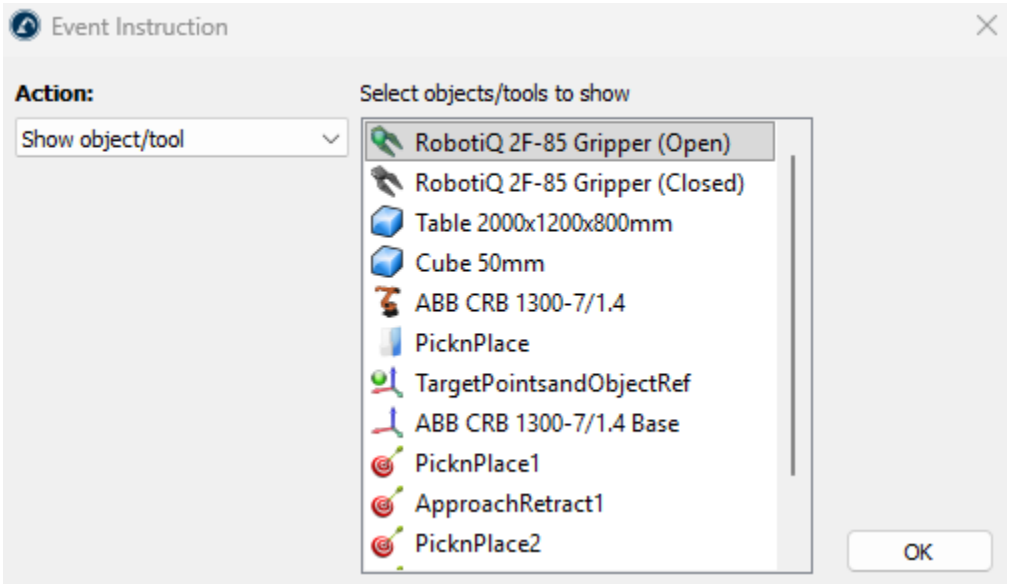
Add a program named ‘Reset’ and introduce an event instruction to reset the whole workspace. This will make sure that if some objects are misplaced unintentionally, they should be replaced at their proper place.

Add Instruction → Simulation Event Instruction

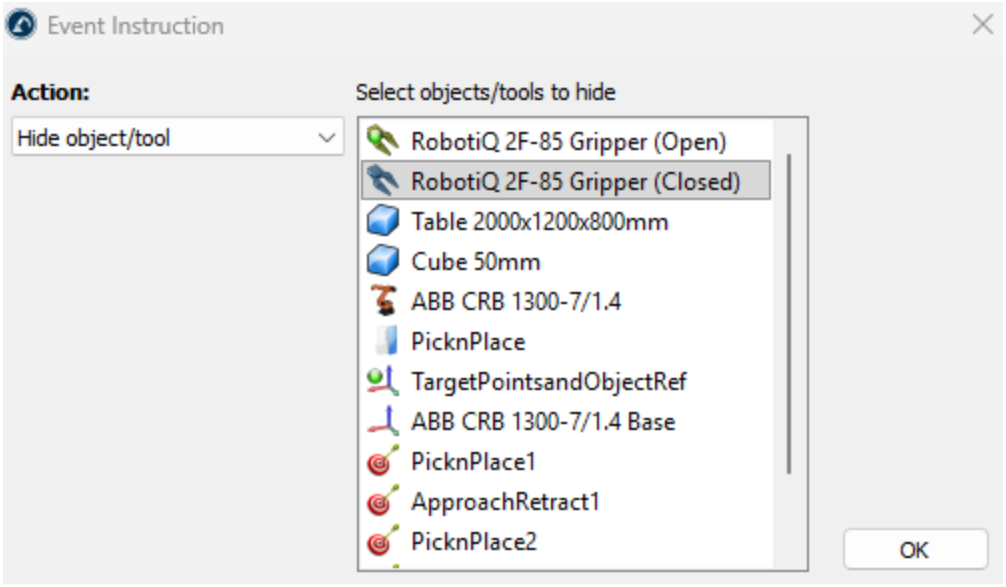
These event instructions are only meant to be performed in the simulation environment.



Two more event instructions are introduced in the same manner for the end-effector, one to show one type of end-effector and turn off the visibility of the other one. In the reset state, open configuration is the active state; therefore, its visibility is turned on.



For the reset state, visibility of the closed gripper is turned off.

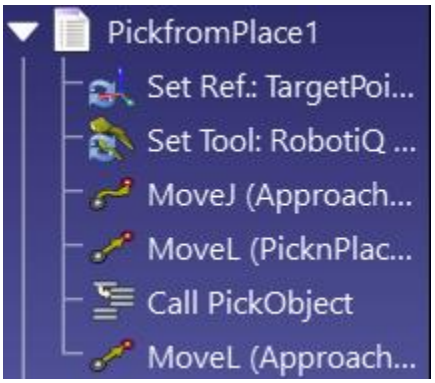


The last instruction is to ensure that the robot should start its operation from the home position. For this purpose, MoveJ instruction is utilized.

Sub-programs

1. Pick from Place 1

The operation sequence is defined as to reach the 'ApproachRetract1' location with the move instruction. Once this pose is attained, the manipulator performs the linear movement to approach the object through 'PicknPlace1'. Then, the 'PickObject' routine is called to perform the attachment and define the proper configuration of the end-effector. After grabbing the object with the end-effector, the manipulator returns to the retract location linearly.



2. Move to Place 2

This is considered as the transition state in which the manipulator moves between pick and drop point. In this case, the manipulator traverse to 'ApproachRetract2' using the move instruction.

3. Drop to Place 2

As manipulator is already coinciding with the drop point's reference frame, only displacement along z-axis is involved which can be achieved through move linear instruction. The robot now moves to 'PicknPlace2' location. Once the manipulator has achieved the desired pose; the drop sequencing is initiated with the help of 'ReleaseObject' routine. Lastly, the manipulator retracts to the 'ApproachRetract2' location.

4. Pick from Place 2

From its current pose, the manipulator reaches the desired object and attains the pose defined by ‘PicknPlace2’ through linear movement. Subsequently, the pickup operation is initiated by calling the ‘PickObject’ routine. Lastly, the manipulator retracts to the ‘ApproachRetract2’ location.

5. Move to Place 1

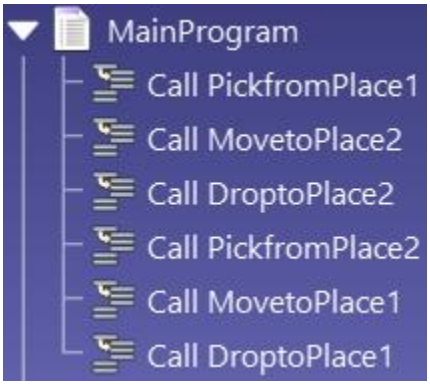
With the object in hold, the manipulator traverses to the ‘ApproachRetract1’ location using the move instruction.

6. Drop to Place 1

The last sequence of operation is the drop operation at ‘PicknPlace1’ followed by the retract operation and in between, the ‘ReleaseObject’ routine is called.

Main Program

The last task is to define the main program so that all the sub-programs can be called sequentially. You can select all desired sub-programs which you want to be a part of the main program, right click and click ‘Make main program’. By default, a new program with the name ‘MainProgram’ is created which is a collection of all the sub-programs that we defined earlier. These programs run in a sequential manner, but you are free to change the sequence.



The final configuration of the whole system appears like this