



baxterTM

user guide

for intera 3.3 software

Welcome!

Thank you for purchasing Baxter, the world's first collaborative robot for manufacturing.

This user guide is designed to provide you with an overview of the robot's features, help support you through the setup process, instruct you on training Baxter for tasks, and outline some basic troubleshooting measures should you need them.

The latest and most detailed information is available on our Intera MFG Wiki page at:
mfg.rethinkrobotics.com

Safety Statement

Complying with ISO 10218-2 requires performing a risk assessment of each application to determine the needed safety performance and safeguarding. ANSI RIA R15.06-2012 is a U.S.-national adoption of ISO 10218-1 & 2.

Users should exercise caution while training Baxter and practicing its motions. The risk of injury is increased when using custom end-effectors, off-vertical motions, and potentially hazardous work pieces.

Rethink Robotics recommends the use of safety glasses when interacting with Baxter, as with other equipment used in industrial environments.

For additional information, reference Rethink Robotics' Safety Documentation: <http://www.rethinkrobotics.com/resources/safety/>

Disclaimer

Every effort is made to ensure that the information in this manual is accurate. This publication could include technical or typographical errors or other inaccuracies. Rethink Robotics, Inc® may make changes to the product described in this publication or to this publication at any time, without notice.

A newer version of this document may be available here:

mfg.rethinkrobotics.com/wiki/Support_Resources



Getting Started with Baxter 5

Setting Up Baxter	5
End of Arm Tooling	5
Grippers - Configuring	6
<i>How to Configure a Vacuum Gripper</i>	6
<i>How to Configure an Electric Parallel Gripper</i>	9
<i>Configuring a Custom Gripper</i>	10
<i>Restore Gripper Configuration</i>	11

Getting to Know Baxter 12

Hardware Overview of Your Baxter Robot	12
<i>Back View</i>	13
Top Down Work Envelope	14
Baxter Dimensions	15
Accessories	16
<i>Grippers</i>	16
Turning On Baxter	19
How to Interact with Baxter	19
<i>Using the Training Cuffs</i>	19
<i>Navigating the Screens</i>	21
<i>Moving the Arms</i>	22
<i>Grasping Objects</i>	23
<i>Active Collision Avoidance</i>	24
How Baxter Interacts	25
<i>Eye Expressions</i>	25
<i>Attention Ring</i>	26
<i>Condition Ring</i>	27
<i>Confused Face Messages</i>	28
<i>"Light Bulb" Tips</i>	28

Quick Start Baxter Tutorial 29

Performing a Task in Just a Few Minutes	29
<i>Turning On Baxter</i>	29
1. Move the Arm	30
Create a New Task	32
Run the Task	36



Getting to Know Intera 37

- What is Intera? 37
- The Basics: What is a Task? 37
- The Basics: What is an Action? 38
 - Labeling Actions* 39
- The Basics: What Is a Subtask? 40
- Navigation: Main Screen 41
- Navigation: Task Map 42
- Navigation: Task Gallery 45
- Navigation: Settings 50
 - About Baxter* 50
 - Hardware Settings* 50
 - Advanced* 51
 - Lock/Unlock* 51

Training a Task In Intera 53

- Training a Pick and Place 53
 - Renaming Tasks* 56
- Training a Hold Action 58
 - How to Train Baxter to Perform a Hold* 59
 - Reordering a Hold Action and Adding a Hold time* 59
- Training a Group of Pick or Place Actions 60
 - How to Train an Action Group* 61
- Creating Paths: Paths, Waypoints, Poses 62
 - How to Create a Custom Path for Baxter* 63
 - Best Practices: Creating Waypoints* 65
 - How to Practice a Path* 65
 - How to Modify a Waypoint* 67
 - How to Modify a Path (includes Practice and Retrain)* 69
- Modifying Actions in Intera 77
 - Modifying a Pick Action* 77
 - Modifying a Place Action* 78
 - Modifying Count* 81
 - How to Modify/Practice an Action* 81
 - Nudge* 83
 - How to Make Detailed Modifications to an Action (Advanced Settings Screen)* 89
- Managing Tasks and Subtasks 94
 - Task Button Bar* 95
 - Subtask Button Bar* 96
 - Coordinating Sub-tasks Across Arms* 97



<i>Changing Gripper Parameters</i>	99
<i>Camera Settings</i>	102
<i>Change Number of Attempts to Pick an Object</i>	103
Motion Presets	105
<i>How to Change Motion Presets at the Task Level</i>	105
<i>How to Change the Motion Preset at the Action Level</i>	106
Specify Object and Custom Gripper Weights	107
<i>How to Specify Object Weight</i>	108

Robot Positioning System **110**

Introduction	110
When to Use the Robot Positioning System	111
How to Create a Landmark	112
<i>Alternate Way to Create a Landmark</i>	116
How to Associate a Landmark to an Action	118
How to Reregister a Landmark	119
Robot Positioning System - Best Practices	123
Frequently Asked Questions about Robot Positioning System	124
Application Examples	125
<i>Unloading Parts from a Rotary Table and Onto Conveyors</i>	125
<i>Transferring Parts from One Conveyor to Another</i>	126
<i>Putting parts into fixtures that require precision</i>	126

Signals **128**

Working with Signals	128
<i>Internal Signals</i>	130
<i>Creating Signals</i>	133
<i>Assigning Signals</i>	135
Signals and their Definitions	138
<i>Terminology</i>	138
<i>Signal Types</i>	139
<i>Task Element Details</i>	141
<i>Signals</i>	142
<i>TASK</i>	142
<i>SUBTASK</i>	144
<i>GRIPPING ACTIONS: PICK, PLACE</i>	144
<i>OTHER ACTIONS: HOLD</i>	145
<i>FAQs - Signals and Their Definitions</i>	146



Maintaining and Supporting Baxter 150

- Powering Down Baxter 150
- Maintaining Baxter 150
 - Cleaning Baxter 150*
 - Replacing the Air Filters 150*
 - Upgrading Software 151*
- Importing and Exporting Tasks 151
 - How to Export Tasks from a Robot 152*
 - How to Import Tasks to a Robot 153*
- Supporting Baxter 154
 - Helping Baxter When It Is Confused 154*
 - Calibrating the Arms 155*
 - Troubleshooting Baxter 157*
 - Exporting Log Files 158*
- Enjoy! 158

Appendix A: Glossary 159

- Glossary 159

Appendix B: Configuring External Devices 160

- Connecting a Modbus Remote Terminal Unit (RTU) to Baxter 162
- Additional Information About the Modbus RTU 164

Appendix C: Support & Warranty 165



Getting Started with Baxter

Setting Up Baxter

To prepare for the arrival and setup of Baxter, read the *Baxter Pre-Delivery Guide*. The document arrives in an email prior to the delivery of your order.

To set up Baxter:

- Locate the installation card that ships with the robot. Follow the instructions to assemble the pedestal (if ordered), and attach Baxter to the pedestal or alternate work surface.
- Follow the instructions included with the gripper kit to install a gripper.

If you misplace these documents, refer to our wiki:

mfg.rethinkrobotics.com/wiki/Support_Resources

End of Arm Tooling

“End of arm tooling” is the term for what is attached to the end of the robot’s arm that allows the robot to interact with parts in its environment. Rethink Robotics offers two different types of End of Arm Tooling (specifically grippers): an electric parallel gripper and a vacuum gripper. You can also attach custom or third party End of Arm Tooling to the robot.

IMPORTANT: When attaching any end of arm tooling to the robot, it is critical that you accurately configure its length and weight. That will maximize the performance and repeatability of the robot. Please follow the steps below to configure your chosen End of Arm Tooling.



Grippers - Configuring

When you attach a Rethink gripper, Baxter automatically detects the gripper type, and prompts you to configure it. Follow the onscreen instructions and enter the length and weight of the gripper.

Configuration teaches Baxter:

- Finger type and open/closed finger position of a parallel gripper
- Vacuum cup size of a vacuum gripper

You can also attach custom or third party grippers the same as Rethink grippers, essentially by entering the gripper's length and weight and then training Baxter. See "Configuring a Custom Gripper" on page 10.

If you change any component of a gripper (such as adding extensions to a vacuum cup) without detaching it, go to **Settings > Hardware Settings > Configure End Effectors** to reconfigure it. If you do not do this, Baxter may have trouble picking up or holding on to objects.

How to Configure a Vacuum Gripper

Using the navigator on the arm with the gripper you want to configure:

1. If the gripper is holding an object, release the object.
2. If not already open, scroll to **Main Screen > Settings > Hardware Settings > Configure End Effectors**.



3. If you need to install or adjust the gripper, depending on where you are standing, select **Behind** or **Front**. The robot automatically presents its hand to you, either in front, or off to the side in back.

Note: You can also manually reposition the arm to access the gripper.

4. Select **Length**, then enter the length of the gripper based on the graphic below, which illustrates where to measure.

Note: If you're using your own adapter tool plate, be sure to take into account its size and weight when configuring.



Gripper Length

Enter Length mm

5. Click **Done**, then select the vacuum cup's **Weight** icon.

Gripper Weight

Enter Weight kg

6. Enter the weight of the gripper and click **Done**.



7. Click **Train**.



The gripper rotates as the robot learns it.

How to Configure an Electric Parallel Gripper

You configure electric parallel grippers the same as you do vacuum grippers. Just enter the gripper's length and weight.

1. If not already open, scroll to **Main Screen > Settings > Hardware Settings > Configure End Effectors**.
2. If you need to install or adjust the gripper, depending on where you are standing, click **Behind** or **Front**. The robot automatically presents its hand to you, either in front, or off to the side in back. You can also move the arm manually to access the gripper.



Here are the weights and lengths for each component in the Electric Parallel gripper kits:

Component	Weight (Kilograms)	Length (Millimeters)
Electric Parallel Gripper without fingers	0.3 Kg	43 mm
Wide Short Finger	0.02 Kg	73 mm
Narrow Short Finger	0.01 Kg	73 mm
Wide Long Finger	0.02 Kg	112 mm
Narrow Long Finger	0.02 Kg	112 mm

Note: If you're using your own adapter tool plate, be sure to take into account its size and weight when configuring.

3. Select **Length**, and enter the length of the gripper based on the graphic, which illustrates where to measure.

Note: The screens for entering the length and weight of Rethink electric parallel grippers are the same as for Rethink vacuum grippers.

4. Click **Done**, then select the gripper's **Weight** icon.
5. Enter the weight of the gripper and click **Done**.
6. Click **Train** to finish configuring the gripper.

Configuring a Custom Gripper

You can also attach a custom gripper to Baxter, for example, a pneumatic gripper actuated using external I/O. You must still configure and train this gripper using the same screens used to configure and train Rethink grippers.

1. Scroll to **Main Screen > Settings > Hardware Settings > Configure End Effectors**.
2. Install the gripper on the end of the robot's arm. Depending on where you are standing, click **Behind** or **Front**. The robot automatically presents its hand to you, either in front, or off to the side in back.

Note: You can also manually reposition the arm to access the gripper.

3. Select **Length**, then enter the length of the gripper. (The graphic illustrates where to measure.)
4. Click **Done**, then select the gripper's **Weight** icon.
5. Enter the weight of the gripper and click **Done**.
6. Click **Train** to finish configuring the gripper.



Note: If you're using a non-actuating gripper, make sure you adjust the weight and length appropriately in order to improve usability and performance.

Restore Gripper Configuration

If you remove a Rethink gripper and then reattach it, you do not have to go through the process of reconfiguring it. This feature allows you to reattach the gripper and get back to work quickly. When you reattach the gripper, Baxter will display the information for the last gripper of that type, and ask you to confirm.



***Note:** The Restore Gripper Configuration feature does not work for non-Rethink end of arm tooling.*

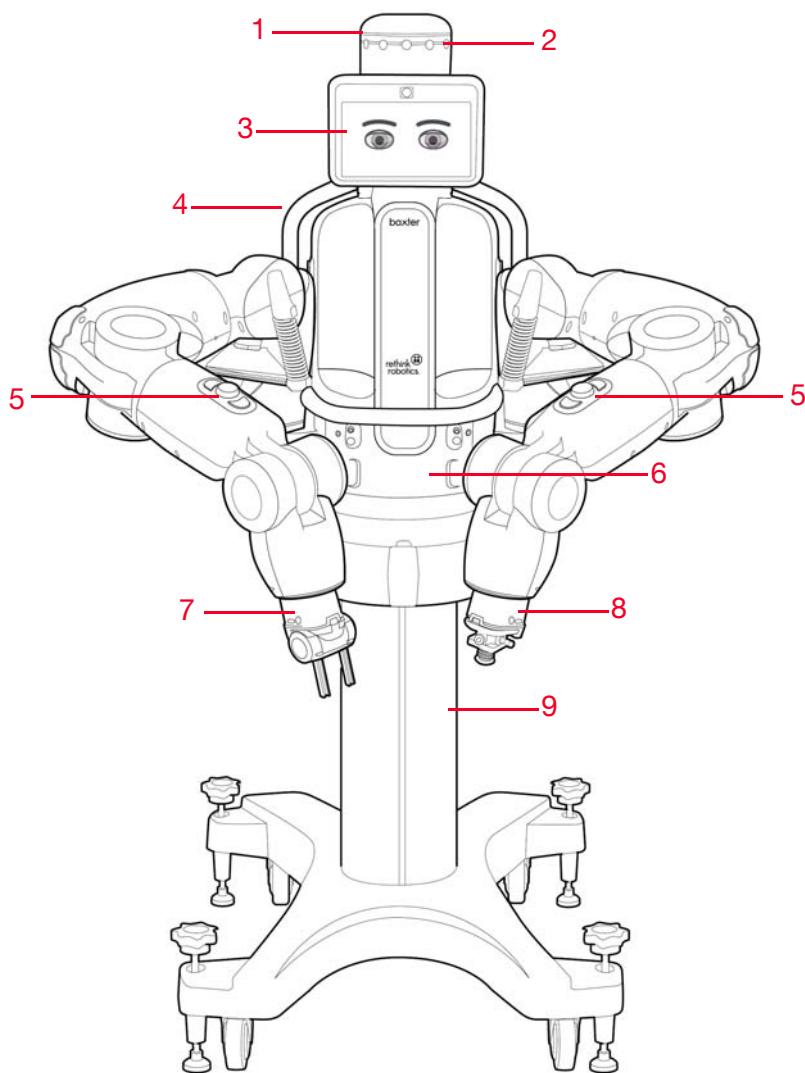
If you're installing the same gripper as before, press **Yes** to use the end effector configuration values saved on the robot. If you're installing a different gripper, select the **Configure** button. The Intera UI will direct you to the configure end effector screen where you can enter the appropriate length and weight for the gripper.

***Note:** Custom finger lengths are supported and can be entered into the software, but custom widths are not currently supported by the software.*



Getting to Know Baxter

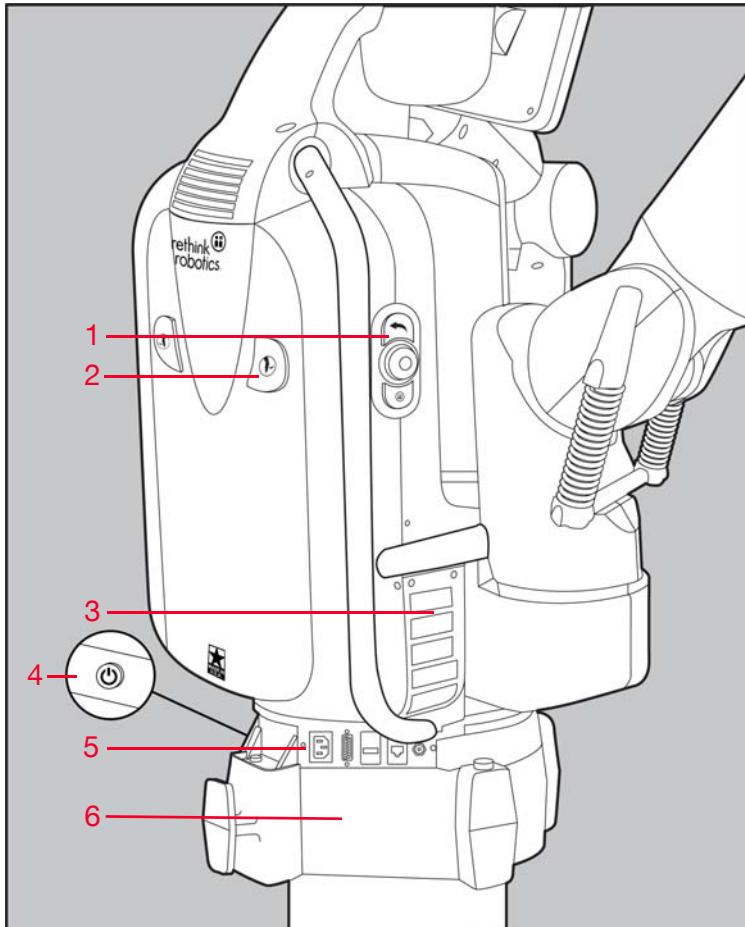
Hardware Overview of Your Baxter Robot



1. Condition ring
2. Attention ring
3. Display
4. Torso
5. Navigator (one on each forearm)
6. Lower front panel
7. Training cuff (shown with parallel gripper)
8. Training cuff (shown with vacuum gripper)
9. Pedestal (optional)



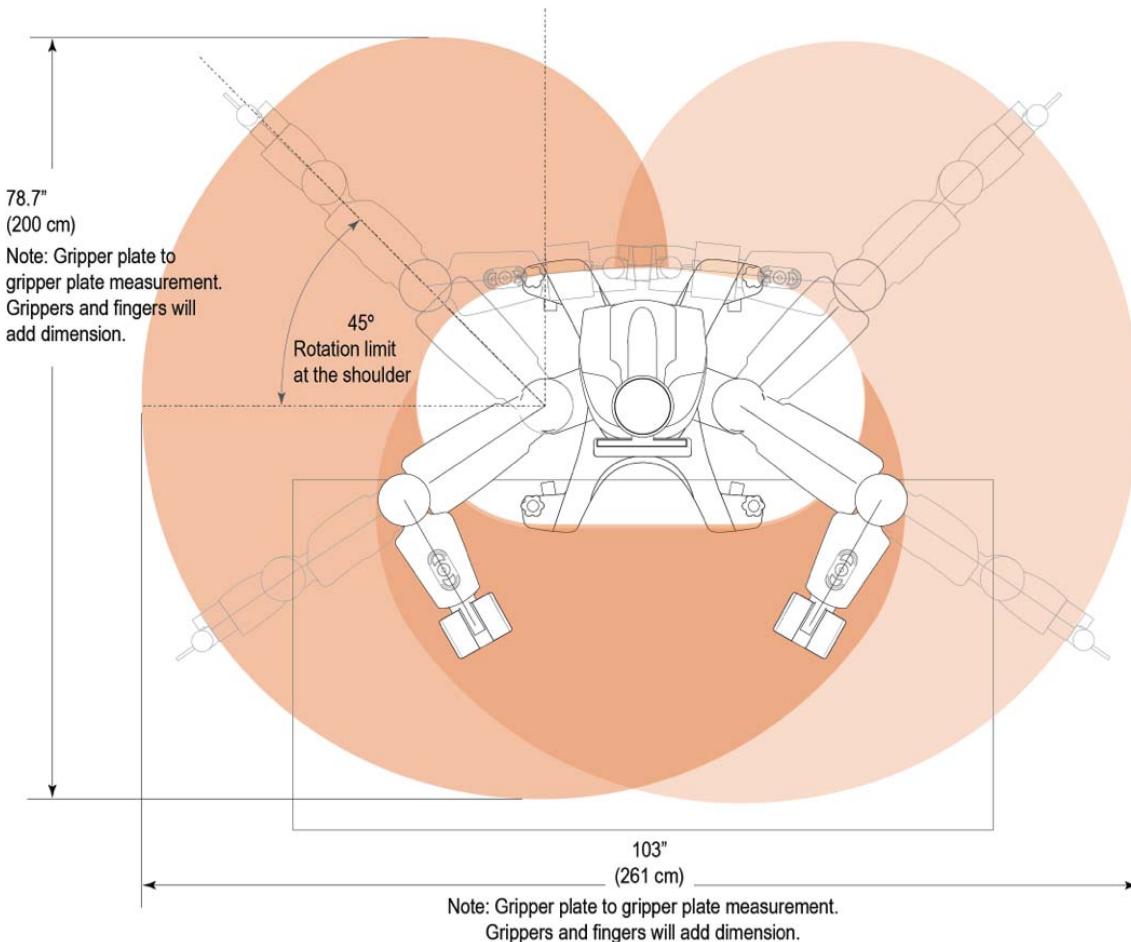
Back View



1. Navigator (one on each side)
2. Non-active (button for future use)
3. Air filter (one on each side)
4. Power button
5. Power and I/O panel (with DB15, USB, and Ethernet Ports)
6. Pedestal (optional)

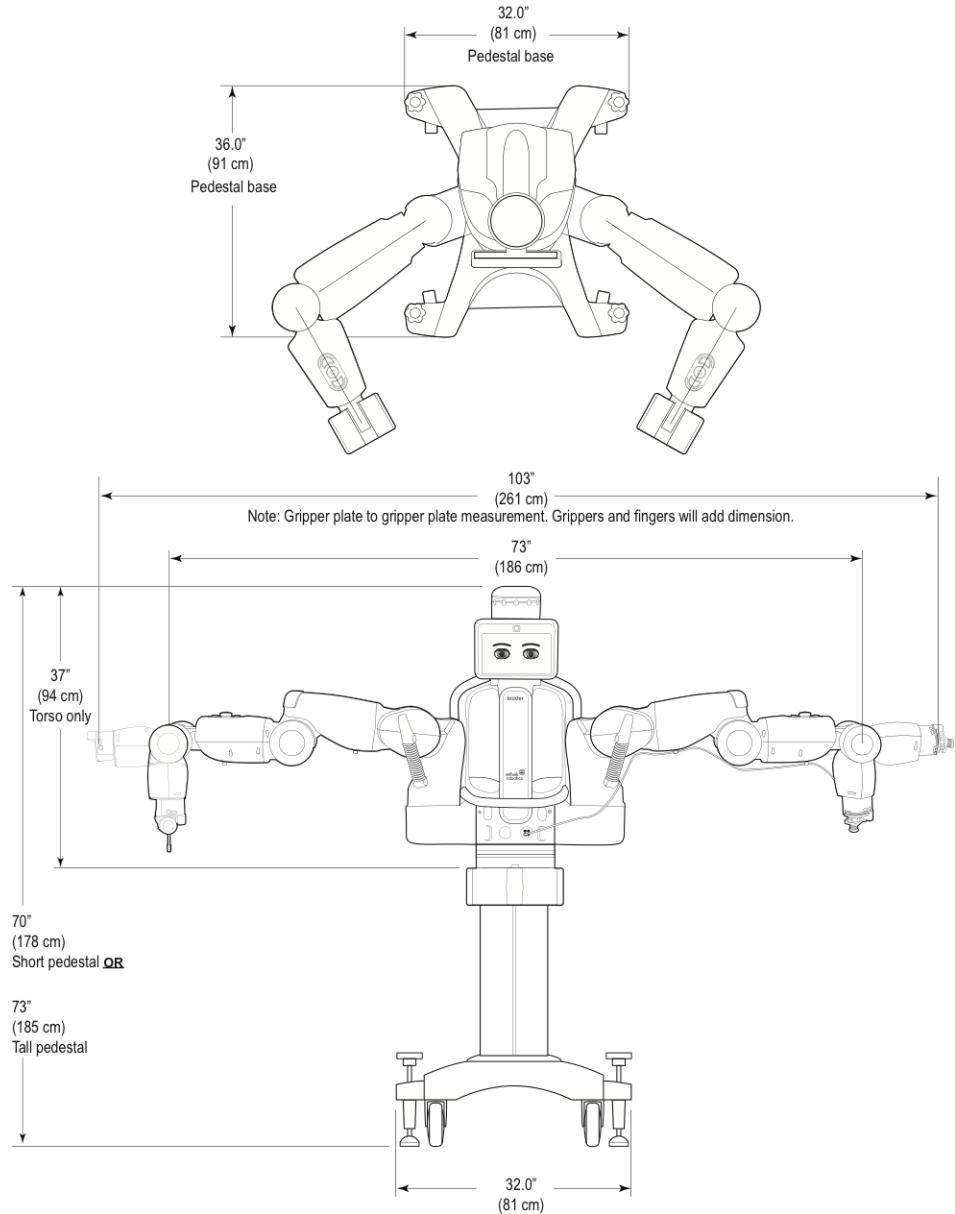


Top Down Work Envelope





Baxter Dimensions





Accessories

Included accessories:

- Power cord
- E-stop button and 10-foot cable

Optional accessories:

- Rethink Robotics Electric Parallel Gripper Kit
- Rethink Robotics Vacuum Cup Gripper Kit
- Baxter pedestal

Grippers

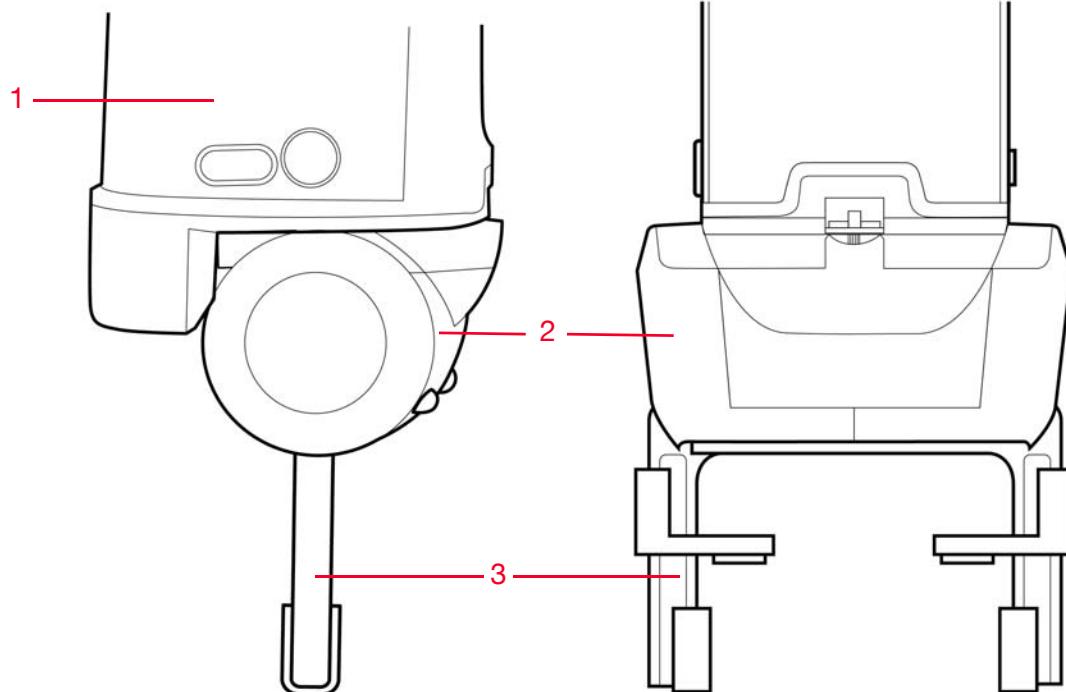
Grippers are the robot's hands—they enable Baxter to grasp and release objects. The grippers attach to the *wrist plate* at the base of the robot's training cuff. Baxter supports two standard Rethink Robotics grippers: electric parallel and vacuum cup, and can support some custom grippers. Please contact Rethink Support with questions about other types of grippers.

After installing or altering a gripper, see “Grippers - Configuring” on page 26 to configure it. (Note that installation and alteration instructions ship with the gripper kit.)

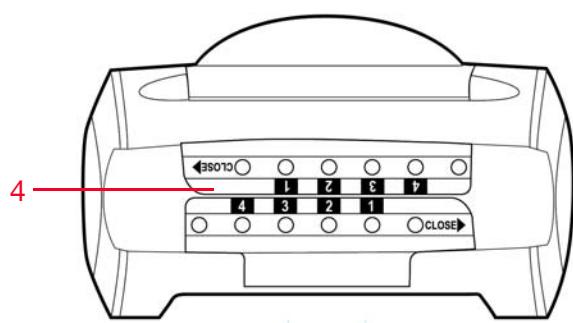
Note: Custom finger lengths are supported and can be entered into the software, but custom widths are not currently supported by the software.



ELECTRIC PARALLEL GRIPPER

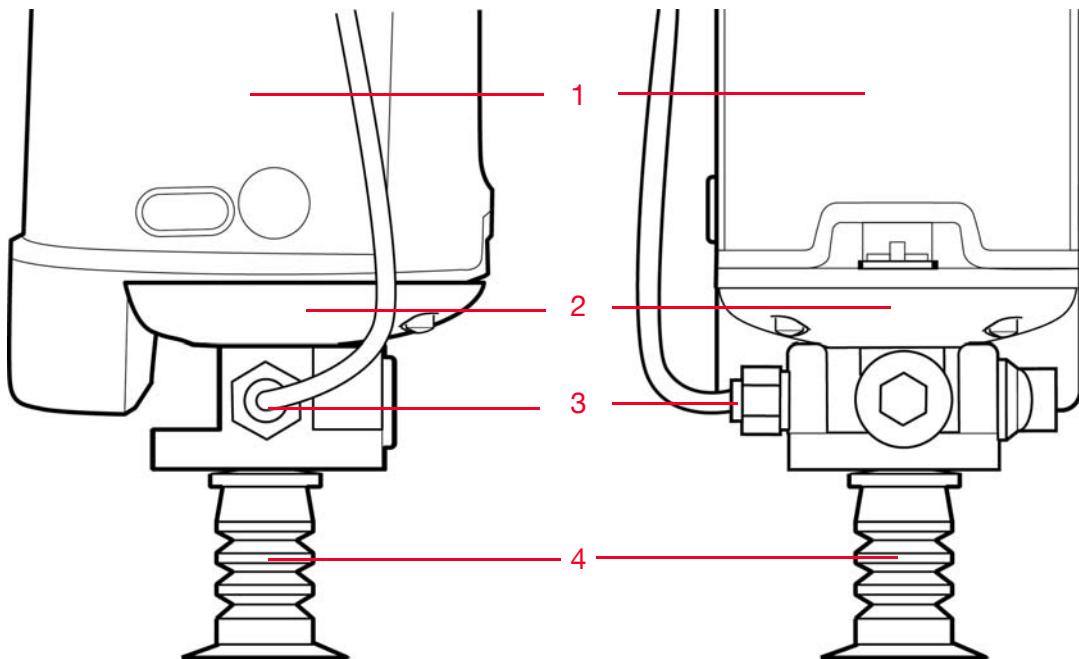


1. Training cuff
2. Gripper body
3. Fingers
4. Finger positions





VACUUM CUP GRIPPER



1. Training cuff
2. Gripper body
3. Pneumatic tube fitting
4. Vacuum cup



Turning On Baxter

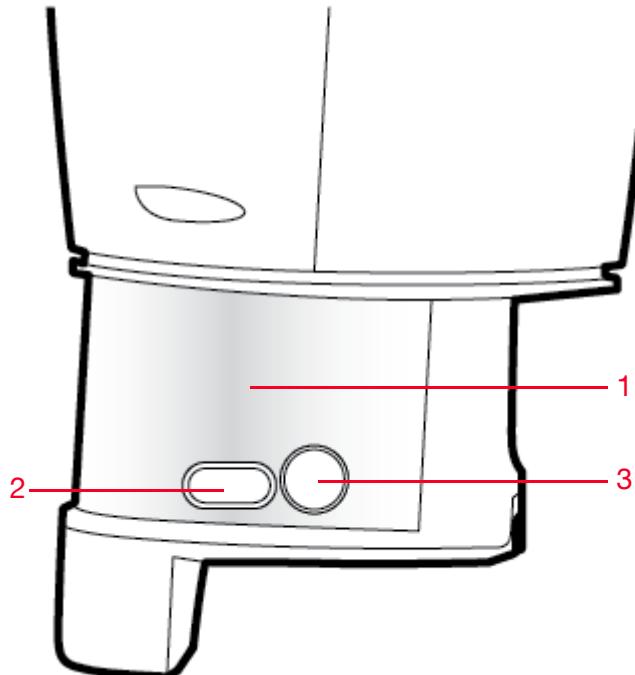
Press the white power button on the lower left back of the robot (see “Back View” on page 13 for the location). The lights on the head turn on, and the main screen appears on the Baxter display.

IMPORTANT: *Never hold the power button down* to shut down Baxter. Just press and release the power button.

How to Interact with Baxter

Using the Training Cuffs

Use the *training cuffs* to move the arms, to manipulate the state of the grippers, and secondarily, to select on-screen options.



1. Training cuff switch
2. Grasp button
3. Action button

Training cuff switch: Squeeze this switch at the indentation in the cuff to move the robot's arm. When this switch is squeezed, the blue indicator on the arm's navigator button lights up.

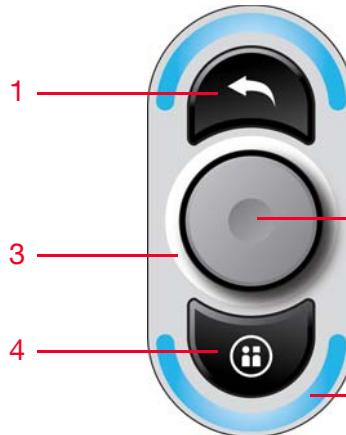
Grasp button: Press to toggle a parallel gripper open or closed, or a vacuum gripper on or off.

Action button: Press to select items on the display screen. Create waypoints, Hold actions; select, copy, or move actions on the task map; create a new subtask; add/create landmarks; outline a visual search area.



Navigating the Screens

Use the *navigator* on either of the arms to scroll to and interact with options on the screen. When you press the OK button (2) (or the action button on the cuff), the white indicators on the navigator light up.



1. Back button
2. Scroll knob (turn to scroll) and OK button (press)
3. OK indicator light
4. Rethink button
5. Training cuff indicator lights

Back button: Press to exit the current screen and return to the previous screen. Will also cancel the last action.

Knob: Scroll the knob to move between on-screen options. Press the knob (OK) to select an option.

OK indicator light: When the action button on the cuff or the OK button on the navigator is pressed, the white indicator around the knob lights up.

Rethink button: Press to display options for the current screen.

Training cuff indicator: When the switch on the cuff is squeezed, the blue indicators along the top and bottom edge of the navigator light up.

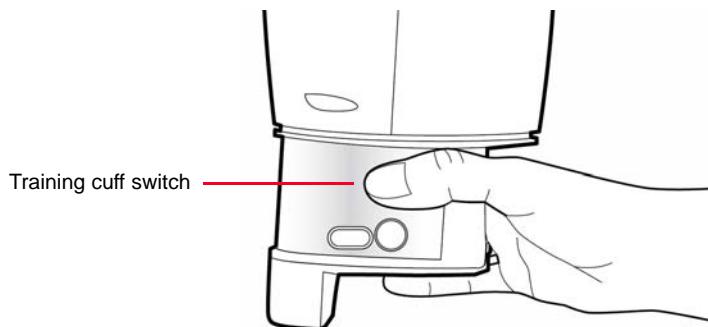


Moving the Arms

To move an arm, squeeze the cuff at the indentation just above the other buttons, and push or pull the arm to the location you want.

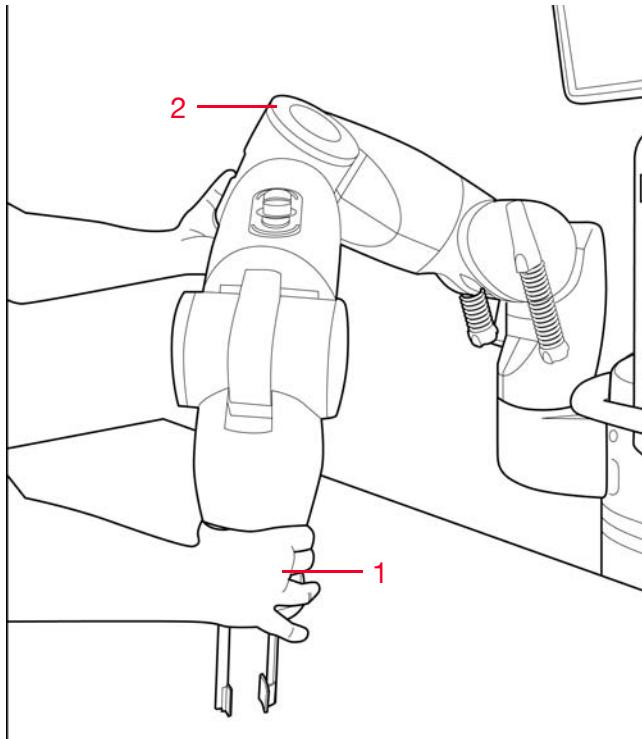
Squeezing the cuff releases the tension and resistance in the arm, making it easier to manipulate. With its seven degrees of freedom—an incredible amount of flexibility—Baxter enhances arm stability by attempting to fix its elbow in position whenever the lower arm is moved.

Note: When the switch is pressed, the blue indicator lights illuminate on the corresponding navigators on the arm and torso.





When grasping the training cuff, you can move the arms by either repositioning the lower arm or changing the height of the elbow.



To move the lower arm: While squeezing the cuff (1), move the robot's arm to the desired location.

To move the elbow: By design, the elbow (2) will try to maintain its current height and will spring back if you do not actively reset it. While squeezing the cuff, move the elbow to the desired position. Continue to hold the elbow at the new location, and release the cuff. This will reset the elbow at the new position.

Grasping Objects

Training involves showing Baxter how to pick up and place objects.

To grasp an object: Position the gripper over the object, press **Grasp**.

To release an object: With an object in the robot's hand, press **Grasp**.

To open or close the gripper without creating a pick or place: Without an object in hand, press **Grasp** twice quickly.



Active Collision Avoidance

Rethink robots are designed to “know” where their arm joints are at any given time and where the head and torso are so as to avoid collisions with any parts of itself. For Baxter, this means one arm won’t collide with the other, its torso, or its head. For Sawyer, the arm will not collide with the torso or head.

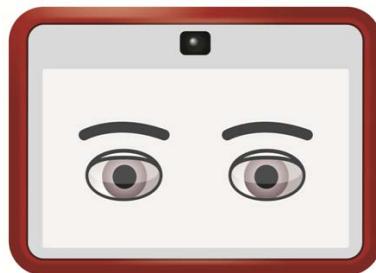


How Baxter Interacts

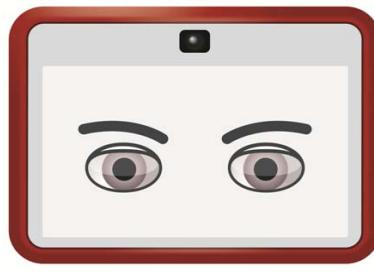
Baxter communicates through a combination of eye expressions, light rings, and thought bubbles. Baxter also responds to touch on a navigator or a training cuff—it stops moving and turns its head in the direction of contact on any of its primary touch points.

Eye Expressions

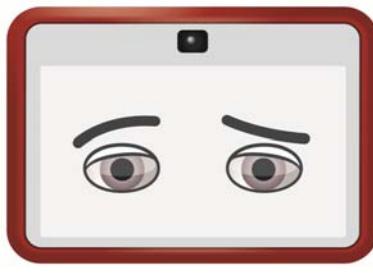
Baxter displays one of six eye expressions in response to what it is doing or what it senses happening in its environment.



Neutral



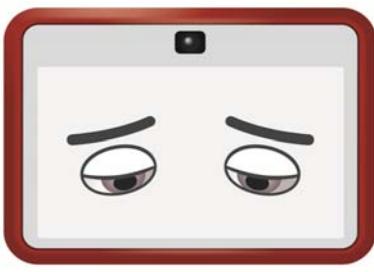
Concentrating



Confused



Surprised



Sad



Sleeping

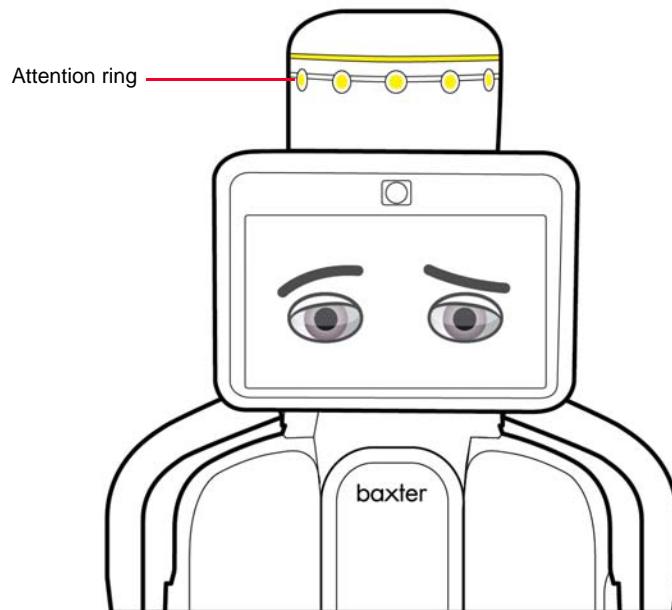


The surprised expression is emphasized with an orange background when Baxter is working and unexpectedly detects someone has entered its space (currently, this only happens when a safety mat is connected and stepped on); Baxter also automatically slows its movement.



Attention Ring

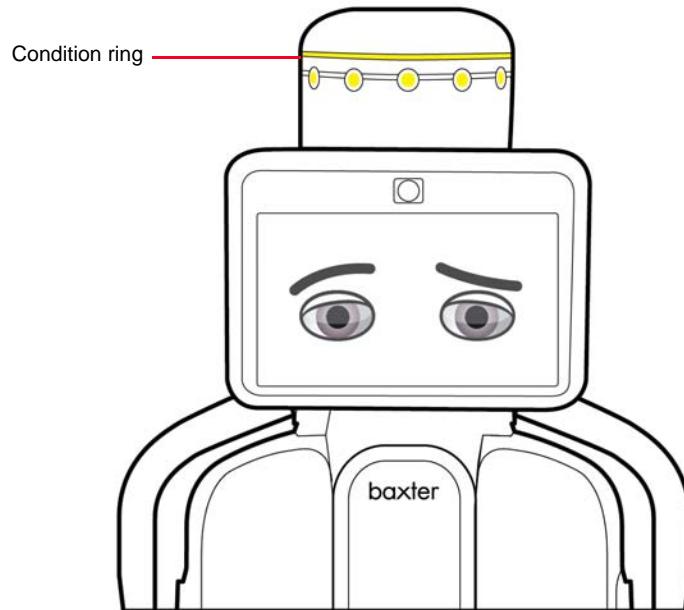
The *attention ring* lights appear in clusters of two or three when Baxter detects movement. When Baxter is confused, the yellow lights in the ring appear and flash simultaneously.





Condition Ring

The *condition ring* communicates the condition of the robot.

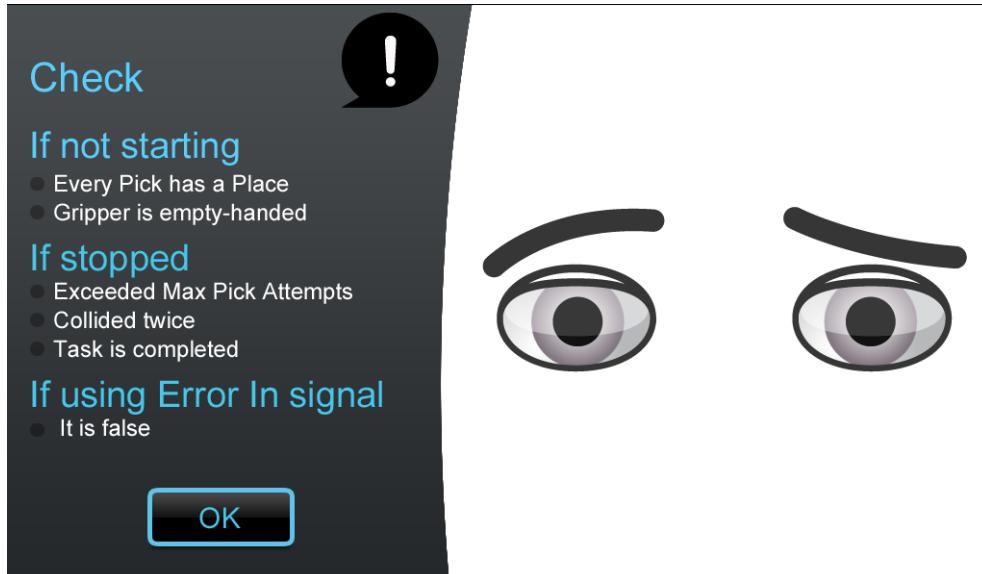


Light Color and Pattern	What it Means
Solid green	Baxter is running a task
Slow pulsing green	Baxter is either idle or being trained by a user
Solid yellow	Baxter is confused and needs user assistance
Slow blinking yellow	Baxter is in sleep mode; its motors are disabled
Fast blinking red	Baxter reports an error



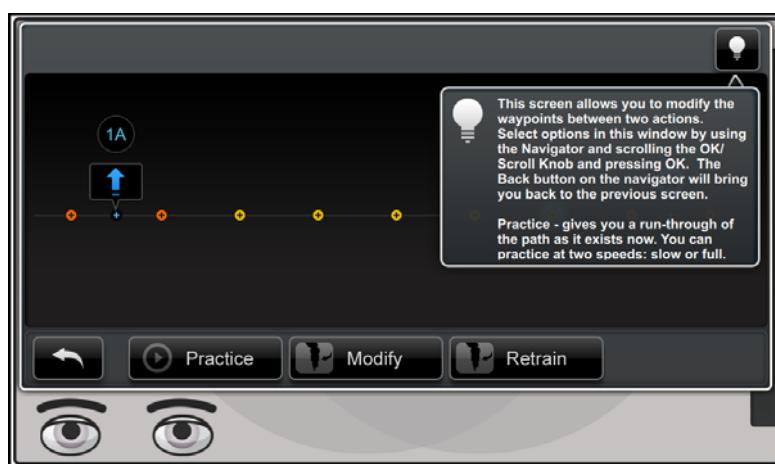
Confused Face Messages

When Baxter becomes confused, the display offers a list of possible explanations and solutions.



"Light Bulb" Tips

When you see a "light bulb" symbol  on a screen, that means there is a tip (or tips) on how to use that functionality. Select the light bulb to display the tip. (These tips are available on the Modify Waypoints Screen, the Advanced Screen, and the Action Practice screens.)





Quick Start Baxter Tutorial

Performing a Task in Just a Few Minutes

Before taking this tutorial, you must set up your robot by following the instructions contained in the documents that ships with the robot and install and configure your end of arm tooling according to the steps described in the “Getting Started with Baxter” chapter.

This tutorial will help you get your robot up and running quickly. The bulk of this User Guide explains Baxter in more detail -- its parts, terminology, how to perform various tasks, create paths, etc. -- but to get a very basic idea of how to operate Baxter, start here.

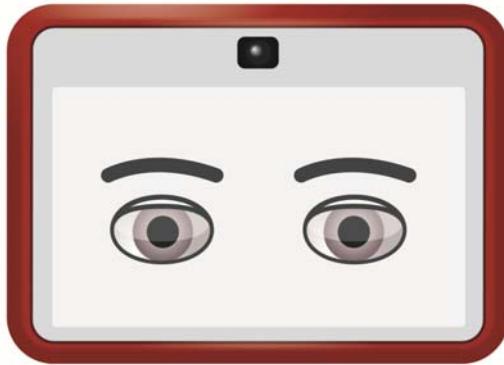
NOTE: For Baxter to perform properly, it should be calibrated (see “Calibrating the Arms” on page 155), before you train this task. Otherwise, Baxter’s arm may pull or miss Picks when performing tasks.

Turning On Baxter

To turn Baxter on, press and release the white power button on the lower left back of the robot.

IMPORTANT: Do not hold the power button down when powering on. It could damage the robot.

The lights on the head turn on, and the display eventually shows a neutral face.



1. Move the Arm

Grab anywhere along Baxter's arm and push and pull on it slightly to feel its resistance. Now, grab the indented portion of the *training cuff*, the part between Baxter's wrist and grippers, and squeeze it just above the buttons on either side. Baxter is now in "Zero G" mode and you can now move the arm easily.



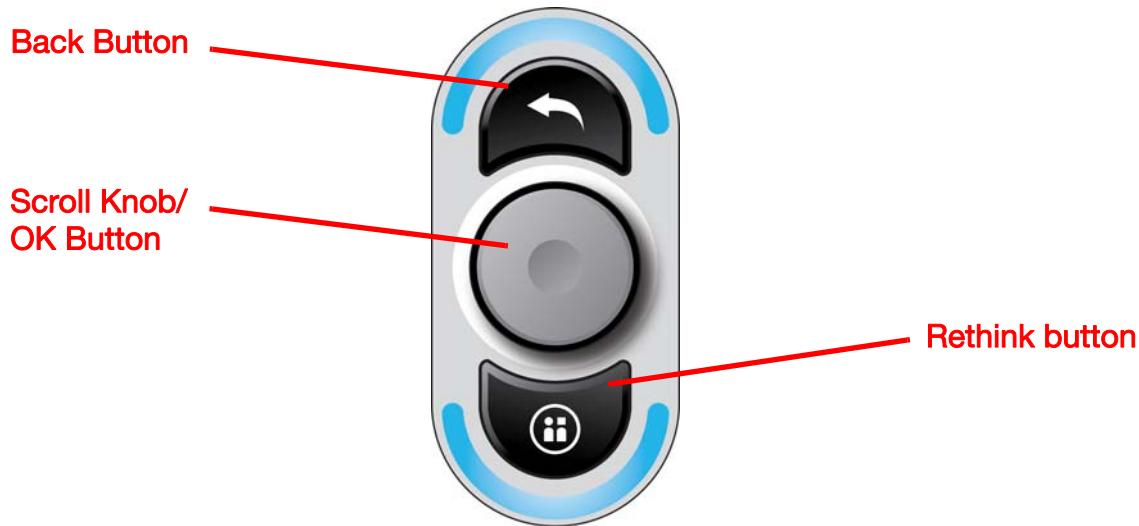


Release the training cuff and the arm becomes (semi-) rigid again. Note that the arm stays in the location and orientation it was in when you stopped squeezing the training cuff. The location and orientation of the arm (its shoulder, elbow, wrist, and so on) is called its *pose*.

THE NAVIGATOR

On both of Baxter's arms, and on either side of Baxter's torso, is the Navigator, a set of buttons and a knob you use to make selections on Baxter. The selections you make on the Navigator are shown on Baxter's display.





Create a New Task

The job you train a robot to perform is called a *task*. A task can be very simple, like the pick and place you’re about to create, or much more complicated, involving multiple pick and place locations, a variety of poses, and sending and receiving signals from other machines and devices.

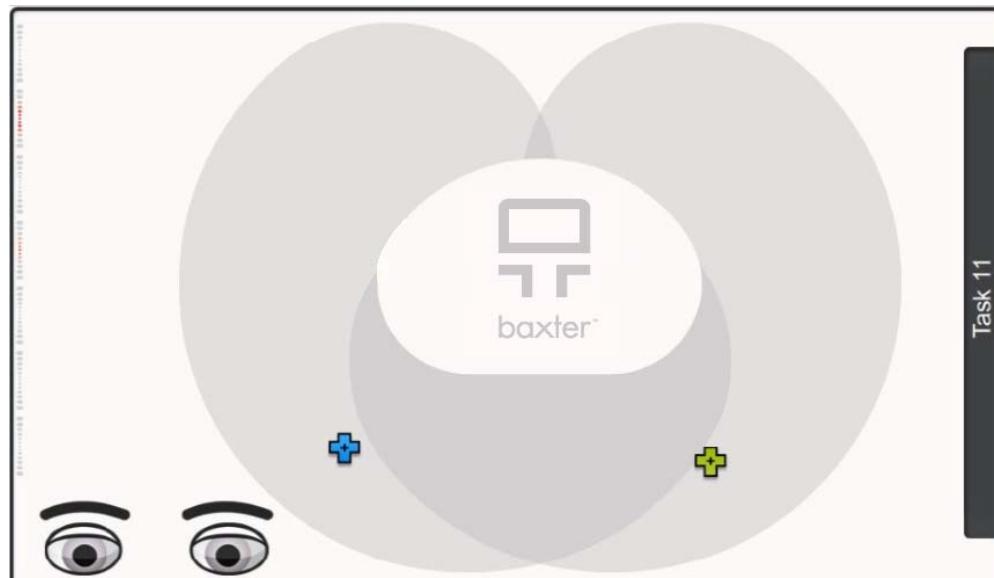
In this module, you will use the selector knob to scroll through options on the robot’s display and press it to make a selection. We refer to pressing the scroll knob to make a selection as “pressing the **OK button**” or sometimes, “press **OK** on the Navigator.”

You press the **Back** button on the Navigator when you want to return to the previous screen.

Scroll the knob to reveal the main button bar and stop scrolling when you reach the New Task icon.



Press OK on the Navigator. Baxter displays a blank *Task Map*.





The Task Map is a top-down view of the Baxter workspace (a.k.a. work envelope).

- Blue icon - the location of the end of the right arm
- Green icon - the location of the end of the left arm
- Dark gray shaded area - where the arm can reach
- Indicators along left side - reflect the current state of each joint in relation to its hard stop limit. Bottom indicator represents the joint closest to the robot's base. Indicator on top is the training cuff. Other joints are represented in order between the cuff and the base.
- Bar on right - task name
- All of attributes of the Task Map are covered in more detail here “Task Map Attributes” on page 43.

Move Baxter’s right arm in Zero G by pressing and holding the training cuff. Watch the blue icon move on screen in response to the movement of the arm.

Notice that when a joint moves closer to its limit, the indicator lines turn from gray to red. If an indicator turns completely red, you have reached the joint’s limit. If the blue arm icon itself turns gray, you will be unable to train an action until you move the arm to a better position. This may be because one of the joints is at a hard limit, which would prevent the robot from performing an action.

A gray arm icon could also mean the arm is too close to the head. The robot has anti-collision software that protects the robot from coming into contact with itself. You can see this in action by trying to push the end of the arm into the head while in zero G.

Grasp an Object

Now that you’re familiar with the features of the Task Map, you will train the robot to pick and place an object and run the task.

Remember: For you to train a pick and place in this exercise, you must have a Rethink gripper attached and configured. To maximize performance and repeatability, the gripper’s length and weight must be configured accurately. If you don’t have a gripper attached, attach one now and refer to the instructions in “Grippers - Configuring” on page 6.

There are two buttons on the training cuff we have not used to this point. The oval button is called the Grasp button. The round button is the Action button.



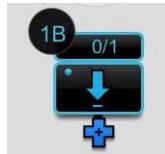
The Grasp button makes the gripper open and close. The Action button brings up additional menus on the Task Map. Press each button now to see the effect of each.

Choose an object for this tutorial and put it in the robot's workspace. With the right (blue icon) arm in Zero G mode, position the gripper on the part (if you're using a vacuum gripper) or around the part (if you're using the electric parallel gripper) and press the Grasp button.

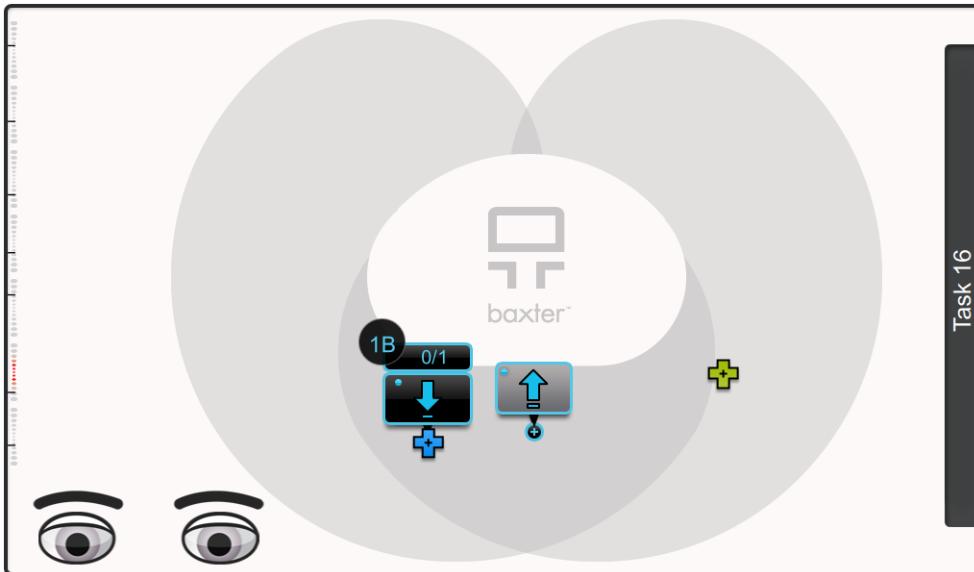
The robot grasps the object and the Task Map displays a Pick icon in the location where the part was picked. Notice that the top left corner of the Pick Icon displays **1A**. That means the action you just trained is the first action in the first subtask.



With the part in the robot's gripper, move the right arm (in Zero G) to where you would like to place it. Press the Grasp button to release the part from the gripper. A Place icon is displayed on the Task Map where you trained the place, along with a subtask number and sequence letter.



Your Task Map will now look something like this:



Run the Task

Return the object to the spot where you trained the pick. Press the **Back** button on the Navigator to display the Main Screen, then scroll to and select **Reset** to start the task. The arm moves to the pick location, picks the part, moves to the place location, places the part, then automatically resets.

Note: The first movement the arm makes, from its pose to the approach point of the first action, is always slower than the speed of the rest of the task. That gives you time to stop the task should the arm be in danger of colliding with a fixture as it moves to the first action.

Congratulations! You have just trained your first task using Baxter!

Press the **Rethink** button on the Navigator and select **Rename**. Then name this task “Task 1” so you can refer back to it as you continue through this guide.

There's much more to learn about training Baxter. Keep reading for more information about how to build tasks for Baxter.



Getting to Know Intera

What is Intera?

Intera is the name given to reflect our robot's interactive software platform. Intera provides an easy-to-use graphical user interface that users can master quickly. The platform allows Baxter and Sawyer robots to be trained by demonstration, using context instead of coordinates to enable non-technical personnel to create and modify programs as needed. It intelligently handles changing environments, while providing an extensible platform that leverages modern tools such as ROS to maximize relevance and flexibility for the modern workforce.

The same Intera software is installed on Sawyer and Baxter, so if you know how to use one robot, you know how to use the other.

In the Quick Start Tutorial chapter (page 29), you learned how to quickly train the robot to perform a pick and a place. Read below for more detailed information about training a task in Intera.

The Basics: What is a Task?

At its most basic, the robot is trained by a user to perform a “Task.”

A task refers to the *job* the robot is trained to perform. A task is made up of many different task elements, called subtasks, actions, paths, and signals. The robot can have many tasks saved in its memory, allowing a user to quickly re-deploy the robot on a number of saved jobs.

To help you become familiar with navigating through Intera, this section explains the basic elements that make up a task, as well as the screens, icons, and settings you will encounter on the robot’s User Interface.

Note: For a glossary of terms found in this User Guide, see “Appendix A: Glossary” on page 159.



The Basics: What is an Action?

The foundation of all work done by the robot is called an *Action*. An Action occurs in a defined location in the workspace and is either a:

- Pick - where the robot attempts to grasp an object.
- Place - where the robot attempts to release an object.
- Hold - a location where the robot will move through, or wait at, during a task.

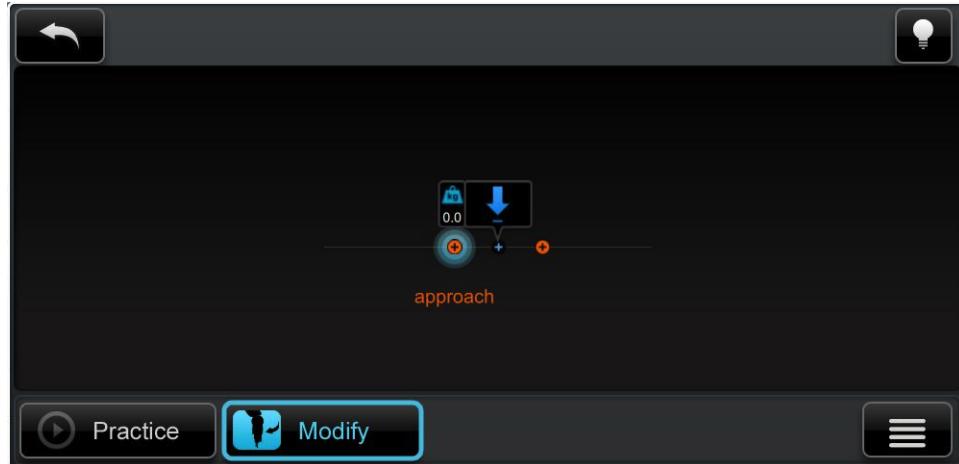
In Intera, each Action consists of 3 specific points:

- Approach Point
- Action Point
- Retract Point

These points are simply the sequence of movements the arm will take while attempting an Action.

For example, when the robot attempts a Place action in a task, the end of the arm begins the Place by approaching the Place location from a particular distance and orientation specified by the person who trained the task. This point is called the “approach point.” After the gripper places the part, the end of the arm Retracts away from the Place location in a manner also specified by the trainer. This point is called the “retract point.”

The image below is a Place Action’s Modify Panel. You can see that for this Place, there are three specific points: from left to right, the Approach > Place > Retract. “Approach” is currently highlighted.

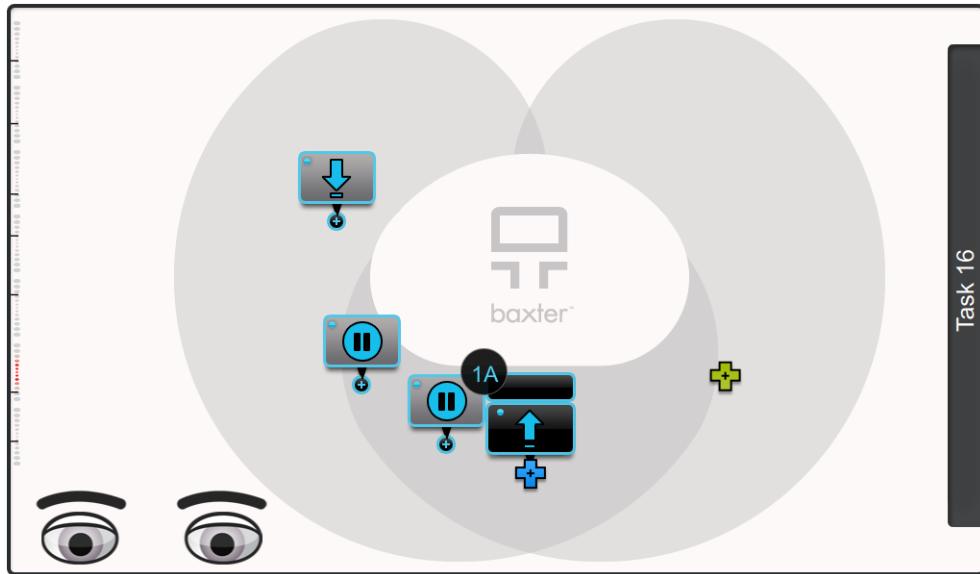


Labeling Actions

Generally, the robot's tasks involve Pick actions followed by Place actions. To make it easier to distinguish one action from another, actions are labeled on the Task Map with the subtask number and appended with a letter. For example, if subtask 1 includes a Pick>Hold>Hold>Place, the actions on the Task Map will be labeled:

- Pick 1A
- Hold 1B
- Hold 1C
- Place 1D

And the Task Map would look like the following illustration. Note that the **Pick 1A** is currently selected:



The Basics: What Is a Subtask?

A subtask is a routine, or sequence of actions, the robot will perform within a task. A task can have one or multiple subtasks. As with actions, subtasks can have counts and signals associated to them.



Navigation: Main Screen

When the robot is fully booted, the first screen to appear on the head display is called the *Main Screen*. Baxter's main screen has two elements: the *button bar*, and *eye expressions*.



1. Current task name
2. Current task options
 - Run – run the task from where the task left off.
 - Reset – reset the count, and restart the task from the beginning.
 - Modify – open the *task map* so the user can make changes to the task's elements.
3. Baxter eyes – The robot uses eye expressions to communicate to users in a familiar way by glancing in the direction in which it's about to move. Baxter also has other expressions to communicate its different states. (See “Eye Expressions” on page 25.)
4. Main options:
 - New – create a task.
 - Tasks – opens the *task gallery*, a visual list of existing saved tasks. See “Navigation: Task Gallery” on page 45.
 - Settings – open Baxter administration and hardware settings. See “Navigation: Settings” on page 50.



- Power – open Baxter power options: Sleep, Restart, Shutdown, and Lock/Unlock (See “Lock/Unlock” on page 51.)

Navigation: Task Map

The *task map* is a graphical representation of the robot’s workspace (or work envelope) from a top-down viewpoint. When you train the robot to perform a task, the map displays icons that represent each of the actions included in the task.

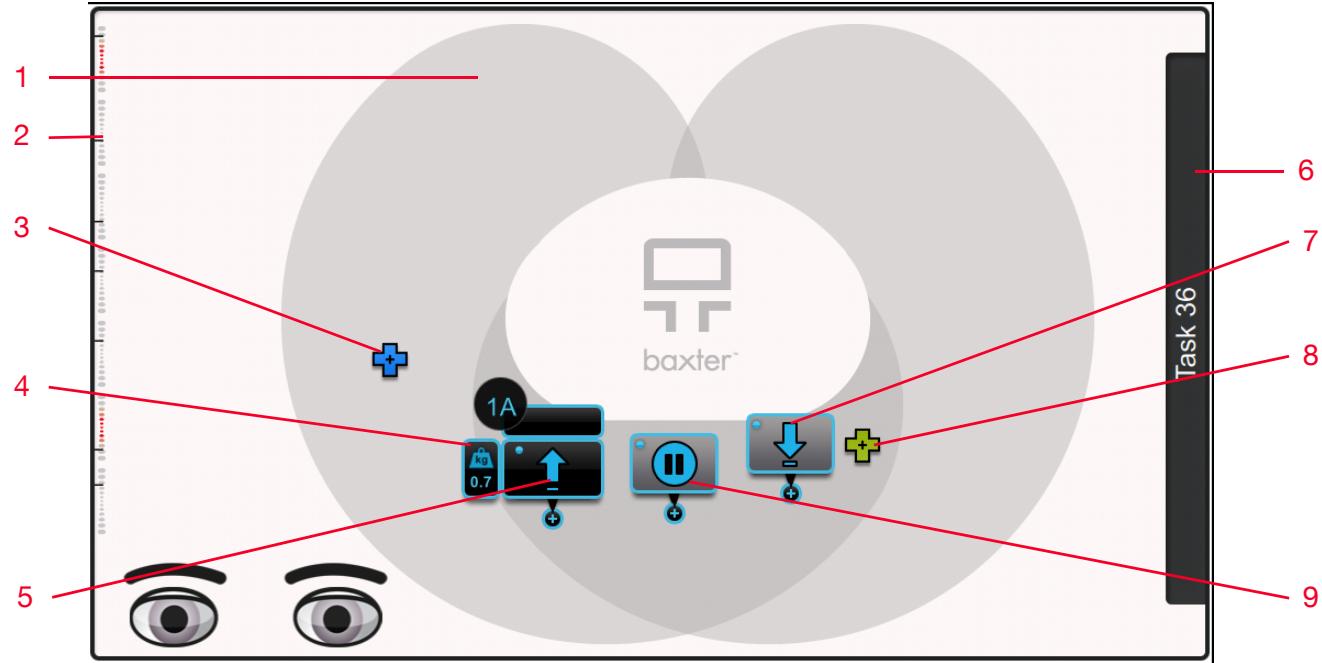
There are several ways to access the Task Map, including:

- Creating a new task
- Modifying an existing task
- Selecting a task from the Task Gallery

The gray shapes —called the *workspace*—represents the maximum mechanical reach of the arms when fully extended. Baxter can execute actions within this area.

Note: The robot’s arms move within a sphere defined by the center point of the shoulder axis and the full reach of the arm. The actual reach of the arm at any given location is affected by the height of the robot’s arm and angles of the joints. Actions are therefore limited at different heights and joint angles. The approach and retract distances for an Action may also limit where the Action can be performed.

Tip: Baxter performs tasks more efficiently when the objects are located close to the center of the workspace. Tasks trained at the extremes of the workspace approach joint limits and are sometimes more difficult to execute consistently.



TASK MAP ATTRIBUTES

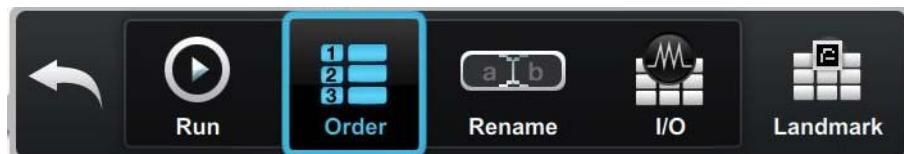
1. Baxter workspace.
2. Joint limit indicators - display the current state of each joint of the active arm in relation to its mechanical limit.
3. Real time location of end of right arm.
4. Weight label - displays the part weight designated for a particular action. (Here, a .7kg weight has been designated for Pick action 1A).
5. Pick action icon - a point in the robot's workspace where the end of arm tooling will attempt to pick an object.
6. Task name - name given to the task currently being modified.
7. Place action icon - a point in the robot's workspace where the end of arm tooling will release a picked object.



8. Location of end of arm - icon that represents the real-time location of the end of the arm.
9. Hold Action - a point in the robot's workspace where the end of the arm will move through, or wait at, when the task is run.



Pressing the Rethink button while on the Task Map opens the task map button bar.

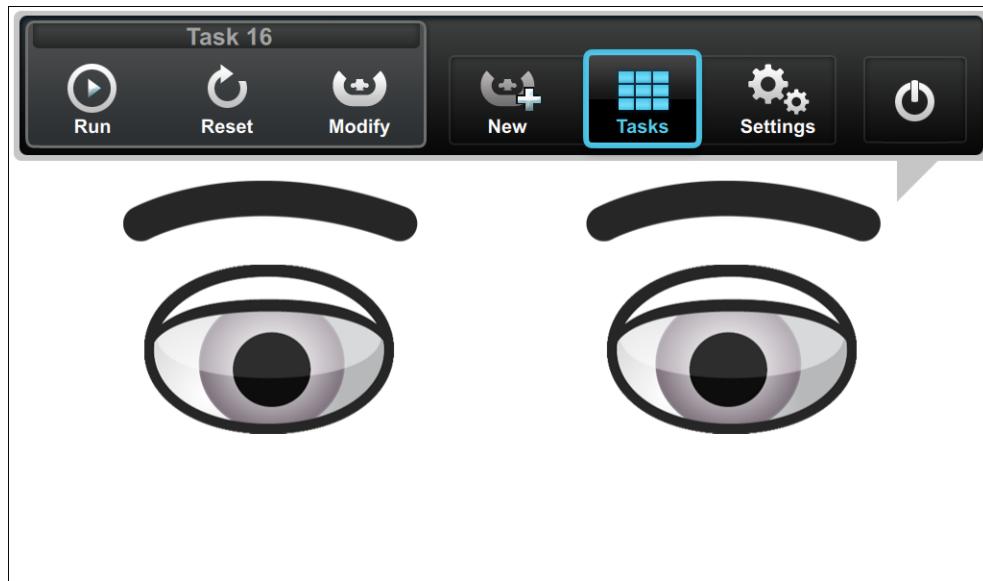


- Back – Close the button bar and return focus to the task map. This can also be done using the Back ↪ button on the Navigator.
- Run – Continue the current task from the point at which it left off.
- Order – Open the task order screen. See “Managing Tasks and Subtasks” on page 94.
- Rename – Modify the name of the task. When starting a new task, Intera will give it a default, numeric value task name, e.g., “Task 7.”
Tip: Entering a unique, descriptive name will help you to more easily identify the task in the task gallery.
- I/O – Open the signals gallery. See “Signals” on page 128.
- Landmark – Open the landmark gallery. (See “Robot Positioning System” on page 110.)

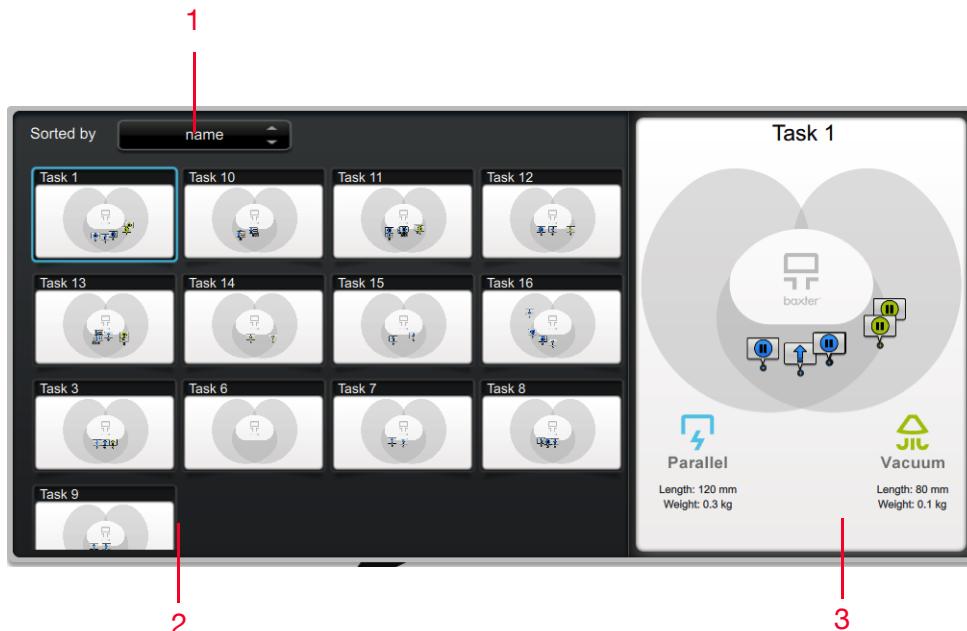
Note: The robot automatically saves tasks as they are created or modified; you do not need to actively save while you train.



Navigation: Task Gallery



The Task Gallery is the list of saved tasks on any robot. It is accessible from the main screen, when you select the **Tasks** button from the menu bar. Use the task gallery to view the details of and select, copy, or delete a trained task.



1. Sort the tasks in the Task Gallery by selecting one of these options in the scroll box:
 - **name** - alphabetically, by name of task
 - **modified** - when the task was last modified
 - **created** - by most recently created task
2. Displays a visual list of all saved tasks, each with the name of the task and a small preview image of the task map.
3. Displays details about the highlighted task.
 - End effector specifics - shows what end effector was used to train the task, along with its parameters (weight and length).
 - Expanded task map view

 - This symbol indicates a mismatch between the installed gripper and the one used in the viewed task.



Note: The task gallery cannot be empty. If you delete all tasks, an empty task is created automatically.

To navigate to a task: Rotate the **knob** to scroll through the tasks.

To select and open a task: Scroll to the task until it is highlighted, and press **OK**. A larger preview of the task map opens in the left panel, and the task gallery button bar opens at the bottom of the screen.

After choosing and selecting a task from the gallery, the User Interface will display a submenu for that task.



- Back – Close the button bar and return focus to the task gallery. This can also be done with the **Back** ↩ button on the Navigator.
- Open – Open the task map for the selected task.
- Rename – Modify the name of the task.
Tip: Rename a task with a descriptive name when you first create it so that you can easily identify it later in the task gallery.
- Delete – Delete the task.

Note: The robot must always have at least one task stored. If only one task exists, and it is deleted, the robot will create a new “empty” task.

IMPORTANT Once the deletion is confirmed, the deleted task cannot be restored.

- Copy – Create a new task based on the current one.

Tip: If you’re training a complicated task, save a record of your changes by copying the task as you build it. You can always refer back to the previous version of the task in the Task Gallery.

- New – Create a new, empty task and open the task map.

You can also delete all the tasks on the robot from the Task Gallery.

To delete all the tasks on the robot:



1. Press the **Rethink** button while in the Task Gallery.
2. On the submenu that appears, select **Delete All**. You will be asked to confirm the deletion of all tasks.



IMPORTANT: Be careful when using **Delete All**. Unless you have previously backed up your tasks, all tasks are gone once you delete them. There is no recovery of tasks without a backup.



Navigation: Settings

About Baxter

About Baxter displays information about your specific Baxter robot: serial number, IP address, current software version, the date of the software's latest build, the current time, and the amount of time this robot has been operating, measured in hours and minutes.



Hardware Settings

Calibrate Screen - Displays a checkerboard pattern and performs a screen calibration for display size and contrast.

Prepare for Moving - Returns Baxter's arm to its home position so it can be moved or shipped. This will also shut down the robot.

Configure End Effectors - Configure the characteristics of the end effectors (type, weight, length) Baxter will be using. See "Grippers - Configuring" on page 26.

Configure Modbus - Configure a Modbus TCP device to communicate with Baxter. See "Appendix B: Configuring External Devices" on page 160.



Advanced

Update Software - Install the latest software on the robot via a USB stick. See “Upgrading Software” on page 151.

Export/Import Tasks - Back up your tasks by exporting to a FAT32-formatted USB stick. Import tasks that were previously backed up to a USB drive. For details see page 152.

Calibrate Arms - Calibrate the arms to troubleshoot common error messages, and to help Baxter maintain its peak performance. For more, see “Calibrating the Arms” on page 155.

Diagnostics - Baxter logs its activity and saves it into files. When an unidentified problem occurs with the robot, your technical support representative may ask you to export log files to a FAT 32 USB device, download them to a PC, and upload the files to Rethink Robotics support. The files may help them to troubleshoot and fix the problem.

To export a log file:

1. Select **Setting > Advanced > Diagnostics**.
2. Insert a formatted FAT 32 USB device into the robot. Make sure there is at least 2 GB of available space on the device.
3. Select **Export Logs to USB > Export**.

The log files copy to the USB device.

Lock/Unlock

You can lock and unlock the robot using a password stored on a USB stick.

This feature helps prevent unauthorized personnel from tampering with the tasks stored on the robot. If it is locked, users can only run, reset, clear errors/confusion, or power cycle on the robot. When Baxter is locked, tasks cannot be created, modified or changed.

If the robot is unlocked, users have access to all of the robot's features, tasks, etc.

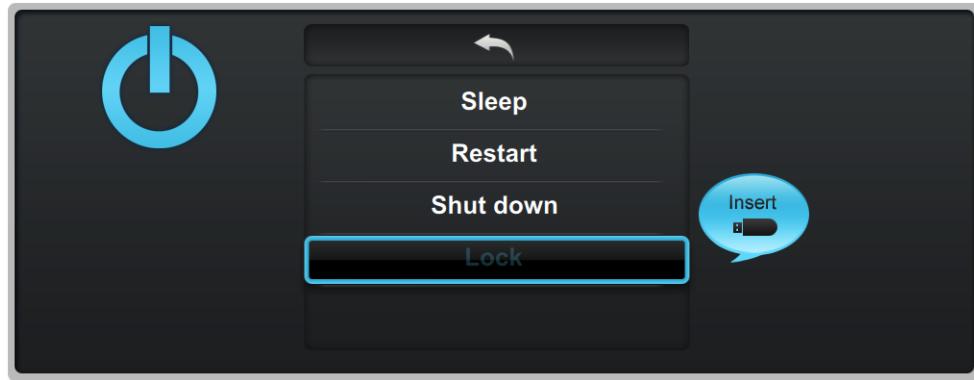
This feature requires a FAT32 USB device containing a particular text file that the robot will recognize as the locking password.

For access to this lock file, contact Rethink Robotics support (support@rethinkrobotics.com).



How to Lock or Unlock Baxter:

1. Insert the USB device containing the .txt file into the USB port on the back of the robot. If not inserted, the lock/unlock option will look like this:



2. Once the stick is inserted, from Baxter's button bar, select **Lock**.
3. Select **Lock** or **Unlock**, depending on what state you want the robot to be in.



Training a Task In Intera

Training a Pick and Place

If you completed the Quick Start Baxter Tutorial (see page 29), you already know how to train a Pick and Place task. In this chapter we're going to start a new task that you can continue to build upon as you read through this User Guide.

Start by training a Pick and Place task:

1. In the main button bar, click **New** to start a new task.



2. Enable zero-G mode by squeezing the training cuff.
3. Move the arm to the location in the robot's workspace where you'd like to pick the object from. If using an electric parallel gripper, poise the fingers to grip the object. If using a vacuum gripper, place the suction cup on the object.



4. Press the **Grasp** button on the training cuff. The robot enables the gripper and grasps the object. The head will nod, indicating a successful action has been created. Verify that the Task Map now displays a Pick icon.



5. Press the **OK** button on the Navigator.
6. Select and press the + button on the Pick's modify panel.





Select the **weight** icon to add the part weight and press **OK**.



IMPORTANT: Always enter the part weight when training a Pick. The robot will account for the weight of the part during movements after the Pick action, ensuring the most accurate trajectories and placement.

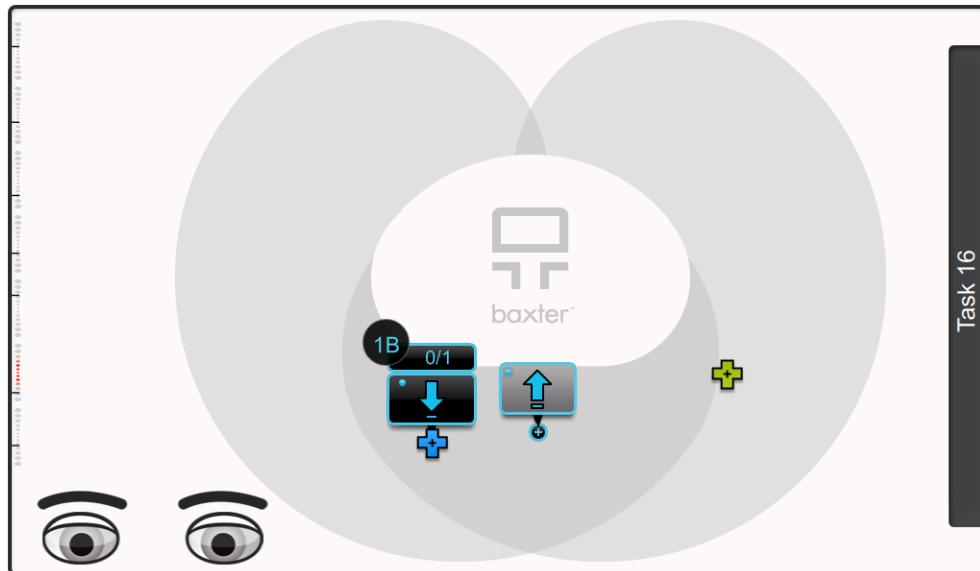
7. Enter the weight of the part and click **OK**. The weight you entered will be displayed on the Modify Panel as well as next to the Pick icon on the Task Map.



8. Press the **Back** button on the Navigator to return to the Task Map.
9. Move the arm in Zero-G to the location where you want to place the object.



10. Press the **Grasp** button once to release the part and create a Place location. The robot releases the object and displays a Place icon on the Task Map.



11. Press **Back** to open the Main Button Bar and select **Reset** or **Run** to perform the task.

Renaming Tasks

Before moving on to the next section, rename the task from the default name to “1st Task.”

1. While on the Task Map, press the **Rethink** button to display the Task menu.
2. Scroll to **Rename**.

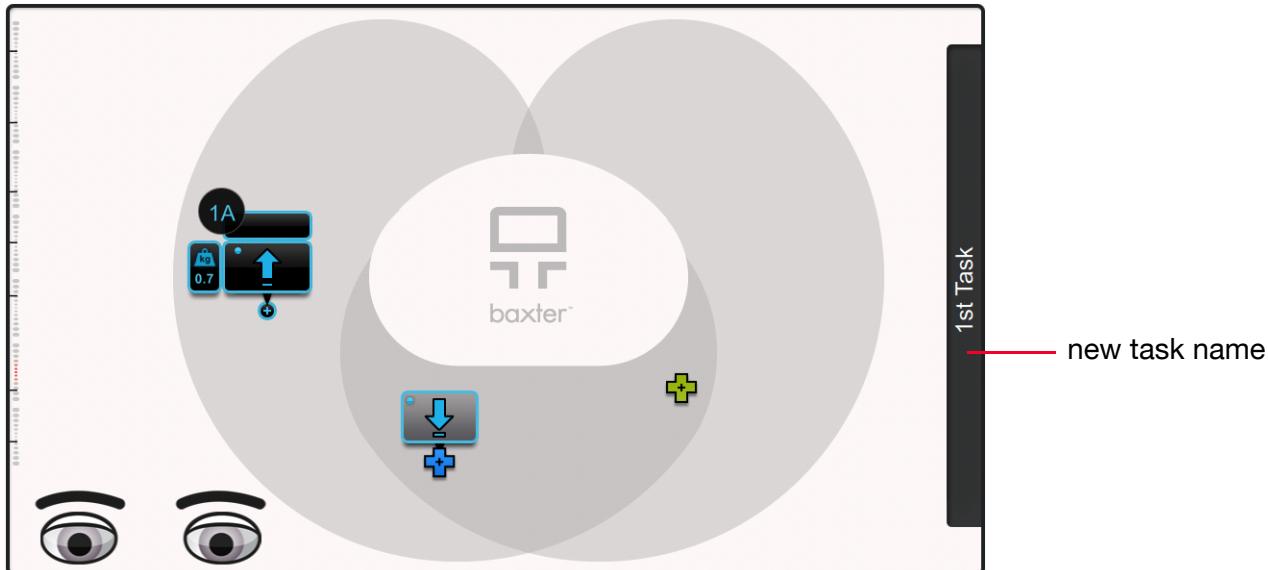


On the Rename panel, enter “1st Task” and press OK.





The name bar on the right side of the screen displays the new task name.



Tip: If the robot's wrist joint is not pointing straight down when performing an action, meaning the action is not vertical, a red circle is displayed next to the blue icon representing the end of Baxter's arm.

Tip: If the arm icon is gray, Baxter cannot create an action in that arm pose. The reason for this may be that the arm is in a "collision bubble," where the robot would hit itself if trained to go to that location. Or the arm may have a joint near its limit as indicated by the Joint Limit Indicator on the left side of the Task Map.

Training a Hold Action

The robot can also be trained to wait at a pose for a defined period of time until a signal is triggered. This feature is useful for such jobs as holding an object in a pose (or a series of poses) for inspection, scanning, labeling, painting, etc. It can be used when you want to move an arm to a neutral location to wait for machine cycling, or for a process to be finished.

Hold actions work with or without a part being grasped, and in any pose or orientation.

Like Pick and Place actions, Hold actions have Approach and Retract points. When a Hold is trained, its default Approach and Retract points are 0cm. Unless these points are modified on the



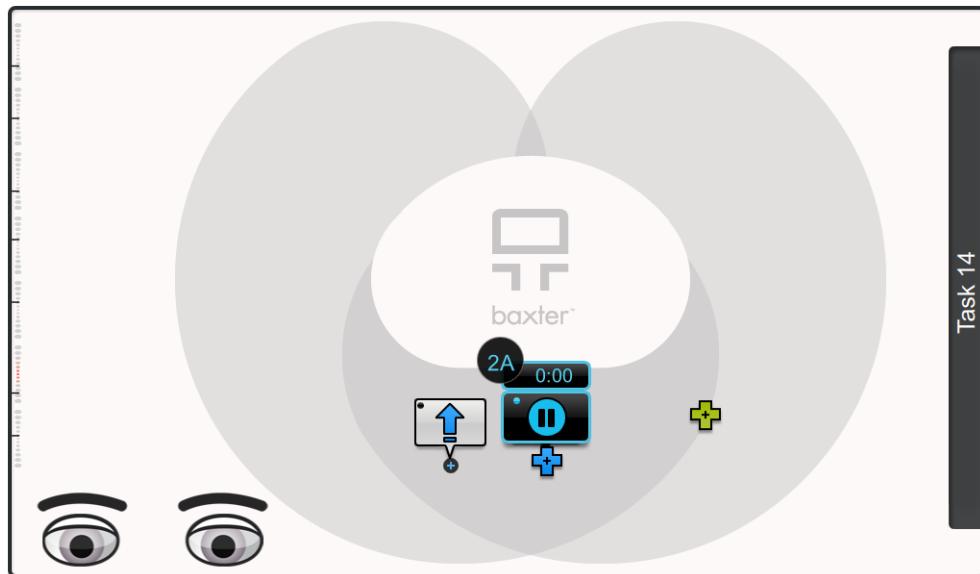
Hold action's modify panel, there is no additional arm movement from the Approach > Hold > Retract.

How to Train Baxter to Perform a Hold

1. While on the Task Map, press the Action button (the round button) on the training cuff to display this submenu:



2. Scroll to the **Hold** icon and press the Action button again to select the Hold.
3. If the Hold was created successfully, the Hold Action will appear on the Task Map and the head will nod.



Reordering a Hold Action and Adding a Hold time

Depending on your task, you may need to reorder a Hold action in the sequence of the task.

In the **1st Task** example you created, the Hold action comes after the Pick and Place Actions are complete because that's the order in which the task was trained. But what if the job was for the



robot to hold a part in front of a barcode reader for two seconds before placing the part in a box? Rather than start over and train the actions in this new order, let's reorder the Hold action for this task so that it takes place before the Place action.

To add a hold time of two seconds:

1. Highlight the Hold action on the Task Map and press **OK** on the Navigator.
2. Select the **+** button on the menu and scroll to the **Time** Icon.
3. Press **OK**.
4. Use the seconds scroll wheel to select **2 Seconds** and press **OK**.
5. Press **Back**.

When the task is run, Baxter's arm will now pause for two seconds at the Hold location before moving to the next action.

To change the order of a Hold:

1. While on the Task Map, select the Rethink Button on the Navigator.
2. Select **Order** from the task menu.
3. On the Order Screen, scroll to the **Hold Action** and press **OK**.
4. Select **Reorder** and press **OK**.

The yellow bar that appears can be moved by scrolling the navigator button.

5. Scroll to change the position of the yellow bar so that it comes between the Pick and the Place Actions.
6. Press **OK**.

Notice that on the order screen, the Hold Action now is sequenced after the Pick and before the Place.

7. Press **Back** to return to the Task Map, then run the task.

Training a Group of Pick or Place Actions

An Action Group refers to two or more like actions combined into one single Task Element. An Action Group is used to combine arrays of individual Picks or Places into one item in order to allow for the advanced settings of all of the individual actions to be shared. Creating an Action Group also



allows for Counts, Part Weights, and Signals to be assigned to the overall group. A common use for Action Groups is box packing where parts will be placed into a grid or array.

Action Groups are particularly useful because the robot remembers the order in which an action group was trained, and will perform actions in that order, while running the task.

In the **1st Task**, you trained the robot to pick a part, hold it for two seconds and place the part to a single location. For this next job, let's say the robot is packing parts into a box that has four parts in each layer.

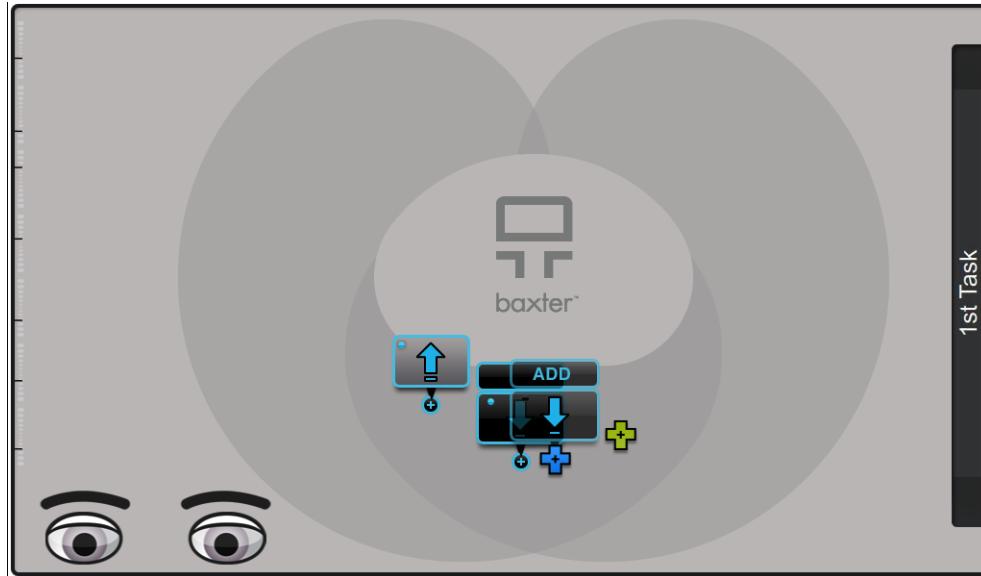
Follow the steps below to train an action group that will allow the robot to place the parts in specific, individual locations inside the box.

How to Train an Action Group

1. Highlight the action you want to use to start a group. Press **OK**.
2. Scroll to and click **Add** on the Modify window.



3. On the Task Map, you'll see an icon labeled "Add." Move the robot's hand to the second desired location, and press the action button. (You can also press **OK**, but the action button is usually more convenient.) The added action is placed in the location.



4. Repeat as many times as needed. (In this example, you'll only need to save three new points inside the box.)
5. When finished, press the **Back** button. (Don't press **OK** to finish. That will continue to add locations to the action group.)
6. Run the task.

NOTE: If, when training the robot, you need to put an object in the gripper and you don't want to copy or add a new action, *double-click* the grasp button.

Creating Paths: Paths, Waypoints, Poses

Before talking about paths, let's first define the terms associated with creating paths in Intera.

A **pose** is a position and orientation of the robot's arm (shoulder, elbow, hand, wrist, etc.) at a waypoint.

Waypoints are poses in a path. Changes in position and/or orientation of the arm are opportunities to create waypoints.



Finally, a **path** is a series of Waypoints (arm poses) the robot travels through while moving to an action in a task. When two actions are trained, the default is for the robot to automatically create a path between those actions. This path will be the most efficient set of poses for the arm to make to travel from one action to another. This is called a **default path** or a **system-generated path**.

There are times when you may want to define the path the robot's arm takes when it moves from one action to another. This is useful to ensure the arm avoids colliding with items in the immediate work area. To accomplish this, you create a path for Baxter. This is called a **custom path**. Only custom paths are displayed on the Task Map, unless you are in Path Mode (accessed when you select the Paths button on an action's modify panel).

Follow the steps below to create and modify paths.

How to Create a Custom Path for Baxter

We will create a Pick and a Place with a user-defined, or *custom* path.

1. Create a Pick for an object.
2. With the object in hand, press the Action Button (the round button on the cuff).

Baxter displays the action bar.

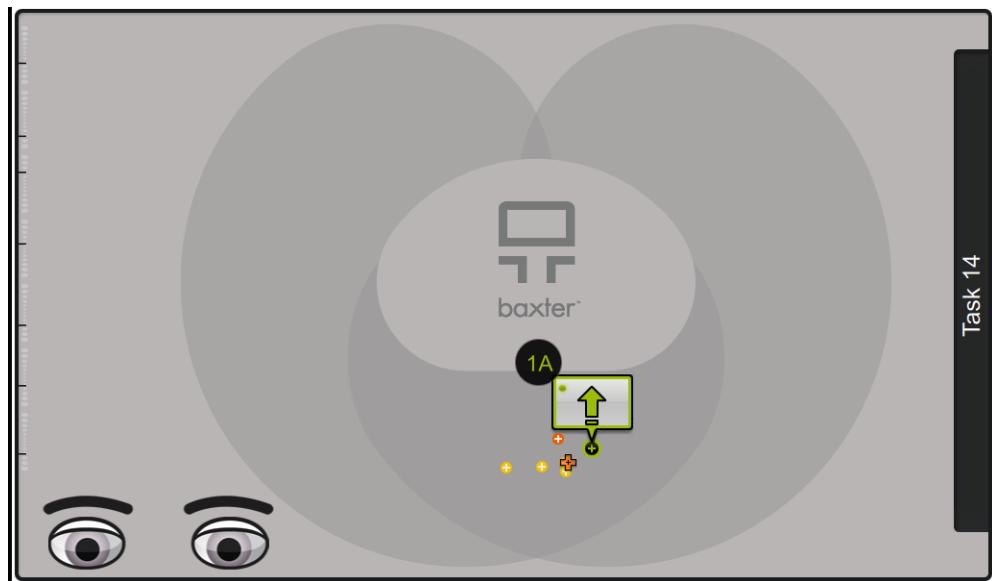


3. Select the Path icon.
4. Elevate the arm above the location of the Pick, pose the arm and hand as desired, and press the Action Button to train the first waypoint, which is the action's first retract point. Because you're creating a custom path, this first waypoint will become the Pick's *Retract point*. A Retract point is the location and pose to which the robot will return after an Action. (This is also the location where the "Go Slow" command ends and normal speed resumes. See "How to Make Detailed Modifications to an Action (Advanced Settings Screen)" on page 89.)

A "path mode" crosshair indicates on the Task Map the location of the Retract point.



5. Move the robot's arm to the next waypoint, keeping in mind the position of the arm and hand as you move it along the path. Press the action button to establish another waypoint.
6. Continue creating the desired path by moving Baxter's arm and hand and pressing the action button on the cuff to create a series of waypoints.



The last waypoint you create before the Place action (in this example) will be automatically set as the Approach point. The Approach point is the location and pose of the hand in anticipation of an action--in this example, the point at which Baxter's arm will begin moving toward the Place location.

On the Task Map, Approach and Retract points are orange.

Transit waypoints are points along the path not associated with an action. They are locations that guide the robot's arm along a path. Transit waypoints on the Task Map are displayed in yellow.

The waypoint closest to the Pick or Place automatically becomes the Approach or Retract point. Keep in mind that the robot moves in curved Joint Coordinates along transit waypoints, but in straight Cartesian Coordinates between the Pick or Place and the Approach and Retract points. This is important when the robot needs to move along a straight line in the direction of the gripper, such as when placing an object into a box or assembling two components. However, while the tran-



sit waypoint poses are approximated as Baxter moves along the path, Pick, Place, and Hold poses are precise.

7. Create the Place by pressing the Grasp button.
8. You can now select Run from the Task Map button bar to run the Pick and Place task.

USAGE NOTES FOR CREATING A PATH:

- When you create a path, Baxter remembers the orientation, angle, and coordinates of its wrist for all waypoints.
- If, when training the robot, you need to put an object in the gripper and you don't want to create a new action, *double-click* the grasp button. Baxter will grab the object without creating a new action.
- If you press the Back button while creating a path, all waypoints in that path are deleted.
- When you run a task, notice that the arm rewinds along the path you created when it returns to the Pick location.
- You can add a custom (user-defined) path between poses by modifying and retraining a path. That path will show up on the Task Map. To do this, modify a pose, select the Paths icon in the Modify Path panel, then select the action where the path leads.

Best Practices: Creating Waypoints

- Generally it is not desirable to define a large number of waypoints when training a path. The more transit points you add, the longer the path and therefore, the more time the task will take to run.
- If you train a number of waypoints relatively close together along a path, the robot's arm movement will become jerky.
- Train just those points needed to avoid obstacles that may be in the robot's path.

How to Practice a Path

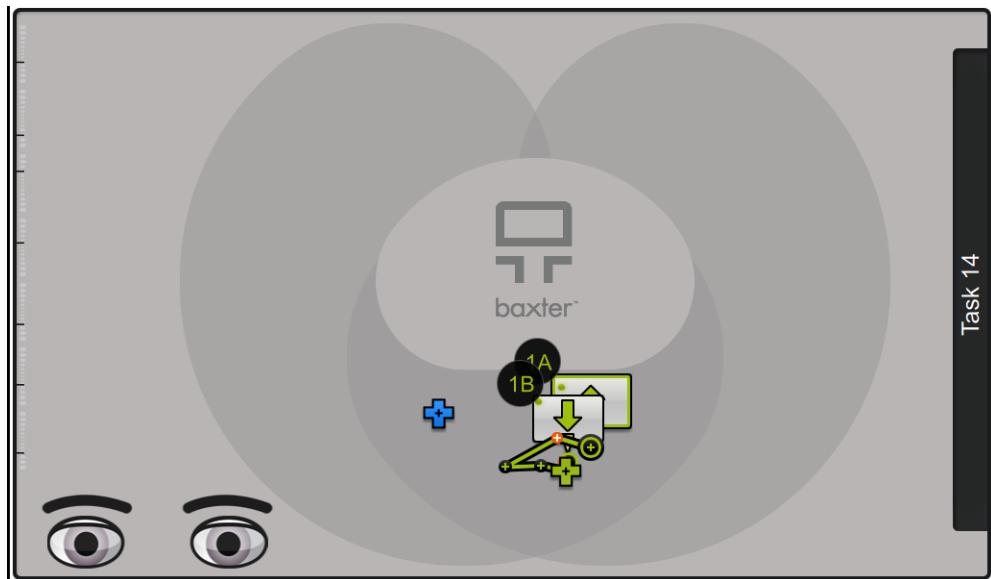
1. Select an action on the Task Map and press **OK** on the Navigator.

The robot displays the Modify Panel for the action.



2. Select the Path icon and press OK.

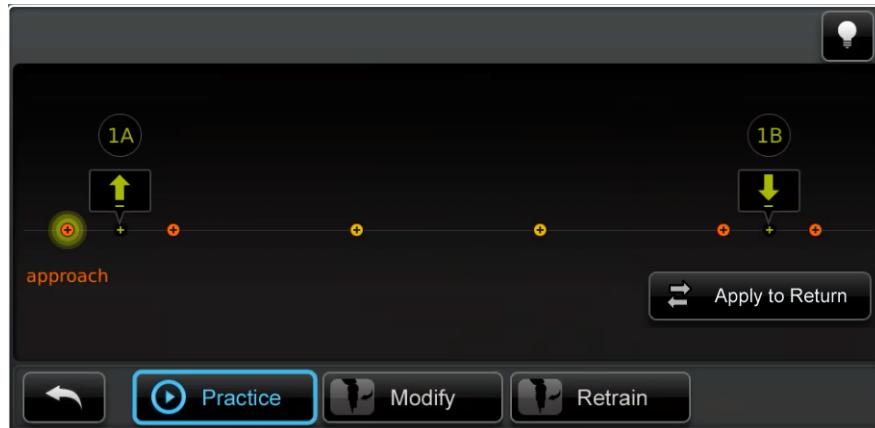
The robot displays the Task Map with the action and its path.



3. On the task map, select the path you want to practice.



The robot displays the Panel for the selected path.



In the illustration, the orange waypoints represent retract and approach points. Yellow points are transit points, the points between the Pick and the Place. The approach pose for the Pick is highlighted.

4. Press the Practice button.
5. Select to practice the path at either slow or full speed. (Slow speed is equal to approximately one-half full speed.)

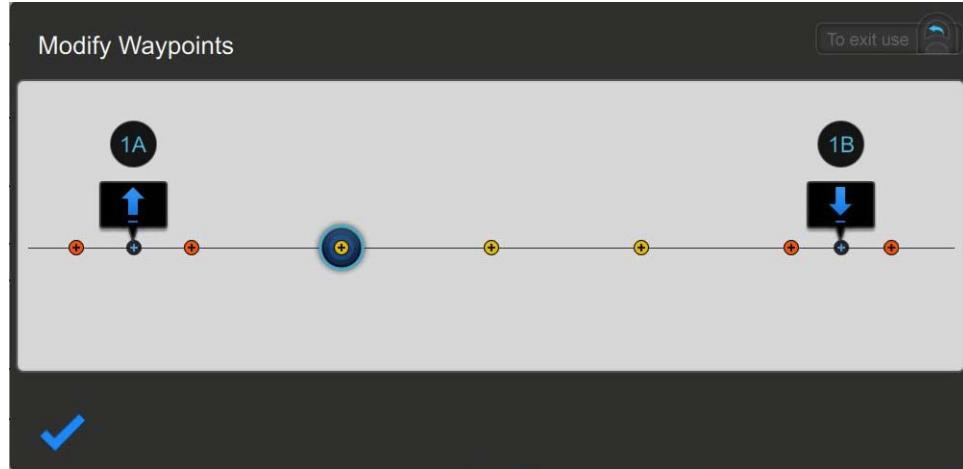
The robot practices the path.

Note: You can grab the cuff while Baxter is in practice mode to stop the arm from moving. That does not alter the waypoints in the path or any of the poses associated with the path.

How to Modify a Waypoint

Note: See “Nudge” on page 83 for details on making precise changes to the robot’s arm and gripper location and orientation.

1. Select the path you want to modify on the Task Map and press **OK**.
2. The robot displays the Modify Transit Panel for that path.
3. Press the **Modify** button.
4. Scroll the knob to the waypoint you want to modify. Each waypoint is highlighted with a halo as you scroll to it. The robot’s arm will move through each waypoint until it reaches the selected waypoint.



5. Grab Baxter's cuff, change its arm location and/or pose and press the action button. A check mark is briefly displayed and the pose for that waypoint is updated.

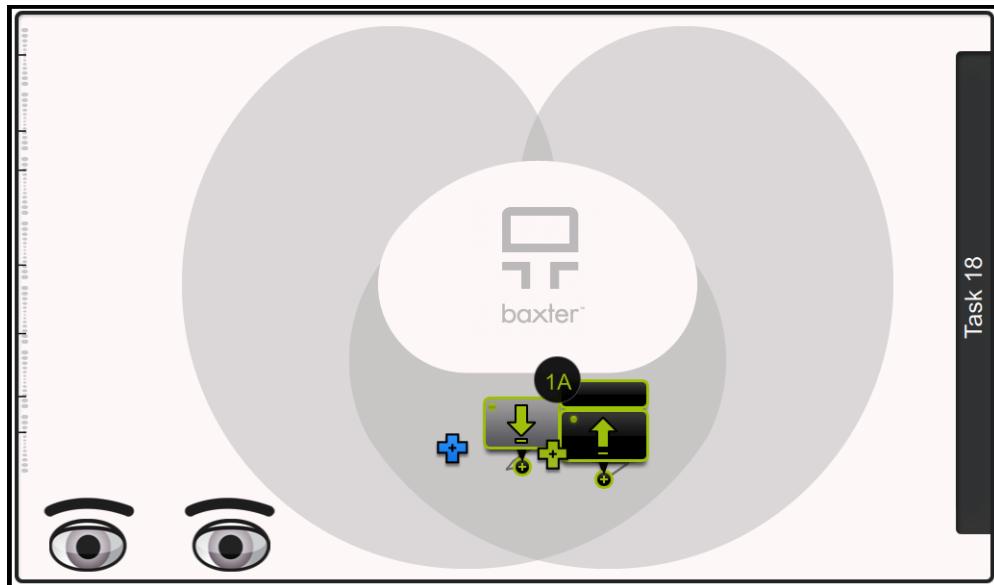
WAYPOINT/PATH NOTES:

- You can scroll through to any waypoint on the path. For example, you can scroll through the path one waypoint at a time—the robot's arm will move to each point in turn—or you can scroll directly to the waypoint you want to modify and the robot's arm will skip directly to that point. Be careful when skipping waypoints using the Modify Transit Panel, though, because it causes the robot's arm to deviate from the path: the arm moves directly between points and it may bump into a fixture or obstacles.
- You cannot nudge (see page 83) a transit waypoint.
- When you see a "light bulb" symbol on a Modify Waypoint screen, that means there is a tip (or tips) on how to use that functionality. Select the light bulb to display the tip.



How to Modify a Path (includes Practice and Retrain)

1. With a task already created, select the action where the path originates. For example, if the path is from a Pick to a Place, scroll to the Pick.

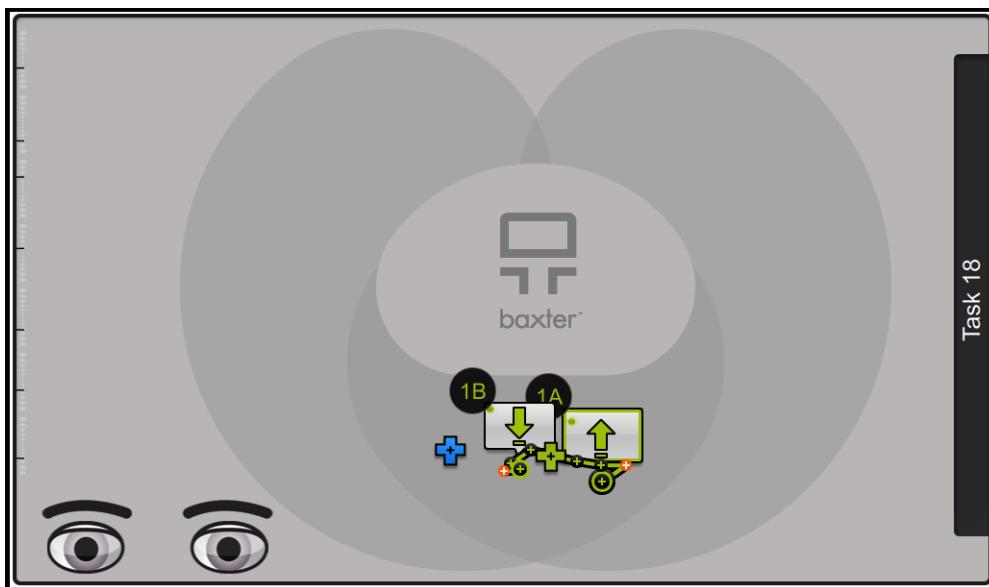




2. Press OK and select the Paths icon from the Modify screen.



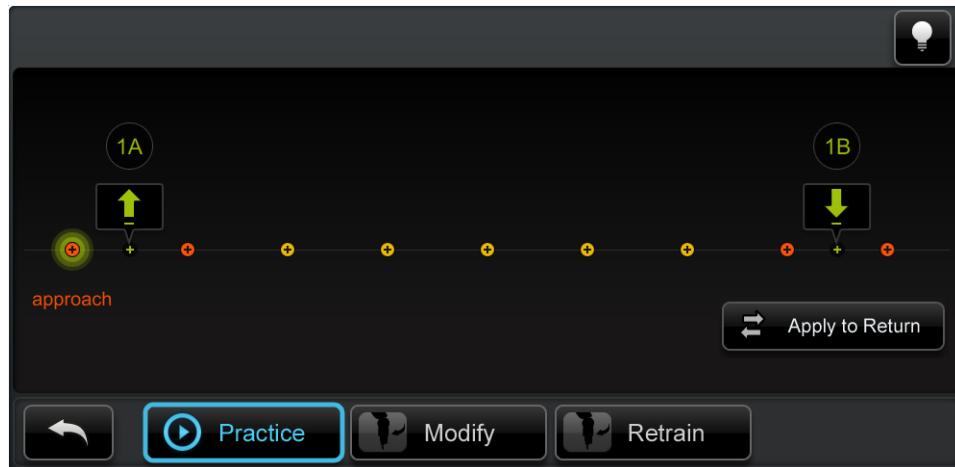
The robot displays the Task Map.





- From the Task Map, scroll through the possible paths, and press OK on the one you want to change.

The robot displays this screen. From here you can choose to practice, modify, or retrain the path.



After making changes to a custom path you can quickly apply those changes to its return path by clicking the **Apply to Return** button. This button is always available when returning to view a path.

PRACTICE



Practice gives you a run-through of the path as it exists now. You can practice at two different speeds: slow and full.

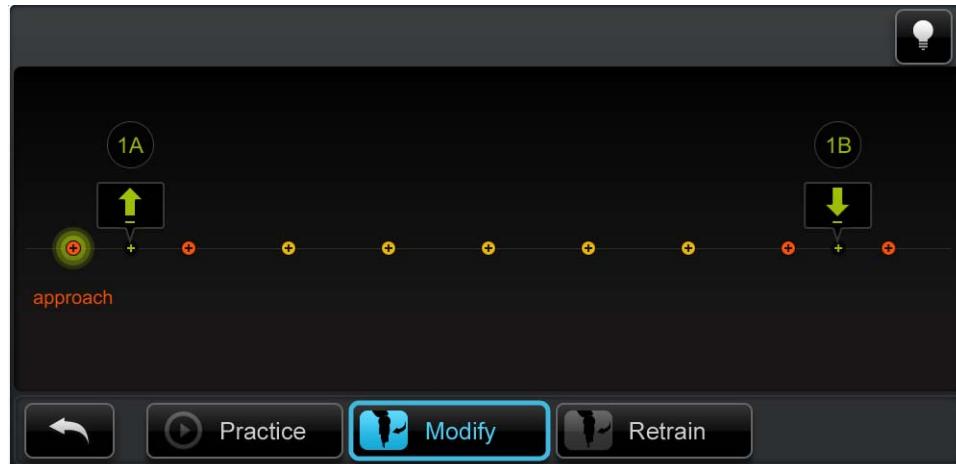
RETRAIN

Retrain enables you to quickly modify an existing path. Pressing Retrain deletes the transit waypoints on the current path, while keeping the pick and place actions and their associated approach



and retract points. (You can then add new waypoints to the path.) Retrain is also an option in the Modify Panel submenu for selected waypoints.

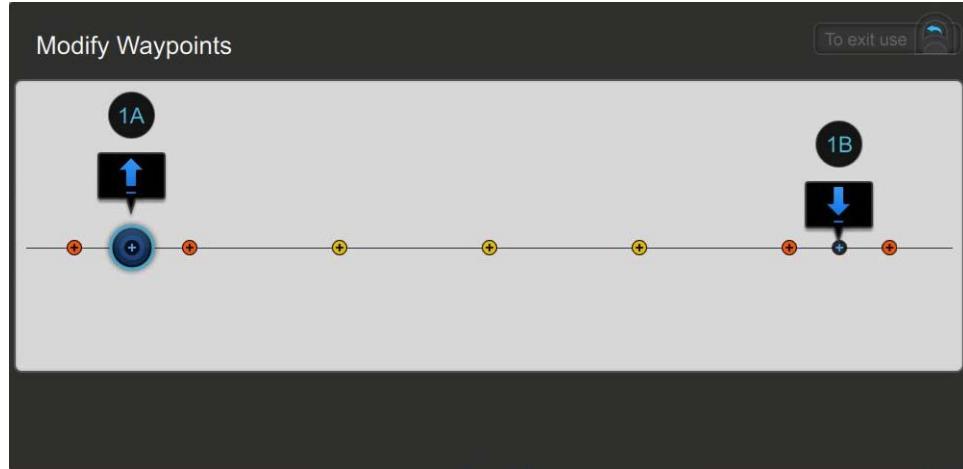
MODIFY



Modify enables you to change one point at a time, whether it's an action, a transit point, an approach or retract.

1. Select Modify.

The robot displays the Modify Waypoints panel.



Select a waypoint by scrolling to it and pressing **OK**.

The submenu changes depending on the kind of waypoint is selected and the kind of action upon which the path is based.

MODIFY PANEL SUBMENU OPTIONS

Back - Returns to the previous screen.

Remove - Deletes the selected waypoint.

Nudge - Allows for precise movement of the location and orientation of end of arm tooling. See “Nudge” on page 83.

Retrain - Enables you to quickly modify an existing path. Pressing Retrain deletes the transit waypoints on the current path, while keeping the pick and place actions and their associated approach and retract points.

Add - Adds waypoints to the path.

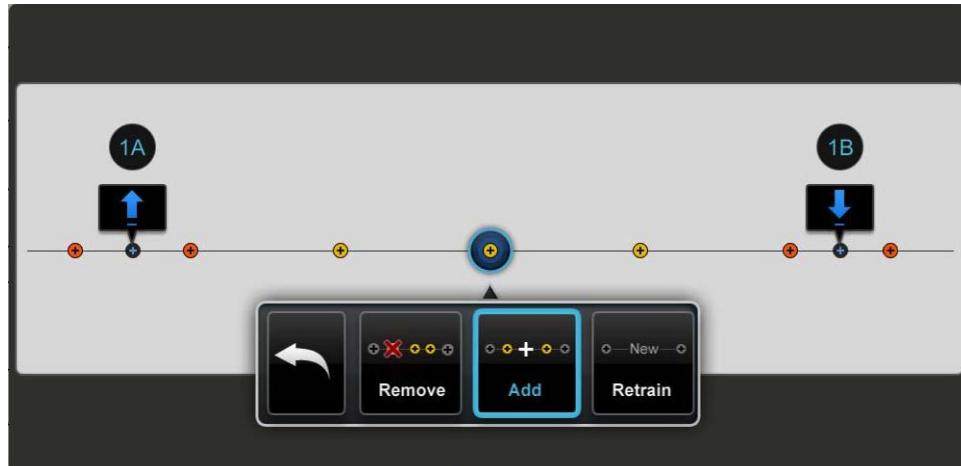
To add a waypoint:



1. Select a path from the Task Map.
2. On the screen, select Modify.

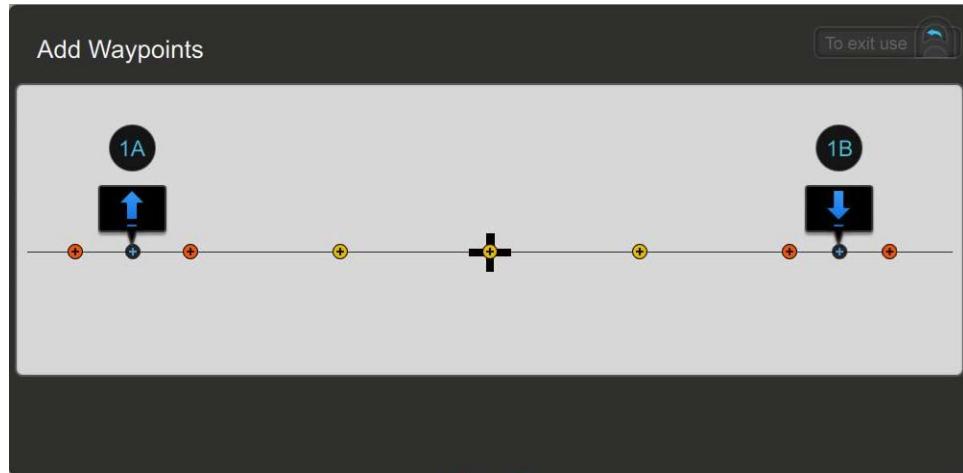
The robot displays the Move Pose Mode screen.

3. Scroll to the waypoint on the path before which you want to insert the new pose/point.
4. Press OK to display the submenu for that waypoint.



5. Select Add.

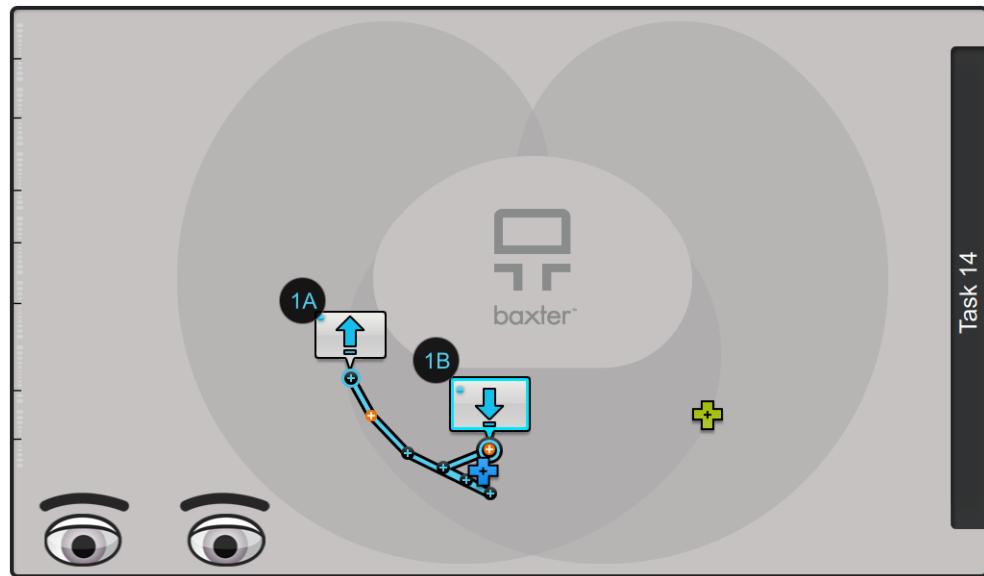
The robot displays the screen in Add Waypoints Mode.



6. Manually position the robot's arm and end of arm tooling in the pose you want to add.
7. Press the round action button on the training cuff to save the new pose.
8. Repeat for each new waypoint.
9. When done, press the Back button.



The modified path is displayed.



Suggestion: Use Practice to run through the path to check the new pose.

NOTE: You may find that some locations or poses will be invalid. That means the location or pose cannot be performed by the robot because it's near a joint limit. On the Task Map this would show up as a grayed out icon. On the screen, the robot displays the message, "Add pose failed."



Modifying Actions in Intera

Select an action on the Task Map and click **OK** to display the Modify Panel for that action.

Modifying a Pick Action

There are a number of reasons you may want to modify a Pick action, including:

- The Pick location will vary more than 0.2 in (0.5 cm).
- You want the task to include a count.
- You need to adjust the entry/exit height
- You want that part of the task to start based on a signal or send out a signal when finished.



1. Back – Close the modify screen and return to the task map.
2. Action – The action, in this example, an (off-vertical) Pick.
3. Copy – Copy the current action (and all its modified details). Creates an action group.

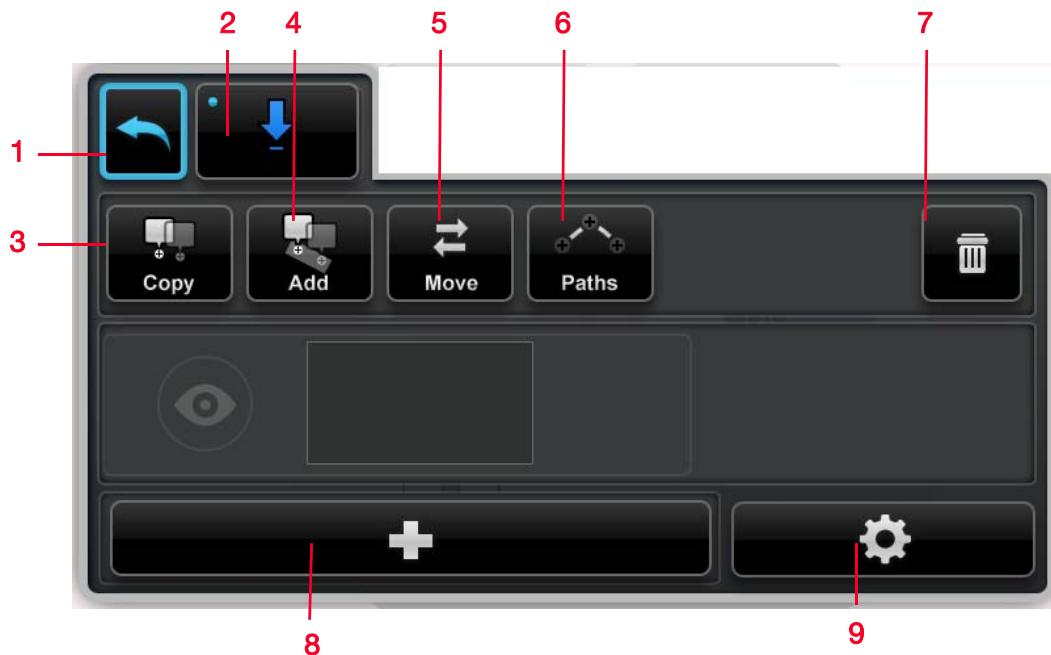


4. Add - Train an action group. Similar to Copy, but this icon enables you to add additional locations to the original, making an action group. (See “Training a Group of Pick or Place Actions” on page 60.) To add to an existing action group, choose any single action in the group, then select Add.
5. Move – Move the location where the action takes place. (Note that this will delete the path to the Action, so you may want to use Action Training and Modify to preserve the path.)
6. Paths - Select from the Task Map a path you want to modify.
7. Delete – Delete the action. You’ll be prompted to confirm the deletion. Once deleted, an action *cannot* be restored.
8. Add Features – Adjust more details:
 - Add Count – Specify the number of times an action should be completed.
 - Add Signal – Specify which defined signal(s) you want to attach.
 - Add Weight - Specify a weight for the object the robot will hold for a task.
 - Add Landmark - Create a landmark used by the Robot Positioning System. See “Robot Positioning System” on page 110.)
9. Practice or modify the approach, retract, or action poses. You can also change the distance Swayer’s arm travels as it approaches or retracts, and determine how slowly the arm should move when approaching or retracting. See “How to Modify/Practice an Action” on page 81.

Modifying a Place Action

As with Picks, there are a myriad of reasons to modify a Place, including:

- To change a Place location.
- To place a specific number of objects (count).
- To adjust the drop height.
- To switch from a drop to a place, or to change the drop height.



1. Back – Close the modify screen and return to the task map.
2. Action – The action, in this example, an (off-vertical) Place.
3. Copy – Copy the current action (and all its modified details).
4. Add - Similar to Copy, but this icon enables you to add additional locations to the original, making an action group. To add to an existing action group, choose any single action in the group, then select Add.
5. Move – Move the current action to another location. (Note that this will delete the path to the Action, so you may want to use Action Training and Modify to preserve the path.)
6. Paths - Select a path from the Task Map to modify.
7. Delete – Delete the action, and remove it from the task map. You'll be prompted to confirm the deletion. Once deleted, an action *cannot* be restored.
8. Add Features – Display a sub-menu where details of the Place can be adjusted, specifically:



- Add Count – Specify the number of times an action should be completed. See “Modifying Count” on page 81.
 - Add Signal – Specify which defined signal(s) you want to attach.
 - Add Landmark - Create a landmark used by the Robot Positioning System. (See “Robot Positioning System” on page 110.)
9. Practice or modify the approach, retract, or action poses. You can also change the distance the robot’s arm travels as it approaches or retracts, and determine how slowly the arm should move when approaching or retracting. See “How to Modify/Practice an Action” on page 81.

To move a Place location:

1. On the modify screen, click **Move**. The task map opens with a temporary “ghost” copy of the location displayed with the move icon over it.
2. Move the arm to drag the ghost to the desired new location.
Tip: Remember to maintain arm pose alignment.
3. Press the action button or the scroll knob on the Navigator.

To copy a Place location:

1. On the Modify screen, click **Copy**. The task map opens with a “temporary ghost” copy of the location displayed with the copy icon over it.
2. Reposition the arm to drag the copy to the desired location.
Tip: Remember to maintain correct arm pose alignment.
3. Press **OK** on the cuff to make the copy.
4. Repeat Steps 3 and 4 as many times as needed.
5. When you reach the final Place location, press **OK** twice. (Alternatively, press **Back** 



Modifying Count

Actions/action groups and tasks have the following counts by default:

- Pick action – unlimited count
- Place action – count of 1
- Action group (for both Pick and Place) – count of 1 for each location within the group.
- Task – unlimited count (it resets automatically)
- Subtask – no count (acts like 1)

You can modify any of these counts, as well as add count to a subtask.

To modify count to an action:

1. In the Modify screen, click the Add Features icon.
2. Click **123**.
3. Select the highlighted box, **Max Count**.
 - If the count is currently unlimited, click the box to un-check it.
 - To make the current count unlimited, click the box.
4. On the count wheel, scroll to the first number, and press **OK**. Repeat for each number, up to a maximum of 9999.
5. When all numbers are entered, on the screen, press **OK**.
6. Click **Done** in the count screen. The new count shows up in the Modify window next to the Add Features button, as well as for the action on the task map.

How to Modify/Practice an Action

After you have trained an action, you can modify its settings in the Modify Action Panel.

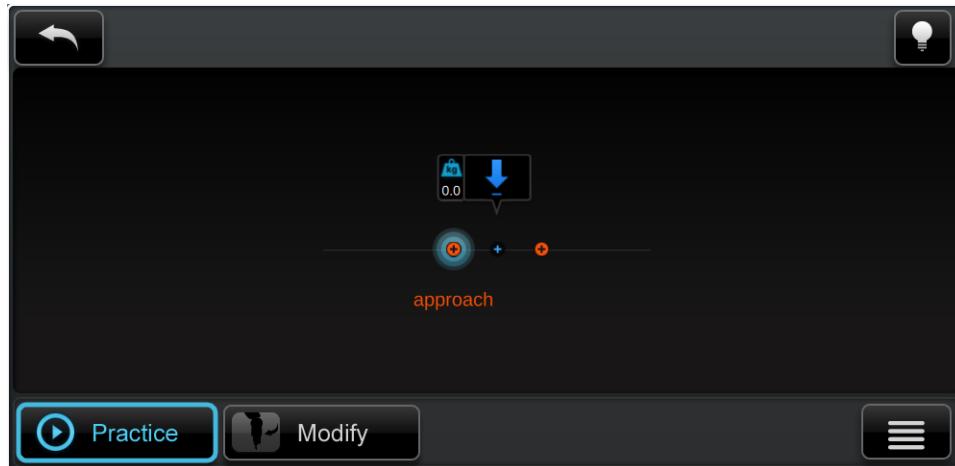
1. Select the action on the Task Map.



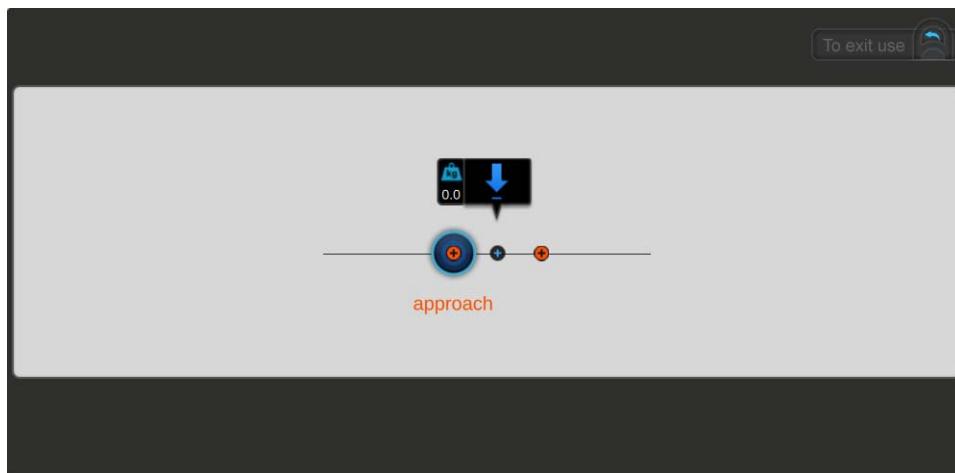
2. Scroll to the gear icon and press **OK**.



The robot displays the modify action screen.



3. Scroll to the Modify button and press it. The robot displays the modify action panel.



4. Scroll to the approach, action (in this example it's a Place), or retract point. Baxter's arm and hand move to the point as you scroll to it.
5. Grab the training cuff, adjust the pose and/or location, and press the action button on the cuff. A check mark is briefly displayed on the screen to register the new pose/location.



Note: Instead of manually adjusting the pose, you can also choose to more precisely nudge the pose after selecting the approach, action, or retract point.

6. Press the Back button when done.

You can have the robot practice the new action by pressing the **Practice** button. If you have assigned a weight to a specific action, the robot will account for the weight while practicing. You will be prompted to choose if you would like to practice with or without signals (output signals only). Practicing with signals will send the signals as it would when running the task.

Nudge

The Nudge feature enables you to finely tune the location and orientation of the tip of the end of arm tooling. The benefit: you can define precisely the position, orientation, and angle where a pick, place or hold action takes place.

You can also use this feature to nudge an action's retract and approach points if the action is non-vertical. You cannot nudge the approach and retract points of a vertical action (specifically an action whose "Snap to Vertical" checkbox is marked on the Advanced Setting screen).

HOW TO NUDGE A POSE

1. Select a pick, place, or hold action on the Task Map.

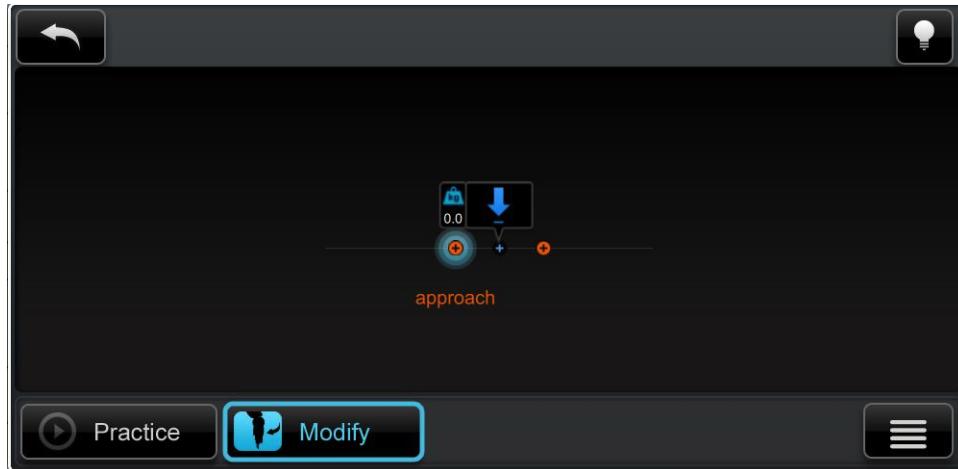
Baxter displays the action's Modify panel.



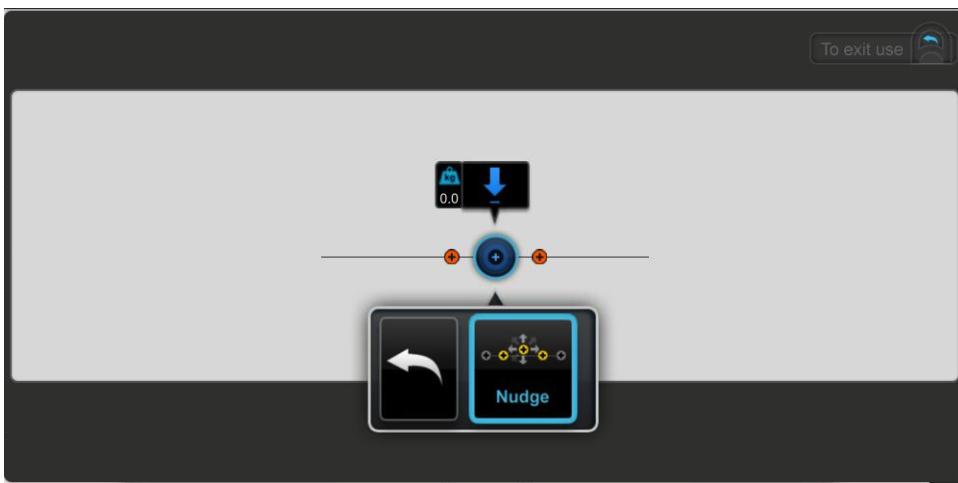
2. Select the gear icon.



The modify panel is displayed.



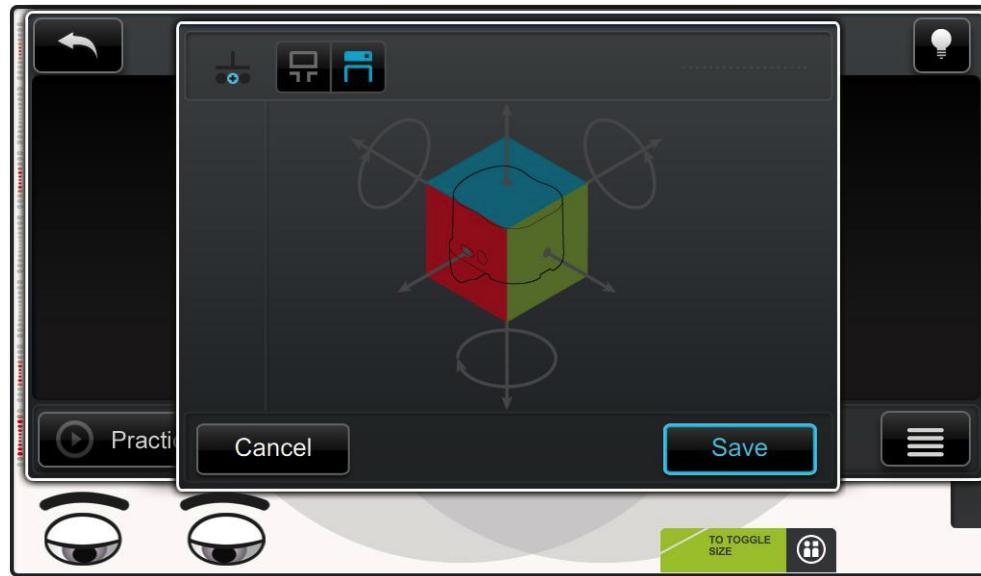
3. Click on the **Modify** button.
4. Select the specific action whose location you want to change. This can be the action itself, or its attract or retract point if the action is non-vertical. The Back and Nudge buttons are displayed.



5. Select **Nudge**.

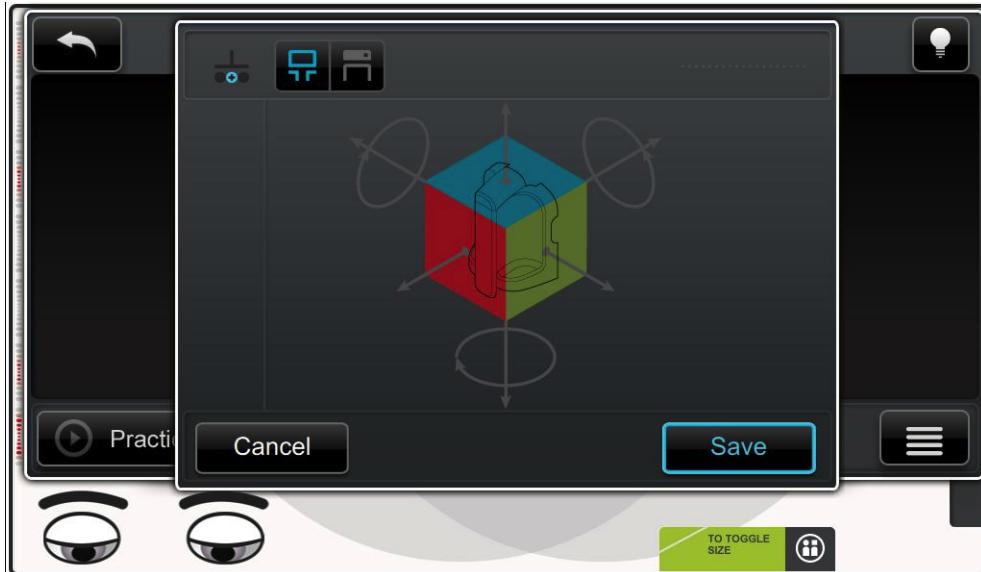


The robot moves its arm to the currently defined pose and displays the options for how the pose can be nudged.



NUUDGE IN RELATION TO CUFF OR ROBOT

You can nudge the arm in relation to the training cuff, shown in the above illustration, or in relation to the robot itself (robot base), shown below.



6. Toggle between the cuff and robot base by highlighting the robot/cuff icon (top left on screen) and pressing **OK**.

The diagram has three panes depicting the X,Y and Z axis of a particular frame. There are also corresponding arrows that represent the potential movement in millimeters along an axis as well as rotating in degrees around an axis .

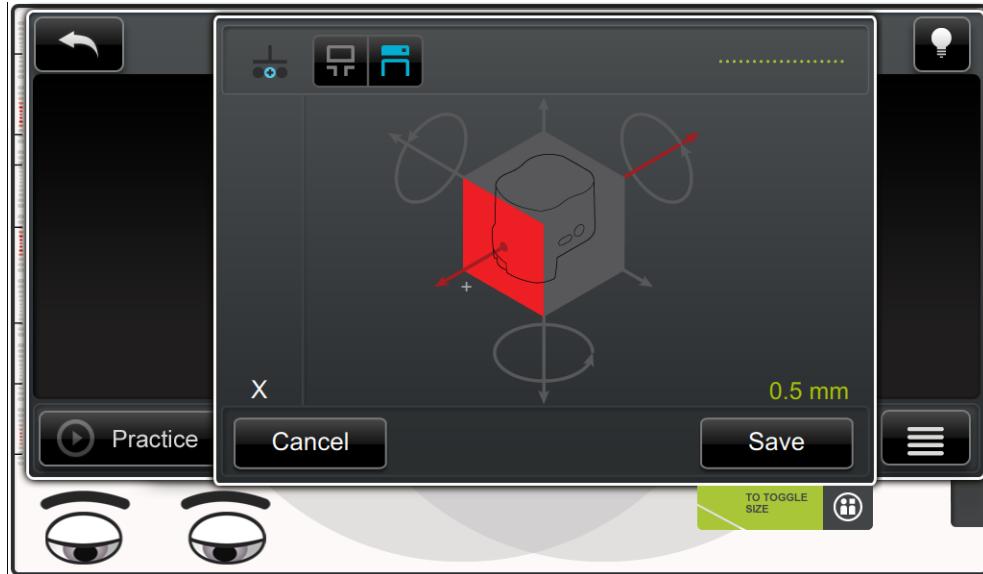


7. Scroll through the different axes and click **OK** to select the direction in which you would like to nudge.

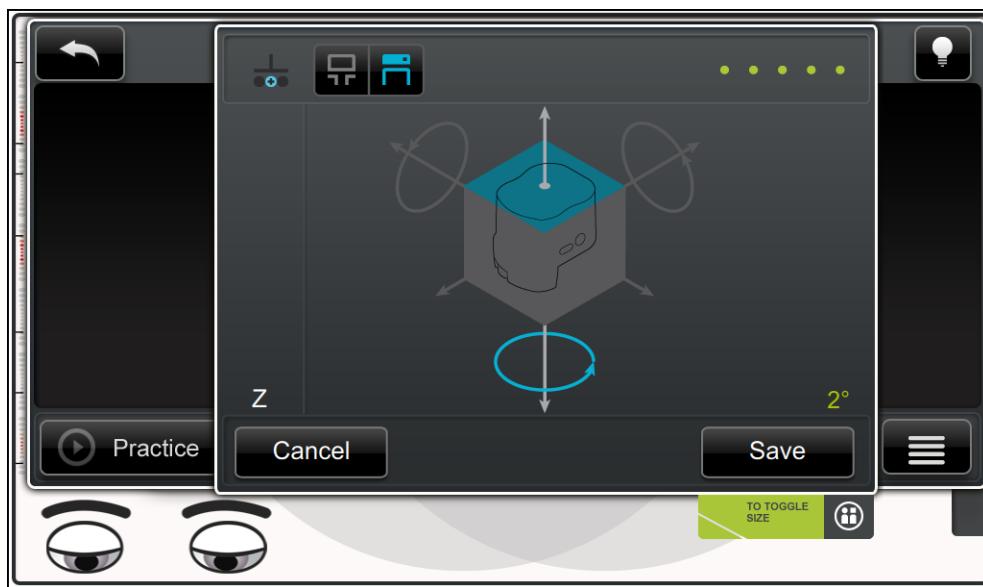
NUDGE IN SMALL OR LARGE INCREMENTS

You can nudge in any direction until you hit a joint limit. You can nudge in large (2mm or degrees) or small (.5mm or degree) increments.

8. Press the Rethink button to toggle between nudging in large and small increments.



The size of the dots in the upper right of the display change depending on your selection. The length or degree of nudge are displayed as numbers in the lower right.





9. Select from among the options, then dial a distance or degree using the scroll knob.

The robot's arm moves to the new location or angle.

10. When done, press the **Save** button.

NOTES ON NUDGE

- If you are nudging a vertical pose (specifically an action that has “Snap to Vertical” checked in the Advanced Settings window,) you cannot nudge the angle of the training cuff. This helps ensure the vertical pose does not change to a non-vertical pose.
- You should move the arm *before* you enter Nudge mode. While in Nudge mode, you can move the arm by squeezing the training cuff and moving the arm as you normally would in zero gravity. If you press the Save button after moving the arm, this becomes the updated pose from which nudges are made.
- Keep in mind that, if you nudge an Action, its Retract and Approach are *not* nudged proportionally. This is true for any Nudge, including Holds with a zero Approach/Retract distance. So, in the case of the Hold, if you nudge, make sure you go to the advanced menu and make the Approaches/Retracts zero distance.
- Pressing the Action button after manually moving the arm to a new pose does NOT save the pose.
- You cannot nudge transit waypoints.

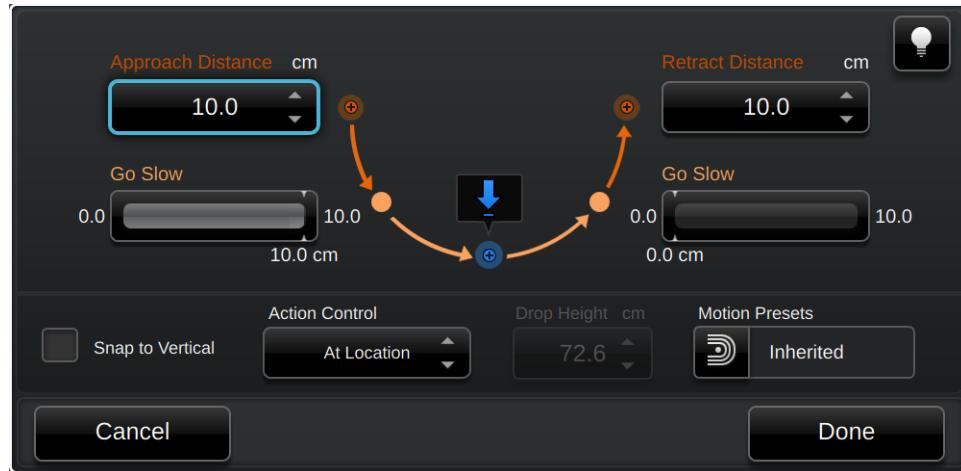
How to Make Detailed Modifications to an Action (Advanced Settings Screen)

1. From the modify action screen (such as in the figure above), press the Advanced





The robot displays the Advanced Settings Screen.



2. Make the desired adjustments to the action, described below, and press **Done** when complete.

Approach Distance – The distance the robot's arm will travel when it approaches its action, for example, a Pick. (More technically, this is the point along a vector perpendicular to the position on the cuff joint from the action's location that will move into position before moving along the vector to the action.)

Retract Distance – The distance the robot's arm will travel when it retracts from an action, for example, a Place. (More technically, this is the point along a vector perpendicular to the position on the cuff joint from the action's location that will move into position before moving along the vector to the action.)

Tip: To align an Approach, Action and Retract, change the Approach and Retract values to 0 and click Done. This aligns the three poses. Now go back to the Advanced Settings and adjust your Approach and Retract to your desired values.

There may be times when you want the Approach and Retract to move in different directions. For example, when picking up an object horizontally from a shelf, the Approach might be parallel to the table while the Retract is perpendicular to the table. In cases such as this, use modify to adjust the magnitude and direction of Approach and Retract after setting them to 0.



Go Slow Approach – You can define here a point between the Approach pose and the Action pose where the arm slows down so the Action will be more precise. This is also the point at which the end effector lines up with the Action. Therefore if "Go Slow" is turned on for the whole Approach, the orientation of the end effector doesn't matter; it aligns with the Action at the very beginning of the Approach (or with the Retract, in the case of "Go Slow Retract.") On the other hand, if "Go Slow" is turned off for the whole Approach, the end effector will gradually change its orientation until it gets to the Action.

Go Slow Retract – You can define here a point between the Action pose and Retract pose where the arm finishes going slow because it is safely away from performing the Action. This is also the point at which the end effector lines up with the Action. Therefore if "Go Slow" is turned on for the whole Retract, the orientation of the end effector doesn't matter; it aligns with the Action at the very beginning of the Retract. On the other hand, if "Go Slow" is turned off for the whole Retract, the end effector will gradually change its orientation as it retracts.

Snap to Vertical – Check this box if you want Actions, Approaches, and Retracts that are performed within 5 degrees of vertical to become truly vertical. This could be useful when packing a multi-layer box or when inserting a peg into a hole. (Note that the robot's Approach, Action, and Retract poses must be aligned and within 5 degrees of vertical for this function to be enabled.) If you want to preserve a slightly off-vertical action, make sure this box is unchecked. You can toggle the checkbox, as long as your pose (Action, Approach or Retract) is within 10 degrees of vertical.

Note: See the tip above, under Retract Distance, to learn how to use the Advanced Settings Screen to align the Approach, Action and Retract poses.

Non-vertical actions are identified by a "dot" symbol. These symbols show up on the Task Map when an action is highlighted or on the action's modify panel.



Examples of non-vertical (left) and vertical (right) place action icons



Action Control Settings – Select here to control when the robot will perform an action. The options are:

- At Location – The robot will grip or release a part when it arrives at the trained position, not when it feels force from an object or achieves a vacuum seal.
- First – The robot will grip or release a part when it either feels contact (if it's using an Electric Parallel Gripper), when it senses a vacuum seal has been achieved (Vacuum Cup Gripper), or when it arrives at the trained position, whichever happens first.
 - The robot will only travel 2cm beyond the end point if it doesn't sense contact or reach the trained position. This is intentional and is meant to ensure the robot doesn't inadvertently collide with other parts of the workspace.
- Contact (Electric Parallel Gripper only) – The robot will grip or release the part when it feels force from the part or work surface.
 - The robot will continue on its approach vector until it comes in contact with a part and then grasp. This setting is helpful when creating stacks.
- Sensed (Vacuum Gripper, Pick Action only) – The robot will grip the part when it senses a vacuum seal.
 - The robot will continue on its approach vector until it senses a vacuum seal on the part, and then grasp. This setting is helpful when creating stacks.
- Mixed – When modifying an Action Group, if the Action Group's individual locations have a mix of other settings -- for example the first place is Contact and the second place is At Location -- you will see Mixed as the setting.

Note: You can change Mixed to another setting, but you cannot change another setting to Mixed. In other words, if you change Mixed to another setting, for example At Location, all of the actions in that group will Pick At Location, but you cannot change from another setting to Mixed since you cannot specify which settings will be assigned to which actions in the group.



Default Action Control Settings - Pick:

- Electric Parallel Gripper - At Location
- Vacuum Gripper - First

Default Action Control Settings - Place:

- Electric Parallel Gripper - At Location
- Vacuum Gripper - At Location

Drop Height – Define here the height above zero at which the robot will perform a Place. For vertical Place actions only. (The zero point on the robot is where the gray lower front panel meets the black metal that connects to the robot's pedestal.)

Motion Presets - Motion presets define how the robot's arm follows a path's waypoints. "Inherited" means the preset is inherited from the task level. See "Motion Presets" on page 91.



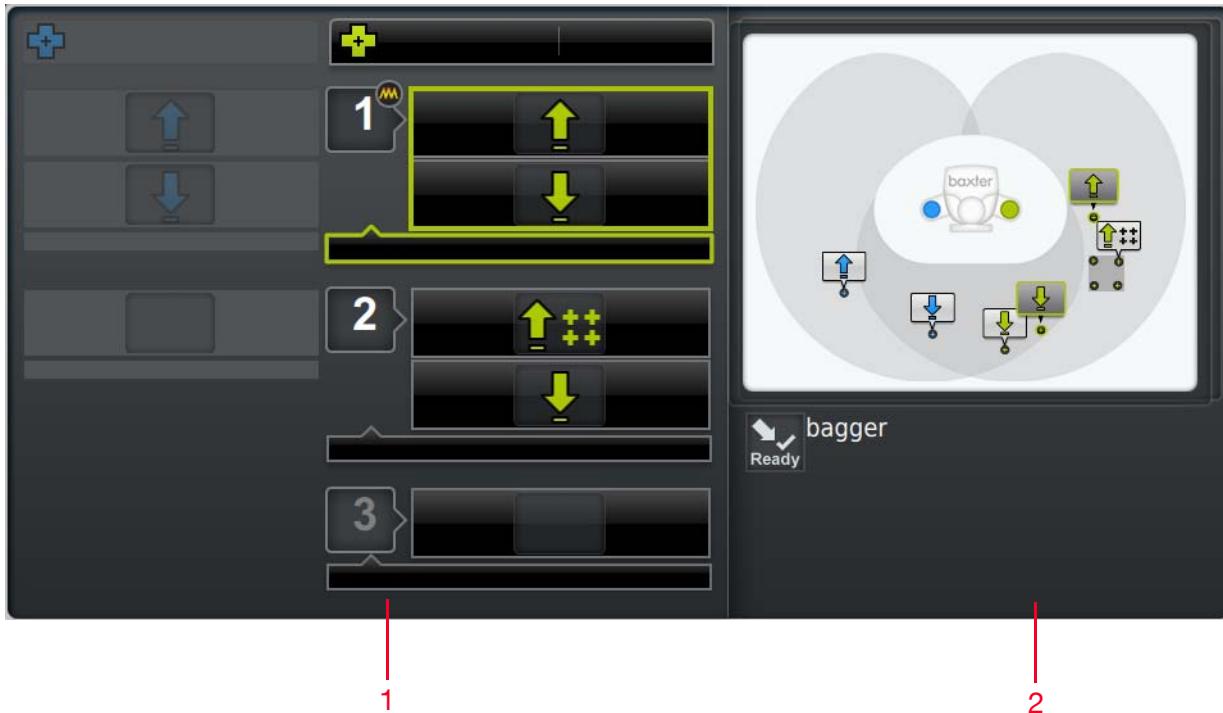
Note: When you see a "light bulb" symbol on the Advanced screen, that means there is a tip (or tips) on how to use that functionality. Select the light bulb to display the tip.



Managing Tasks and Subtasks

Use the Task Order screen to view the current subtask order, and rearrange and combine subtasks. You can also use this screen to manage task settings and add counts or signals to a task or sub-task.

To open the Task Order screen, in the task map, press the **Rethink** button, and click **Order**.

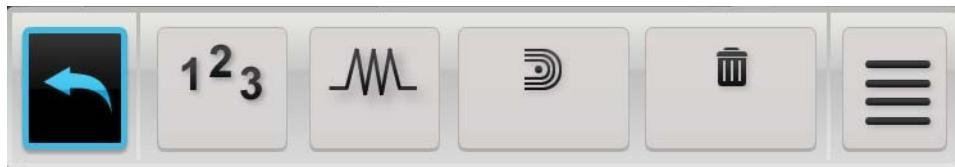


1. Displays a visual list of all subtasks (numbered in order of execution), and the corresponding Pick, Place, and/or Hold actions (including action groups) within each subtask.
2. Displays the task map for the selected task, and below it, any signals that correspond to the entire task or subtask. In this example, you can see a ready signal.

To make changes to the whole task, select the bar at the top. To make changes to any subtask, select the subtask and press **OK**. The button bar for the task or subtask appears.



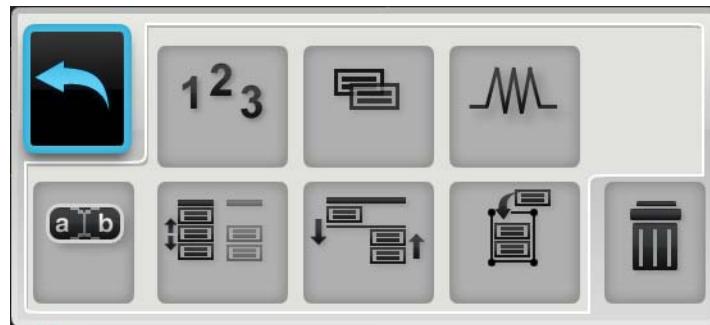
Task Button Bar



- Back – Close the button bar and return focus to the task order screen.
- Count – Change count to a task.
- Signals – Add a signal to a subtask, or remove one. (See “Working with Signals” on page 128.)
- Motion Presets - Define how the arm follows a path's waypoints. These presets will help the robot complete a wider variety of tasks. (See “Motion Presets” on page 91.)
- Clear Actions – Delete all the subtasks. This helps you quickly retrain a task by allowing you to reset the subtasks while not affecting any signals, task settings, etc. you have already set up. While actions are cleared, the following are retained: signals and landmarks set up for the task; task settings (e.g., end effector settings, camera settings (see “Camera Settings” on page 102), arm reset behavior, drops/missed picks count); motion presets, task-level counts and signals.
- More Options – Access end effector settings (see “Changing Gripper Parameters” on page 99); camera settings; and adjust the number of times the robot will attempt Picks (see “Change Number of Attempts to Pick an Object” on page 103).



Subtask Button Bar



Top row, left to right:

- Back – Close the button bar and return focus to the task order screen.
- Count – Add count to a subtask.
- Copy – Copy a subtask and all its details to the same location on the Task Map. The copied subtask will appear on the Task Map just below the original.
- Signals – Add a signal to a subtask, or remove one. (See “Signals” on page 89.)

Second row, left to right:

- Rename - Create or change the name of the selected sub-task.
- Reorder – Reorder sub-tasks for the selected arm. You can also merge incomplete sub-tasks.
- Coordinate Order (Across Arms) – Shift when a sub-task begins on one arm in relation to a sub-task on the other arm. If the sub-task is already coordinated, this button becomes “uncoordinate.”
- Combine – Combine two sub-tasks on one arm into one sub-task. The highlighted sub-task moves into the sub-task below it.
- Delete – Delete the current sub-task on an arm.

TO NAME/RENAME A SUBTASK:

1. From the Order screen, highlight the subtask you want to name and press **OK**.
2. Select “Rename” from the subtask menu.
3. Use the text wheel to create a name for the selected subtask.
4. Select **OK**.

You can also modify or delete subtask names.



COMBINING SUBTASKS

Combining subtasks lets the robot decide when to complete part of a task. For example, if three parts need to go onto a tray, but the order does not matter, you can combine each subtask into one large subtask, and the robot will determine in which order to complete the subtasks.

1. Select the first subtask of the two.
2. In the subtask order button bar, click the combine icon. The selected subtask merges with the one below it.

Note: Note: It is not possible to split combined subtasks.

TO REORDER SUBTASKS:

1. Select the reorder icon in the subtask button bar.
2. Scroll up or down. A yellow line appears to indicate the new location.
3. Press **OK** to move or merge the subtask.

Coordinating Sub-tasks Across Arms

Coordinating arms allows Baxter to conduct a sub-task on one arm before conducting a sub-task on the other arm.

1. Pick the sub-task you want to move relative to the other arm. Press **OK**.



2. Click the coordinate icon in the sub-task order button bar. A yellow line appears in the list of sub-tasks on the other arm (In this example, the user wants to move the second sub-task on the left [green] arm.).



3. Scroll to the right or left to move the yellow line until it appears after the sub-task that the selected sub-task should follow.



4. Press OK.



After the sub-tasks are coordinated, a gray line appears between them to indicate the order.



To uncoordinate sub-tasks across arms, highlight the sub-task used to coordinate arms and scroll to the coordinate option. It will now read "uncoordinate". Select it and that coordination will be removed.

Changing Gripper Parameters

Apply custom gripper settings to optimize gripper performance with a wide variety of objects.

Note: These settings apply to both Rethink Robotics grippers and custom grippers, and are applied for all actions on an arm for that task.

To change a setting:

1. Create a task.
2. Go to the Task Order screen.



3. Select the task bar at the top of the screen, and press OK.



4. Click the More Options  icon.
5. Click the End Effector Settings icon.



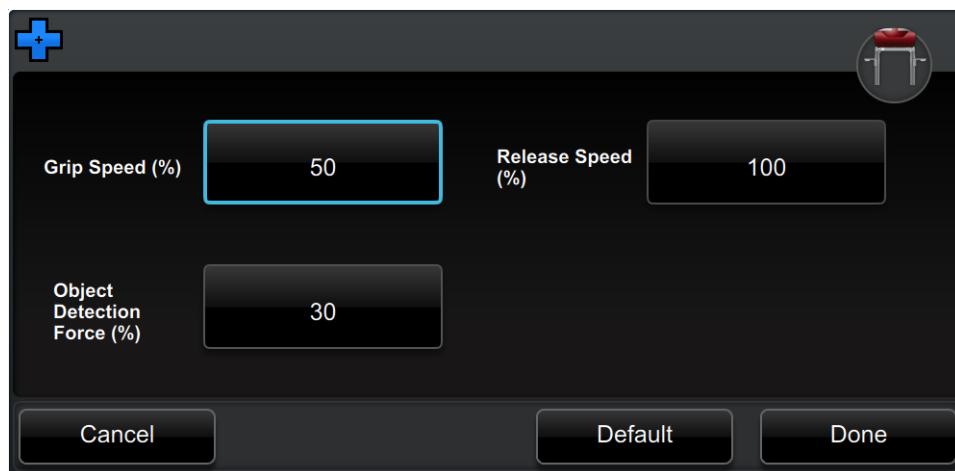
6. Change the setting.



ELECTRIC PARALLEL GRIPPER SETTINGS:

- Grip Speed (%): Gripper closing speed as a percentage of maximum.
- Release Speed (%): Gripper opening speed as a percentage of maximum.
- Object Detection Force (%): Threshold for object detection—the amount of force required to detect when an object is grasped before the robot moves to the next action.

Note: If the force is set too high, a soft object in the gripper may not be detected. Set high initially, and back it off if the gripper is crushing things.



VACUUM CUP GRIPPER SETTINGS:

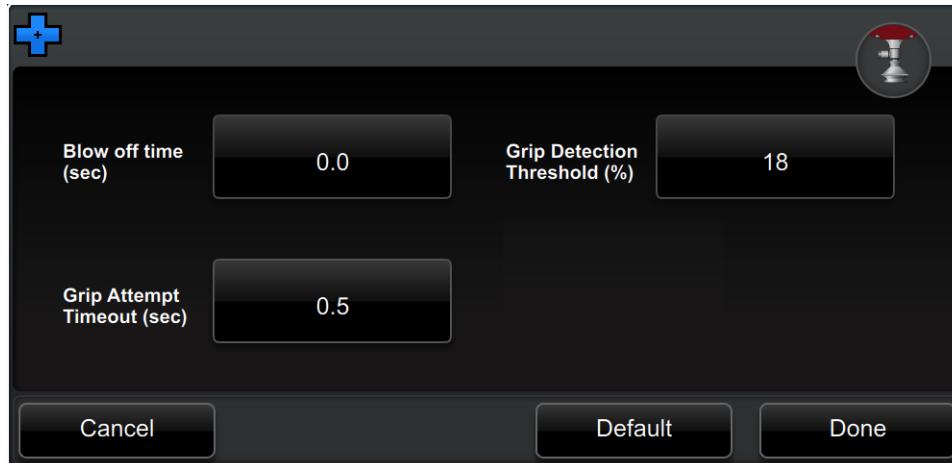
- Blow off time (sec): Amount of time the vacuum gripper will blow air when releasing an object. (Note that this feature requires additional “plumbing.”)
- Grip Detection Threshold (%): Threshold in which the vacuum sensor detects a successful pick. This setting is intended for custom vacuum cup grippers and/or 3rd party vacuum cups, and is dependent on the mechanical properties of the vacuum cup mechanism.

Examples:

- For a single piece of paper, which is likely porous, set a lower threshold. That tells the robot the material is porous and the gripper sensor will trigger at a lower threshold.
- For a rigid non-porous object, less of a vacuum is required, so set a higher threshold. Setting a higher threshold ensures the object in the gripper does not trigger prematurely.



- When using a custom vacuum gripper, it is a best practice to start with a higher percentage, and reduce the value as needed.
- Grip Attempt Timeout (sec): Duration in which a vacuum end effector attempts to grip an object before stopping the attempt.



Camera Settings

You can change the settings for the robot's cameras to adjust for the amount and quality of light the robot perceives in the work area. (Technical note: the adjustments apply primarily to the lighting variables of gain, white balance, and exposure.) You can make adjustments using a simple slider or you can let the robot adjust its own camera lighting settings automatically by selecting the Auto checkbox. This capability is available per task.

Note: When you create a new task, the current camera settings are inherited based on the previous open task.

HOW TO CHANGE THE LIGHTING SETTINGS FOR THE ROBOT'S CAMERAS

1. On the Task Order screen, highlight a task and press OK to display the task bar buttons.



1. Select the More Options icon
2. Select the camera settings button



There is a slider underneath a live camera image. You can scroll among the slider, the “Auto” option, the Back button, and the Done button. (If “Auto” is active, the slider is not needed, therefore it’s not highlighted.)

3. Press OK to activate the slider.
4. Scroll to move the slider left or right.

Scrolling left reduces the amount of light the camera allows in. Scrolling to the right lets in more light. (This is also known as decreasing or increasing the gain, respectively.)

5. When you’re satisfied with the appearance of the live image, press **Done**.

Change Number of Attempts to Pick an Object

Baxter attempts to pick up a specific object by default twice. (If it misses the first time, it will try again, and if misses a second time, Baxter will stop.) You can change this setting to be any number between 1 and 99.

This setting also applies to parts being knocked out of Baxter's hand, for example.

This setting is changed at the task level, not the action level.

To change this setting:



1. With the task highlighted on the Task Order screen, select the advanced settings icon.



2. Scroll to **Attempts** on the resulting menu and select it.





The default is 2 attempts.

3. Highlight the spin box.
4. Change the number of attempts to your preferred number. (A check box also gives you the option of stopping Baxter immediately if the object is dropped.)
5. Press the **Return** button on the to make the change for that task.

Note: Although you can change this setting on either arm, the setting applies to both arms. You cannot assign different settings to each arm. Further, the setting applies to both drops and misses. They are counted separately, but the number to apply is specified once.

Motion Presets

Motion presets define how the robot's arm follows a path's waypoints. These presets will help the robot complete a wider variety of tasks.

There are four types of presets:

- **Fast** ("fast, direct") - The default preset. This is the way the robot has followed waypoints along its path in the past. (If you upgrade a task from a previous version of the software, this is the preset that will be selected.)
- **Moderate** ("moderate, deliberate") - The robot will follow the waypoints closely, and in a controlled manner (that is, more closely and controlled than either the Fast or Express presets.)
- **Express** ("fast, most direct") - The robot will use the path's waypoints more as a guide to follow rather than as a specific target to hit. With this preset, the robot is optimized to complete its actions faster than the other presets. Use this preset when cycle time is paramount and path precision is not critical.
- **Slow** ("slow, most deliberate") - Allows you to run the whole task, or individual actions, at a very slow speed to ensure the robot is behaving the way you expect before running the task at its desired final speed.

You set presets at the task level, as you do for signals, counts, and advanced settings, but you also have the option of overriding the preset at the action level (i.e., pick, place, or hold).

How to Change Motion Presets at the Task Level

1. From the Task Order screen, highlight the task.



2. Press **OK** to display the menu button bar.



3. Select the Motion presets icon.

The current or default motion preset is displayed, for example:



4. Scroll through the motion presets options and select a preset.
5. Press **OK**.

How to Change the Motion Preset at the Action Level

1. Highlight the pick, place, or hold action on the Task Map and press **OK**.
2. On the modify action screen, select the icon.
3. Select the Advanced icon on the modify screen to display the advanced action modification screen.



The default setting for the action is "Inherited." That means the setting is the same as, or inherited, from the task level.

4. Change the motion preset for the individual action by selecting the Motion Presets icon.
5. Scroll through the options.
6. Press **OK** to select one of the presets, then press **Done** to save it.

IMPORTANT NOTE ABOUT MOTION PRESETS

The destination action -- the action to which the arm is moving along the path -- is where you apply the motion preset. For example, if you need the robot's arm to move in a controlled, deliberate fashion from a Hold to a Pick, you would open the advanced screen for the Pick and choose Explicit.

Specify Object and Custom Gripper Weights

To achieve optimized arm performance, you can specify the weight of the object the robot will handle as well as the weight of custom grippers you may use. This feature makes the robot capable of recognizing weight – and adjusting to it – so arm movements will be smoother and more efficient.



For instance, if you input a payload of three pounds (1.4 kg) for the gripper weight, the robot will recognize that weight, so when you squeeze the training cuff to move the robot's arm, it will feel essentially weightless and performance may also improve. (Note, however, that if you adjust the weight of a custom gripper and do not specify that change, the robot's arm performance will likely suffer.)

The robot can support up to five pounds (2.2 kg) on each of its end arms. An end effector is made up of the plate and its attachments (electronic gripper or vacuum, fingers).

Remember to account for the weight and length of the adapter plate when configuring custom grippers.

NOTE: The five-pound limit includes both the weight of the objects the robot will be handling and the weight of the end effector itself, so keep that in mind when specifying total gripper weight.

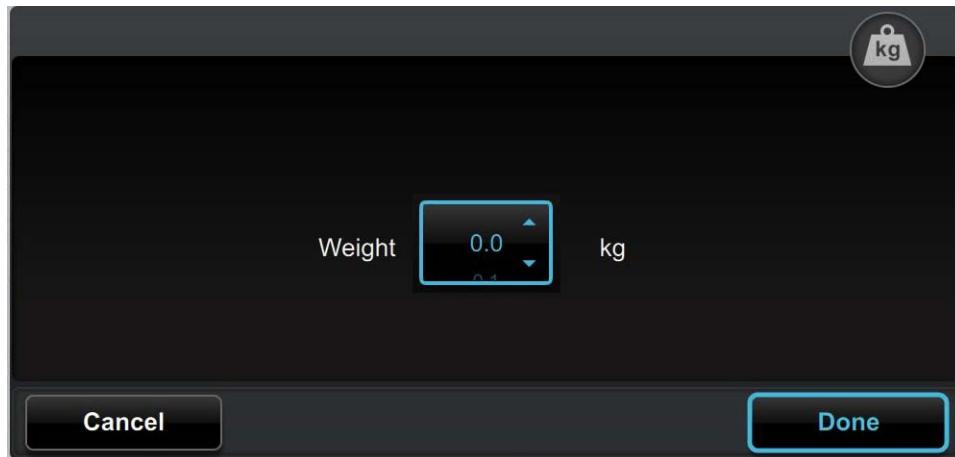
How to Specify Object Weight

1. Select an action or action group on the Task Map and press **OK** to display the Modify Task Window.
2. Select the Add Features + button and press **OK** to display the task button bar.





3. Highlight the Weight icon and press OK to display the weight entry screen.



4. Select the spin box, enter the desired weight and press OK.
5. Press Done to confirm the weight.
6. Press the Back button on the once to return to the Task Modify screen.

The weight appears in the Modify window with its own Added Features button.

7. Press the Back button again to return to the Task Map.



Robot Positioning System

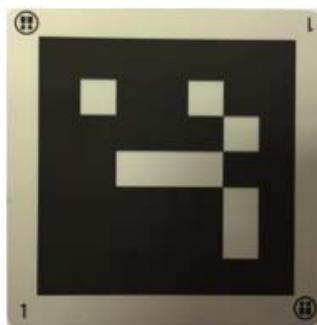
Introduction

Collaborative robots are designed to work next to humans and to be deployed quickly from one task or work cell to another. This design is a beneficial one but it also introduces a whole new set of challenges for robotics. For example, moving robots from one work cell to another means the locations of Pick, Place, and Hold actions for a task must be adjusted in the new work area.

And when human operators accidentally bump into tables, conveyors, or the robot itself, the locations where the robot expected to perform those actions get moved. That would make the robot fail to perform the tasks.

For these reasons we have designed the Robot Positioning System feature. In a new workflow using this feature, an operator can quickly reregister the robot to the workspace and continue working without having to retrain a task or spend time trying to fine-tune a previously trained task.

Please keep in mind that at present the Robot Positioning System only works while setting up the task; it does not dynamically adjust as the robot performs the task. Also, the Robot Positioning System is not used to load tasks or make decisions on what action to perform.



Sample landmark



A landmark is a design or mark placed in the field of view of an imaging system to be used as a point of reference. It's similar in some ways to a registration mark used in printing to keep the different images aligned.

Rethink Robotics currently supplies landmarks on anodized aluminum with an adhesive backing. These landmarks are placed on the surface of modules in the work area.

If Baxter -- or the surface containing the landmark on which an action takes place -- moves, either accidentally or not, you can use the Robot Positioning System feature to re-register the action in relation to the original location and quickly reorient Baxter in relation to the task.

Please note this feature is designed for relatively small changes in location. The movement between the original location and the re-registered one should not be more than a distance +/- 50mm and/or a rotation of 10 degrees.

IMPORTANT:

- When re-registering to a previously associated landmark, it's very important that you do not move the arm from the original location. Doing this will introduce error into the associated actions. Therefore, we recommend pressing the OK button on the other arm or Baxter's torso.
- For certain tasks you may need to mechanically fix Landmarks to the workspace. For example: tasks where the surface is particularly hot, which would affect the adhesive on the landmark, allowing it to drift.

When to Use the Robot Positioning System

This feature is useful whenever you anticipate that the location of an action (usually a Pick or Place) may change and you want to quickly reregister Baxter to the new location.

The change in location could be caused by:

- rolling Baxter away for routine calibration and then moving the robot back into the same work cell.
- operator error, for example, accidentally bumping a table.
- planned moves of the robot and setup, for example, when moving an entire work cell to a new location on the factory floor.



IMPORTANT:

- Assign every action in a task to a landmark.
- Use unique Landmark numbers for each module and for each arm.
- Do not reuse Landmark numbers in the same task.

Tip: When precision is required, move the landmark as close as possible to the location where the action takes place.

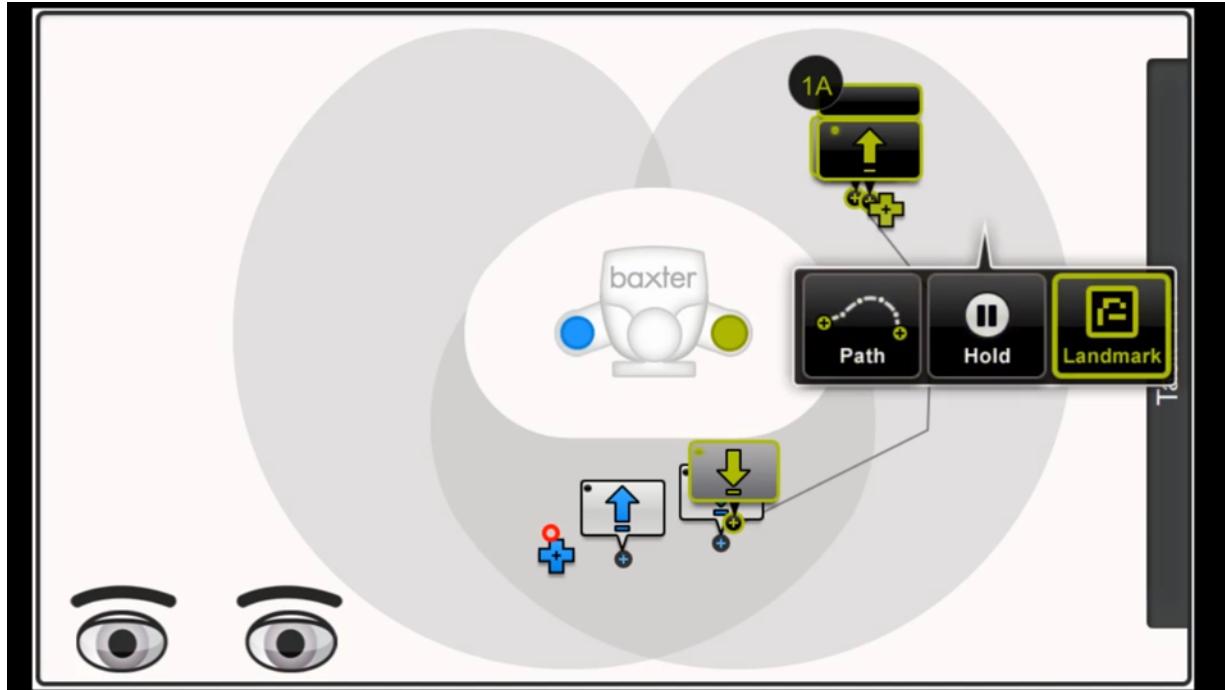
How to Create a Landmark

IMPORTANT:

- Make sure there are no obstructions between the end effector and the landmark.
- Make sure Landmarks are placed on a flat, horizontal, leveled surface and that the surface is properly cleaned before placing the landmark.
- The further away a landmark is from the actions, the less precise it might be when reregistering the robot. We recommend landmarks be no further than 50 cm away from the actions associated with them.
- To allow Baxter's imaging system to see the landmark, position the end effector camera approximately 20cm above the landmark, with the landmark in the center of the field of vision. You may also need to ensure there is sufficient light by adjusting the gain.

To create a Landmark with the Action Button:

1. In the Task Map, press the action button on the training cuff and select **Landmark** from the submenu.

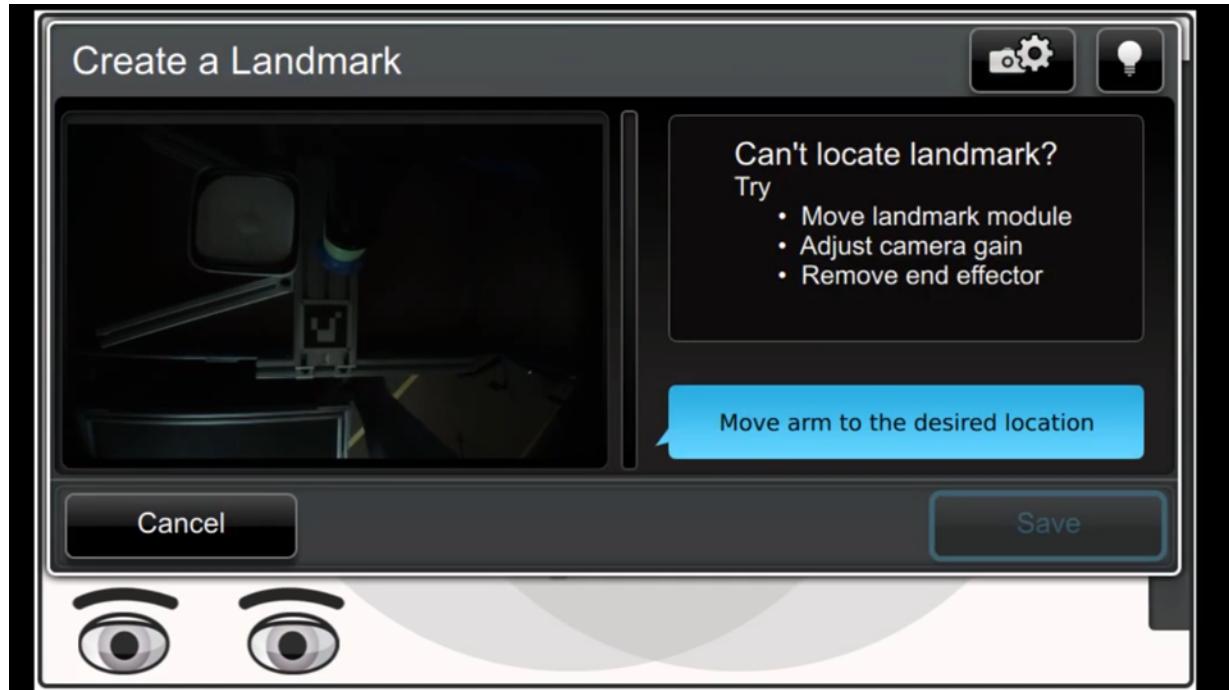


2. Position Baxter's arm approximately 20cm above the landmark, with the landmark centered in the camera's display.

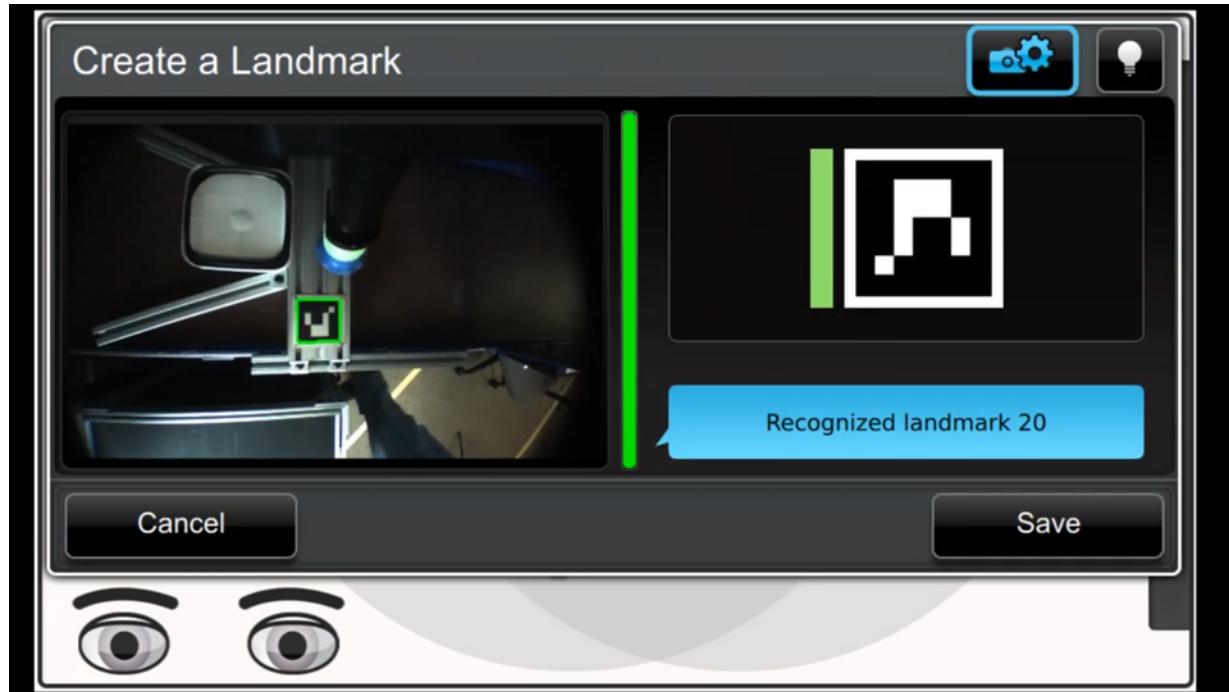
Baxter should recognize the landmark and highlight it with a green outline.

If Baxter cannot locate the landmark, as in the example below, do one of the following:

- make sure the camera is 20cm above the landmark.
- adjust the gain to make the image brighter or darker.
- move the landmark module so that it's closer to the center of the camera's point of view.
- remove the end effector.

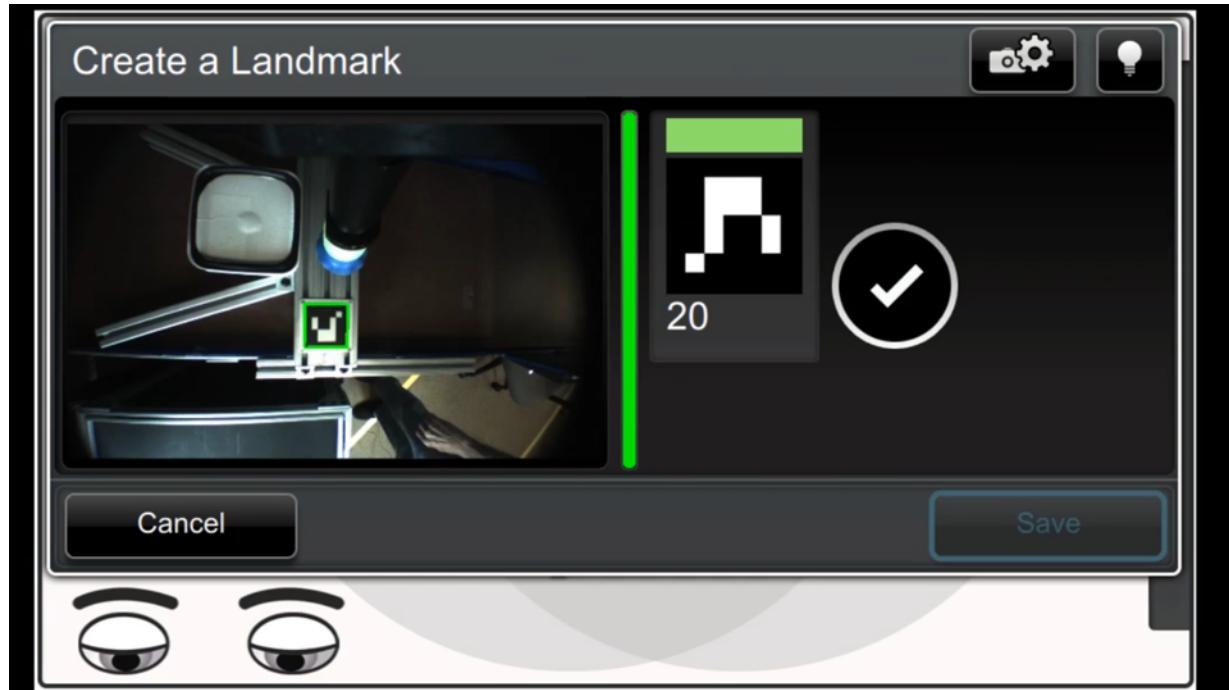


The strength of the recognition is displayed by the height and brightness of the vertical green bar in the middle of the screen: taller and brighter = stronger recognition. The landmark is recognized when it is outlined in green and a message displays with the landmark's number, as in the example below.



3. Save the landmark by selecting **Save**, but, to avoid moving the camera, press the **OK** button on the torso or the other arm.

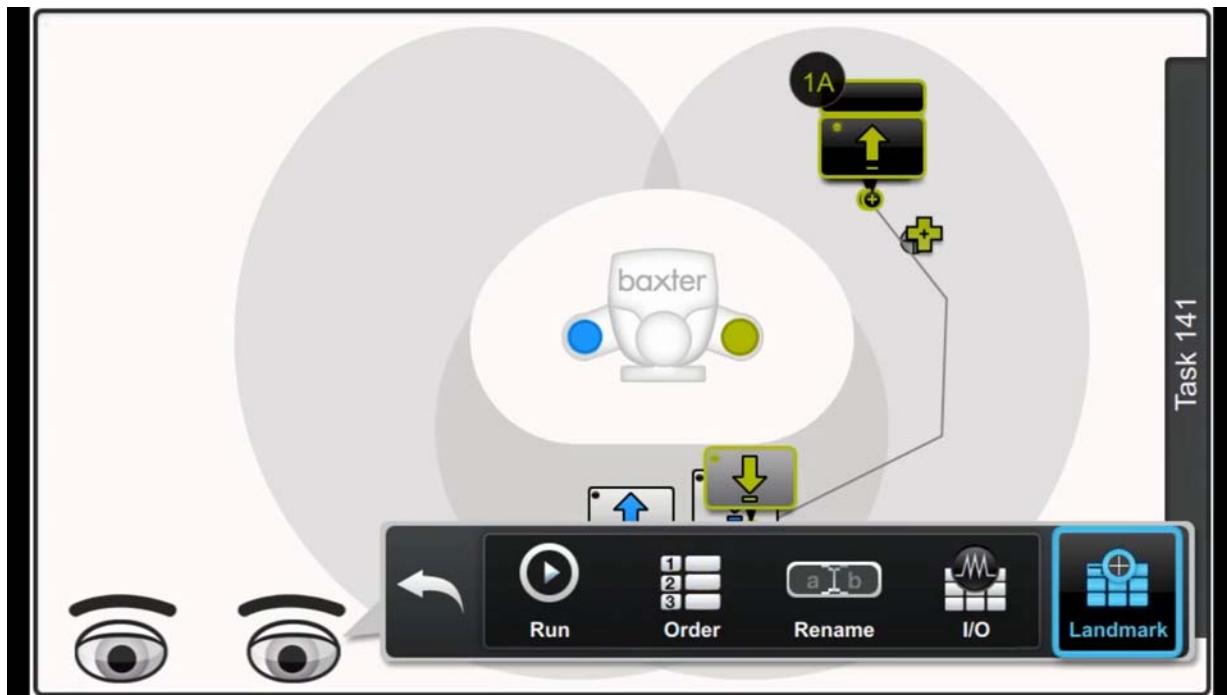
When the landmark is saved, the screen displays a checkmark and the number of the landmark, as in the example below.



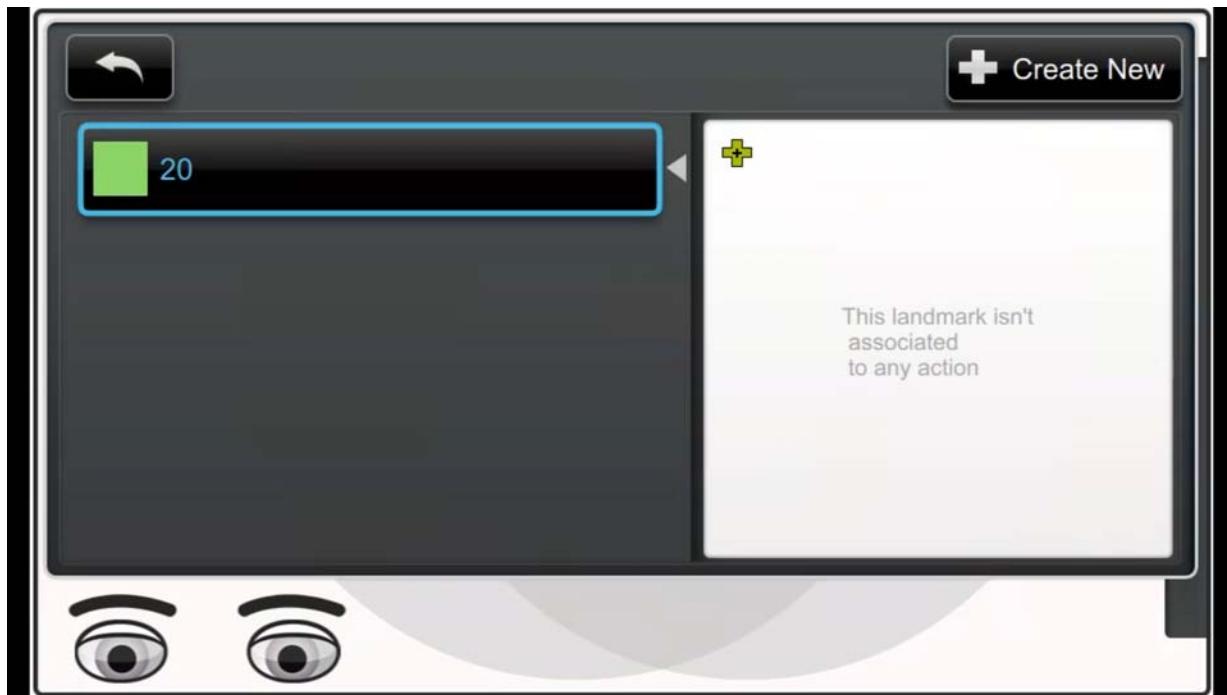
Alternate Way to Create a Landmark

You can also create a landmark this way:

1. Press the **Rethink** button.
2. Select the landmark gallery icon.



3. Select **Create New**.



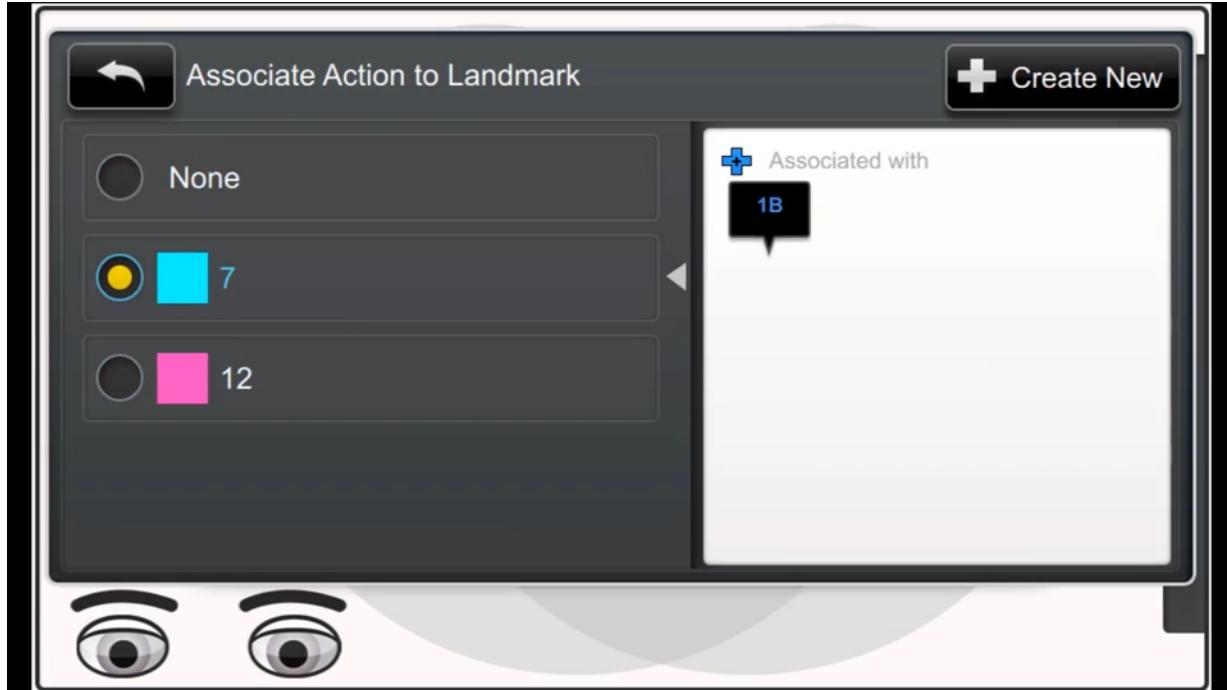
That displays the Create a Landmark screen, as seen previously.

4. Save the landmark as before, using the navigator button on the torso or other arm.

How to Associate a Landmark to an Action

1. Select the action on the Task Map.
2. Select
3. Select

The Associate Action to Landmark screen displays.



4. Select the landmark for the action.

In this example, landmark 7 is associated with Pick 1B.

5. Press OK.

NOTE: You can also press the Rethink button and the landmark icon to display the landmark gallery, where you can associate multiple actions to a single Landmark. This is a more efficient approach when there are multiple actions that need to be associated to Landmarks.

How to Reregister a Landmark

After one or more of the action locations has been moved -- whether because the robot has been moved, a fixture has been inadvertently bumped, Baxter has been inadvertently bumped, etc. -- it's easy to reregister the actions using the Robot Positioning System rather than retrain them.



Note: When re-registering to a previously associated landmark, it's very important that you do not move the arm from the original location. Doing this will introduce error into the associated actions.

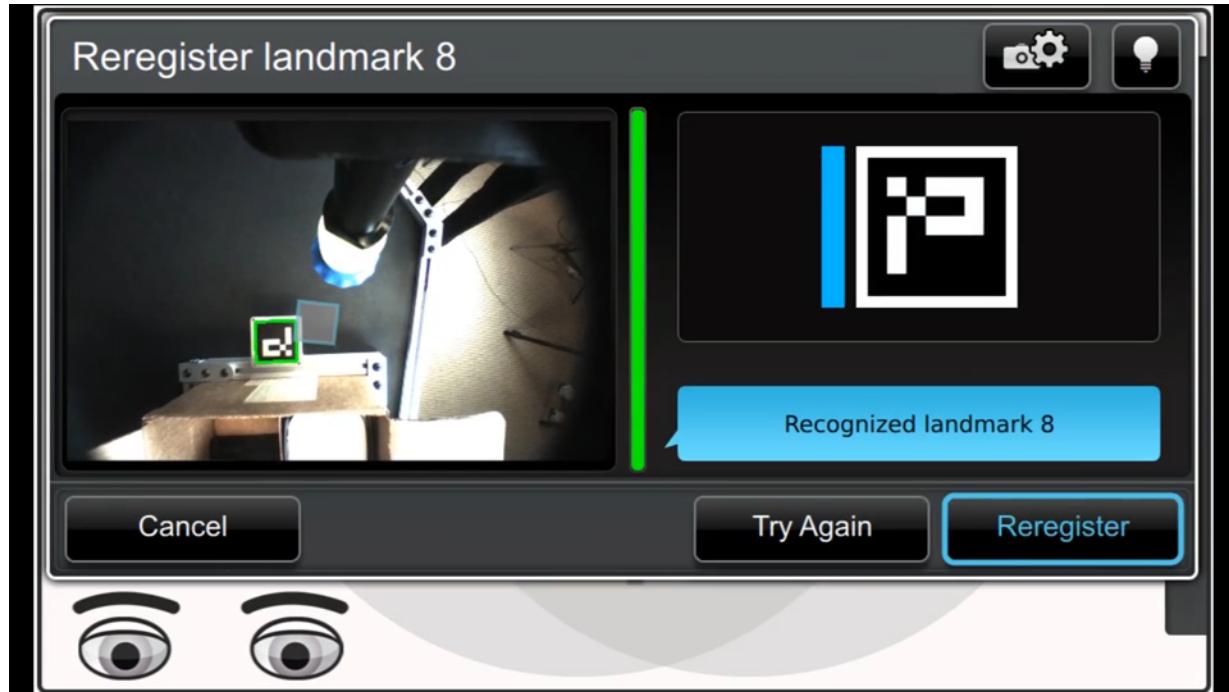
To reregister a landmark:

1. Press the Rethink button.
2. Select the landmark gallery.



3. Select the landmark to be reregistered. In this case, it's landmark 8.
4. Select **Reregister**.
5. Press **Start**.

The arm moves to its original location.



A gray outline indicates the location where the landmark was originally saved.

If the new location of the landmark is no more than 50mm and/or 10 degrees rotation from the original landmark location, Baxter will recognize the new location.

If the new location is too far off, just move Baxter or the module until the green outline is displayed.

6. Press **Reregister**.



The landmark is reregistered (note the checkmark) and the location is realigned with the original task location.



Robot Positioning System - Best Practices

- To make it easier to move Baxter back into its approximate, original position, especially if it's common to move the robot periodically, many customers find it useful to mark Baxter's work location on the floor.
- When registering a landmark, give the arm a moment to settle (a good rule of thumb is 5 seconds) before registering. The same applies when reregistering.
- When registering the robot to a landmark, avoid pressing the OK button on the arm being registered; this makes the arm shake and may register the robot incorrectly. You can use the OK button on the other arm, or on either side of the torso.
- Custom paths are not redefined when you reregister using Robot Positioning System, so you may want to keep your waypoints limited in number and relatively far apart from the Approach and Retract points in order to give more flexibility to the robot.
- If you do need to avoid an obstacle, use a series of Holds and associate those to a landmark instead of using paths to avoid the obstacle.
- Keep the surface containing the landmark(s) level.
- Pay attention to the height of the robot when reregistering. The height should be the same when reregistering as it was when originally registered. So, for example, if you unlock the casters on the pedestal and raise the robot to roll it out of its work location temporarily, remember to return the robot to its original height when you roll it back.
- The weight of the grippers can affect the accuracy of Baxter's arm movements. If you register landmarks for a task with the grippers attached, make sure the grippers are also attached when you reregister. Conversely, if the grippers were off when registering, they should be off when reregistering.
- If you have registered landmarks for both arms, reregister first one arm, then the other. It may take several reregistrations to complete the process. That is, you may need to go back additional times to reregister one arm after you have reregistered (and/or adjusted) the other.



Frequently Asked Questions about Robot Positioning System

Q: I tried registering a landmark, but I get a message that says “Multiple landmarks are visible.” What should I do?

A: The robot is seeing more than one landmark. Cover one of the landmarks with your hand until just one is recognized. Then, press Save and keep your hand covering the other landmark until the desired landmark is saved.

Q: I registered a landmark but now I would like to remove it. What do I do?

A: Select the action whose landmark you’d like to remove and go into the landmark gallery panel as described above. Once in this gallery, select and the landmark will be de-associated from that particular action.

Q: When I relocate the robot, will my custom paths be erased?

A: No. The path will remain the same but the end points (i.e. Approach/Retract) will move to the new locations.

Q: When I have Action Groups, do I need to associate each action to a landmark or may I just associate the whole Action group to the landmark?

A: If you have an Action Group and would like to associate all actions to a landmark, you can just associate the whole Action Group in one step.

Q: Are there any special lighting requirements for using landmarks?

A: Recognition of landmarks works best with soft, diffused lighting that does not produce hard shadows.

Q: Does the landmark need to be trained at the same orientation as the pick or place action?

A: No. You could have your landmark mounted horizontally to the frame of the conveyor even though Baxter will pick in an off-vertical orientation on the conveyor.

Q: Can a landmark be associated with a Hold?



A: Yes.

Q: What do I do when a landmark is damaged?

A: Replace it with a new landmark and use the Overwrite feature in the landmarks gallery. Remember to overwrite the Landmark before removing it from the workspace.

Q: What do I do when I need to associate waypoints in a path to a Landmark?

A: It's not currently possible to associate paths to a Landmark. Therefore, if you absolutely need to make sure waypoints are associated to a Landmark, please turn these waypoints into Holds and associate those to a Landmark.

Application Examples

Unloading Parts from a Rotary Table and Onto Conveyors

TASK DESCRIPTION

Baxter is working on a rotary table offloading parts from the table and placing them on conveyors to either side of the robot using both arms. The rotary table has more than one nest where it will pick parts from. The table and conveyor are on wheels and this line is constantly being moved. The engineer doesn't like to have to train the task every single time they have to move the line somewhere else.

SOLUTION

Even though there is more than one nest on the rotary table, the pick location is the same for all of them. Therefore landmarks only need to be applied on one of the nests and the conveyor belts.

1. Train PICK and PLACE for RIGHT arm.
2. Train PICK and PLACE for LEFT arm.
3. Position one landmark on each of the conveyors so that Baxter can clearly reach and see them.
4. Position one landmark on each of the sides of one nest so that each of the arms can reach and see its respective landmarks.



5. Create and associate each action to its relevant landmark.

Done.

Transferring Parts from One Conveyor to Another

TASK DESCRIPTION

Baxter is working at the end of a line transferring parts from one conveyor to another and is using both arms. Operators work near the conveyors and often bump into these, moving them out of place and Baxter has problems picking or placing. The engineer would like to be able to re-register the robot without having to retrain the task or even retrain specific actions.

SOLUTION

Even though there are only two modules, four landmarks will need to be placed because each arm is independent of the other.

1. Train PICK and PLACE for RIGHT arm.
2. Train PICK and PLACE for LEFT arm.
3. Position one landmark on each of the conveyors for each of the arms so that both arms can reach and see its respective landmarks.
4. Train each ACTION to its relevant landmark.

Done.

Putting parts into fixtures that require precision

TASK DESCRIPTION

Baxter is working near a bench with a fixture that is often being moved depending on what task is being run that day. The fixture requires a lot of precision during the placement and the engineer would like to be able to replace the fixture and not have to retrain the task every time.

SOLUTION

In order to increase precision, place the landmark as close to the action as possible.



1. Train PICK and PLACE.
2. Position one landmark on the fixture close to the actions.
3. Train each ACTION to its relevant landmark.

Done.



Signals

Working with Signals

Robots running Intera 3.3 (Sawyer and Baxter) can send (out) or receive (in) digital (on/off, true/false, yes/no) signals from other machines. Baxter has two built-in signal ports on its DB15 connector, one In and one Out. You can attach and configure additional devices via the robot's Ethernet port.

The robot sends and recognizes the following signals:



Note: Unless otherwise noted, level signals are active when true; edge signals look for false-to-true transitions.

You'll notice that the robot displays in-context descriptions of each of the signals in its signal library, so if you're unsure which signal you want to use in your current situation, these descriptions will help.

- In Ready (level and edge) – Signals to the robot that the external world is ready for it to perform an action. In some cases, the In Ready signal also tells the robot to reset counts. You can configure a ready signal as Gate or Enable. See “Ready Signal: Configure as Gate or Enable” on page 137.



- In Error (level) – Signals to the robot that an external machine or device has an error. Baxter stops and waits until In Error becomes false before resuming where it left off.
- In Pause - Freezes the robot when true. (Pause can sometimes be more convenient than an Error In signal, which causes the robot to display a confused face and requires someone to interact with the robot to "fix" whatever caused the error in signal.)
- In Skip (level) - For subtasks, signals that the task should skip to the next subtask. This signal is only considered when a subtask is between action sequences. For Hold actions, causes the hold action to terminate immediately instead of waiting until the time interval has elapsed.
- In Do Subtask - Allows you to more easily tell the robot what to do and when to do it. If a task has all its subtasks with the Do Subtask signal assigned, each subtask is, in effect, in "listening" mode. You can send a Do Subtask signal at any point to instruct the robot to perform a specific subtask and then wait there until the next signal tells the robot where to go and what to do. This provides greater flexibility when working with a ready signal and reduces ladder logic in a PLC. If all subtasks do not have the "Do Subtask" signal associated with them, the robot will perform its task typically, skipping those subtasks that have "Do Subtask" associated but not asserted.
- Reset In - Baxter can receive a Reset In Signal if a PLC triggers the robot to reset its task to start at the first action. This is essentially the same as stopping the robot and pressing the reset button. Returns all counts to 0.
- Reset Count - Resets count when true.
- Out Done (level) - This signals that the robot is done with an action, subtask or task, i.e., that the count is complete. This signal may stay true for an arbitrary length of time, but never less than 0.5 seconds.
- Out Count (edge) - Signals that the robot has executed an action that has changed a count.
- Out Error (level) - Signals that the robot has a problem and has stopped executing the task.
- Out Confusion (level) - Signals that the robot may need materials or help. User intervention is likely required.
- Out Action Started (edge) - Signals that the robot is starting on an action or subtask, i.e., that the arm has reached the action's approach point. In other words, start pulses at the approach pose of the first action of the subtask. You can define the duration of a pulse for as much as 5 seconds or as little as .5 seconds. The default is 1 second.
- Out Action Ended (edge) - Signals that the robot has finished an action or subtask, i.e., pulses at the retract pose of the last action in the subtask. You can



define the duration of a pulse for as much as 5 seconds or as little as .5 seconds. The default is 1 second.

- Reset Out - Baxter can send a Reset Out Signal to reset a PLC when the user presses “Reset” in the robot’s UI. Returns all counts to 0.
- Hold Active - Sent when the robot’s arm arrives at a Hold location. When the robot’s arm settles in the hold pose, the Active signal is sent out and will remain true for the duration you have defined, whether 5 seconds, 30 minutes or unlimited. When that time is up, the robot will move on to its next location and wait for the next signal.

Note: The duration of the Active signal is equal to whichever is longer: the configured hold time or the pulse duration.

Internal Signals

Internal signals do not involve communications between the robot and outside devices. Rather, they use the existing signaling system for communication within the robot itself.

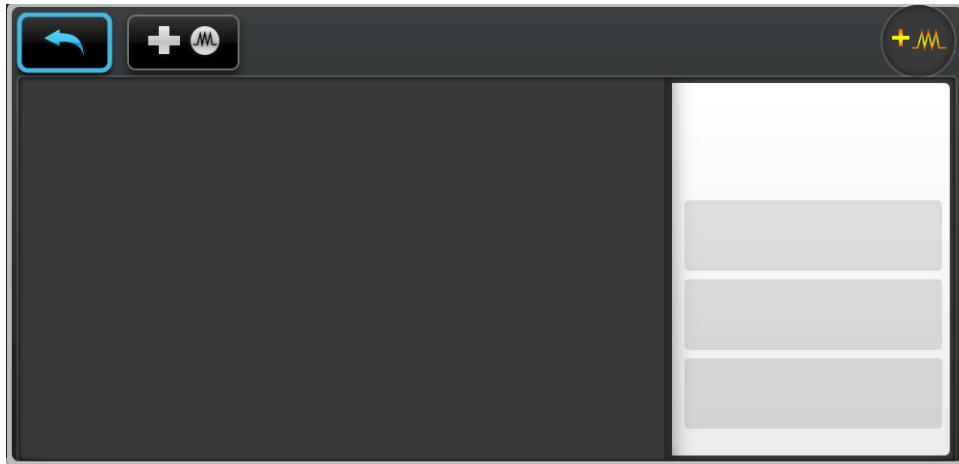
To set up, internal signals need to have an input and output on the same line, and the trigger logic of the input (Gate/Enable) needs to work with the duration of the output, i.e., Gate will work with a duration of 0.5 seconds, but Enable will likely need a longer duration.

Note: Baxter will show confusion when an internal input signal does not have a corresponding output signal on the same line.

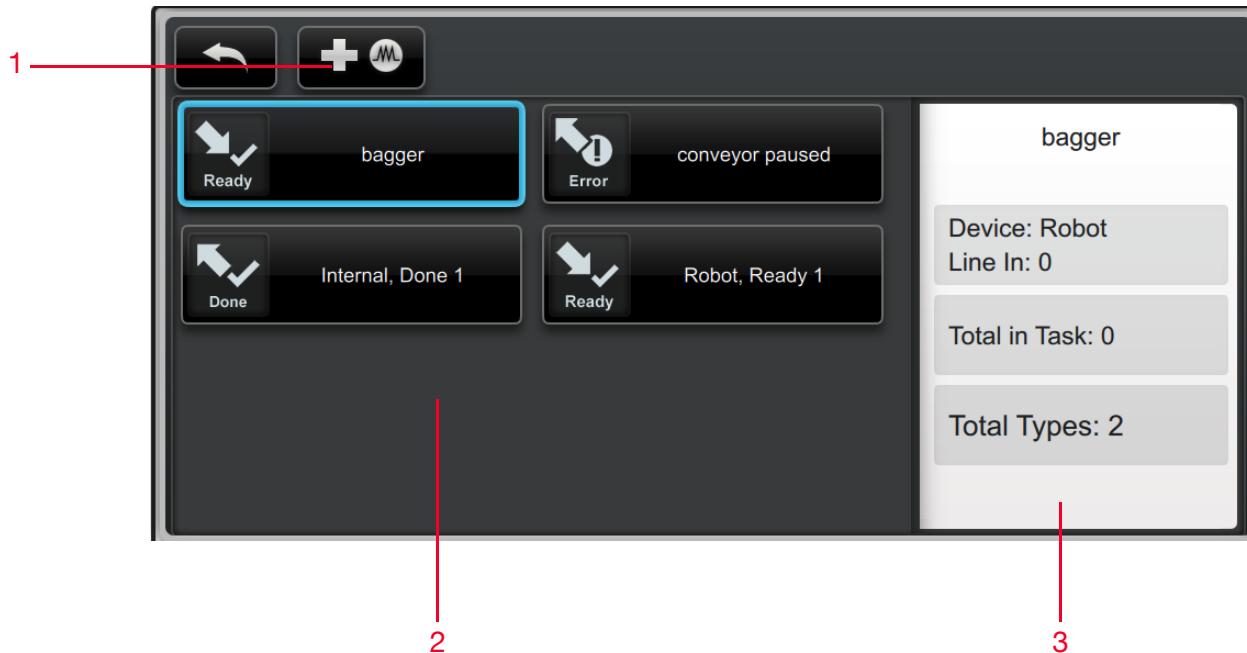


THE SIGNALS GALLERY

From the Task Map, press the Rethink button, and click I/O to access the *signals gallery*. Use this screen to view available signals, or add new ones.



When you first access the signals gallery, it will be blank, as in the illustration above, if no signals have been created yet. If signals do exist in the gallery, the display will be similar to the one below.



1. Create signal – Add a signal to the gallery.
2. Displays a list of all signals defined for the current task.
3. Displays the details of the selected signal.
 - Name of the signal
 - Device name and line number
 - Number of places the signal is added to the task
 - Number of the type (e.g., In Ready) of signals created



Creating Signals

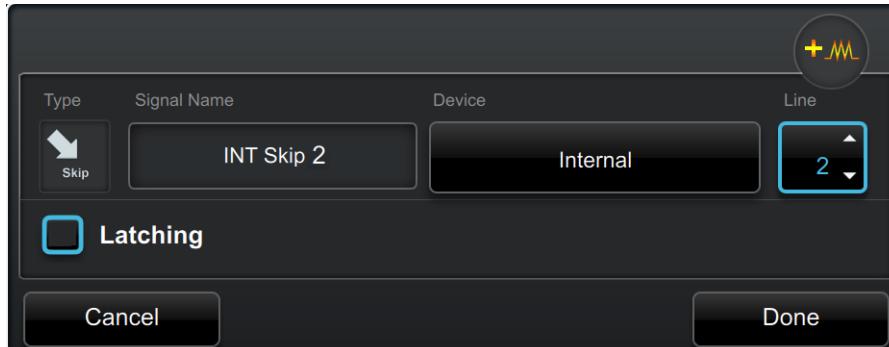
To set up signals, first attach and configure the device (see “Appendix B: Configuring External Devices” on page 160), then, create a signal and store it in the signal gallery:

1. In the task gallery, click the create signal icon.
2. In the add signal screen click the type of signal. “In” signal icons feature a right-and-down pointing arrow (first and second rows, below); “out” signals all have a left-and-up pointing arrow (third and fourth rows).



In the configure signal screen, change the name if it is not clear to your team; then reference the correct device, and the line number.

3. When complete, click **Done**. You can later return to modify any of these details.





LATCHING SIGNAL

Checking the Latching box allows output signals to maintain a high or low state until another signal tells it otherwise. Use polarity to define when it should stay high or low.

This feature is helpful when using third party grippers because it allows you to keep the grippers open or closed using signals.

For example, you have attached and configured a 3rd party gripper that requires a constant signal state to grasp, and the inverse signal state to release. Follow these steps:

1. Create a Hold that will act as your pick.
2. Configure an Active signal for this Hold on Device 1, Line 2 and check the “Latching” box (see the image above).
3. In the Association Panel set the Polarity to High. That tells the gripper to grasp;



the signal stays high until you specify it to go low.

4. Create another Hold (which could be the next Hold or any other Hold) where you want to place the part.
5. Associate the Active signal to this Hold on Device 1, Line 2. Toggle the Polarity to



Low to release the part.

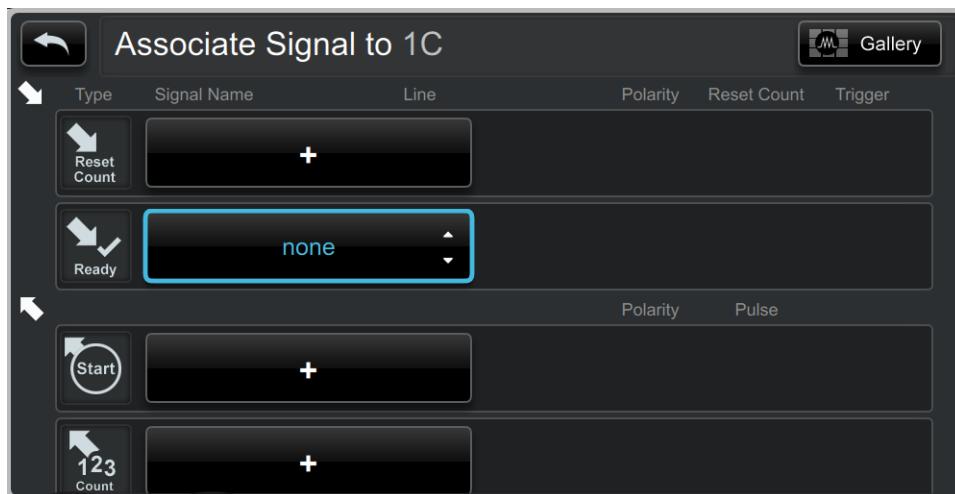


Assigning Signals

To assign a signal:

1. Select an action, task, or subtask.
2. From its modify screen, select the + button.
3. Select **Signal** from the submenu.

The robot displays the Associate Signal screen, where you can both define and assign a signal:



4. Select the plus (+) button to define a signal and its attributes.
5. When finished, click the **Back** icon on the screen.

DISPLAY STATE OF INPUT SIGNALS; TEST OUTPUT SIGNALS

In the Association panel a light next to each signal indicates the state of the signals once the signal has been associated with an action.

INPUTS

- Green = signal is received.



- Gray = signal is not being received.
- Red = robot is not communicating with the external device.



OUTPUTS

To test outputs, press the Rethink Button while you are in the Association Panel.

- Gray light = signal is on a false state.
- Green light = signal is on a true state.

The light will flash for the correct duration of the signal.

If the signal is latching, the light will switch to the correct state and remain.

SIGNALS NOTES

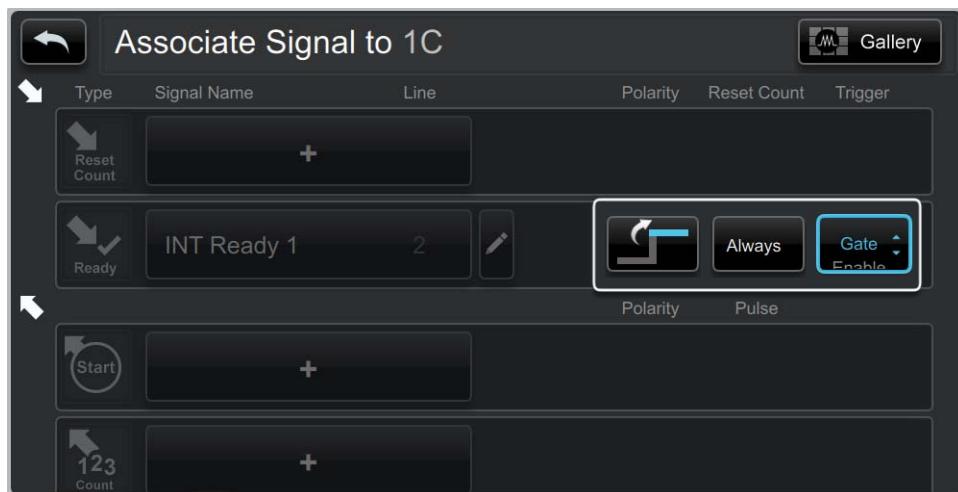
- You can easily invert the signals the robot sends or receives by clicking on the invert signal icon  in the Polarity column.
- Pulse - For more flexibility when integrating with external devices, you can define the duration of a pulse for an Out signal. You can set it to as much as 5 seconds or as little as .5 seconds. The default is 1 second.
- To disable an existing signal, select none in the spin box next to the signal, then return with the Back button. This removes the association of the signal from the current action, subtask, or at the task level, but the signal itself remains available, so it can be associated to other actions, subtasks, or at the task level.
- Internal signals default names all begin with INT. (e.g. INT Ready 1)
- As of Intera 3.2, the designation Torso is now Robot, and default names all begin with Robot. (e.g. Robot, Ready).



- The "Reset Count" policy for a Ready signal can be set to **never**, and you can separately assign a Reset Count signal, if needed.

READY SIGNAL: CONFIGURE AS GATE OR ENABLE

For pick and place actions with a Ready signal, you have the option of instructing the robot what to do if a signal that was true when the robot started the action becomes false. For example, in the case of using a part present signal, if the part disappears (and the signal is false), you can choose whether you want the robot to continue moving or to stop.



- If **Enable** is chosen for the signal, and the part present signal disappears (turns false), the robot will stop, go back to its approach, and wait for the signal to be true before trying again.
- If **Gate** is chosen for the signal, (the gate being defined as the action's approach), once past the gate, even if the part present signal disappears, the arm will continue.



Signals and their Definitions

This section discusses available signals and their exact definitions. It applies to both Sawyer and Baxter robots.

Terminology

Signal: a signal transmits a piece of information. Signals are binary, meaning they are only on or off. A signal can be an input signal received by Intera from an external source or an output signal sent by Intera to an external source.

Polarity: a binary signal is either true or false. By default, when Intera reads an input signal, it interprets a zero as false and a one as true. Conversely, when it sends an output signal, it sends a zero for false and a one for true. How zero and one relate to the real world depends upon how the external devices are wired and configured.

Invert: Intera has the option to invert an input or output signal. An inverted signal is simply interpreted as opposite to the default. Namely, a zero is considered true and a one is considered false.

Signal classes: Intera uses two main types of signals:

Level: the information is contained solely in the current value of the signal. This represents the state of something, e.g., whether a part is present or whether an operation is done. Input level signals are used in one of two ways:

- **Gate (input):** Intera waits at specific points in a task for a gate signal to be true. Once true, the value of the gate is ignored until that point in the task is reached again. To be reliably recognized, a gate input should have a minimum duration of 0.5 seconds.
- **Enable (input):** Intera will wait for an enable signal to be true before executing particular actions, then requires that the enable signal remain true or it will abort the action as soon as possible.

Pulse (output), edge (input): the information is contained in the change of the signal from false to true (the rising edge of the signal.) How long the signal remains true doesn't matter.



This represents the occurrence of an event. For example, when an action is started, a pulse is sent.

- An edge input should remain true for a minimum of 0.5 seconds to ensure it is reliably recognized.
- Pulse outputs are true for at least 0.5 seconds.
- Other notes

In advanced applications, there are some cases where it makes sense to connect a pulse output to a level input, or a level output to an edge input. For a contrived example, a user could connect a Done (output level) signal to a Task Reset (input edge) and the task would reset when the Done signal became true.

Task element: a task element is a piece of a task to which signals can be assigned. Elements include: the overall task itself, a subtask, an action group, an individual action. Action Sequences are also task elements, though signals cannot be attached directly to them.

Signal types: Intera has a predefined set of signal types. In order to define a signal, a type must be selected, it must be named, and it must be associated with a particular input or output. Once defined, a signal can only be connected to an input or output of a task element of the same type.

Signal Types

Intera has a predefined set of signal types. While a type implies a certain high-level meaning, the details of how any signal operates are dependent upon the element to which the signal is attached. The main types are described below.

INPUTS

Ready (edge and level): when true, indicates that the task element is okay to run. In many cases the edge of the Ready signal is also used to reset the count or error state of the connected element.

Skip (level): when true, causes Intera to move on from the current element.

Task Reset In (edge and level): pauses (level) and resets (edge) the task.

Error In (level): stops the task.



Pause (level, enable): pauses the task when asserted

Do Subtask (level): when true, signal the subject is allowed to run.

Reset Count (edge): reset count for the associated task element.

OUTPUTS

Done (level): the count for the particular task element has been reached. Stays true until the count is reset.

Note: while Done is a level, due to the automatic resetting of counts it will often stay on for only a short period of time, but always at least 0.5 seconds.

Increment Count (pulse): the count for a particular task element has just changed. This only happens if the element is successful.

Start (pulse): a task element is starting.

End (pulse): a task element is finishing.

Task Reset Out (pulse): the Task has been reset.

Error Out (level): the Task has an internal error.

Confusion (level): the Task is in a confused state.

Active (pulse): the associated hold is currently executing.

INTERACTION OF READY AND CONFUSION

In most cases, connecting the Ready signal to a task element will suppress that element from causing confusion (and, therefore, setting the Confusion signal true). Specifically, if a task is not currently running any actions, Intera makes an effort to determine whether a Ready signal becoming true would allow the task to proceed or whether the rising edge of Ready will reset a count or error state. If so, Intera suppresses the confusion signal.



Task Element Details

ACTION

An action is a Pick, Place or Hold that is performed at a particular pose in the workspace. Signals can be attached to a single Action that is not part of an Action Group.

ACTION GROUP

An Action Group is a Pick or Place that is applied, usually with common parameters, to a group of poses in the workspace. Signals can only be applied to an Action Group as a whole, not to individual Actions within the group. For input signals, it is as if the signal was attached to each Action individually. For output signals, Intera generates the logical OR or logical AND of the outputs from each of the Actions, depending upon what makes sense for the particular signal type. For example, the Done signal is true if and only if all of the Actions are done (logical AND), whereas any of the pulses (e.g., Start) occur when any one of the Actions generate them (logical OR).

ACTION SEQUENCE

An Action Sequence (“AS”) consists of five lists of Actions executed in a defined order each time the AS is started. The lists are:

1. Zero or more Holds, executed sequentially
2. One or more Picks, from which one is chosen
3. Zero or more Holds, executed sequentially
4. One or more Places, from which one is chosen
5. Zero or more Holds, executed sequentially

In a Subtask with a single AS, the AS will start if a Pick is available to run. In a Subtask with multiple AS's, an AS will start if both a Pick and a Place are available to run. Normally, an Action Sequence will run to completion once it starts. However, if the start condition becomes false before a Pick has been completed, the AS will abort. If an AS has picked a part but there is no available Place to run, the AS will continue to run and wait until the Place becomes available.

SUBTASK

A Subtask comprises one or more Action Sequences.



Subtasks are defined to end when no Action Sequence is running and either:

- a. all of the Place counts for all Actions in all Action Sequences in the Subtask are full or,
- b. the Skip signal is true.

Errors (e.g., pick or place failures) will not cause a Subtask to end and will not cause the count to be incremented. Therefore, if errors prevent Actions from running and Place counts can't be satisfied, the Subtask may never end.

Signals

In Intera, there are a number of task elements to which signals can be attached. These include the overall task, subtasks, action groups, as well as individual actions. The meaning of a signal is dependent both upon its type and the place to which the signal is connected. Discussed below are each of the places that support signals and each of the signals they support.

TASK

These signals are accessed in the Task Order screen. For non-coordinated tasks, these signals can be attached separately to the task for each arm, except where noted. For coordinated tasks, these signals are attached to the overall task.

INPUTS

- **Ready (level (gate)):** checked before starting the first subtask of a task. When true, allows the task to start.
- **Task Reset In (edge, level):** as an edge, resets the task. The reset happens immediately, all counts are reset and, if running, the task is started over from the beginning. As a level, the task will remain paused while true.
 - Acts like pressing Reset from the menu, but will not start the task if it's not already running.
 - If either arm is gripping an object at the time of reset the task will not run and Intera will show confusion.
- **Error In (level):** when true, the robot is stopped. The user will need to press the run or reset button. The Error In signal can be asserted at any point in task execution, or even during training.
 - The Error In signal can be assigned to the task for each arm, but asserting the signal on either task will stop both arms.



- **Pause (level):** At any point during task execution, if the pause signal is asserted, task execution will be paused until the pause signal is cleared.
- **Reset Count (edge):** When the Reset Count pulse is detected, the task count will be reset.

OUTPUTS

- **Done (level):** true when the task is done, i.e., it has completed the number of cycles specified by the count.
 - Goes to false immediately when the task is reset.
 - A task is only considered Done when it has completed the specified count of cycles. If error conditions prevent the task from finishing then its count will not increment and it will not be Done.
- **Increment Count (pulse):** sent when the task has completed its last subtask and the task count has been incremented.
 - If a subtask is unable to complete because of errors, then the task won't complete and this signal will not be sent.
 - If all subtasks are complete either because their count is done and has not been reset or Skip is asserted, the task count will increment rapidly until a subtask starts to run. In this case, the Task Increment Count signal won't be particularly useful.
- **Error Out (level):** true when there is an error that prevents the task from running.
 - Error Out is asserted when there are alert stop messages from the UI including EStop.
 - In a non-coordinated task, though the task for each arm has a separate Error Out signal, they are usually both the same. (If Task Editor is used to set the FailureStrategy to Stop, then an Action failure on one arm can assert Error Out on that arm without asserting it on the other.)
 - If the Error Out signal is due to task-related causes such as failure to pick or place, it will be reset when the task is reset. If the signal is due to hardware or external errors, it will remain asserted through a reset until the error condition itself is cleared.
 - For non-error conditions see Confusion below.
- **Confusion (level):** true when the robot is showing confusion.
 - In a non-coordinated task, it is possible for one arm's task to be paused due to confusion while the other arm's task continues to run.
- **Reset (pulse):** signals pulses true when the task is reset by any means.



SUBTASK

INPUTS

- **Ready (edge, gate):** when true, allows the subtask to start. When a task reaches a subtask with a false Ready signal, the task will pause until the signal becomes true. The rising edge of Ready immediately resets the subtask count.
- **Skip (gate):** when true, causes the subtask to finish as soon as it can and the task to continue to the next subtask.
 - A Skip signal is only examined between Action Sequences. A subtask will never end while in the middle of an Action Sequence.
 - If Skip is true when a subtask with count of one is about to start, the subtask will be skipped.
 - A skipped subtask will increment its count and send an Increment Count signal.
- **Do Subtask (level):** When associated, signal must be asserted for subtask execution to occur.
- **Reset Count (edge):** When the Reset Count pulse is detected, the subtask count will be reset.

OUTPUTS

- **Done (level):** true when the subtask is done, i.e., it has completed the number of cycles specified by its count.
- **Increment Count (pulse):** sent when the subtask has completed a cycle and the subtask count has been incremented.
- **Start (pulse):** signal pulses true when subtask execution starts
- **End (pulse):** signal pulses true when subtask execution ends

GRIPPING ACTIONS: PICK, PLACE

INPUTS

- **Ready (enable, edge):** when false, prevents the action from running. When true, allows the action to run if all other conditions are met (e.g., count.) In addition, the rising of edge of Ready immediately resets the count the on the action depending upon the reset always/full setting.
 - If Ready goes false while an action is running and the action has not yet taken place (i.e., the part has not yet been picked or placed), then the action is aborted immediately. If the action has been performed then it completes normally regardless of the state of Ready.



- **Reset Count (edge):** When the Reset Count pulse is detected, the action's count will be reset.

OUTPUTS

- **Done (level):** true when the Action's count has been reached.
 - Due to the automatic reset of counts, if a Ready signal isn't connected Done may be on for only a short time, but in no case less than 0.5 seconds.
 - Internally, Intera counts the number of parts available for a Pick or Place. Therefore, Picks count down and assert Done when the number of available parts reaches 0. If a Pick fails a certain number of times in a row, normally 2, one of two actions will be taken:
 - 1.The Pick goes into an error state which leaves the count alone, so Done will not be asserted. This is the default.
 - 2.If the SoftFailureLimitAction attribute is set to EMPTY, the count will be set to zero (a sensed empty condition) and as a result Done will be asserted.
- **Increment Count (pulse):** sent when the Action has succeeded and the count has changed.
 - The pulse is sent when the Action successfully picks or places a part and the count has changed. This is typically after the retract motion has completed, but may be earlier if the Action is aborted during the retract phase, e.g., if the user grabs the cuff.
 - The internal count is not changed when an error occurs, therefore the increment count signal will not be pulsed.
- **Start (pulse):** sent when the action is starting its approach.
 - One pulse will be sent for each retry of the Action.
- **End (pulse):** sent when the action has finished its retract.
 - The pulse is sent when the Action has completed, whether successful or not, typically after the retract motion.
 - One pulse will be sent for each retry of the Action.

OTHER ACTIONS: HOLD

INPUTS

- **Ready (level (enable)):** when false, prevents the action from running. When true, allows the action to run if all other conditions are met (e.g., count.)



- If Ready goes false before the hold time is up, then the Hold is aborted immediately and the Action Sequence will wait until Ready becomes true before restarting the Hold.
- Since Holds in an Action Sequence are executed sequentially, the task will be paused when it encounters a Hold with a false Ready signal.
- **Skip (level (gate)):** if true any time during the hold period, the Hold acts as if the wait time has completed.
 - The Hold will always execute the approach and retract regardless of the Skip signal. However, the Skip signal can reduce the hold time to effectively zero.

OUTPUTS

- **Done (level):** not applicable.
- **Increment Count (pulse):** not applicable.
- **Start (pulse):** sent when the Hold is starting its approach.
- **End (pulse):** sent when the Hold has finished its retract.
- **Active (pulse):** When the hold is active / currently holding at the hold pose, the active signal will be pulsed true.

ACTION GROUPS

When actions are in a group, signals can only be attached to the action group and not to individual actions within the group. When an input signal is attached to an action group, it acts as if it was attached to each action within the group individually. When an output signal is attached to an action group, it is true if the output of any of the individual actions is true, and false otherwise (i.e., the outputs of the individual actions are OR'ed together.)

FAQs - Signals and Their Definitions

Q: How long will a Done signal stay true after completing a Task, Subtask, Action?

A: In general, the Done signal remains true until the count on the element is reset. When using automatic reset, i.e., the Ready signal isn't being used, Action counts are reset when their containing Subtask ends, whereas Subtask counts are reset when the Task completes. Due to the automatic resetting (e.g., in a Subtask with only one Place Action, the Subtask will end when the count is satisfied and then immediately reset the count). Done signals may only stay true for a short period of time, but in any case no less than 0.5 seconds.



Q: Why doesn't my Action run the second time through even though the Ready signal has always been true?

A: Check the count, which is only reset on the rising edge of Ready. Therefore, if you have a count of 1 and you leave Ready true, the Action will only happen once and the count will never be reset. If you're using Ready to control a Pick, set the count to infinity. If you're using Ready to control a Place and are using multiple subtasks, you can either set the count to infinity and use the Skip signal to move on to the next subtask or make sure that you cycle Ready false to true to reset the count.

Q: How do I use signaling and an external PLC to run Action Sequences out of order?

A: There are two main approaches to using signaling to run sequences out of order using an external PLC:

1. Put multiple Action Sequences into single Subtask using the combine operation on the Task Order screen. Use Ready signals on the Picks to choose which Action Sequence to run.

a. Counts should be set to infinity for all Picks and Places, since the single subtask never ends and the sequence is being controlled externally.

b. The Ready signal for a Pick must be held true until the action completes. A way to reliably synchronize is to monitor the Increment Count signal and use that to clear Ready. An alternative approach would be to use the Increment Count on the corresponding Place, which would allow the Pick to repeat if the part was dropped. Which approach is better is highly dependent on the application.

2. Put each Action Sequence into its own Subtask and use the Subtask Skip signal to choose which Subtask to run.

a. All Subtasks should have their own Skip signal connected. In the idle case, all of the Skip signals should be true.

b. To run a particular Subtask, the PLC should set that Subtask's Skip signal to false. To operate reliably, the signal should remain false until the PLC receives a positive acknowledgment that the Subtask has started. This could be from the Start signal of the first Action, for example.



- c. In order to ensure that the Subtask doesn't start again immediately upon completing the Action Sequence, Skip should be cleared immediately upon receiving the indication that the sequence has started. Since Skip is only looked at when no Action Sequence is running, this will allow the current sequence to complete and not restart.
- d. Do not use the Subtask Ready signal.

Q: If I invert an output signal (e.g. Done) and have not started the task, what will the output be and when does it become inverted?

A: Inversion takes place as soon as the modification is saved in the task and the changes have time to propagate throughout the system, typically a second or so.

Q: In the case of a HOLD -> PICK -> PLACE, why won't the robot go back to the first HOLD when PICK READY is not enabled?

A: Action Sequences are defined to not start unless there is an available Pick. This is to prevent the robot from getting stuck in a sequence when a better choice might come along. For example, a Skip signal might want to end the Subtask, but that can't happen in the middle of an Action Sequence.

Q: If two input signals become true at the same time (e.g. Pick Ready and Subtask Skip), can I predict what will happen? Is there a way to understand what has a higher probability of getting chosen?

A: Relying on the robot's behavior in this case is fundamentally unreliable system design. This is a classic synchronization problem and there is fundamentally no way to predict which signal will take priority. Whichever one Intera happens to notice first will take effect. Note that even in a procedural language, where you might be checking one signal then the other in a loop in order to decide what to do next, if they change at literally the same time you can't predict which one will be noticed first.

In situations where two signals might change at nearly the same time and predictable behavior is defined and required, we highly recommend that you use a PLC and logic that creates a window in which both signals are examined and an unambiguous decision is made, which is then communicated to and synchronized with Intera by asserting only one signal representing the required action.

Q: Where can I get the most up-to-date information on signals, Intera 3.1, or anything else related to Baxter?



Please visit our Wiki at: mfg.rethinkrobotics.com



Maintaining and Supporting Baxter

Powering Down Baxter

1. Clear the area around the robot.
2. If performing maintenance, grab the training cuff or turn the knob to cause the head to move to the side. (Pressing the button may run or reset a task, so it's safer and simpler to grab the cuff or turn the knob.) Moving the head to the side makes it easier to remove the front cover. If the robot does not have power, carefully move the head manually.
3. Press the white power button on the robot pedestal.

The shutdown process is complete when the green and yellow lights around the head shut off.

Unplug the power cord from the wall outlet/power source.

Maintaining Baxter

Cleaning Baxter

To clean Baxter, periodically wipe it down with a clean, damp cloth.

Replacing the Air Filters

Check the Baxter air filters at least every six months, and replace them as necessary. Baxter has two air filters, one on each side of the torso.

- To remove a filter, use a 2 mm hex key to remove the two screws from the air filter; remove the filter.
- To install a filter, use a 2 mm hex key to install the two screws and attach the filter to the side of the torso.



Upgrading Software

IMPORTANT

- Please refer to the Read Me First guide posted on the FTP site for best practices when upgrading.
- You must be running Intera release 3.2.0, 3.2.1, 3.2.2, or 3.2.3 to upgrade to Intera 3.3. If you have an older version of the software, you must first upgrade to Intera 3.2.0, 3.2.1, 3.2.2, or 3.2.3 before Baxter can be upgraded to 3.3.
- Please export all your tasks prior to upgrading to Intera 3.3.
- Should you need to downgrade from the Intera 3.3 beta, please contact support.
- We recommend using a USB stick with at least 4GB of storage.

TO UPGRADE:

1. On Baxter, go to **Main Screen > Settings > Advanced > Update Software**.
2. Insert the USB device that includes the software upgrade, and wait for Baxter to recognize the device.
3. Select the version of software you want to install.
4. Select **Yes Continue** in the confirmation dialog.
A status screen appears as the software loads.
5. When the software finishes installing, restart Baxter.

Importing and Exporting Tasks

You can copy tasks from one robot and transfer them to another using a USB memory stick. Details of the task are included in the transfer. You only need to update the device referenced. This capability is useful when a factory is using multiple robots to perform the same task.

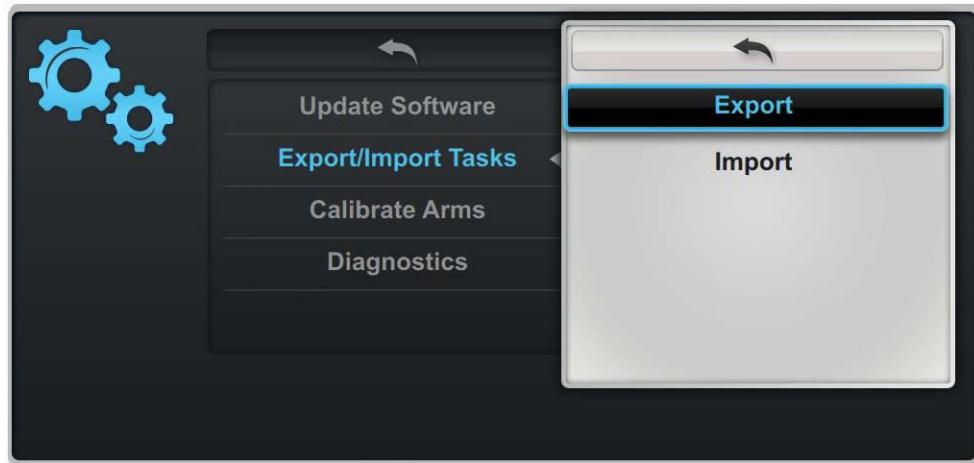
When transferring tasks to a different robot, variability across robots and fiducials are not taken into account.

Note: If you have multiple exports of tasks on the same USB stick, only the most recent set of exported tasks will be imported.

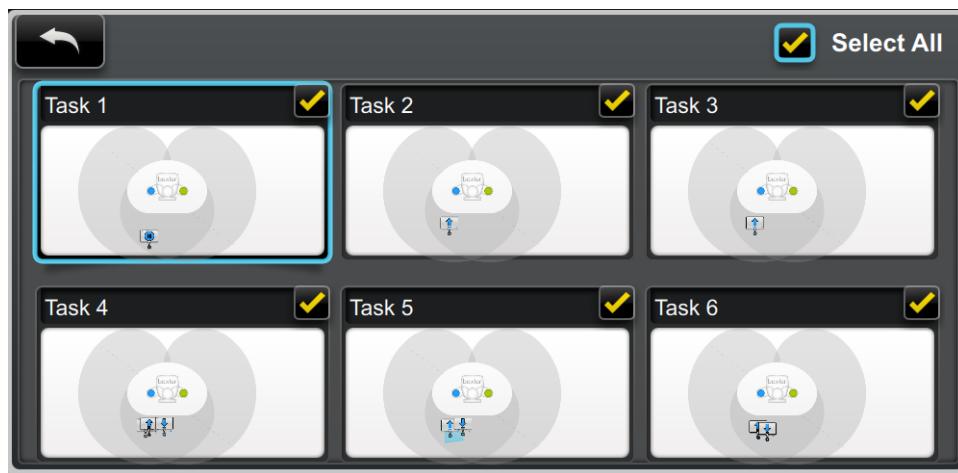


How to Export Tasks from a Robot

1. Insert a USB device, formatted as FAT32, into the USB port on Baxter.
2. On the robot, go to **Main Screen > Settings > Advanced > Export/Import Tasks**.
3. Scroll to **Export**.



4. Press OK. The robot displays its tasks. All tasks are selected by default.



5. To select just some of the tasks, scroll to each particular task and select it or deselect it by pressing OK when it is highlighted.



6. When the tasks you want to export are highlighted, press the **Rethink** button on the Navigator, then select **Export**.

A bar displays the progress of the export to the USB.

Note: If you press the **Back** button during the export process, the export will be canceled.

How to Import Tasks to a Robot

1. Insert a USB device that contains exported tasks into the USB port on the robot.
2. On the robot that will receive the copied tasks, go to **Main Screen > Settings > Advanced > Export/Import**.
3. Scroll to **Import**.
4. Press the **OK** button on the Navigator.

A bar displays the progress of the import.

Note: If you press the **Back** button during the import process, the import will be canceled.

COPIED TASKS

If the robot already had a task named “Task1” in its Task Gallery, the newly created name would be incremented by .1, e.g., “Task1.1.”

A TROUBLESHOOTING CHECKLIST FOR IMPORTED AND EXPORTED TASKS

Because the physical world of the two Baxters – the robot from which the task was imported and the robot to which the copied task was exported – are not identical, you may need to refine these tasks. Here is a checklist of troubleshooting situations you may encounter:

- The grippers may not match.
- Different robot heights – The task runs but the robot is missing picks and places because the height of the pedestal is different (the wheels have been adjusted; the original robot is mounted to a table) etc.
- The task runs but the robot’s position from the work surface is different from the original, in distance and/or angle.



The task runs but the workspace is different from the original in some way. In this case, the operator has to tweak the layout of the workspace or the path of the robot's arm.

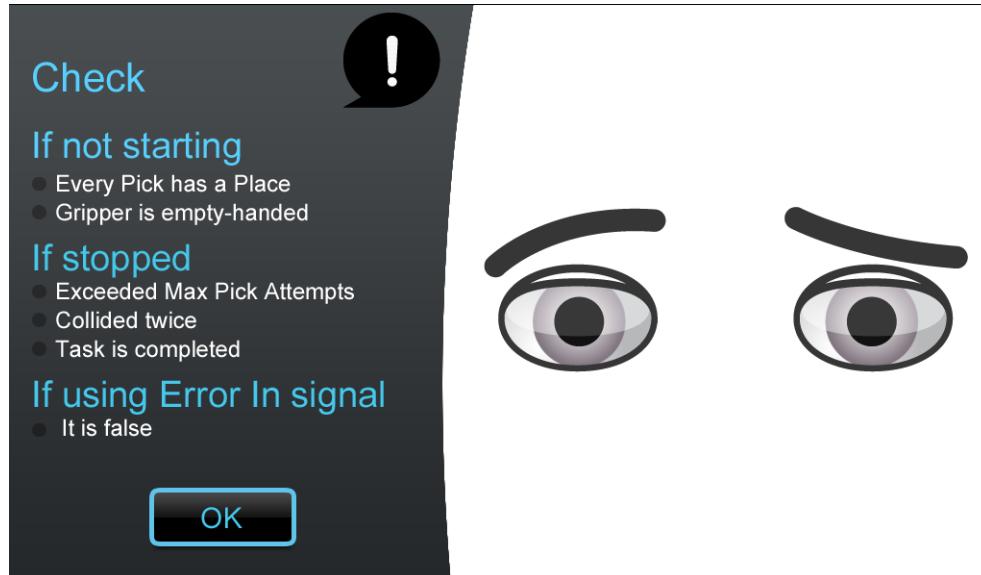
Supporting Baxter

Occasionally, Baxter encounters problems that can inhibit its ability to perform tasks. This section describes the types of problems that may occur, and the actions you can take to resolve them.

Typically you will be able to solve the problem yourself. In some cases, however, you may need to contact your technical support representative for assistance.

Helping Baxter When It Is Confused

When Baxter encounters a problem, it typically will stop and display a message on the screen.



When you see this type of message, follow instructions on the screen to resolve the issue.

Other types of “confused” screens concern external devices not being connected, internal signals not being fully defined, and end effectors that are not configured.



If the issue is not solved even after shutting down and restarting Baxter, contact your technical support representative for assistance.

Calibrating the Arms

Calibrate the arms to troubleshoot common error messages, and to help Baxter maintain its peak performance. It is also recommended that calibration is performed according to the following schedule:

- Weekly – perform a Basic calibration on both arms.
- Monthly and whenever the robot is shipped – perform a Factory and Spring calibration on both arms.

IMPORTANT During calibration, Baxter's arms must be able to move freely in all directions, starting from the base of the robot on up.

IMPORTANT During calibration, Baxter's force sensing is turned off. Before calibrating the arms, remove the grippers, and make sure the workspace is clear above the base of the robot. (Baxter's arms calibrate above the work surface.) If it is not possible to clear the workspace, move Baxter away from the workspace, and clear a 5-foot radius around the robot. While calibration is active, it is important to maintain a distance outside of this 5-foot radius.

If one (or both) of the Baxter arms begins to lose pick and place accuracy while performing trained tasks, try calibrating the arm(s) to fix the problem. Baxter offers three types of arm calibration:

- Basic – performs a basic calibration on the arm or arms selected; use as a first step in fixing the problem. The process takes approximately 15 seconds per arm.
- Factory – performs a thorough calibration of all joints on the arm or arms selected; use if spring calibration does not fix the problem. This process takes approximately 7 minutes 20 seconds per arm.

Note: Always follow up a Factory calibration with a Spring calibration.

- Spring – performs a shoulder calibration on the arm or arms selected; use if basic calibration does not correct the problem. The process takes approximately 3 minutes 20 seconds per arm.



To calibrate an arm:

1. Make sure there is a clear, 5-foot radius around and above the base of the robot.

If Baxter is attached to a pedestal, and it is not possible to clear the workspace around the robot:

- Mark the current location of Baxter by drawing on or affixing tape to the floor near the edges or corners of the pedestal. (This action will help to re-position Baxter after moving it for calibration.)
- Roll Baxter on its pedestal, away from the workspace and into an area large enough for the arms to move freely.

2. Make sure Baxter is level.

3. Scroll to **Main Screen > Settings > Advanced > Calibrate Arms**; then select a form of calibration (**Basic**, **Factory**, or **Spring**).



4. Select one or both arms; then press **OK** to begin the calibration.
5. Move away from the robot, and make sure all persons are clear of Baxter.

The selected arm moves as follows for each type of calibration:

- Basic – the arm drops to the side of the robot, then rises up and extends outward.
- Factory – the arm moves to the side of the robot, then rises up and extends outward in a few different directions.
- Spring – the arm moves to the side of the robot with the elbow at the lowest point, then rises up and extends outward.



6. When the robot finishes calibrating, if a Factory calibration was performed:
 - a. Reboot the robot.
 - b. Scroll to **Main Screen > Settings > Advanced > Calibrate Arms**, and select **Spring Calibration**.

Wait for the robot to complete the Spring calibration.

7. Return Baxter to its workspace (if moved).
 8. If the grippers were removed, reinstall them.
 9. Restart the robot.
10. If the grippers were removed and reinstalled, configure and train each gripper.

If after calibration, Baxter still does not function correctly, try retraining and running the task. If the problem is not solved, contact your technical support representative.

Troubleshooting Baxter

To troubleshoot Baxter:

- If Baxter isn't behaving as expected, restart the robot, then try running the task again.
- If Baxter is having trouble accurately picking and placing objects, try calibrating the arms (see "Calibrating the Arms" on page 155 for instructions).

If this does not solve the problem, please consult the knowledge base in the Rethink Robotics customer support portal. If the problem persists, contact your authorized Rethink Robotics service provider for technical support.



Exporting Log Files

Baxter logs its daily activity and saves it into files. When an unidentified problem occurs with the robot, your technical support representative may ask you to export log files to a FAT 32 USB device, download them to a PC, and upload the files to Rethink Robotics support. The files may help them to troubleshoot and fix the problem.

To export a log file:

1. Touch one of the four buttons on the robot to open the button bar.
2. Select **Setting > Advanced > Diagnostics**.
3. Insert a formatted FAT 32 USB device into Baxter. Make sure there is at least 2 GB of available space on the device.
4. Select **Export Logs to USB > Export**.

The log files copy to the USB device.

To retrieve the files from the USB device and email them to Rethink Robotics, insert the USB device into a PC, copy the files from the device to the PC, then attach the files to an email message, and send the message to the requester.

Enjoy!

This concludes the core content of the *Baxter User Guide*. Please refer regularly to our wiki, located here:

mfg.rethinkrobotics.com

On behalf of the entire team at Rethink Robotics, we wish you great success with your Baxter robot, and hope you find it to be a valuable solution for your business.

Several appendices follow, with more detailed examples of information contained in this document.



Appendix A: Glossary

Glossary

- Action Control - The parameter to tell the robot when the gripper should actuate.
- Actions - A Pick, a Place, or a Hold.
- Approach Point - The pose that immediately precedes an action.
- Custom Path - A path for which you define the specific waypoints and associated poses along the robot's path.
- Default Path - A path in which you allow the robot to create waypoints along the path automatically.
- GUI - Graphical User Interface. The robot's "head" display, where you see the robot's menus, eye expressions, modify screens, task map, etc.
- Homing screen - (Sawyer robot only) The screen displayed after Sawyer has booted up. The arm then performs a Homing Sequence so the robot can recognize where each joint is in real space. During this sequence, each joint will move approximately 5 degrees.
- Motion Preset - The parameter that defines how precisely the robot's arm follows a path's waypoints.
- Path - The arm movement between two actions.
- Pose - A position and orientation of the robot's arm at a location.
- Retract Point - The pose that immediately follows an action.
- Subtask - One or more actions that may be grouped together and performed in sequence.
- Task - Made up of one or many subtasks on the robot's arm (or arms). A Task includes everything the robot is trained to perform.
- Task map - A top down view of the robot's workspace, often known as the "work envelope."
- UI - User Interface. The means by which you interact with the robot's software, for example, by entering weights and lengths, scrolling through screens, making selections, etc.
- Waypoint - A location in space that the arm will move to along a path.



Appendix B: Configuring External Devices

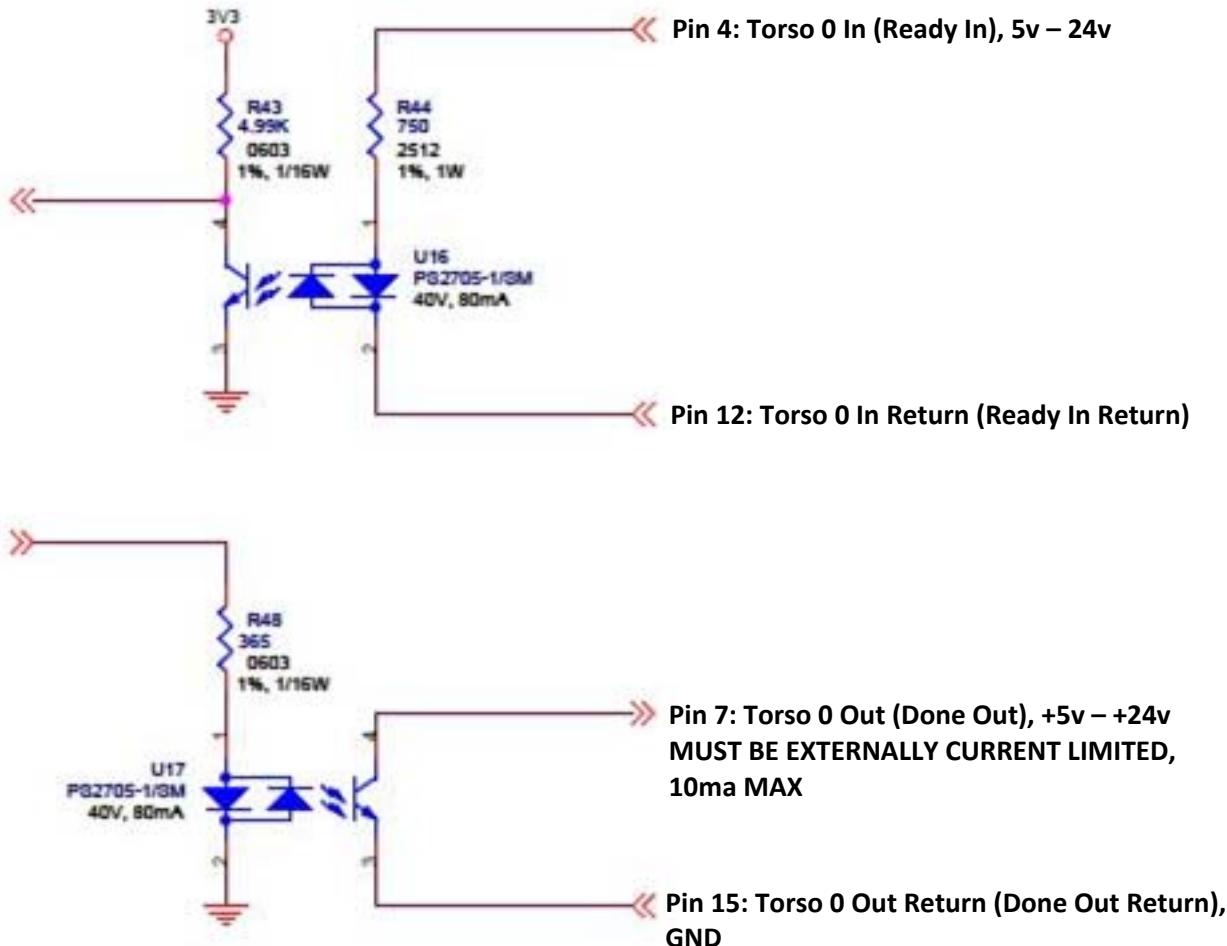
Note: For the latest information, please refer to our wiki at: mfg.rethinkrobotics.com

Baxter communicates to external machines via its open logic ports in one integrated DB15 connector or by adding an external Ethernet-connected Modbus Remote Terminal Unit (RTU).

Attach a terminal block to the DB15 connector to accommodate two signal lines—one in and one out—and provide lines for the e-stop and a safety mat.

TABLE 1. DB15 Pinouts

DB15 Pin		
1	E-Stop	E Stop Switch
9	E-Stop	E Stop Switch
2	Safety Mat	To +5 (6) via safety mat cable
10	Safety Mat	To GND (8) for safety mat
3	Safety Mat	To +5 (5) for safety mat
11	Safety Mat	To GND (13) via safety mat cable
4	Torso 0 In	Torso 0 In (Ready In)
12	Torso 0 In Return	Torso 0 In Return (Ready In Return)
5	1K to +5	To (3) for safety mat
13	GND	To (11) via safety mat cable
6	1K to +5	To (2) via safety mat cable
14	GND	
7	Torso 0 Out	Torso 0 Out (Done Out), +5v - +24v, MUST BE LIMITED TO 10ma MAX
15	Torso 0 Out Return	Torso 0 Out Return (Done Out Return), GND
8	GND	To (10) for safety mat



Note:

- Torso 0 In is FALSE when current is flowing, and TRUE otherwise. A disconnected input will appear as TRUE.
- Torso 0 Out is open when FALSE, and closed (current flowing) when TRUE.



Connecting a Modbus Remote Terminal Unit (RTU) to Baxter

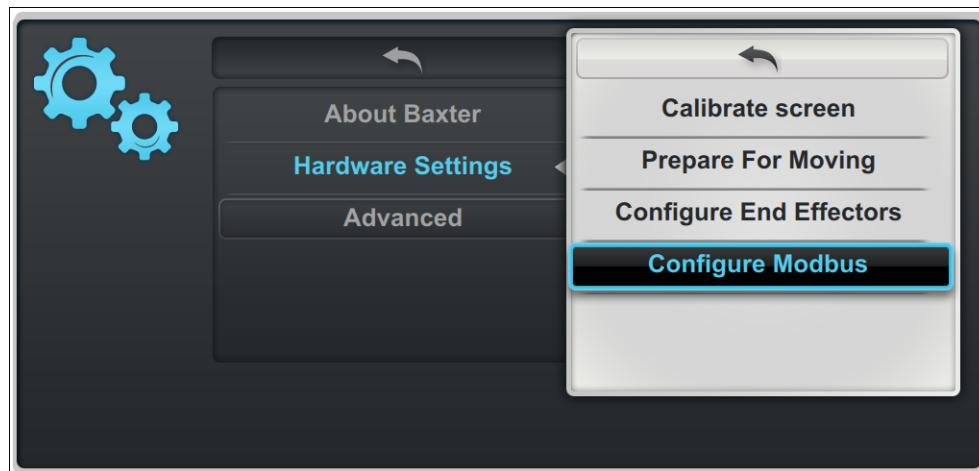
1. Set the IP address of the Modbus device to:

169.254.#.#

where # is any number between and including 1 to 254, and the subnet mask is 255.255.0.0. (Please refer to the RTU manufacturer's instructions for assigning the IP address.)

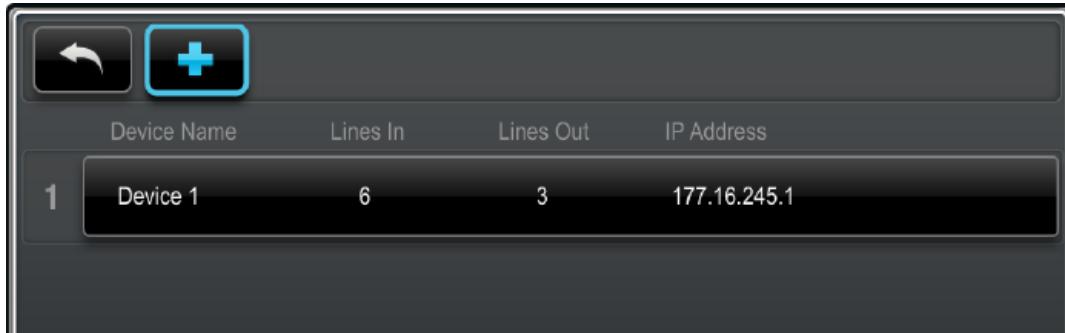
Note: With Baxter 1.1 software, Rethink Robotics qualified the Moxa ioLogik E1212 Remote Terminal Unit.

2. With Baxter shut down and powered off, attach the device to the Ethernet port.
3. Turn on the device.
4. Power Baxter on.
5. Go to **Settings > Hardware Settings > Configure Modbus.**





- In the device list screen, select the + icon to add a device.



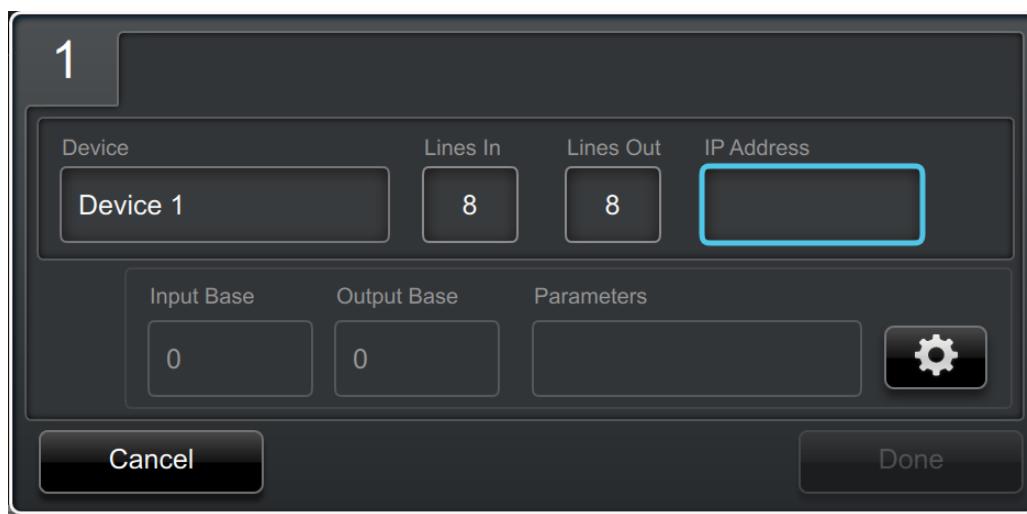
- In the configure new device screen, enter the following:

- Device Name – Keep the default, or enter a descriptive name for the device (maximum 16 characters).

Tip: Create an easily recognizable name. This makes it easier to identify when creating and adding signals to a task.

- Number of Lines In and Lines Out
- IP Address

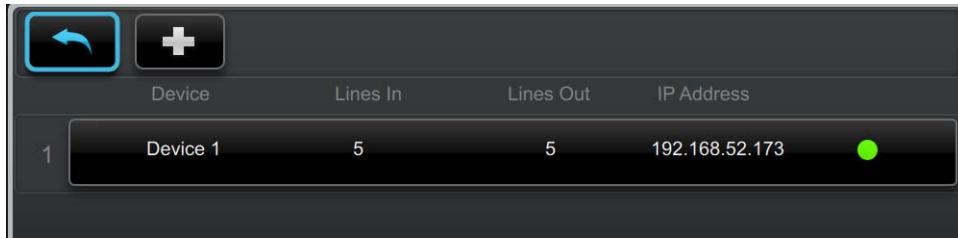
To change the default Input and Output Base of 0, and parameters, select the Advanced icon.



- Click **Done** to save the new device.



When you have a successful connection to a Modbus device, the UI displays a green indicator:



Additional Information About the Modbus RTU

Please note the following when setting up signals:

- For ModBus digital (binary) signals, Baxter uses function code 2 (Read Discrete Inputs) for reads and function code 5 (Write Single Coil) for writes.
- The ModBus device must be configured in SLAVE mode. When using settings other than the default, you must set the input and output base addresses in the Baxter configuration panel if other than 0/0, and the inputs and outputs must be digital (discrete coils, not registers) and contiguous.



Appendix C: Support & Warranty

The Baxter robot comes with a one (1) year limited warranty.

If there is a problem with your robot and you are unable to resolve it, try shutting down and restarting the robot. If the problem persists, contact your authorized Rethink Robotics service provider for technical support. You will need to provide the model and serial number of the robot experiencing the problem. These can be found on the back of the robot near the power button.

If the product is no longer within the warranty period, the authorized Rethink Robotics service provider will provide an estimate of the technical support or repair costs.

