

---

# Pick[+] client guide

Pick[+] version 1.3

Start guide for the Pick[+] client

---

This document serves as a brief guide on how to use *Pick[+]* from the client application. Its structure responds to the main actions the user can perform from it, always in the context of *Pick & Place* applications.

There are several video tutorials that cover some of the most common functionalities of *Pick[+]* and that serve as a complement to this document. They can be found in the following link: <https://www.bitmetrics.es/resources>

Before getting started with this guide, a **brief reminder on client execution**.<sup>1</sup>

## Start client

There are two possibilities to start client application:

1. *Standalone* client. Client that is pre-installed (or can be installed) in the server PC.
2. External client. Client installed in the user PC.

In both cases, the client consists of an executable file inside a packaged client folder that is dependent on the operating system it is intended to be executed.

In order to start client make sure that:

- server PC is on,
- (if external) client PC is connected via ethernet in the socket that is dedicated to it,
- (if external) ethernet adapter of client PC used for communication with the server PC is configured with the IP address 200.200.1.1 (or any other in the same

---

<sup>1</sup> See *Installation guide* for more details on client execution and other installation aspects necessary to use *Pick[+]*. Also, see <https://www.bitmetrics.es/resources>, video titled *Install Pick[+]*, for an alternative explanation and examples.

---

network, with mask 255.255.255.0, except for the address 200.200.1.100—the server PC address).

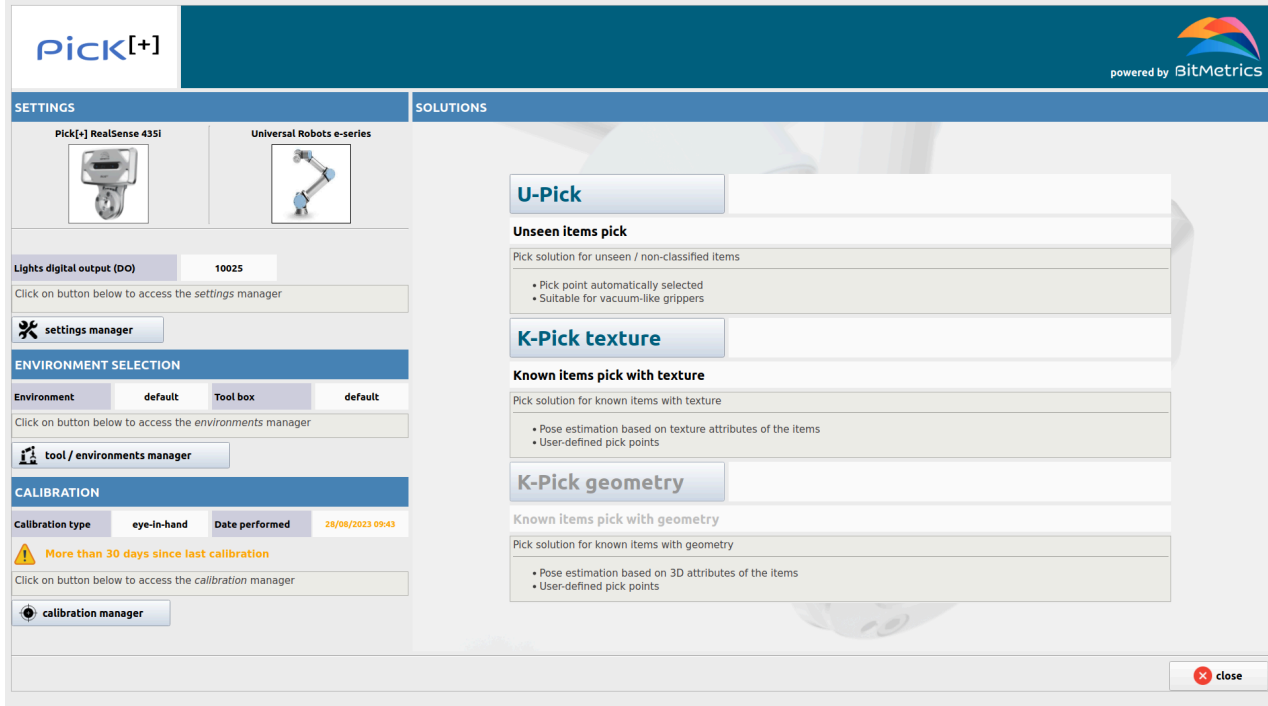
To start the client application (if there is no shortcut available in the desktop), execute the *executable* file (*pickplus.exe* for Windows or *pickplus.sh* for Ubuntu) that is contained in the client folder named *pickplus*.

If everything goes well you will see the introduction pop-up indicating success in connectivity to the *Pick[+]* server that will later lead to the selection page (application landing page).



## Main concepts / landing page

The application landing page or *selection page* presents the user with the main functionalities of *Pick[+]*.



We have two main functionality blocks:

- **Configuration block.** It contains the functions under the following headings:
  - *settings.* Hardware settings and selection (camera and robot selection, robot digital outputs and camera-robot setting),
  - *environment and tool selection.* Definition and selection of sets of pick boxes and robot tools.
  - *calibration.* Camera-robot calibration.
- **Solutions block.** It contains the solutions available (by default in the standard bundle) under the current version V1.3:
  - *U-Pick.* Solution for unseen items (i.e., not *known* by the user),
  - *K-Pick texture.* Solution for items that are known by the user and can be scanned with the help of the robot and camera,

- 
- \**K-Pick geometry*. Not available in V1.3.

In essence, actions and setting options under the configuration block are not solution or application dependent. Any application we can create and execute will assume global *Pick[+]* configuration (i.e., setting values, calibration and environment / tool selection).

Solutions are all based on a first stage consisting of image segmentation. In *U-Pick* such segmentation is performed assuming all objects belong to the same generic category (*object*). In *K-Pick* instances are classified into the categories defined by the user. In *K-Pick texture* the identification of all instances in the pick scene is made based on texture (i.e., RGB) attributes whereas in *K-Pick geometry* the identification is based on geometric attributes derived from the point clouds captured by the camera.

Unlike *U-Pick*, *K-Pick* builds upon knowledge of the objects involved in the pick application. Information about the objects is provided by the user by means of the scan process: the user places the object in a particular position over a scanning mat in order for the robot, with camera in hand, to perform a scanning trajectory around it. From this scan two data types can be obtained:

- RGB images
- Point clouds

Instance segmentation models are learnt from scan-based RGB images (in the case of the solution *K-pick texture*). Point clouds are used to build 2.5D models that can then be used for pose estimation of the items in the pick scene.

---

## What do you want to do? (*main actions*)

In this section we address some of the main actions that can be performed by the user.

### Solution settings

- **Change application settings**  
Change hardware settings. Go to [Application settings](#)
- **Define / edit application environment**  
Define / edit boxes or containers in the picking environment. Go to [Application environment](#)
- **Define / edit tool bounding box**  
Define / edit tool collision boxes for the different robot actuators. Go to [Tool collisions box](#)
- **Calibrate camera-robot**  
Set camera-robot calibration, *eye-in-hand* or *hand-eye*. Go to [Calibration](#)

### Data management

- **Add object sample to data repository**  
Scan object to generate sample and save it in samples repository. Go to [Add sample](#)
- **Delete data from repository**  
Manage user data repositories. Go to [Data management](#)
- **Inspect / delete old applications**  
Manage user applications. Go to [Applications](#)
- **Create / upload 3D sample models**  
Create 2.5D views from scanned objects (from captured point clouds) or upload 3D models for object alignment. Go to [Models](#)
- **Create / edit pick points**  
Define / edit picking points on 2.5D or 3D models previously created or uploaded. Go to [Picking points](#)

---

## Application creation and execution

- **Execute U-Pick**  
Execute pre-trained *U-Pick* solution for the *unseen / unknown* context. Go to [Execute application](#)
- **Create new K-Pick application**  
Create new *K-Pick* application from scanned samples of the objects involved in the application. Go to [Create new K-Pick application](#)
- **Execute existing K-Pick application**  
Execute K-Pick application in repository (i.e. previously created). Go to [Execute application](#)

## Logs and alarms

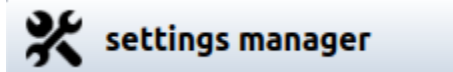
- **Access application logs**  
Access both server and client logs. Go to [Logs](#)
- **Interpret alarm pop-up**  
Brief description of the alarm codes. Go to [Alarms](#)

## Solution settings



### Application settings<sup>2</sup>

We can edit / modify the camera setting and the lights digital output (on which lights are connected).<sup>3</sup> Some other hardware options (like e.g., camera and robot manufacturer) are disabled for the default bundle of V1.3.

To modify settings click on



in the selection page. By doing so we will move to the settings page.

CALIBRATION	ROBOT	CAMERA	LIGHTS
Select hardware setting <ul style="list-style-type: none"> <li>• Camera in robot arm: <i>eye-in-hand</i></li> <li>• Camera away from robot: <i>hand-eye</i></li> </ul> <input checked="" type="radio"/> Eye-in-hand <input type="radio"/> Hand-eye	Universal Robots e-series  Select UR e-series robot <ul style="list-style-type: none"> <li><input checked="" type="radio"/> UR3</li> <li><input type="radio"/> other</li> </ul>	Pick[+] RealSense 435i  Select camera from drop-down menu <input type="text" value="RS435i"/>	Select robot digital output where light source is connected from drop-down menu If light source is not connected to robot select --- none selected --- <input type="text" value=""/>
<input type="button" value="save changes"/>			

In the settings page we have three configurable options:

- **Calibration / camera setting**

<sup>2</sup> See <https://www.bitmetrics.es/resources>, video titled *Configure Pick[+]*, for an alternative explanation and examples.

<sup>3</sup> In future releases there will be more robot and camera options to select from.

---


We can choose between eye-in-hand (camera mounted in the robot) or hand-eye (external camera mounted).

- **Robot**

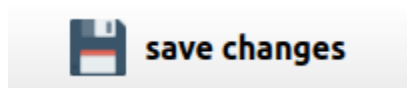
We can choose between UR3 and any other e-series model

- **Lights**

We need to select the robot digital output (DO) to which lights are connected (an integer between 0 and 7).

 It is possible to work with lights disconnected but results both when preparing and / or executing the application might not be optimal.

When done click on



Once changes are saved we can return to the main selection page or close application.



## Application environment

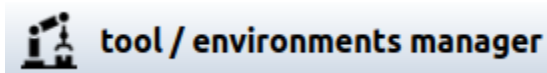
We define application environments by means of sets of 3D containers or boxes from which we wish to perform a *pick* action.

Definition of containers is particularly useful in order to assess possible collisions at the time of assessing a *pick* trajectory.




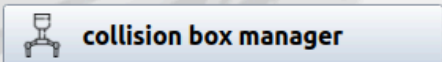
**NOTE:** It is possible to work with the default environment which is, essentially, an empty environment (i.e., without boxes).



To define / edit an environment click on



to move to the environments page.

PICK ENVIRONMENTS	TOOL COLLISION BOXES
	
<p>Access <i>environments manager</i></p> <p>Actions</p> <ul style="list-style-type: none"> <li>• <b>Create</b> or <b>delete</b> 3D containers</li> <li>• <b>Create, edit, select</b> as active or <b>delete</b> environment</li> </ul> <p><i>Note: an environment can be composed by a single container. To define picking contexts environments are selected, not directly containers</i></p>	<p>Access <i>tool collision box manager</i></p> <p>Actions</p> <ul style="list-style-type: none"> <li>• <b>Create, select</b> as active or <b>delete</b> tool collision bounding box</li> </ul>
	

Click on



to open the environments manager pop-up. We are ready to define / edit environments and / or boxes.

## Boxes

To define / edit a new box from which we wish to perform *pick* actions we need to select the box / container manager radio button under the Actions heading.

### ACTIONS

Choose between box and environment manager

**Box / container manager**

**Pick environment manager**

### Create new box

Boxes are defined as a set of 3D coordinates with respect to the robot base that form a 3D rectangle. To define the rectangle we need to provide four points: **origin**, **p1**, **p2** and **p3**.

In order to proceed with the definition of a new box, select the radio button Create new box under the heading Actions on boxes.

#### Actions on boxes

- **Create** new box manually or with the help of the robot
- Select existing box to **edit** or **remove** from repository

*Note: only boxes that do not belong to any pick environment can be deleted*

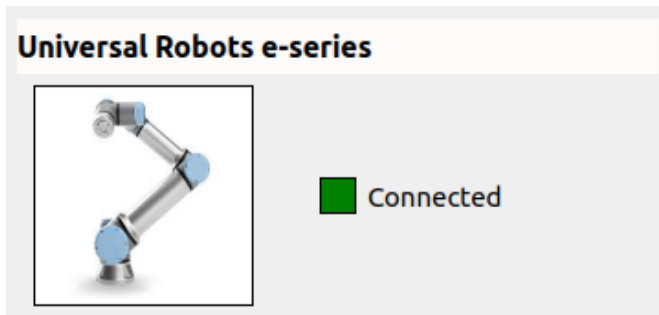
**Create new box**       **Edit existing box**

Select an ID for the new box

**Box ID**

and provide the four coordinates that are necessary to define it. We can do so fully manually or assisted by the robot.

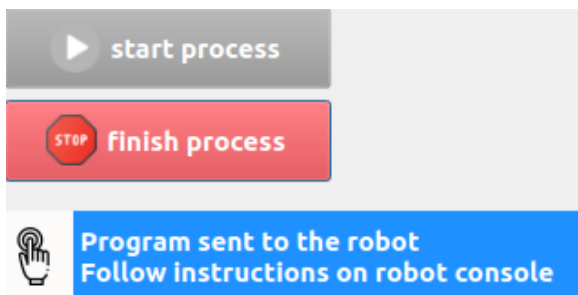
To proceed with the robot assistance first make sure that the robot is communicating with the *Pick[+]* server.



Click on



to send script to the robot and start the coordinate-recording process. Once the script is sent follow instructions on the robot console.



The robot will ask you to provide the four required coordinates sequentially. To provide a coordinate we need to place the robot TCP on the desired coordinate and confirm (the robot will be, along the process, in *free-drive* mode).

**⚠ WARNING:** Make sure that the robot TCP is well defined before proceeding.

This process can be stopped any time or automatically finished after providing the fourth coordinate.

---

Coordinates provided by the robot can be manually edited any time before saving. Once finished, click on



## Edit box

To edit an existing box in the data repository first select the radio button Edit existing box under the heading Actions on boxes.

**Actions on boxes**

- **Create** new box manually or with the help of the robot
- Select existing box to **edit** or **remove** from repository

*Note: only boxes that do not belong to any pick environment can be deleted*

Create new box  Edit existing box

Select the box you want to edit or delete from the drop-down menu.

Select box from list below

---- None selected ---- ▾

Click on

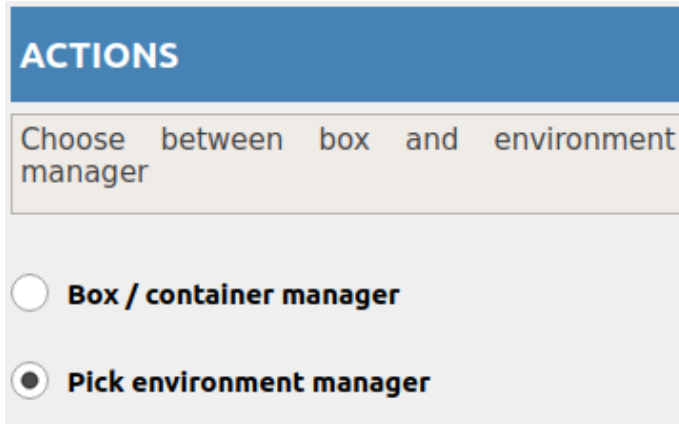


respectively to save changes or delete the box.

**⚠ WARNING:** Only boxes not included in an environment can be deleted.

## Environments

To define / edit an environment select the radio button Pick environment manager under the Actions heading.



**ACTIONS**

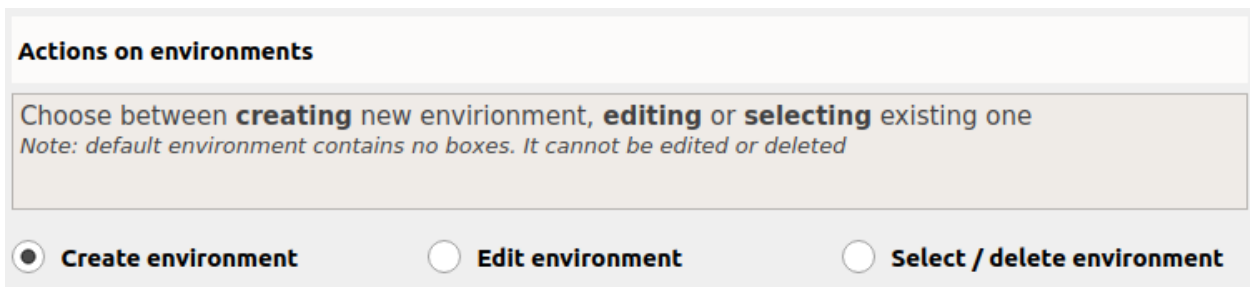
Choose between box and environment manager

Box / container manager

Pick environment manager

### Create new environment

To create a new environment select the radio button Create environment under the heading Actions on environments.




**Actions on environments**

Choose between **creating** new environment, **editing** or **selecting** existing one  
*Note: default environment contains no boxes. It cannot be edited or deleted*

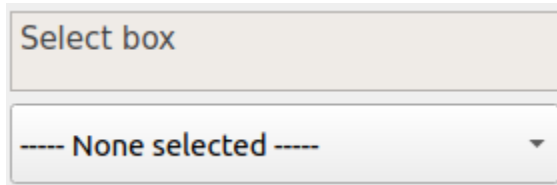
Create environment       Edit environment       Select / delete environment

First of all, provide an ID for the new environment and verify it.



---

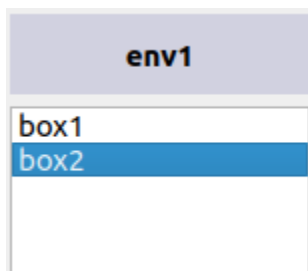
In order to add a box to the newly created environment we need to select it from the drop-down menu.



Once selected we need to click on



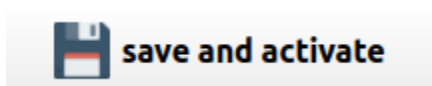
to add box to environment. To remove a box previously added select it from the list view



and click on



Save environment box into the data repository by clicking on



**⚠ WARNING:** By saving the current environment we automatically select it as the active one.

## Edit environment

To edit an existing environment select the radio button Edit environment under the heading Actions on environments.

**Actions on environments**

Choose between **creating** new environment, **editing** or **selecting** existing one  
*Note: default environment contains no boxes. It cannot be edited or deleted*

Create environment       Edit environment       Select / delete environment

Select environment from the drop-down menu

**Select environment**

---- None selected ----

Once selected, we proceed as in the [creation of a new environment](#) by adding / removing boxes and saving it when finished.

## Select / delete environment

To activate or delete an environment select the radio button Select / delete environment under the heading Actions on environments.

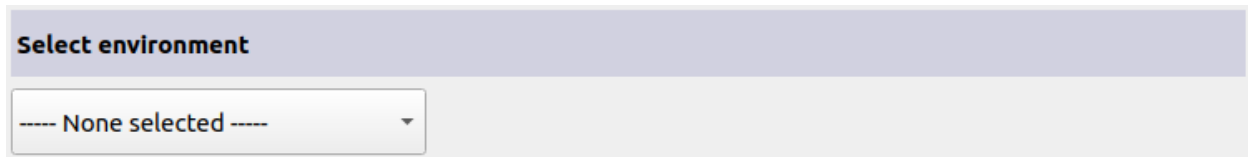
**Actions on environments**

Choose between **creating** new environment, **editing** or **selecting** existing one  
*Note: default environment contains no boxes. It cannot be edited or deleted*

Create environment       Edit environment       Select / delete environment

---

Select environment from the drop-down menu



Click on



respectively to either activate or delete the selected environment.

## Tool collisions box<sup>4</sup>

In order to deal with collisions related to the tool in *pick* trajectories we allow the user to define a 3D rectangle as the tool bounding box that will be used as the tool reference for collision assessment.

To define / edit a new tool bounding box click on




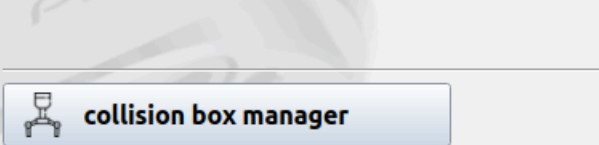


to move to the environments page.

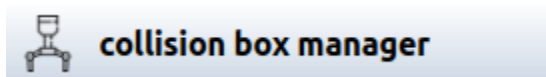
---

<sup>4</sup> See <https://www.bitmetrics.es/resources>, video titled *Tool collision box*, for an alternative explanation and examples.



PICK ENVIRONMENTS	TOOL COLLISION BOXES
	
<p>Access <i>environments</i> manager</p> <p>Actions</p> <ul style="list-style-type: none"><li>• <b>Create</b> or <b>delete</b> 3D containers</li><li>• <b>Create, edit, select</b> as active or <b>delete</b> environment</li></ul> <p><i>Note: an environment can be composed by a single container. To define picking contexts environments are selected, not directly containers</i></p>	<p>Access <i>tool collision box</i> manager</p> <p>Actions</p> <ul style="list-style-type: none"><li>• <b>Create, select</b> as active or <b>delete</b> tool collision bounding box</li></ul>
	

Click on



to open the tool collision manager pop-up. We are ready to define / edit tool bounding boxes.

## Create new tool collision box

Select radio button Create new tool collision box under the heading Actions on tool collision box.

### Actions on tool collision box

Create **new** tool collisions box in repository or select existing one to **edit, delete** or **activate**

*Note: new box will be set as active in configuration*

**Create tool collision box**

**Edit tool collision box**

Set ID and the 3D rectangle parameters to define the new collision box.

<b>Box ID</b>	<input type="text"/>
<b>Width (x)</b>	<input type="text"/> mm.
<b>Height (z)</b>	<input type="text"/> mm.
<b>Depth (y)</b>	<input type="text"/> mm.
<b>Offset (d)</b>	<input type="text"/> mm.

*Note: the 3D rectangle is defined with respect to the robot TCP. The offset needs to be a value smaller than or equal to 0 that indicates how far from the TCP we place the origin of the 3D rectangle (generally they will coincide. I.e., value 0).*

Once set click on



to save and activate the new collision box.

## Edit collision box

To edit existing collision box select the radio button Edit tool collision box under the heading Actions on tool collision box.

**Actions on tool collision box**

Create **new** tool collisions box in repository or select existing one to **edit, delete or activate**  
*Note: new box will be set as active in configuration*

Create tool collision box       Edit tool collision box

Select collision box from the drop-down menu

Select tool collisions box from repository

---- None selected ---- ▾

We can edit, delete or simply activate the selected collision box. When finished click on



according to the desired action.

## Calibration

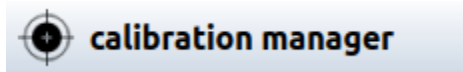
With calibration we aim at relating the camera and robot spaces. More specifically, we look for the transformation (rotation + translation) that needs to be implemented in order to express camera coordinates in terms of robot coordinates.

---

We can perform two types of robot-camera calibration:

- **Eye-in-hand calibration**<sup>5</sup>. In the case when the camera is mounted on the robot.
- **Hand-eye calibration**. In the case when the camera is outside the robot.





To proceed with calibration we need to click on



to move to the calibration page.

**⚠ WARNING:** Make sure the calibration type set is the desired one. See the [Application settings](#) section if it needs to be changed before proceeding with calibration.

Camera and robot need to be connected in order to calibrate (necessary to enable calibration button). In the case of the robot, it needs to be in remote mode so that the calibration script can be sent from the *Pick[+]* server.

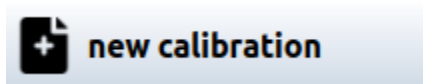
<b>Pick[+] RealSense 435i</b>	
	 Connected
<b>Universal Robots e-series</b>	
	 Connected

---

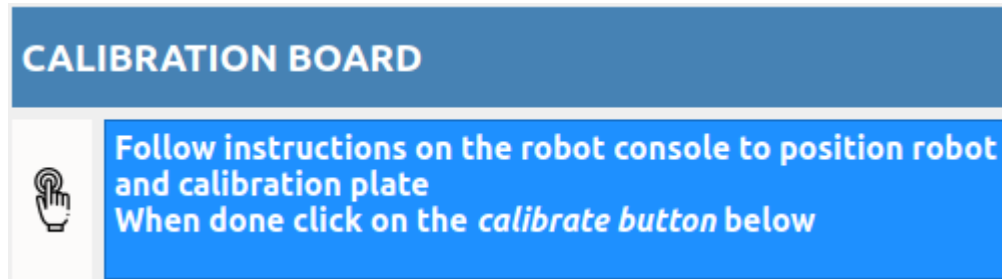
<sup>5</sup> See <https://www.bitmetrics.es/resources>, video titled *Perform eye-in-hand calibration*, for an alternative explanation and examples.

---

Click on



to proceed with the calibration process and follow the instructions on the robot console.



If successful in the process a new calibration matrix will be saved.

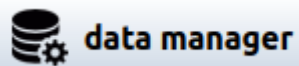
---

## Data management

To manage data in repository click on

A blue button with a white gradient background and the text "K-Pick texture" in bold blue font.

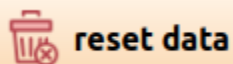
and then on

A blue button with a white gradient background. On the left is a black icon of a database cylinder with a gear. To the right is the text "data manager" in bold black font.

to access the data page.

We can manage application, sample, model and pick point data from the respective managers in a custom manner.

We can also reset and empty the whole data repository by clicking on the button

An orange button with a white gradient background. On the left is a red icon of a trash bin with a red 'X' over it. To the right is the text "reset data" in bold black font.

By doing so we will delete all custom data and reset the repository as it was during installation.

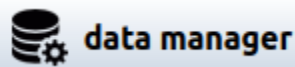
## Add sample<sup>6</sup>

By **sample** we mean the collection of images / data created during the scanning process that belong to a particular item or object. We can have different samples in our data repository that can later be used in the design of different applications.

To proceed with the addition or edition of a sample we click on



and



to move to the data page. Click on



to open the samples manager pop-up.

**NOTE:** There is an alternative way to access the samples manager via the application creation page.

## Create new sample

To create a new sample select the radio button create sample under the heading Actions.

---

<sup>6</sup> See <https://www.bitmetrics.es/resources>, video titled *K-pick: scan objects*, for an alternative explanation and examples.

### ACTIONS

Select between **creating** new sample or **editing / deleting** existing one

**Create sample**

**Inspect / edit sample**

Enter and verify the new sample ID.

### Enter sample name

Choose new sample name

Next we need to indicate the type of sample / object we are dealing with. Samples can present variability in two ways:

- **Variable geometry:** the object does not have a 3D model. Different instances of the same object are not identical in shape (e.g., organic items).
- **Variable texture:** Different instances of the same object are not identical in terms of texture (e.g., Trident boxes of different flavors).

## Views

To grasp geometry we capture what we call **views**. They consist of collections of RGB images and point clouds that refer to a surface section (i.e., a view from the camera) of the scanned object.

Each view allows us to produce a 2.5D model of the object from the point clouds and robot poses. As such (2.5D), it does not capture the whole surface of the object. To do so we will need to add extra views.

We will have as many 2.5D models of the objects as views we have.



---

**⚠ WARNING:** Note that, in order to work in a 3D environment (i.e., a 3D application), we need to make sure that the views generated will cover the whole surface of the object.

## Textures

To grasp different textures of an object we capture what we call **textures**. Textures need to be representative of all the variability that will be found in the production environment (in the data augmentation phase during training of the instance segmentation model new cases will be produced by interpolation of the provided examples).

**⚠ WARNING:** Only textures not covered by views need to be added. Texture images produced during the scan of a view are also used.

If the object does not present variability in geometry we will require at least one view. If it presents variability in geometry we will require only textures (at least one texture).

**Variable** if the object can present variations in shape and/or appearance or texture (e.g., an organic object).  
**Invariable** otherwise (e.g., a mechanical part)

**variable**       **invariable**

**Does the object present variability in shape?**

**yes, its geometry may vary**       **no**

When done with the choice between variability / invariability click on



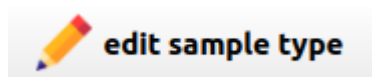
At this point we can edit the type of sample we are creating. Samples can be

- **2D:** samples correspond to one face (or a few faces) of the object. They do not cover the whole surface of the object. They can only be used for 2D applications.

- **3D**: samples cover the whole surface of the object. They can be used both for 2D and 3D applications.

Sample type	2D
Click on button below to <b>edit sample type</b> <i>Note: by default sample is assumed 2D (i.e., views do not cover whole surface of the object)</i>	

By default samples are considered 2D. Click on



to edit sample type.



What type of sample do you have?
Is the set of sample views <b>partial</b> (cover object surface partially) or <b>complete</b> (cover whole object surface)? <i>Note: mark/update after all views have been added</i>
<input checked="" type="checkbox"/> <b>partial (2D)</b> <input type="checkbox"/> <b>complete (3D)</b>

Select the desired type and click on



## Scan for views and textures

To scan first make sure that camera and robot are connected (and that robot is in remote mode).

**Pick[+] RealSense 435i** Connected**Universal Robots e-series** Connected

To start scan process click on



depending on whether we are scanning a texture or a view.

Follow instructions on the robot console to place the camera-robot in the right way with respect to the scanning mat.



**Move robot until image marks fit scanning mat corners  
Object to be scanned must lie at the center of the mat  
Click on the 'start scan' button when finished**

When done click on



and from then on follow instructions on the robot console.

---

When all desired textures and / or views have been produced click on

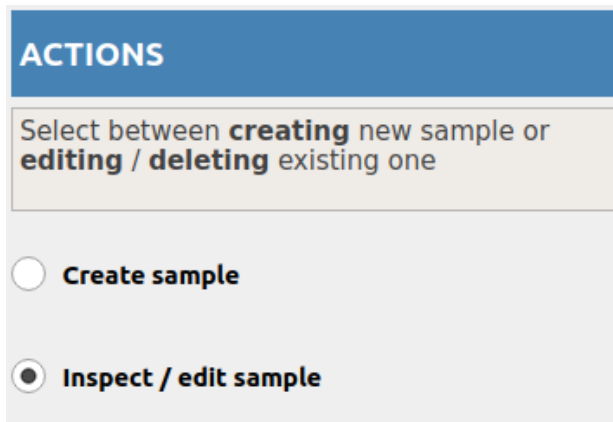


To abandon scanning process any time click on

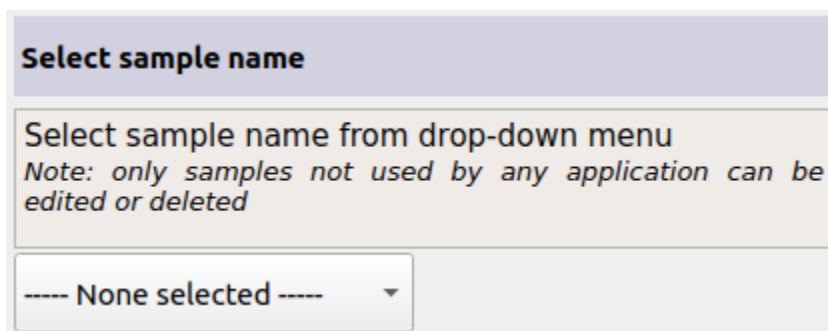


## Edit sample

To edit or inspect a sample select the radio button Edit / inspect sample under the heading Actions .

A screenshot of a software interface panel titled "ACTIONS" in a blue header. Below the header, there is a light beige box containing the text "Select between **creating** new sample or **editing / deleting** existing one". Below this box, there are two radio button options: "Create sample" (which is unselected) and "Inspect / edit sample" (which is selected with a black dot).

Select sample from the drop-down menu

A screenshot of a software interface panel titled "Select sample name" in a grey header. Below the header, there is a light beige box containing the text "Select sample name from drop-down menu" and a note: "Note: only samples not used by any application can be edited or deleted". Below this box, there is a white drop-down menu with the text "---- None selected ----" and a downward-pointing arrow.

---

and proceed as with the creation of a new sample to edit sample features and to add / remove views and textures.

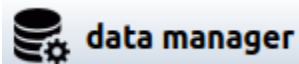
## Applications


To access applications first click on



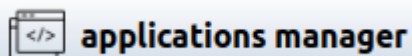
**K-Pick texture**


and then click on



 **data manager**

To open the applications manager pop up click on



 **applications manager**

Select application you wish to inspect / delete

### Applications repository

#### Applications manager

##### Delete application

Click on button below to **remove** selected application

- Trident\_big
- Woods**
- app1
- battery\_app

<b>Application type</b>		3D			
<b>Module</b>		texture			
<b>Date created</b>		day	05/07/2022	hour	19:07
<b>Last time executed</b>		day	06/07/2022	hour	10:59
<b>Prediction model</b>		<input checked="" type="checkbox"/>			
<b>Categories</b>					
<b>Positive</b>			<b>Negative</b>		
Cherry_cat	<b>Samples</b>			<b>Samples</b>	
Pine_cat					
<b>Ring_cat</b>	Ring				

**delete**

**close**

To delete from data repository click on



## Models<sup>7</sup>

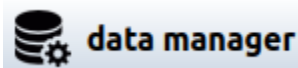
In order to deal with object alignment and thus allow for a precise place position of the picked object we need the 2.5 / 3D model of the object. To align, we offer two possibilities:

- Work with the 2.5D models that can be generated from the scanned views.
- Work with a 3D model provided by the user.

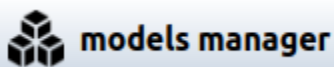
To allow any of these two options we need to open the `models` manager. To do do, click on



and then click on



To open the `models` manager pop-up click on



### Add / delete 3D model

To add / delete a 3D model select the radio button `Add / delete model` under the heading `Manager` selection.

---

<sup>7</sup> See <https://www.bitmetrics.es/resources>, video titled *K-pick: generate models*, for an alternative explanation and examples.

## MANAGER SELECTION

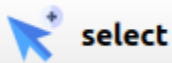
- Add / delete model**
- Generate model**
- Link / unlink model to sample**

To upload a new 3D model to the data repository first select the units with which the model to be uploaded has been generated.

### Model metric units

---- No unit selected ---- ▾

Next select model from local repository by clicking on



When selected click on



To delete model from repository select it from the list of available models

### Models in repository

Trident



---

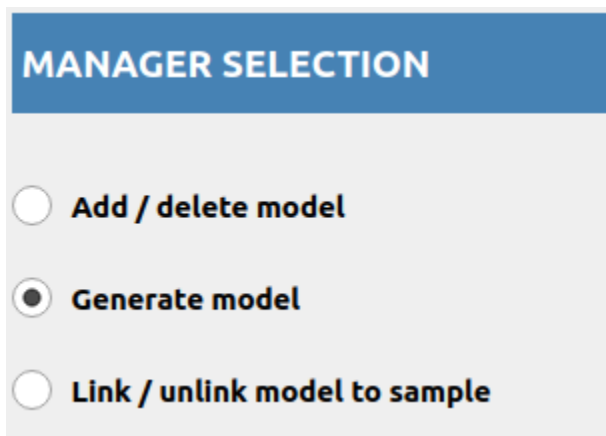
and click on



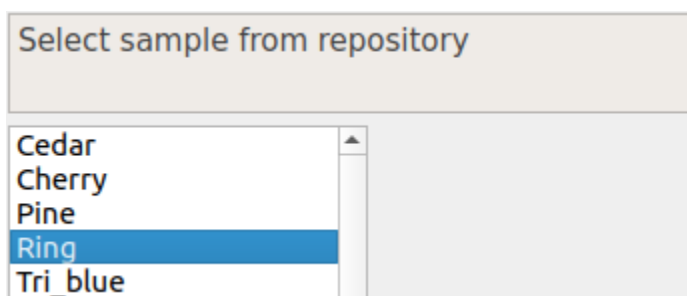
## Generate model

In the lack of 3D models for the samples we deal with we can generate 2.5D models from the scanned views.

To generate models from sample views select radio button **Generate model** under the heading **Manager selection**.

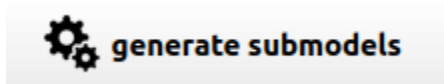


Select sample from list



---

and click on

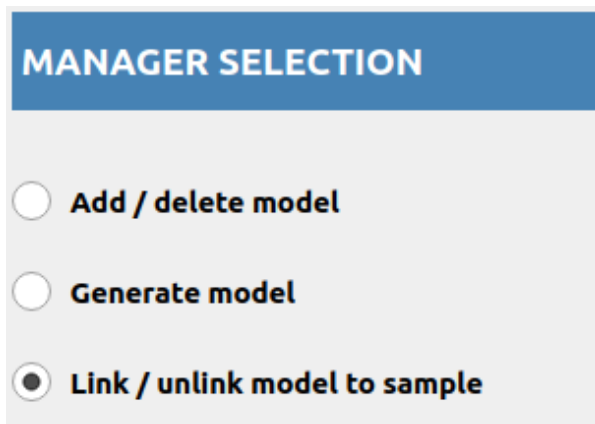


2.5 view models will be automatically generated for the selected sample.

## Link / unlink 3D model to sample

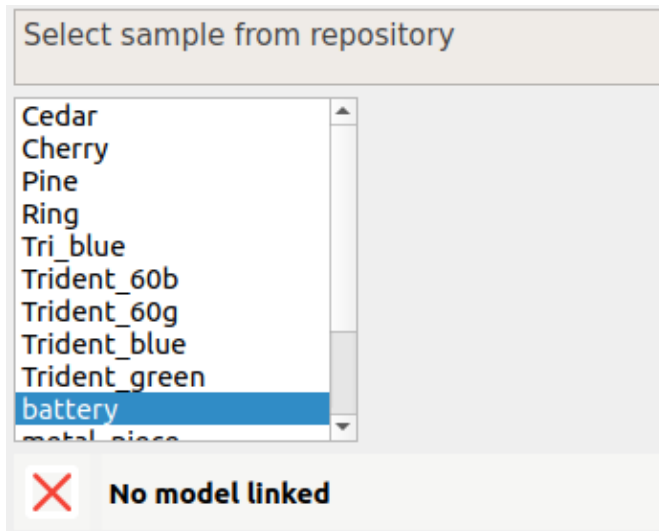
3D models are not linked by default to a particular sample / object. In fact, the same 3D model can be potentially linked to more than one sample. To indicate this relationship between the model and a particular sample we need to explicitly *link* the model to the sample.

To link / unlink 3D model to / from sample select radio button Link / unlink model to sample under the heading Manager selection.

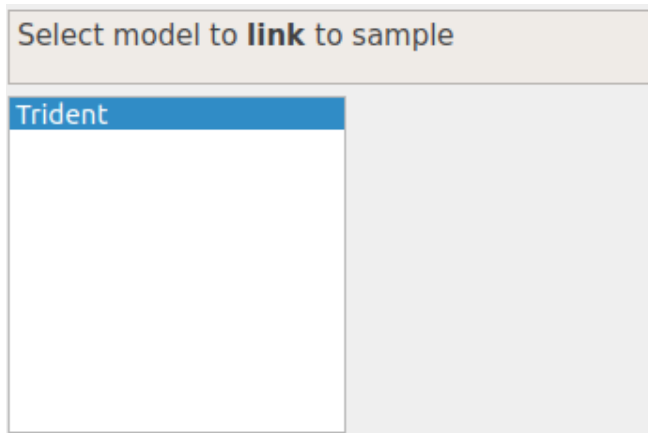


---

First of all, select sample from list



If the selected sample is not linked to any 3D model we will be shown a list with all the models available.

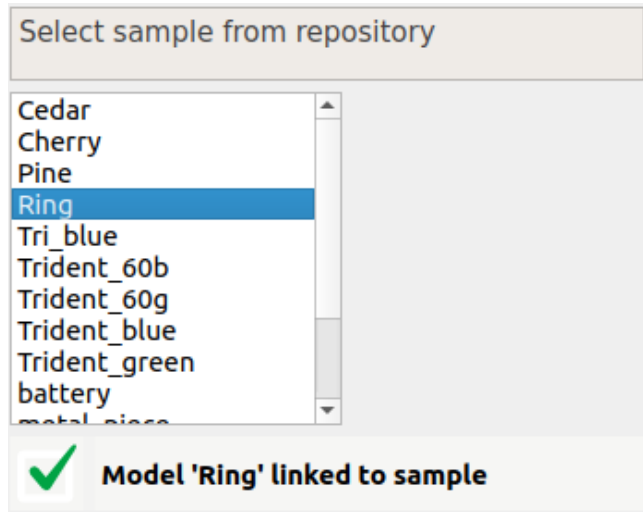


Select desired model and click on



to link the selected model to the selected sample.

If, on the other hand, the selected sample is already linked to a model, we first need to *unlink* it before linking it to another model.



To *unlink* click on



## Picking points<sup>8</sup>

We can select pick points on 2.5D and 3D object models to grab the item. By doing so we enable the user to define oriented, precise place positions for the items.

In a 3D bin picking context it is advisable to select a sufficient number of picking points on the object in order to maximize the likelihood that, in the face of occlusions in the candidate object to be picked, there will be at least one feasible picking point. For example, if we are dealing with 2.5D models generated from the scanned views, we could select at least one picking point per view to maximize the likelihood of the presence of feasible picking points in the scene. Likewise, if we are dealing with revolution solids it is advisable to define picking points as revolution points (see [Revolution points](#) below).

<sup>8</sup> See <https://www.bitmetrics.es/resources>, video titled *K-pick: picking points*, for an alternative explanation and examples.

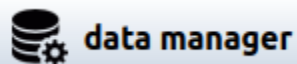
---


To proceed click on



**K-Pick texture**

and then click on



 **data manager**

to access the data page. Once in it click on



 **points manager**

to open points manager pop-up.

## Define new picking points set

To define a new set of pick points select the radio button Define new picking points set under the heading Actions.

### ACTIONS

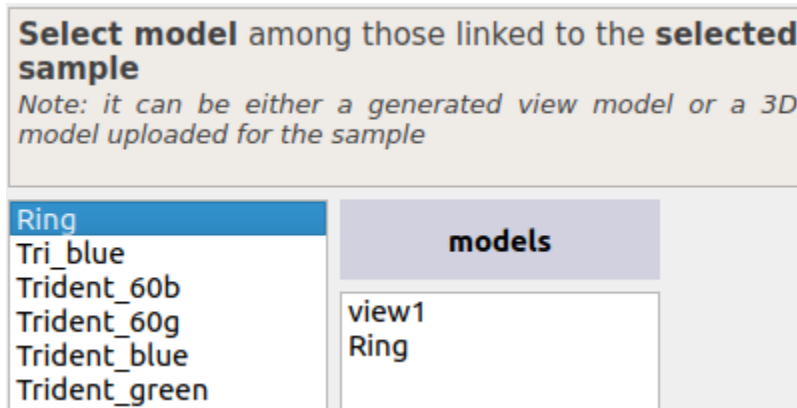
Select between **defining new** picking points set on 3D model, **editing** existing ones or **linking / unlinking** picking points sets to / from models

**Define new picking points set**

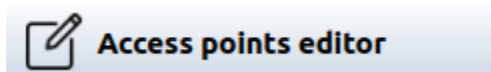
**Link / edit picking points set**

First we select the sample for which we want to define a new set of picking points. In the **Models** list we will see all the models that are linked to the selected sample (if any): we can have a 3D model uploaded from the local repository that is linked to the selected sample and the 2.5D models generated from the scanned views of the sample.

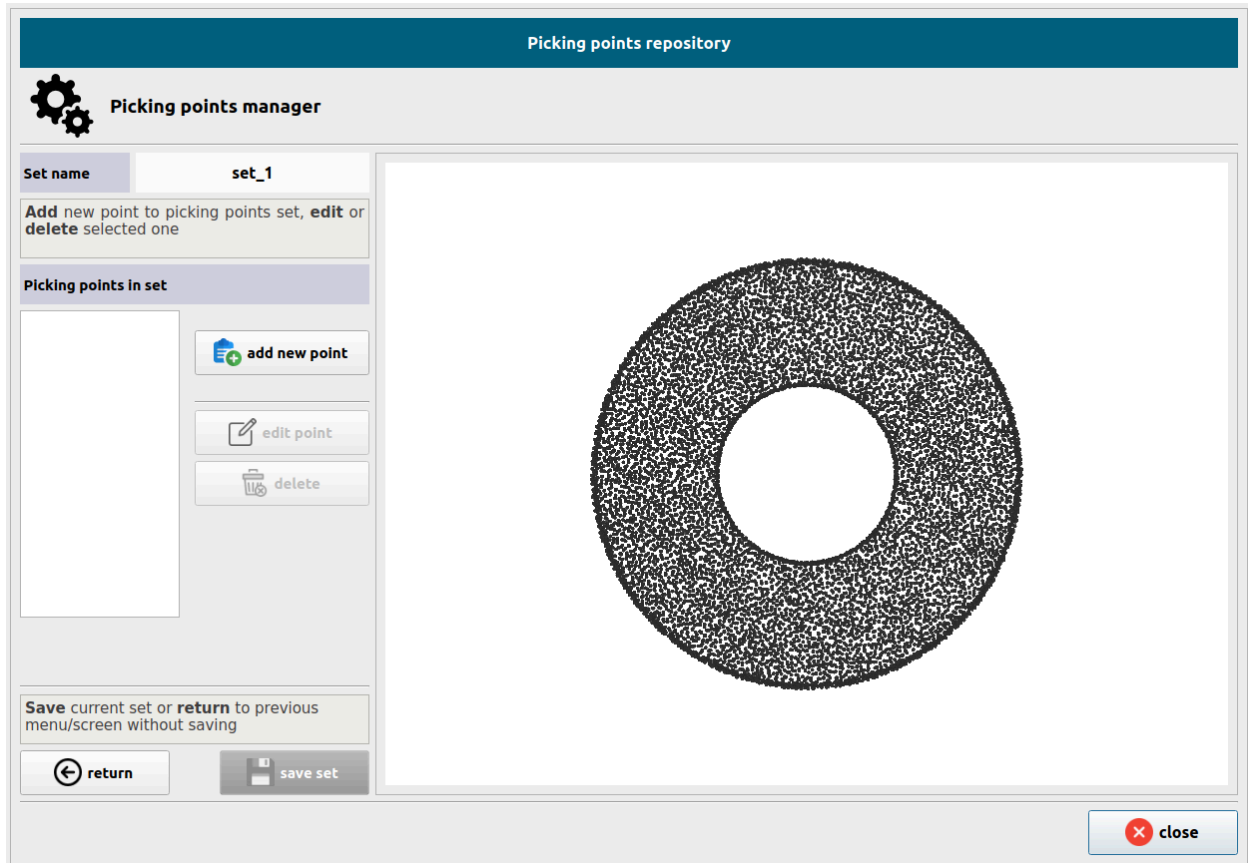
*Note: go to [Models](#) to see how to link 3D models to a sample and / or how to generate 2.5D models from views.*



We can define picking point sets for all the models independently. To proceed select one model from the list and click on



to open the pick point editor pop-up.



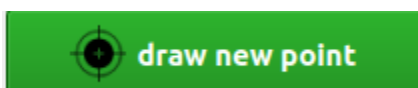
Picking points sets are named automatically: the first set we define for a given model will be named `set_1`, the second set will be named `set_2` and so on. Names are not editable.

To add a picking point to the newly created set click on

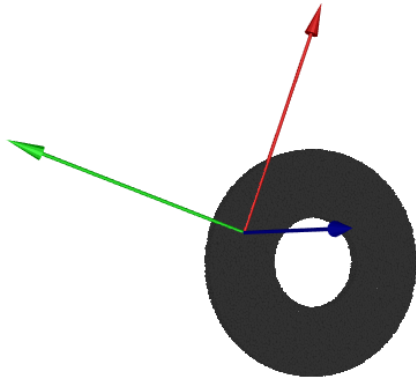


To place a new point on the model we can:

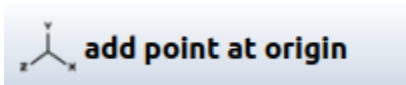
1. Click on



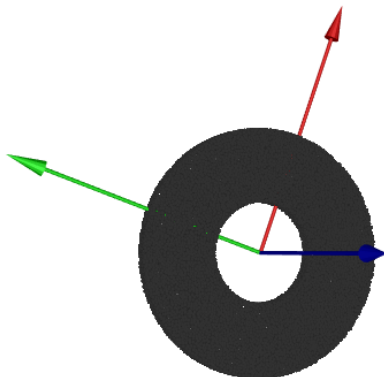
and then click on the model around the area where we want to define the new picking point (it can be refined afterwards)



2. Click on



And place a new picking point at the origin / center of mass of the model (it can be refined later)



To refine / adjust point click on





### Refine picking point

Refine picking point on the model by adjusting its **position** and **orientation** values

<b>X</b>	0,00
<b>Y</b>	0,00
<b>Z</b>	0,00
<b>RX</b>	0,00
<b>RY</b>	0,00
<b>RZ</b>	0,00

When done click on



to confirm. Click on



and, when the picking point set is completely defined, on



---

Any time during the definition of a picking point set we can edit or delete points already added to the list. Select point and click on



to do so respectively.

## Revolution points

If the object presents symmetries we can define what we call **revolution** points (e.g., a ring like the one we have considered as an example in this section is a revolution solid along the Z-axis—normal to the surface).

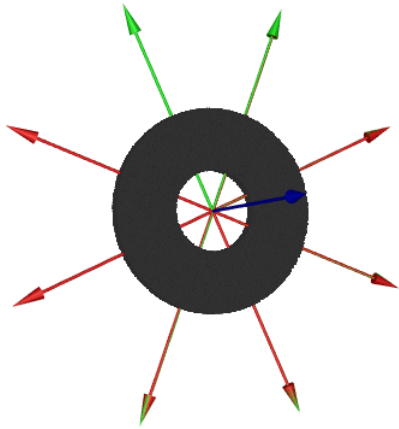
Check the box `revolution point`



and click on

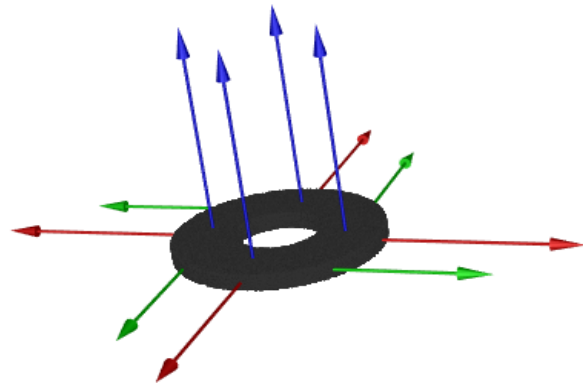


By default eight points will be considered.



We can adjust the number of points, radius and orientations as desired.

Radius	
15	<input type="range"/>
Number of Points	
4	<input type="text"/>
RX	0,00
RY	0,00
RZ	0,00



When done click on



## Link / edit picking points set

To edit or link / unlink a set of pick points to a sample select the radio button Link / edit picking points set under the heading Actions.

## ACTIONS

Select between **defining new** picking points set on 3D model, **editing** existing ones or **linking / unlinking** picking points sets to / from models

- Define new picking points set
- Link / edit picking points set

To link / edit a set related to a 2.5D or 3D model we first select the sample of interest and the model.

### Sample and model

Select **model** among those linked to the **selected sample**

*Note: it can be either a generated view model or a 3D model uploaded for the sample*

Ring  
Tri\_blue  
Trident\_60b  
Trident\_60g  
Trident\_blue  
Trident\_green

#### models

view1  
Ring

We will be able to see all the picking points sets that have been defined for this model and sample in the Sets list.

**Sets**

set\_1

We select the desired set and select the radio button `link / unlink set to model`.

Select action to perform on selected set: **edit / delete** or **link** to model

**Edit set**       **Link / unlink set to model**

Click on



to link set to model. If already linked click on



to unlink.

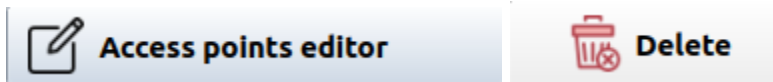
We can also edit / delete the selected set by selecting the radio button `Edit set`.

Select action to perform on selected set: **edit / delete** or **link** to model

**Edit set**       **Link / unlink set to model**

---

Click on



either to edit or delete the set.

**⚠ WARNING:** Only sets that are not linked to any model can be deleted.

## Application creation and execution

### Create new K-Pick application<sup>9</sup>

The term *K-Pick* stands for *known pick* in the sense that objects to be picked are known by the user. Since they are known the user will generally be able to scan them with the help of the robot and camera and create image / point-cloud *samples* from them (see the section on [adding samples](#)).

We can create new *K-Pick* applications with different collections of known objects. To proceed with the creation or edition of a *K-pick* application click on



and then on



to move to the application creation page.

### Create application

In order to create a new application we need to follow the next steps.

#### Step 1. Select ID

Select and verify ID for the new application.

---

<sup>9</sup> See <https://www.bitmetrics.es/resources>, video titled *K-Pick application creation*, for an alternative explanation and examples.

---

**New application name**  
Choose name for new application

## Step 2. Select application type

There are three types of applications:

- **2D** applications.  
Applications where there is no overlap (or little overlap) between objects and only a view or face of the object is of relevance (i.e., the camera always observes the same face or view of the object).
- **2DR** applications.  
Applications where there is no overlap (or little overlap) between objects but more than one view of the object is possible (i.e., the object can rotate and be seen by the camera with different orientations).
- **3D** applications.  
Applications where overlap between objects is indeed possible and all views of the object are possible (e.g., 3D random bin picking).

Select the type of application by checking the corresponding box

**Select application type**  
Application types:

- **2D**: one face of the object / small or no overlap between objects
- **2D with rotation (2DR)**: multiple object faces / small or no overlap between objects
- **3D**: multiple object faces / overlap between objects

When finished, click on **done**

**2D**       **2DR**       **3D**



---

and then click on



### Step 3. Select categories

Categories are meant to represent the groups of objects on which the same action is performed by the robot (e.g., all objects that need to be placed in Container A constitute a category whereas all those that need to be placed in Container B constitute another. Two place actions constitute two categories).

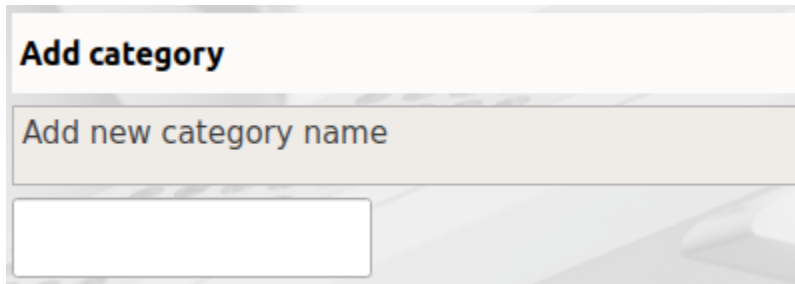
There is a default category which we call negatives. This category is meant to contain all objects known by the user that can occur in the pick scene but that no robot action needs to be performed on them (i.e., they do not need to be picked, they are ignored). The negatives category can be left empty if there is no known, well defined object in the pick scene on which the robot needs to perform no action.

The aim of the negatives category is to help the instance segmentation model in production to avoid false positives (see [Main concepts / landing page](#) for a better understanding of the aim of the instance segmentation model). There are actually two ways to do so:

- Negatives category (**weak**). By adding samples of objects to the negatives category we force the instance segmentation model to *ignore* such items during the training stage and thus learn to *ignore* them in production.
- Add all known, frequently occurring objects as *positive* categories (**strong**). By doing so we force the instance segmentation model to recognize and *identify* all such objects during the training stage and thus learn to *identify* them in production. Later, in order to discard all such objects, we will need to associate a robot pick and place action on them to remove them from further pick iterations in the pick scene.

---

To add a new category enter the category name in the corresponding text field



The screenshot shows a form titled "Add category". It contains a text input field with the placeholder text "Add new category name" and a button with a plus sign icon.

and click on



To remove a previously added category from the list of categories select it first



The screenshot shows a list titled "Categories". The list contains two items: "negatives" and "apples". The "apples" item is selected, indicated by a blue highlight.

and then click on



When done click on



---

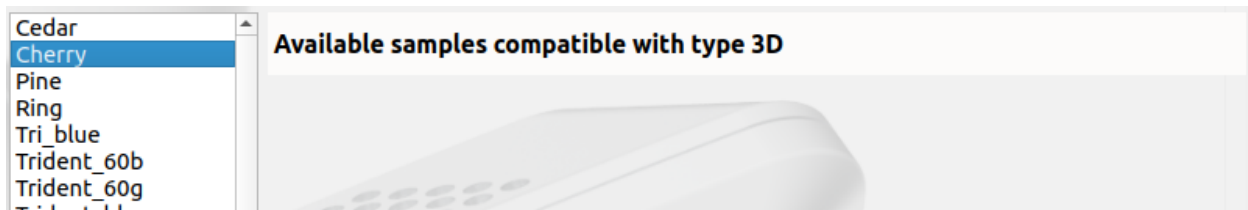
## Step 4. Add objects / samples to categories

It is time to add objects / samples to all the categories that we have defined (see [Add sample](#) to learn how to add samples to data repository). As mentioned above, the robot is meant to execute the same action on all objects under the same category.

To add a sample to a category first select the desired category from the corresponding list



and then select a sample from the list of all compatible samples for the given application type.



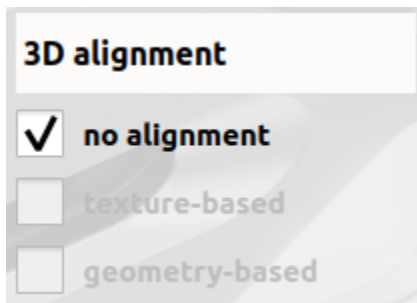
Before adding the sample to the category we need to select the type of alignment we wish to perform in order to determine the object pose in the scene. We contemplate three possibilities:

- **No alignment** (default)  
No alignment is performed. The pick point is assessed automatically based on the analysis of the point cloud / surface of the object in the pick scene.
- **Texture-based**  
Object alignment with the model or view is based on texture attributes. Suitable for objects that are *rich* in texture and have several distinguishable texture features.

- **Geometry-based**

Object alignment with the model or view is based on geometric attributes.

Depending on the type of object / sample some alignment options might not be available (e.g., a sample that has no model or view since it presents geometric variability among the different instances of the item—e.g., an organic item—will have only available the no-alignment option).

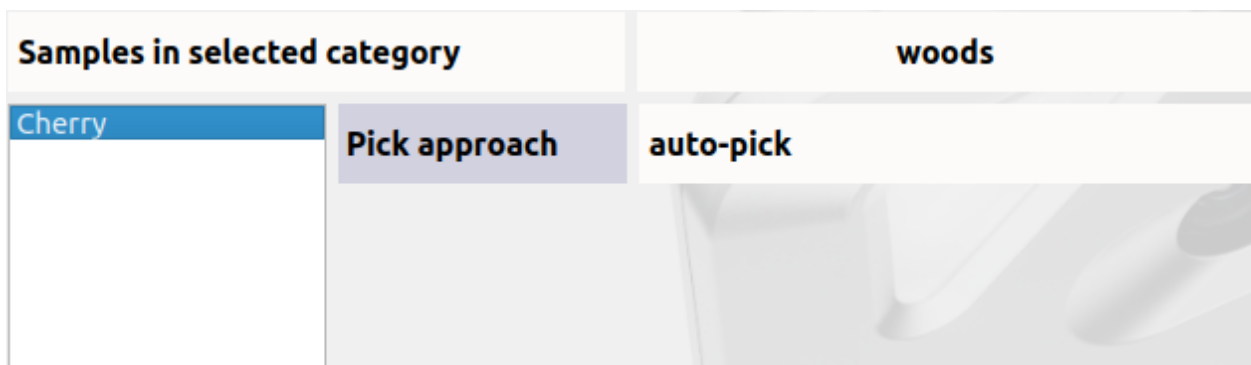


Once selected click on



to add sample to category.

In case a previously added sample needs to be removed first select it



---

and then click on



When done click on



## Step 5. Save and train application

Click on



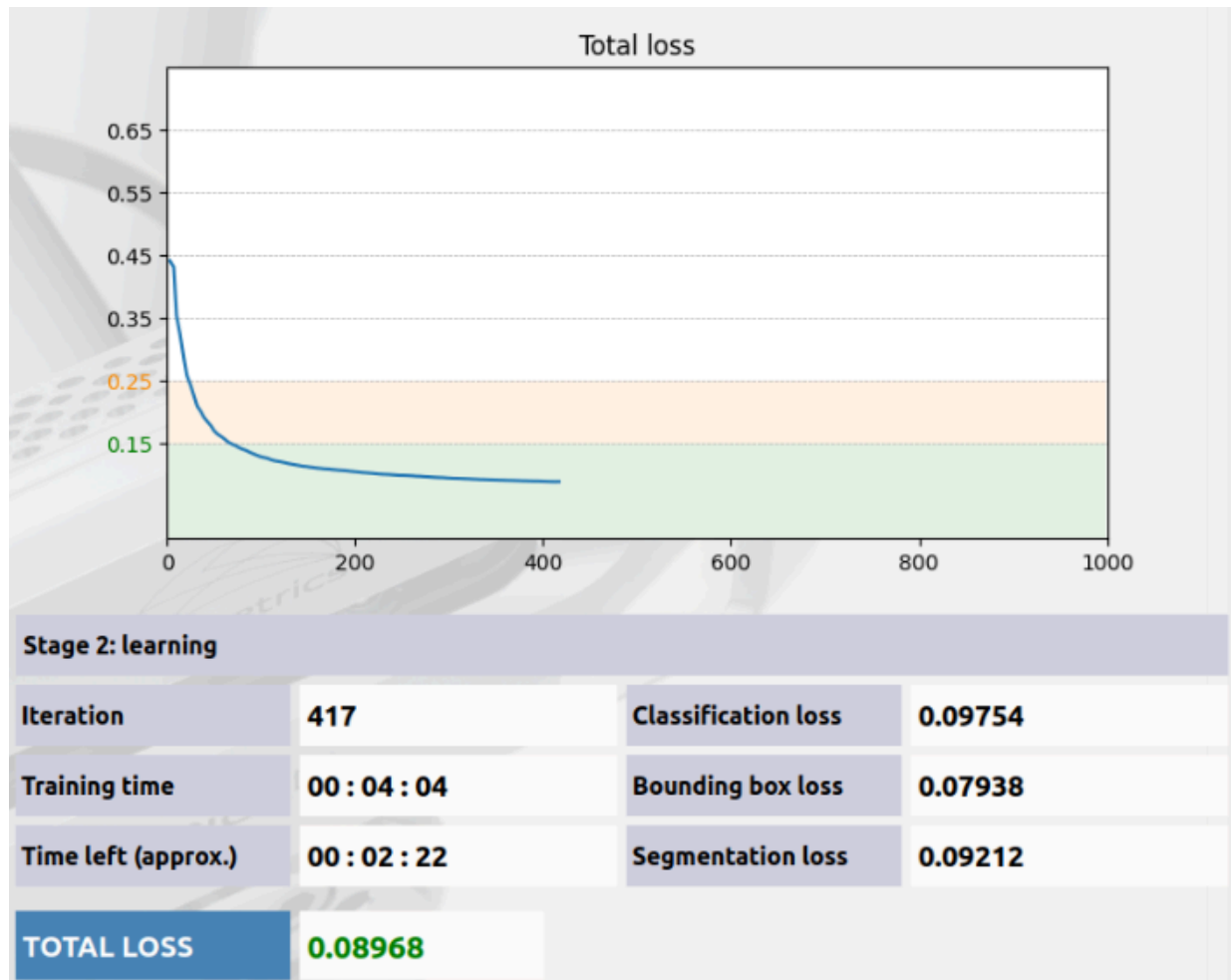
to save the application into the data repository. Before it can be executed the associated instance segmentation model needs to be trained: we need to learn the model that, in production, will be able to identify and locate all object instances in the pick context.

The application model can be trained right after the saving or any other time.

To train the application model click on



to move to the training widget.



There are two stages in the training process:

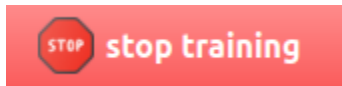
- **Stage 1:** data generation.  
Stage that takes between two and five minutes to generate training data.
- **Stage 2:** learning.  
Learning process.

Training metrics are monitored and plotted to see progress during Stage 2.

**Total loss** is calculated from the classification and segmentation loss. Total loss is a good indicator of the goodness or badness of the learned model.

---

We can stop the training process any time by clicking on



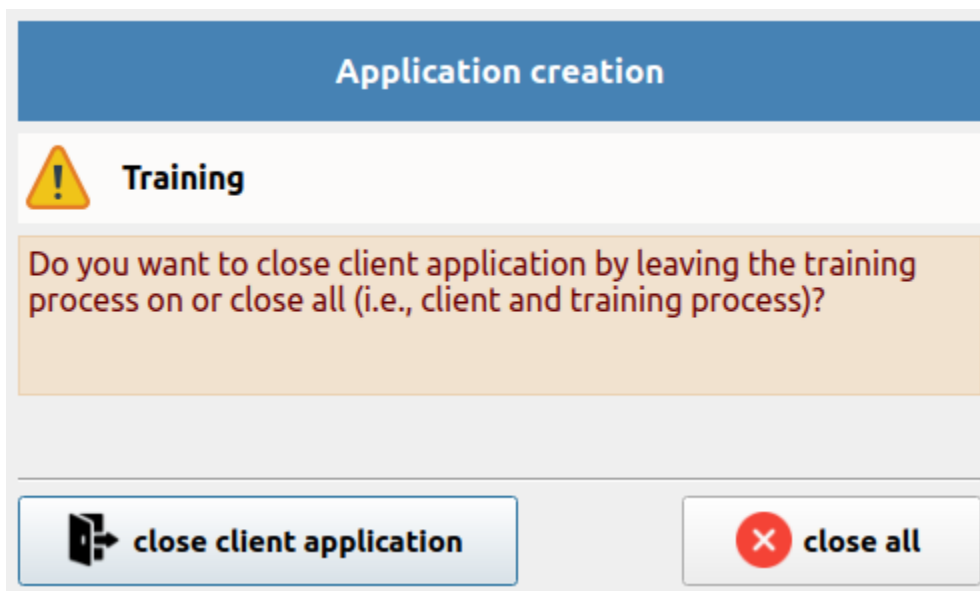
### Close client while training

Both when training and executing an application we can disconnect the client without stopping the server. To do so click on



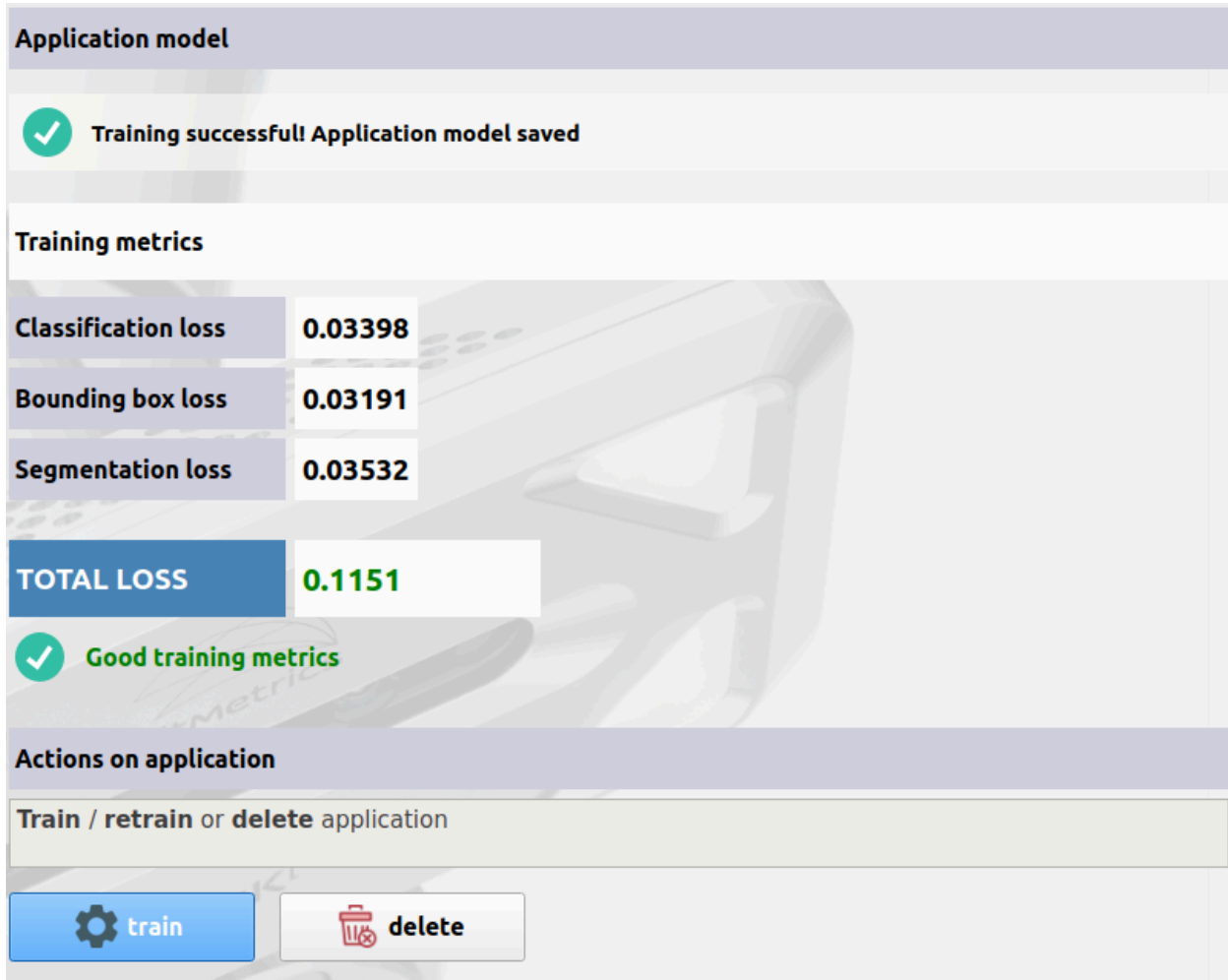
A pop-up will open to offer two possibilities:

- Close client and server (and thus interrupt training).
- Close only the client application and leave server on.



By starting the client again we go back to the training process widget if the training has not been finished.

After training we summarize results as



The screenshot displays the 'Application model' section of the BitMetrics interface. It features a green checkmark icon and the text 'Training successful! Application model saved'. Below this, the 'Training metrics' section is shown, containing a table of loss values:

Classification loss	0.03398
Bounding box loss	0.03191
Segmentation loss	0.03532
<b>TOTAL LOSS</b>	<b>0.1151</b>

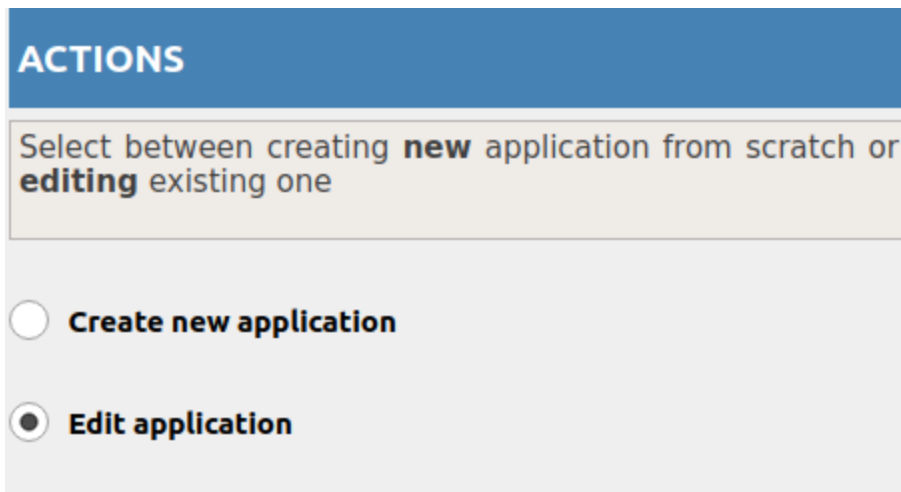
Below the table, there is another green checkmark icon and the text 'Good training metrics'. The 'Actions on application' section includes a text label 'Train / retrain or delete application' and two buttons: a blue 'train' button with a gear icon and a grey 'delete' button with a trash can icon.

A total loss under 0.15 is generally considered very good (green color). A loss between 0.15 and 0.25 is generally sufficient (orange color). A loss above 0.25 is, on many occasions, not sufficient to solve the picking context (red color). In the latter case the application design should be checked and the training problem identified (possibilities on why training is not fully satisfactory are endless but the most common cause is that instances of different samples or categories are very similar and hard to distinguish during the training process).



## Edit application

To edit an existing application select the radio button **Edit application** under the **Actions** heading.



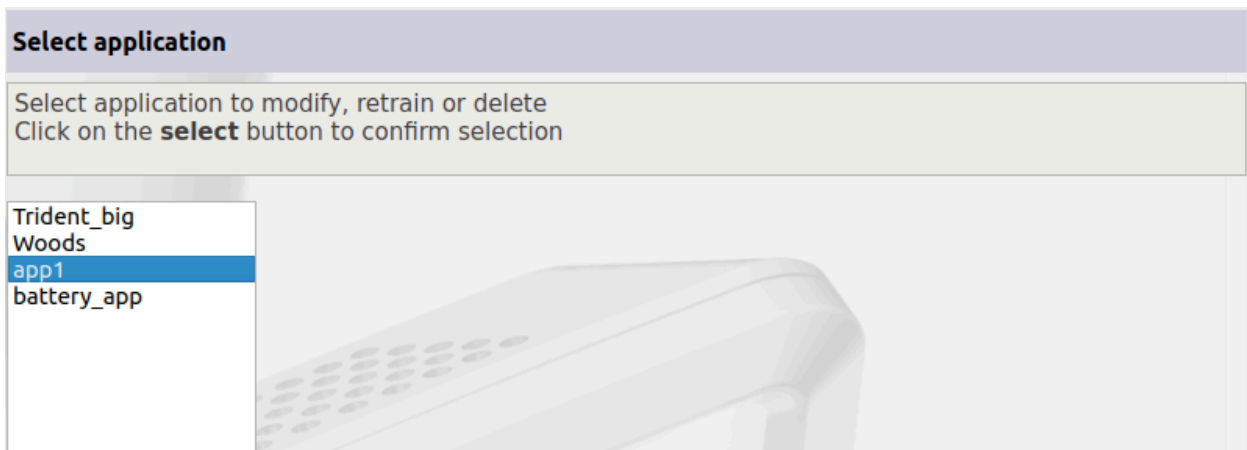
**ACTIONS**

Select between creating **new** application from scratch or **editing** existing one

**Create new application**

**Edit application**

Select application from the list



**Select application**

Select application to modify, retrain or delete  
Click on the **select** button to confirm selection


- Trident\_big
- Woods
- app1**
- battery\_app

and click on



to select.

---

Click on the corresponding  **edit** button to edit application (name, type, categories or samples).

Proceed as with the application creation to save and retrain.

## Execute application

### Pre-execution checks

Follow these steps before proceeding with application execution.

#### 1. Check application settings

Make sure that hardware and calibration settings are as required.

*Note: see [Application settings](#) to check / edit camera settings and lights connection.*

#### 2. Check environment

Make sure that both the pick environment and the tool bounding box selected are the desired ones for the application execution.

*Note: see [Application environment](#) to define / select boxes or environments and [Tool collisions box](#) to define / select bounding box for the actuator.*

#### 3. Check calibration

Make sure the pair camera-robot is properly calibrated before proceeding with application execution.

*Note: see [Calibration](#) if you need to calibrate camera-robot.*

---

## K-Pick application<sup>10</sup>

*K-Pick* is suitable in contexts where we are dealing with a few known objects. As such:

- Objects are available to the user and scanning them is not particularly costly (in the sense that we are dealing with a small number of objects).

To select a *K-Pick* application to execute click on

A rectangular button with a light blue gradient background and the text "K-Pick texture" in a bold, dark blue font.

and then on

A blue rectangular button with a white gear icon on the left and the text "execute K-Pick application" in white font.

The application pop-up will open with all *K-Pick* applications in the repository listed.

---

<sup>10</sup> See <https://www.bitmetrics.es/resources>, video titled *K-Pick: execution*, for an alternative explanation and examples.

**Application selection**

### Application selection manager

Select repository
application from

Trident\_big  
Woods  
app1  
battery\_app

Application type

Date created

day

hour

Last time executed

day

hour

Consistency check

Click on button below to proceed with application execution

**execution manager**

Application categories


Positive		Negative	
	Samples		Samples

**close**


We select the application we wish to execute. By doing so a consistency check is performed:

- A check on the consistency of the application is carried out. We check that
  - the application has been trained and that there is a suitable model for the candidate segmentation,
  - models are present—if required by the type of alignment of the items,
  - picking points on models have been defined—if required by the type of pick of the samples,

- A check on the consistency of the execution environment is also carried out. We check that:
  - a box environment has been selected,
  - a tool bounding box is also selected,
  - a calibration exists for the selected camera setting.

Woods			
Application type	3D		
Date created	day	05/07/2022	hour 19:07
Last time executed	day	06/07/2022	hour 10:59
Consistency check			
<b>No pick points defined for model 'view1' linked to sample 'Ring'</b>			
<b>Fix all consistency issues in order to proceed with execution</b>			

If the check is not passed we need to fix all the issues indicated in order to be able to proceed with execution of the application.

app1			
Application type	3D		
Date created	day	14/09/2022	hour 09:28
Last time executed	day	---	hour ---
Consistency check			
<b>Consistency check successful! You can proceed with execution</b>			

---

If the check is passed we can proceed with execution by clicking on



By doing so we move to the execution visualization page and carry out two actions in an orderly manner:

1. We send all application details to the execution server to configure it.
2. We start the server that listens to the robot messages coming from execution.

At this point we can start the robot routine from the robot console.

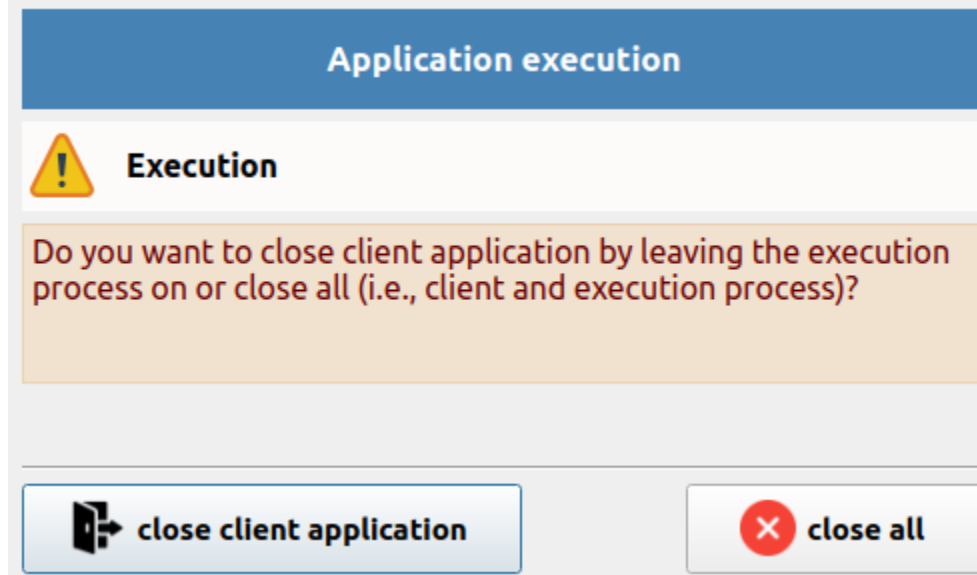
### **Close client while executing**

Both when training and executing an application we can disconnect the client without stopping the server. To do so click on



A pop-up will open to offer two possibilities:

- Close client and server (and thus stop the execution and robot server).
- Close only the client application and leave the servers on. In this case, by starting the client again we go back to the execution visualization page if the training has not been finished.



## U-Pick application<sup>11</sup>

*U-Pick* is suitable in contexts where:

- **In pick:** objects / items to be picked are not *known* or, if they are, there is a large number of them, making the scan process time consuming and the training process very hard (in the case that we are dealing with many references). Robot pick pose is estimated based on the point-cloud surface of the object.
- **In place:** place is not oriented (i.e., undefined). *Place* position and orientation of the object is only approximate.

Unlike *K-Pick*, the *U-Pick* approach builds on a pre-trained model.

To proceed click on



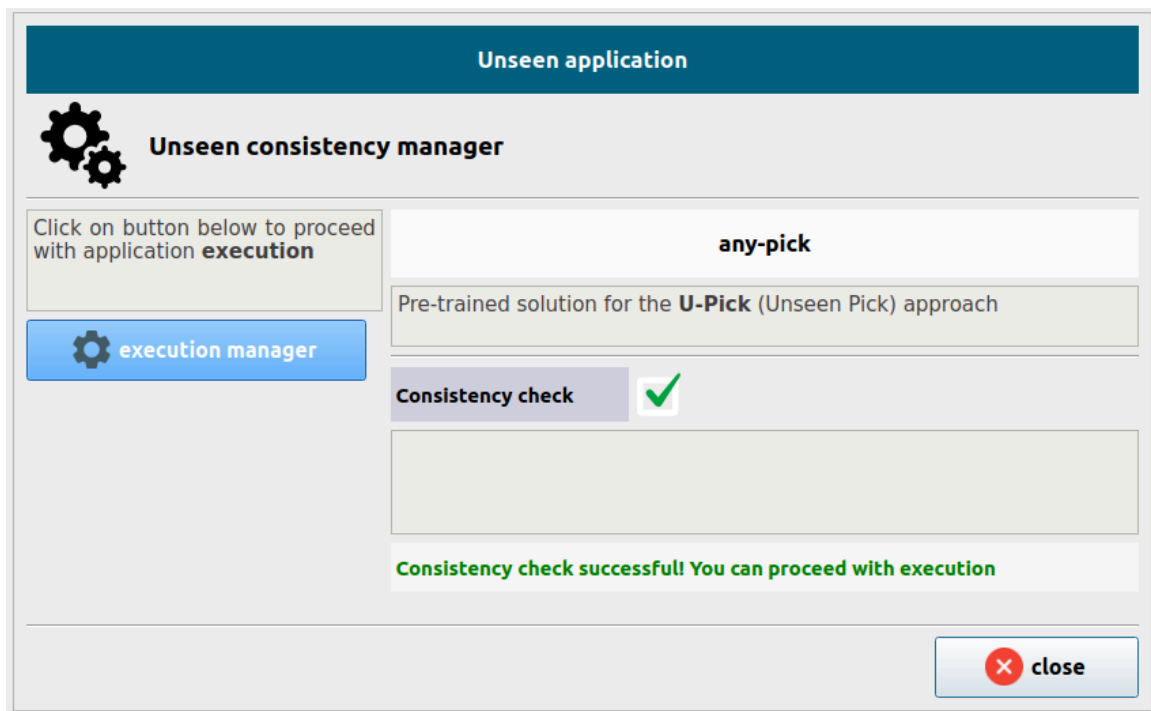
---

<sup>11</sup> See <https://www.bitmetrics.es/resources>, video titled *Create a U-Pick application*, for an alternative explanation and examples.

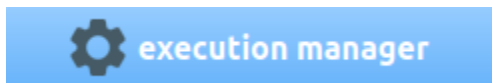
and then click on



By doing so a check on the consistency of the application environment is carried out (see [K-Pick application](#)).



If the consistency check is passed we can proceed and click on



to access the execution visualization window (see [K-Pick application](#) for more details). As with *K-Pick*, the client application can be closed during execution (see [K-Pick application](#)).



---

## Logs and alarms

### Logs

In the event of errors or unexpected outcomes logs should be checked. We have access to both *Pick[+] server* and *Pick[+] client* logs.

To access logs click on




We can visualize *Pick[+] client* application logs by clicking on



A pop-up with the most recent logs will be displayed.

**Logs register**


 **Application logs**

```
----- SESSION START -----
----- 08-09-2022 12:14:47 -----
09/08/2022 12:14:50: (WARNING) Retrying (Retry(total=1, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'NewConnectionError('<urllib3.connection.HTTPConnection object at
0x7eff48910e48>: Failed to establish a new connection: [Errno 113] No route to host,'): /device_info
09/08/2022 12:14:55: (WARNING) Retrying (Retry(total=0, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'NewConnectionError('<urllib3.connection.HTTPConnection object at
0x7eff46cd8160>: Failed to establish a new connection: [Errno 113] No route to host,'): /device_info
09/08/2022 12:14:56: (CRITICAL) [C1] / 'device_info' request. No response from server.
Exception: HTTPConnectionPool(host='192.168.1.100', port=5000): Max retries exceeded with url: /device_info
(Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7eff46cd8358>: Failed to
establish a new connection: [Errno 113] No route to host,))

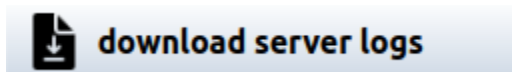
----- SESSION START -----
----- 08-09-2022 12:16:09 -----
09/08/2022 12:16:12: (WARNING) Retrying (Retry(total=1, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'NewConnectionError('<urllib3.connection.HTTPConnection object at
0x7f5a025ffe10>: Failed to establish a new connection: [Errno 113] No route to host,'): /device_info
09/08/2022 12:16:17: (WARNING) Retrying (Retry(total=0, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'NewConnectionError('<urllib3.connection.HTTPConnection object at
0x7f5a00cb1128>: Failed to establish a new connection: [Errno 113] No route to host,'): /device_info
09/08/2022 12:16:18: (CRITICAL) [C1] / 'device_info' request. No response from server.
Exception: HTTPConnectionPool(host='192.168.1.100', port=5000): Max retries exceeded with url: /device_info
(Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f5a00cb1320>: Failed to
establish a new connection: [Errno 113] No route to host,))

----- SESSION START -----
```

File located at /logs in application root folder

 close

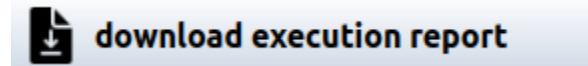
We can also download *Pick[+]* server logs by clicking on



A file named *pickplus\_server.log* will be downloaded and stored in the local *downloads* folder.

---

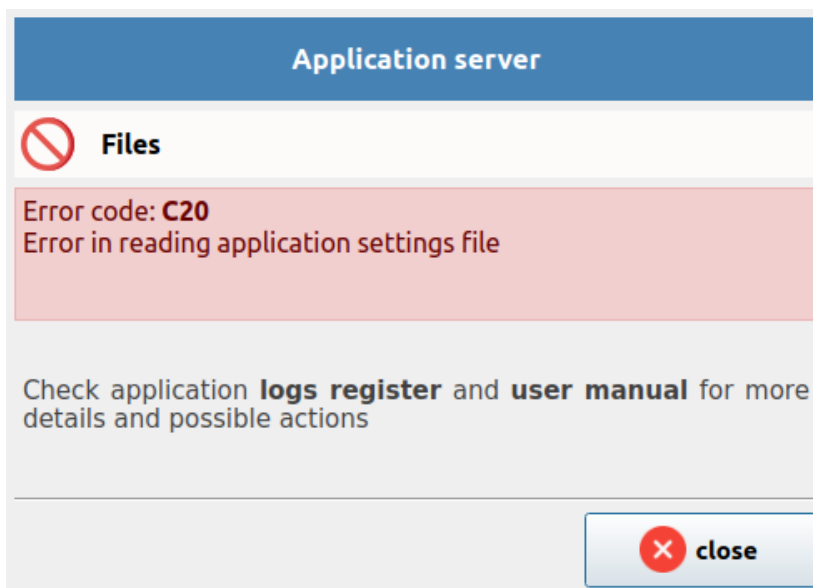
An execution report with data regarding the last execution iteration can also be downloaded by clicking on



A folder named `exec_report.zip` will be downloaded and stored in the local `downloads` folder.

## Alarms

In the face of an unexpected outcome the client application generally shows an alarm pop-up:



The alarm pop-up has the following components:

- *Heading*. The heading (or main title) indicates the general origin or main functionality affected by the error or warning.
- *Title*. The title briefly states the specific origin of the error or warning.
- *Code*. Alarm code (more on it below).
- *Message*. Error raised and explanation of it (when necessary).

---

## Alarm types

**First coordinate of the alarm code.** By error / warning level.

- *warning (W)*. Action by the user required. Incidence that does not affect the functioning of the solution. The user can continue using the solution.
- *error (E)*. Action by the user required. Incidence that generally needs fixing but that does not require closing the application.
- *critical (C)*. Action by the user required. Application is closed since, in order to make it work, exit and restart after fixing the problem is necessary.

**Second coordinate of the alarm code.** By origin.

- *unknown (0)*. Unknown / undefined origin.
- *communications (1)*. Communications error between any nodes of the solution.
- *server (2)*. Errors with origin in server.
- *data (3)*. Errors with origin in the data repository.
- *client (4)*. Errors derived from the client application.
- *execution (5)*. Errors / warnings during execution.

**Third coordinate of the alarm code.** By object type.

- *file (0)*. Error / warning caused by problems with a particular file.
- *Installation (1)*. Error / warning caused by problems in installation.

To gather more information on the cause of the alarm *server* and *client* log records need to be checked (see [Logs](#) section).