Tutorial

This Tutorial is intended to make you familiar with some basic procedures while working with CIROS. It goes beyond the content of the Getting Started tutorial and has been written for users who have already basic experience in working with CIROS. It is assumed that you have already read and worked with the Getting Started when beginning with this example.

The main intent is to provide information on how to add and handle mechanisms as well as how to handle the most frequent graphical modeling tasks.

The tutorial covers:

- Usage of materials
- Usage of mechanisms
  - Conveyor belt
  - Replicator
  - Trashcan
- Modeling of templates
- Modeling of primitive, inactive objects

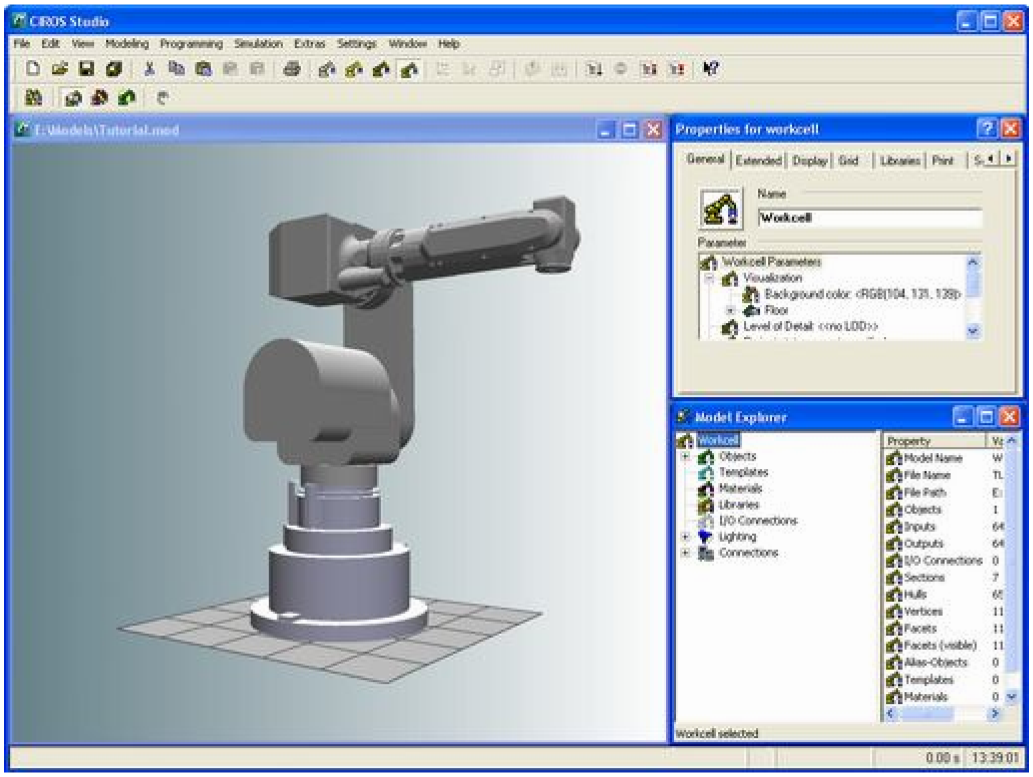
If your version of CIROS includes the sensor simulation, the tutorial additionally explains:

- Usage of sensor types
  - Distance sensor
  - Color sensor
- Utilization of the information given by the sensors to:
  - control the conveyor belt
  - move the robot relatively
  - sort items by color
- Randomized creation of simple objects

Step 1: Creating the Workcell

Create a new workcell and add `Demo Robot` from the `Demo Robots` library. Save the workcell as `Tutorial.mod`

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

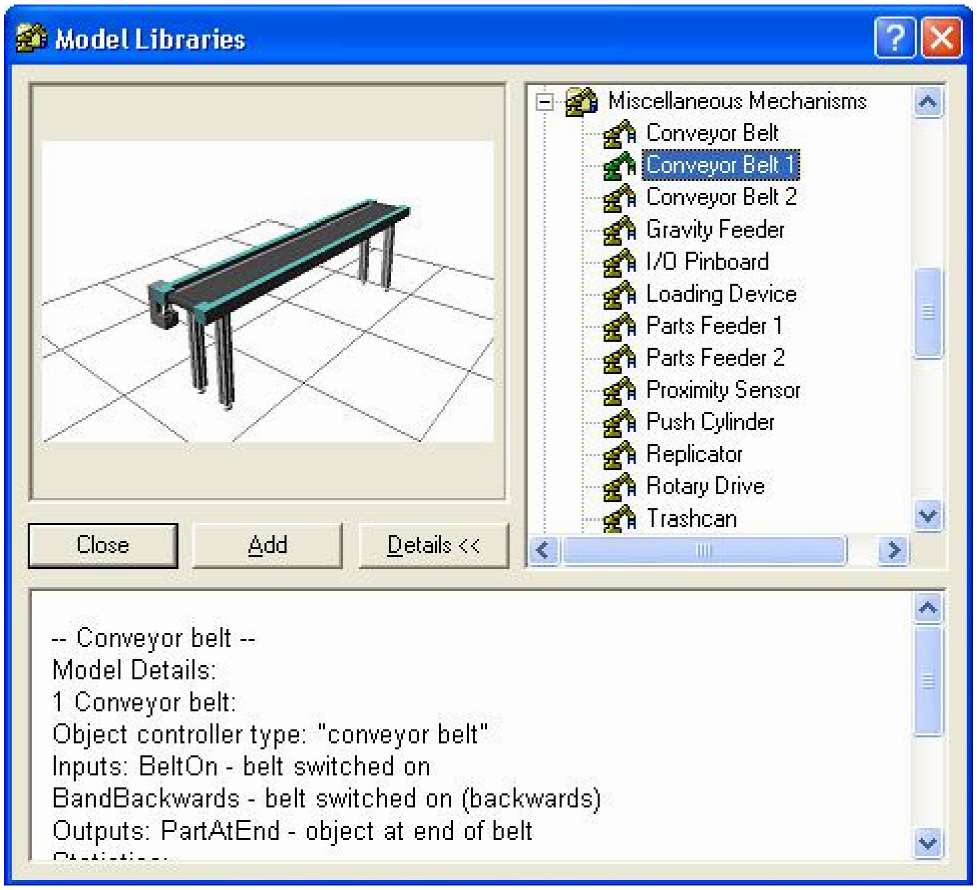


Step 2: Adding a Conveyor Belt

Adding the conveyor belt:

Open the model libraries and add `Conveyor Belt 1` from the `Miscellaneous Mechanisms` library.

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)



-- Conveyor belt --
Model Details:
1 Conveyor belt:
Object controller type: "conveyor belt"
Inputs: BeltOn - belt switched on
BandBackwards - belt switched on (backwards)
Outputs: PartAtEnd - object at end of belt

The added conveyor belt has two inputs which can be used to control the state (on/off) and the direction of the transportation mechanism. To switch the belt on, the input `BeltOn` must be set to `1.` By default the conveyor belt transports items from the drive to the end. `BeltBackw=1` changes the transportation direction. The output `PartAtEnd` indicates whether an object has reached the end of the conveyor belt.

The mechanism "grasps" any grip point that touches the conveyor surface and transports the corresponding object until the grip point reaches the conveyor's end. As soon as an object reaches the end, the output `PartAtEnd` will be set to `1.`

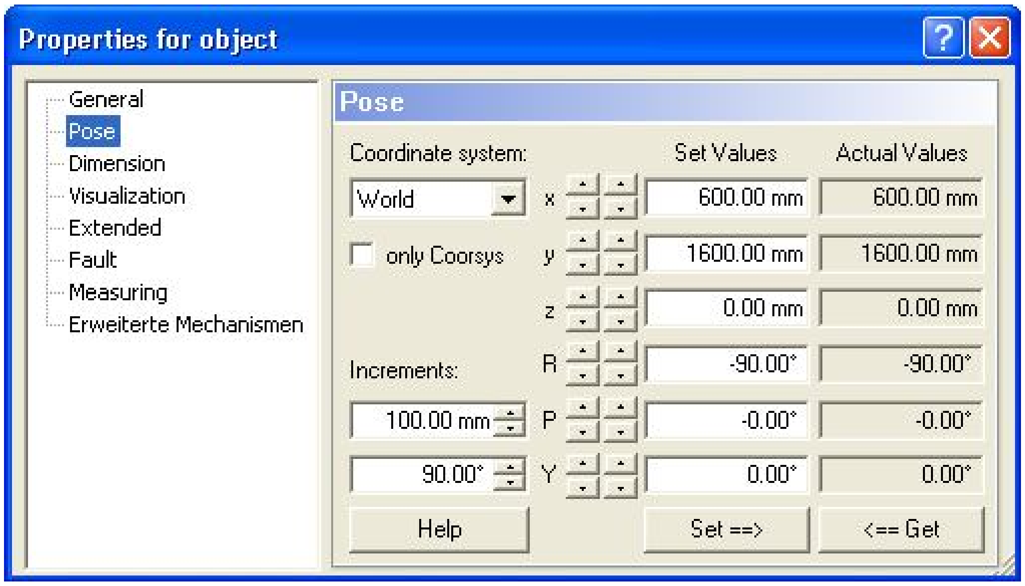
Positioning of the conveyor belt:

After importing, the conveyor belt is positioned in the origin of the workcell's coordinate system. The conveyor belt shall be turned by 180°, and then moved so that the end of the belt is positioned in the workspace of the robot.

Select the Conveyor belt in the *model explorer* and open the dialog *properties for object.* There, select the *Pose* tab and choose Coordinate system `World.`

Turn the object by reducing the absolute roll angle to -90°. Then move the object to the x,y,z coordinates given in the following picture:

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)



Properties for object

- General
- Pose
- Dimension
- Visualization
- Extended
- Fault
- Measuring
- Erweiterte Mechanismen

Pose

Coordinate system: World

only Coorsys [ ]

	Set Values	Actual Values
x	600.00 mm	600.00 mm
y	1600.00 mm	1600.00 mm
z	0.00 mm	0.00 mm
R	-90.00°	-90.00°
P	-0.00°	-0.00°
Y	0.00°	0.00°

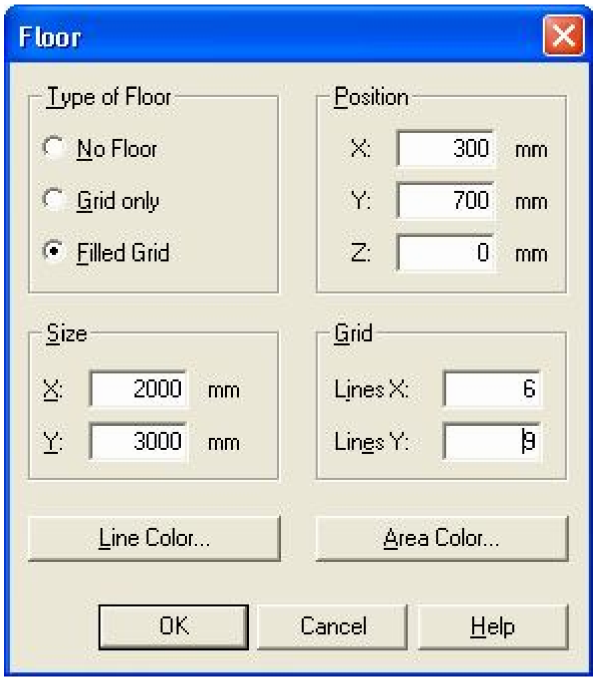
Increments: 100.00 mm, 90.00°

Help | Set ==> | <== Get

Customizing the floor:

Adjust the floor of the workcell so that the all objects find place. Leave some space at the beginning and at the end of the belt for e.g. the replicator socket, which will be added in the following chapters.

Use command *Floor* from the context menu of the workcell to open the dialog *Floor.* There, enter the following values:



Floor

Type of Floor: No Floor [✓ Filled Grid] Grid only

Position		
X:	300	mm
Y:	700	mm
Z:	0	mm

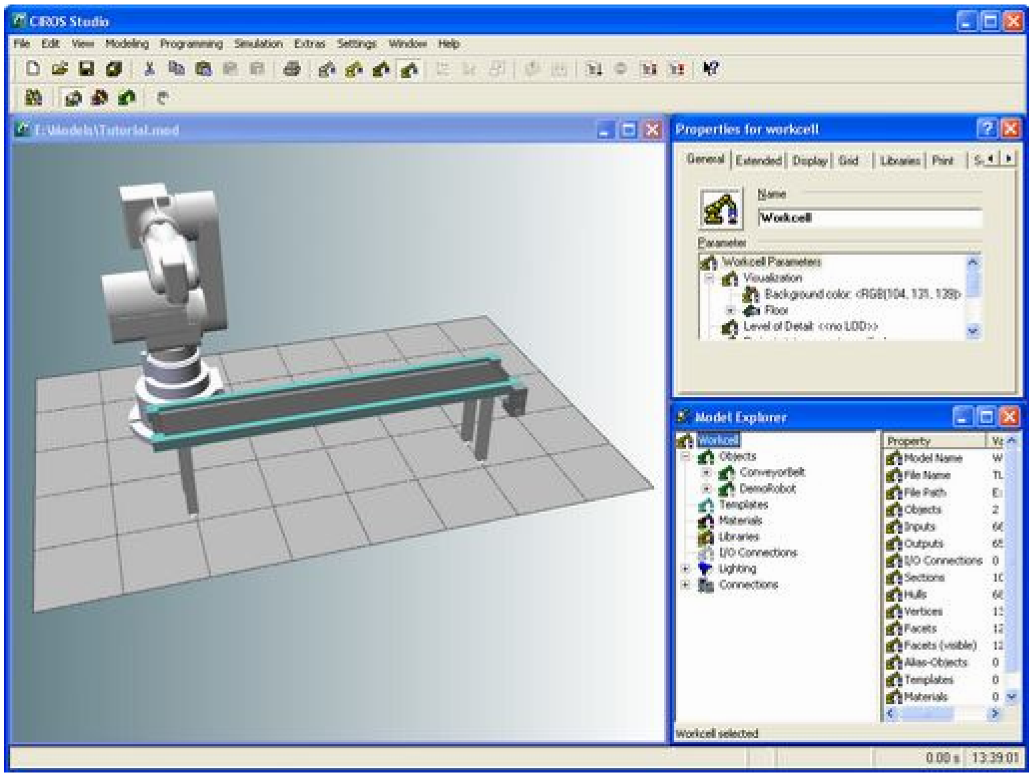
Size		
X:	2000	mm
Y:	3000	mm

Grid	
Lines X:	6
Lines Y:	9

Line Color... | Area Color... | OK | Cancel | Help

Finally your workcell should look like this:

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)


CIROS Studio
File Edit View Modeling Programming Simulation Extras Settings Window Help
E:\Models\Tutorial.mod
Properties for workcell
General Extended Display Grid Libraries Print
Model Explorer
Property
Workcell selected
0.00 s 13:39:01

Name: Workcell

Parameter:
- Workcell Parameters
  - Visualization
    - Background color: <RGB(104, 131, 139)>
    - Floor
  - Level of Detail: <<no LOD>>

Workcell
- Objects
  - ConveyorBelt
  - DemoRobot
- Templates
- Materials
- Libraries
- I/O Connections
- Lighting
- Connections

Property	Va
Model Name	W
File Name	T
File Path	E
Objects	2
Inputs	6
Outputs	6
I/O Connections	0
Sections	1
Hulls	6
Vertices	1
Facets	1
Facets (visible)	1
Alias-Objects	0
Templates	0
Materials	0

Step 3: Creating the Workpiece

In this step, you create a simple workpiece from a geometric primitive. The workpiece shall have the following properties:

- Box shaped, 150mm x 150mm x 150mm size
- Grip point on the bottom, so that the workpiece can be grasped by the conveyor belt.
- Grip point on the top, so that it can be grasped by the robot.

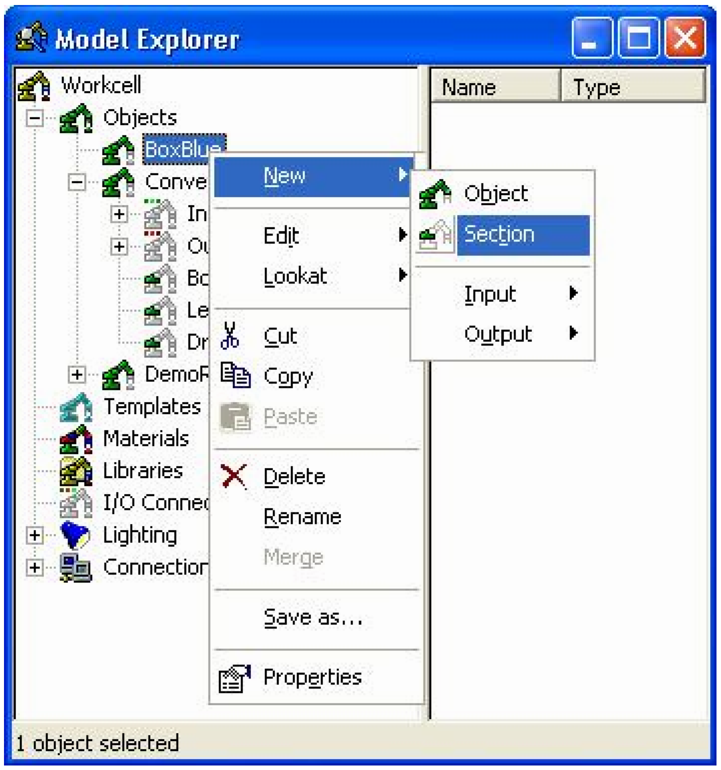
Creating a simple box using the model explorer:

Select the *Objects* entry of the model explorer. Use Command *New>Object* from the context menu to create a new object. Rename the new object to `BoxBlue.`

The new object ist positioned in the origin of the world coordinate system. To make sure that you can see it during further modeling, move the object e.g. to world coordinates `x,y,z = 1000mm, 0mm, 0mm.`

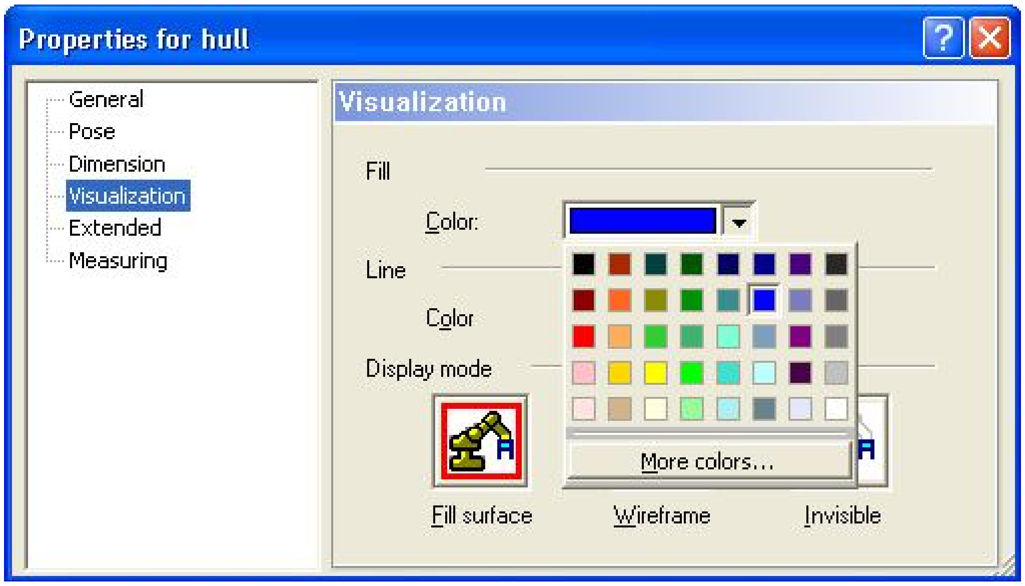
For the new object select commend *New>Section* from the context menu to create a new section. The section will automatically be named `Base.`

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)



In the section, create a new box the same way using command *New>Geometric Primitives>Box*

Select the box and use the property tab *General* to enter new values for `Length x,y,z = 150.00mm.` Then select the *Visualization* tab to change the color to `blue.`

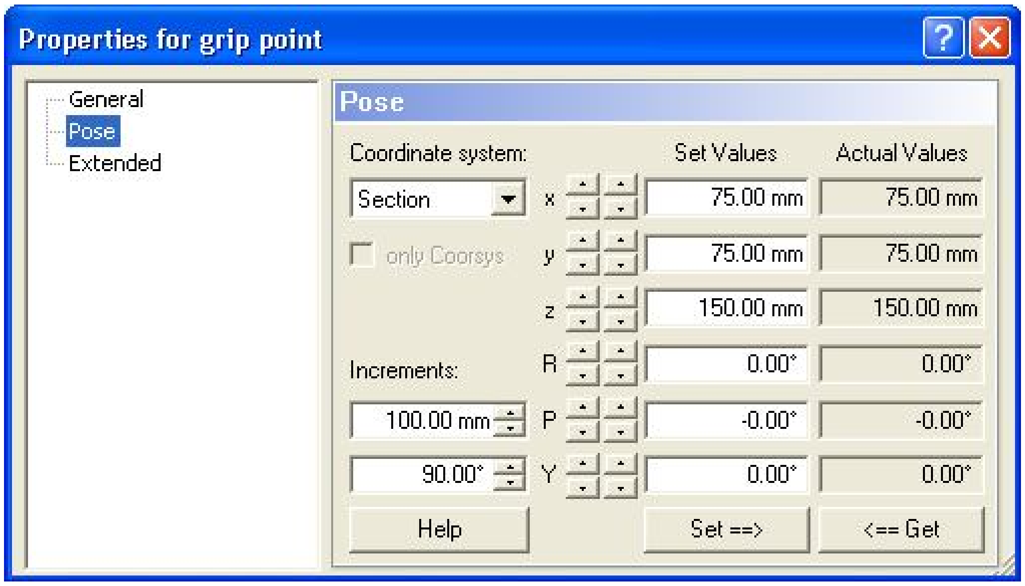


Adding grip points:

To add grip points to the box, select the section `base.` Then use command *New>Grip Point* from the context menu twice to add two grip points. Rename the first grip point to `GPTop,` the second to `GPBottom.`

The grip point `GPTop` shall be placed in the middle of the top surface where it can be grasped by the robot's gripper, the replicator and the trashcan. Move the grip point in the section (Coordinate system: Section) to `x,y,z = 75mm, 75mm, 150mm.`

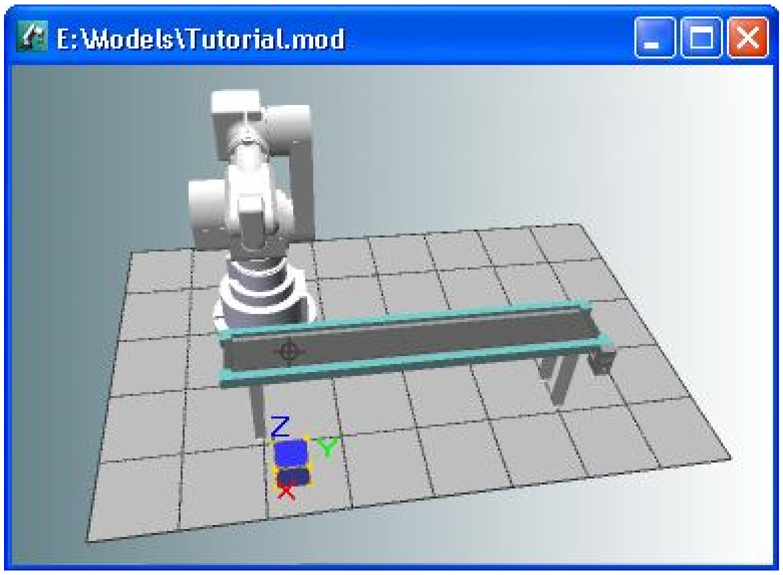
You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)



Coordinate system:		Set Values	Actual Values
Section	x	75.00 mm	75.00 mm
only Coorsys	y	75.00 mm	75.00 mm
	z	150.00 mm	150.00 mm
Increments:	R	0.00°	0.00°
100.00 mm	P	-0.00°	-0.00°
90.00°	Y	0.00°	0.00°

Leave the grip point `GPBottom` at position `x,y,z = 0mm.` The grip point is located at the bottom of the box (where it can be grasped by the conveyor belt) and at the edge which first reaches the conveyor's end.

Your workcell should now look like this:



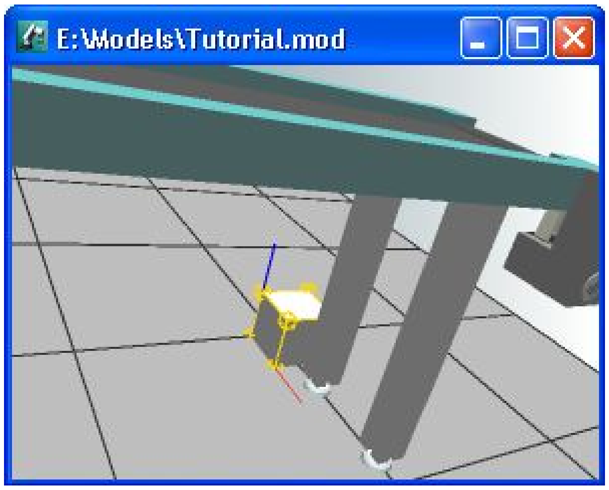
Step 4: Modeling the Replicator Socket

A replicator is a pure mechanism without any visualization. To avoid items that appear just out of nothing, a simple socket for the replicator has to be modeled. Later the replicator will be placed into the socket.

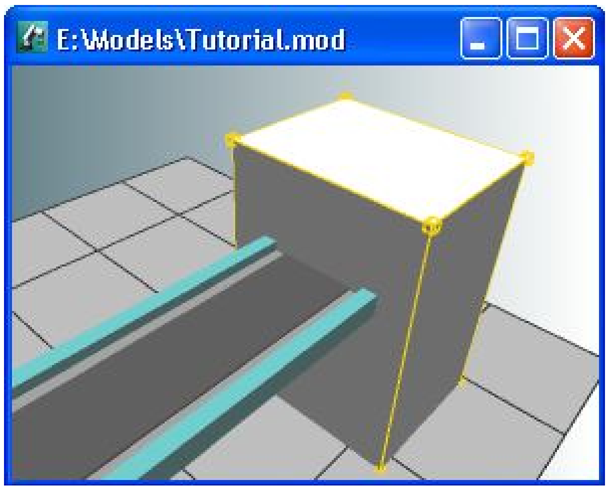
Creating the socket:

Create a new object and rename it to `ReplicatorSocket.` Move the object to position `x,y,z = 500mm, 1400mm, 0mm.` Then create a new section and a new box inside the section.

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)



Resize the box as done in the previous chapter: Open the *properties* dialog of the box and change the *length* values to x,y,z = `500mm, 400mm, 700mm`

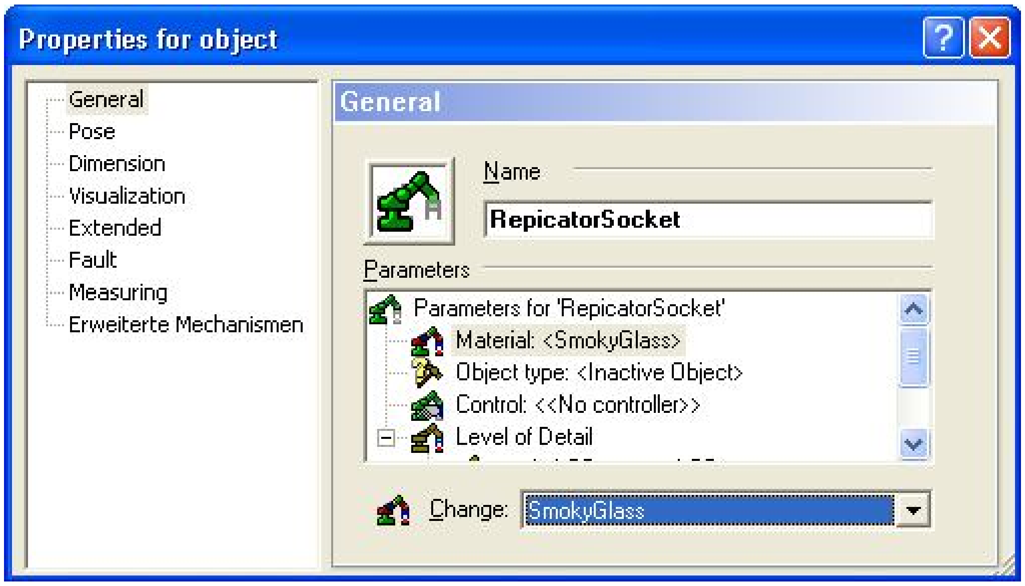


Assigning a material:

Open the model libraries and add the material `Smoky Glass` from the `ModLibs > Materials` library.

Then select the object *ReplicatorSocket* in the *model explorer.* Open the *General* tab of the *Properties for Object* dialog and change the material to `SmokyGlass.`

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)


Properties for object

General
Pose
Dimension
Visualization
Extended
Fault
Measuring
Erweiterte Mechanismen

General

Name
RepicatorSocket

Parameters
Parameters for 'RepicatorSocket'
Material: <SmokyGlass>
Object type: <Inactive Object>
Control: <<No controller>>
Level of Detail

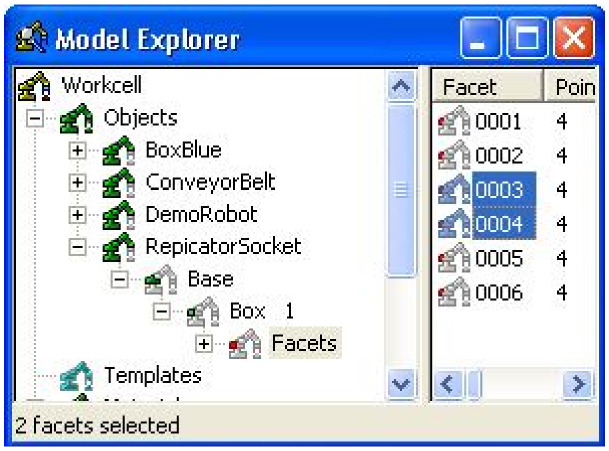
Change: SmokyGlass

The properties of the surface are now corresponding to the properties of the material `SmokyGlass.` The surface will become e.g. semi transparent.

Deleting unneeded Facets:

Use menu command *Settings>Model Explorer* to open the dialog *Model Explorer Options.* Select the *Display* Tab and check the filter option *Facets.* The model explorer will now additionally display facets in the tree view.

Select the facets number 3 and 4 of the box (the bottom and the facet which is pierced by the conveyor belt).


Model Explorer

Workcell
Objects
BoxBlue
ConveyorBelt
DemoRobot
RepicatorSocket
Base
Box 1
Facets
Templates

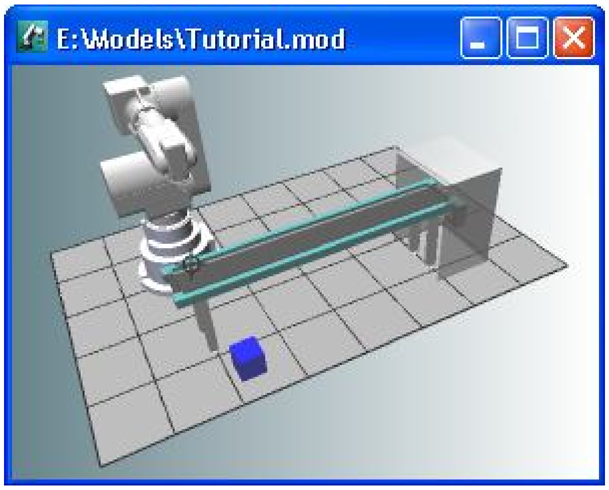
Facet	Poin
0001	4
0002	4
0003	4
0004	4
0005	4
0006	4

2 facets selected

Use command *delete* from the context menu to delete the selected facets.

Your workcell should now look like this:

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

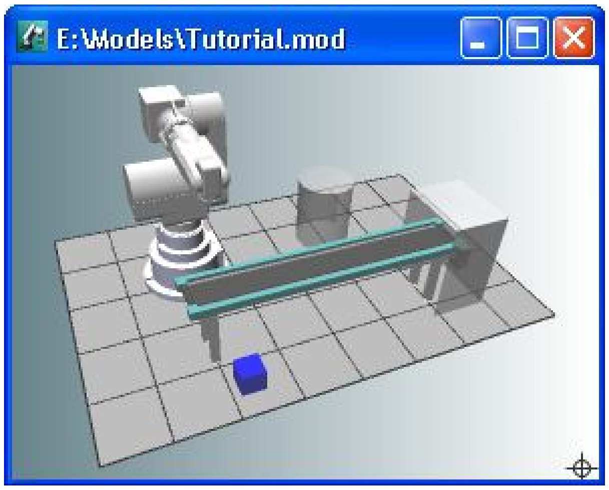


Step 5: Modeling the Trashcan

Similar to the replicator, the Trashcan does not have any visualization. For visualization purposes, a geometric primitive will be added and modified:

Create a new object and rename it to `Trash.` Add a section and a cylinder. Open the general properties of the cylinder and change the following properties: `Radius = 200mm, Height = 500mm`.

Move the object `Trash` to a position where it can be reached by the robot, (`x,y,z = -100mm, 1100mm, 0mm`) and assign the material `SmokyGlass` to the object.



Step 6: Mechanism Trashcan

Open the *model libraries* and add the mechanism *Trashcan* from the library *Miscellaneous Mechanisms.*

About the mechanism:

The mechanism `Trashcan` is equipped with three gripper points. Each of them is assigned to an input. If an object is "grasped" by one of the gripper points, it will be deleted from the workcell.

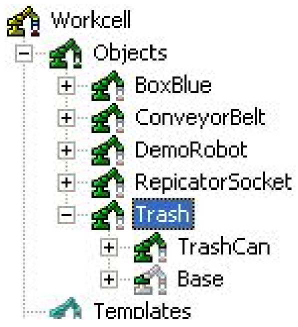
If an object's grip point is in the range of the mechanism's gripper point, the object can be deleted by setting the corresponding input from `0` to `1`.

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

Adding the mechanism to object Trash:

Select the object `TrashCan.` Use the *pose* tab of the dialog *Properties for Object* to move the object to the x- and y-coordinates of object `Trash` (World `x,y = -100mm, 1100mm`). Choose `z = 500mm.` The mechanism is now placed in the middle of the top surface of object `Trash.`

Then select the object `TrashCan` in the *model explorer* and drag it to object `Trash.` After releasing the mouse button, the mechanism becomes a part of object `trash.` It is displayed as a child entry of object `Trash` in the *model explorer.*



```
Workcell
  Objects
    BoxBlue
    ConveyorBelt
    DemoRobot
    RepicatorSocket
    Trash
      TrashCan
      Base
  Templates
```

Object `Trash` is now a so called "meta object" of the object `TrashCan.`

Step 7: Mechanism Replicator

About templates:

Templates are construction plans for objects. Similar to objects, they have an own entry in the top level of the *model explorer.*

To create a template, just model the desired object, and then drag the object to the templates entry using the model explorer. The object will then be removed from the model, and a corresponding construction plan will be added to the templates section.

To provide replication of objects on base of a template, the object/template must have at least one grip point.

About the replicator mechanism:

The Replicator mechanism provides the ability to generate objects based on templates. The names of the templates can be added to the extended properties of a replicator object.

To create an object based on a template, the input of the replicator which is corresponding to a template has to be set from `0` to `1`. The object will then be created and added to the model at the replicator's gripper point.

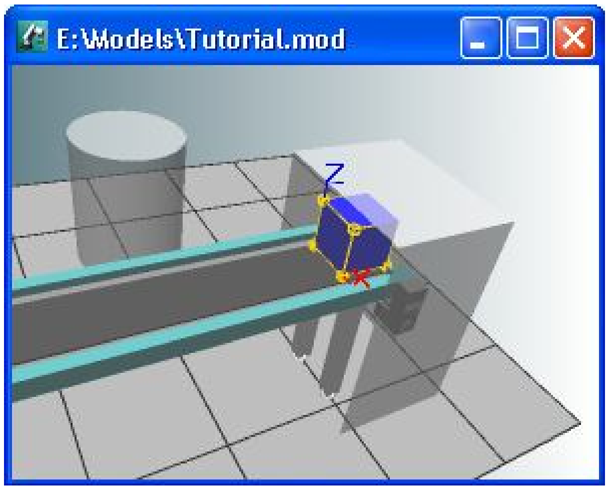
Replicators can be added on two way:

- Adding from the model library `Miscellaneous Mechanisms:`
  A replicator added from the model library is added at the origin of the world coordinate system and has to be configured manually.
- Adding via the context menu of an object *(Edit>Create replicator):*
  When creating a replicator this way, the replicator will be created at the first grip point of the selected object. The replicator will be automatically configured to create the selected object at its original pose. The object itself will be moved to the templates section.

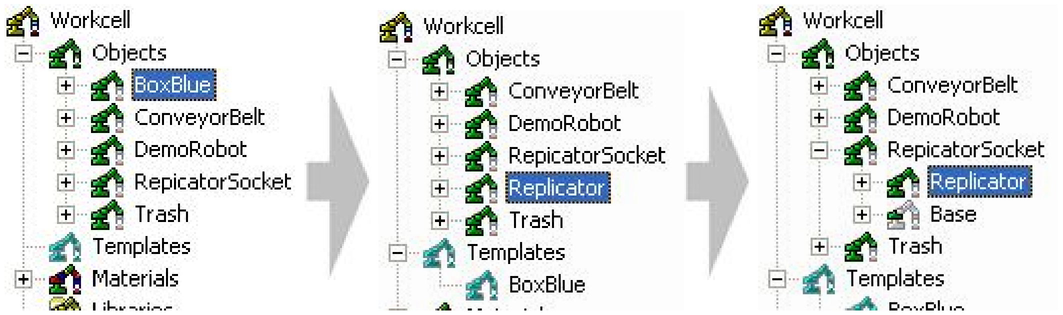
Adding a replicator:

For this example, use the second way of adding a replicator: First move the object `BoxBlue` to the position at which it shall later be replicated. Move it to the drive side of the conveyor on the inside of the replicator socket (World coordinates `x,y,z = 685mm, 1400mm, 494mm).`

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)



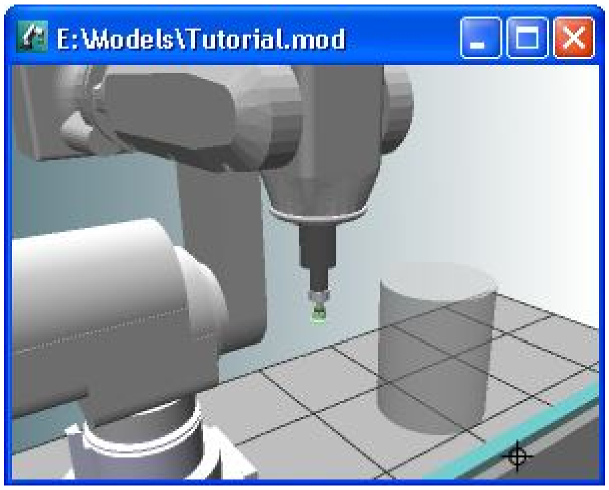
Now select the object `BoxBlue` in the *Model Explorer* and execute command *Edit>Create replicator* from the context menu. The object `BoxBlue` will be removed from the model, and a corresponding construction plan will be added to the templates section. At the same time, a new object `Replicator` is created. Add the object `Replicator` to the object `ReplicatorSocket.`



Step 8: Setup Robot

Adding the gripper:

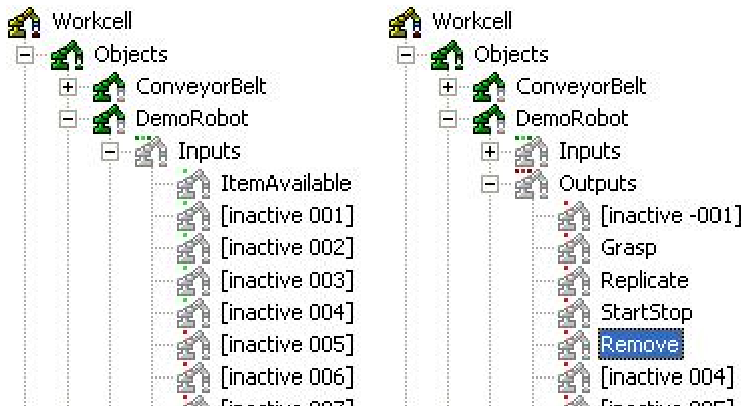
The replicated objects have a grip point in the center of their top which shall be grasped by a vacuum gripper. Open the model library `Miscellaneous Grippers` and add the gripper `vacuum gripper` to your model.



Renaming and activating I/Os:

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

The robots needs a number of I/Os to control the mechanisms and the gripper. Digital Outputs are needed to grasp an object, to trigger the replicator, to start the conveyor belt and to destroy objects in the trashcan. Additionally, the robot needs at least one digital input which indicates whether an object has reached the conveyor belt's end. Use the model explorer to rename the I/Os of the robot as shown in the following image. By renaming an I/O, it is automatically activated:



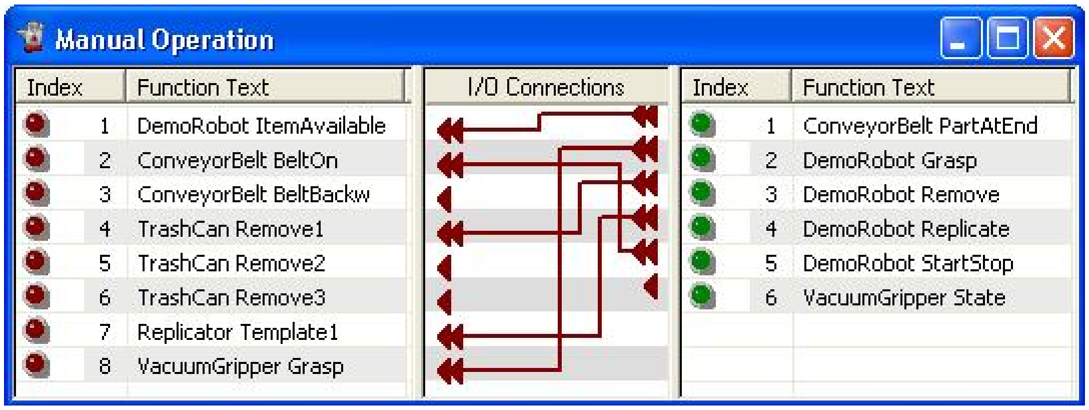
```
Workcell                              Workcell
  Objects                               Objects
    ConveyorBelt                          ConveyorBelt
    DemoRobot                             DemoRobot
      Inputs                                Inputs
        ItemAvailable                       Outputs
        [inactive 001]                        [inactive -001]
        [inactive 002]                        Grasp
        [inactive 003]                        Replicate
        [inactive 004]                        StartStop
        [inactive 005]                        Remove
        [inactive 006]                        [inactive 004]
```

Note:

For Mitsubishi robots the I/Os for gripper control are fixed and pre-defined. Instead of `Grasp` use the predefined output `HCLOSE1.`

Connecting inputs and outputs:

Open the *Manual Operation* window *(Modeling>Manual Operation).* Connect the I/Os of the robot with the corresponding I/Os of the mechanisms and the gripper. The following image shows the required connections. For the screenshot, some columns of the *Manual Operation* window have been hidden. To show/hide elements of the window, use the corresponding commands from the window's context menu.



Manual Operation

Index	Function Text
1	DemoRobot ItemAvailable
2	ConveyorBelt BeltOn
3	ConveyorBelt BeltBackw
4	TrashCan Remove1
5	TrashCan Remove2
6	TrashCan Remove3
7	Replicator Template1
8	VacuumGripper Grasp

I/O Connections

Index	Function Text
1	ConveyorBelt PartAtEnd
2	DemoRobot Grasp
3	DemoRobot Remove
4	DemoRobot Replicate
5	DemoRobot StartStop
6	VacuumGripper State

The robot can now control the mechanisms and the gripper by setting or resetting digital Outputs. A positive signal at input `ItemAvailable` indicates that an object has reached the end of the conveyor belt.

Step 9: Programming the Robot

Position list:

To program the robot, three positions are required: The initial position of the robot, the picking position at the end of the conveyor belt and the placing position at the trashcan.

Create a new position list, and add the initial position of the robot three times.

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)

Picking position:

Since the boxes are replicated with their first grip point at world coordinates `x,y,z = 760mm, 1475mm, 644mm`, and then only moved in y-direction, the x and z coordinates for the picking position are `x,z = 760mm, 644mm.` The box stops at the conveyors end, which is at `y = 4mm.` Because the upper grip point is in the middle of the box it is at `y = 4mm - 75mm = -71mm.` The resulting position is:

`x,y,z = 760mm, -71mm, 644mm`

Placing position:

The world coordinates of the trashcan's gripper point are `x,y,z = -100mm, 1100mm, 500mm.` Since the gripper point of the mechanism shall "grasp" the box at the upper grip point, these are also the coordinates for the placing position:

`x,y,z = -100mm, 1100mm, 500mm`

An abstract robot program shall perform the following steps:

Create an object:	Set output `replicate` from `0` to `1`
Switch on conveyor belt:	Set output `StartStop = 1`
Wait until item at end:	Wait until input `ItemAvailable = 1`
Switch off conveyor belt:	Set output `StartStop = 0`
Move to picking position:	Move to position 50mm above the grip point, then move straight to the grip point
Grasp:	Set output `Grasp = 1` (Mitsubishi: Command `HCLOSE1`)
Move to placing position:	Move up straight, then move to a position 200mm above the trashcan and move down straight to the placing position
Open gripper:	Set output `Grasp = 0` (Mitsubishi: Command `HOPEN1`)
Remove object from trashcan:	Set output `remove` from `0` to `1`

Example for an IRL robot program:

```
PROGRAM IRL;

IMPORT DATALIST 'Tutorial.PSL';
VAR
   TEACH ROBTARGET : POS1;
   TEACH ROBTARGET : POS2;
   TEACH ROBTARGET : POS3;

  INPUT BOOL : ItemAvailable        AT 0;

  OUTPUT BOOL : Grasp        AT 0;
  OUTPUT BOOL : Replicate        AT 1;
  OUTPUT BOOL : StartStop        AT 2;
  OUTPUT BOOL : Remove         AT 3;

   INT : i;

BEGIN

FOR i := 1 to 10
```

```
    {replicate object BoxBlue}
     Replicate := 1; Replicate :=0;

    {start belt}
     StartStop := 1;

    {move robot to init position}
    MOVE LIN POS1;

    {wait for box}
     WAIT FOR ItemAvailable;
     StartStop := 0;

    {pick object}
    MOVE PTP POS2 @ POSITION (0.0,0.0,-50.0);
    MOVE LIN POS2;
     GRASP := TRUE;
    MOVE LIN POS2 @ POSITION (0.0,0.0,-100.0);

    {place object}
    MOVE LIN POS3 @ POSITION (0.0,0.0,-200.0);
    MOVE LIN POS3;
     GRASP := FALSE;
    MOVE PTP POS3 @ POSITION (0.0,0.0,-200.0);

    {empty trashcan}
     Remove := 1; Remove := 0;
  ENDFOR;

  ENDPROGRAM;
```

Note:

The pre-installed samples of CIROS include an already completely modeled and programmed version of this example. Click here to open the pre-installed example. Please note that depending on your version of CIROS the current workcell may be closed automatically.

You created this PDF from an application that is not licensed to print to novaPDF printer (http://www.novapdf.com)