Webots™ is a development environment used to model, program and simulate mobile robots.

With Webots™, you can design complex robotics scenarios, including possibly several different robots, interacting in a shared environment. The properties of each object, such as shape, color, texture, mass or friction, are individually adjustable. A large choice of simulated sensors and actuators is available to equip each robot. The robot controllers can be programmed with the built-in IDE or any third party development environment. The behavior of the robots are run in physically realistic worlds. The controller programs can optionally be transferred to existing real robots.

Webots™ is used in over 1000 universities and research centers worldwide. With Webots™, you take advantage of a proven technology that has been developed in partnership with the Swiss Federal Institute of Technology in Lausanne (EPFL), thoroughly tested, well documented and actively improved for over 15 years. By using Webots™, you will save a considerable amount of development time.

**KEY FEATURES**

- Models and simulates any mobile robot, including wheeled, legged and flying robots.
- Includes a complete library of sensors and actuators.
- Has a built-in 3D world and robot editor with 3D import capability.
- Lets you program the robots in C/C++, Python, Java, ROS, Matlab™ or over TCP/IP.
- Runs on Windows, Mac OS X and Linux 32/64 platforms.
- Uses ODE (Open Dynamics Engine) library for real-time physics simulation.
- Transfers to real mobile robots: e-puck™, NAO™, DARwIn-OP™, Pionneer™, etc.
- Includes over 170 examples with open source code for 3D models and robot controllers.
- Lets you simulate multi-agent systems with communication facilities.
- Creates HD movies of your simulations.

![Webots™ user interface: scene tree, 3D view and integrated development environment (IDE)](image-url)
Sensor Library

A large choice of sensors can be plugged into your robot model:

- Distance sensors (IR, US and laser)
- Cameras (1D, 2D, spherical)
- Range finders
- Light sensors
- Touch (pressure or bumper) sensors
- Global Positioning Sensors (GPS)
- Receivers (inter-robot communication)
- Position and force sensors for servos
- Incremental wheel encoders
- Inertial Units (3D),
- Accelerometers (3D)
- Gyroscopes (3D)
- Digital compasses (3D)

Sensor parameters may be tuned individually: range, noise, response, field of view, etc.

Actuator Library

Similarly, a number of actuators can be added to your robot model:

- Differential wheel motor units
- Servo motors: legs, arms, wheels, etc.
- Linear motors (pistons)
- LEDs
- Emitters (inter-robot communication)
- Grippers
- Displays (LCD)
- Pens (drawing)

World and Robot Editor

With Webots™, it is easy to create state-of-the-art virtual environments for your robot simulations, using advanced graphics, with lights, shading, texture mapping, shadows, etc. Moreover, Webots™ allows you to import 3D models from most modeling software through the VRML97 standard.

You can create worlds as large as you need and Webots™ will optimize them to enable fast simulations.

Sophisticated robots models can be built by assembling chains of servo nodes. This allows you for example to easily create a robot with articulated legs, arms, or a pan / tilt camera.

You may also place several cameras on the same robot to perform binocular stereo vision, or a 360 degree vision system.

Faster than real-time

Webots™ PRO simulation engine uses virtual time, thus making it possible to run a simulation often much faster than real robots. Depending on the complexity of the setup and the power of your computer, a simulation can run up to 300 times faster than a real setup.

The simulation time step can be adjusted to suit your needs (precision versus speed trade-off). A step-by-step mode is available to analyze in detail the behavior of the simulation.

Open Physics Library

Simulating complex robotic devices including articulated mechanical parts requires a precise physics simulation. Webots™ relies on the powerful ODE (Open Dynamics Engine) to perform an accurate physics simulation. For each component of a robot or solid body, it is possible to specify the bounding objects for collision detection, the center of mass, the mass distribution matrix (or use primitives for simple geometries), the static and kinematic friction coefficients, the bounciness, etc.

You can also extend the physics simulation programmatically with a physics plugin. Servo devices can be controlled by your program in position, velocity or torque/force. The control parameters for the servo can be individually adjusted in your controller programs.

User Interaction

The graphical user interface of Webots™ allows you to easily interact with the robots and their environment. Robots and other objects can be translated, rotated and resized with the mouse, very intuitively and while the simulation is running. Similarly it is possible to apply forces and torques interactively to test the robustness of a mechanical system.

Robot Window

The built-in robot windows continuously display sensor and actuator information so that the status of each device can be monitored while the simulation is running. Actuators may also be controlled from the same robot window. Such windows can be extended by the user to design custom graphical user interfaces, using the Qt library.
Programming Interface

Programming your robot using the C language is as simple as this:

```c
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>

int main() {
  wb_robot_init(); // Initialization
  wb_device_tag lr = wb_robot_get_device("pd0");
  wb_distance_sensor_enable(lr, 32);
  // Perform 32 ms simulation steps
  while(wb_robot_step(32) == 0) {
    if (wb_distance_sensor_get_value(lr) > 100)
      wb_differential_wheels_set_speed(0, 0);
    else
      wb_differential_wheels_set_speed(10, 10);
  }
}
```

In this example, a “differential wheels” robot is equipped with an infrared distance sensor named “lr”. The robot will stop moving if the distance sensor detects an obstacle and it will resume its motion when the obstacle is no longer detected. Application programming interfaces (APIs) are also provided for the C++, Java, Python and Matlab™ languages. In addition it is also possible to control robots from ROS or from a standard TCP/IP Interface.

Supervising Simulations

Research experiments often need to interact automatically with the simulation. The supervisor capability allows you to write a program responsible for supervising an experiment. For example, a supervisor program can move objects, change their properties, send messages to robots, record robot trajectories, take a snapshot or record a video of the simulation.

The supervisor capability can be used for optimization algorithms where a large number of robot configurations or control parameters has to be evaluated, as in genetic programming, neural networks, machine learning, etc.

Transfer to Real Robots

Once tested in simulation your robot controllers can be transferred to real robots:

- **e-puck**: Finite state automata graphical programming, cross-compilation, and remote control execution modes are available.

- **Nao™**: Using either Choregraphe™ or NaoQi™ (Aldebaran Robotics), you can write controller programs and connect them either to the simulated or to the real Nao™ robot.

- **DARwIn-OP™**: The C++ Webots API allows you to run the same controller programs on both the simulated and the real DARwIn-OP™ robot exhibiting the same behaviors.

- **KUKA youBot™**: The realistic model of this mobile manipulator allows you to run the same control programs in simulation and on the real robot.

- **Your own robot**: The Webots™ user guide explains how to build a Webots™ cross-compilation system for your very own robot.

Robotics Applications

Webots™ has been used by our customers for many years in multiple robotic R&D projects in these fields:

- Mobile robot design and prototyping (wheeled, legged or flying robots)
- Multi-agent systems
- Robot locomotion
- Machine vision
- Artificial life and evolutionary robotics
- Modular robotics
- Surgical robotics
- Robot soccer (RoboCup)

In addition, Webots™ is widely used for teaching robotics and C/C++/Java programming in universities. Moreover Webots™ is used in online programming contests.

The robot window allows you to monitor both simulated and real robots.

[www.cyberbotics.com](http://www.cyberbotics.com)
Webots™ PRO: the Research Tool

Webots™ PRO is designed for research and development projects. It includes the possibility to create supervisor processes for controlling robotic experiments, an extended physics programming capability and fast simulation modes for CPU intensive simulations.

Webots™ EDU: for the Classroom

Webots™ EDU is tailored for classrooms. Students learn how to model robots, create their own environments and program the behavior of the robots. To validate their models, they can optionally transfer their control programs to a real robot.

<table>
<thead>
<tr>
<th>Webots™</th>
<th>EDU</th>
<th>PRO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervisor capability</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Physics plugin programming</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Fast simulation mode</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Robot design and programming</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Transfer to real robots</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Floating license option</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Multi-platform: Windows, Mac &amp; Linux</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>One year Premier Service included</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

System Requirements

**Operating system:** Windows 7 or 8, Mac OS X 10.9 or 10.10, Ubuntu Linux 14.04 or 14.10.

**Graphics card:** nVidia or ATI, at least 512 MB RAM.

**CPU:**
dual or quad-core, at least 2 GB RAM.

**Hard disk:**
at least 3GB free space.

Documentation

Webots™ comes with complete documentation, including two extensively illustrated manuals provided as PDF and HTML files:

Webots™ User Guide explains how to install and get started with Webots™. This manual includes a step-by-step tutorial for modelling and programming your own robot, and describes a number of sample simulations included in the Webots™ distribution. It explains the basic principles of Webots™ and shows the procedure for transferring your control programs to real robots.

Webots™ Reference Manual provides a complete description of each item that can be simulated with Webots™, including robots, sensors, actuators and passive objects. The programming interfaces for C, C++, Java, Python and Matlab are exhaustively documented and examplified.

In addition, a large number of sample simulations with open source code can be used as a starting point for developing your own simulations.

User Support: Premier Service

All Webots™ licenses include a one year Premier Service with personal e-mail support, maintenance and upgrades. Cyberbotics also maintains a user forum where questions are answered within 24 business hours.