

CoppeliaSim Legacy remote API

`simxCheckCollision`: Checks whether two entities collide.

`simxCheckDistance`: Measures the distance between two entities.

`simxCreateDummy`: Creates a [dummy](#) in the scene.

`simxGetObjectHandle`: Retrieves an object handle based on its path and alias.

`simxGetObjectPosition`

`simxGetObjectOrientation`

`simxGetObjectVelocity`: Retrieves the linear and angular velocity of an object.

`simxGetPingTime`: Retrieves the time needed for a command to be sent to the server, executed, and sent back. That time depends on various factors like the client settings, the network load, whether a simulation is running, whether the simulation is real-time, the simulation time step, etc. The function is blocking. This is a remote API helper function.

`simxGetVisionSensorDepthBuffer`: Retrieves the depth buffer of a vision sensor.

`simxGetVisionSensorImage`: Retrieves the image of a vision sensor.

`simxReadProximitySensor`: Reads the state of a proximity sensor. This function doesn't perform detection, it merely reads the result from a previous call to [sim.handleProximitySensor](#) (`sim.handleProximitySensor` is called in the default main script). See also [simxGetObjectGroupData](#).

`simxReadVisionSensor`: Reads the state of a vision sensor. This function doesn't perform detection, it merely reads the result from a previous call to [sim.handleVisionSensor](#) (`sim.handleVisionSensor` is called in the default main script).

`simxSetObjectOrientation`

`simxSetObjectPosition`

`simxStart`: Starts a communication thread with the server (i.e. CoppeliaSim). A same client may start several communication threads (but only one communication thread for a given IP and port). This should be the very first remote API function called on the client side. Make sure to start an appropriate remote API server service on the server side, that will wait for a connection. See also [simxFinish](#). This is a remote API helper function.

`simxStartSimulation`: Requests a start of a simulation (or a resume of a paused simulation).

`simxStopSimulation`

`simxSynchronous`: Enables or disables the stepped mode for the remote API server service that the client is connected to. The function is blocking. While in stepped mode, the client application is in charge of triggering the next simulation step. Only pre-enabled remote API server services will successfully execute this function.

`simxSynchronousTrigger`