



## Remote API Functions (C/C++)

### simxAddStatusBarMessage

regular API equivalent: [simAddStatusBarMessage](#)  
ROS service equivalent: [simRosAddStatusBarMessage](#)

Description	Adds a message to the status bar.
C synopsis	<code>simxInt simxAddStatusBarMessage(simxInt clientID,const simxChar* message,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>message</b> : the message to display <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxAppendStringSignal (ROS service equivalent: [simRosAppendStringSignal](#))

Description	DEPRECATED. Refer to <a href="#">simxWriteStringStream</a> instead.  Appends a string to a string signal. If that signal is not yet present, it is added. See also <a href="#">simxSetStringSignal</a> .
C synopsis	<code>simxInt simxAppendStringSignal(simxInt clientID,const simxChar* signalName,const simxUChar* signalValueToAppend,simxInt signalLength,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValueToAppend</b> : value to append to the signal. That value may contain any value, including embedded zeros. <b>signalLength</b> : size of the signalValueToAppend string. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxAuxiliaryConsoleClose

regular API equivalent: [simAuxiliaryConsoleClose](#)  
ROS service equivalent: [simRosAuxiliaryConsoleClose](#)

Description	Closes an auxiliary console window. See also <a href="#">simxAuxiliaryConsoleOpen</a> .
C synopsis	<code>simxInt simxAuxiliaryConsoleClose(simxInt clientID,simxInt consoleHandle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>consoleHandle</b> : the handle of the console window, previously returned by the <a href="#">simxAuxiliaryConsoleOpen</a> command <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxAuxiliaryConsoleOpen

regular API equivalent: [simAuxiliaryConsoleOpen](#)  
ROS service equivalent: [simRosAuxiliaryConsoleOpen](#)

Description	Opens an auxiliary console window for text display. This console window is different from the application main console window. Console window handles are shared across all simulator scenes. See also <a href="#">simxAuxiliaryConsolePrint</a> , <a href="#">simxAuxiliaryConsoleShow</a> and <a href="#">simxAuxiliaryConsoleClose</a> .
C synopsis	<code>simxInt simxAuxiliaryConsoleOpen(simxInt clientID,const simxChar* title,simxInt maxLines,simxInt mode,simxInt* position,simxInt* size,simxFloat* textColor,simxFloat* backgroundColor,simxInt* consoleHandle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>title</b> : the title of the console window <b>maxLines</b> : the number of text lines that can be displayed and buffered <b>mode</b> : bit-coded value. Bit0 set indicates that the console window will automatically close at simulation end, bit1 set indicates that lines will be wrapped, bit2 set indicates that the user can close the console window, bit3 set indicates that the console will automatically be hidden during simulation pause, bit4 set indicates that the console will not automatically hide when the user switches to another scene. <b>position</b> : the initial position of the console window (x and y value). Can be NULL <b>size</b> : the initial size of the console window (x and y value). Can be NULL <b>textColor</b> : the color of the text (rgb values, 0-1). Can be NULL <b>backgroundColor</b> : the background color of the console window (rgb values, 0-1). Can be NULL

	<b>consoleHandle</b> : pointer to a location receiving the handle of the created console <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxAuxiliaryConsolePrint**  
regular API equivalent: [simxAuxiliaryConsolePrint](#)  
ROS service equivalent: [simRosAuxiliaryConsolePrint](#)

Description	Prints to an auxiliary console window. See also <a href="#">simxAuxiliaryConsoleOpen</a> .
C synopsis	<code>simxInt simxAuxiliaryConsolePrint(simxInt clientID,simxInt consoleHandle,const simxChar* txt,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>consoleHandle</b> : the handle of the console window, previously returned by the <a href="#">simxAuxiliaryConsoleOpen</a> command <b>txt</b> : the text to append, or NULL to clear the console window <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxAuxiliaryConsoleShow**  
regular API equivalent: [simxAuxiliaryConsoleShow](#)  
ROS service equivalent: [simRosAuxiliaryConsoleShow](#)

Description	Shows or hides an auxiliary console window. See also <a href="#">simxAuxiliaryConsoleOpen</a> and <a href="#">simxAuxiliaryConsoleClose</a> .
C synopsis	<code>simxInt simxAuxiliaryConsoleShow(simxInt clientID,simxInt consoleHandle,simxChar showState,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>consoleHandle</b> : the handle of the console window, previously returned by the <a href="#">simxAuxiliaryConsoleOpen</a> command <b>showState</b> : indicates whether the console should be hidden (0) or shown (!=0) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxBreakForceSensor**  
regular API equivalent: [simxBreakForceSensor](#)  
ROS service equivalent: [simRosBreakForceSensor](#)

Description	Allows breaking a force sensor during simulation. A broken force sensor will lose its positional and orientational constraints. See also <a href="#">simxReadForceSensor</a> .
C synopsis	<code>simxInt simxBreakForceSensor(simxInt clientID,simxInt forceSensorHandle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>forceSensorHandle</b> : handle of the force sensor <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxClearFloatSignal**  
regular API equivalent: [simxClearFloatSignal](#)  
ROS service equivalent: [simRosClearFloatSignal](#)

Description	Clears a float signal (removes it). See also <a href="#">simxSetFloatSignal</a> , <a href="#">simxClearIntegerSignal</a> and <a href="#">simxClearStringSignal</a> .
C synopsis	<code>simxInt simxClearFloatSignal(simxInt clientID,const simxChar* signalName,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal or an empty string to clear all float signals <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxClearIntegerSignal**  
regular API equivalent: [simxClearIntegerSignal](#)  
ROS service equivalent: [simRosClearIntegerSignal](#)

Description	Clears an integer signal (removes it). See also <a href="#">simxSetIntegerSignal</a> , <a href="#">simxClearFloatSignal</a> and <a href="#">simxClearStringSignal</a> .
C synopsis	<code>simxInt simxClearIntegerSignal(simxInt clientID,const simxChar* signalName,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal or an empty string to clear all integer signals <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxClearStringSignal**regular API equivalent: [simxClearStringSignal](#)ROS service equivalent: [simRosClearStringSignal](#)

Description	Clears a string signal (removes it). See also <a href="#">simxSetStringSignal</a> , <a href="#">simxClearIntegerSignal</a> and <a href="#">simxClearFloatSignal</a> .
C synopsis	<code>simxInt simxClearStringSignal(simxInt clientID,const simxChar* signalName,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal or an empty string to clear all string signals <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxCloseScene**regular API equivalent: [simxCloseScene](#)ROS service equivalent: [simRosCloseScene](#)

Description	Closes current scene, and switches to another open scene. If there is no other open scene, a new scene is then created. Should only be called when simulation is not running and is only executed by <a href="#">continuous remote API server services</a> . See also <a href="#">simxLoadScene</a> .
C synopsis	<code>simxInt simxCloseScene(simxInt clientID,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxCopyPasteObjects**regular API equivalent: [simxCopyPasteObjects](#)ROS service equivalent: [simRosCopyPasteObjects](#)

Description	Copies and pastes objects, together with all their associated calculation objects and child scripts. To copy and paste whole models, you can simply copy and paste the model base object.
C synopsis	<code>simxInt simxCopyPasteObjects(simxInt clientID,const simxInt* objectHandles,simxInt objectCount,simxInt** newObjectHandles,simxInt* newObjectCount,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandles</b> : an array containing the handles of the objects to copy <b>objectCount</b> : the number of handles the above array contains <b>newObjectHandles</b> : a pointer to a value receiving an array of handles of newly created objects. Individual objects of a new model are not returned, but only the model base. The array remains valid until next remote API function is called. <b>newObjectCount</b> : a pointer to a value receiving the number of handles the above array contains <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxCreateBuffer (regular API equivalent: [simxCreateBuffer](#))**

Description	Creates a buffer. The buffer needs to be released with <a href="#">simxReleaseBuffer</a> except otherwise explicitly specified. This is a remote API helper function.
C synopsis	<code>simxUChar* simxCreateBuffer(simxInt bufferSize)</code>
C parameters	<b>bufferSize</b> : size of the buffer in bytes
C return value	A pointer to the created buffer
Other languages	<a href="#">Python</a> , <a href="#">Matlab</a>

**simxCreateDummy**regular API equivalent: [simxCreateDummy](#)ROS service equivalent: [simRosCreateDummy](#)

Description	Creates a <b>dummy</b> in the scene.
C synopsis	simxInt simxCreateDummy(simxInt clientID,simxFloat size,const simxUChar* colors,simxInt* dummyHandle,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>size</b> : the size of the dummy. <b>colors</b> : 4*3 bytes (0-255) for ambient_diffuse RGB, 3 reserved values (set to zero), specular RGB and emissive RGB. Can be NULL for default colors. <b>dummyHandle</b> : pointer to a value that will receive the dummy handle. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxDisplayDialog

regular API equivalent: [simDisplayDialog](#)  
ROS service equivalent: [simRosDisplayDialog](#)

Description	Displays a generic dialog box during simulation (and only during simulation!). Use in conjunction with <a href="#">simxGetDialogResult</a> , <a href="#">simxGetDialogInput</a> and <a href="#">simxEndDialog</a> . Use <a href="#">custom user interfaces</a> instead if a higher customization level is required.
C synopsis	simxInt simxDisplayDialog(simxInt clientID,const simxChar* titleText,const simxChar* mainText,simxInt dialogType,const simxChar* initialText,simxFloat* titleColors,simxFloat* dialogColors,simxInt* dialogHandle,simxInt* uiHandle,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>titleText</b> : Title bar text <b>mainText</b> : Information text <b>dialogType</b> : a <a href="#">generic dialog style</a> <b>initialText</b> : Initial text in the edit box if the dialog is of type <code>sim_dlgstyle_input</code> . Cannot be NULL! <b>titleColors</b> : Title bar color (6 <code>simxFloat</code> values for RGB for background and foreground), can be NULL for default colors <b>dialogColors</b> : Dialog color (6 <code>simxFloat</code> values for RGB for background and foreground), can be NULL for default colors <b>dialogHandle</b> : handle of the generic dialog (different from custom user interface handle!! (see hereafter)). This handle should be used with the following functions: <a href="#">simxGetDialogResult</a> , <a href="#">simxGetDialogInput</a> and <a href="#">simxEndDialog</a> . <b>uiHandle</b> : a pointer to a value accepting the handle of the corresponding custom user interface. Can be NULL <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxEndDialog

regular API equivalent: [simEndDialog](#)  
ROS service equivalent: [simRosEndDialog](#)

Description	Closes and releases resource from a previous call to <a href="#">simxDisplayDialog</a> . Even if the dialog is not visible anymore, you should release resources by using this function (however at the end of a simulation, all dialog resources are automatically released).
C synopsis	simxInt simxEndDialog(simxInt clientID,simxInt dialogHandle,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>dialogHandle</b> : handle of generic dialog (return value of <a href="#">simxDisplayDialog</a> ) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxEraseFile (ROS service equivalent: [simRosEraseFile](#))

Description	Erases a file on the server side. This function is used by several other functions internally (e.g. <a href="#">simxLoadModel</a> ). See also <a href="#">simxTransferFile</a> . This is a remote API helper function.
C synopsis	simxInt simxEraseFile(simxInt clientID,const simxChar* fileName_serverSide,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>fileName_serverSide</b> : the file to erase on the server side. For now, do not specify a path (the file will be erased in the remote API plugin directory) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxFinish

Description	Ends the communication thread. This should be the very last remote API function called on the client
-------------	--

	side. <code>simxFinish</code> should only be called after a successful call to <code>simxStart</code> . This is a remote API helper function.
C synopsis	<code>simxVoid simxFinish(simxInt clientID)</code>
C parameters	<b>clientID</b> : the client ID. refer to <code>simxStart</code> . Can be -1 to end all running communication threads.
C return value	none
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetAndClearStringSignal

Description	DEPRECATED. Refer to <code>simxReadStream</code> instead.  Gets the value of a string signal, then clears it. Useful to retrieve continuous data from the server. See also <code>simxGetStringSignal</code> .
C synopsis	<code>simxInt simxGetAndClearStringSignal(simxInt clientID,const simxChar* signalName,simxUChar** signalValue,simxInt* signalLength,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <code>simxStart</code> . <b>signalName</b> : name of the signal <b>signalValue</b> : pointer to a pointer receiving the value of the signal. The signal value will remain valid until next remote API call <b>signalLength</b> : pointer to a location receiving the value of the signal length, since it may contain any data (also embedded zeros). <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Since this function will clear a read signal, and we cannot afford to wait for a reply (well, we could, but that would mean a blocking operation), the function operates in a special mode and should be used as in following example:  <pre>// Initialization phase: simxUChar* signal; simxInt sLength; simxGetAndClearStringSignal(cid,"sig",&amp;signal,&amp;sLength,simx_opmode_streaming);  // while we are connected: while (simxGetConnectionId(cid)!=-1) {     if (simxGetAndClearStringSignal(cid,"sig",&amp;signal,&amp;sLength,simx_opmode_buffer)==         simx_return_ok)     {         // A signal was retrieved!         // Enable streaming again (was automatically disabled with the positive event):         simxGetAndClearStringSignal(cid,"sig",&amp;signal,&amp;sLength,simx_opmode_streaming);     }     .. }</pre>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetArrayParameter

regular API equivalent: `simGetArrayParameter`  
ROS service equivalent: `simRosGetArrayParameter`

Description	Retrieves 3 values from an array. See the <a href="#">array parameter identifiers</a> . See also <code>simxSetArrayParameter</code> , <code>simxGetBooleanParameter</code> , <code>simxGetIntegerParameter</code> , <code>simxGetFloatingParameter</code> and <code>simxGetStringParameter</code> .
C synopsis	<code>simxInt simxGetArrayParameter(simxInt clientID,simxInt paramIdentifier,simxFloat* paramValues,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <code>simxStart</code> . <b>paramIdentifier</b> : an <a href="#">array parameter identifier</a> <b>paramValues</b> : a pointer to 3 values that will receive the parameters <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code> (if not called on a regular basis)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetBooleanParameter

regular API equivalent: `simGetBooleanParameter`  
ROS service equivalent: `simRosGetBooleanParameter`

Description	Retrieves a boolean value. See the <a href="#">Boolean parameter identifiers</a> . See also <code>simxSetBooleanParameter</code> , <code>simxGetIntegerParameter</code> , <code>simxGetFloatingParameter</code> , <code>simxGetArrayParameter</code> and <code>simxGetStringParameter</code> .
C synopsis	<code>simxInt simxGetBooleanParameter(simxInt clientID,simxInt paramIdentifier,simxUChar* paramValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <code>simxStart</code> . <b>paramIdentifier</b> : a <a href="#">Boolean parameter identifier</a>

	<b>paramValue:</b> a pointer that will receive the parameter value <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code> (if not called on a regular basis)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetCollisionHandle

regular API equivalent: [simGetCollisionHandle](#)  
ROS service equivalent: [simRosGetCollisionHandle](#)

Description	Retrieves a collision object handle based on its name. If the client application is launched from a <a href="#">child script</a> , then you could also let the child script figure out what handle correspond to what collision object, and send the handles as additional arguments to the client application during its launch. See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetCollisionHandle(simxInt clientID,const simxChar* collisionObjectName,simxInt* handle,simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>collisionObjectName:</b> name of the collision object. If possible, don't rely on the <a href="#">automatic name adjustment mechanism</a> , and always specify the full collision object name, including the #: if the collision object is "myCollision", specify "myCollision#", if the collision object is "myCollision#0", specify "myCollision#0", etc. <b>handle:</b> pointer to a value that will receive the handle <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetConnectionId

Description	Returns the ID of the current connection. Use this function to track the connection state to the server. See also <a href="#">simxStart</a> . This is a remote API helper function.
C synopsis	<code>simxInt simxGetConnectionId(simxInt clientID)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> .
C return value	a connection ID, or -1 if the client is not connected to the server. Different connection IDs indicate temporary disconnections in-between.
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetDialogInput

regular API equivalent: [simGetDialogInput](#)  
ROS service equivalent: [simRosGetDialogInput](#)

Description	Queries the text the user entered into a generic dialog box of style <code>sim_dlgstyle_input</code> . To be used after <a href="#">simxDisplayDialog</a> was called and after <a href="#">simxGetDialogResult</a> returned <code>sim_dlgret_ok</code> .
C synopsis	<code>simxInt simxGetDialogInput(simxInt clientID,simxInt dialogHandle,simxChar** inputText,simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>dialogHandle:</b> handle of generic dialog (return value of <a href="#">simxDisplayDialog</a> ) <b>inputText:</b> pointer to a pointer receiving the string the user entered. The pointer is valid until another remote API function is called. <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetDialogResult

regular API equivalent: [simGetDialogResult](#)  
ROS service equivalent: [simRosGetDialogResult](#)

Description	Queries the result of a dialog box. To be used after <a href="#">simxDisplayDialog</a> was called.
C synopsis	<code>simxInt simxGetDialogResult(simxInt clientID,simxInt dialogHandle,simxInt* result,simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>dialogHandle:</b> handle of generic dialog (return value of <a href="#">simxDisplayDialog</a> ) <b>result:</b> pointer receiving the <a href="#">result value</a> .  Note. If the return value is <code>sim_dlgret_still_open</code> , the dialog was not closed and no button was pressed. Otherwise, you should free resources with <a href="#">simxEndDialog</a> (the dialog might not be visible anymore, but is still present) <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetDistanceHandle**regular API equivalent: [simxGetDistanceHandle](#)ROS service equivalent: [simRosGetDistanceHandle](#)

Description	Retrieves a distance object handle based on its name. If the client application is launched from a <a href="#">child script</a> , then you could also let the child script figure out what handle correspond to what distance object, and send the handles as additional arguments to the client application during its launch. See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetDistanceHandle(simxInt clientID,const simxChar* distanceObjectName,simxInt* handle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>distanceObjectName</b> : name of the distance object. If possible, don't rely on the <a href="#">automatic name adjustment mechanism</a> , and always specify the full distance object name, including the #: if the distance object is "myDistance", specify "myDistance#", if the distance object is "myDistance#0", specify "myDistance#0", etc. <b>handle</b> : pointer to a value that will receive the handle <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_one-shot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetFloatingParameter**regular API equivalent: [simxGetFloatingParameter](#)ROS service equivalent: [simRosGetFloatingParameter](#)

Description	Retrieves a floating point value. See the <a href="#">floating-point parameter identifiers</a> . See also <a href="#">simxSetFloatingParameter</a> , <a href="#">simxGetBooleanParameter</a> , <a href="#">simxGetIntegerParameter</a> , <a href="#">simxGetArrayParameter</a> and <a href="#">simxGetStringParameter</a> .
C synopsis	<code>simxInt simxGetFloatingParameter(simxInt clientID,simxInt paramIdentifier,simxFloat* paramValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>paramIdentifier</b> : a <a href="#">floating parameter identifier</a> <b>paramValue</b> : a pointer that will receive the parameter value <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_one-shot_wait</code> (if not called on a regular basis)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetFloatSignal**regular API equivalent: [simxGetFloatSignal](#)ROS service equivalent: [simRosGetFloatSignal](#)

Description	Gets the value of a float signal. Signals are cleared at simulation start. See also <a href="#">simxSetFloatSignal</a> , <a href="#">simxClearFloatSignal</a> , <a href="#">simxGetIntegerSignal</a> and <a href="#">simxGetStringSignal</a> .
C synopsis	<code>simxInt simxGetFloatSignal(simxInt clientID,const simxChar* signalName,simxFloat* signalValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValue</b> : pointer to a location receiving the value of the signal <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetInMessageInfo**

Description	Retrieves information about the last received message from the server. This is a remote API helper function. See also <a href="#">simxGetOutMessageInfo</a> .  If the client didn't receive any command reply from the server for a while, the data retrieved with this function won't be up-to-date. In order to avoid this, you should start at least one streaming command, which will guarantee regular message income.
C synopsis	<code>simxInt simxGetInMessageInfo(simxInt clientID,simxInt infoType,simxInt* info)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>infoType</b> : an <a href="#">inbox message info type</a> <b>info</b> : pointer to a <code>simxInt</code> value receiving the requested information
C return value	-1 in case of an error
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetIntegerParameter**regular API equivalent: [simxGetIntegerParameter](#)

ROS service equivalent: [simRosGetIntegerParameter](#)

Description	Retrieves an integer value. See the <a href="#">integer parameter identifiers</a> . See also <a href="#">simxSetIntegerParameter</a> , <a href="#">simxGetBooleanParameter</a> , <a href="#">simxGetFloatingParameter</a> , <a href="#">simxGetArrayParameter</a> and <a href="#">simxGetStringParameter</a> .
C synopsis	<code>simxInt simxGetIntegerParameter(simxInt clientID,simxInt paramIdentifier,simxInt* paramValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>paramIdentifier</b> : an <a href="#">integer parameter identifier</a> <b>paramValue</b> : a pointer that will receive the parameter value <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code> (if not called on a regular basis)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetIntegerSignal**regular API equivalent: [simGetIntegerSignal](#)ROS service equivalent: [simRosGetIntegerSignal](#)

Description	Gets the value of an integer signal. Signals are cleared at simulation start. See also <a href="#">simxSetIntegerSignal</a> , <a href="#">simxClearIntegerSignal</a> , <a href="#">simxGetFloatSignal</a> and <a href="#">simxGetStringSignal</a> .
C synopsis	<code>simxInt simxGetIntegerSignal(simxInt clientID,const simxChar* signalName,simxInt* signalValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValue</b> : pointer to a location receiving the value of the signal <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetJointForce**regular API equivalent: [simGetJointForce](#)ROS service equivalent: [simRosGetJointState](#)

Description	Retrieves the force or torque applied to a joint along/about its active axis. This function retrieves meaningful information only if the joint is prismatic or revolute, and is dynamically enabled. With the <a href="#">Bullet</a> engine, this function returns the force or torque applied to the joint motor (torques from joint limits are not taken into account). With the <a href="#">ODE</a> or <a href="#">Vortex</a> engine, this function returns the total force or torque applied to a joint along/about its z-axis. See also <a href="#">simxSetJointForce</a> , <a href="#">simxReadForceSensor</a> and <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetJointForce(simxInt clientID,simxInt jointHandle,simxFloat* force,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>force</b> : the force or the torque applied to the joint along/about its z-axis <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetJointMatrix**regular API equivalent: [simGetJointMatrix](#)ROS service equivalent: [simRosGetJointMatrix](#)

Description	Retrieves the intrinsic transformation matrix of a joint (the transformation caused by the joint movement). See also <a href="#">simxSetSphericalJointMatrix</a> .
C synopsis	<code>simxInt simxGetJointMatrix(simxInt clientID,simxInt jointHandle,simxFloat* matrix,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>matrix</b> : pointer to 12 <code>simxFloat</code> values. See the regular API equivalent function for details <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetJointPosition**regular API equivalent: [simGetJointPosition](#)ROS service equivalent: [simRosGetJointState](#)

Description	Retrieves the intrinsic position of a joint. This function cannot be used with spherical joints (use <a href="#">simxGetJointMatrix</a> instead). See also <a href="#">simxSetJointPosition</a> and <a href="#">simxGetObjectGroupData</a> .
-------------	--



C synopsis	<code>simxInt simxGetJointPosition(simxInt clientID,simxInt jointHandle,simxFloat* position,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>position</b> : intrinsic position of the joint. This is a one-dimensional value: if the joint is revolute, the rotation angle is returned, if the joint is prismatic, the translation amount is returned, etc. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetLastCmdTime

Description	Retrieves the simulation time of the last fetched command (i.e. when the last fetched command was processed on the server side). The function can be used to verify how "fresh" a command reply is, or whether a command reply was recently updated. For example: <pre> if (simxGetVisionSensorImage(cid,handle,res,&amp;img,0,sim_opmode_buffer)==simx_return_ok)     imageAcquisitionTime=simxGetLastCmdTime(cid); </pre> <p>This is a remote API helper function.</p>
C synopsis	<code>simxInt simxGetLastCmdTime(simxInt clientID)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> .
C return value	The simulation time in milliseconds when the command reply was generated, or 0 if simulation was not running.
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetLastErrors

regular API equivalent: [simGetLastError](#)  
ROS service equivalent: [simRosGetLastErrors](#)

Description	Retrieves the last 50 errors that occurred on the server side, and clears the error buffer there. Only errors that occurred because of this client will be reported.
C synopsis	<code>simxInt simxGetLastErrors(simxInt clientID,simxInt* errorCnt,simxChar** errorStrings,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>errorCnt</b> : pointer receiving the number of error strings returned <b>errorStrings</b> : pointer to a string pointer receiving the address of an error string (individual strings are separated by a zero character). The address is valid until next remote API function is called <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls) when not debugging. For debugging purposes, use <code>simx_opmode_oneshot_wait</code> .
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetModelProperty

regular API equivalent: [simGetModelProperty](#)  
ROS service equivalent: [simRosGetModelProperty](#)

Description	Retrieves the properties of a model. See also <a href="#">simxSetModelProperty</a> .
C synopsis	<code>simxInt simxGetModelProperty(simxInt clientID,simxInt objectHandle,simxInt* prop,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>prop</b> : pointer to a location receiving the <a href="#">model property value</a> <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls), or <code>simx_opmode_oneshot_wait</code> (depending on the intended usage)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxGetObjectChild

regular API equivalent: [simGetObjectChild](#)  
ROS service equivalent: [simRosGetObjectChild](#)

Description	Retrieves the handle of an object's child object. See also <a href="#">simGetObjectParent</a> .
C synopsis	<code>simxInt simxGetObjectChild(simxInt clientID,simxInt parentObjectHandle,simxInt childIndex,simxInt* childObjectHandle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>parentObjectHandle</b> : handle of the object

	<b>childIndex</b> : zero-based index of the child's position. To retrieve all children of an object, call the function by increasing the index until the child handle is -1 <b>childObjectHandle</b> : pointer to a value receiving the handle of the child object. If the value is -1, there is no child at the given index <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectFloatParameter**regular API equivalent: [simGetObjectFloatParameter](#)ROS service equivalent: [simRosGetObjectFloatParameter](#)

Description	Retrieves a floating-point parameter of an object. See also <a href="#">simxSetObjectFloatParameter</a> and <a href="#">simxGetObjectIntParameter</a> .
C synopsis	<code>simxInt simxGetObjectFloatParameter(simxInt clientID, simxInt objectHandle, simxInt parameterID, simxFloat* parameterValue, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>parameterID</b> : identifier of the parameter to retrieve. See the <a href="#">list of all possible object parameter identifiers</a> <b>parameterValue</b> : pointer to a location that will receive the value of the parameter <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls), or <code>simx_opmode_oneshot_wait</code> (depending on the intended usage)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectGroupData (ROS service equivalent: [simRosGetObjectGroupData](#))**

Description	Simultaneously retrieves data of various objects in a V-REP scene.
C synopsis	<code>simxInt simxGetObjectGroupData(simxInt clientID, simxInt objectType, simxInt dataType, simxInt* handlesCount, simxInt** handles, simxInt* intDataCount, simxInt** intData, simxInt* floatDataCount, simxFloat** floatData, simxInt* stringDataCount, simxChar** stringData, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectType</b> : a <a href="#">scene object type</a> , or <code>sim_appobj_object_type</code> for all <a href="#">scene objects</a> . <b>dataType</b> : the type of data that is desired: <ul style="list-style-type: none"> <li>0: retrieves the object names (in <code>stringData</code>.)</li> <li>1: retrieves the <a href="#">object types</a> (in <code>intData</code>)</li> <li>2: retrieves the parent object handles (in <code>intData</code>)</li> <li>3: retrieves the absolute object positions (in <code>floatData</code>. There are 3 values for each object (x,y,z))</li> <li>4: retrieves the local object positions (in <code>floatData</code>. There are 3 values for each object (x,y,z))</li> <li>5: retrieves the absolute object orientations as <a href="#">Euler angles</a> (in <code>floatData</code>. There are 3 values for each object (alpha,beta,gamma))</li> <li>6: retrieves the local object orientations as <a href="#">Euler angles</a> (in <code>floatData</code>. There are 3 values for each object (alpha,beta,gamma))</li> <li>7: retrieves the absolute object orientations as quaternions (in <code>floatData</code>. There are 4 values for each object (qx,qy,qz,qw))</li> <li>8: retrieves the local object orientations as quaternions (in <code>floatData</code>. There are 4 values for each object (qx,qy,qz,qw))</li> <li>9: retrieves the absolute object positions and orientations (as <a href="#">Euler angles</a>) (in <code>floatData</code>. There are 6 values for each object (x,y,z,alpha,beta,gamma))</li> <li>10: retrieves the local object positions and orientations (as <a href="#">Euler angles</a>) (in <code>floatData</code>. There are 6 values for each object (x,y,z,alpha,beta,gamma))</li> <li>11: retrieves the absolute object positions and orientations (as quaternions) (in <code>floatData</code>. There are 7 values for each object (x,y,z,qx,qy,qz,qw))</li> <li>12: retrieves the local object positions and orientations (as quaternions) (in <code>floatData</code>. There are 7 values for each object (x,y,z,qx,qy,qz,qw))</li> <li>13: retrieves proximity sensor data (in <code>intData</code> (2 values): detection state, detected object handle. In <code>floatData</code> (6 values): detected point (x,y,z) and detected surface normal (nx,ny,nz))</li> <li>14: retrieves force sensor data (in <code>intData</code> (1 values): force sensor state. In <code>floatData</code> (6 values): force (fx,fy,fz) and torque (tx,ty,tz))</li> <li>15: retrieves joint state data (in <code>floatData</code> (2 values): position, force/torque)</li> <li>16: retrieves joint properties data (in <code>intData</code> (2 values): joint type, joint mode (bit16=hybrid operation). In <code>floatData</code> (2 values): joint limit low, joint range (-1.0 if joint is cyclic))</li> <li>17: retrieves the object linear velocity (in <code>floatData</code>. There are 3 values for each object (vx,vy,vz))</li> <li>18: retrieves the object angular velocity as <a href="#">Euler angles</a> per seconds (in <code>floatData</code>. There are 3 values for each object (dAlpha,dBeta,dGamma))</li> <li>19: retrieves the object linear and angular velocity (in <code>floatData</code>. There are 6 values for each object (vx,vy,vz,dAlpha,dBeta,dGamma))</li> </ul> <b>handlesCount</b> (output): the number of returned object handles. Can be NULL. <b>handles</b> (output): the object handles. The pointer remains valid until the next remote API call. Can be NULL.

	<p><b>intDataCount</b> (output): the number of returned integer values. Can be NULL.</p> <p><b>intData</b> (output): the integer values. The pointer remains valid until the next remote API call. Can be NULL.</p> <p><b>floatDataCount</b> (output): the number of returned float values. Can be NULL.</p> <p><b>floatData</b> (output): the float values. The pointer remains valid until the next remote API call. Can be NULL.</p> <p><b>stringDataCount</b> (output): the number of returned strings. Can be NULL.</p> <p><b>stringData</b> (output): the strings. Each string is separated by the zero char ('\0'). The pointer remains valid until the next remote API call. Can be NULL.</p> <p><b>operationMode</b>: a <a href="#">remote API function operation mode</a>. Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code> or <code>simx_opmode_streaming</code>.</p>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectHandle**regular API equivalent: [simGetObjectHandle](#)ROS service equivalent: [simRosGetObjectHandle](#)

Description	Retrieves an object handle based on its name. If the client application is launched from a <a href="#">child script</a> , then you could also let the child script figure out what handle correspond to what objects, and send the handles as additional arguments to the client application during its launch. See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetObjectHandle(simxInt clientID,const simxChar* objectName,simxInt* handle,simxInt operationMode)</code>
C parameters	<p><b>clientID</b>: the client ID. refer to <a href="#">simxStart</a>.</p> <p><b>objectName</b>: name of the object. If possible, don't rely on the <a href="#">automatic name adjustment mechanism</a>, and always specify the full object name, including the #: if the object is "myJoint", specify "myJoint#", if the object is "myJoint#0", specify "myJoint#0", etc.</p> <p><b>handle</b>: pointer to a value that will receive the handle</p> <p><b>operationMode</b>: a <a href="#">remote API function operation mode</a>. Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code></p>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectIntParameter**regular API equivalent: [simGetObjectIntParameter](#)ROS service equivalent: [simRosGetObjectIntParameter](#)

Description	Retrieves an integer parameter of a object. See also <a href="#">simxSetObjectIntParameter</a> and <a href="#">simxGetObjectFloatParameter</a> .
C synopsis	<code>simxInt simxGetObjectIntParameter(simxInt clientID,simxInt objectHandle,simxInt parameterID,simxInt* parameterValue,simxInt operationMode)</code>
C parameters	<p><b>clientID</b>: the client ID. refer to <a href="#">simxStart</a>.</p> <p><b>objectHandle</b>: handle of the object</p> <p><b>parameterID</b>: identifier of the parameter to retrieve. See the <a href="#">list of all possible object parameter identifiers</a></p> <p><b>parameterValue</b>: pointer to a location that will receive the value of the parameter</p> <p><b>operationMode</b>: a <a href="#">remote API function operation mode</a>. Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls), or <code>simx_opmode_oneshot_wait</code> (depending on the intended usage)</p>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectOrientation**regular API equivalent: [simGetObjectOrientation](#)ROS service equivalent: [simRosGetObjectPose](#)

Description	Retrieves the orientation ( <a href="#">Euler angles</a> ) of an object. See also <a href="#">simxSetObjectOrientation</a> , <a href="#">simxGetObjectPosition</a> and <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetObjectOrientation(simxInt clientID,simxInt objectHandle,simxInt relativeToObjectHandle,simxFloat* eulerAngles,simxInt operationMode)</code>
C parameters	<p><b>clientID</b>: the client ID. refer to <a href="#">simxStart</a>.</p> <p><b>objectHandle</b>: handle of the object</p> <p><b>relativeToObjectHandle</b>: indicates relative to which reference frame we want the orientation. Specify -1 to retrieve the absolute orientation, <code>sim_handle_parent</code> to retrieve the orientation relative to the object's parent, or an object handle relative to whose reference frame you want the orientation</p> <p><b>eulerAngles</b>: pointer to 3 values receiving the Euler angles (alpha, beta and gamma)</p> <p><b>operationMode</b>: a <a href="#">remote API function operation mode</a>. Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)</p>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectParent**

regular API equivalent: [simGetObjectParent](#)  
 ROS service equivalent: [simRosGetObjectParent](#)

Description	Retrieves the handle of an object's parent object. See also <a href="#">simxGetObjectChild</a> and <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetObjectParent(simxInt clientID,simxInt objectHandle,simxInt* parentObjectHandle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>parentObjectHandle</b> : pointer to a value receiving the handle of the parent object. If the value is -1, the object has no parent <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectPosition**  
 regular API equivalent: [simGetObjectPosition](#)  
 ROS service equivalent: [simRosGetObjectPose](#)

Description	Retrieves the position of an object. See also <a href="#">simxSetObjectPosition</a> , <a href="#">simxGetObjectOrientation</a> and <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetObjectPosition(simxInt clientID,simxInt objectHandle,simxInt relativeToObjectHandle,simxFloat* position,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>relativeToObjectHandle</b> : indicates relative to which reference frame we want the position. Specify -1 to retrieve the absolute position, <code>sim_handle_parent</code> to retrieve the position relative to the object's parent, or an object handle relative to whose reference frame you want the position <b>position</b> : pointer to 3 values receiving the position <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjects**  
 regular API equivalent: [simGetObjects](#)  
 ROS service equivalent: [simRosGetObjects](#)

Description	Retrieves object handles of a given type, or of all types (i.e. all object handles). See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetObjects(simxInt clientID,simxInt objectType,simxInt* objectCount,simxInt** objectHandles,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectType</b> : <a href="#">object type</a> ( <code>sim_object_shape_type</code> , <code>sim_object_joint_type</code> , etc., or <code>sim_handle_all</code> for any type of object) <b>objectCount</b> : pointer to a value that will receive the number of retrieved handles <b>objectHandles</b> : pointer to a pointer that will receive an object handle array. The array remains valid until next remote API function is called. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectSelection**  
 regular API equivalent: [simGetObjectSelection](#)  
 ROS service equivalent: [simRosGetObjectSelection](#)

Description	Retrieves all selected object's handles. See also <a href="#">simxSetObjectSelection</a> .
C synopsis	<code>simxInt simxGetObjectSelection(simxInt clientID,simxInt** objectHandles,simxInt* objectCount,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandles</b> : pointer to a pointer that will receive an object handle array. The array remains valid until next remote API function is called. <b>objectCount</b> : pointer to a value that will receive the number of elements in above's array. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls), or <code>simx_opmode_oneshot_wait</code> depending on the intent.
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetObjectVelocity**  
 regular API equivalent: [simGetObjectVelocity](#)

ROS publisher equivalent: `simros_strmcmd_get_twist_status`

Description	Retrieves the linear and angular velocity of an object. See also <a href="#">simxGetObjectPosition</a> , <a href="#">simxGetObjectOrientation</a> and <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetObjectVelocity(simxInt clientID, simxInt objectHandle, simxFLOAT* linearVelocity, simxFLOAT* angularVelocity, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>linearVelocity</b> : pointer to 3 values receiving the linearVelocity (vx, vy, vz). Can be NULL. <b>angularVelocity</b> : pointer to 3 values receiving the angularVelocity (dAlpha, dBeta, dGamma). Can be NULL. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

## simxGetOutMessageInfo

Description	Retrieves information about the next message to send to the server. This is a remote API helper function. See also <a href="#">simxGetInMessageInfo</a> .
C synopsis	<code>simxInt simxGetOutMessageInfo(simxInt clientID, simxInt infoType, simxInt* info)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>infoType</b> : an <a href="#">outbox message info type</a> <b>info</b> : pointer to a <code>simxInt</code> value receiving the requested information
C return value	-1 in case of an error
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

## simxGetPingTime

Description	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.
C synopsis	<code>simxInt simxGetPingTime(simxInt clientID, simxInt* pingTime)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>pingTime</b> : a pointer to a <code>simxInt</code> value accepting the ping time in milliseconds.
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

## simxGetStringParameter

regular API equivalent: `simGetStringParameter`ROS service equivalent: `simRosGetStringParameter`

Description	Retrieves a string value. See the <a href="#">string parameter identifiers</a> . See also <a href="#">simxGetBooleanParameter</a> , <a href="#">simxGetIntegerParameter</a> , <a href="#">simxGetArrayParameter</a> and <a href="#">simxGetFloatingParameter</a> .
C synopsis	<code>simxInt simxGetStringParameter(simxInt clientID, simxInt paramIdentifier, simxChar** paramValue, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>paramIdentifier</b> : a <a href="#">string parameter identifier</a> <b>paramValue</b> : a pointer to a pointer that will receive the string. The string pointer is valid until next remote API function is called. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code> (if not called on a regular basis)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

## simxGetStringSignal

regular API equivalent: `simGetStringSignal`ROS service equivalent: `simRosGetStringSignal`

Description	Gets the value of a string signal. Signals are cleared at simulation start. See also <a href="#">simxSetStringSignal</a> , <a href="#">simxGetAndClearStringSignal</a> , <a href="#">simxClearStringSignal</a> , <a href="#">simxGetIntegerSignal</a> and <a href="#">simxGetFloatSignal</a> .
C synopsis	<code>simxInt simxGetStringSignal(simxInt clientID, const simxChar* signalName, simxUChar** signalValue, simxInt* signalLength, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValue</b> : pointer to a pointer receiving the value of the signal. The signal value will remain valid until next remote API call <b>signalLength</b> : pointer to a location receiving the value of the signal length, since it may contain any data (also embedded zeros). <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)

C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetUIButtonProperty**regular API equivalent: [simGetUIButtonProperty](#)ROS service equivalent: [simRosGetUIButtonProperty](#)

Description	Retrieves the properties of a custom user interface button. See also <a href="#">simxSetUIButtonProperty</a> .
C synopsis	<code>simxInt simxGetUIButtonProperty(simxInt clientID,simxInt uiHandle,simxInt uiButtonID,simxInt* prop,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>uiHandle</b> : handle of the custom user interface <b>uiButtonID</b> : handle (or id) of the custom user interface button <b>prop</b> : pointer to an integer accepting a <a href="#">button property value</a> . <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetUIEventButton**regular API equivalent: [simGetUIEventButton](#)ROS service equivalent: [simRosGetUIEventButton](#)

Description	Gets the button handle (i.e. ID) and some auxiliary values of the last occurred event in that custom user interface. The function will then clear the event. When a custom user interface button is pressed, a slider is moved or an edit box is changed, an event is registered and stored in the custom user interface.
C synopsis	<code>simxInt simxGetUIEventButton(simxInt clientID,simxInt uiHandle,simxInt* uiEventButtonID,simxInt* auxValues,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>uiHandle</b> : handle of the custom user interface <b>uiEventButtonID</b> : id of the UI button where an event occurred, or -1 if no event occurred <b>auxValues</b> : 2 values (can be NULL). value1: the <a href="#">button property</a> value2: for sliders: slider state (0-1000), for stay down buttons: down state (0 or 1), for up/down event buttons: up/down events (0 or 1) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Since this function will clear a read event, and we cannot afford to wait for a reply (well, we could, but that would mean a blocking operation), the function operates in a special mode and should be used as in following example:  <pre>// Initialization phase: int uiHandle=-1; simxGetUIHandle(cid,"UI",&amp;uiHandle,simx_opmode_oneshot_wait); int buttonEventID=-1; simxGetUIEventButton(cid,uiHandle,&amp;buttonEventID,NULL,simx_opmode_streaming);  // while we are connected: while (simxGetConnectionId(cid)!=-1) {     if ((simxGetUIEventButton(cid,uiHandle,&amp;buttonEventID,NULL,simx_opmode_buffer)==         simx_return_ok)&amp;&amp;(buttonEventID!=-1) )     {         // A button was pressed/edited/changed. React to it here!         // Enable streaming again (was automatically disabled with the positive event):         simxGetUIEventButton(cid,uiHandle,&amp;buttonEventID,NULL,simx_opmode_streaming);     }     .. }</pre>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetUIHandle**regular API equivalent: [simGetUIHandle](#)ROS service equivalent: [simRosGetUIHandle](#)

Description	Retrieves the handle of a custom user interface. If the client application is launched from a <a href="#">child script</a> , then you could also let the child script figure out what handle correspond to what UIs, and send the handles as additional arguments to the client application during its launch. See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxGetUIHandle(simxInt clientID,const simxChar* uiName,simxInt* handle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>uiName</b> : name of the UI. If possible, don't rely on the <a href="#">automatic name adjustment mechanism</a> , and always specify the full object name, including the #: if the UI is "myUI", specify "myUI#", if the UI is "myUI#0", specify "myUI#0", etc.

	<b>handle:</b> pointer to a value that will receive the handle <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetUISlider**

regular API equivalent: [simGetUISlider](#)  
ROS service equivalent: [simRosGetUISlider](#)

Description	Gets the slider position of a custom user interface button (must be slider-type button). See also <a href="#">simxSetUISlider</a> .
C synopsis	<code>simxInt simxGetUISlider(simxInt clientID,simxInt uiHandle,simxInt uiButtonID,simxInt* position,simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>uiHandle:</b> handle of the custom user interface <b>uiButtonID:</b> handle of a button inside the specified custom user interface <b>position:</b> pointer to an integer receiving the slider position (value between 0 and 1000) <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetVisionSensorDepthBuffer**

regular API equivalent: [simGetVisionSensorDepthBuffer](#)  
ROS service equivalent: [simRosGetVisionSensorDepthBuffer](#)

Description	Retrieves the depth buffer of a vision sensor. The returned data doesn't make sense if <a href="#">simHandleVisionSensor</a> wasn't called previously ( <code>simHandleVisionSensor</code> is called by default in the main script if the vision sensor is not tagged as explicit handling). Use the <a href="#">simxGetLastCmdTime</a> function to verify the "freshness" of the retrieved data. See also <a href="#">simxGetVisionSensorImage</a> .
C synopsis	<code>simxInt simxGetVisionSensorDepthBuffer(simxInt clientID,simxInt sensorHandle,simxInt* resolution,simxFloat** buffer,simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>sensorHandle:</b> handle of the vision sensor <b>resolution:</b> pointer to 2 <code>simxInt</code> values receiving the resolution of the image <b>buffer:</b> pointer to a pointer to the depth buffer data. The data remains valid until next remote API function is called (i.e. the data is automatically released). Values are in the range of 0-1 (0=closest to sensor, 1=farthest from sensor). <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxGetVisionSensorImage**

regular API equivalent: [simGetVisionSensorImage](#)  
ROS service equivalent: [simRosGetVisionSensorImage](#)

Description	Retrieves the image of a vision sensor. The returned data doesn't make sense if <a href="#">simHandleVisionSensor</a> wasn't called previously ( <code>simHandleVisionSensor</code> is called by default in the main script if the vision sensor is not tagged as explicit handling). Use the <a href="#">simxGetLastCmdTime</a> function to verify the "freshness" of the retrieved data. See also <a href="#">simxSetVisionSensorImage</a> , <a href="#">simxGetVisionSensorDepthBuffer</a> and <a href="#">simxReadVisionSensor</a> .
C synopsis	<code>simxInt simxGetVisionSensorImage(simxInt clientID,simxInt sensorHandle,simxInt* resolution,simxUChar** image,simxUChar options,simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>sensorHandle:</b> handle of the vision sensor <b>resolution:</b> pointer to 2 <code>simxInt</code> values receiving the resolution of the image <b>image:</b> pointer to a pointer to the image data. The data remains valid until next remote API function is called (i.e. the data is automatically released) <b>options:</b> image options, bit-coded: bit0 set: each image pixel is a byte (greyscale image), otherwise each image pixel is a rgb byte-triplet <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxJointGetForce (DEPRECATED)**

Description	DEPRECATED. See <a href="#">simxGetJointForce</a> instead.
C synopsis	<code>simxInt simxJointGetForce(simxInt clientID,simxInt jointHandle,simxFloat* force,simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> .

	<b>jointHandle</b> : handle of the joint <b>force</b> : the force or the torque applied to the joint along/about its z-axis <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxLoadModel**regular API equivalent: [simLoadModel](#)ROS service equivalent: [simRosLoadModel](#)

Description	Loads a previously saved model. See also <a href="#">simxLoadUI</a> , <a href="#">simxLoadScene</a> and <a href="#">simxTransferFile</a> .
C synopsis	<code>simxInt simxLoadModel(simxInt clientID,const simxChar* modelPathAndName,simxUChar options,simxInt* baseHandle,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>modelPathAndName</b> : the model filename, including the path and extension ("ttm"). The file is relative to the client or server system depending on the <b>options</b> value (see next argument) <b>options</b> : options, bit-coded: bit0 set: the specified file is located on the client side (in that case the function will be blocking since the model first has to be transferred to the server). Otherwise it is located on the server side <b>baseHandle</b> : the loaded model base. Can be NULL. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxLoadScene**regular API equivalent: [simLoadScene](#)ROS service equivalent: [simRosLoadScene](#)

Description	Loads a previously saved scene. Should only be called when simulation is not running and is only executed by <a href="#">continuous remote API server services</a> . See also <a href="#">simxCloseScene</a> , <a href="#">simxLoadModel</a> , <a href="#">simxLoadUI</a> and <a href="#">simxTransferFile</a> .
C synopsis	<code>simxInt simxLoadScene(simxInt clientID,const simxChar* scenePathAndName,simxUChar options,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>scenePathAndName</b> : the scene filename, including the path and extension ("ttt"). The file is relative to the client or server system depending on the <b>options</b> value (see next argument) <b>options</b> : options, bit-coded: bit0 set: the specified file is located on the client side (in that case the function will be blocking since the scene first has to be transferred to the server). Otherwise it is located on the server side <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxLoadUI**regular API equivalent: [simLoadUI](#)ROS service equivalent: [simRosLoadUI](#)

Description	Loads previously saved custom user interfaces. See also <a href="#">simxLoadModel</a> , <a href="#">simxLoadScene</a> and <a href="#">simxTransferFile</a> .
C synopsis	<code>simxInt simxLoadUI(simxInt clientID,const simxChar* uiPathAndName,simxUChar options,simxInt* count,simxInt** uiHandles,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>uiPathAndName</b> : the ui filename, including the path and extension ("ttb"). The file is relative to the client or server system depending on the <b>options</b> value (see next argument) <b>options</b> : options, bit-coded: bit0 set: the specified file is located on the client side (in that case the function will be blocking since the UI file first has to be transferred to the server). Otherwise it is located on the server side <b>count</b> : a pointer to a value receiving the number of loaded custom user interfaces (one UI file might contain several custom user interfaces). <b>uiHandles</b> : a pointer to a pointer receiving the loaded UI handles. The user is in charge of releasing the buffer with <a href="#">simxReleaseBuffer</a> (*uiHandles) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxPauseCommunication**



Description	Allows to temporarily halt the communication thread from sending data. This can be useful if you need to send several values to V-REP that should be received and evaluated at the same time. This is a remote API helper function.
C synopsis	<code>simxInt simxPauseCommunication(simxInt clientID,simxUChar pause)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>pause</b> : whether the communication thread should pause or run normally.  Usage example:  <pre>simxPauseCommunication(clientID,1); simxSetJointPosition(clientID, joint1Handle, joint1Value, simx_opmode_oneshot); simxSetJointPosition(clientID, joint2Handle, joint2Value, simx_opmode_oneshot); simxSetJointPosition(clientID, joint3Handle, joint3Value, simx_opmode_oneshot); simxPauseCommunication(clientID, 0);  // Above's 3 joints will be received and set on the V-REP side at the same time</pre>
C return value	0 in case of operation success.
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxPauseSimulation**  
regular API equivalent: [simPauseSimulation](#)  
ROS service equivalent: [simRosPauseSimulation](#)

Description	Requests a pause of a simulation. See also <a href="#">simxStartSimulation</a> and <a href="#">simxStopSimulation</a> .
C synopsis	<code>simxInt simxPauseSimulation(simxInt clientID,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function is <code>simx_opmode_oneshot</code> .
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

### simxQuery

Description	Sends a query string to V-REP, and waits for a reply string. Query and reply strings can be accessed via string signals. This function allows for instance to have a <a href="#">child script</a> , another remote API client or a <a href="#">ROS node</a> handle special requests coming from this remote API client, then send a reply back.  Usage example where a child script handles a request:  <pre>// Following is the remote API client side: simxUChar* replyData; simxInt replySize; if (simxQuery(clientID,"request","send me a 42",12,"reply",&amp;replyData,&amp;replySize,5000)==0)     printf("The reply is: %s\n",replyData  -- This is the child script side. The child script is non-threaded and -- following part executed at each simulation pass: req=simGetStringSignal("request") if (req) then     simClearStringSignal("request")     if (req=="send me a 42") then         simSetStringSignal("reply","42\0") -- will be automatically cleared by the client     end end end</pre>
C synopsis	<code>simxInt simxQuery(simxInt clientID,const simxChar* signalName,const simxUChar* signalValue,simxInt signalLength,const simxChar* retSignalName,simxUChar** retSignalValue,simxInt* retSignalLength,simxInt timeOutInMs)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal that contains the request string <b>signalValue</b> : pointer to the request string. <b>signalLength</b> : the size of the request string, since it may contain any data (also embedded zeros). <b>retSignalName</b> : name of the signal that contains the reply string <b>retSignalValue</b> : pointer to a pointer receiving the value of the reply string. The string pointer will remain valid until next remote API call <b>retSignalLength</b> : pointer to a location receiving the value of the reply string length, since it may contain any data (also embedded zeros). <b>timeOutInMs</b> : the maximum time in milliseconds that the function will wait for a reply.
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxReadCollision**  
regular API equivalent: [simReadCollision](#)  
ROS service equivalent: [simRosReadCollision](#)

Description	Reads the collision state of a registered collision object. This function doesn't perform collision detection, it merely reads the result from a previous call to <a href="#">simHandleCollision</a> ( <code>simHandleCollision</code> is called in the default main script). See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxReadCollision(simxInt clientID,simxInt collisionObjectHandle,simxUChar* collisionState,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>collisionObjectHandle</b> : handle of the collision object <b>collisionState</b> : a pointer to a value receiving the collision state (0: not colliding) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxReadDistance**regular API equivalent: [simReadDistance](#)ROS service equivalent: [simRosReadDistance](#)

Description	Reads the distance that a registered distance object measured. This function doesn't perform minimum distance calculation, it merely reads the result from a previous call to <a href="#">simHandleDistance</a> ( <code>simHandleDistance</code> is called in the default main script). See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxReadDistance(simxInt clientID,simxInt distanceObjectHandle,simxFloat* minimumDistance,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>distanceObjectHandle</b> : handle of the distance object <b>minimumDistance</b> : a pointer to a value receiving the minimum distance <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxReadForceSensor**regular API equivalent: [simReadForceSensor](#)ROS service equivalent: [simRosReadForceSensor](#)

Description	Reads the force and torque applied to a force sensor (filtered values are read), and its current state ('unbroken' or 'broken'). See also <a href="#">simxBreakForceSensor</a> , <a href="#">simxGetJointForce</a> and <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxReadForceSensor(simxInt clientID,simxInt forceSensorHandle,simxUChar* state,simxFloat* forceVector,simxFloat* torqueVector,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>forceSensorHandle</b> : handle of the force sensor <b>state</b> : pointer to a byte value receiving the state of the force sensor. Can be NULL bit 0 set: force and torque data is available, otherwise it is not (yet) available (e.g. when not enough values are present for the filter) bit 1 set: force sensor is broken, otherwise it is still intact ('unbroken') <b>forceVector</b> : pointer to 3 float values receiving the force vector. Can be NULL <b>torqueVector</b> : pointer to 3 float values receiving the torque vector. Can be NULL <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxReadProximitySensor**regular API equivalent: [simReadProximitySensor](#)ROS service equivalent: [simRosReadProximitySensor](#)

Description	Reads the state of a proximity sensor. This function doesn't perform detection, it merely reads the result from a previous call to <a href="#">simHandleProximitySensor</a> ( <code>simHandleProximitySensor</code> is called in the default main script). See also <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxReadProximitySensor(simxInt clientID,simxInt sensorHandle,simxUChar* detectionState,simxFloat* detectedPoint,simxInt* detectedObjectHandle,simxFloat* detectedSurfaceNormalVector,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>sensorHandle</b> : handle of the proximity sensor <b>detectionState</b> : pointer to a value receiving the detection state (0=no detection). Can be NULL. <b>detectedPoint</b> : pointer to 3 values receiving the detected point coordinates (relative to the sensor reference frame). Can be NULL. <b>detectedObjectHandle</b> : pointer to a value receiving the handle of the detected object. Can be NULL. <b>detectedSurfaceNormalVector</b> : pointer to 3 values receiving the normal vector (normalized) of the detected surface. Relative to the sensor reference frame. Can be NULL <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxReadStringStream**

Description	Gets the value of a string signal, then clears it. Useful to retrieve continuous data from the server. See also <a href="#">simxWriteStringStream</a> .
C synopsis	<code>simxInt simxReadStringStream(simxInt clientID,const simxChar* signalName,simxUChar** signalValue,simxInt* signalLength,simxInt operationMode)</code>
C parameters	<p><b>clientID</b>: the client ID. refer to <a href="#">simxStart</a>.  <b>signalName</b>: name of the signal  <b>signalValue</b>: pointer to a pointer receiving the value of the signal. The signal value will remain valid until next remote API call  <b>signalLength</b>: pointer to a location receiving the value of the signal length, since it may contain any data (also embedded zeros).  <b>operationMode</b>: a <a href="#">remote API function operation mode</a>. Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls). <code>simx_opmode_oneshot_wait</code> is forbidden. Use a construction like following in order to continuously exchange data with V-REP:</p> <p>Remote API client side:</p> <pre>// Initialization phase: simxUChar* signal; simxInt sLength; simxReadStringStream(cid,"toClient",&amp;signal,&amp;sLength,simx_opmode_streaming);  // while we are connected: while (simxGetConnectionId(cid)!=-1) {     if (simxReadStringStream(cid,"toClient",&amp;signal,&amp;sLength,simx_opmode_buffer)==         simx_return_ok)     {         // Data produced by the child script was retrieved! Send it back to the child script:         simxWriteStringStream(cid,"fromClient",signal,sLength,simx_opmode_oneshot);     } }</pre> <p>Server side (V-REP), from a non-threaded child script:</p> <pre>if (sim_call_type==sim_childscriptcall_initialization) then     -- initialization phase:     i=0     lastReceived=-1 end  if (sim_call_type==sim_childscriptcall_actuation) then     -- First send a stream of integers that count up:     dat=simGetStringSignal('toClient')     if not dat then         dat=''     end     dat=dat..simPackInts({i})     i=i+1     simSetStringSignal('toClient',dat)      -- Here receive the integer stream in return and check if each number is correct:     dat=simGetStringSignal('fromClient')     if dat then         simClearStringSignal('fromClient')         dat=simUnpackInts(dat)         for j=1,#dat,1 do             if (dat[j]~=lastReceived+1) then                 print('Error')             else                 io.write('.')                 lastReceived=dat[j]             end         end     end end end</pre>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxReadVisionSensor**regular API equivalent: [simReadVisionSensor](#)ROS service equivalent: [simRosReadVisionSensor](#)

Description	Reads the state of a vision sensor. This function doesn't perform detection, it merely reads the result from a previous call to <a href="#">simHandleVisionSensor</a> ( <code>simHandleVisionSensor</code> is called in the default main script). See also <a href="#">simxGetVisionSensorImage</a> and <a href="#">simxGetObjectGroupData</a> .
C synopsis	<code>simxInt simxReadVisionSensor(simxInt clientID,simxInt sensorHandle,simxUChar* detectionState,simxFloat** auxValues,simxInt** auxValuesCount,simxInt operationMode)</code>
C parameters	<p><b>clientID</b>: the client ID. refer to <a href="#">simxStart</a>.  <b>sensorHandle</b>: handle of the vision sensor</p>

	<p><b>detectionState:</b> pointer to a byte that receives the detection state (i.e. the trigger state). Can be NULL</p> <p><b>auxValues:</b> auxiliary values returned from the <a href="#">applied filters</a>. By default V-REP returns one packet of 15 auxiliary values: the minimum of {intensity, red, green, blue, depth value}, the maximum of {intensity, red, green, blue, depth value}, and the average of {intensity, red, green, blue, depth value}. If additional filter components return values, then they will be appended as packets to the first packet. AuxValues can be NULL if auxValuesCount is also NULL. The user is in charge of releasing the auxValues buffer with <a href="#">simxReleaseBuffer</a>(*auxValues).</p> <p><b>auxValuesCount:</b> contains information about the number of auxiliary value packets and packet sizes returned in auxValues. The first value is the number of packets, the second is the size of packet1, the third is the size of packet2, etc. Can be NULL if auxValues is also NULL. The user is in charge of releasing the auxValuesCount buffer with <a href="#">simxReleaseBuffer</a>(*auxValuesCount).</p> <p><b>operationMode:</b> a <a href="#">remote API function operation mode</a>. Recommended operation modes for this function are <code>simx_opmode_streaming</code> (the first call) and <code>simx_opmode_buffer</code> (the following calls)</p>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

#### simxReleaseBuffer (regular API equivalent: [simReleaseBuffer](#))

Description	Releases a buffer previously created with <a href="#">simxCreateBuffer</a> or a buffer returned by a remote API function. This is a remote API helper function.
C synopsis	<code>simxVoid simxReleaseBuffer(simxUChar* buffer)</code>
C parameters	<b>buffer:</b> buffer to be released
C return value	none
Other languages	<a href="#">Python</a> , <a href="#">Matlab</a>

#### simxRemoveModel regular API equivalent: [simRemoveModel](#) ROS service equivalent: [simRosRemoveModel](#)

Description	Removes a model from the scene. See also <a href="#">simxRemoveObject</a> .
C synopsis	<code>simxInt simxRemoveModel(simxInt clientID, simxInt objectHandle, simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle:</b> handle of the model to remove (object should be flagged as model base). <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code> (or <code>simx_opmode_oneshot_wait</code> )
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

#### simxRemoveObject regular API equivalent: [simRemoveObject](#) ROS service equivalent: [simRosRemoveObject](#)

Description	Removes a scene object. See also <a href="#">simxRemoveModel</a> .
C synopsis	<code>simxInt simxRemoveObject(simxInt clientID, simxInt objectHandle, simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle:</b> handle of the object to remove <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code> (or <code>simx_opmode_oneshot_wait</code> )
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

#### simxRemoveUI regular API equivalent: [simRemoveUI](#) ROS service equivalent: [simRosRemoveUI](#)

Description	Removes a custom user interface.
C synopsis	<code>simxInt simxRemoveUI(simxInt clientID, simxInt uiHandle, simxInt operationMode)</code>
C parameters	<b>clientID:</b> the client ID. refer to <a href="#">simxStart</a> . <b>uiHandle:</b> handle of the custom user interface <b>operationMode:</b> a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code> (or <code>simx_opmode_oneshot_wait</code> )
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

#### simxSetArrayParameter regular API equivalent: [simSetArrayParameter](#) ROS service equivalent: [simRosSetArrayParameter](#)

Description	Sets 3 values of an <a href="#">array parameter</a> . See also <a href="#">simxGetArrayParameter</a> , <a href="#">simxSetBooleanParameter</a> , <a href="#">simxSetIntegerParameter</a> and <a href="#">simxSetFloatingParameter</a> .
C synopsis	<code>simxInt simxSetArrayParameter(simxInt clientID, simxInt paramIdentifier, const simxFloat*</code>

	paramValues,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>paramIdentifier</b> : an <a href="#">array parameter identifier</a> <b>paramValues</b> : the array containing the 3 values to set <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetBooleanParameter**regular API equivalent: [simSetBooleanParameter](#)ROS service equivalent: [simRosSetBooleanParameter](#)

Description	Sets a <a href="#">boolean parameter</a> . See also <a href="#">simxGetBooleanParameter</a> , <a href="#">simxSetIntegerParameter</a> , <a href="#">simxSetArrayParameter</a> and <a href="#">simxSetFloatingParameter</a> .
C synopsis	<code>simxInt simxSetBooleanParameter(simxInt clientID,simxInt paramIdentifier,simxUChar paramValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>paramIdentifier</b> : a <a href="#">Boolean parameter identifier</a> <b>paramValue</b> : the parameter value <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetFloatingParameter**regular API equivalent: [simSetFloatingParameter](#)ROS service equivalent: [simRosSetFloatingParameter](#)

Description	Sets a <a href="#">floating point parameter</a> . See also <a href="#">simxGetFloatingParameter</a> , <a href="#">simxSetBooleanParameter</a> , <a href="#">simxSetArrayParameter</a> and <a href="#">simxSetIntegerParameter</a> .
C synopsis	<code>simxInt simxSetFloatingParameter(simxInt clientID,simxInt paramIdentifier,simxFloat paramValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>paramIdentifier</b> : a <a href="#">floating parameter identifier</a> <b>paramValue</b> : the parameter value <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetFloatSignal**regular API equivalent: [simSetFloatSignal](#)ROS service equivalent: [simRosSetFloatSignal](#)

Description	Sets the value of a float signal. If that signal is not yet present, it is added. See also <a href="#">simxGetFloatSignal</a> , <a href="#">simxClearFloatSignal</a> , <a href="#">simxSetIntegerSignal</a> and <a href="#">simxSetStringSignal</a> .
C synopsis	<code>simxInt simxSetFloatSignal(simxInt clientID,const simxChar* signalName,simxFloat signalValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValue</b> : value of the signal <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetIntegerParameter**regular API equivalent: [simSetIntegerParameter](#)ROS service equivalent: [simRosSetIntegerParameter](#)

Description	Sets an <a href="#">integer parameter</a> . See also <a href="#">simxGetIntegerParameter</a> , <a href="#">simxSetBooleanParameter</a> , <a href="#">simxSetArrayParameter</a> and <a href="#">simxSetFloatingParameter</a> .
C synopsis	<code>simxInt simxSetIntegerParameter(simxInt clientID,simxInt paramIdentifier,simxInt paramValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>paramIdentifier</b> : an <a href="#">integer parameter identifier</a> <b>paramValue</b> : the parameter value <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetIntegerSignal**regular API equivalent: [simSetIntegerSignal](#)ROS service equivalent: [simRosSetIntegerSignal](#)

Description	Sets the value of an integer signal. If that signal is not yet present, it is added. See also <a href="#">simxGetIntegerSignal</a> , <a href="#">simxClearIntegerSignal</a> , <a href="#">simxSetFloatSignal</a> and <a href="#">simxSetStringSignal</a> .
C synopsis	<code>simxInt simxSetIntegerSignal(simxInt clientID,const simxChar* signalName,simxInt signalValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValue</b> : value of the signal <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetJointForce**regular API equivalent: [simSetJointForce](#)ROS service equivalent: [simRosSetJointForce](#)

Description	Sets the maximum force or torque that a joint can exert. This function has no effect when the joint is not dynamically enabled, or when it is a spherical joint. See also <a href="#">simxGetJointForce</a> .
C synopsis	<code>simxInt simxSetJointForce(simxInt clientID,simxInt jointHandle,simxFloat force,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>force</b> : the maximum force or torque that the joint can exert <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetJointPosition**regular API equivalent: [simSetJointPosition](#)ROS service equivalent: [simRosSetJointPosition](#)

Description	Sets the intrinsic position of a joint. May have no effect depending on the joint mode. This function cannot be used with spherical joints (use <a href="#">simxSetSphericalJointMatrix</a> instead). If you want to set several joints that should be applied at the exact same time on the V-REP side, then use <a href="#">simxPauseCommunication</a> . See also <a href="#">simxGetJointPosition</a> and <a href="#">simxSetJointTargetPosition</a> .
C synopsis	<code>simxInt simxSetJointPosition(simxInt clientID,simxInt jointHandle,simxFloat position,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>position</b> : position of the joint (angular or linear value depending on the joint type) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_oneshot</code> or <code>simx_opmode_streaming</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetJointTargetPosition**regular API equivalent: [simSetJointTargetPosition](#)ROS service equivalent: [simRosSetJointTargetPosition](#)

Description	Sets the target position of a joint if the joint is in torque/force mode (also make sure that the joint's motor and position control are enabled). See also <a href="#">simxSetJointPosition</a> .
C synopsis	<code>simxInt simxSetJointTargetPosition(simxInt clientID,simxInt jointHandle,simxFloat targetPosition,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>targetPosition</b> : target position of the joint (angular or linear value depending on the joint type) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_oneshot</code> or <code>simx_opmode_streaming</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetJointTargetVelocity**regular API equivalent: [simSetJointTargetVelocity](#)ROS service equivalent: [simRosSetJointTargetVelocity](#)

Description	Sets the intrinsic target velocity of a non-spherical joint. This command makes only sense when the joint
-------------	---

	mode is: (a) motion mode: the joint's motion handling feature must be enabled ( <a href="#">simHandleJoint</a> must be called (is called by default in the main script), and the joint motion properties must be set in the joint settings dialog), (b) torque/force mode: the dynamics functionality and the joint motor have to be enabled (position control should however be disabled)
C synopsis	<code>simxInt simxSetJointTargetVelocity(simxInt clientID,simxInt jointHandle,simxFloat targetVelocity,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>jointHandle</b> : handle of the joint <b>targetVelocity</b> : target velocity of the joint (linear or angular velocity depending on the joint-type) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_oneshot</code> or <code>simx_opmode_streaming</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetModelProperty**regular API equivalent: [simSetModelProperty](#)ROS service equivalent: [simRosSetModelProperty](#)

Description	Sets the properties of a model. See also <a href="#">simxGetModelProperty</a> .
C synopsis	<code>simxInt simxSetModelProperty(simxInt clientID,simxInt objectHandle,simxInt prop,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>prop</b> : a <a href="#">model property value</a> <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetObjectFloatParameter**regular API equivalent: [simSetObjectFloatParameter](#)ROS service equivalent: [simRosSetObjectFloatParameter](#)

Description	Sets a floating-point parameter of an object. See also <a href="#">simxGetObjectFloatParameter</a> and <a href="#">simxSetObjectIntParameter</a> .
C synopsis	<code>simxInt simxSetObjectFloatParameter(simxInt clientID,simxInt objectHandle,simxInt parameterID,simxFloat parameterValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>parameterID</b> : identifier of the parameter to set. See the <a href="#">list of all possible object parameter identifiers</a> <b>parameterValue</b> : the desired value of the parameter <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetObjectIntParameter**regular API equivalent: [simSetObjectIntParameter](#)ROS service equivalent: [simRosSetObjectIntParameter](#)

Description	Sets an integer parameter of an object. See also <a href="#">simxGetObjectIntParameter</a> and <a href="#">simxSetObjectFloatParameter</a> .
C synopsis	<code>simxInt simxSetObjectIntParameter(simxInt clientID,simxInt objectHandle,simxInt parameterID,simxInt parameterValue,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>parameterID</b> : identifier of the parameter to set. See the <a href="#">list of all possible object parameter identifiers</a> <b>parameterValue</b> : the desired value of the parameter <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetObjectOrientation**regular API equivalent: [simSetObjectOrientation](#)ROS service equivalent: [simRosSetObjectQuaternion](#)

Description	Sets the orientation ( <a href="#">Euler angles</a> ) of an object. Dynamically simulated objects will implicitly be reset before the command is applied (i.e. similar to calling <a href="#">simResetDynamicObject</a> just before). See also <a href="#">simxGetObjectOrientation</a> and <a href="#">simxSetObjectPosition</a> .
C synopsis	<code>simxInt simxSetObjectOrientation(simxInt clientID,simxInt objectHandle,simxInt relativeToObjectHandle,const simxFloat* eulerAngles,simxInt operationMode)</code>

C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>relativeToObjectHandle</b> : indicates relative to which reference frame the orientation is specified. Specify -1 to set the absolute orientation, <code>sim_handle_parent</code> to set the orientation relative to the object's parent, or an object handle relative to whose reference frame the orientation is specified. <b>eulerAngles</b> : Euler angles (alpha, beta and gamma) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetObjectParent**regular API equivalent: [simSetObjectParent](#)ROS service equivalent: [simRosSetObjectParent](#)

Description	Sets an object's parent object. See also <a href="#">simxGetObjectParent</a> .
C synopsis	<code>simxInt simxSetObjectParent(simxInt clientID, simxInt objectHandle, simxInt parentObject, simxUChar keepInPlace, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object that will become child of the parent object. Can be combined with <a href="#">sim_handleflag_assembly</a> , if the two objects can be assembled via a predefined assembly transformation (refer to the <b>assembling</b> option in the <a href="#">object common properties</a> ). In that case, <b>parentObject</b> can't be -1, and <b>keepInPlace</b> should be set to false. <b>parentObject</b> : handle of the object that will become parent, or -1 if the object should become parentless <b>keepInPlace</b> : indicates whether the object's absolute position and orientation should stay same <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code> or <code>simx_opmode_oneshot_wait</code> depending on the intent
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetObjectPosition**regular API equivalent: [simSetObjectPosition](#)ROS service equivalent: [simRosSetObjectPosition](#)

Description	Sets the position of an object. Dynamically simulated objects will implicitly be reset before the command is applied (i.e. similar to calling <a href="#">simResetDynamicObject</a> just before). See also <a href="#">simxGetObjectPosition</a> and <a href="#">simxSetObjectOrientation</a> .
C synopsis	<code>simxInt simxSetObjectPosition(simxInt clientID, simxInt objectHandle, simxInt relativeToObjectHandle, const simxFloat* position, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandle</b> : handle of the object <b>relativeToObjectHandle</b> : indicates relative to which reference frame the position is specified. Specify -1 to set the absolute position, <code>sim_handle_parent</code> to set the position relative to the object's parent, or an object handle relative to whose reference frame the position is specified. <b>position</b> : the position values (x, y and z) <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetObjectSelection (ROS service equivalent: [simRosSetObjectSelection](#))**

Description	Sets the selection state for objects. See also <a href="#">simxGetObjectSelection</a> .
C synopsis	<code>simxInt simxSetObjectSelection(simxInt clientID, const simxInt* objectHandles, simxInt objectCount, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>objectHandles</b> : an array of object handles <b>objectCount</b> : the number of elements in the array <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code> or <code>simx_opmode_oneshot_wait</code> depending on the intent.
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetSphericalJointMatrix**regular API equivalent: [simSetSphericalJointMatrix](#)ROS service equivalent: [simRosSetSphericalJointMatrix](#)

Description	Sets the intrinsic orientation matrix of a spherical joint object. This function cannot be used with non-spherical joints (use <a href="#">simxSetJointPosition</a> instead). See also <a href="#">simxGetJointMatrix</a> ..
C synopsis	<code>simxInt simxSetSphericalJointMatrix(simxInt clientID, simxInt jointHandle, simxFloat* matrix, simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> .



	<b>jointHandle</b> : handle of the joint <b>matrix</b> : pointer to 12 simxFloat values. See the regular API equivalent function for details <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function are <code>simx_opmode_oneshot</code> or <code>simx_opmode_streaming</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetStringSignal**  
regular API equivalent: [simSetStringSignal](#)  
ROS service equivalent: [simRosSetStringSignal](#)

Description	Sets the value of a string signal. If that signal is not yet present, it is added. See also <a href="#">simxAppendStringSignal</a> , <a href="#">simxGetStringSignal</a> , <a href="#">simxClearStringSignal</a> , <a href="#">simxSetIntegerSignal</a> and <a href="#">simxSetFloatSignal</a> .
C synopsis	<code>simxInt simxSetStringSignal(simxInt clientID,const simxChar* signalName,const simxUChar* signalValue,simxInt signalLength,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValue</b> : value of the signal (which may contain any value, including embedded zeros) <b>signalLength</b> : size of the signalValue string. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetUIButtonLabel**  
regular API equivalent: [simSetUIButtonLabel](#)  
ROS service equivalent: [simRosSetUIButtonLabel](#)

Description	Sets the up-state and down-state labels of a custom user interface button.
C synopsis	<code>simxInt simxSetUIButtonLabel(simxInt clientID,simxInt uiHandle,simxInt uiButtonID,const simxChar* upStateLabel,const simxChar* downStateLabel,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>uiHandle</b> : handle of the custom user interface <b>uiButtonID</b> : handle (or ID) of the custom user interface button <b>upStateLabel</b> : pointer to a string containing the label of the button when it is up. Cannot be NULL. <b>downStateLabel</b> : pointer to a string containing the label of the button when it is down. Cannot be NULL. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetUIButtonProperty**  
regular API equivalent: [simSetUIButtonProperty](#)  
ROS service equivalent: [simRosSetUIButtonProperty](#)

Description	Sets the properties of a custom user interface button. See also <a href="#">simxGetUIButtonProperty</a> .
C synopsis	<code>simxInt simxSetUIButtonProperty(simxInt clientID,simxInt uiHandle,simxInt uiButtonID,simxInt prop,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>uiHandle</b> : handle of the custom user interface <b>uiButtonID</b> : handle (or ID) of the custom user interface button <b>prop</b> : a <a href="#">button property value</a> . <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetUISlider**  
regular API equivalent: [simSetUISlider](#)  
ROS service equivalent: [simRosSetUISlider](#)

Description	Sets the slider position of a custom user interface button (must be a slider-type button). See also <a href="#">simxGetUISlider</a> .
C synopsis	<code>simxInt simxSetUISlider(simxInt clientID,simxInt uiHandle,simxInt uiButtonID,simxInt position,simxInt operationMode)</code>
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>uiHandle</b> : handle of the custom user interface <b>uiButtonID</b> : id of the button (slider) in the custom user interface <b>position</b> : slider position. valid values are between 0 and 1000 <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>

C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSetVisionSensorImage**regular API equivalent: [simSetVisionSensorImage](#)ROS service equivalent: [simRosSetVisionSensorImage](#)

Description	Sets the image of a vision sensor (and applies any image processing filter if specified in the vision sensor dialog). Make sure the vision sensor is flagged as <b>use external image</b> . The "regular" use of this function is to first read the data from a vision sensor with <a href="#">simxGetVisionSensorImage</a> , do some custom filtering, then write the modified image to a passive vision sensor. The alternate use of this function is to display textures, video images, etc. by using a vision sensor object (without however making use of the vision sensor functionality), since a vision sensor can be "looked through" like camera objects.
C synopsis	simxInt simxSetVisionSensorImage(simxInt clientID,simxInt sensorHandle,simxUChar* image,simxInt bufferSize,simxUChar options,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>sensorHandle</b> : handle of the vision sensor <b>image</b> : pointer to the image data <b>bufferSize</b> : size of the image data <b>options</b> : image options, bit-coded: bit0 set: each image pixel is a byte (greyscale image), otherwise each image pixel is a rgb byte-triplet <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxStart**

Description	Starts a communication thread with the server (i.e. V-REP). 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 <a href="#">simxFinish</a> . This is a remote API helper function.
C synopsis	simxInt simxStart(const simxChar* connectionAddress,simxInt connectionPort,simxUChar waitUntilConnected,simxUChar doNotReconnectOnceDisconnected,simxInt timeOutInMs,simxInt commThreadCycleInMs)
C parameters	<b>connectionAddress</b> : the ip address where the server is located (i.e. V-REP) <b>connectionPort</b> : the port number where to connect <b>waitUntilConnected</b> : if different from zero, then the function blocks until connected (or timed out). <b>doNotReconnectOnceDisconnected</b> : if different from zero, then the communication thread will not attempt a second connection if a connection was lost. <b>timeOutInMs</b> : connection time-out in milliseconds (for the first connection). <b>commThreadCycleInMs</b> : indicates how often data packets are sent back and forth. Reducing this number improves responsiveness, and a default value of 5 is recommended.
C return value	the client ID, or -1 if the connection to the server was not possible (i.e. a timeout was reached). A call to <a href="#">simxStart</a> should always be followed at the end with a call to <a href="#">simxFinish</a> if <a href="#">simxStart</a> didn't return -1
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxStartSimulation**regular API equivalent: [simStartSimulation](#)ROS service equivalent: [simRosStartSimulation](#)

Description	Requests a start of a simulation (or a resume of a paused simulation). This function is only executed by <a href="#">continuous remote API server services</a> . See also <a href="#">simxPauseSimulation</a> and <a href="#">simxStopSimulation</a> .
C synopsis	simxInt simxStartSimulation(simxInt clientID,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code> .
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxStopSimulation**regular API equivalent: [simStopSimulation](#)ROS service equivalent: [simRosStopSimulation](#)

Description	Requests a stop of the running simulation. See also <a href="#">simxStartSimulation</a> and <a href="#">simxPauseSimulation</a> .
C synopsis	simxInt simxStopSimulation(simxInt clientID,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation modes for this function is <code>simx_opmode_oneshot</code> .
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSynchronous (ROS service equivalent: simRosSynchronous)**

Description	Enables or disables the synchronous operation mode for the remote API server service that the client is connected to. The function is blocking. While in synchronous operation 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. See also <a href="#">simxSynchronousTrigger</a> and the <a href="#">remote API overview</a> . This is a remote API helper function.
C synopsis	simxInt simxSynchronous(simxInt clientID,simxUChar enable)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>enable</b> : the enable state of the synchronous operation
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxSynchronousTrigger (ROS service equivalent: simRosSynchronousTrigger)**

Description	Sends a synchronization trigger signal to the server. The function is blocking. The server needs to be previously enabled for synchronous operation via the <a href="#">simxSynchronous</a> function. The trigger signal will inform V-REP to execute the next simulation step (i.e. to call <a href="#">simHandleMainScript</a> ). While in synchronous operation mode, the client application is in charge of triggering the next simulation step, otherwise simulation will stall. See also the <a href="#">remote API overview</a> . This is a remote API helper function.
C synopsis	simxInt simxSynchronousTrigger(simxInt clientID)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> .
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxTransferFile (ROS service equivalent: simRosTransferFile)**

Description	Allows transferring a file from the client to the server. This function is used by several other functions internally (e.g. <a href="#">simxLoadModel</a> ). See also <a href="#">simxEraseFile</a> . This is a remote API helper function.
C synopsis	simxInt simxTransferFile(simxInt clientID,const simxChar* filePathAndName,const simxChar* fileName_serverSide,simxInt timeout,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>filePathAndName</b> : the local file name and path (i.e. on the client side) <b>fileName_serverSide</b> : a file name under which the transferred file will be saved on the server side. For now, do not specify a path (the file will be saved in the remote API plugin directory) <b>timeout</b> : a timeout value in milliseconds <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot_wait</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>

**simxWriteStringStream (ROS service equivalent: simRosAppendStringSignal)**

Description	Appends a string to a string signal. If that signal is not yet present, it is added. See also <a href="#">simxReadStringStream</a> .
C synopsis	simxInt simxWriteStringStream(simxInt clientID,const simxChar* signalName,const simxUChar* signalValueToAppend,simxInt signalLength,simxInt operationMode)
C parameters	<b>clientID</b> : the client ID. refer to <a href="#">simxStart</a> . <b>signalName</b> : name of the signal <b>signalValueToAppend</b> : value to append to the signal. That value may contain any value, including embedded zeros. <b>signalLength</b> : size of the signalValueToAppend string. <b>operationMode</b> : a <a href="#">remote API function operation mode</a> . Recommended operation mode for this function is <code>simx_opmode_oneshot</code>
C return value	a <a href="#">remote API function return code</a>
Other languages	<a href="#">Python</a> , <a href="#">Java</a> , <a href="#">Matlab</a> , <a href="#">Octave</a> , <a href="#">Urbi</a>