Remote API Functions (Matlab)

simxAddStatusbarMessage (regular API equivalent: simAddStatusbarMessage)

Description	Adds a message to the status bar.
Matlab synopsis	[number returnCode]=simxAddStatusbarMessage(number clientID,string message,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **message**: the message to display **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxAppendStringSignal

Description	DEPRECATED. Refer to simxWriteStringStream instead. Appends a string to a string signal. If that signal is not yet present, it is added. To pack/unpack integers/floats into/from a string, refer to simxPackInts, simxPackFloats, simxUnpackInts and simxUnpackFloats. See also simxSetStringSignal.
Matlab synopsis	[number returnCode]=simxAppendStringSignal(number clientID,string signalName,string signalValueToAppend,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **signalValueToAppend**: value to append to the signal. That value may contain any value, including embedded zeros. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi

simxAuxiliaryConsoleClose (regular API equivalent: simAuxiliaryConsoleClose)

Description	Closes an auxiliary console window. See also simxAuxiliaryConsoleOpen.
Matlab synopsis	[number returnCode]=simxAuxiliaryConsoleClose(number clientID,number consoleHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **consoleHandle**: the handle of the console window, previously returned by the simxAuxiliaryConsoleOpen command **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxAuxiliaryConsoleOpen (regular API equivalent: simAuxiliaryConsoleOpen)

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 simxAuxiliaryConsolePrint, simxAuxiliaryConsoleShow and simxAuxiliaryConsoleClose.
Matlab synopsis	[number returnCode,number consoleHandle]=simxAuxiliaryConsoleOpen(number clientID,string title,number maxLines,number mode,array position,array size,array textColor,array backgroundColor,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **title**: the title of the console window **maxLines**: the number of text lines that can be displayed and buffered

	mode: 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. **position**: the initial position of the console window (x and y value). Can be [] for default values **size**: the initial size of the console window (x and y value). Can be [] for default values **textColor**: the color of the text (rgb values, 0-1). Can be [] for default values **backgroundColor**: the background color of the console window (rgb values, 0-1). Can be [] for default values **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **consoleHandle**: the handle of the created console
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxAuxiliaryConsolePrint (regular API equivalent: simAuxiliaryConsolePrint)

Description	Prints to an auxiliary console window. See also simxAuxiliaryConsoleOpen.
Matlab synopsis	[number returnCode]=simxAuxiliaryConsolePrint(number clientID,number consoleHandle,string txt,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **consoleHandle**: the handle of the console window, previously returned by the simxAuxiliaryConsoleOpen command **txt**: the text to append, or [] to clear the console window **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxAuxiliaryConsoleShow (regular API equivalent: simAuxiliaryConsoleShow)

Description	Shows or hides an auxiliary console window. See also simxAuxiliaryConsoleOpen and simxAuxiliaryConsoleClose.
Matlab synopsis	[number returnCode]=simxAuxiliaryConsoleShow(number clientID,number consoleHandle,boolean showState,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **consoleHandle**: the handle of the console window, previously returned by the simxAuxiliaryConsoleOpen command **showState**: indicates whether the console should be hidden (false) or shown (true) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxBreakForceSensor (regular API equivalent: simBreakForceSensor)

Description	Allows breaking a force sensor during simulation. A broken force sensor will lose its positional and orientational constraints. See also simxReadForceSensor.
Matlab synopsis	[number returnCode]=simxBreakForceSensor(number clientID,number forceSensorHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **forceSensorHandle**: handle of the force sensor **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxCallScriptFunction (regular API equivalent: simCallScriptFunctionEx)

Description	Remotely calls a V-REP script function. When calling simulation scripts, then simulation must be running (and threaded scripts must still be running, i.e. not ended yet). Refer to this section for additional

	details.
Matlab synopsis	[number returnCode,array outInts,array outFloats,string outStrings,array outBuffer]=simxCallScriptFunction(number clientID,string scriptDescription,number scriptHandleOrType,string functionName,array inInts,array inFloats,string inStrings,array inBuffer,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **scriptDescription**: the name of the scene object where the script is attached to, or an empty string if the script has no associated scene object. **scriptHandleOrType**: the handle of the script, otherwise the type of the script: *sim_scripttype_mainscript* (0): the main script will be called. *sim_scripttype_childscript* (1): a child script will be called. *sim_scripttype_jointctrlcallback* (4): a joint control callback script will be called. *sim_scripttype_contactcallback* (5): the contact callback script will be called. *sim_scripttype_customizationscript* (6): a customization script will be called. *sim_scripttype_generalcallback* (7): the general callback script will be called. **functionName**: the name of the Lua function to call in the specified script. **inInts**: the input integer values that are handed over to the script function. Can be []. **inFloats**: the input floating-point values that are handed over to the script function. Can be []. **inStrings**: the input strings that are handed over to the script function. Each string should be terminated with a zero char, e.g. ['Hello' 0 'world!' 0]. Can be an empty string. **inBuffer**: the input buffer that is handed over to the script function. Can be []. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **outInts**: the returned integer values. **outFloats**: the returned floating-point values. **outStrings**: the returned strings. Each string is terminated with a zero char. **outBuffer**: the returned buffer.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxClearFloatSignal (regular API equivalent: simClearFloatSignal)

Description	Clears a float signal (removes it). See also simxSetFloatSignal, simxClearIntegerSignal and simxClearStringSignal.
Matlab synopsis	[number returnCode]=simxClearFloatSignal(number clientID,string signalName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal or an empty string to clear all float signals **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxClearIntegerSignal (regular API equivalent: simClearIntegerSignal)

Description	Clears an integer signal (removes it). See also simxSetIntegerSignal, simxClearFloatSignal and simxClearStringSignal.
Matlab synopsis	[number returnCode]=simxClearIntegerSignal(number clientID,string signalName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal or an empty string to clear all integer signals **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxClearStringSignal (regular API equivalent: simClearStringSignal)

Description	Clears a string signal (removes it). See also simxSetStringSignal, simxClearIntegerSignal and simxClearFloatSignal.
Matlab synopsis	[number returnCode]=simxClearStringSignal(number clientID,string signalName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal or an empty string to clear all string signals **operationMode**: a remote API function operation mode. Recommended operation mode for this

	function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxCloseScene (regular API equivalent: simCloseScene)

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 continuous remote API server services. See also simxLoadScene.
Matlab synopsis	[number returnCode]=simxCloseScene(number clientID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxCopyPasteObjects (regular API equivalent: simCopyPasteObjects)

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.
Matlab synopsis	[number returnCode,array newObjectHandles]=simxCopyPasteObjects(number clientID,array objectHandles,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandles**: an array containing the handles of the objects to copy **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **newObjectHandles**: an array of handles of newly created objects. Individual objects of a new model are not returned, but only the model base.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxCreateBuffer (regular API equivalent: simCreateBuffer)

Description	Creates a buffer. The buffer needs to be released with simxReleaseBuffer except otherwise explicitly specified. This is a remote API helper function.
Matlab synopsis	[libpointer buffer]=simxCreateBuffer(number bufferSize)
Matlab parameters	**bufferSize**: size of the buffer in bytes
Matlab return values	**buffer**: a pointer to the created buffer
Other languages	C/C++, Python

simxCreateDummy (regular API equivalent: simCreateDummy)

Description	Creates a dummy in the scene.
Matlab synopsis	[number returnCode,number dummyHandle]=simxCreateDummy(number clientID,number size,array colors,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **size**: the size of the dummy. **colors**: 4*3 bytes (0-255) for ambient_diffuse RGB, 3 reserved values (set to zero), specular RGB and emissive RGB. Can be [] for default colors. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return value	**returnCode**: a remote API function return code **dummyHandle**: handle of the created dummy.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxDisplayDialog (regular API equivalent: simDisplayDialog)

Description	Displays a generic dialog box during simulation (and only during simulation!). Use in conjunction with simxGetDialogResult, simxGetDialogInput and simxEndDialog. Use custom user interfaces instead if a

	higher customization level is required.
Matlab synopsis	[number returnCode,number dialogHandle,number uiHandle]=simxDisplayDialog(number clientID,string titleText,string mainText,number dialogType,string initialText,array titleColors,array dialogColors,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **titleText**: Title bar text **mainText**: Information text **dialogType**: a generic dialog style **initialText**: Initial text in the edit box if the dialog is of type sim_dlgstyle_input. **titleColors**: Title bar color (6 number values for RGB for background and foreground), can be [] for default colors **dialogColors**: Dialog color (6 number values for RGB for background and foreground), can be [] for default colors **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **dialogHandle**: handle of the generic dialog (different from an OpenGl-based custom UI handle!! (see hereafter)). This handle should be used with the following functions: simxGetDialogResult, simxGetDialogInput and simxEndDialog. **uiHandle**: the handle of the corresponding OpenGl-based custom UI.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxEndDialog (regular API equivalent: simEndDialog)

Description	Closes and releases resource from a previous call to simxDisplayDialog. 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).
Matlab synopsis	[number returnCode]=simxEndDialog(number clientID,number dialogHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **dialogHandle**: handle of generic dialog (return value of simxDisplayDialog) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxEraseFile

Description	Erases a file on the server side. This function is used by several other functions internally (e.g. simxLoadModel). See also simxTransferFile. This is a remote API helper function.
Matlab synopsis	[number returnCode]=simxEraseFile(number clientID,string fileName_serverSide,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **fileName_serverSide**: 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) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxFinish

Description	Ends the communication thread. This should be the very last remote API function called on the client side. simxFinish should only be called after a successfull call to simxStart. This is a remote API helper function.
Matlab synopsis	simxFinish(number clientID)
Matlab parameters	**clientID**: the client ID. refer to simxStart. Can be -1 to end all running communication threads.
Matlab return values	none
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetAndClearStringSignal

Description	DEPRECATED. Refer to simxReadStringStream instead. Gets the value of a string signal, then clears it. Useful to retrieve continuous data from the server. To pack/unpack integers/floats into/from a string, refer to simxPackInts, simxPackFloats, simxUnpackInts and simxUnpackFloats. See also simxGetStringSignal.
Matlab synopsis	[number returnCode,string signalValue]=simxGetAndClearStringSignal(number clientID,string signalName,number operationMode)

Matlab parameters	**clientID**: the client ID. refer to simxStart.
	signalName: name of the signal
	operationMode: a remote API function operation mode. 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:
	
	`% Initialization phase:`
	`[err,signal]=vrep.simxGetAndClearStringSignal(clientID,'sig',`
	`vrep.simx_opmode_streaming);`
	`% while we are connected:`
	`while (vrep.simxGetConnectionId(clientID)~=-1)`
	`[err,signal]=vrep.simxGetAndClearStringSignal(clientID,'sig',`
	`vrep.simx_opmode_buffer);`
	`if (err==vrep.simx_return_ok)`
	`% A signal was retrieved.`
	`% Enable streaming again (was automatically disabled with the positive event):`
	`[err,signal]=vrep.simxGetAndClearStringSignal(clientID,'sig',`
	`vrep.simx_opmode_streaming);`
	`end`
	
	`..`
	
	`end`

Matlab return values	**returnCode**: a remote API function return code **signalValue**: the signal data (that may contain any value, including embedded zeros).
Other languages	C/C++, Python, Java, Octave, Urbi

simxGetArrayParameter (regular API equivalent: simGetArrayParameter)

Description	Retrieves 3 values from an array. See the array parameter identifiers. See also simxSetArrayParameter, simxGetBooleanParameter, simxGetIntegerParameter, simxGetFloatingParameter and simxGetStringParameter.
Matlab synopsis	[number returnCode,array paramValues]=simxGetArrayParameter(number clientID,number paramIdentifier,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: an array parameter identifier **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking (if not called on a regular basis)
Matlab return values	**returnCode**: a remote API function return code **paramValues**: 3 values representing the array parameter
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetBooleanParameter (regular API equivalent: simGetBoolParameter)

Description	Retrieves a boolean value. See the Boolean parameter identifiers. See also simxSetBooleanParameter, simxGetIntegerParameter, simxGetFloatingParameter, simxGetArrayParameter and simxGetStringParameter.
Matlab synopsis	[number returnCode,boolean paramValue]=simxGetBooleanParameter(number clientID,number paramIdentifier,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: a Boolean parameter identifier **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking (if not called on a regular basis)
Matlab return values	**returnCode**: a remote API function return code **paramValue**: the parameter value
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetCollectionHandle (regular API equivalent: simGetCollectionHandle)

Description	Retrieves a collection handle based on its name. If the client application is launched from a child script,

	then you could also let the child script figure out what handle correspond to what collection, and send the handles as additional arguments to the client application during its launch. See also simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number handle]=simxGetCollectionHandle(number clientID,string collectionName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **collectionName**: name of the collection. If possibe, don't rely on the automatic name adjustment mechanism, and always specify the full collection name, including the #: if the collection is 'myCollection', specify 'myCollection#', if the collection is 'myCollection#0', specify 'myCollection#0', etc. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **handle**: the handle
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetCollisionHandle (regular API equivalent: simGetCollisionHandle)

Description	Retrieves a collision object handle based on its name. If the client application is launched from a child script, 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 simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number handle]=simxGetCollisionHandle(number clientID,string collisionObjectName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **collisionObjectName**: name of the collision object. If possibe, don't rely on the automatic name adjustment mechanism, 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. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **handle**: the handle
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetConnectionId

Description	Returns the ID of the current connection. Use this function to track the connection state to the server. See also simxStart. This is a remote API helper function.
Matlab synopsis	[number connectionId]=simxGetConnectionId(number clientID)
Matlab parameters	**clientID**: the client ID. refer to simxStart.
Matlab return values	**connectionId**: a connection ID, or -1 if the client is not connected to the server. Different connection IDs indicate temporary disconections in-between.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetDialogInput (regular API equivalent: simGetDialogInput)

Description	Queries the text the user entered into a generic dialog box of style sim_dlgstyle_input. To be used after simxDisplayDialog was called and after simxGetDialogResult returned sim_dlgret_ok.
Matlab synopsis	[number returnCode,string inputText]=simxGetDialogInput(number clientID,number dialogHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **dialogHandle**: handle of generic dialog (return value of simxDisplayDialog) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **inputText**: the string the user entered.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetDialogResult (regular API equivalent: simGetDialogResult)

Description	Queries the result of a dialog box. To be used after simxDisplayDialog was called.
Matlab synopsis	[number returnCode,number result]=simxGetDialogResult(number clientID,number dialogHandle,number operationMode)

Matlab parameters	**clientID**: the client ID. refer to simxStart. **dialogHandle**: handle of generic dialog (return value of simxDisplayDialog) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code **result**: the result value. Note. If the return value is sim_dlgret_still_open, the dialog was not closed and no button was pressed. Otherwise, you should free resources with simxEndDialog (the dialog might not be visible anymore, but is still present)
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetDistanceHandle (regular API equivalent: simGetDistanceHandle)

Description	Retrieves a distance object handle based on its name. If the client application is launched from a child script, 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 simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number handle]=simxGetDistanceHandle(number clientID,string distanceObjectName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **distanceObjectName**: name of the distance object. If possibe, don't rely on the automatic name adjustment mechanism, 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. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **handle**: the handle
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetFloatingParameter (regular API equivalent: simGetFloatParameter)

Description	Retrieves a floating point value. See the floating-point parameter identifiers. See also simxSetFloatingParameter, simxGetBooleanParameter, simxGetIntegerParameter, simxGetArrayParameter and simxGetStringParameter.
Matlab synopsis	[number returnCode,number paramValue]=simxGetFloatingParameter(number clientID,number paramIdentifier,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: a floating parameter identifier **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking (if not called on a regular basis)
Matlab return values	**returnCode**: a remote API function return code **paramValue**: the parameter value
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetFloatSignal (regular API equivalent: simGetFloatSignal)

Description	Gets the value of a float signal. Signals are cleared at simulation start. See also simxSetFloatSignal, simxClearFloatSignal, simxGetIntegerSignal and simxGetStringSignal.
Matlab synopsis	[number returnCode,number signalValue]=simxGetFloatSignal(number clientID,string signalName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **signalValue**: the value of the signal
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetInMessageInfo

Description	Retrieves information about the last received message from the server. This is a remote API helper function. See also simxGetOutMessageInfo.

	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.
Matlab synopsis	[number result,number info]=simxGetInMessageInfo(number clientID,number infoType)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **infoType**: an inbox message info type
Matlab return values	**result**: -1 in case of an error **info**: the requested information
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetIntegerParameter (regular API equivalent: simGetInt32Parameter)

Description	Retrieves an integer value. See the integer parameter identifiers. See also simxSetIntegerParameter, simxGetBooleanParameter, simxGetFloatingParameter, simxGetArrayParameter and simxGetStringParameter.
Matlab synopsis	[number returnCode,number paramValue]=simxGetIntegerParameter(number clientID,number paramIdentifier,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: an integer parameter identifier **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking (if not called on a regular basis)
Matlab return values	**returnCode**: a remote API function return code **paramValue**: the parameter value
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetIntegerSignal (regular API equivalent: simGetIntegerSignal)

Description	Gets the value of an integer signal. Signals are cleared at simulation start. See also simxSetIntegerSignal, simxClearIntegerSignal, simxGetFloatSignal and simxGetStringSignal.
Matlab synopsis	[number returnCode,number signalValue]=simxGetIntegerSignal(number clientID,string signalName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **signalValue**: the value of the signal
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetJointForce (regular API equivalent: simGetJointForce)

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 Bullet engine, this function returns the force or torque applied to the joint motor (torques from joint limits are not taken into account). With the ODE or Vortex engine, this function returns the total force or torque applied to a joint along/about its z-axis. See also simxSetJointForce, simxReadForceSensor and simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number force]=simxGetJointForce(number clientID,number jointHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **force**: the force or the torque applied to the joint along/about its z-axis
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetJointMatrix (regular API equivalent: simGetJointMatrix)

Description	Retrieves the intrinsic transformation matrix of a joint (the transformation caused by the joint movement). See also simxSetSphericalJointMatrix.
Matlab synopsis	[number returnCode,array matrix]=simxGetJointMatrix(number clientID,number jointHandle,number operationMode)

Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **matrix**: 12 number values. See the regular API equivalent function for details
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetJointPosition (regular API equivalent: simGetJointPosition)

Description	Retrieves the intrinsic position of a joint. This function cannot be used with spherical joints (use simxGetJointMatrix instead). See also simxSetJointPosition and simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number position]=simxGetJointPosition(number clientID,number jointHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **position**: 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.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

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:
	
	`[err,res,img]=vrep.simxGetVisionSensorImage(clientID,handle,0,vrep.simx_opmode_buffer);`
	`if (err==vrep.simx_return_ok)`
	`  imageAcquisitionTime=vrep.simxGetLastCmdTime(clientID);`
	`end`
	
	If some streaming commands are running, simxGetLastCmdTime will always retrieve the current simulation time, otherwise, only the simulation time of the last command that retrieved data from V-REP. This is a remote API helper function.

Matlab synopsis	[number cmdTime]=simxGetLastCmdTime(number clientID)
Matlab parameters	**clientID**: the client ID. refer to simxStart.
Matlab return values	**cmdTime**: the simulation time in milliseconds when the command reply was generated, or 0 if simulation was not running.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetLastErrors (regular API equivalent: simGetLastError)

Description	Retrieves the last 50 errors that occured on the server side, and clears the error buffer there. Only errors that occured because of this client will be reported.
Matlab synopsis	[number returnCode,cell errorStrings]=simxGetLastErrors(number clientID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls) when not debugging. For debugging purposes, use simx_opmode_blocking.
Matlab return values	**returnCode**: a remote API function return code **errorStrings**: all error strings
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetModelProperty (regular API equivalent: simGetModelProperty)

Description	Retrieves the properties of a model. See also simxSetModelProperty.
Matlab synopsis	[number returnCode,number prop]=simxGetModelProperty(number clientID,number objectHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart.

	objectHandle: handle of the object **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls), or simx_opmode_blocking (depending on the intended usage)
Matlab return values	**returnCode**: a remote API function return code **prop**: the model property value
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectChild (regular API equivalent: simGetObjectChild)

Description	Retrieves the handle of an object's child object. See also simxGetObjectParent.
Matlab synopsis	[number returnCode,number childObjectHandle]=simxGetObjectChild(number clientID,number parentObjectHandle,number childIndex,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **parentObjectHandle**: handle of the object **childIndex**: 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 **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **childObjectHandle**: the handle of the child object. If the value is -1, there is no child at the given index
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectFloatParameter (regular API equivalent: simGetObjectFloatParameter)

Description	Retrieves a floating-point parameter of a object. See also simxSetObjectFloatParameter and simxGetObjectIntParameter.
Matlab synopsis	[number returnCode,number parameterValue]=simxGetObjectFloatParameter(number clientID,number objectHandle,number parameterID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **parameterID**: identifier of the parameter to retrieve. See the list of all possible object parameter identifiers **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls), or simx_opmode_blocking (depending on the intended usage)
Matlab return values	**returnCode**: a remote API function return code **parameterValue**: the value of the parameter
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectGroupData

Description	Simultaneously retrieves data of various objects in a V-REP scene.
Matlab synopsis	[number returnCode,array handles,array intData,array floatData,array stringData]=simxGetObjectGroupData(number clientID,number objectType,number dataType,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectType**: a scene object type, sim_appobj_object_type for all scene objects, or a collection handle. **dataType**: the type of data that is desired: 0: retrieves the object names (in stringData.) 1: retrieves the object types (in intData) 2: retrieves the parent object handles (in intData) 3: retrieves the absolute object positions (in floatData. There are 3 values for each object (x,y,z)) 4: retrieves the local object positions (in floatData. There are 3 values for each object (x,y,z)) 5: retrieves the absolute object orientations as Euler angles (in floatData. There are 3 values for each object (alpha,beta,gamma)) 6: retrieves the local object orientations as Euler angles (in floatData. There are 3 values for each object (alpha,beta,gamma)) 7: retrieves the absolute object orientations as quaternions (in floatData. There are 4 values for each object (qx,qy,qz,qw)) 8: retrieves the local object orientations as quaternions (in floatData. There are 4 values for each object (qx,qy,qz,qw)) 9: retrieves the absolute object positions and orientations (as Euler angles) (in floatData.

	There are 6 values for each object (x,y,z,alpha,beta,gamma)) 10: retrieves the local object positions and orientations (as Euler angles) (in floatData. There are 6 values for each object (x,y,z,alpha,beta,gamma)) 11: retrieves the absolute object positions and orientations (as quaternions) (in floatData. There are 7 values for each object (x,y,z,qx,qy,qz,qw)) 12: retrieves the local object positions and orientations (as quaternions) (in floatData. There are 7 values for each object (x,y,z,qx,qy,qz,qw)) 13: retrieves proximity sensor data (in intData (2 values): detection state, detected object handle. In floatData (6 values): detected point (x,y,z) and detected surface normal (nx,ny,nz)) 14: retrieves force sensor data (in intData (1 values): force sensor state. In floatData (6 values): force (fx,fy,fz) and torque (tx,ty,tz)) 15: retrieves joint state data (in floatData (2 values): position, force/torque) 16: retrieves joint properties data (in intData (2 values): joint type, joint mode (bit16=hybid operation). In floatData (2 values): joint limit low, joint range (-1.0 if joint is cyclic)) 17: retrieves the object linear velocity (in floatData. There are 3 values for each object (vx,vy,vz)) 18: retrieves the object angular velocity as Euler angles per seconds (in floatData. There are 3 values for each object (dAlpha,dBeta,dGamma)) 19: retrieves the object linear and angular velocity (in floatData. There are 6 values for each object (vx,vy,vz,dAlpha,dBeta,dGamma)) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking or simx_opmode_streaming.
Matlab return values	**returnCode**: a remote API function return code **handles**: the object handles. **intData**: the integer values. **floatData**: the float values. **stringData**: the string values.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectHandle (regular API equivalent: simGetObjectHandle)

Description	Retrieves an object handle based on its name. If the client application is launched from a child script, 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 simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number handle]=simxGetObjectHandle(number clientID,string objectName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectName**: name of the object. If possibe, don't rely on the automatic name adjustment mechanism, 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. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **handle**: the handle
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectIntParameter (regular API equivalent: simGetObjectInt32Parameter)

Description	Retrieves an integer parameter of a object. See also simxSetObjectIntParameter and simxGetObjectFloatParameter.
Matlab synopsis	[number returnCode,number parameterValue]=simxGetObjectIntParameter(number clientID,number objectHandle,number parameterID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **parameterID**: identifier of the parameter to retrieve. See the list of all possible object parameter identifiers **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls), or simx_opmode_blocking (depending on the intended usage)
Matlab return values	**returnCode**: a remote API function return code **parameterValue**: the value of the parameter
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectOrientation (regular API equivalent: simGetObjectOrientation)

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

simxGetObjectParent (regular API equivalent: simGetObjectParent)

Description	Retrieves the handle of an object's parent object. See also simxGetObjectChild and simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number parentObjectHandle]=simxGetObjectParent(number clientID,number objectHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **parentObjectHandle**: the handle of the parent object. If the value is -1, the object has no parent
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectPosition (regular API equivalent: simGetObjectPosition)

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

simxGetObjects (regular API equivalent: simGetObjects)

Description	Retrieves object handles of a given type, or of all types (i.e. all object handles). See also simxGetObjectGroupData.
Matlab synopsis	[number returnCode,array objectHandles]=simxGetObjects(number clientID,number objectType,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectType**: object type (sim_object_shape_type, sim_object_joint_type, etc., or sim_handle_all for any type of object **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **objectHandles**: an array containing object handles.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectSelection (regular API equivalent: simGetObjectSelection)

Description	Retrieves all selected object's handles. See also simxSetObjectSelection.

Matlab synopsis	[number returnCode,array objectHandles]=simxGetObjectSelection(number clientID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls), or simx_opmode_blocking depending on the intent.
Matlab return values	**returnCode**: a remote API function return code **objectHandles**: an array containing the handles of all selected objects
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetObjectVelocity (regular API equivalent: simGetObjectVelocity)

Description	Retrieves the linear and angular velocity of an object. See also simxGetObjectPosition, simxGetObjectOrientation and simxGetObjectGroupData.
Matlab synopsis	[number returnCode,array linearVelocity,array angularVelocity]=simxGetObjectVelocity(number clientID,number objectHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **linearVelocity**: 3 values representing the linear velocity (vx, vy, vz). **angularVelocity**: 3 values representing the angular velocity (dAlpha, dBeta, dGamma).
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetOutMessageInfo

Description	Retrieves information about the next message to send to the server. This is a remote API helper function. See also simxGetInMessageInfo.
Matlab synopsis	[number result,number info]=simxGetOutMessageInfo(number clientID,number infoType)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **infoType**: an outbox message info type
Matlab return values	**result**: -1 in case of an error **info**: the requested information
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

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.
Matlab synopsis	[number returnCode,number pingTime]=simxGetPingTime(number clientID)
Matlab parameters	**clientID**: the client ID. refer to simxStart.
Matlab return values	**returnCode**: a remote API function return code **pingTime**: the ping time in milliseconds.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetStringParameter (regular API equivalent: simGetStringParameter)

Description	Retrieves a string value. See the string parameter identifiers. See also simxGetBooleanParameter, simxGetIntegerParameter, simxGetArrayParameter and simxGetFloatingParameter.
Matlab synopsis	[number returnCode,string paramValue]=simxGetStringParameter(number clientID,number paramIdentifier,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: a string parameter identifier **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking (if not called on a regular basis)
Matlab return values	**returnCode**: a remote API function return code **paramValue**: the parameter value (a string)
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetStringSignal (regular API equivalent: simGetStringSignal)

Description	Gets the value of a string signal. Signals are cleared at simulation start. To pack/unpack integers/floats into/from a string, refer to simxPackInts, simxPackFloats, simxUnpackInts and simxUnpackFloats. See also simxSetStringSignal, simxReadStringStream, simxClearStringSignal, simxGetIntegerSignal and simxGetFloatSignal.
Matlab synopsis	[number returnCode,string signalValue]=simxGetStringSignal(number clientID,string signalName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **signalValue**: the signal data (that may contain any value, including embedded zeros).
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetUIButtonProperty (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxGetUIEventButton (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxGetUIHandle (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxGetUISlider (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxGetVisionSensorDepthBuffer (regular API equivalent: simGetVisionSensorDepthBuffer)

Description	Retrieves the depth buffer of a vision sensor as a pointer. The returned data doesn't make sense if simHandleVisionSensor wasn't called previously (simHandleVisionSensor is called by default in the main script if the vision sensor is not tagged as explicit handling). Use the simxGetLastCmdTime function to verify the *freshness* of the retrieved data. See also simxGetVisionSensorDepthBuffer2 and simxGetVisionSensorImage.
Matlab synopsis	[number returnCode,array resolution,libpointer buffer]=simxGetVisionSensorDepthBuffer(number clientID,number sensorHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the vision sensor **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)

Matlab return values	**returnCode**: a remote API function return code
	resolution: 2 number values representing the resolution of the image
	buffer: a libpointer object to the data. To access individual depth buffer pixels, use following code:
	`buffer.setDataType('singlePtr',1,resolution(1)*resolution(2));`
	`buffer.value(pixelIndex);`
	Values are in the range of 0-1 (0=closest to sensor, 1=farthest from sensor). The buffer remains valid until next call to a simx-function.

Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetVisionSensorDepthBuffer2 (regular API equivalent: simGetVisionSensorDepthBuffer)

Description	Retrieves the depth buffer of a vision sensor as an image array. The returned data doesn't make sense if simHandleVisionSensor wasn't called previously (simHandleVisionSensor is called by default in the main script if the vision sensor is not tagged as explicit handling). Use the simxGetLastCmdTime

	function to verify the *freshness* of the retrieved data. This function is much slower than simxGetVisionSensorDepthBuffer. See also simxGetVisionSensorImage.
Matlab synopsis	[number returnCode,array resolution,matrix buffer]=simxGetVisionSensorDepthBuffer2(number clientID,number sensorHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the vision sensor **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **resolution**: 2 number values representing the resolution of the image **buffer**: the depth buffer data. Values are in the range of 0-1 (0=closest to sensor, 1=farthest from sensor).
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetVisionSensorImage (regular API equivalent: simGetVisionSensorImage)

Description	Retrieves the image of a vision sensor as a pointer. The returned data doesn't make sense if simHandleVisionSensor wasn't called previously (simHandleVisionSensor is called by default in the main script if the vision sensor is not tagged as explicit handling). Use the simxGetLastCmdTime function to verify the *freshness* of the retrieved data. See also simxGetVisionSensorImage2, simxSetVisionSensorImage, simxGetVisionSensorDepthBuffer and simxReadVisionSensor.
Matlab synopsis	[number returnCode,array resolution,libpointer image]=simxGetVisionSensorImage(number clientID,number sensorHandle,number options,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the vision sensor **options**: image options, bit-coded: bit0 set: each image pixel is a byte (greyscale image), otherwise each image pixel is a rgb byte-triplet **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)

Matlab return values	**returnCode**: a remote API function return code
	resolution: 2 number values representing the resolution of the image
	image: a libpointer object to the data. To access individual pixels, use following code:
	`image.setDataType('uint8Ptr',1,resolution(1)*resolution(2)*bytesPerPixel);`
	`image.value(pixelIndex);`
	Values are in the range of 0-255. The buffer remains valid until next call to a simx-function.

Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxGetVisionSensorImage2 (regular API equivalent: simGetVisionSensorImage)

Description	Retrieves the image of a vision sensor as an image array. The returned data doesn't make sense if simHandleVisionSensor wasn't called previously (simHandleVisionSensor is called by default in the main script if the vision sensor is not tagged as explicit handling). Use the simxGetLastCmdTime function to verify the *freshness* of the retrieved data. See also simxGetVisionSensorImage, simxSetVisionSensorImage, simxGetVisionSensorDepthBuffer and simxReadVisionSensor.
Matlab synopsis	[number returnCode,array resolution,matrix image]=simxGetVisionSensorImage2(number clientID,number sensorHandle,number options,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the vision sensor **options**: image options, bit-coded: bit0 set: each image pixel is a byte (greyscale image), otherwise each image pixel is a rgb byte-triplet **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **resolution**: 2 number values representing the resolution of the image **image**: the image data. Values are in the range of 0-255.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxJointGetForce (REPRECATED)

Description	DEPRECATED. See simxGetJointForce instead.
Matlab synopsis	[number returnCode,number force]=simxJointGetForce(number clientID,number jointHandle,number

	operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **force**: the force or the torque applied to the joint along/about its z-axis
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxLoadModel (regular API equivalent: simLoadModel)

Description	Loads a previously saved model. See also simxLoadScene and simxTransferFile.
Matlab synopsis	[number returnCode,number baseHandle]=simxLoadModel(number clientID,string modelPathAndName,number options,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **modelPathAndName**: the model filename, including the path and extension ('ttm'). The file is relative to the client or server system depending on the **options** value (see next argument) **options**: 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 **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code **baseHandle**: the loaded model base.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxLoadScene (regular API equivalent: simLoadScene)

Description	Loads a previously saved scene. Should only be called when simulation is not running and is only executed by continuous remote API server services. See also simxCloseScene, simxLoadModel, and simxTransferFile.
Matlab synopsis	[number returnCode]=simxLoadScene(number clientID,string scenePathAndName,number options,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **scenePathAndName**: the scene filename, including the path and extension ('ttt'). The file is relative to the client or server system depending on the **options** value (see next argument) **options**: 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 **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxLoadUI (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxPackFloats

Description	Packs an array of floats into a string. This is a remote API helper function. See also simxUnpackFloats and simxPackInts.
Matlab synopsis	[string packedData]=simxPackFloats(array floatValues)
Matlab parameters	**floatValues**: an array of numbers we wish to pack as floats
Matlab return values	**packedData**: a string that contains the packed values. Each values takes exactly 4 bytes in the string.
Other languages	Java, Octave, Python, Urbi, Lua

simxPackInts

Description	Packs an array of integers into a string. This is a remote API helper function. See also simxUnpackInts and simxPackFloats.
Matlab synopsis	[string packedData]=simxPackInts(array intValues)
Matlab parameters	**intValues**: an array of numbers we wish to pack as integers
Matlab return values	**packedData**: a string that contains the packed values. Each values takes exactly 4 bytes in the string.
Other languages	Java, Octave, Python, Urbi, Lua

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.
Matlab synopsis	[number returnCode]=simxPauseCommunication(number clientIDboolean pause)

Matlab parameters	**clientID**: the client ID. refer to simxStart.
	pause: whether the communication thread should pause or run normally.
	
	Usage example:
	
	`vrep.simxPauseCommunication(clientID,1);`
	`vrep.simxSetJointPosition(clientID,joint1Handle,joint1Value,vrep.simx_opmode_oneshot);`
	`vrep.simxSetJointPosition(clientID,joint2Handle,joint2Value,vrep.simx_opmode_oneshot);`
	`vrep.simxSetJointPosition(clientID,joint3Handle,joint3Value,vrep.simx_opmode_oneshot);`
	`vrep.simxPauseCommunication(clientID,0);`
	
	`% Above's 3 joints will be received and set on the V-REP side at the same time`

Matlab return values	**returnCode**: 0 in case of operation success.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxPauseSimulation (regular API equivalent: simPauseSimulation)

Description	Requests a pause of a simulation. See also simxStartSimulation and simxStopSimulation.
Matlab synopsis	[number returnCode]=simxPauseSimulation(number clientID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **operationMode**: a remote API function operation mode. Recommended operation modes for this function is simx_opmode_oneshot.
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxQuery

Description	DEPRECATED. Refer to simxCallScriptFunction instead.
	
	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 child script, another remote API client or a ROS node handle special requests coming from this remote API client, then send a reply back. To pack/unpack integers/floats into/from a string, refer to simxPackInts, simxPackFloats, simxUnpackInts and simxUnpackFloats.
	
	Usage example where a child script handles a request:
	
	`% Following is the remote API client side:`
	`[res,replyData]=vrep.simxQuery(clientID,'request','send me a 42','reply',5000)`
	`if (res==vrep.simx_return_ok)`
	`  fprintf('The reply is: %s\n',replyData);`
	`end`
	
	`-- 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
```

Matlab synopsis	[number returnCode string retSignalValue]=simxQuery(number clientID,string signalName,string signalValue,string retSignalName,number timeOutInMs)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal that contains the request string **signalValue**: the request string. **retSignalName**: name of the signal that contains the reply string **timeOutInMs**: the maximum time in milliseconds that the function will wait for a reply.
Matlab return value	**returnCode**: a remote API function return code **retSignalValue**: the reply string
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxReadCollision (regular API equivalent: simReadCollision)

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 simHandleCollision (simHandleCollision is called in the default main script). See also simxGetObjectGroupData.
Matlab synopsis	[number returnCode,boolean collisionState]=simxReadCollision(number clientID,number collisionObjectHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **collisionObjectHandle**: handle of the collision object **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **collisionState**: the collision state (false: not colliding)
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxReadDistance (regular API equivalent: simReadDistance)

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 simHandleDistance (simHandleDistance is called in the default main script). See also simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number minimumDistance]=simxReadDistance(number clientID,number distanceObjectHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **distanceObjectHandle**: handle of the distance object **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **minimumDistance**: the minimum distance. If the distance object wasn't handled yet, the distance value will be larger than 1e36.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxReadForceSensor (regular API equivalent: simReadForceSensor)

Description	Reads the force and torque applied to a force sensor (filtered values are read), and its current state ('unbroken' or 'broken'). See also simxBreakForceSensor, simxGetJointForce and simxGetObjectGroupData.
Matlab synopsis	[number returnCode,number state,array forceVector,array torqueVector]=simxReadForceSensor(number clientID,number forceSensorHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **forceSensorHandle**: handle of the force sensor **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **state**: the state of the force sensor: 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') **forceVector**: 3 values representing the force vector **torqueVector**: 3 values representing the torque vector
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxReadProximitySensor (regular API equivalent: simReadProximitySensor)

Description	Reads the state of a proximity sensor. This function doesn't perform detection, it merely reads the result from a previous call to simHandleProximitySensor (simHandleProximitySensor is called in the default main script). See also simxGetObjectGroupData.
Matlab synopsis	[number returnCode,boolean detectionState,array detectedPoint,number detectedObjectHandle,array detectedSurfaceNormalVector]=simxReadProximitySensor(number clientID,number sensorHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the proximity sensor **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **detectionState**: the detection state (false=no detection) **detectedPoint**: the detected point coordinates (relative to the sensor reference frame) **detectedObjectHandle**: the handle of the detected object **detectedSurfaceNormalVector**: the normal vector (normalized) of the detected surface. Relative to the sensor reference frame
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxReadStringStream

Description	Gets the value of a string signal, then clears it. Useful to retrieve continuous data from the server. To pack/unpack integers/floats into/from a string, refer to simxPackInts, simxPackFloats, simxUnpackInts and simxUnpackFloats. See also simxWriteStringStream.
Matlab synopsis	[number returnCode,string signalValue]=simxReadStringStream(number clientID,string signalName,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls). simx_opmode_blocking is forbidden. Use a construction like following in order to continuously exchange data with V-REP: Remote API client side:

```
% Initialization phase:
[err,signal]=vrep.simxReadStringStream(clientID,'toClient',
        vrep.simx_opmode_streaming);
% while we are connected:
while (vrep.simxGetConnectionId(clientID)~=-1)
  [err,signal]=vrep.simxReadStringStream(clientID,'toClient',
        vrep.simx_opmode_buffer);
  if (err==vrep.simx_return_ok)
    % Data produced by the child script was retrieved! Send it back to the child script:
    vrep.simxWriteStringStream(clientID,'fromClient',signal,
        vrep.simx_opmode_oneshot);
  end
end
```

Server side (V-REP), from a non-threaded child script:

```
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
```

Matlab return values	**returnCode**: a remote API function return code **signalValue**: the signal data (that may contain any value, including embedded zeros)
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxReadVisionSensor (regular API equivalent: simReadVisionSensor)

Description	Reads the state of a vision sensor. This function doesn't perform detection, it merely reads the result from a previous call to simHandleVisionSensor (simHandleVisionSensor is called in the default main script). See also simxGetVisionSensorImage and simxGetObjectGroupData.
Matlab synopsis	[number returnCode,boolean detectionState,array auxData,array auxPacketInfo]=simxReadVisionSensor(number clientID,number sensorHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the vision sensor **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_streaming (the first call) and simx_opmode_buffer (the following calls)
Matlab return values	**returnCode**: a remote API function return code **detectionState**: the detection state (i.e. the trigger state) **auxData**: all auxiliary values returned from the applied filters. 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 after the first packet. **auxPacketInfo**: an array containing an entry for each returned packet. Each entry represents the number of values in each packets. Use this info to extract individual packets from **auxData**.
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxReleaseBuffer (regular API equivalent: simReleaseBuffer)

Description	Releases a buffer previously created with simxCreateBuffer or a buffer returned by a remote API function. This is a remote API helper function.
Matlab synopsis	simxReleaseBuffer(libpointer buffer)
Matlab parameters	**buffer**: buffer to be released
Matlab return values	none
Other languages	C/C++, Python

simxRemoveModel (regular API equivalent: simRemoveModel)

Description	Removes a model from the scene. See also simxRemoveObject.
Matlab synopsis	[number returnCode]=simxRemoveModel(number clientID,number objectHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the model to remove (object should be flagged as *model base*). **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot (or simx_opmode_blocking)
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxRemoveObject (regular API equivalent: simRemoveObject)

Description	Removes a scene object. See also simxRemoveModel.
Matlab synopsis	[number returnCode]=simxRemoveObject(number clientID,number objectHandle,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart.

	objectHandle: handle of the object to remove **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot (or simx_opmode_blocking)
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxRemoveUI (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxSetArrayParameter (regular API equivalent: simSetArrayParameter)

Description	Sets 3 values of an array parameter. See also simxGetArrayParameter, simxSetBooleanParameter, simxSetIntegerParameter and simxSetFloatingParameter.
Matlab synopsis	[number returnCode]=simxSetArrayParameter(number clientID,number paramIdentifier,array paramValues,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: an array parameter identifier **paramValues**: the array containing the 3 values to set **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetBooleanParameter (regular API equivalent: simSetBoolParameter)

Description	Sets a boolean parameter. See also simxGetBooleanParameter, simxSetIntegerParameter, simxSetArrayParameter and simxSetFloatingParameter.
Matlab synopsis	[number returnCode]=simxSetBooleanParameter(number clientID,number paramIdentifier,boolean paramValue,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: a Boolean parameter identifier **paramValue**: the parameter value **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetFloatingParameter (regular API equivalent: simSetFloatParameter)

Description	Sets a floating point parameter. See also simxGetFloatingParameter, simxSetBooleanParameter, simxSetArrayParameter and simxSetIntegerParameter.
Matlab synopsis	[number returnCode]=simxSetFloatingParameter(number clientID,number paramIdentifier,number paramValue,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: a floating parameter identifier **paramValue**: the parameter value **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetFloatSignal (regular API equivalent: simSetFloatSignal)

Description	Sets the value of a float signal. If that signal is not yet present, it is added. See also simxGetFloatSignal, simxClearFloatSignal, simxSetIntegerSignal and simxSetStringSignal.
Matlab synopsis	[number returnCode]=simxSetFloatSignal(number clientID,string signalName,number signalValue,number operationMode)

Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **signalValue**: value of the signal **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetIntegerParameter (regular API equivalent: simSetInt32Parameter)

Description	Sets an integer parameter. See also simxGetIntegerParameter, simxSetBooleanParameter, simxSetArrayParameter and simxSetFloatingParameter.
Matlab synopsis	[number returnCode]=simxSetIntegerParameter(number clientID,number paramIdentifier,number paramValue,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **paramIdentifier**: an integer parameter identifier **paramValue**: the parameter value **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetIntegerSignal (regular API equivalent: simSetIntegerSignal)

Description	Sets the value of an integer signal. If that signal is not yet present, it is added. See also simxGetIntegerSignal, simxClearIntegerSignal, simxSetFloatSignal and simxSetStringSignal.
Matlab synopsis	[number returnCode]=simxSetIntegerSignal(number clientID,string signalName,number signalValue,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **signalValue**: value of the signal **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetJointForce (regular API equivalent: simSetJointForce)

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 simxGetJointForce.
Matlab synopsis	[number returnCode]=simxSetJointForce(number clientID,number jointHandle,number force,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **force**: the maximum force or torque that the joint can exert **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetJointPosition (regular API equivalent: simSetJointPosition)

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 simxSetSphericalJointMatrix instead). If you want to set several joints that should be applied at the exact same time on the V-REP side, then use simxPauseCommunication. See also simxGetJointPosition and simxSetJointTargetPosition.
Matlab synopsis	[number returnCode]=simxSetJointPosition(number clientID,number jointHandle,number position,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint

	position: position of the joint (angular or linear value depending on the joint type) **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_oneshot or simx_opmode_streaming
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetJointTargetPosition (regular API equivalent: simSetJointTargetPosition)

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 simxSetJointPosition.
Matlab synopsis	[number returnCode]=simxSetJointTargetPosition(number clientID,number jointHandle,number targetPosition,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **targetPosition**: target position of the joint (angular or linear value depending on the joint type) **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_oneshot or simx_opmode_streaming
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetJointTargetVelocity (regular API equivalent: simSetJointTargetVelocity)

Description	Sets the intrinsic target velocity of a non-spherical joint. This command makes only sense when the joint mode is in torque/force mode: the dynamics functionality and the joint motor have to be enabled (position control should however be disabled)
Matlab synopsis	[number returnCode]=simxSetJointTargetVelocity(number clientID,number jointHandle,number targetVelocity,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **targetVelocity**: target velocity of the joint (linear or angular velocity depending on the joint-type) **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_oneshot or simx_opmode_streaming
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetModelProperty (regular API equivalent: simSetModelProperty)

Description	Sets the properties of a model. See also simxGetModelProperty.
Matlab synopsis	[number returnCode]=simxSetModelProperty(number clientID,number objectHandle,number prop,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **prop**: a model property value **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetObjectFloatParameter (regular API equivalent: simSetObjectFloatParameter)

Description	Sets a floating-point parameter of a object. See also simxGetObjectFloatParameter and simxSetObjectIntParameter.
Matlab synopsis	[number returnCode]=simxSetObjectFloatParameter(number clientID,number objectHandle,number parameterID,number parameterValue,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **parameterID**: identifier of the parameter to set. See the list of all possible object parameter identifiers **parameterValue**: the desired value of the parameter **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot

Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetObjectIntParameter (regular API equivalent: simSetObjectInt32Parameter)

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

simxSetObjectOrientation (regular API equivalent: simSetObjectOrientation)

Description	Sets the orientation (Euler angles) of an object. Dynamically simulated objects will implicitely be reset before the command is applied (i.e. similar to calling simResetDynamicObject just before). See also simxGetObjectOrientation and simxSetObjectPosition.
Matlab synopsis	[number returnCode]=simxSetObjectOrientation(number clientID,number objectHandle,number relativeToObjectHandle,array eulerAngles,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **relativeToObjectHandle**: indicates relative to which reference frame the orientation is specified. Specify -1 to set the absolute orientation, sim_handle_parent to set the orientation relative to the object's parent, or an object handle relative to whose reference frame the orientation is specified. **eulerAngles**: Euler angles (alpha, beta and gamma) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetObjectParent (regular API equivalent: simSetObjectParent)

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

simxSetObjectPosition (regular API equivalent: simSetObjectPosition)

Description	Sets the position of an object. Dynamically simulated objects will implicitely be reset before the command is applied (i.e. similar to calling simResetDynamicObject just before). See also simxGetObjectPosition and simxSetObjectOrientation.
Matlab synopsis	[number returnCode]=simxSetObjectPosition(number clientID,number objectHandle,number

	relativeToObjectHandle,array position,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandle**: handle of the object **relativeToObjectHandle**: indicates relative to which reference frame the position is specified. Specify -1 to set the absolute position, sim_handle_parent to set the position relative to the object's parent, or an object handle relative to whose reference frame the position is specified. **position**: the position values (x, y and z) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetObjectSelection

Description	Sets the selection state for objects. See also simxGetObjectSelection.
Matlab synopsis	[number returnCode]=simxSetObjectSelection(number clientID,array objectHandles,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **objectHandles**: an array of object handles **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot or simx_opmode_blocking depending on the intent.
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetSphericalJointMatrix (regular API equivalent: simSetSphericalJointMatrix)

Description	Sets the intrinsic orientation matrix of a spherical joint object. This function cannot be used with non-spherical joints (use simxSetJointPosition instead). See also simxGetJointMatrix..
Matlab synopsis	[number returnCode]=simxSetSphericalJointMatrix(number clientID,number jointHandle,array matrix,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **jointHandle**: handle of the joint **matrix**: 12 number values. See the regular API equivalent function for details **operationMode**: a remote API function operation mode. Recommended operation modes for this function are simx_opmode_oneshot or simx_opmode_streaming
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetStringSignal (regular API equivalent: simSetStringSignal)

Description	Sets the value of a string signal. If that signal is not yet present, it is added. To pack/unpack integers/floats into/from a string, refer to simxPackInts, simxPackFloats, simxUnpackInts and simxUnpackFloats. See also simxWriteStringStream, simxGetStringSignal, simxClearStringSignal, simxSetIntegerSignal and simxSetFloatSignal.
Matlab synopsis	[number returnCode]=simxSetStringSignal(number clientID,string signalName,string signalValue,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **signalValue**: value of the signal (which may contain any value, including embedded zeros) **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetUIButtonLabel (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxSetUIButtonProperty (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxSetUISlider (DEPRECATED)

Description	DEPRECATED. Use the Qt-based custom user interfaces, via simxCallScriptFunction instead.

simxSetVisionSensorImage (regular API equivalent: simSetVisionSensorImage)

Description	Sets the image of a vision sensor (and applies any image processing filter if specified in the vision sensor dialog). The image is provided as a libpointer. Make sure the vision sensor is flagged as **use external image**. The *regular* use of this function is to first read the data from a vision sensor with simxGetVisionSensorImage, 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. See also simxSetVisionSensorImage2.
Matlab synopsis	[number returnCode]=simxSetVisionSensorImage(number clientID,number sensorHandle,libpointer image,number bufferSize,number options,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the vision sensor **image**: the image data **bufferSize**: size of the image data **options**: image options, bit-coded: bit0 set: each image pixel is a byte (greyscale image), otherwise each image pixel is a rgb byte-triplet **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSetVisionSensorImage2 (regular API equivalent: simSetVisionSensorImage)

Description	Sets the image of a vision sensor (and applies any image processing filter if specified in the vision sensor dialog). The image is provided as an image array. Make sure the vision sensor is flagged as **use external image**. The *regular* use of this function is to first read the data from a vision sensor with simxGetVisionSensorImage2, 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. See also simxSetVisionSensorImage which is much faster.
Matlab synopsis	[number returnCode]=simxSetVisionSensorImage2(number clientID,number sensorHandle,matrix image,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **sensorHandle**: handle of the vision sensor **image**: the image data **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

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 simxFinish. This is a remote API helper function.
Matlab synopsis	[number clientID]=simxStart(string connectionAddress,number connectionPort,boolean waitUntilConnected,boolean doNotReconnectOnceDisconnected,number timeOutInMs,number commThreadCycleInMs)
Matlab parameters	**connectionAddress**: the ip address where the server is located (i.e. V-REP) **connectionPort**: the port number where to connect **waitUntilConnected**: if true, then the function blocks until connected (or timed out).

	doNotReconnectOnceDisconnected: if true, then the communication thread will not attempt a second connection if a connection was lost. **timeOutInMs**: if positive: the connection time-out in milliseconds for the first connection attempt. In that case, the time-out for blocking function calls is 5000 milliseconds. if negative: its positive value is the time-out for blocking function calls. In that case, the connection time-out for the first connection attempt is 5000 milliseconds. **commThreadCycleInMs**: indicates how often data packets are sent back and forth. Reducing this number improves responsiveness, and a default value of 5 is recommended.
Matlab return values	**clientID**: the client ID, or -1 if the connection to the server was not possible (i.e. a timeout was reached). A call to simxStart should always be followed at the end with a call to simxFinish if simxStart didn't return -1
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxStartSimulation (regular API equivalent: simStartSimulation)

Description	Requests a start of a simulation (or a resume of a paused simulation). This function is only executed by continuous remote API server services. See also simxPauseSimulation and simxStopSimulation.
Matlab synopsis	[number returnCode]=simxStartSimulation(number clientID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot.
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxStopSimulation (regular API equivalent: simStopSimulation)

Description	Requests a stop of the running simulation. See also simxStartSimulation and simxPauseSimulation.
Matlab synopsis	[number returnCode]=simxStopSimulation(number clientID,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **operationMode**: a remote API function operation mode. Recommended operation modes for this function is simx_opmode_oneshot.
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSynchronous

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 simxSynchronousTrigger and this section. This is a remote API helper function.
Matlab synopsis	[number returnCode]=simxSynchronous(number clientID,boolean enable)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **enable**: the enable state of the synchronous operation
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxSynchronousTrigger

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 simxSynchronous function. The trigger signal will inform V-REP to execute the next simulation step (i.e. to call simHandleMainScript). While in synchronous operation mode, the client application is in charge of triggering the next simulation step, otherwise simulation will stall. See also this section. This is a remote API helper function.
Matlab synopsis	[number returnCode]=simxSynchronousTrigger(number clientID)
Matlab parameters	**clientID**: the client ID. refer to simxStart.
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxTransferFile

Description	Allows transferring a file from the client to the server. This function is used by several other functions internally (e.g. simxLoadModel). See also simxEraseFile. This is a remote API helper function.
Matlab synopsis	[number returnCode]=simxTransferFile(number clientID,string filePathAndName,string fileName_serverSide,number timeOut,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **filePathAndName**: the local file name and path (i.e. on the client side) **fileName_serverSide**: 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) **timeOut**: a timeout value in milliseconds **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_blocking
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua

simxUnpackFloats

Description	Unpacks a string into an array of floats. This is a remote API helper function. See also simxPackFloats and simxUnpackInts.
Matlab synopsis	[array floatValues]=simxUnpackFloats(string packedData)
Matlab parameters	**packedData**: a string that contains the packed values. Each values takes exactly 4 bytes in the string.
Matlab return values	**floatValues**: an array of numbers that were unpacked as floats
Other languages	Java, Octave, Python, Urbi, Lua

simxUnpackInts

Description	Unpacks a string into an array of integers. This is a remote API helper function. See also simxPackInts and simxUnpackFloats.
Matlab synopsis	[array intValues]=simxUnpackInts(string packedData)
Matlab parameters	**packedData**: a string that contains the packed values. Each values takes exactly 4 bytes in the string.
Matlab return values	**intValues**: an array of numbers that were unpacked as integers
Other languages	Java, Octave, Python, Urbi, Lua

simxWriteStringStream

Description	Appends a string to a string signal. If that signal is not yet present, it is added. To pack/unpack integers/floats into/from a string, refer to simxPackInts, simxPackFloats, simxUnpackInts and simxUnpackFloats. See also simxReadStringStream.
Matlab synopsis	[number returnCode]=simxWriteStringStream(number clientID,string signalName,string signalValueToAppend,number operationMode)
Matlab parameters	**clientID**: the client ID. refer to simxStart. **signalName**: name of the signal **signalValueToAppend**: value to append to the signal. That value may contain any value, including embedded zeros. **operationMode**: a remote API function operation mode. Recommended operation mode for this function is simx_opmode_oneshot
Matlab return values	**returnCode**: a remote API function return code
Other languages	C/C++, Python, Java, Octave, Urbi, Lua