

---

# **Blender-VR API**

*Release 1.0*

**Blender-VR Team**

August 19, 2015

<b>1 Processor File</b>	<b>2</b>
1.1 Examples . . . . .	2
1.2 OSC API . . . . .	3
<b>2 Source Code</b>	<b>8</b>
2.1 Utils . . . . .	8
2.2 Modules . . . . .	9
<b>3 Indices and tables</b>	<b>30</b>
<b>Python Module Index</b>	<b>31</b>

The documentation of the Blender-VR project includes the part that is exposed to the user to be used in the processor file, as well as the core code of Blender-VR.

---

## Processor File

---

### 1.1 Examples

For more examples, check the `processor` files in the [Samples Repository](#) of the Blender-VR project.

#### 1.1.1 Basic Example

This is a basic processor file which can be considered a barebone and a start point for your own. All it does is to synchronize all the objects between the master and the slaves machines.

```

1  import blendervr
2
3  if blendervr.is_console():
4      class Processor(blendervr.processor.getProcessor()):
5          def __init__(self, console):
6              global try_wait_user_name, try_chooser, try_console_arc_balls
7              super(Processor, self).__init__(console)
8
9          def useLoader(self):
10             return True
11
12 elif blendervr.is_creating_loader():
13     import bpy
14
15     class Processor(blendervr.processor.getProcessor()):
16         def __init__(self, creator):
17             super(Processor, self).__init__(creator)
18
19 elif blendervr.is_virtual_environment():
20     import bge
21
22     class Processor(blendervr.processor.getProcessor()):
23         def __init__(self, parent):
24             super(Processor, self).__init__(parent)
25
26         if self.blenderVR.isMaster():
27             self.blenderVR.getSceneSynchronizer().\
28                 getItem(bge.logic).activate(True, True)

```

The file is split in three parts:

1. *Console*

2. *Update Loader*
3. *Virtual Environment*

The processor file is called three times, and each time a section of it is called.

## Console

The console part of the code is called first by the `console`. This runs before your `.blend` file is even loaded. The `useLoader()` determines if you need Blender-VR to modify your `.blend` on-the-fly.

Most of the time this won't need to change. The exception is when the file being loaded was already modified to work with Blender-VR (e.g., the file generated on-the-fly after running it once).

```
if blendervr.is_console():
    class Processor(blendervr.processor.getProcessor()):
        def __init__(self, console):
            global try_wait_user_name, try_chooser, try_console_arc_balls
            super(Processor, self).__init__(console)

        def useLoader(self):
            return True
```

## Update Loader

If a project requires specific changes in the `.blend` file they are introduced here. This is the place where a specific `Actuator` can be added for a Head-Mounted display for example.

```
elif blendervr.is_creating_loader():
    import bpy

    class Processor(blendervr.processor.getProcessor()):
        def __init__(self, creator):
            super(Processor, self).__init__(creator)
```

## Virtual Environment

This part of the code is called when the `.blend` file is loaded in the Blender Game Engine. The most basic usage is to synchronize all the scene objects, as it's being done here.

```
elif blendervr.is_virtual_environment():
    import bge

    class Processor(blendervr.processor.getProcessor()):
        def __init__(self, parent):
            super(Processor, self).__init__(parent)

            if self.blenderVR.isMaster():
                self.blenderVR.getSceneSynchronizer().\
                    getItem(bge.logic).activate(True, True)
```

## 1.2 OSC API

How to use the OSC API integrated in blenderVR.

## 1.2.1 Introduction

OSC is integrated as a BlenderVR plugin, to be defined in the .xml configuration file. The OSC API is e.g. used for BlenderVR synchroniztion with the Max/MSP based [Sound Rendering Engine](#).

As a start, you'll want to adapt BlenderVR [configuration file](#) to your architecture, process detailed in the [how to use OSC](#) section.

For the sake of illustration, say you're using a configuration file which `<osc>` subsection looks like:

```
<plugins>

  <osc host='localhost' port='3819' configuration='Laptop SPAT' max_audio_objects='8'>

    <user listener='Binaural 1' viewer='user A' />
    <user listener='Ambisonic' />
    <user listener='Stereo' />

  </osc>

</plugins>
```

Every parameter defined in these lines will be sent to the OSC client at BlenderVR startup (but for osc host and port), see the [how to use OSC](#) section for more details on each flag.

Received by the OSC client at BlenderVR start:

```
/global configuration Laptop SPAT
/global max_audio_objects 8
/user 0 name Binaural 1
/user 1 name Ambisonic
/user 2 name Stereo
```

```
OSC = self.blenderVR.getPlugin('osc')
```

with `self` representing the BlenderVR processor object, granting access to BlenderVR OSC module and its API (OSC in python code bellow refers to this module).

From there, BlenderVR OSC API proposes 4 different class of messages: `global`, `user`, `object` and `objectUser`.

## 1.2.2 Global Messages

Global messages can be used for global configuration of the Sound Engine (e.g. global volume, start, etc.).

```
osc_global = OSC.getGlobal()
osc_global.start(True)
osc_global.mute(True)
osc_global.volume('%45')
```

will send the following messages to the Sound Engine:

```
/global start 1
/global mute 1
/global volume %45
```

with volume being either absolute balue (e.g. `%45`) or +/- relative add (e.g. `+3` or `-7`).

### 1.2.3 User Messages

User messages can be used for user specific configuration of the Sound Engine (e.g. `user volume`, “user start”, etc.). See OSC users as listeners, or rather as the media + rendering technique that produces a sound (speakers set + ambisonic, headset + binaural, etc.).

```
osc_user = OSC.getUser('Binaural 1')
# or equivalently
bvr_user = self.blenderVR.getUserByName('user A')
osc_user = OSC.getUser(bvr_user)
# --
osc_user.start(True)
osc_user.mute(True)
osc_user.volume('%45')
```

The first line grants access to the OSC user named `Binaural 1` in the configuration file (attached to BlenderVR “user A”, see above). Thanks to the definition of user / listener in the configuration file, each BlenderVR user position/orientation (‘user A’ here) will be synchronized from BlenderVR to the sound rendering engine.

The next lines will send the following messages to the Sound Engine:

```
/user 1 start 1
/user 1 mute 1
/user 1 volume %45
```

and BlenderVR will constantly update osc user position with messages like:

```
/user 1 position 1. 0. 0. 0. 0. 0. -1. -0. 0. 0. 0. 0. -1. 0. 0. 0. 0. 1.
```

where the 16 floats represent the 4x4 homogeneous Matrix of user position/orientation in the virtual world.

### 1.2.4 Object Messages

Object messages can be used for object specific configuration of the Sound Engine (e.g. `object volume`, `object start`, etc.). See OSC objects as a virtual sound source instantiated in the Sound Engine, that will be attached to a BlenderVR object (e.g. a Blender `KX_Game_Object`) in the scene and eventually heard by one/many OSC user/listener (see `objectUser` messages below).

```
scene = bge.logic.getCurrentScene()
kx_object = scene.objects['Cube']
osc_object = OSC.getObject(kx_object)
osc_object.sound('HeyPachuco.wav')
osc_object.start(True)
osc_object.mute(False)
osc_object.volume('%45')
```

The first line grants access to the OSC object that will be attached to the `KX_GameObject Cube` in the blender scene. This first line triggers a callback that will synchronize the object position in the The next lines will send the following messages to the Sound Engine:

```
/object 1 sound HeyPachuco.wav
/object 1 start 1
/object 1 mute 0
/object 1 volume %45
```

and BlenderVR will constantly update osc object position with messages like:

```
/object 1 position 0.54156 0.132934 -0.830085 0. -0.840592 0.07291 -0.536739 0. -0.01083 0.98844 0.15122
```

## 1.2.5 ObjectUser Messages

This class of messages allow to dynamically route object sounds to osc users (listeners) audio input. basically, sending:

```
/objectUser 1 0 mute 0
```

will tell the sound engine to route osc object 1 to osc user 0 (Binaural 1 here, see above), hence the listener Binaural 1 will hear the sound of kx\_object 1.

The line `SC.getObjectUser(osc_object, osc_user)` grants access to the OSC objectUser that will control the link between the sound from the osc object (attached to the blender object ``Cube) to the osc user ``Binaural 1'. The next two lines will send the following messages to the Sound Engine:

```
/objectUser 1 0 mute 0
/objectUser 1 0 volume %50
```

## 1.2.6 Example

The basic-osc.blend in the BlenderVR [samples](#) will send the following OSC messages to the Sound Engine (it's actually the code in the basic-osc.processor.py along with the osc plugin definition in the ``//blender-vr/configuration/main.xml' configuration file that will send the following messages):

```
/global configuration Laptop SPAT
/global max_audio_objects 20
/global volume %40
/global start 1
/global mute 0
/object 1 sound HeyPachuco.wav
/object 1 loop 1
/object 1 volume %45
/object 1 start 1
/object 1 position 0.54156 0.132934 -0.830085 0. -0.840592 0.07291 -0.536739 0. -0.01083 0.98844 0.15122
/object 1 mute 0
/user 2 name Ambisonic
/user 2 hrtf 0
/user 2 volume %50
/user 2 position
/user 2 start 0
/user 2 mute 0
/user 2 warmth 0
/user 2 brightness 0
/user 2 presence 0
/user 2 reverb_volume 0
/user 2 running_reverb 0
/user 2 late_reverb 0
/user 2 envelop 0
/user 2 heavyness 0
/user 2 livelyness 0
/user 0 name Binaural 1
/user 0 hrtf 0
/user 0 volume %80
/user 0 position 1. 0. 0. 0. 0. -1. -0. 0. 0. 0. -1. 0. 0. 0. 0. 1.
/user 0 start 1
```



```
/user 0 mute 0
/user 0 warmth 0
/user 0 brightness 0
/user 0 presence 0
/user 0 reverb_volume 0
/user 0 running_reverb 0
/user 0 late_reverb 0
/user 0 envelop 0
/user 0 heavyness 0
/user 0 liveliness 0
/user 1 name Binaural 2
/user 1 hrtf 0
/user 1 volume %50
/user 1 position 1. 0. 0. 0. 0. 0. -1. -0. 0. 0. 0. 0. -1. 0. 0. 0. 0. 0. 1.
/user 1 start 0
/user 1 mute 0
/user 1 warmth 0
/user 1 brightness 0
/user 1 presence 0
/user 1 reverb_volume 0
/user 1 running_reverb 0
/user 1 late_reverb 0
/user 1 envelop 0
/user 1 heavyness 0
/user 1 liveliness 0
/user 3 name Stereo
/user 3 hrtf 0
/user 3 volume %50
/user 3 position
/user 3 start 0
/user 3 mute 0
/user 3 warmth 0
/user 3 brightness 0
/user 3 presence 0
/user 3 reverb_volume 0
/user 3 running_reverb 0
/user 3 late_reverb 0
/user 3 envelop 0
/user 3 heavyness 0
/user 3 liveliness 0
/objectUser 1 0 volume %50
/objectUser 1 0 mute 0
/object 1 position 0.529771 0.133939 -0.837498 0. -0.848072 0.071046 -0.525097 0. -0.01083 0.98844 0.151
/object 1 position 0.517878 0.134918 -0.844748 0. -0.855386 0.069169 -0.513353 0. -0.01083 0.98844 0.151
(... moving objects / users position updates ...)
```

## 2.1 Utils

Files to support the Blender VR application.

### 2.1.1 Daemon

This script runs in the clients and is responsible for spawning the Blender Player.

```
class daemon.Daemon (BlenderVR_modules)
```

```
    Bases: object
```

```
    Background management of the Blender Player and related stuff.
```

```
    main ()
```

```
        Start the Daemon, quits any instance of BlenderPlayer running.
```

```
    processCommand (command, argument)
```

```
        Run the received commands
```

#### Parameters

- **command** (*str*) – Command to execute in the client machine
- **argument** – Value depends on the command

```
    write (*messages)
```

```
        Send message to the client
```

```
        Parameters messages (list) – all the messages to send to the client (i.e., console commands)
```

```
daemon.main ()
```

```
    Main function to start the daemon.
```

```
    Prepare execution (daemonize if necessary), then build a Daemon and call its main() method to manage background communications.
```

### 2.1.2 Update Loader

Script that runs in Blender in background mode to transform the `.blend` file into a Blender-VR ready file. A few Logic Bricks are created among other changes in the initial scene.

## 2.2 Modules

Main modules for the Blender-VR application.

### 2.2.1 blendervr package

#### Subpackages

**blendervr.console** package

#### Subpackages

**blendervr.console.gui** package

#### Submodules

##### **blendervr.console.gui.console** module

**class** `blendervr.console.gui.console.GUI`

Bases: `object`

**display\_screen\_sets** (*screenSets*)

**main** ()

**quit** ()

**start** ()

##### **blendervr.console.gui.logger** module

**class** `blendervr.console.gui.logger.Logger` (*parent, config\_index*)

Bases: `blendervr.console.base.Base`

**clear** ()

**flush** ()

**quit** ()

**set\_log\_level** (*log\_level*)

**set\_stream\_state** (*stream, state*)

**start** ()

**write** (*\*messages*)

##### **blendervr.console.gui.options** module

**class** `blendervr.console.gui.options.GUI`

Bases: `object`

**is\_options\_window\_opened** ()

**quit** ()

**start** ()

**blendervr.console.gui.screen module****class** `blendervr.console.gui.screen.GUI`Bases: `object``getFrame()``is_log_window_opened()``quit()``start()`**blendervr.console.gui.screens module****class** `blendervr.console.gui.screens.GUI`Bases: `object``quit()``show(state)``start()``update_gui()`**Module contents****blendervr.console.logic package****Submodules****blendervr.console.logic.console module****class** `blendervr.console.logic.console.Logic`Bases: `object``compile_BC()``get_blender_player_state()``load_configuration_file()``quit()``receivedFromVirtualEnvironment(message)``sendToVirtualEnvironment(command, argument)``set_screen_set()``start()``start_simulation()``stop_simulation()``update_user_files(force=False)`

**blendervr.console.logic.file\_name module**

```
class blendervr.console.logic.file_name.FileName (file_name, anchor=None)
    Bases: object

    strip (anchor)

    unstrip (anchor)
```

**blendervr.console.logic.screen module**

```
class blendervr.console.logic.screen.Logic (net_console)
    Bases: object

    adapt_simulation_files_to_screen (loader_file, blender_file, processor_files)

    ask_blender_player_to_quit ()

    daemon_is_running ()

    getHostname ()

    get_blender_player_state ()

    is_master ()

    quit ()

    restartDaemon ()

    send_to_blender_player (command, argument='')

    setConfiguration (configuration, complements)

    setHierarchy (informations)

    setNetworkClient (origin, client, addr)

    set_BlenderVR_state (state)

    start ()
```

**blendervr.console.logic.screens module**

```
class blendervr.console.logic.screens.Logic
    Bases: object

    adapt_simulation_files_to_screen (loader_file, blender_file, processor_files)

    getMaster ()

    getScreen (screen_name)

    getScreensNumber ()

    getStates ()

    quit ()

    send_to_blender_player (command, message)

    set_screens (configurations, net_console, master_name, port, complements)

    start ()

    start_simulation ()

    stop_simulation ()
```

## Module contents

### blendervr.console.qt package

#### Submodules

#### blendervr.console.qt.console module

**class** `blendervr.console.qt.console.GUI`

Bases: `blendervr.console.gui.console.GUI`

**addListenTo** (*socket, callback, data=None*)

**addTimeout** (*time, callback*)

**cb\_add\_configuration\_path** ()

**cb\_close** ()

**cb\_load\_configuration\_file** ()

**cb\_processor\_window** ()

**cb\_remove\_configuration\_path** (\*args)

**cb\_set\_blender\_file** ()

**cb\_set\_configuration\_file** ()

**cb\_set\_current\_tab** ()

**cb\_set\_link\_processor\_to\_blender** ()

**cb\_set\_processor\_file** ()

**cb\_set\_screen\_set** ()

**cb\_start** ()

**cb\_stop** ()

**cb\_update\_liste\_paths** (\*args)

**display\_screen\_sets** (*screenSets*)

**getWindow** ()

**main** ()

**quit** ()

**removeListenTo** (*tag*)

**start** ()

**updateStatus** (*message, state=None*)

**update\_processor** ()

**class** `blendervr.console.qt.console.MainWindow` (*owner, profile, profile\_indices*)

Bases: `blendervr.tools.gui.qt.Common`, `dummy.PyQt4.QMainWindow`

`blendervr.console.qt.console.quit` ()

**blendervr.console.qt.logger module****class** `blendervr.console.qt.logger.Logger` (*parent, config\_index, window, log\_level\_selector*)Bases: *blendervr.console.gui.logger.Logger*`cb_set_log_level()``clear()`**blendervr.console.qt.options module****class** `blendervr.console.qt.options.GUI` (*parent*)Bases: *blendervr.console.base.Base, blendervr.console.gui.options.GUI*`BlenderVR_state_changed(state)``cb_debug_daemon()``cb_debug_processor()``cb_executables()``cb_reload_processor()``cb_restart_daemons()``close()``getMenu()``getmenu(index)``quit()``start()`**blendervr.console.qt.screen module****class** `blendervr.console.qt.screen.GUI`Bases: *blendervr.console.gui.screen.GUI*`cb_toggle_stderr_state()``cb_toggle_stdout_state()``close()``getMenu(index)``is_log_window_opened()``quit()``start()`**blendervr.console.qt.screens module****class** `blendervr.console.qt.screens.GUI`Bases: *blendervr.console.gui.screens.GUI*`close_all()``quit()``start()``update_gui()`

**blendervr.console.qt.tools module****class** `blendervr.console.qt.tools.MessagesColors`Bases: `object``getColors` (*level*)**Module contents****blendervr.console.xml package****Subpackages****blendervr.console.xml.device package****Subpackages****blendervr.console.xml.device.hmd package****Submodules****blendervr.console.xml.device.hmd.base module****class** `blendervr.console.xml.device.hmd.base.Device` (*parent, name, attrs*)Bases: `blendervr.console.xml.device.base.Base`**blendervr.console.xml.device.hmd.oculus\_dk2 module****class** `blendervr.console.xml.device.hmd.oculus_dk2.Device` (*parent, name, attrs*)Bases: `blendervr.console.xml.device.hmd.base.Device`**Module contents**`blendervr.console.xml.device.hmd.Device` (*parent, name, attrs*)**Submodules****blendervr.console.xml.device.base module****class** `blendervr.console.xml.device.base.Base` (*parent, name, attrs*)Bases: `blendervr.console.xml.base.XML`**blendervr.console.xml.device.planovision module****class** `blendervr.console.xml.device.planovision.Device` (*parent, name, attrs*)Bases: `blendervr.console.xml.device.screen.Screen`



**blendervr.console.xml.device.screen module**

```
class blendervr.console.xml.device.screen.Screen (parent, name, attrs)
    Bases: blendervr.console.xml.device.base.Base

    characters (string)

    endElement (name)
```

**blendervr.console.xml.device.wall module**

```
class blendervr.console.xml.device.wall.Device (parent, name, attrs)
    Bases: blendervr.console.xml.device.screen.Screen
```

**Module contents**

```
blendervr.console.xml.device.getDevice (parent, name, attrs)
```

**Submodules****blendervr.console.xml.base module**

```
class blendervr.console.xml.base.XML (parent, name, attrs)
    Bases: xml.sax.handler.ContentHandler, blendervr.console.base.Base

    endElement (name)

    getBoolean (value)

    getConfiguration ()

    getMain ()

    getParser ()

    getVector (vector, size, none_value=None)

    getXML_FileName ()

    getXML_LineNumber ()

    getXML_Position ()

    is_exe (filename)

    print_warning (msg)

    raise_error (msg)

    startElement (name, attrs)

    which (filename)

class blendervr.console.xml.base.mono (parent, name, attrs)
    Bases: blendervr.console.xml.base.XML

    characters (string)

    endElement (name)

    startElement (name, attrs)

class blendervr.console.xml.base.single (parent, name, attrs)
    Bases: blendervr.console.xml.base.XML

    characters (string)
```

**endElement** (*name*)

**startElement** (*name, attrs*)

#### **blendervr.console.xml.behavior module**

**class** `blendervr.console.xml.behavior.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.reusable.XML`

**characters** (*string*)

**endElement** (*name*)

#### **blendervr.console.xml.blendervr module**

**class** `blendervr.console.xml.blendervr.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

#### **blendervr.console.xml.common\_system module**

**class** `blendervr.console.xml.common_system.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.reusable.XML`

#### **blendervr.console.xml.computer module**

**class** `blendervr.console.xml.computer.container` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

**class** `blendervr.console.xml.computer.main` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

#### **blendervr.console.xml.display module**

**class** `blendervr.console.xml.display.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.reusable.XML`

**characters** (*string*)

**endElement** (*name*)

**class** `blendervr.console.xml.display.graphic_buffer` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.mono`

#### **blendervr.console.xml.plugins module**

**class** `blendervr.console.xml.plugins.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

#### **blendervr.console.xml.processors module**

**class** `blendervr.console.xml.processors.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.single`

**characters** (*string*)

**getConfiguration** ()

#### **blendervr.console.xml.reusable module**

**class** `blendervr.console.xml.reusable.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

**blendervr.console.xml.root module**

**class** `blendervr.console.xml.root.XML` (*parent*)

Bases: `blendervr.console.xml.base.XML`

**getConfiguration** ()

**blendervr.console.xml.screen module**

**class** `blendervr.console.xml.screen.container` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

**class** `blendervr.console.xml.screen.main` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

**blendervr.console.xml.starter module**

**class** `blendervr.console.xml.starter.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.common_system.XML`

**characters** (*string*)

**blendervr.console.xml.system module**

**class** `blendervr.console.xml.system.Library` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.mono`

**class** `blendervr.console.xml.system.XML` (*parent, name, attrs*)

Bases: `blendervr.console.xml.common_system.XML`

**class** `blendervr.console.xml.system.blenderplayer` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.single`

**characters** (*string*)

**class** `blendervr.console.xml.system.daemon` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.single`

**characters** (*string*)

**class** `blendervr.console.xml.system.log` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.mono`

**class** `blendervr.console.xml.system.login` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.mono`

**blendervr.console.xml.user module**

**class** `blendervr.console.xml.user.container` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

**class** `blendervr.console.xml.user.main` (*parent, name, attrs*)

Bases: `blendervr.console.xml.base.XML`

**Module contents**

**class** `blendervr.console.xml.Configure` (*parent, config\_paths, config\_file*)

Bases: `xml.sax.handler.ContentHandler`, `xml.sax.handler.EntityResolver`,  
`blendervr.console.base.Base`

**getConfiguration** ()

**resolveEntity** (*publicID, systemID*)

## Submodules

### blendervr.console.base module

**class** `blendervr.console.base.Base` (*parent*)

Bases: `blendervr.base.Base`

`getConsole()`

`profile`

`quit()`

### blendervr.console.console module

**class** `blendervr.console.console.Console` (*profile\_file*)

Bases: `blendervr.console.logic.console.Logic`, `blendervr.console.qt.console.GUI`

`logger`

`plugins`

`profile`

`quit()`

`start()`

### blendervr.console.exceptions module

**exception** `blendervr.console.exceptions.Main`

Bases: `Exception`

### blendervr.console.profile module

**class** `blendervr.console.profile.Profile` (*configuration\_file*)

Bases: `object`

`appendValue(index, value, write=True)`

`dump()`

`getValue(index)`

`lock(lock)`

`prependValue(index, value, write=True)`

`setDefault(default, node=None, root=None)`

`setValue(index, value, write=True)`

### blendervr.console.screen module

**class** `blendervr.console.screen.Screen` (*screens, name, net\_console*)

Bases: `blendervr.console.base.Base`, `blendervr.console.logic.screen.Logic`,  
`blendervr.console.qt.screen.GUI`

`getName()`

`logger`

`main_logger`

`quit()`

**start ()**

### **blendervr.console.screens module**

**class** `blendervr.console.screens.Screens` (*parent*)

Bases: `blendervr.console.base.Base`, `blendervr.console.logic.screens.Logic`,  
`blendervr.console.qt.screens.GUI`

**quit ()**

**start ()**

### **Module contents**

`blendervr.console.main ()`

`blendervr.console.stripAnchor` (*anchor, path*)

`blendervr.console.unstripAnchor` (*anchor, path*)

### **blendervr.interactor package**

#### **Subpackages**

#### **blendervr.interactor.arc\_ball package**

#### **Submodules**

#### **blendervr.interactor.arc\_ball.console module**

#### **Module contents**

#### **Submodules**

#### **blendervr.interactor.head\_controlled\_navigation module**

#### **blendervr.interactor.landmarks module**

#### **blendervr.interactor.laser module**

#### **blendervr.interactor.object\_chooser module**

#### **blendervr.interactor.reset\_objects module**

#### **blendervr.interactor.viewpoint module**

#### **blendervr.interactor.wavefront\_obj module**

## Module contents

### blendervr.loader package

#### Submodules

#### blendervr.loader.base module

`class blendervr.loader.base.Base` (*parent*)

Bases: `blendervr.base.Base`

`process` (*controller*)

`blendervr.loader.base.main` ()

## Module contents

`class blendervr.loader.Creator` (*logger*)

Bases: `object`

`process` ()

`blendervr.loader.main` ()

### blendervr.player package

#### Subpackages

#### blendervr.player.network package

#### Subpackages

#### blendervr.player.network.synchronizer package

#### Subpackages

#### blendervr.player.network.synchronizer.objects package

#### Submodules

`blendervr.player.network.synchronizer.objects.item_armature_bone` module

`blendervr.player.network.synchronizer.objects.item_armature_channel` module

`blendervr.player.network.synchronizer.objects.item_armature_object` module

`blendervr.player.network.synchronizer.objects.item_base` module

`blendervr.player.network.synchronizer.objects.item_camera` module

**blendervr.player.network.synchronizer.objects.item\_default module**

**blendervr.player.network.synchronizer.objects.item\_font module**

**blendervr.player.network.synchronizer.objects.item\_light module**

**blendervr.player.network.synchronizer.objects.item\_object module**

**blendervr.player.network.synchronizer.objects.item\_root module**

**blendervr.player.network.synchronizer.objects.item\_scene module**

**blendervr.player.network.synchronizer.objects.master module**

**blendervr.player.network.synchronizer.objects.slave module**

**Module contents**

**Module contents**

**Submodules**

**blendervr.player.network.connector module**

**blendervr.player.network.controller module**

**Module contents**

**blendervr.player.screen package**

**Subpackages**

**blendervr.player.screen.hmd package**

**Submodules**

**blendervr.player.screen.hmd.base module**

**blendervr.player.screen.hmd.oculus\_dk2 module**

**Module contents**

## Submodules

**blendervr.player.screen.base** module

**blendervr.player.screen.planovision** module

**blendervr.player.screen.wall** module

## Module contents

### Submodules

**blendervr.player.base** module

**blendervr.player.buffer** module

**class** `blendervr.player.buffer.Buffer`

Bases: `object`

**addPrefix** (*prefix*)

**boolean** (*data=None*)

**command** (*data=None*)

**float** (*data=None*)

**integer** (*data=None*)

**isEmpty** ()

**itemID** (*data=None*)

**matrix\_3x3** (*data=None*)

**matrix\_4x4** (*data=None*)

**size** (*data=None*)

**string** (*data=None*)

**subBuffer** (*data=None*)

**unsigned\_char** (*data=None*)

**vector\_3** (*data=None*)

**vector\_4** (*data=None*)

`blendervr.player.buffer.data_size` (*data\_type*)

**blendervr.player.device** module



**blendervr.player.exceptions module****exception** `blendervr.player.exceptions.Common` (*reason*)Bases: `Exception`**exception** `blendervr.player.exceptions.Controller` (*reason*)Bases: `blendervr.player.exceptions.Common`**exception** `blendervr.player.exceptions.Main` (*reason*)Bases: `blendervr.player.exceptions.Common`**exception** `blendervr.player.exceptions.Processor` (*reason*)Bases: `blendervr.player.exceptions.Common`**exception** `blendervr.player.exceptions.Processor_Invalid_Device` (*reason*)Bases: `blendervr.player.exceptions.Common`**exception** `blendervr.player.exceptions.Processor_Invalid_Device_Method` (*reason*)Bases: `blendervr.player.exceptions.Processor_Invalid_Device`**exception** `blendervr.player.exceptions.Synchronizer` (*reason*)Bases: `blendervr.player.exceptions.Common`**exception** `blendervr.player.exceptions.User` (*reason*)Bases: `blendervr.player.exceptions.Common`**exception** `blendervr.player.exceptions.VirtualEnvironment` (*reason*)Bases: `blendervr.player.exceptions.Common`**blendervr.player.keyboardAndMouse module****blendervr.player.splash module****blendervr.player.user module****Module contents****blendervr.plugins package****Subpackages****blendervr.plugins.oculus\_dk2 package****Subpackages****blendervr.plugins.oculus\_dk2.virtual\_environment package****Submodules****blendervr.plugins.oculus\_dk2.virtual\_environment.user module****Module contents**

### Submodules

`blendervr.plugins.oculus_dk2.xml` module

### Module contents

`blendervr.plugins.osc` package

### Subpackages

`blendervr.plugins.osc.virtual_environment` package

### Submodules

`blendervr.plugins.osc.virtual_environment.base` module

`blendervr.plugins.osc.virtual_environment.object` module

`blendervr.plugins.osc.virtual_environment.objectuser` module

`blendervr.plugins.osc.virtual_environment.user` module

### Module contents

### Submodules

`blendervr.plugins.osc.client` module

`blendervr.plugins.osc.exceptions` module

`blendervr.plugins.osc.msg` module

`blendervr.plugins.osc.xml` module

### Module contents

`blendervr.plugins.vrpn` package

### Subpackages

`blendervr.plugins.vrpn.virtual_environment` package

### Submodules

`blendervr.plugins.vrpn.virtual_environment.analog` module

`blendervr.plugins.vrpn.virtual_environment.button` module

`blendervr.plugins.vrpn.virtual_environment.text` module

`blendervr.plugins.vrpn.virtual_environment.tracker` module

### Module contents

`blendervr.plugins.vrpn.xml` package

### Submodules

`blendervr.plugins.vrpn.xml.analog` module

`blendervr.plugins.vrpn.xml.button` module

`blendervr.plugins.vrpn.xml.sensor` module

`blendervr.plugins.vrpn.xml.text` module

`blendervr.plugins.vrpn.xml.tracker` module

`blendervr.plugins.vrpn.xml.transformation` module

`blendervr.plugins.vrpn.xml.vrpn_base` module

### Module contents

### Module contents

### Submodules

`blendervr.plugins.base` module

`blendervr.plugins.exceptions` module

`blendervr.plugins.xml` module

## Module contents

**blendervr.processor package**

### Submodules

**blendervr.processor.base module**

**blendervr.processor.default module**

## Module contents

**blendervr.tools package**

### Subpackages

**blendervr.tools.gui package**

### Submodules

**blendervr.tools.gui.qt module**

**class** `blendervr.tools.gui.qt.Common` (*owner, profile, profile\_indices*)

Bases: `object`

**moveEvent** (*event*)

**resizeEvent** (*event*)

**setGeometry** ()

**start** ()

**class** `blendervr.tools.gui.qt.Dialog` (*owner, profile, profile\_indices*)

Bases: `blendervr.tools.gui.qt.Common`, `dummy.PyQt4.QDialog`

**cb\_close** ()

**cb\_display\_log\_window** ()

**getMenu** (*name, shortcut*)

**setMenuAction** (*menu\_action*)

**start** ()

## Module contents

`blendervr.tools.gui.insertWidgetInsideAnother` (*parent, child*)

`blendervr.tools.gui.load` (*ui\_file, parent\_widget*)

### Submodules

**blendervr.tools.connector module**

**class** `blendervr.tools.connector.Client` (*controller, module, screen\_name*)

Bases: `blendervr.tools.connector.Common`

**class** `blendervr.tools.connector.Common`

Bases: `object`

**BUFFER\_LEN = 1024**

**SIZE\_LEN = 10**

**close()**

**fileno()**

**getClient()**

**run()**

**send** (*command, argument=''*)

**setCallback** (*callback*)

**setClient** (*client, callback=None*)

**setWait** (*block*)

**class** `blendervr.tools.connector.Server` (*client*)

Bases: `blendervr.tools.connector.Common`

**getClientInformation()**

**blendervr.tools.logger module** Handle all the errors, warnings and debug info

**class** `blendervr.tools.logger.Console` (*msg='Console logger: '*)

Bases: `object`

**write** (*\*messages*)

**class** `blendervr.tools.logger.Logger` (*name*)

Bases: `logging.Logger`

**addLoginWindow** (*login\_window, addName=False*)

**getVerbsities()**

**get\_position()**

**log\_position()**

**log\_traceback** (*error*)

**setLevel** (*verbosity*)

`blendervr.tools.logger.getLogger` (*name*)

**blendervr.tools.profiler module****blendervr.tools.protocol module**

`blendervr.tools.protocol.composeMessage` (*command, argument=''*)

`blendervr.tools.protocol.decomposeMessage` (*message*)

**Module contents**`blendervr.tools.getLibsPath()`

The path for the external libs

`blendervr.tools.getModulePath()`

The path of the current module

`blendervr.tools.getRootPath()`

The root path of the project

**Submodules****blendervr.base module****class** `blendervr.base.Base` (*parent*)

Bases: object

Main struct for the blender vr environment

**Parameters** *parent* (*module or class*) –**getMainRunningModule** ()**Return type** module**getParent** ()

Returns the module that inherits the Base class

**Return type** module or class**logger**

Logger of the main running module

**Return type** callback function**quit** ()**blendervr.exceptions module****exception** `blendervr.exceptions.PluginError` (*hasToClear=True, msg=''*)

Bases: Exception

**hasToClear** ()**Module contents**

Main module of the BlenderVR application

`blendervr.is_console` ()

Check if it is in console mode.

**Return type** bool`blendervr.is_creating_loader` ()

Check if BPY is available.

**Return type** bool`blendervr.is_virtual_environment` ()

Check if the Blender Game Engine is available.

**Return type** bool

`blendervr.main()`

`blendervr.run()`

---

## Indices and tables

---

- [genindex](#)
- [modindex](#)
- [search](#)



**b**

- blendervr, 28
- blendervr.base, 28
- blendervr.console, 19
  - blendervr.console.base, 18
  - blendervr.console.console, 18
  - blendervr.console.exceptions, 18
  - blendervr.console.gui, 10
    - blendervr.console.gui.console, 9
    - blendervr.console.gui.logger, 9
    - blendervr.console.gui.options, 9
    - blendervr.console.gui.screen, 10
    - blendervr.console.gui.screens, 10
  - blendervr.console.logic, 12
    - blendervr.console.logic.console, 10
    - blendervr.console.logic.file\_name, 11
    - blendervr.console.logic.screen, 11
    - blendervr.console.logic.screens, 11
  - blendervr.console.profile, 18
  - blendervr.console.qt, 14
    - blendervr.console.qt.console, 12
    - blendervr.console.qt.logger, 13
    - blendervr.console.qt.options, 13
    - blendervr.console.qt.screen, 13
    - blendervr.console.qt.screens, 13
    - blendervr.console.qt.tools, 14
  - blendervr.console.screen, 18
  - blendervr.console.screens, 19
  - blendervr.console.xml, 17
    - blendervr.console.xml.base, 15
    - blendervr.console.xml.behavior, 16
    - blendervr.console.xml.blendervr, 16
    - blendervr.console.xml.common\_system, 16
    - blendervr.console.xml.computer, 16
    - blendervr.console.xml.device, 15
      - blendervr.console.xml.device.base, 14
      - blendervr.console.xml.device.hmd, 14
        - blendervr.console.xml.device.hmd.base, 14
        - blendervr.console.xml.device.hmd.oculus\_dk2, 14
      - blendervr.console.xml.device.planovision, 14
      - blendervr.console.xml.device.screen, 15
      - blendervr.console.xml.device.wall, 15
    - blendervr.console.xml.display, 16
    - blendervr.console.xml.plugins, 16
    - blendervr.console.xml.processors, 16
    - blendervr.console.xml.reusable, 16
    - blendervr.console.xml.root, 17
    - blendervr.console.xml.screen, 17
    - blendervr.console.xml.starter, 17
    - blendervr.console.xml.system, 17
    - blendervr.console.xml.user, 17
- blendervr.exceptions, 28
- blendervr.loader, 20
  - blendervr.loader.base, 20
- blendervr.player.buffer, 22
- blendervr.player.exceptions, 23
- blendervr.tools, 28
  - blendervr.tools.connector, 27
  - blendervr.tools.gui, 26
    - blendervr.tools.gui.qt, 26
  - blendervr.tools.logger, 27
  - blendervr.tools.profiler, 27
  - blendervr.tools.protocol, 27

**d**

- daemon, 8

**u**

- update\_loader, 8

## A

adapt\_simulation\_files\_to\_screen()  
     (blendervr.console.logic.screen.Logic method),  
     11  
 adapt\_simulation\_files\_to\_screen()  
     (blendervr.console.logic.screens.Logic method), 11  
 addListenTo() (blendervr.console.qt.console.GUI method), 12  
 addLoginWindow() (blendervr.tools.logger.Logger method),  
     27  
 addPrefix() (blendervr.player.buffer.Buffer method), 22  
 addTimeout() (blendervr.console.qt.console.GUI method), 12  
 appendValue() (blendervr.console.profile.Profile method), 18  
 ask\_blender\_player\_to\_quit()  
     (blendervr.console.logic.screen.Logic method),  
     11

## B

Base (class in blendervr.base), 28  
 Base (class in blendervr.console.base), 18  
 Base (class in blendervr.console.xml.device.base), 14  
 Base (class in blendervr.loader.base), 20  
 blenderplayer (class in blendervr.console.xml.system), 17  
 blendervr (module), 28  
 blendervr.base (module), 28  
 blendervr.console (module), 19  
 blendervr.console.base (module), 18  
 blendervr.console.console (module), 18  
 blendervr.console.exceptions (module), 18  
 blendervr.console.gui (module), 10  
 blendervr.console.gui.console (module), 9  
 blendervr.console.gui.logger (module), 9  
 blendervr.console.gui.options (module), 9  
 blendervr.console.gui.screen (module), 10  
 blendervr.console.gui.screens (module), 10  
 blendervr.console.logic (module), 12  
 blendervr.console.logic.console (module), 10  
 blendervr.console.logic.file\_name (module), 11  
 blendervr.console.logic.screen (module), 11  
 blendervr.console.logic.screens (module), 11  
 blendervr.console.profile (module), 18  
 blendervr.console.qt (module), 14  
 blendervr.console.qt.console (module), 12  
 blendervr.console.qt.logger (module), 13  
 blendervr.console.qt.options (module), 13  
 blendervr.console.qt.screen (module), 13  
 blendervr.console.qt.screens (module), 13  
 blendervr.console.qt.tools (module), 14  
 blendervr.console.screen (module), 18  
 blendervr.console.screens (module), 19  
 blendervr.console.xml (module), 17  
 blendervr.console.xml.base (module), 15  
 blendervr.console.xml.behavior (module), 16  
 blendervr.console.xml.blendervr (module), 16  
 blendervr.console.xml.common\_system (module), 16  
 blendervr.console.xml.computer (module), 16  
 blendervr.console.xml.device (module), 15  
 blendervr.console.xml.device.base (module), 14  
 blendervr.console.xml.device.hmd (module), 14  
 blendervr.console.xml.device.hmd.base (module), 14  
 blendervr.console.xml.device.hmd.oculus\_dk2 (module), 14  
 blendervr.console.xml.device.planovision (module), 14  
 blendervr.console.xml.device.screen (module), 15  
 blendervr.console.xml.device.wall (module), 15  
 blendervr.console.xml.display (module), 16  
 blendervr.console.xml.plugins (module), 16  
 blendervr.console.xml.processors (module), 16  
 blendervr.console.xml.reusable (module), 16  
 blendervr.console.xml.root (module), 17  
 blendervr.console.xml.screen (module), 17  
 blendervr.console.xml.starter (module), 17  
 blendervr.console.xml.system (module), 17  
 blendervr.console.xml.user (module), 17  
 blendervr.exceptions (module), 28  
 blendervr.loader (module), 20  
 blendervr.loader.base (module), 20  
 blendervr.player.buffer (module), 22  
 blendervr.player.exceptions (module), 23  
 blendervr.tools (module), 28  
 blendervr.tools.connector (module), 27  
 blendervr.tools.gui (module), 26  
 blendervr.tools.gui.qt (module), 26

- blendervr.tools.logger (module), 27  
 blendervr.tools.profiler (module), 27  
 blendervr.tools.protocol (module), 27  
 BlenderVR\_state\_changed() (blendervr.console.qt.options.GUI method), 13  
 boolean() (blendervr.player.buffer.Buffer method), 22  
 Buffer (class in blendervr.player.buffer), 22  
 BUFFER\_LEN (blendervr.tools.connector.Common attribute), 27
- ## C
- cb\_add\_configuration\_path() (blendervr.console.qt.console.GUI method), 12  
 cb\_close() (blendervr.console.qt.console.GUI method), 12  
 cb\_close() (blendervr.tools.gui.qt.Dialog method), 26  
 cb\_debug\_daemon() (blendervr.console.qt.options.GUI method), 13  
 cb\_debug\_processor() (blendervr.console.qt.options.GUI method), 13  
 cb\_display\_log\_window() (blendervr.tools.gui.qt.Dialog method), 26  
 cb\_executables() (blendervr.console.qt.options.GUI method), 13  
 cb\_load\_configuration\_file() (blendervr.console.qt.console.GUI method), 12  
 cb\_processor\_window() (blendervr.console.qt.console.GUI method), 12  
 cb\_reload\_processor() (blendervr.console.qt.options.GUI method), 13  
 cb\_remove\_configuration\_path() (blendervr.console.qt.console.GUI method), 12  
 cb\_restart\_daemons() (blendervr.console.qt.options.GUI method), 13  
 cb\_set\_blender\_file() (blendervr.console.qt.console.GUI method), 12  
 cb\_set\_configuration\_file() (blendervr.console.qt.console.GUI method), 12  
 cb\_set\_current\_tab() (blendervr.console.qt.console.GUI method), 12  
 cb\_set\_link\_processor\_to\_blender() (blendervr.console.qt.console.GUI method), 12  
 cb\_set\_log\_level() (blendervr.console.qt.logger.Logger method), 13  
 cb\_set\_processor\_file() (blendervr.console.qt.console.GUI method), 12  
 cb\_set\_screen\_set() (blendervr.console.qt.console.GUI method), 12  
 cb\_start() (blendervr.console.qt.console.GUI method), 12  
 cb\_stop() (blendervr.console.qt.console.GUI method), 12  
 cb\_toggle\_stderr\_state() (blendervr.console.qt.screen.GUI method), 13  
 cb\_toggle\_stdout\_state() (blendervr.console.qt.screen.GUI method), 13  
 cb\_update\_liste\_paths() (blendervr.console.qt.console.GUI method), 12  
 characters() (blendervr.console.xml.base.mono method), 15  
 characters() (blendervr.console.xml.base.single method), 15  
 characters() (blendervr.console.xml.behavior.XML method), 16  
 characters() (blendervr.console.xml.device.screen.Screen method), 15  
 characters() (blendervr.console.xml.display.XML method), 16  
 characters() (blendervr.console.xml.processors.XML method), 16  
 characters() (blendervr.console.xml.starter.XML method), 17  
 characters() (blendervr.console.xml.system.blenderplayer method), 17  
 characters() (blendervr.console.xml.system.daemon method), 17  
 clear() (blendervr.console.gui.logger.Logger method), 9  
 clear() (blendervr.console.qt.logger.Logger method), 13  
 Client (class in blendervr.tools.connector), 27  
 close() (blendervr.console.qt.options.GUI method), 13  
 close() (blendervr.console.qt.screen.GUI method), 13  
 close() (blendervr.tools.connector.Common method), 27  
 close\_all() (blendervr.console.qt.screens.GUI method), 13  
 command() (blendervr.player.buffer.Buffer method), 22  
 Common, 23  
 Common (class in blendervr.tools.connector), 27  
 Common (class in blendervr.tools.gui.qt), 26  
 compile\_BC() (blendervr.console.logic.console.Logic method), 10  
 composeMessage() (in module blendervr.tools.protocol), 27  
 Configure (class in blendervr.console.xml), 17  
 Console (class in blendervr.console.console), 18  
 Console (class in blendervr.tools.logger), 27  
 container (class in blendervr.console.xml.computer), 16  
 container (class in blendervr.console.xml.screen), 17  
 container (class in blendervr.console.xml.user), 17  
 Controller, 23  
 Creator (class in blendervr.loader), 20
- ## D
- daemon (class in blendervr.console.xml.system), 17  
 Daemon (class in daemon), 8  
 daemon (module), 8  
 daemon\_is\_running() (blendervr.console.logic.screen.Logic method), 11  
 data\_size() (in module blendervr.player.buffer), 22  
 decomposeMessage() (in module blendervr.tools.protocol), 27  
 Device (class in blendervr.console.xml.device.hmd.base), 14  
 Device (class in blendervr.console.xml.device.hmd.oculus\_dk2), 14  
 Device (class in blendervr.console.xml.device.planovision), 14  
 Device (class in blendervr.console.xml.device.wall), 15  
 Device() (in module blendervr.console.xml.device.hmd), 14

- Dialog (class in `blendervr.tools.gui.qt`), 26
- `display_screen_sets()` (`blendervr.console.gui.console.GUI` method), 9
- `display_screen_sets()` (`blendervr.console.qt.console.GUI` method), 12
- `dump()` (`blendervr.console.profile.Profile` method), 18
- ## E
- `endElement()` (`blendervr.console.xml.base.mono` method), 15
- `endElement()` (`blendervr.console.xml.base.single` method), 15
- `endElement()` (`blendervr.console.xml.base.XML` method), 15
- `endElement()` (`blendervr.console.xml.behavior.XML` method), 16
- `endElement()` (`blendervr.console.xml.device.screen.Screen` method), 15
- `endElement()` (`blendervr.console.xml.display.XML` method), 16
- ## F
- `FileName` (class in `blendervr.console.logic.file_name`), 11
- `fileno()` (`blendervr.tools.connector.Common` method), 27
- `float()` (`blendervr.player.buffer.Buffer` method), 22
- `flush()` (`blendervr.console.gui.logger.Logger` method), 9
- ## G
- `get_blender_player_state()` (`blendervr.console.logic.console.Logic` method), 10
- `get_blender_player_state()` (`blendervr.console.logic.screen.Logic` method), 11
- `get_position()` (`blendervr.tools.logger.Logger` method), 27
- `getBoolean()` (`blendervr.console.xml.base.XML` method), 15
- `getClient()` (`blendervr.tools.connector.Common` method), 27
- `getClientInformation()` (`blendervr.tools.connector.Server` method), 27
- `getColors()` (`blendervr.console.qt.tools.MessagesColors` method), 14
- `getConfiguration()` (`blendervr.console.xml.base.XML` method), 15
- `getConfiguration()` (`blendervr.console.xml.Configure` method), 17
- `getConfiguration()` (`blendervr.console.xml.processors.XML` method), 16
- `getConfiguration()` (`blendervr.console.xml.root.XML` method), 17
- `getConsole()` (`blendervr.console.base.Base` method), 18
- `getDevice()` (in module `blendervr.console.xml.device`), 15
- `getFrame()` (`blendervr.console.gui.screen.GUI` method), 10
- `getHostname()` (`blendervr.console.logic.screen.Logic` method), 11
- `getLibsPath()` (in module `blendervr.tools`), 28
- `getLogger()` (in module `blendervr.tools.logger`), 27
- `getMain()` (`blendervr.console.xml.base.XML` method), 15
- `getMainRunningModule()` (`blendervr.base.Base` method), 28
- `getMaster()` (`blendervr.console.logic.screens.Logic` method), 11
- `getMenu()` (`blendervr.console.qt.options.GUI` method), 13
- `getmenu()` (`blendervr.console.qt.options.GUI` method), 13
- `getMenu()` (`blendervr.console.qt.screen.GUI` method), 13
- `getMenu()` (`blendervr.tools.gui.qt.Dialog` method), 26
- `getModulePath()` (in module `blendervr.tools`), 28
- `getName()` (`blendervr.console.screen.Screen` method), 18
- `getParent()` (`blendervr.base.Base` method), 28
- `getParser()` (`blendervr.console.xml.base.XML` method), 15
- `getRootPath()` (in module `blendervr.tools`), 28
- `getScreen()` (`blendervr.console.logic.screens.Logic` method), 11
- `getScreensNumber()` (`blendervr.console.logic.screens.Logic` method), 11
- `getStates()` (`blendervr.console.logic.screens.Logic` method), 11
- `getValue()` (`blendervr.console.profile.Profile` method), 18
- `getVector()` (`blendervr.console.xml.base.XML` method), 15
- `getVerboisities()` (`blendervr.tools.logger.Logger` method), 27
- `getWindow()` (`blendervr.console.qt.console.GUI` method), 12
- `getXML_FileName()` (`blendervr.console.xml.base.XML` method), 15
- `getXML_LineNumber()` (`blendervr.console.xml.base.XML` method), 15
- `getXML_Position()` (`blendervr.console.xml.base.XML` method), 15
- `graphic_buffer` (class in `blendervr.console.xml.display`), 16
- `GUI` (class in `blendervr.console.gui.console`), 9
- `GUI` (class in `blendervr.console.gui.options`), 9
- `GUI` (class in `blendervr.console.gui.screen`), 10
- `GUI` (class in `blendervr.console.gui.screens`), 10
- `GUI` (class in `blendervr.console.qt.console`), 12
- `GUI` (class in `blendervr.console.qt.options`), 13
- `GUI` (class in `blendervr.console.qt.screen`), 13
- `GUI` (class in `blendervr.console.qt.screens`), 13
- ## H
- `hasToClear()` (`blendervr.exceptions.PluginError` method), 28
- ## I
- `insertWidgetInsideAnother()` (in module `blendervr.tools.gui`), 26
- `integer()` (`blendervr.player.buffer.Buffer` method), 22
- `is_console()` (in module `blendervr`), 28
- `is_creating_loader()` (in module `blendervr`), 28
- `is_exe()` (`blendervr.console.xml.base.XML` method), 15
- `is_log_window_opened()` (`blendervr.console.gui.screen.GUI` method), 10
- `is_log_window_opened()` (`blendervr.console.qt.screen.GUI` method), 13
- `is_master()` (`blendervr.console.logic.screen.Logic` method), 11

- is\_options\_window\_opened()  
(blendervr.console.gui.options.GUI method), 9
- is\_virtual\_environment() (in module blendervr), 28
- isEmpty() (blendervr.player.buffer.Buffer method), 22
- itemID() (blendervr.player.buffer.Buffer method), 22
- ## L
- Library (class in blendervr.console.xml.system), 17
- load() (in module blendervr.tools.gui), 26
- load\_configuration\_file() (blendervr.console.logic.console.Logic method), 10
- lock() (blendervr.console.profile.Profile method), 18
- log (class in blendervr.console.xml.system), 17
- log\_position() (blendervr.tools.logger.Logger method), 27
- log\_traceback() (blendervr.tools.logger.Logger method), 27
- logger (blendervr.base.Base attribute), 28
- logger (blendervr.console.console.Console attribute), 18
- logger (blendervr.console.screen.Screen attribute), 18
- Logger (class in blendervr.console.gui.logger), 9
- Logger (class in blendervr.console.qt.logger), 13
- Logger (class in blendervr.tools.logger), 27
- Logic (class in blendervr.console.logic.console), 10
- Logic (class in blendervr.console.logic.screen), 11
- Logic (class in blendervr.console.logic.screens), 11
- login (class in blendervr.console.xml.system), 17
- ## M
- Main, 18, 23
- main (class in blendervr.console.xml.computer), 16
- main (class in blendervr.console.xml.screen), 17
- main (class in blendervr.console.xml.user), 17
- main() (blendervr.console.gui.console.GUI method), 9
- main() (blendervr.console.qt.console.GUI method), 12
- main() (daemon.Daemon method), 8
- main() (in module blendervr), 29
- main() (in module blendervr.console), 19
- main() (in module blendervr.loader), 20
- main() (in module blendervr.loader.base), 20
- main() (in module daemon), 8
- main\_logger (blendervr.console.screen.Screen attribute), 18
- MainWindow (class in blendervr.console.qt.console), 12
- matrix\_3x3() (blendervr.player.buffer.Buffer method), 22
- matrix\_4x4() (blendervr.player.buffer.Buffer method), 22
- MessagesColors (class in blendervr.console.qt.tools), 14
- mono (class in blendervr.console.xml.base), 15
- moveEvent() (blendervr.tools.gui.qt.Common method), 26
- ## P
- PluginError, 28
- plugins (blendervr.console.console.Console attribute), 18
- prependValue() (blendervr.console.profile.Profile method), 18
- print\_warning() (blendervr.console.xml.base.XML method), 15
- process() (blendervr.loader.base.Base method), 20
- process() (blendervr.loader.Creator method), 20
- processCommand() (daemon.Daemon method), 8
- Processor, 23
- Processor\_Invalid\_Device, 23
- Processor\_Invalid\_Device\_Method, 23
- profile (blendervr.console.base.Base attribute), 18
- profile (blendervr.console.console.Console attribute), 18
- Profile (class in blendervr.console.profile), 18
- ## Q
- quit() (blendervr.base.Base method), 28
- quit() (blendervr.console.base.Base method), 18
- quit() (blendervr.console.console.Console method), 18
- quit() (blendervr.console.gui.console.GUI method), 9
- quit() (blendervr.console.gui.logger.Logger method), 9
- quit() (blendervr.console.gui.options.GUI method), 9
- quit() (blendervr.console.gui.screen.GUI method), 10
- quit() (blendervr.console.gui.screens.GUI method), 10
- quit() (blendervr.console.logic.console.Logic method), 10
- quit() (blendervr.console.logic.screen.Logic method), 11
- quit() (blendervr.console.logic.screens.Logic method), 11
- quit() (blendervr.console.qt.console.GUI method), 12
- quit() (blendervr.console.qt.options.GUI method), 13
- quit() (blendervr.console.qt.screen.GUI method), 13
- quit() (blendervr.console.qt.screens.GUI method), 13
- quit() (blendervr.console.screen.Screen method), 18
- quit() (blendervr.console.screens.Screens method), 19
- quit() (in module blendervr.console.qt.console), 12
- ## R
- raise\_error() (blendervr.console.xml.base.XML method), 15
- receivedFromVirtualEnvironment()  
(blendervr.console.logic.console.Logic method), 10
- removeListenTo() (blendervr.console.qt.console.GUI method), 12
- resizeEvent() (blendervr.tools.gui.qt.Common method), 26
- resolveEntity() (blendervr.console.xml.Configure method), 17
- restartDaemon() (blendervr.console.logic.screen.Logic method), 11
- run() (blendervr.tools.connector.Common method), 27
- run() (in module blendervr), 29
- ## S
- Screen (class in blendervr.console.screen), 18
- Screen (class in blendervr.console.xml.device.screen), 15
- Screens (class in blendervr.console.screens), 19
- send() (blendervr.tools.connector.Common method), 27
- send\_to\_blender\_player() (blendervr.console.logic.screen.Logic method), 11
- send\_to\_blender\_player() (blendervr.console.logic.screens.Logic method), 11
- sendToVirtualEnvironment() (blendervr.console.logic.console.Logic method), 10



- Server (class in `blendervr.tools.connector`), 27
- `set_BlenderVR_state()` (`blendervr.console.logic.screen.Logic` method), 11
- `set_log_level()` (`blendervr.console.gui.logger.Logger` method), 9
- `set_screen_set()` (`blendervr.console.logic.console.Logic` method), 10
- `set_screens()` (`blendervr.console.logic.screens.Logic` method), 11
- `set_stream_state()` (`blendervr.console.gui.logger.Logger` method), 9
- `setCallback()` (`blendervr.tools.connector.Common` method), 27
- `setClient()` (`blendervr.tools.connector.Common` method), 27
- `setConfiguration()` (`blendervr.console.logic.screen.Logic` method), 11
- `setDefault()` (`blendervr.console.profile.Profile` method), 18
- `setGeometry()` (`blendervr.tools.gui.qt.Common` method), 26
- `setHierarchy()` (`blendervr.console.logic.screen.Logic` method), 11
- `setLevel()` (`blendervr.tools.logger.Logger` method), 27
- `setMenuAction()` (`blendervr.tools.gui.qt.Dialog` method), 26
- `setNetworkClient()` (`blendervr.console.logic.screen.Logic` method), 11
- `setValue()` (`blendervr.console.profile.Profile` method), 18
- `setWait()` (`blendervr.tools.connector.Common` method), 27
- `show()` (`blendervr.console.gui.screens.GUI` method), 10
- `single` (class in `blendervr.console.xml.base`), 15
- `size()` (`blendervr.player.buffer.Buffer` method), 22
- `SIZE_LEN` (`blendervr.tools.connector.Common` attribute), 27
- `start()` (`blendervr.console.console.Console` method), 18
- `start()` (`blendervr.console.gui.console.GUI` method), 9
- `start()` (`blendervr.console.gui.logger.Logger` method), 9
- `start()` (`blendervr.console.gui.options.GUI` method), 9
- `start()` (`blendervr.console.gui.screen.GUI` method), 10
- `start()` (`blendervr.console.gui.screens.GUI` method), 10
- `start()` (`blendervr.console.logic.console.Logic` method), 10
- `start()` (`blendervr.console.logic.screen.Logic` method), 11
- `start()` (`blendervr.console.logic.screens.Logic` method), 11
- `start()` (`blendervr.console.qt.console.GUI` method), 12
- `start()` (`blendervr.console.qt.options.GUI` method), 13
- `start()` (`blendervr.console.qt.screen.GUI` method), 13
- `start()` (`blendervr.console.qt.screens.GUI` method), 13
- `start()` (`blendervr.console.screen.Screen` method), 18
- `start()` (`blendervr.console.screens.Screens` method), 19
- `start()` (`blendervr.tools.gui.qt.Common` method), 26
- `start()` (`blendervr.tools.gui.qt.Dialog` method), 26
- `start_simulation()` (`blendervr.console.logic.console.Logic` method), 10
- `start_simulation()` (`blendervr.console.logic.screens.Logic` method), 11
- `startElement()` (`blendervr.console.xml.base.mono` method), 15
- `startElement()` (`blendervr.console.xml.base.single` method), 16
- `startElement()` (`blendervr.console.xml.base.XML` method), 15
- `stop_simulation()` (`blendervr.console.logic.console.Logic` method), 10
- `stop_simulation()` (`blendervr.console.logic.screens.Logic` method), 11
- `string()` (`blendervr.player.buffer.Buffer` method), 22
- `strip()` (`blendervr.console.logic.file_name.FileName` method), 11
- `stripAnchor()` (in module `blendervr.console`), 19
- `subBuffer()` (`blendervr.player.buffer.Buffer` method), 22
- Synchronizer, 23
- ## U
- `unsigned_char()` (`blendervr.player.buffer.Buffer` method), 22
- `unstrip()` (`blendervr.console.logic.file_name.FileName` method), 11
- `unstripAnchor()` (in module `blendervr.console`), 19
- `update_gui()` (`blendervr.console.gui.screens.GUI` method), 10
- `update_gui()` (`blendervr.console.qt.screens.GUI` method), 13
- `update_loader` (module), 8
- `update_processor()` (`blendervr.console.qt.console.GUI` method), 12
- `update_user_files()` (`blendervr.console.logic.console.Logic` method), 10
- `updateStatus()` (`blendervr.console.qt.console.GUI` method), 12
- User, 23
- ## V
- `vector_3()` (`blendervr.player.buffer.Buffer` method), 22
- `vector_4()` (`blendervr.player.buffer.Buffer` method), 22
- VirtualEnvironment, 23
- ## W
- `which()` (`blendervr.console.xml.base.XML` method), 15
- `write()` (`blendervr.console.gui.logger.Logger` method), 9
- `write()` (`blendervr.tools.logger.Console` method), 27
- `write()` (`daemon.Daemon` method), 8
- ## X
- XML (class in `blendervr.console.xml.base`), 15
- XML (class in `blendervr.console.xml.behavior`), 16
- XML (class in `blendervr.console.xml.blendervr`), 16
- XML (class in `blendervr.console.xml.common_system`), 16
- XML (class in `blendervr.console.xml.display`), 16
- XML (class in `blendervr.console.xml.plugins`), 16
- XML (class in `blendervr.console.xml.processors`), 16
- XML (class in `blendervr.console.xml.reusable`), 16
- XML (class in `blendervr.console.xml.root`), 17
- XML (class in `blendervr.console.xml.starter`), 17
- XML (class in `blendervr.console.xml.system`), 17