UniOSC Manual

vers. 1.7

monoflow

new media experience

Contents

Chapter 1

Introduction

- **UniOSC** is a tool to easy create Unity applications which can be controlled by hard/software that uses the OSC protocol for communication. OSC is a protocol for distributed systems that is mainly used in the music industry and is often used as an alternative to MIDI. If you need more information about OSC please visit http://opensoundcontrol.org/ . For the OSC communication UniOSC uses a modified version of the OSCsharp library. You can use UniOSC to send/receive OSC messages to/from other devices that are connected via Wi-Fi or create your own GUI-app for controlling another Unity application via OSC but it is strongly recommended to use a third party software like TouchOSC for creating the GUI part. For more info about **TouchOSC** please visit http://hexler.net
- **UniOSC** works in play- and edit-mode. That means you can check/setup your system without entering the play mode of Unity. You could also use it to remote control the Editor.

Chapter 2

Installation

1. Import the UniOSC package from the Assetstore. You should now have a folder named UniOSC with the following structure in your Unity project:

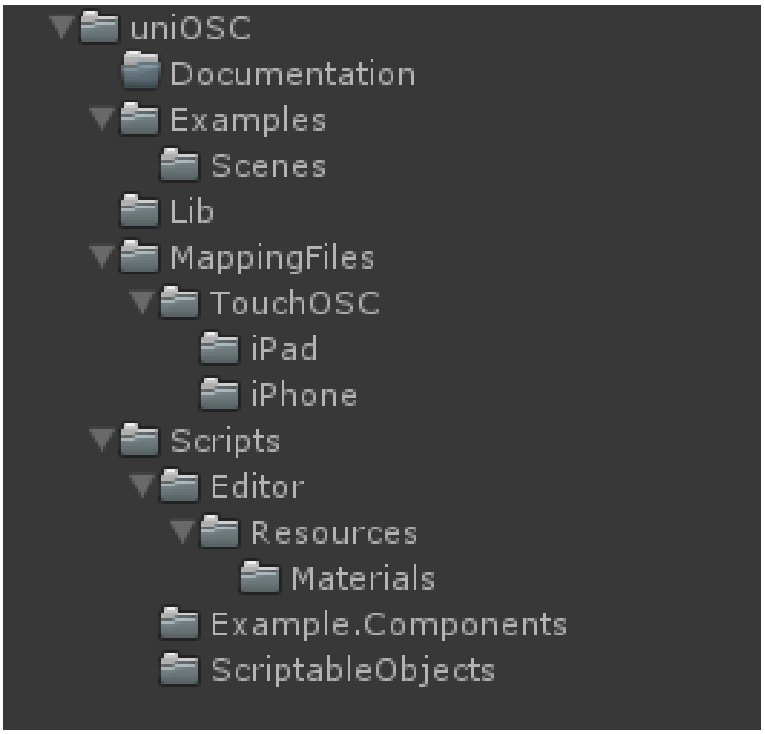


Figure 2.1: package structure

   - Examples: UniOSC comes with a demo scene to show all components in a ready setup. For your mobile device we provide the UniOSC.Mobile scene.
   - Lib: Here is the OSCsharp dll located
   - MappingFiles: Preconfigured mapping files for TouchOSC
   - Scripts: All the c# code

2. After installation you should see a menu item under ‘ Window/UniOSC’. If there is no UniOSC menu item you have to reimport the unitypackage or close & reopen Unity.

3. You should change the script execution order for the UniOSCConnection class in order to prevent problems with auto connecting UniOSC components at start time. Go to “Edit/Project Settings/Script Execution Order” and set the value for the UniOSCConnection class to a negative value so it is executed before other scripts.

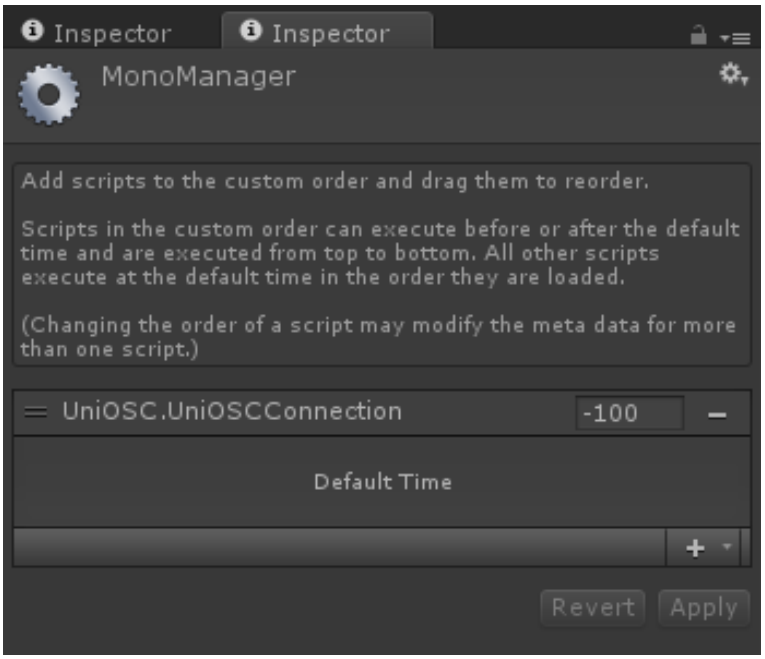


Figure 2.2: Script Execution Order

4. For a quick test with a mobile device it is recommend getting the TouchOSC app. Or you make an app from the UniOSC.Mobile scene and run it on your device.

Chapter 3

UniOSC main components

3.1 OSCEditor

The OSC Editor is a tool to administrate your OSC setup in an easy and visual way and to speed up your workflow. You have access to the main components and tools to create a working OSC setup. It is also very useful for tracing the OSC data flow. For more info please look at UniOSC Editor Interface.

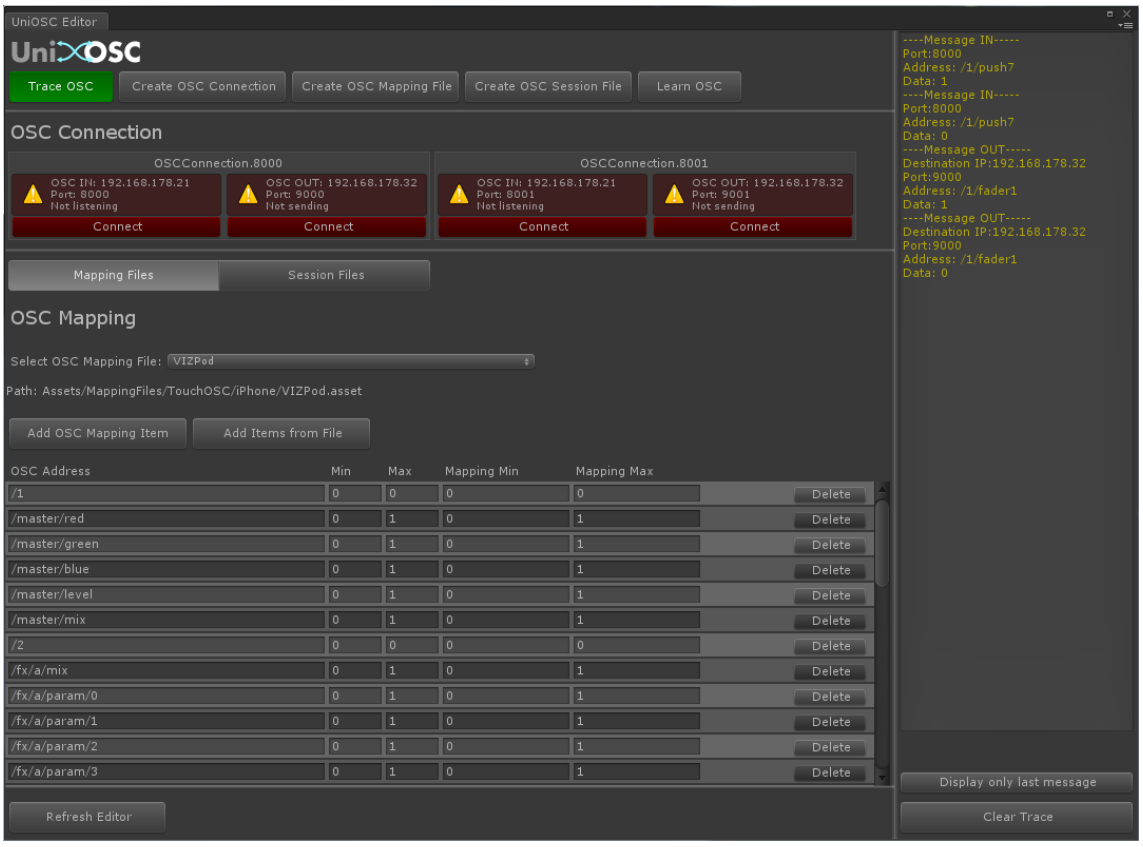


Figure 3.1: OSCEditor

3.2 OSCConnection

- The OSCConnection is a component that handles all the network related tasks of the OSC communication. You can start listening on a local port for receiving incoming OSC data or open an OSC Out connection ready to send data to an IP address on a given port. Depending on your transmission type you can specify the IP addresses. The inspector gives you visual feedback if you use not a valid address to make the setup as easy as possible.

- Connect On Start: If you select this option the connection is established automatically when your Unity app starts or when you enter the play mode in the editor.

- OSC IN: If you use the Unicast mode you can only specify the port the connection is listening on as in this mode the local IP address is used. In Multicast mode you specify the multicast group address.(see below)

- OSC OUT: You have to specify a port , IP address and the transmission type for outgoing OSC data. When using the Broadcast mode the standard broadcast address is used.

- Transmission Type :

  - **Unicast** (default) : one-to-one transmission.
    You only specify a single IP address for sending or listening to.

  - **Multicast**: one-to-many transmission.
    With multicast, the message is sent to a multicast address, and the network delivers it only to those hosts that have indicated that they want to receive messages sent on that address. A valid multicast address lies in the range between 224.0.0.0 – 239.255.255.255 .
    The range of addresses between 224.0.0.0 & 224.0.0.255, inclusive, is reserved for the use of routing protocols and other low-level topology discovery or maintenance protocols, such as gateway discovery and group membership reporting and should not be used due the special purpose they are destined to.
    `iana IPv4 Multicast Address Space Registry`

  - **Broadcast**: one-to-all transmission.
    In this mode you use the fixed broadcast address 255.255.255.255 to send your message to all hosts on the network. Regardless of the OSC IN transmission type of your host it will receive the broadcast message if the port matches. Broadcasting is the most general communication method, and is also the most intensive in the sense that a large number of messages are required.

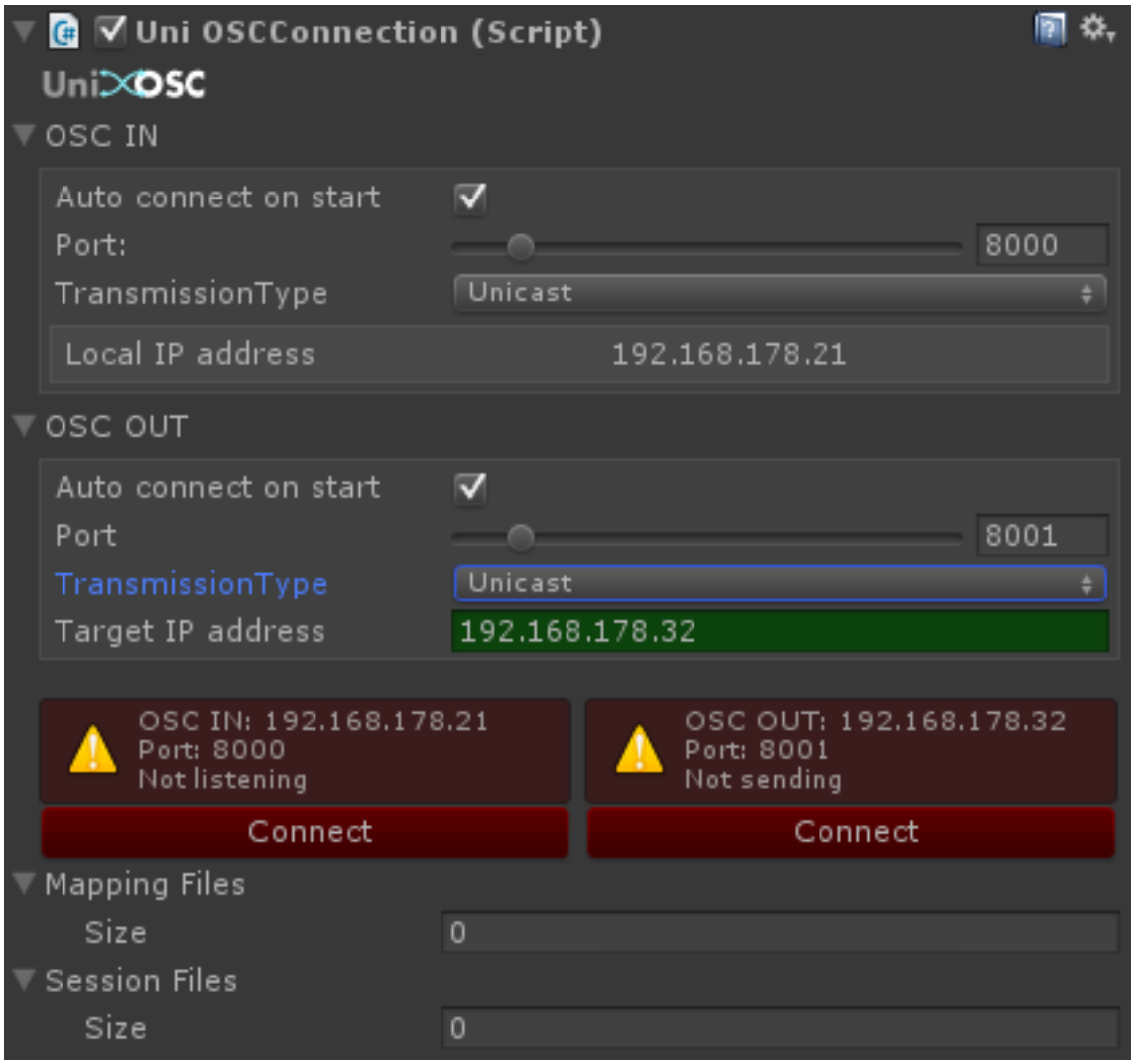


Figure 3.2: OSCConnection

3.3 OSC Mapping file

- The data of an OSC message is defined by your OSC sending application and so sometimes out of your control. On the other side these values don't comply with your workflow. It is useful to have some sort of transformation of the data before it gets handled by your GameObject scripts. In this case you can create a mapping file that you can attach to an OSC connection.
- The mapping file acts as a filter so the data of all the messages that are included in the mapping file are mapped to new values before they pass through to the Unity GameObjects.
- A mapping file is a Scriptable Object asset that is located in your Assets folder so you can create one and copy it to several Unity projects.
- For every OSC address you want to remap the data you have to create a mapping item entry in the mapping file.
- UniOSC comes with some preconfigured mappings files to remap the default layouts that are included with TouchOSC. They are located in 'UniOSC/MappingFiles'.

3.4 OSCMappingItem

- A OSC message consists of an OSC Address Pattern followed by zero or more OSC Arguments that contain the data in form of different data types .(For more info please look at the OSC Specification.)
- To map the data part of the message you first specify the range of the data when it arrives (min/max) and the range to what it should be mapped (MapMin/MapMax).
- Normally the data comes in a normalized range from 0 to 1, or -1 to 1. It's always a good practice to send OSC data in this way when you create your own GUI app. You can map the values afterwards with the help of a mapping file or remap the data in your OSCEventTarget components.

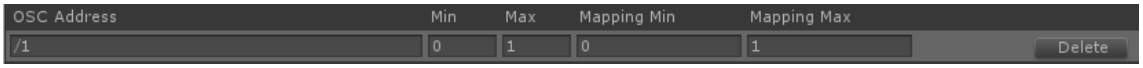


Figure 3.3: OSCMappingItem

3.5 OSC Session file

- When you work with a tool like TouchOSC you will realize that your GUI will reset when you start TouchOSC after its task is suspended. To be able to set the GUI to the latest state from your Unity session it is possible to store the last OSC data for a given OSC address.
- The OSC Session file is a container like a mapping file where you specify the addresses that should be stored. After the file is attached to your OSC Connection you store the latest data that belongs to the addresses. You are able to send all the data to your mobile device with one click so the GUI gets updated.(At the moment only numbers&strings as data are supported).

3.6 OSCEventTarget

To handle OSC data on a Unity GameObject you can add a component that is derived from the abstract class OS↩ CEventTarget. It handles all the connections to the OSCConnection and the filtering, so you can get the data and do whatever you want with your GameObject. UniOSC comes with some premade components (look in the Scripts.↩ Examples folder). The OSCEventTargetImplementation class is good blueprint for your own implementations.

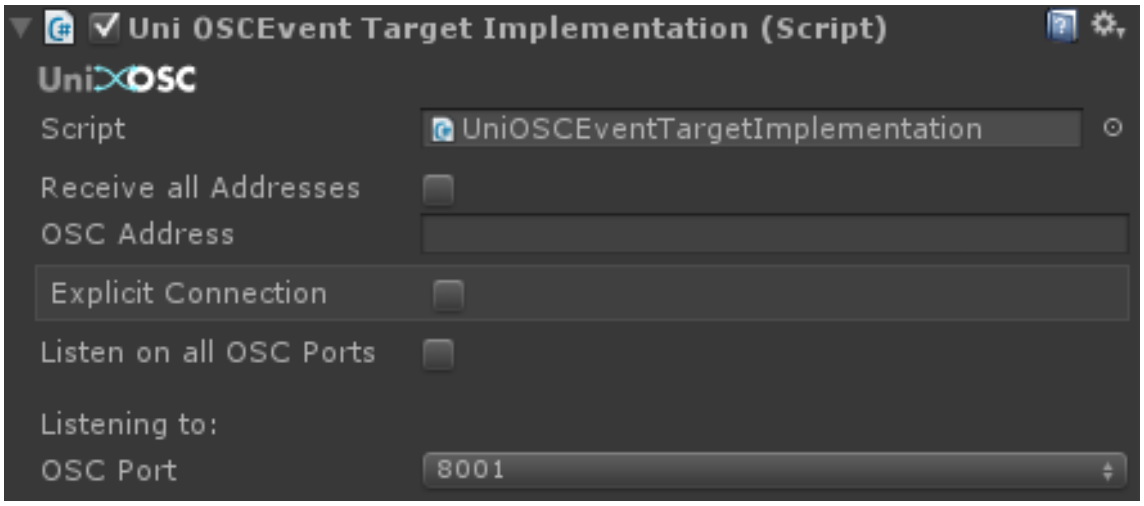


Figure 3.4: OSCEventTargetImplementation Inspector

3.7 OSCEventDispatcher

This is the opposite component as it's passing some message to a OSCconnection for sending it out to another device.

Chapter 4

UniOSC Editor Interface

- Open the OSC Editor under ‘Window/UniOSC/OSCEditor’

- At the top of the OSCEditor you see the main buttons:

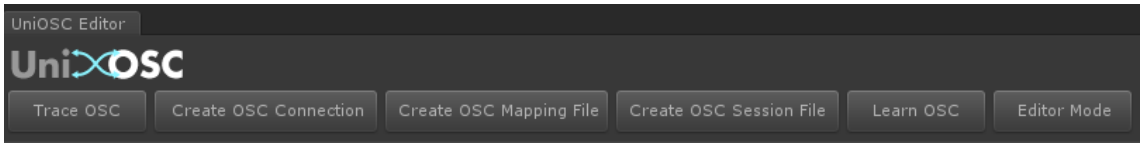


Figure 4.1: OSCEditor

4.1 Trace OSC

- For tracing all incoming & outgoing OSC messages you can toggle the trace OSC button. If you trace a text field appears on the right side of the editor which displays all the OSC messages that Unity receives/send.

- Before Unity can receive/ send OSC data you have to turn on the OSC connections with the ports you want to listen/send to.

- You can clear the content of the text field or use the ‘Display only last message‘ option (useful when your get a constant data stream like info from a Gyro).

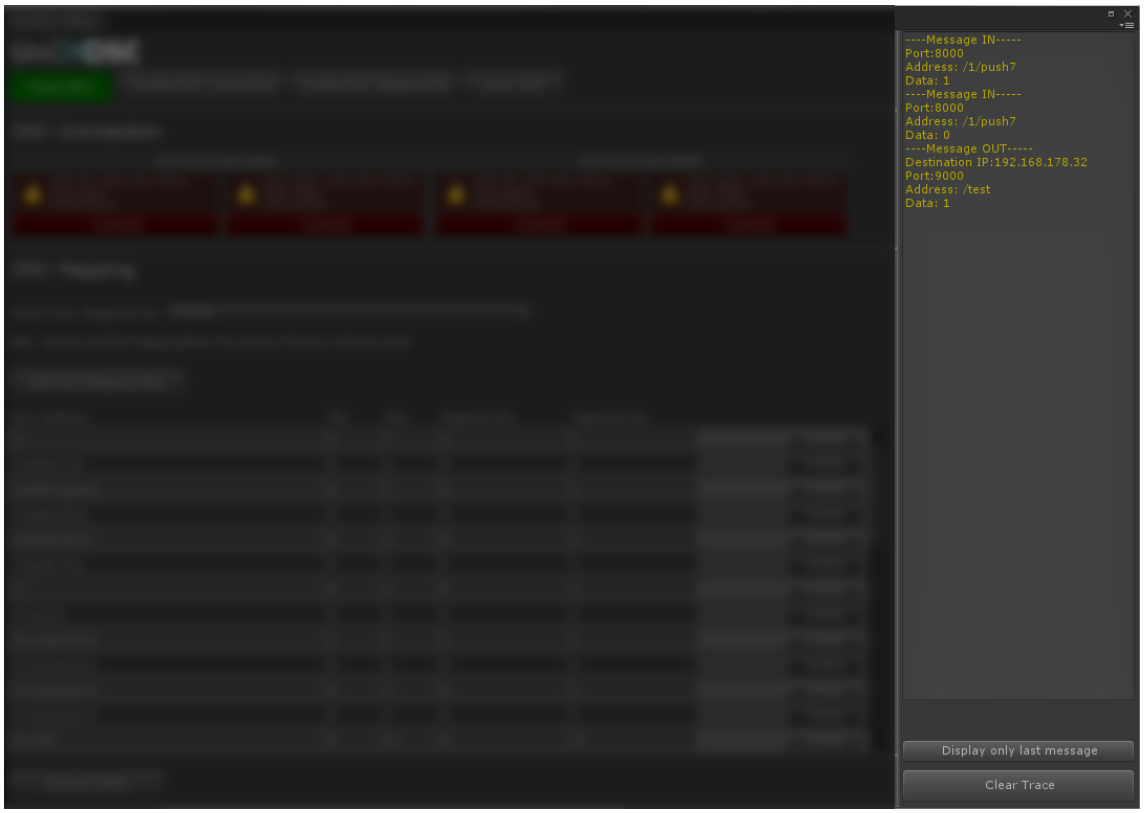


Figure 4.2: Tracing

4.2 OSC Connections

- The UniOSC Editor displays all OSCConnections that are in your project hierarchy currently enabled. If you have any problems with an OSCConnection not displayed hit the 'Refresh Editor' button to force a rescan for available OSCConnection instances.

- You see the same status boxes like in the Component Inspector. You can start and stop the connections but if you want to administrate a connection you have to select the OSCConnection GameObject in the hierarchy and work with the component inspector (Just click onto one of the status boxes and the OSCConnection get selected in the project hierarchy for editing).

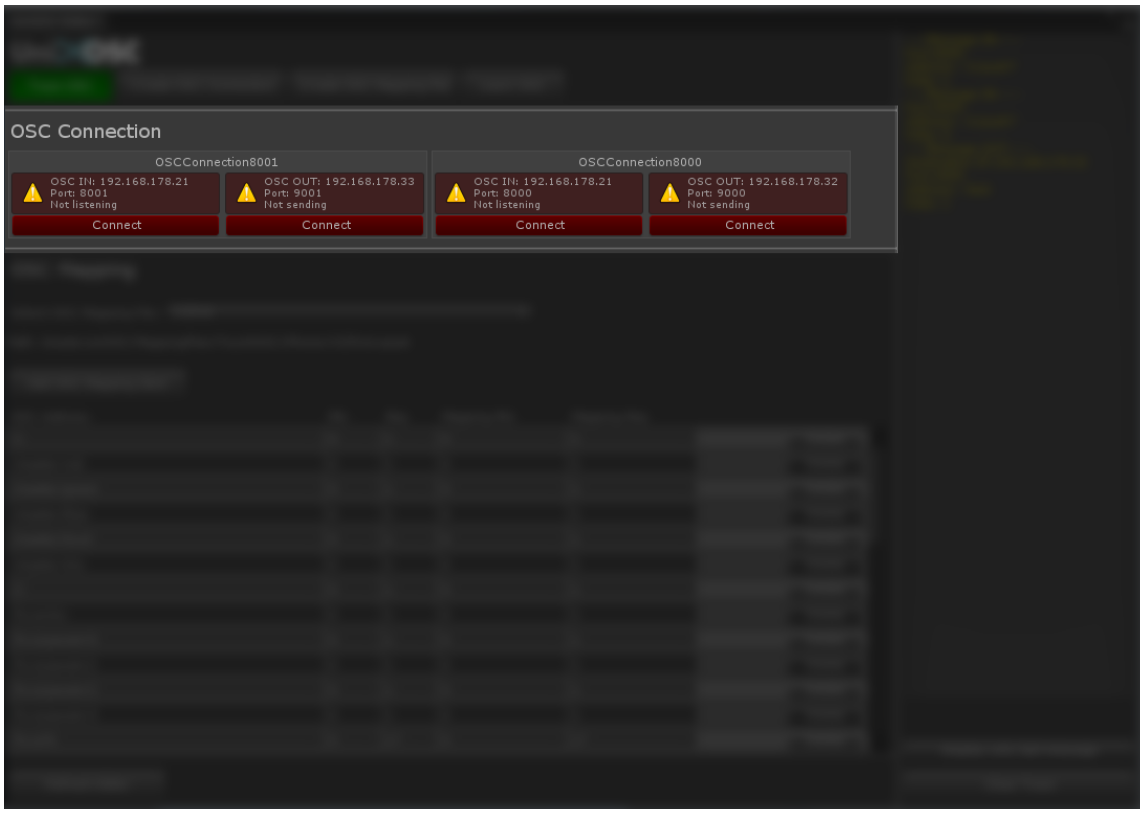


Figure 4.3: OSCConnection

- When an OSC connection is not connected it is displayed red. This means that no OSC data will be received/send.

Figure 4.4: OSCConnection 'off'

When you hit the connect button the connection establish the underlying network resources for communication. The connection box turns to green and a tiny button with a green status light appears.

Figure 4.5: OSCConnection 'on'

You're now ready to go. In some circumstances it is useful to stop the message flow into Unity, but trace incoming OSC messages. In this case you can click on the status light to set the communication in a kind of 'pause' mode. You can trace the OSC data but the data is blocked and is not send to other GameObjects. This is useful when you want to test your connections but don't want that the properties of your GameObjects are changed.

Figure 4.6: OSCConnection 'paused'

4.3 OSC Mapping

- You can attach multiple mapping files to an OSC Connection, but to prevent troubles with duplicated entries that cause only confusion with overriding mapping values it is recommended to use only one mapping file for an OSCConnection.

- To create a mapping file just click the 'Create mapping file' button in the editor or go to the Unity Editor menu under GameObject/Create Other/UniOSC/OSC Mapping File.

- In the mapping area you see a dropdown list of all available mapping files in your project. The current selected file is displayed below the dropdown list.

- If you want to attach a mapping file to a connection you can drop a mapping file asset from the Project Browser onto the connection status boxes in the editor or in the component inspector. The other way is to click on the path of the current displayed mapping file and drag to a connection box.

- If you want to add a mapping item entry to the current mapping file just hit the 'Add OSC Mapping Item' button and a new empty mapping item is appended to the mapping file. (The address is left empty but the other values are preconfigured with default values.

- To specify which message a mapping item is related to you enter a string into the OSC address text field or you can enter the learn mode.

- If you have a mapping file and need most of the addresses from another file that already exists, you can add all addresses from that file if you select an asset (mapping or session file) via the 'add Items from file' button. You can also just drag&drop the file asset from the hierarchy onto the button.

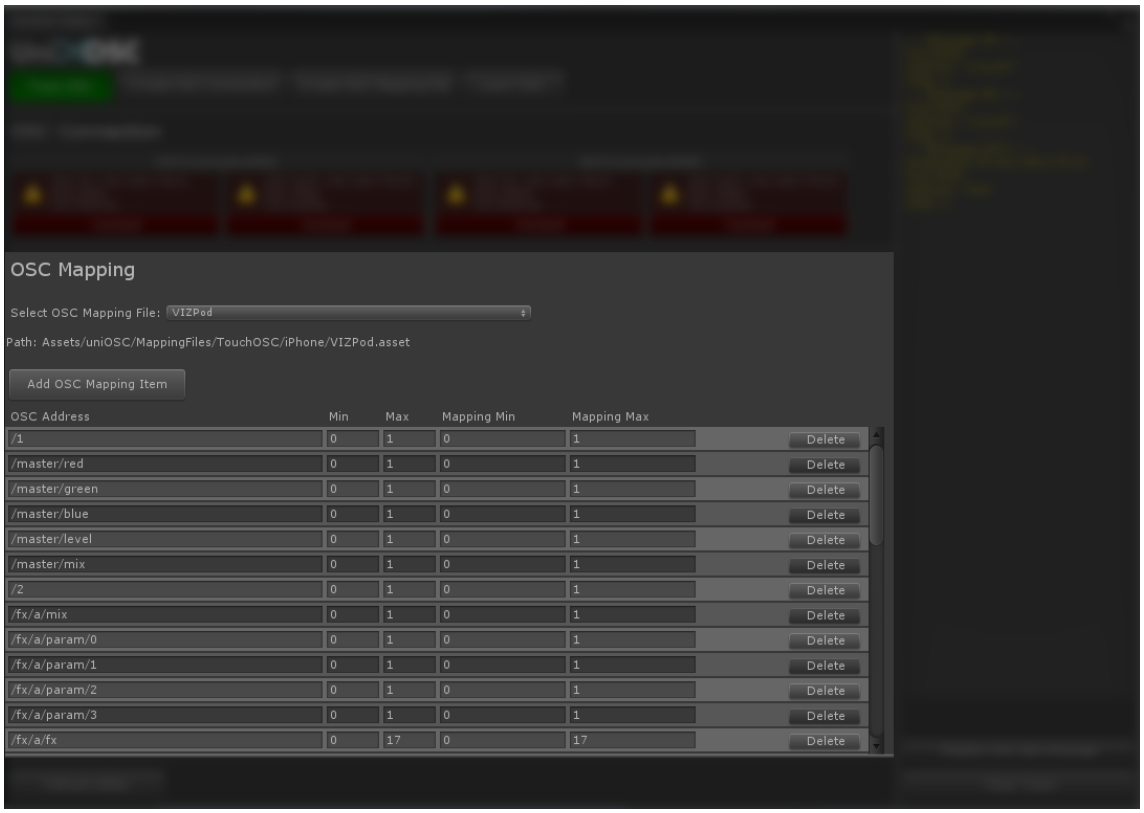


Figure 4.7: Mapping

4.4 OSC Session

- To store the latest data values that come with an OSC Message you can create a Session File. The behavior to create and assign it to an OSC Connection is the same as with a Mapping File.

- When you added a Session File to an OSC Connection the last data of the specified addresses are now stored to the file.

- The current values that are stored could be watched if you select the Session File from the popup. The first four data values from an address are displayed. (Normally only one data value comes with a OSC message)

- When you attached a Session File to an OSC Connection a button ‘Send Session Data’ appears in the component inspector. When you click the button (OSC Out Connection has to be enabled) all the data (OSC messages) from the attached files is sent on the Out Connection. This is mainly for initialization of your GUI (TouchOSC) useful, so all the elements are in sync with your last settings.

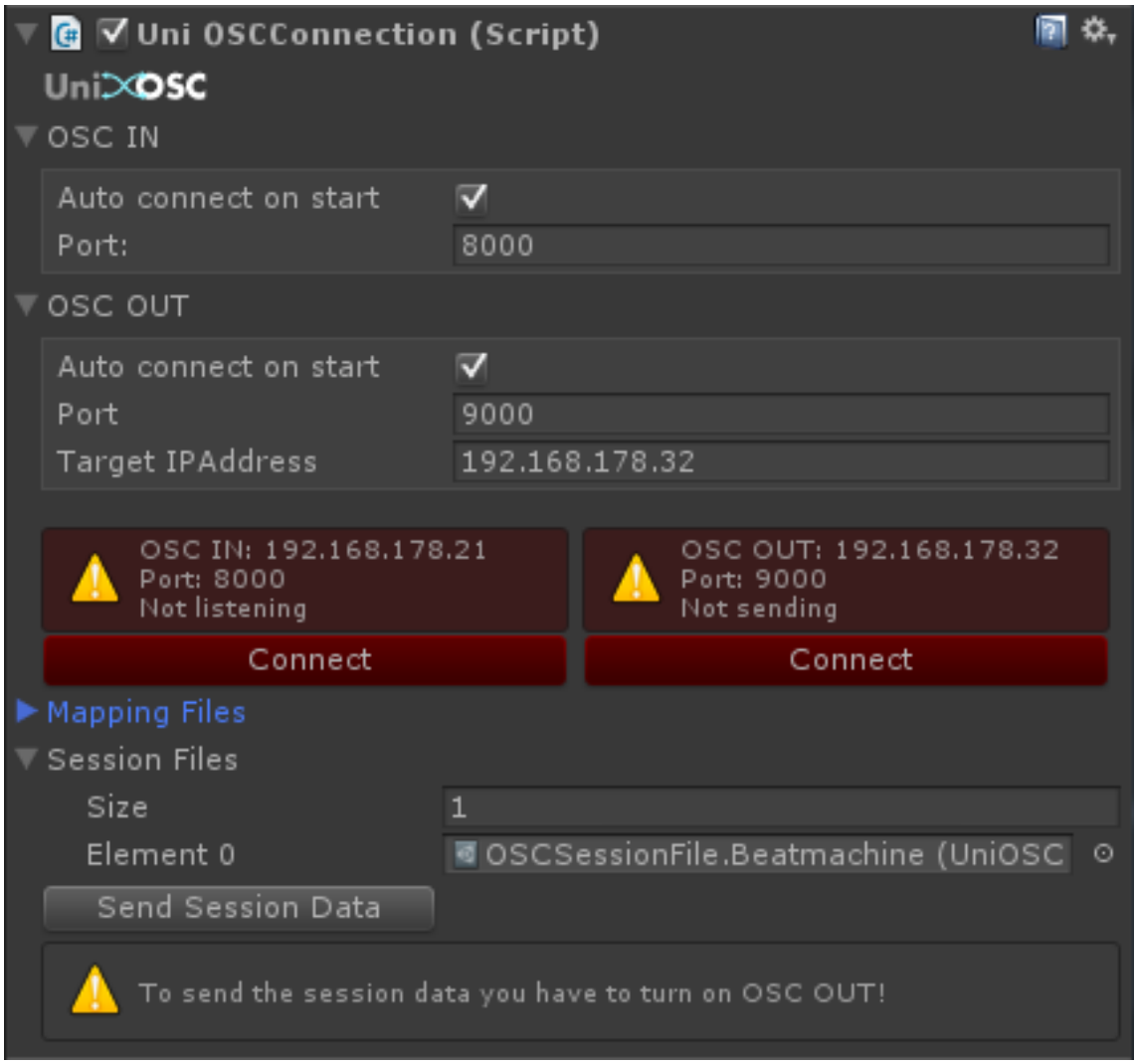


Figure 4.8: Session

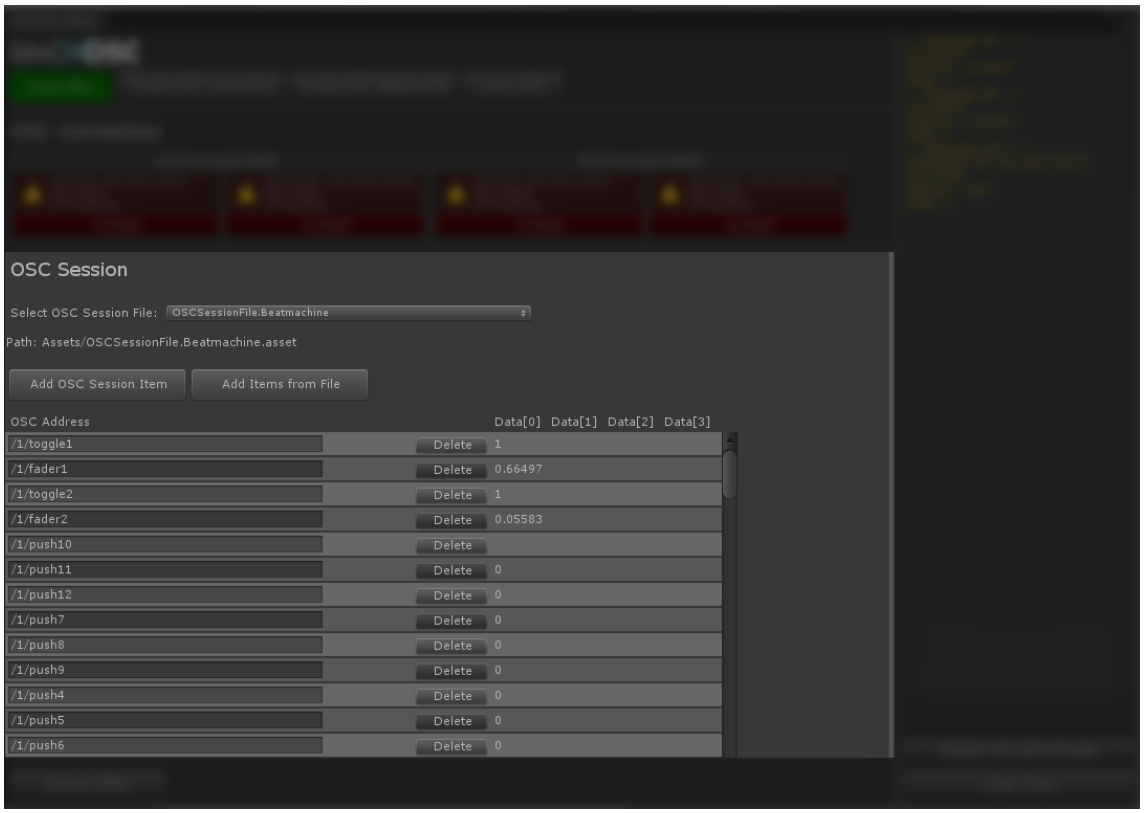


Figure 4.9: Session

4.5 Learn OSC

- In this mode a 'learn' button for every mapping/session item appears. If you push and the button the address is set automatically to the address of the actual message that arrives on a port that you are listen to. So you don't need to write it by hand and can be sure of the right spelling.

- In learn mode the OSC data is not routed into Unity so you don't have to think about if some values of your Unity GameObjects are changed when you are in edit mode.

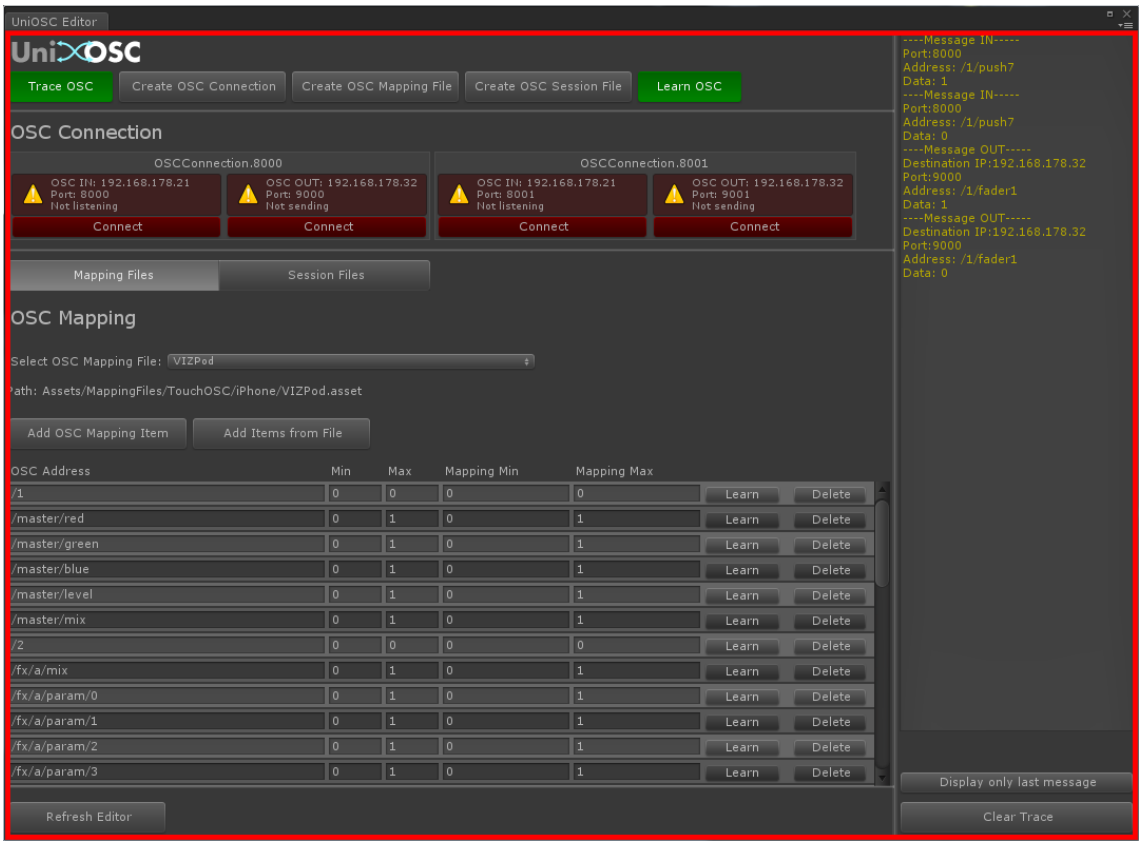


Figure 4.10: Learn mode

4.6 Editor Mode

- A huge benefit of UniOSC is the possibility to work also in Edit mode so you don't have to change to Play mode if you want to test your data flow.
- When you disable this mode you can still trace your incoming data in the UniOSCEditor but the OSC data is not routed any further.
- When you enable the Editor Mode you have to keep in mind that your incoming OSC data can change your scene permanently! This mode is useful if you want to create some editor tools.

Chapter 5

Basic workflow (with TouchOSC)

5.1 Setup a OSC connection

- Click the ‘Create OSC Connection’ button. You should see now the new created OSC connection in the editor.
- IN Connection: enter a port number of your choice (or use Port(outgoing) from TouchOSC). The transmission type should be left to 'Unicast'
- OUT Connection: specify the port (or use Port(incoming) from TouchOSC and the IP address (Local IP address from your mobile device).

5.2 Setup TouchOSC

- Go to: Settings/Connections/OSC
- Host(target IP address): has to match with the local IP address of the computer running Unity.
- Port(outgoing): has to match with the OSC IN port of an OSCConnection.

5.3 Test communication

- Push the connect buttons to establish a OSC connection.
- Use the Trace option in the Editor to check if OSC data is received by Unity.
- Send some OSC data from your app on your mobile device. You should see now the OSC messages in the trace text field. If you see nothing you have to check if your OSC connection is running, that your ports are matching or if you have a reliable network connection.

5.4 Map OSC data (Optional)

- Create a OSC mapping file
- Create OSC mapping item
- Learn OSC for setting a mapping item

5.5 Handle data in Unity

- Attach one of the UniOSC scripting components to your GameObjects you want to control and specify the address and port you want to listen to. (See Scripting classes for more info)
- If you stay in edit mode you have to be aware that if you have an open OSCConnection you can change right now your GameObject properties via OSC live in the Unity editor!

Chapter 6

Components

- To work with the OSC data inside Unity you have to add one of the components to your GameObject or write a component by yourself, based on a base class that UniOSC provides. For easy access to the UniOSC editor all component inspectors have a small icon at the top where you can click on to open the UniOSC editor.

- You always have to set the OSC address so the component can filter if a message should be handled respectively what OSC address should be sent.

- If you select the option 'Listen to all Addresses' your component receives all messages, this is handy if you want to listen to multiple OSC addresses, but are too lazy to add every message to the _oscAddresses list in code.

- All receiving components have in common that you have to specify the port you want to listen to. You can listen on one port or use the 'Listen on all Ports' option where the component listens on all available IN ports in your Scene.

- If you change a port the component reconnects it to the appropriate OSCConnection, so you can change your port live without restarting your scene.

- As this approach is not so flexible when you change the settings on your OSCConnection frequently but want to bind to only one specific port there is a third approach how you specify your binding:

- **Explicit Connection**. If you select this option you don't have to specify a port but only select the OSC↩ Connection you want to bind to. Your component uses now always the actual port from your OSCConnection. When you change the settings of your OSCConnection your components also bind to the new port or IP-↩ Address.

- In play mode the inspector gives you visual feedback to which port the component is actual listen to.

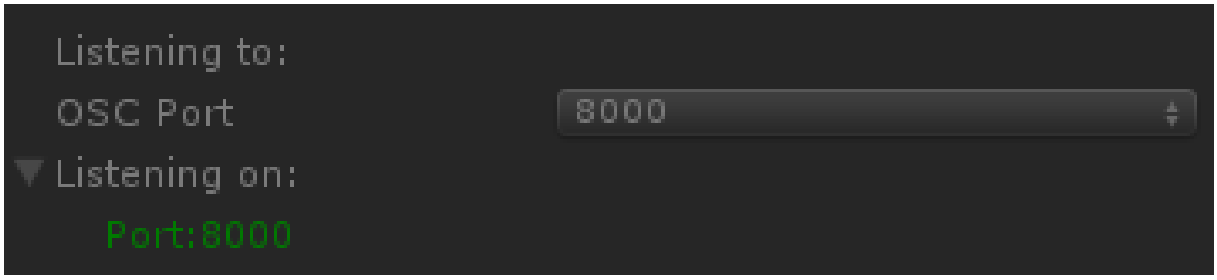


Figure 6.1: Listen to running connection

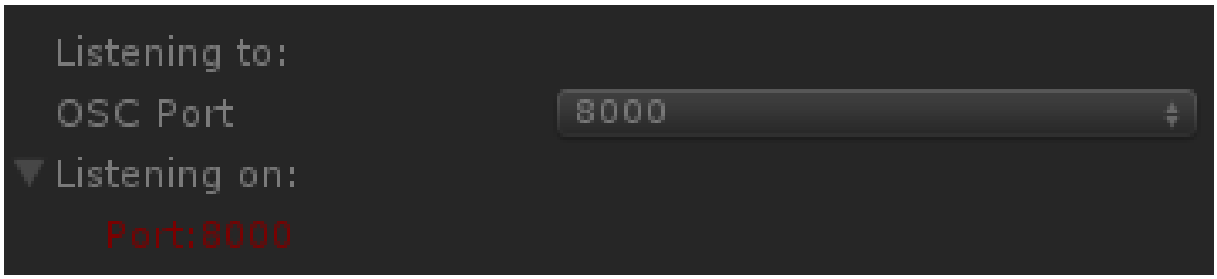


Figure 6.2: Listen but connection not running

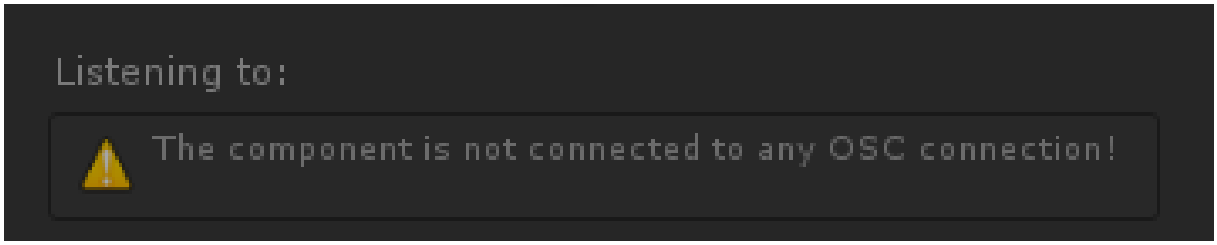


Figure 6.3: Not connected

- For receiving OSC Data you can use the UniOSCEventTarget class as a starting point to subclass from. The main method you have to implement by yourself is the 'OnOSCMessageReceived' method
- Additionally to the OnOSCMessageReceived method calling on your subclass there is also the OSC↩ MessageReceived event fired to handle the data on other classes that subscribe to this event.
- We provide the **UniOSCEventTargetImplementation** class where you could see how your own class should look like and what parsing possibilities you have.

6.1 Example Components

Most of the example components are assuming that the OSC data comes in normalized values between 0 and 1. It's up to you how you want to remap the data. You can make a mapping item for the address and make so a global remapping with a mapping file or you can remap the values on the component level with a scale factor.

6.1.1 Toggle

- This is a universal component to enable/disable most of the existing Unity components.
- With the 'ComponentToToggle' dropdown list you select one of the component types that are attached to the current GameObject you want to toggle.
- The data of the OSC message you use should be 0(disabled) or 1 (enabled).

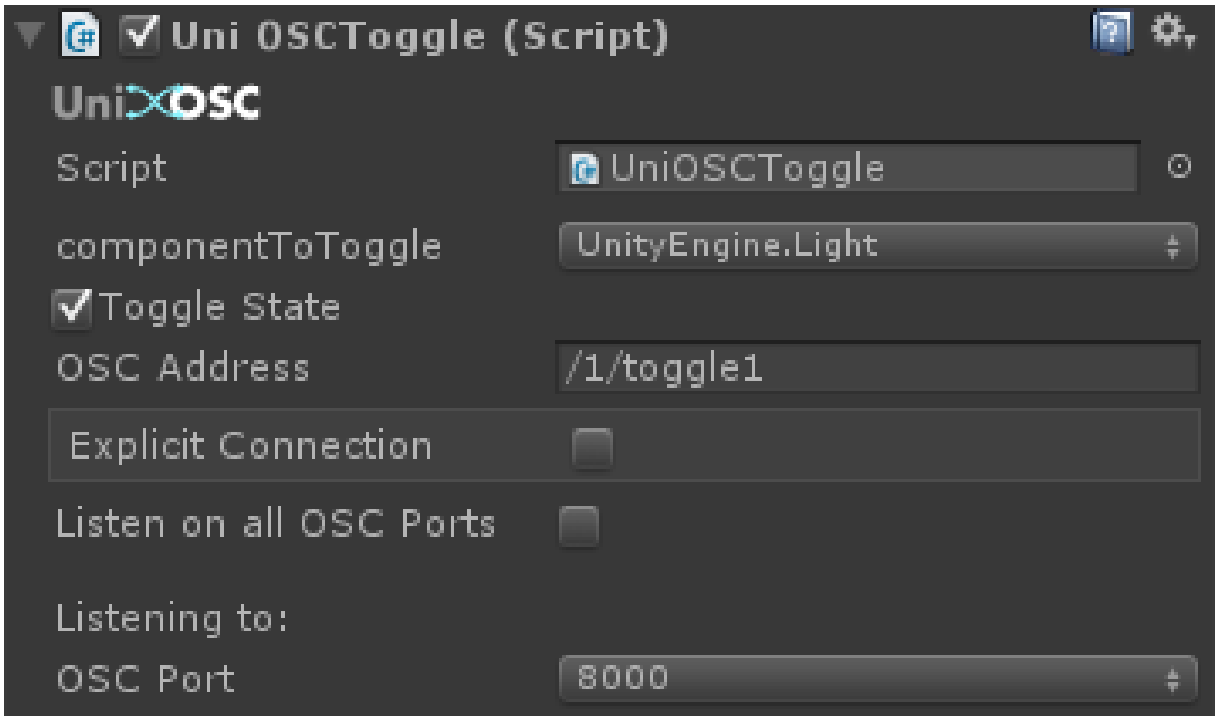


Figure 6.4: Toggle

6.1.2 Change Color

- For every color channel you can set a OSC address.

- The 'Shared Material' option is only in play mode relevant as in editor mode you always change the shared material property (Changes are stored permanently). In play mode you normally change the material (If you leave the play mode all changes are lost). If you are unfamiliar with the difference between material and shared material please look at the Unity documentation.

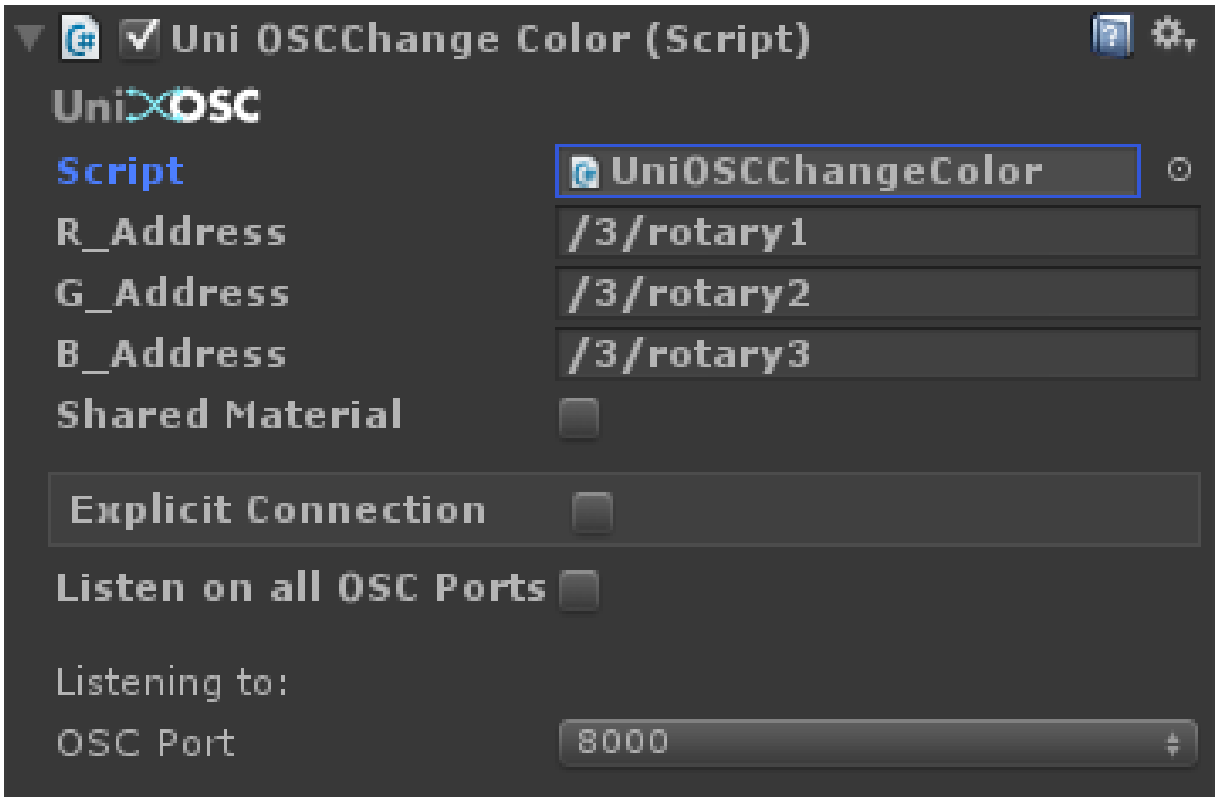


Figure 6.5: ChangeColor

6.1.3 Scale GameObject

- You can specify a scale factor for your scaling. The incoming OSC data gets multiplied by the factor.

Figure 6.6: Scale GameObject

6.1.4 Rotate GameObject

- For every axis you set the OSC address and a rotation factor.

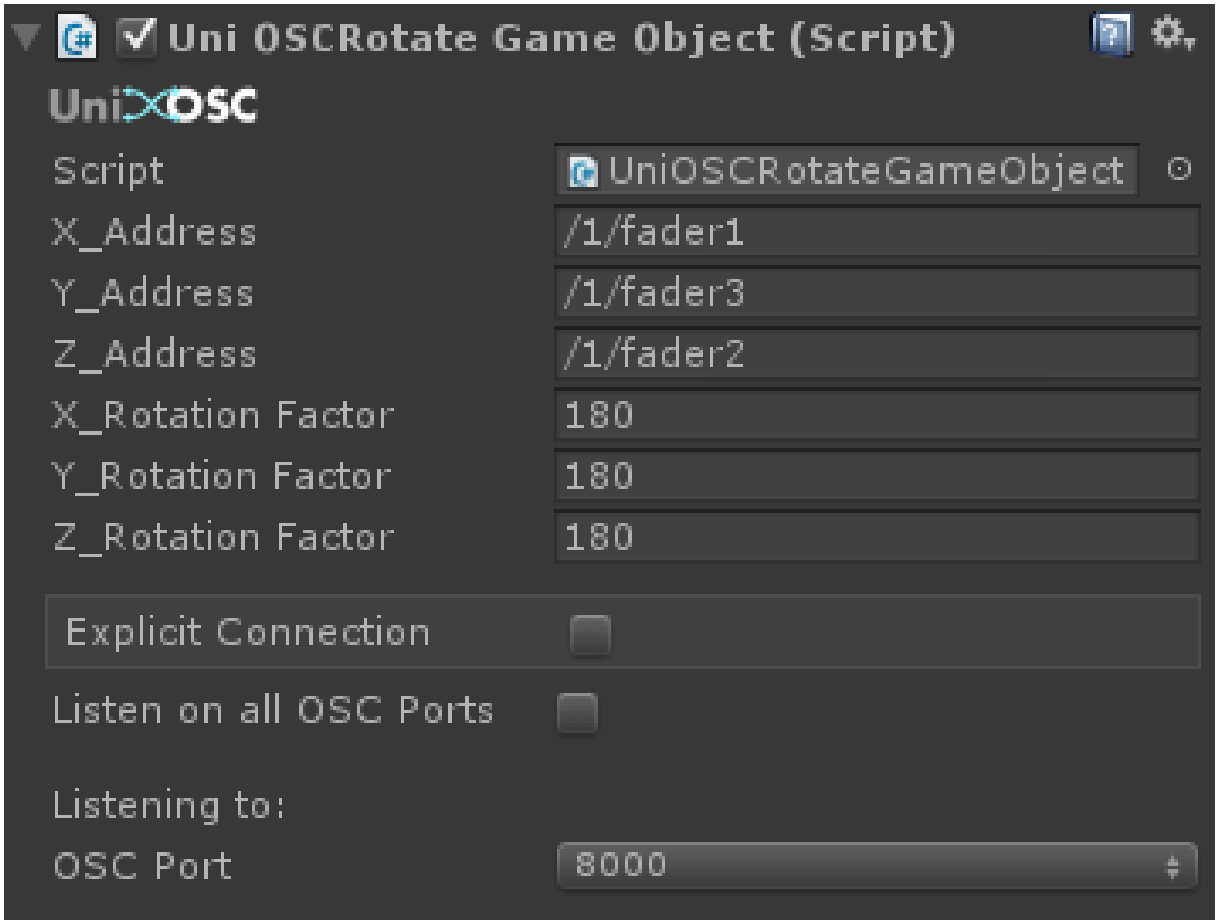


Figure 6.7: Rotate GameObject

6.1.5 Move GameObject

- In Screen Mode this component moves a GameObject in front of the main camera over the whole screen area. It assumes an OSC message with data for x and y axis normalized. (Like the XY Pad from TouchOSC . Just specify the offset from the near clipping plane from the camera. The object will move in front of the camera.

- In Relative Mode the current data values are added to the current position. So you can move your object out of the screen

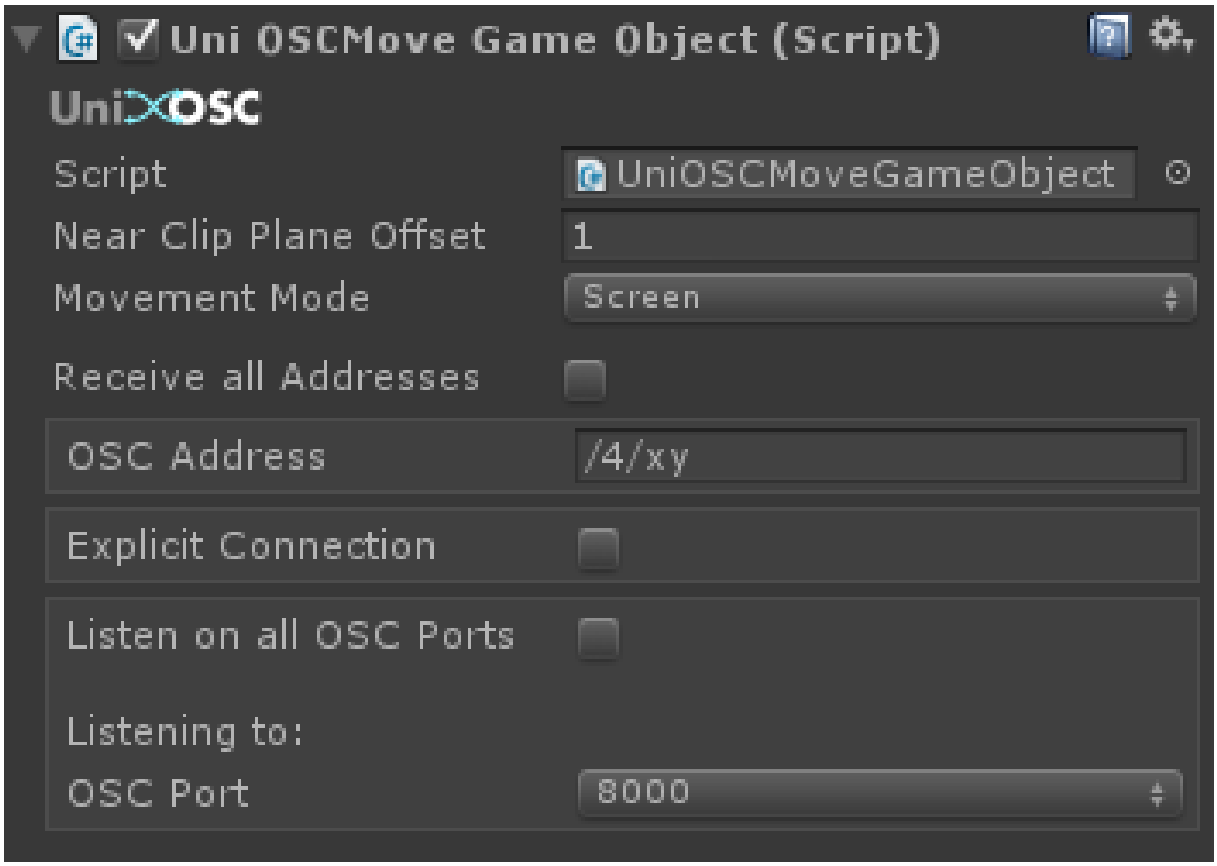


Figure 6.8: Move GameObject

6.1.6 TouchOSC Gyro Rotate

- This component is for receiving the gyro data from a TouchOSC device. (Always use the '/accxyz' string as the OSC address)

- In TouchOSC you can turn on an option to send the accelerometer data from your device. (TouchOSC Settings/Options/Accelerometer (/xyz))

- TouchOSC send the '/accxyz' message now permanently.

- The data of every axis comes in the range from -1 to +1.

Figure 6.9: Gyro Rotate

6.1.7 Send Button

- With this component you can send an OSC message. You have to turn on the 'Show GUI' option to actually see a GUI button on your screen otherwise it's more of an abstract button you can trigger from other scripts.

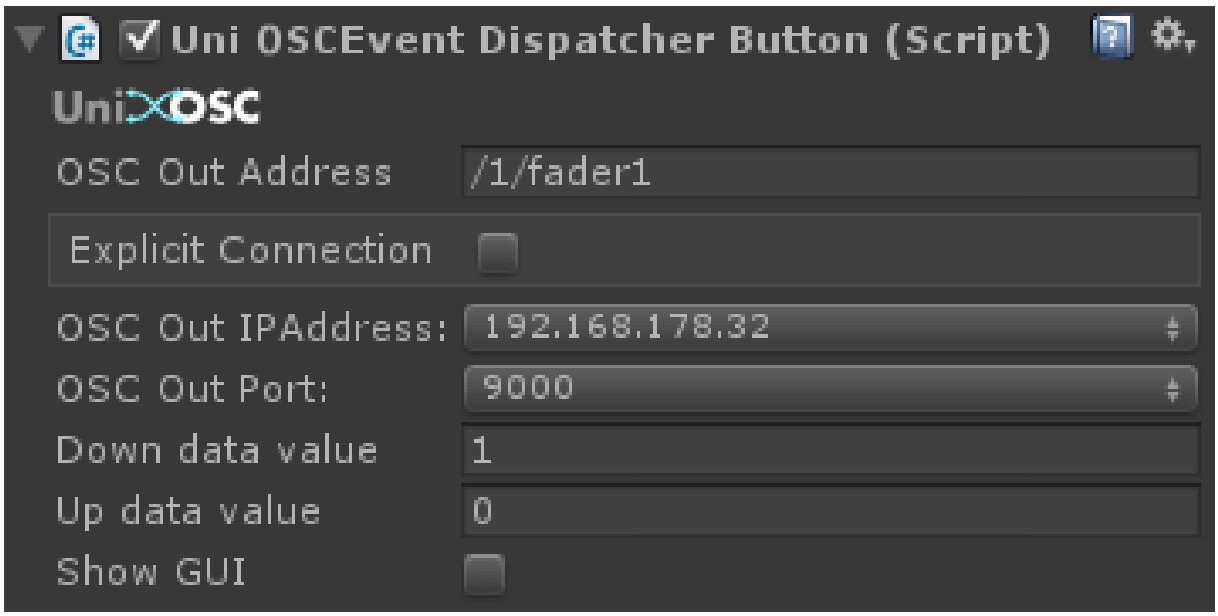


Figure 6.10: Send Button

6.1.8 OSC GUI

- This component is intended for testing and debugging the OSCConnections on a running app and gives you parts of the UniOSC editor functionality.

Figure 6.11: OSC GUI component

6.1.9 Transform Sender

- With this component you can send the transform data of a GameObject in a continuously way. If you don't specify a Tracked GameObject the current hosting GameObject is used.

6.1.10 JavaScript Communication

- UniOSC is written in C# so if you want to code in JS you have to know that there are some limitations. There is only one direction of where the C# world can communicate with the JS world . This is a general problem of Unity. So first you have to decide from which direction your communication should flow (C# -> JS or JS -> C#. Depending on that the scripts of one Language must be located in the Plugins folder to be compiled

first. Look at http://docs.unity3d.com/Manual/ScriptCompileOrderFolders.html for more info. To prevent a lot of trouble with compilation problems we only suggest that you use C# -> JS communication so you only have to locate your JS scripts in the Plugins folder. We provide a JSBridge↩ Demo.unityPackage that installs a little demo scene with a demo setup that you can use as a starting point for your own JS scripting. As we are limited in the way to send custom data types from C# to JS we have to parse our OSC data before we can send only primitive types like int or strings and the one way communication we strongly suggest to only script in C#!

Chapter 7

Scripting classes

- Working with Components is easy and you have visual inspectors for the set up but if you need to handle OSCData in classes that don't derive from a MonoBehaviour or prefer to work mostly in pure code you can use the class based versions of the OSCDispachers and OSCReceiver classes.
- The common handling for class based instances are:
  1. Create a instance with calling one of the constructors. There are several class constructors you can use to specify your type of instance like the way you can specify it on the component based versions. You only have to override the constructors you want to use and call the base constructor of that type
  2. Call instance.Enable(). Without enabling, the OSC data handling will not be initialized. Later the Enable() methods could re-enable your instance when it is temporary disabled.
  3. Call instance.Disable() if you want to disable the instance only temporary.
  4. Call instance.Dispose() if you want to delete the instance. This is important to make sure that the On↩ OSCMessageReceived method isn't called any further or none of your Event callbacks are still called, even if your instance is set to null or the class that hosting your instance is destroyed.
- The **UniOSCEventTargetCBImplementation** and **UniOSCEventDispatcherCBImplementation** classes are good blueprints for your own scripting
- If you want to use the class based scripts in a Unity Editor class you have to consider a couple of additional steps. The problem is that your scripts will lose the references to the OSCConnections when you change the playmode state of Unity. For the instances that don't use the ExplicitConnection feature you only have to re-enable your instance to force a new connection set up on your instance. For the other instances you have to re-create a reference to the OSCConection from the InstanceID and call the SetExplicitConnection() method. We provide a UniOSCTestEditor script that shows exactly what steps you have to do for your own Editors.(In Unity go to: Window/UniOSC/Test/ScriptTestEditor)

7.1 Receiving OSC data

To create a class that needs to listen to an OSC Message you can create a class that is a subclass from the **UniOSCEventTargetCB** class.

7.1.1 Constructors

You can choose between several constructors to get the same features you know from the component versions:

- **UniOSCEventTargetCB(int oscPort)**
  Your instance auto connect to the first OSCConnection that has the given IN port and handles all OSC↩ Addresses

- **UniOSCEventTargetCB(string oscAddress)**
  Your instance auto connects to all available OSCConnections and handle all OSCMessages that match the given OSCAddress
- **UniOSCEventTargetCB(string oscAddress, int oscPort)**
  Your instance only react to a OSCMessage that comes from a given port with a given address pattern
- **UniOSCEventTargetCB(UniOSCConnection con)**
  This is the Explicit Connection feature. You listen to all OSCMessages that come from a given OSC↩ Connection, regardless if the port is changed later.
- **UniOSCEventTargetCB(string oscAddress, UniOSCConnection con)**
  Explicit Connection feature but you specify the OSCAddress pattern that you want to react to.

7.2 Sending OSC data

For sending data it is almost the same like receiving

7.2.1 Constructors

You can only choose between two constructors:

- **UniOSCEventDispatcherCB(string oscOutAddress, string oscOutIPAddress,int oscPort)**
  The OSC data you have added are sent with the OSCOutAddress to a given IPAddress on a given port when there is a OSCConnection with a matching IPAddress/port available
- **UniOSCEventDispatcherCB(string oscOutAddress, UniOSCConnection explicitConnection)**
  An OSC Message is sent with a given OSCConnection. The current settings of the OSCConnection determine the IPAddress/port

Chapter 8

Sending OSC

- For sending OSC Data you can create your own subclasses from the abstract class UniOSCEventDispatcher or UniOSCEventDispatcherCB. You specify an OSC address, the port and an IP address. (The UniOSC↩ EventDispatcherImplementation class is a good starting point for your own implementations.)
- UniOSC supports sending OscMessages and OscBundles. The default state of the components is sending an OscMessage. If you try to add a OscMessage without setting the bundle mode you get a debug warning that this is not possible. You can change at any time from bundle mode to normal message mode back (SetBundleMode(false);), but then you have to reassign your data with AppendData().
- UniOSC creates a default OSC Message/Bundle for you in the background so you don't have to explicit add one, but if you want to send several OSCMessages with one component you can use a bundle or add separate OSCMessages. We provide an MultiAddress/MultiConnection example to show you how this could be done.
- When you want to clear the data you can call the ClearData() method.
- If you need to send OSC data constantly (from an Input device for example) you can specify the “sendInterval” property were you set the interval in milliseconds the component should send OSC data. (Sending too much OSC Messages in a short timespan can cause transmission problems)

8.1 Message Mode

- If you work on the OscMessage level you just call the AppendData(data) method and add the data to your OscMessage you want to send. You can add as much data as you want as long as the data type is supported by the OSC protocol.
- Supported types are: Int32, Int64, Single, Double, String, Byte[], OscTimeTag, Char, Color, Boolean.

8.2 Bundle Mode

- If you want to send a OSCBundle you just call the SetBundleMode(true); After that you can append several OscMessages with the AppendData(oscMessage) method.
- The main advantage of Bundles is that you can encapsulate several OscMessages into one OscBundle and only need to force one sending of the whole bundle. This also reduces the network traffic. Keep in mind that your OscReceiving app needs to support the handling of bundles. If it doesn’t support bundles you need to send every OscMessage with a separate call.

Chapter 9

Common pitfalls

- If an OSCConnection is connected the port of the connection is in use, so no other OSC connection can use this port. To prevent confusion it's always a good practice to always use different ports for every OSC↩ Connection.
- If you use multiple mapping files on an OSCConnection are sure that the address spaces don't overlap. Otherwise only the mapping values of the last mapping file that is parsed will be used for remapping.
- The OSC message appears in the trace text field but my game objects don't receive a message:
-
  - Are you in learning mode?
  - Is the OSCConnection paused?
  - Check your port at the GameObject component.
- When you use the gyro data from TouchOSC other OSC controlled objects that listen on the same port can begin to react sluggish. It's best to use a separate device for the gyro and use another device on another port for the other GameObjects.
- Depending on your network topology you will have a latency when you use your mobile device to send/receive OSC data. To get the best performance we recommend setup an Ad-Hoc Network to get the lowest latency for best user experience. See the Links section for further information.

Chapter 10

Known Issues

- Only float/int data types are mappable
- Mapping files .assets cause issues when importing in a project with other Unity version from the version of creation. Please read the ‘Read.Me.First!.txt’ file in the ‘Mapping Files’ folder.

Chapter 11

History

Version 1.7 – 2015.09.01

- Multicast & Broadcast support
  - In the OSCConnection component you specify now the transmission type (Unicast, Multicast or Broadcast)
- Improved IP address validation
  - The OSCConnection inspector gives visual feedback if a chosen IP address is not valid. At runtime you get error messages in the console that makes debugging easy.
- OSCSharp.dll update
- Documentation update

Version 1.6 – 2015.04.29

- Sending OSC Bundle support
  - A OSC Bundle encapsulates several OSC messages into one package to reduce network traffic overhead
  - You can always change between the old Message mode (default) and the Bundle mode
- OSCEventArgs has now the Packet property (OscPacket) instead of a Message property for supporting Osc↩ Messages and OscBundles. If you have legacy code where you access the message of a UniOSCEventArgs object you have to change it this way: OscMessage msg = ((OscMessage)my UniOSCEventArgsObj.Packet);
- Changing a property of a Dispatcher/EventTarget (from inspector or via code) now works transparently.
  - No need for manually re-enable your component/class (auto-reconfiguration).
- AppendData is much more flexible
  - You don't have to enable your dispatcher/eventTarget before add some data.
  - Data is now persistent. No need to reassign your data when your component is disabled once.
- AppendData works in Message and in Bundle mode almost the same way
  - Message mode: AppendData(myDataType);
  - Bundle mode: AppendData(myOSCMessage);
- New demo classes

  - MultiAddressSender to show how to send multiple OSC messages with one script (several individual messages or one bundle)
  - MultiConnectionSender to show how to send one message or bundle via several connections with one script.

- Flux Timeline Editor support
  - You can now sent OSC messages at points in time with the Flux Timeline editor http://u3d.as/content/nuno-afonso/flux

- Cleanup folder structure
  - All code that relies on external assets are now under UniOSC/scripts/External
  - Some Editor classes are rearranged into new folders

- Documentation update

Version 1.5 – 2015.02.27

- Unity 5 ready
- Added Editor Mode
  - You have to enable the editor mode now explicit to route your OSC data to the Components.
- Changed the Component Inspector for Dispatchers and Receivers
  - The base Inspectors are now drawing the DefaultInspector .
  - You can now create your own scripts based on the UniOSCEventTarget & UniOSCEventDispatcher and all public properties are displayed right out of the box. (You don't need to write a custom inspector any more to show your own public properties)
- Added a Mode option at UniOSCMoveGameObject.
  - You have now a Screen mode (like before) and a Relative Mode (additive movement that is not bound to the camera rect)
- Added a UniOSCTransformSender component for sending the transform data (position & rotation) of a GameObjects continuously

Version 1.4 - 2014.11.24

- Class based versions of OSCEventTarget and OSCDispatcher. You can now handle OSC messages in classes that don't derive from MonoBehaviour. This makes it possible to work only in code and in editors.
- Created a demo editor to show the new class based feature to use in Unity Editors.
- Created a demo scene with class based sending and receiving.
- Fixed a bug with the explicit connection mode (OSC data of OSCEventDispatchers was always reset when a status changed event was fired from a OSCConnection)
- Added a ClearData method to the OSCEventDispatcher class so you can clear the OSC data.
- Added an OSCMessageReceived event to the OSCEventTarget classes. This event is additionally fired when the OnOSCMessageReceived method is called.
- Documentation update

Version 1.3 - 2014.11.07

- Added Explicit Connection mode for OSCEventTargets and OSCEventDispatchers.
- UniOSCGUI updated: You can now change the port and IP-Address in the GUI at runtime for a flexible setup.
- Mobile Example scene updated with new UniOSCGUI and added several components that use the Explicit Connection.
- Added the JSBridgeDemo.unitypackage for showing how to send data to a JavaScript class.
- Documentation update

Version 1.2 - 2014.07.02

- UniOSCEventDispatcher can now send more than only one float value as data. You can add as much data as you want as long as the data type is supported by the OSC protocol (Int32,Int64,Single,Double,String,Byte[],OscTimeTag,Char,Color,Boolean)
- Added a UniOSCEventDispatcherImplementation class as a blueprint for OSC sending components.
- Added External InputDevice scripts (Third party Assets need to be installed to work)
  - Send OSC data with the SpacNavigator Controller (Asset from Patrik Hogenboom) : https://www.assetstore.unity3d.com/en/#!/content/9774
  - Send OSC data with Razor Hydra Controller from Sixsense Studios: https://www.assetstore.unity3d.com/en/#!/content/7953
- Unity 4.5 Bug fixed (Tracing of OSC messages causes an exception when there were more than 15000 chars in the TextField. Tracing TextField now displays maximal 8192 chars.
- Documentation update

Version 1.1 - 2014.05.29

- Session file support
  - Store the latest data that comes with a OSC message
  - Send all data to update the GUI state of an external app like TouchOSC
- Change in OSCsharp lib: made the TypeTag property of a OSC message accessible
- Type of the data could now be verified with the Typetag string. No GetType() necessary if you have performance problems
- receiveAllAddresses property is now accessible via the component inspector
- UniOSCEventArgs could now be filtered by Group, AddressRoot or AddressIndex if the OSC address matches a specific pattern
- Documentation update

Version 1.0 - 2014.05.13

- Initial release

Chapter 12

Credits

UniOSC uses a modified version of **OSCsharp** written by Valentin Simonov based on Bespoke Open Sound Control Library by Paul Varolik

- Original version https://github.com/valyard/OSCsharp
- Modified version: https://github.com/sloopidoopi/OSCsharp

- GuiScaler class from: https://gist.github.com/darktable/2018687
- **TouchOSC** by Hexler: http://hexler.net/software/touchosc

Chapter 13

Links

- **OSC**: http://opensoundcontrol.org/spec-1_0
- **TouchOSC** control reference: http://hexler.net/docs/touchosc-controls-reference
- Establishing a Computer-to-Computer (Ad-Hoc) network without a router
  Windows:
  http://windows.microsoft.com/en-us/windows/set-computer-to-computer-adhoc-network#1↩
  TC=windows-7
  OSX:
  http://support.apple.com/kb/PH10666
  Multicast-addresses:
  http://www.iana.org/assignments/multicast-addresses/multicast-addresses.↩
  xhtml

Chapter 14

Support

If you need support or have any question/suggestions please contact us.

- Website: http://uniosc.monoflow.org
- Email: info@monoflow.org
- Unity Forum: http://forum.unity3d.com/threads/247204

monoflow
new media experience

Chapter 15

Namespace Documentation

15.1 Package UniOSC

Classes

- class UniOSCAbstractItem

  Uni OSC abstract item is the base class for Mapping/Session Items.
- class UniOSCChangeColor

  Change the color of the material from the GameObjects.
- class UniOSCChangeColorEditor
- class UniOSCConnection

  This class is responsible for all the network related tasks.
- class UniOSCConnectionEditor
- class UniOSCEditor

  Editor for the administration of OSCconnections, mapping files.
- class UniOSCEditorConfigObj

  UniOSC editor config object.
- class UniOSCEventArgs

  A wrapper to a OscMessage class to also store the port and have a quick way to access the message address.
- class UniOSCEventDispatcher

  This is the abstract class you should subclass from when you want to sent OSC data
- class UniOSCEventDispatcherButton

  Dispatcher button that forces a OSCConnection to send a OSC Message.
- class UniOSCEventDispatcherButtonEditor

  Uni OSC event dispatcher button editor.
- class UniOSCEventDispatcherCB
- class UniOSCEventDispatcherCBImplementation

  This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.
- class UniOSCEventDispatcherCBRawImplementation

  This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.
- class UniOSCEventDispatcherCBSimple

  This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.
- class UniOSCEventDispatcherEditor
- class UniOSCEventDispatcherImplementation

  This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.

- class UniOSCEventDispatcherMultiAddressSender

  This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.
- class UniOSCEventDispatcherMultiConnectionSender

  This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.
- class UniOSCEventDispatcherSlider

  Dispatcher toggle that forces a OSCConnection to send a OSC Message.
- class UniOSCEventDispatcherToggle

  Dispatcher toggle that forces a OSCConnection to send a OSC Message.
- class UniOSCEventTarget

  UniOSC event target.
- class UniOSCEventTargetCB

  UniOSC event target for class based scripting.
- class UniOSCEventTargetCBImplementation

  This class is a blueprint for your own implementations of the abstract class OSCDispatcherTargetCB //Don't forget the base callings !!!! The OnOSCMessageReceived method is where you should parse the OSC data
- class UniOSCEventTargetEditor
- class UniOSCEventTargetImplementation

  This class is a blueprint for your own implementations of the abstract class OSCDispatcherTarget //Don't forget the base callings !!!! The OnOSCMessageReceived method is where you should parse the OSC data
- class UniOSCFileObj
- class UniOSCGUI

  GUI class that mimics the UniOSC editor interface for runtime use You can start/stop the OSCConnections and trace OSC data messages
- class UniOSCMappingFileObj

  Mapping file class .
- class UniOSCMappingFileObjEditor
- class UniOSCMappingItem

  Uni OSC mapping item.
- class UniOSCMappingItemEditor
- class UniOSCMoveGameObject

  Moves a GameObject in normalized coordinates (ScreenToWorldPoint)
- class UniOSCReceiver

  Uni OSC receiver.
- class UniOSCRotateGameObject

  Rotates (localRotation) the hosting game object.
- class UniOSCRotateGameObjectEditor
- class UniOSCRotateGameObjectTouchOSCGyro
- class UniOSCScaleGameObject

  Uni OSC scale game object.
- class UniOSCScriptTestEditor

  Editor for the administration of OSCconnections, mapping files.
- class UniOSCSessionFileObj

  OSC Session file class .
- class UniOSCSessionFileObjEditor
- class UniOSCSessionItem

  Uni OSC mapping item.
- class UniOSCSessionItemEditor
- class UniOSCToggle

  With this class you can toggle most of the Unity Components on/off The data of the OSC message should be only 0(off) or 1(on)
- class UniOSCToggleEditor
- class UniOSCTransformSender
- class UniOSCTransmitter

15.2 Package UnityEngine

Classes

- class GUIScaler

  Usage:

Chapter 16

Class Documentation

16.1 UnityEngine.GUIScaler Class Reference

Usage:

Static Public Member Functions

- static void Initialize (float scale)

  Initialize the gui scaler with a specific scale.
- static void Initialize ()

  Initialize the gui scaler using the detected screen dpi.
- static void Begin ()

  All gui elements drawn after this will be scaled.
- static void End ()

  Restores the default gui scale.

Properties

- static Vector3 GuiScale `[get]`

16.1.1 Detailed Description

Usage:

(optional) Call GUIScaler.Initialize() in Start(), Awake() or OnEnable() (only needed once) Call GUIScaler.Begin() at the top of your OnGUI() methods Call GUIScaler.End() at the bottom of your OnGUI() methods

WARNING: If you don't match Begin() and End() strange things will happen.

16.1.2 Member Function Documentation

16.1.2.1 static void UnityEngine.GUIScaler.Begin ( ) `[static]`

All gui elements drawn after this will be scaled.

16.1.2.2 static void UnityEngine.GUIScaler.End ( ) [static]

Restores the default gui scale.

All gui elements drawn after this will not be scaled.

16.1.2.3 static void UnityEngine.GUIScaler.Initialize (float *scale*) [static]

Initialize the gui scaler with a specific scale.

16.1.2.4 static void UnityEngine.GUIScaler.Initialize ( ) [static]

Initialize the gui scaler using the detected screen dpi.

16.1.3 Property Documentation

16.1.3.1 Vector3 UnityEngine.GUIScaler.GuiScale [static],[get]

16.2 UniOSC.UniOSCAbstractItem Class Reference

Uni OSC abstract item is the base class for Mapping/Session Items.

Public Attributes

- bool isLearning
- string address = ""

16.2.1 Detailed Description

Uni OSC abstract item is the base class for Mapping/Session Items.

16.2.2 Member Data Documentation

16.2.2.1 string UniOSC.UniOSCAbstractItem.address = ""

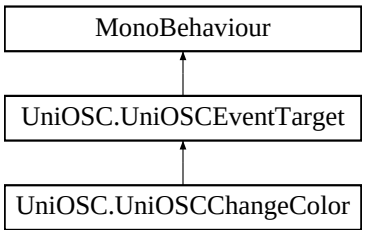
16.2.2.2 bool UniOSC.UniOSCAbstractItem.isLearning

16.3 UniOSCAutoRun Class Reference

16.4 UniOSC.UniOSCChangeColor Class Reference

Change the color of the material from the GameObjects.

Inheritance diagram for UniOSC.UniOSCChangeColor:



Public Member Functions

- override void OnEnable ()

  Enable this component and reinitialize.
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- string R_Address
- string G_Address
- string B_Address
- bool sharedMaterial

Additional Inherited Members

16.4.1 Detailed Description

Change the color of the material from the GameObjects.

Option to choose between Material and SharedMaterial

16.4.2 Member Function Documentation

16.4.2.1 override void UniOSC.UniOSCChangeColor.OnEnable ( ) [virtual]

Enable this component and reinitialize.

If a property of the component is changed via the inspector we force a OnEnable to update the status of the component. In general the component disconnects from all OSCConnections and try to find a new OSCConnection to connect to with a matching port. If you change properties via code you should call this explicit.

Reimplemented from UniOSC.UniOSCEventTarget.

16.4.2.2 override void UniOSC.UniOSCChangeColor.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implements UniOSC.UniOSCEventTarget.

16.4.3 Member Data Documentation

16.4.3.1 string UniOSC.UniOSCChangeColor.B_Address

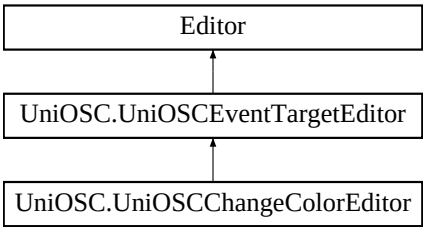
16.4.3.2 string UniOSC.UniOSCChangeColor.G_Address

16.4.3.3 string UniOSC.UniOSCChangeColor.R_Address

16.4.3.4 bool UniOSC.UniOSCChangeColor.sharedMaterial

16.5 UniOSC.UniOSCChangeColorEditor Class Reference

Inheritance diagram for UniOSC.UniOSCChangeColorEditor:



Public Member Functions

- override void OnInspectorGUI ()

Additional Inherited Members

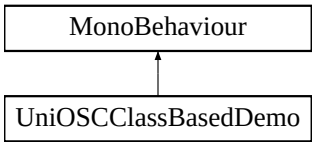
16.5.1 Member Function Documentation

16.5.1.1 override void UniOSC.UniOSCChangeColorEditor.OnInspectorGUI ( )

16.6 UniOSCClassBasedDemo Class Reference

Demo to show how to use the class based scripts.

Inheritance diagram for UniOSCClassBasedDemo:



Public Attributes

- string OSCAddress
- int OSCPort
- UniOSCConnection OSCConnection
- string OSCAddressOUT
- string OSCIPAddressOUT = "192.168.178.32"
- int OSCPortOUT

- UniOSCConnection OSCConnectionOUT
- Light Light1
- Light Light2
- Light Light3
- bool sendData
- float sendInterval =1000

16.6.1 Detailed Description

Demo to show how to use the class based scripts.

16.6.2 Member Data Documentation

16.6.2.1 Light UniOSCClassBasedDemo.Light1

16.6.2.2 Light UniOSCClassBasedDemo.Light2

16.6.2.3 Light UniOSCClassBasedDemo.Light3

16.6.2.4 string UniOSCClassBasedDemo.OSCAddress

16.6.2.5 string UniOSCClassBasedDemo.OSCAddressOUT

16.6.2.6 UniOSCConnection UniOSCClassBasedDemo.OSCConnection

16.6.2.7 UniOSCConnection UniOSCClassBasedDemo.OSCConnectionOUT

16.6.2.8 string UniOSCClassBasedDemo.OSCIPAddressOUT = "192.168.178.32"

16.6.2.9 int UniOSCClassBasedDemo.OSCPort

16.6.2.10 int UniOSCClassBasedDemo.OSCPortOUT

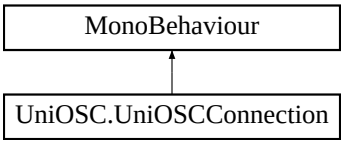
16.6.2.11 bool UniOSCClassBasedDemo.sendData

16.6.2.12 float UniOSCClassBasedDemo.sendInterval =1000

16.7 UniOSC.UniOSCConnection Class Reference

This class is responsible for all the network related tasks.

Inheritance diagram for UniOSC.UniOSCConnection:



Public Member Functions

- void ValidateOscInIPAddress ()
- void ValidateOscOutIPAddress ()

- void Awake ()
- IEnumerator Start ()
- void ConnectOSC ()

  creates internally an UniOSCReciver which handles all the Network setup.
- void DisconnectOSC ()

  Disconnects and destroys the OSCConnection.
- void Force_SetupChanged_IN ()
- void ConnectOSCOut ()

  Connects the OSC out.
- void DisconnectOSCOut ()

  Disconnects and release the OSC out connection.
- void Force_SetupChanged_OUT ()
- void RenderGUI ()

  Renders the GUI of a OSCConnection in the GameView.
- void SendOSCMessage (object sender, UniOSCEventArgs args)

  Sends the OSC message.
- void SendTestMessage ()

  Sends the test message.
- void SendSessionData ()

  Sends the session data.

Static Public Member Functions

- static void Init ()

  Init this instance.
- static void Update_AvailableOSCSettings ()

  Updates the available ports.

Public Attributes

- bool autoConnectOSCIn = true
- bool oscOut = true
- bool autoConnectOSCOut = true
- bool foldoutOSCOut = true
- bool foldoutOSCIn = true
- bool redrawFlag
- bool dispatchOSC = true
- bool dispatchOSCOut = true
- List< UniOSCMappingFileObj > oscMappingFileObjList = new List<UniOSCMappingFileObj>()
- List< UniOSCSessionFileObj > oscSessionFileObjList = new List<UniOSCSessionFileObj>()
- bool SendSessionDataOnStart

Static Public Attributes

- static string localIPAddress = null
- static bool isOSCLearning = false
- static bool isEditorEnabled = false

Properties

- static List< UniOSCConnection > Instances [get]
- static List< int > AvailableINPorts [get]
- static List< int > AvailableOUTPorts [get]
- static List< string > AvailableOUTIPAddresses [get]
- OSCsharp.Net.TransmissionType transmissionTypeIn [get, set]
- OSCsharp.Net.TransmissionType transmissionTypeOut [get, set]
- int oscPort [get, set]
- string oscInIPAddress [get, set]
- IPAddress oscInIPAddressAsIPAddress [get]
- bool hasValidOscIPAddress [get]
- string oscOutIPAddress [get, set]
- IPAddress oscOutIPAddressAsIPAddress [get]
- bool hasValidOscOutIPAddress [get]
- int oscOutPort [get, set]
- bool isConnected [get]
- bool isConnectedOut [get]
- bool hasOSCMappingFileAttached [get]
- bool hasOSCSessionFileAttached [get]

Events

- EventHandler< UniOSCEventArgs > OSCMessageReceivedRaw
- EventHandler< UniOSCEventArgs > OSCMessageReceived
- EventHandler< UniOSCEventArgs > OSCMessageSend
- Action< UniOSCConnection > ConnectionInStatusChange
- Action< UniOSCConnection > ConnectionOutStatusChange

16.7.1 Detailed Description

This class is responsible for all the network related tasks.

It is a wrapper for OSCsharp and handles the event system for the Unity components.

16.7.2 Member Function Documentation

16.7.2.1 void UniOSC.UniOSCConnection.Awake ( )

16.7.2.2 void UniOSC.UniOSCConnection.ConnectOSC ( )

creates internally an UniOSCReciver which handles all the Network setup.

Called from GUI/Inspector

16.7.2.3 void UniOSC.UniOSCConnection.ConnectOSCOut ( )

Connects the OSC out.

16.7.2.4 void UniOSC.UniOSCConnection.DisconnectOSC ( )

Disconnects and destroys the OSCConnection.

16.7.2.5 void UniOSC.UniOSCConnection.DisconnectOSCOut ( )

Disconnects and release the OSC out connection.

16.7.2.6 void UniOSC.UniOSCConnection.Force_SetupChanged_IN ( )

16.7.2.7 void UniOSC.UniOSCConnection.Force_SetupChanged_OUT ( )

16.7.2.8 static void UniOSC.UniOSCConnection.Init ( ) [static]

Init this instance.

Is called from Awake and OSCAutoRun

16.7.2.9 void UniOSC.UniOSCConnection.RenderGUI ( )

Renders the GUI of a OSCConnection in the GameView.

This is different from the rendering in the editor/inspector

16.7.2.10 void UniOSC.UniOSCConnection.SendOSCMessage (object *sender,* UniOSCEventArgs *args*)

Sends the OSC message.

Parameters

sender	Sender.
args	UniOSCEventArgs

16.7.2.11 void UniOSC.UniOSCConnection.SendSessionData ( )

Sends the session data.

This is useful for updating the GUI of TouOSC for example with the last data values from incomming OSC messages. You have to add a OSC Session file to the OSCConnection to use this feature.

16.7.2.12 void UniOSC.UniOSCConnection.SendTestMessage ( )

Sends the test message.

Only for testing the OSC Out connection.

16.7.2.13 IEnumerator UniOSC.UniOSCConnection.Start ( )

16.7.2.14 static void UniOSC.UniOSCConnection.Update_AvailableOSCSettings ( ) [static]

Updates the available ports.

Should be called when a OSCConnection changes the Port.

16.7.2.15 void UniOSC.UniOSCConnection.ValidateOscInIPAddress ( )

16.7.2.16 void UniOSC.UniOSCConnection.ValidateOscOutIPAddress ( )

16.7.3 Member Data Documentation

16.7.3.1 bool UniOSC.UniOSCConnection.autoConnectOSCIn = true

16.7.3.2 bool UniOSC.UniOSCConnection.autoConnectOSCOut = true

16.7.3.3 bool UniOSC.UniOSCConnection.dispatchOSC = true

16.7.3.4 bool UniOSC.UniOSCConnection.dispatchOSCOut = true

16.7.3.5 bool UniOSC.UniOSCConnection.foldoutOSCIn = true

16.7.3.6 bool UniOSC.UniOSCConnection.foldoutOSCOut = true

16.7.3.7 bool UniOSC.UniOSCConnection.isEditorEnabled = false [static]

16.7.3.8 bool UniOSC.UniOSCConnection.isOSCLearning = false [static]

16.7.3.9 string UniOSC.UniOSCConnection.localIPAddress = null [static]

16.7.3.10 List<**UniOSCMappingFileObj**> UniOSC.UniOSCConnection.oscMappingFileObjList = new List<**UniOSCMappingFileObj**>()

16.7.3.11 bool UniOSC.UniOSCConnection.oscOut = true

16.7.3.12 List<**UniOSCSessionFileObj**> UniOSC.UniOSCConnection.oscSessionFileObjList = new List<**UniOSCSessionFileObj**>()

16.7.3.13 bool UniOSC.UniOSCConnection.redrawFlag

16.7.3.14 bool UniOSC.UniOSCConnection.SendSessionDataOnStart

16.7.4 Property Documentation

16.7.4.1 List<int> UniOSC.UniOSCConnection.AvailableINPorts [static],[get]

16.7.4.2 List<string> UniOSC.UniOSCConnection.AvailableOUTIPAddresses [static],[get]

16.7.4.3 List<int> UniOSC.UniOSCConnection.AvailableOUTPorts [static],[get]

16.7.4.4 bool UniOSC.UniOSCConnection.hasOSCMappingFileAttached [get]

16.7.4.5 bool UniOSC.UniOSCConnection.hasOSCSessionFileAttached [get]

16.7.4.6 bool UniOSC.UniOSCConnection.hasValidOscIPAddress [get]

16.7.4.7 bool UniOSC.UniOSCConnection.hasValidOscOutIPAddress [get]

16.7.4.8 List<**UniOSCConnection**> UniOSC.UniOSCConnection.Instances [static],[get]

16.7.4.9 bool UniOSC.UniOSCConnection.isConnected [get]

16.7.4.10 bool UniOSC.UniOSCConnection.isConnectedOut [get]

16.7.4.11 string UniOSC.UniOSCConnection.oscInIPAddress [get],[set]

16.7.4.12 IPAddress UniOSC.UniOSCConnection.oscInIPAddressAsIPAddress [get]

16.7.4.13 string UniOSC.UniOSCConnection.oscOutIPAddress [get], [set]

16.7.4.14 IPAddress UniOSC.UniOSCConnection.oscOutIPAddressAsIPAddress [get]

16.7.4.15 int UniOSC.UniOSCConnection.oscOutPort [get], [set]

16.7.4.16 int UniOSC.UniOSCConnection.oscPort [get], [set]

16.7.4.17 OSCsharp.Net.TransmissionType UniOSC.UniOSCConnection.transmissionTypeIn [get], [set]

16.7.4.18 OSCsharp.Net.TransmissionType UniOSC.UniOSCConnection.transmissionTypeOut [get], [set]

16.7.5 Event Documentation

16.7.5.1 Action<UniOSCConnection> UniOSC.UniOSCConnection.ConnectionInStatusChange

16.7.5.2 Action<UniOSCConnection> UniOSC.UniOSCConnection.ConnectionOutStatusChange

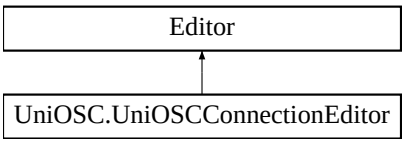
16.7.5.3 EventHandler<UniOSCEventArgs> UniOSC.UniOSCConnection.OSCMessageReceived

16.7.5.4 EventHandler<UniOSCEventArgs> UniOSC.UniOSCConnection.OSCMessageReceivedRaw

16.7.5.5 EventHandler<UniOSCEventArgs> UniOSC.UniOSCConnection.OSCMessageSend

16.8 UniOSC.UniOSCConnectionEditor Class Reference

Inheritance diagram for UniOSC.UniOSCConnectionEditor:



Public Member Functions

- override void OnInspectorGUI ()

Static Public Member Functions

- static void LoadTextures ()
- static void Show (string label, SerializedProperty list)
- static void ShowOSCReciverStatus (UniOSCConnection oscConnection)

Static Public Attributes

- static Texture2D texTestMessage
- static Texture2D texON
- static Texture2D texOFF

Protected Member Functions

- void ForceUpdate ()

Protected Attributes

- string[] _TransmissionTypes
- int _TransmissionTypeIndex = 0
- string _oldOSCMulticastIPAddress
- string _currOSCMulticastIPAddress
- bool _isValidOSCMulticastIPAddress
- string[] _TransmissionTypesOut
- int _TransmissionTypeIndexOut = 0
- string _oldOSCMulticastIPAddressOut
- string _currOSCMulticastIPAddressOut
- bool _isValidOSCMulticastIPAddressOut

16.8.1 Member Function Documentation

16.8.1.1 void UniOSC.UniOSCConnectionEditor.ForceUpdate ( ) [protected]

16.8.1.2 static void UniOSC.UniOSCConnectionEditor.LoadTextures ( ) [static]

16.8.1.3 override void UniOSC.UniOSCConnectionEditor.OnInspectorGUI ( )

16.8.1.4 static void UniOSC.UniOSCConnectionEditor.Show (string *label,* SerializedProperty *list*) [static]

16.8.1.5 static void UniOSC.UniOSCConnectionEditor.ShowOSCReciverStatus (UniOSCConnection *oscConnection*) [static]

16.8.2 Member Data Documentation

16.8.2.1 string UniOSC.UniOSCConnectionEditor._currOSCMulticastIPAddress [protected]

16.8.2.2 string UniOSC.UniOSCConnectionEditor._currOSCMulticastIPAddressOut [protected]

16.8.2.3 bool UniOSC.UniOSCConnectionEditor._isValidOSCMulticastIPAddress [protected]

16.8.2.4 bool UniOSC.UniOSCConnectionEditor._isValidOSCMulticastIPAddressOut [protected]

16.8.2.5 string UniOSC.UniOSCConnectionEditor._oldOSCMulticastIPAddress [protected]

16.8.2.6 string UniOSC.UniOSCConnectionEditor._oldOSCMulticastIPAddressOut [protected]

16.8.2.7 int UniOSC.UniOSCConnectionEditor._TransmissionTypeIndex = 0 [protected]

16.8.2.8 int UniOSC.UniOSCConnectionEditor._TransmissionTypeIndexOut = 0 [protected]

16.8.2.9 string [] UniOSC.UniOSCConnectionEditor._TransmissionTypes [protected]

16.8.2.10 string [] UniOSC.UniOSCConnectionEditor._TransmissionTypesOut [protected]

16.8.2.11 Texture2D UniOSC.UniOSCConnectionEditor.texOFF [static]

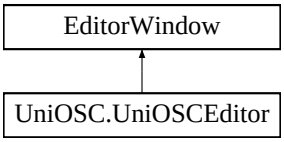
16.8.2.12 Texture2D UniOSC.UniOSCConnectionEditor.texON [static]

16.8.2.13 Texture2D UniOSC.UniOSCConnectionEditor.texTestMessage [static]

16.9 UniOSC.UniOSCEditor Class Reference

Editor for the administration of OSCconnections, mapping files.

Inheritance diagram for UniOSC.UniOSCEditor:



Public Member Functions

- void OnEnable ()
- void OnDisable ()
- void OnHierarchyChange ()

Static Public Member Functions

- static void Init ()

  Init this instance.
- static void OSCLearning (bool flag)

  When entering the OSC learning mode the editor connects all mapping files to the event system so the OSC address for a mapping item can be recorded.

Public Attributes

- const float TRACEWIDTH = 250f

Properties

- static bool isOSCLearning `[get]`
- static UniOSCEditor Instance `[get]`
- static bool IsOpen `[get]`

Events

- static EventHandler< UniOSCEventArgs > OSCMessageReceived

16.9.1 Detailed Description

Editor for the administration of OSCconnections, mapping files.

You can also trace the OSC data flow .

16.9.2 Member Function Documentation

16.9.2.1 static void UniOSC.UniOSCEditor.Init ( ) `[static]`

Init this instance.

Called everytime the editor is opened or when we have to update the editor (After creating a OSCConnection,hit the 'Refresh' button)

16.9.2.2 void UniOSC.UniOSCEditor.OnDisable ( )

16.9.2.3 void UniOSC.UniOSCEditor.OnEnable ( )

16.9.2.4 void UniOSC.UniOSCEditor.OnHierarchyChange ( )

16.9.2.5 static void UniOSC.UniOSCEditor.OSCLearning (bool *flag*) [static]

When entering the OSC learning mode the editor connects all mapping files to the event system so the OSC address for a mapping item can be recorded.

Parameters

flag	If set to `true` flag.

16.9.3 Member Data Documentation

16.9.3.1 const float UniOSC.UniOSCEditor.TRACEWIDTH = 250f

16.9.4 Property Documentation

16.9.4.1 UniOSCEditor UniOSC.UniOSCEditor.Instance [static],[get]

16.9.4.2 bool UniOSC.UniOSCEditor.IsOpen [static],[get]

16.9.4.3 bool UniOSC.UniOSCEditor.isOSCLearning [static],[get]

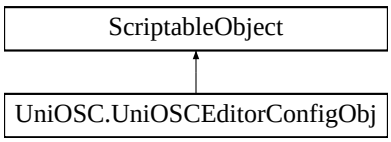
16.9.5 Event Documentation

16.9.5.1 EventHandler<UniOSCEventArgs> UniOSC.UniOSCEditor.OSCMessageReceived [static]

16.10 UniOSC.UniOSCEditorConfigObj Class Reference

UniOSC editor config object.

Inheritance diagram for UniOSC.UniOSCEditorConfigObj:



Public Member Functions

- void OnEnable ()

Public Attributes

- int selectedMappingFileObjIndex

- int selectedSessionFileObjIndex
- Vector2 configTraceScrollpos
- bool isOSCTracing
- bool isOSCLearning
- bool isEditorEnabled
- bool isLastMessageTracing
- GUISkin mySkin
- GUIStyle learnStyle
- List< UniOSCMappingFileObj > OSCMappingFileObjList
- List< UniOSCSessionFileObj > OSCSessionFileObjList
- int toolbarInt = 0
- Texture2D tex_LearnFrame
- Texture2D tex_logo

16.10.1 Detailed Description

UniOSC editor config object.

Storage of all the UniOSCEditor settings

16.10.2 Member Function Documentation

16.10.2.1 void UniOSC.UniOSCEditorConfigObj.OnEnable ( )

16.10.3 Member Data Documentation

16.10.3.1 Vector2 UniOSC.UniOSCEditorConfigObj.configTraceScrollpos

16.10.3.2 bool UniOSC.UniOSCEditorConfigObj.isEditorEnabled

16.10.3.3 bool UniOSC.UniOSCEditorConfigObj.isLastMessageTracing

16.10.3.4 bool UniOSC.UniOSCEditorConfigObj.isOSCLearning

16.10.3.5 bool UniOSC.UniOSCEditorConfigObj.isOSCTracing

16.10.3.6 GUIStyle UniOSC.UniOSCEditorConfigObj.learnStyle

16.10.3.7 GUISkin UniOSC.UniOSCEditorConfigObj.mySkin

16.10.3.8 List<UniOSCMappingFileObj> UniOSC.UniOSCEditorConfigObj.OSCMappingFileObjList

16.10.3.9 List<UniOSCSessionFileObj> UniOSC.UniOSCEditorConfigObj.OSCSessionFileObjList

16.10.3.10 int UniOSC.UniOSCEditorConfigObj.selectedMappingFileObjIndex

16.10.3.11 int UniOSC.UniOSCEditorConfigObj.selectedSessionFileObjIndex

16.10.3.12 Texture2D UniOSC.UniOSCEditorConfigObj.tex_LearnFrame

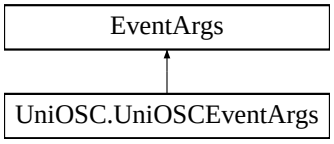
16.10.3.13 Texture2D UniOSC.UniOSCEditorConfigObj.tex_logo

16.10.3.14 int UniOSC.UniOSCEditorConfigObj.toolbarInt = 0

16.11 UniOSC.UniOSCEventArgs Class Reference

A wrapper to a OscMessage class to also store the port and have a quick way to access the message address.

Inheritance diagram for UniOSC.UniOSCEventArgs:



Public Member Functions

- UniOSCEventArgs (int port, OscPacket packet)

Public Attributes

- string IPAddress

Properties

- OscPacket Packet `[get]`
- string Address `[get]`
- int Port `[get]`
- int Group `[get]`
- string AddressRoot `[get]`
- int AddressIndex `[get]`

16.11.1 Detailed Description

A wrapper to a OscMessage class to also store the port and have a quick way to access the message address.

UniOSC use this class for the internal communication

this is a paragraph

See also

UniOSC.OSCEventTarget

UniOSC.OSCEventTarget

16.11.2 Constructor & Destructor Documentation

16.11.2.1 UniOSC.UniOSCEventArgs.UniOSCEventArgs (int *port*, OscPacket *packet*)

16.11.3 Member Data Documentation

16.11.3.1 string UniOSC.UniOSCEventArgs.IPAddress

16.11.4 Property Documentation

16.11.4.1 string UniOSC.UniOSCEventArgs.Address [get]

16.11.4.2 int UniOSC.UniOSCEventArgs.AddressIndex [get]

16.11.4.3 string UniOSC.UniOSCEventArgs.AddressRoot [get]

16.11.4.4 int UniOSC.UniOSCEventArgs.Group [get]

16.11.4.5 OscPacket UniOSC.UniOSCEventArgs.Packet [get]

16.11.4.6 int UniOSC.UniOSCEventArgs.Port [get]

16.12 UniOSC.UniOSCEventDispatcher Class Reference

This is the abstract class you should subclass from when you want to sent OSC data

Inheritance diagram for UniOSC.UniOSCEventDispatcher:

Public Member Functions

- virtual void Awake ()
- virtual void Start ()
- virtual void OnEnable ()
- void ForceSetupChange (bool resetMessage)

  This method is mainly for the EventDispatcherEditor to force an update of the internal message setup
- virtual void OnDestroy ()
- virtual void OnDisable ()
- void SetBundleMode (bool _isBundle)

  Sets the bundle mode.
- virtual void SendOSCMessage ()

  Sends the OSC message.
- void AppendData (object _data)

  Appends the data.
- void ClearData ()
- void StartSendIntervalTimer ()
- void StopSendIntervalTimer ()

Public Attributes

- float sendInterval =100

Protected Member Functions

- void _OnTimedEvent (object source, System.Timers.ElapsedEventArgs e)
- virtual void _Update ()
- void _OnConnectionOutStatusChanged (UniOSCConnection con)
- void _ConnectToOSCConnections ()
- void _DisconnectFromOSCConnections ()
- void _SetupOSCMessage (bool _isBundle)
- void _SendOSCMessage (UniOSCEventArgs args)

Protected Attributes

- string _oscOutAddress = "/"
- string _oscOutIPAddress
- int _oscOutPort
- bool _useExplicitConnection
- UniOSCConnection _explicitConnection
- OscPacket _OSCpkg
- UniOSCEventArgs _OSCeArg
- System.Timers.Timer _sendIntervalTimer
- bool _isOSCDirty
- object _mylock = new object()
- bool _drawDefaultInspector = true
- List< UniOSCConnection > _myOSCConnections = new List<UniOSCConnection>()

Properties

- string oscOutAddress `[get, set]`
- string oscOutIPAddress `[get, set]`
- int oscOutPort `[get, set]`
- bool isBundle `[get]`
- bool useExplicitConnection `[get, set]`
- UniOSCConnection explicitConnection `[get, set]`

16.12.1 Detailed Description

This is the abstract class you should subclass from when you want to sent OSC data

16.12.2 Member Function Documentation

16.12.2.1 void UniOSC.UniOSCEventDispatcher._ConnectToOSCConnections ( ) [protected]

16.12.2.2 void UniOSC.UniOSCEventDispatcher._DisconnectFromOSCConnections ( ) [protected]

16.12.2.3 void UniOSC.UniOSCEventDispatcher._OnConnectionOutStatusChanged (UniOSCConnection *con*) [protected]

16.12.2.4 void UniOSC.UniOSCEventDispatcher._OnTimedEvent (object *source*, System.Timers.ElapsedEventArgs *e*) [protected]

16.12.2.5 void UniOSC.UniOSCEventDispatcher._SendOSCMessage (UniOSCEventArgs *args*) [protected]

16.12.2.6 void UniOSC.UniOSCEventDispatcher._SetupOSCMessage (bool *_isBundle*) [protected]

16.12.2.7 virtual void UniOSC.UniOSCEventDispatcher._Update ( ) [protected], [virtual]

Reimplemented in UniOSC.UniOSCTransformSender.

16.12.2.8 void UniOSC.UniOSCEventDispatcher.AppendData (object *_data*)

Appends the data.

Depending on your bundle mode the AppendData method works in a different way. If you use bundles you can append multiple OscMessages. If you don't use bundles (default) you append data to your OscMessage We only can append data types that are supported by the OSC specification: (Int32,Int64,Single,Double,String,Byte[],Osc↩ TimeTag,Char,Color,Boolean)

Parameters

_data	_data.

16.12.2.9 virtual void UniOSC.UniOSCEventDispatcher.Awake ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventDispatcherSlider, UniOSC.UniOSCEventDispatcherToggle, UniOSC.UniOSCEventDispatcherButton, UniOSC.UniOSCEventDispatcherMultiConnectionSender, UniOSC.UniOSCEventDispatcherMultiAddressSender, and UniOSC.UniOSCEventDispatcherImplementation.

16.12.2.10 void UniOSC.UniOSCEventDispatcher.ClearData ( )

16.12.2.11 void UniOSC.UniOSCEventDispatcher.ForceSetupChange (bool *resetMessage*)

This method is mainly for the EventDispatcherEditor to force an update of the internal message setup

Parameters

resetMessage	

16.12.2.12 virtual void UniOSC.UniOSCEventDispatcher.OnDestroy ( ) [virtual]

16.12.2.13 virtual void UniOSC.UniOSCEventDispatcher.OnDisable ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventDispatcherMultiAddressSender, UniOSC.UniOSCEventDispatcherMultiConnectionSender, UniOSC.UniOSCEventDispatcherImplementation, UniOSC.UniOSCEventDispatcherSlider, UniOSC.UniOSCEventDispatcherToggle, UniOSC.UniOSCEventDispatcherButton, and UniOSC.UniOSCTransformSender.

16.12.2.14 virtual void UniOSC.UniOSCEventDispatcher.OnEnable ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventDispatcherSlider, UniOSC.UniOSCEventDispatcherToggle, UniOSC.UniOSCEventDispatcherButton, UniOSC.UniOSCEventDispatcherMultiConnectionSender, UniOSC.UniOSCEventDispatcherMultiAddressSender, UniOSC.UniOSCEventDispatcherImplementation, and UniOSC.UniOSCTransformSender.

16.12.2.15 virtual void UniOSC.UniOSCEventDispatcher.SendOSCMessage ( ) [virtual]

Sends the OSC message.

Reimplemented in UniOSC.UniOSCEventDispatcherSlider.

16.12.2.16 void UniOSC.UniOSCEventDispatcher.SetBundleMode (bool *_isBundle*)

Sets the bundle mode.

You can change the mode at any time but you have to be careful what data you trying to append with UniOSC.UniOSCEventDispatcher.AppendData(object)

Parameters

_isBundle	If set to `true` is bundle.

16.12.2.17 virtual void UniOSC.UniOSCEventDispatcher.Start ( ) [virtual]

16.12.2.18 void UniOSC.UniOSCEventDispatcher.StartSendIntervalTimer ( )

16.12.2.19 void UniOSC.UniOSCEventDispatcher.StopSendIntervalTimer ( )

16.12.3 Member Data Documentation

16.12.3.1 bool UniOSC.UniOSCEventDispatcher._drawDefaultInspector = true [protected]

16.12.3.2 UniOSCConnection UniOSC.UniOSCEventDispatcher._explicitConnection [protected]

16.12.3.3 bool UniOSC.UniOSCEventDispatcher._isOSCDirty [protected]

16.12.3.4 object UniOSC.UniOSCEventDispatcher._mylock = new object() [protected]

16.12.3.5 List<UniOSCConnection> UniOSC.UniOSCEventDispatcher._myOSCConnections = new List<UniOSCConnection>() [protected]

16.12.3.6 UniOSCEventArgs UniOSC.UniOSCEventDispatcher._OSCeArg [protected]

16.12.3.7 string UniOSC.UniOSCEventDispatcher._oscOutAddress = "/" [protected]

16.12.3.8 string UniOSC.UniOSCEventDispatcher._oscOutIPAddress [protected]

16.12.3.9 int UniOSC.UniOSCEventDispatcher._oscOutPort [protected]

16.12.3.10 OscPacket UniOSC.UniOSCEventDispatcher._OSCpkg [protected]

16.12.3.11 System.Timers.Timer UniOSC.UniOSCEventDispatcher._sendIntervalTimer [protected]

16.12.3.12 bool UniOSC.UniOSCEventDispatcher._useExplicitConnection [protected]

16.12.3.13 float UniOSC.UniOSCEventDispatcher.sendInterval =100

16.12.4 Property Documentation

16.12.4.1 UniOSCConnection UniOSC.UniOSCEventDispatcher.explicitConnection [get], [set]

16.12.4.2 bool UniOSC.UniOSCEventDispatcher.isBundle [get]

16.12.4.3 string UniOSC.UniOSCEventDispatcher.oscOutAddress [get], [set]

16.12.4.4 string UniOSC.UniOSCEventDispatcher.oscOutIPAddress [get], [set]

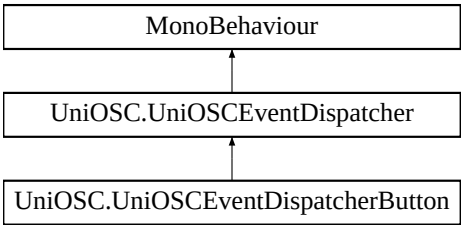
16.12.4.5 int UniOSC.UniOSCEventDispatcher.oscOutPort [get], [set]

16.12.4.6 bool UniOSC.UniOSCEventDispatcher.useExplicitConnection [get], [set]

16.13 UniOSC.UniOSCEventDispatcherButton Class Reference

Dispatcher button that forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherButton:



Public Member Functions

- override void Awake ()
- override void OnEnable ()
- override void OnDisable ()
- void SendOSCMessageDown ()

  Sends the OSC message with the downOSCDataValue.
- void SendOSCMessageUp ()

  Sends the OSC message with the upOSCDataValue.

Public Attributes

- float downOSCDataValue =1
- float upOSCDataValue =0
- bool showGUI
- float xPos
- float yPos

Additional Inherited Members

16.13.1 Detailed Description

Dispatcher button that forces a OSCConnection to send a OSC Message.

Two separate states: Down and Up

16.13.2 Member Function Documentation

16.13.2.1 override void UniOSC.UniOSCEventDispatcherButton.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.13.2.2 override void UniOSC.UniOSCEventDispatcherButton.OnDisable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.13.2.3 override void UniOSC.UniOSCEventDispatcherButton.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.13.2.4 void UniOSC.UniOSCEventDispatcherButton.SendOSCMessageDown ( )

Sends the OSC message with the downOSCDataValue.

16.13.2.5 void UniOSC.UniOSCEventDispatcherButton.SendOSCMessageUp ( )

Sends the OSC message with the upOSCDataValue.

16.13.3 Member Data Documentation

16.13.3.1 float UniOSC.UniOSCEventDispatcherButton.downOSCDataValue =1

16.13.3.2 bool UniOSC.UniOSCEventDispatcherButton.showGUI

16.13.3.3 float UniOSC.UniOSCEventDispatcherButton.upOSCDataValue =0

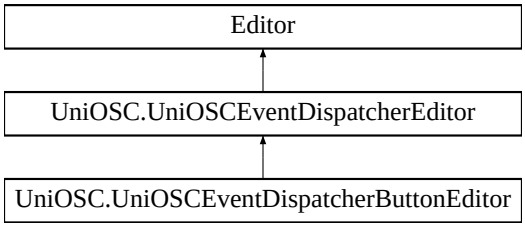
16.13.3.4 float UniOSC.UniOSCEventDispatcherButton.xPos

16.13.3.5 float UniOSC.UniOSCEventDispatcherButton.yPos

16.14 UniOSC.UniOSCEventDispatcherButtonEditor Class Reference

Uni OSC event dispatcher button editor.

Inheritance diagram for UniOSC.UniOSCEventDispatcherButtonEditor:



Public Member Functions

- override void OnEnable ()
- override void OnInspectorGUI ()

Protected Attributes

- SerializedProperty downOSCDataValueProp
- SerializedProperty upOSCDataValueProp
- SerializedProperty ShowGUIProp
- SerializedProperty xProp
- SerializedProperty yProp

Additional Inherited Members

16.14.1 Detailed Description

Uni OSC event dispatcher button editor.

16.14.2 Member Function Documentation

16.14.2.1 override void UniOSC.UniOSCEventDispatcherButtonEditor.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcherEditor.

16.14.2.2 override void UniOSC.UniOSCEventDispatcherButtonEditor.OnInspectorGUI ( )

16.14.3 Member Data Documentation

16.14.3.1 SerializedProperty UniOSC.UniOSCEventDispatcherButtonEditor.downOSCDataValueProp [protected]

16.14.3.2 SerializedProperty UniOSC.UniOSCEventDispatcherButtonEditor.ShowGUIProp [protected]

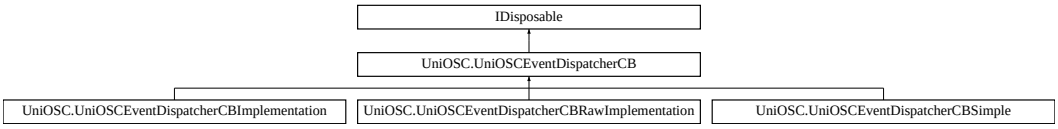
16.14.3.3 SerializedProperty UniOSC.UniOSCEventDispatcherButtonEditor.upOSCDataValueProp [protected]

16.14.3.4 SerializedProperty UniOSC.UniOSCEventDispatcherButtonEditor.xProp [protected]

16.14.3.5 SerializedProperty UniOSC.UniOSCEventDispatcherButtonEditor.yProp [protected]

16.15 UniOSC.UniOSCEventDispatcherCB Class Reference

Inheritance diagram for UniOSC.UniOSCEventDispatcherCB:



Public Member Functions

- UniOSCEventDispatcherCB (string __oscOutAddress, string __oscOutIPAddress, int __oscPort)

  Sets the explicitConnection property to the new UniOSCConnection when we have turned on the useExplicit↩ Connection mode
- UniOSCEventDispatcherCB (string __oscOutAddress, UniOSCConnection __explicitConnection)
- virtual void Awake ()
- virtual void Enable ()

  Enable this instance.
- virtual void Disable ()

  Disable this instance.
- virtual void OnDestroy ()
- void ForceSetupChange (bool resetMessage)

  This method forces an reconfiguration with the current settings.
- void SetBundleMode (bool _isBundle)

  Sets the bundle mode.

- void SendOSCMessage ()

  Sends the OSC message.
- void AppendData (object _data)

  Appends the data.
- void ClearData ()

  Clears all data.
- void UpdateDataAt (int index, object value)
- void StartSendIntervalTimer ()

  Starts the send interval timer.
- void StopSendIntervalTimer ()

  Stops the send interval timer.
- void Dispose ()

  Performs application-defined tasks associated with freeing, releasing, or resetting resources.

Public Attributes

- float sendInterval =100

Protected Member Functions

- void _OnTimedEvent (object source, System.Timers.ElapsedEventArgs e)
- void _OnConnectionOutStatusChanged (UniOSCConnection con)
- void _ConnectToOSCConnections ()
- void _DisconnectFromOSCConnections ()
- void _SetupOSCMessage (bool _isBundle)
- void _SendOSCMessage (UniOSCEventArgs args)

Protected Attributes

- OscPacket _OSCpkg
- UniOSCEventArgs _OSCeArg
- System.Timers.Timer _sendIntervalTimer
- bool _isOSCDirty
- object _mylock = new object()
- string _oscOutAddress
- string _oscOutIPAddress
- int _oscOutPort
- bool _useExplicitConnection
- UniOSCConnection _explicitConnection

Properties

- bool isEnabled `[get]`
- string oscOutAddress `[get, set]`
- string oscOutIPAddress `[get, set]`
- int oscOutPort `[get, set]`
- bool isBundle `[get]`
- bool useExplicitConnection `[get, set]`
- UniOSCConnection explicitConnection `[get, set]`

16.15.1 Constructor & Destructor Documentation

16.15.1.1 UniOSC.UniOSCEventDispatcherCB.UniOSCEventDispatcherCB (string *__oscOutAddress,* string *__oscOutIPAddress,* int *__oscPort*)

Sets the explicitConnection property to the new UniOSCConnection when we have turned on the useExplicit↩ Connection mode

Parameters

newCon	

16.15.1.2 UniOSC.UniOSCEventDispatcherCB.UniOSCEventDispatcherCB (string *__oscOutAddress,* UniOSCConnection *__explicitConnection*)

16.15.2 Member Function Documentation

16.15.2.1 void UniOSC.UniOSCEventDispatcherCB._ConnectToOSCConnections ( ) [protected]

16.15.2.2 void UniOSC.UniOSCEventDispatcherCB._DisconnectFromOSCConnections ( ) [protected]

16.15.2.3 void UniOSC.UniOSCEventDispatcherCB._OnConnectionOutStatusChanged (UniOSCConnection *con*) [protected]

16.15.2.4 void UniOSC.UniOSCEventDispatcherCB._OnTimedEvent (object *source,* System.Timers.ElapsedEventArgs *e*) [protected]

16.15.2.5 void UniOSC.UniOSCEventDispatcherCB._SendOSCMessage (UniOSCEventArgs *args*) [protected]

16.15.2.6 void UniOSC.UniOSCEventDispatcherCB._SetupOSCMessage (bool *_isBundle*) [protected]

16.15.2.7 void UniOSC.UniOSCEventDispatcherCB.AppendData (object *_data*)

Appends the data.

Depending on your bundle mode the AppendData method works in a different way. If you use bundles you can append multiple OscMessages. If you don't use bundles (default) you append data to your OscMessage We only can append data types that are supported by the OSC specification: (Int32,Int64,Single,Double,String,Byte[],Osc↩TimeTag,Char,Color,Boolean)

Parameters

_data	_data.

16.15.2.8 virtual void UniOSC.UniOSCEventDispatcherCB.Awake ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventDispatcherCBSimple, UniOSC.UniOSCEventDispatcherCBRaw↩Implementation, and UniOSC.UniOSCEventDispatcherCBImplementation.

16.15.2.9 void UniOSC.UniOSCEventDispatcherCB.ClearData ( )

Clears all data.

16.15.2.10 virtual void UniOSC.UniOSCEventDispatcherCB.Disable ( ) [virtual]

Disable this instance.

Reimplemented in UniOSC.UniOSCEventDispatcherCBImplementation, UniOSC.UniOSCEventDispatcherCB↩Simple, and UniOSC.UniOSCEventDispatcherCBRawImplementation.

16.15.2.11 void UniOSC.UniOSCEventDispatcherCB.Dispose ( )

Performs application-defined tasks associated with freeing, releasing, or resetting resources.

Call Dispose when you are finished using the UniOSC.UniOSCEventDispatcherCB. The Dispose method leaves the UniOSC.UniOSCEventDispatcherCB in an unusable state. After calling Dispose, you must release all references to the UniOSC.UniOSCEventDispatcherCB so the garbage collector can reclaim the memory that the UniOSC.UniOSCEventDispatcherCB was occupying.

16.15.2.12 virtual void UniOSC.UniOSCEventDispatcherCB.Enable ( ) [virtual]

Enable this instance.

Reimplemented in UniOSC.UniOSCEventDispatcherCBSimple, UniOSC.UniOSCEventDispatcherCBImplementation, and UniOSC.UniOSCEventDispatcherCBRawImplementation.

16.15.2.13 void UniOSC.UniOSCEventDispatcherCB.ForceSetupChange (bool *resetMessage*)

This method forces an reconfiguration with the current settings.

Normaly you don't need to call this method. You can specify if you want to reset all the data from the OSC Message/Bundle that is used when sending data out.

Parameters

resetMessage	

16.15.2.14 virtual void UniOSC.UniOSCEventDispatcherCB.OnDestroy ( ) [virtual]

16.15.2.15 void UniOSC.UniOSCEventDispatcherCB.SendOSCMessage ( )

Sends the OSC message.

16.15.2.16 void UniOSC.UniOSCEventDispatcherCB.SetBundleMode (bool *_isBundle*)

Sets the bundle mode.

You can change the mode at any time but you have to be careful what data you trying to append with UniOSC.UniOSCEventDispatcherCB.AppendData(object)

Parameters

_isBundle	If set to `true` is bundle.

16.15.2.17 void UniOSC.UniOSCEventDispatcherCB.StartSendIntervalTimer ( )

Starts the send interval timer.

This is useful when you need to send OSC data frequently. With the sendInterval property you specify the interval in milliseconds

16.15.2.18 void UniOSC.UniOSCEventDispatcherCB.StopSendIntervalTimer ( )

Stops the send interval timer.

16.15.2.19 void UniOSC.UniOSCEventDispatcherCB.UpdateDataAt (int *index*, object *value*)

16.15.3 Member Data Documentation

16.15.3.1 UniOSCConnection UniOSC.UniOSCEventDispatcherCB._explicitConnection [protected]

16.15.3.2 bool UniOSC.UniOSCEventDispatcherCB._isOSCDirty [protected]

16.15.3.3 object UniOSC.UniOSCEventDispatcherCB._mylock = new object() [protected]

16.15.3.4 UniOSCEventArgs UniOSC.UniOSCEventDispatcherCB._OSCeArg [protected]

16.15.3.5 string UniOSC.UniOSCEventDispatcherCB._oscOutAddress [protected]

16.15.3.6 string UniOSC.UniOSCEventDispatcherCB._oscOutIPAddress [protected]

16.15.3.7 int UniOSC.UniOSCEventDispatcherCB._oscOutPort [protected]

16.15.3.8 OscPacket UniOSC.UniOSCEventDispatcherCB._OSCpkg [protected]

16.15.3.9 System.Timers.Timer UniOSC.UniOSCEventDispatcherCB._sendIntervalTimer [protected]

16.15.3.10 bool UniOSC.UniOSCEventDispatcherCB._useExplicitConnection [protected]

16.15.3.11 float UniOSC.UniOSCEventDispatcherCB.sendInterval =100

16.15.4 Property Documentation

16.15.4.1 UniOSCConnection UniOSC.UniOSCEventDispatcherCB.explicitConnection [get], [set]

16.15.4.2 bool UniOSC.UniOSCEventDispatcherCB.isBundle [get]

16.15.4.3 bool UniOSC.UniOSCEventDispatcherCB.isEnabled [get]

16.15.4.4 string UniOSC.UniOSCEventDispatcherCB.oscOutAddress [get], [set]

16.15.4.5 string UniOSC.UniOSCEventDispatcherCB.oscOutIPAddress [get], [set]

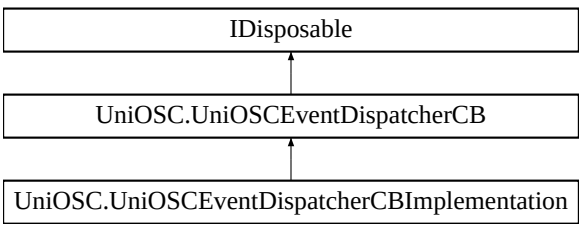
16.15.4.6 int UniOSC.UniOSCEventDispatcherCB.oscOutPort [get], [set]

16.15.4.7 bool UniOSC.UniOSCEventDispatcherCB.useExplicitConnection [get], [set]

16.16 UniOSC.UniOSCEventDispatcherCBImplementation Class Reference

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherCBImplementation:



Public Member Functions

- UniOSCEventDispatcherCBImplementation (string _oscOutAddress, string _oscOutIPAddress, int _oscPort)

  You have to override the constructors you want to use from the base class UniOSC.UniOSCEventDispatcherCode↩ Based class.
- UniOSCEventDispatcherCBImplementation (string _oscOutAddress, UniOSCConnection _explicit↩ Connection)
- override void Awake ()
- override void Enable ()

  Enable this instance.
- override void Disable ()

  Disable this instance.
- void SetDataAtIndex0 (bool val)

  Just a demo method to show how you can change the data of your OSC Message

Additional Inherited Members

16.16.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.

//Don't forget the base callings !!!!

16.16.2 Constructor & Destructor Documentation

16.16.2.1 UniOSC.UniOSCEventDispatcherCBImplementation.UniOSCEventDispatcherCBImplementation (string *_oscOutAddress,* string *_oscOutIPAddress,* int *_oscPort*)

You have to override the constructors you want to use from the base class UniOSC.UniOSCEventDispatcherCode↩ Based class.

16.16.2.2 UniOSC.UniOSCEventDispatcherCBImplementation.UniOSCEventDispatcherCBImplementation (string *_oscOutAddress,* UniOSCConnection *_explicitConnection*)

16.16.3 Member Function Documentation

16.16.3.1 override void UniOSC.UniOSCEventDispatcherCBImplementation.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.16.3.2 override void UniOSC.UniOSCEventDispatcherCBImplementation.Disable ( ) [virtual]

Disable this instance.

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.16.3.3 override void UniOSC.UniOSCEventDispatcherCBImplementation.Enable ( ) [virtual]

Enable this instance.

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.16.3.4 void UniOSC.UniOSCEventDispatcherCBImplementation.SetDataAtIndex0 (bool *val*)

Just a demo method to show how you can change the data of your OSC Message

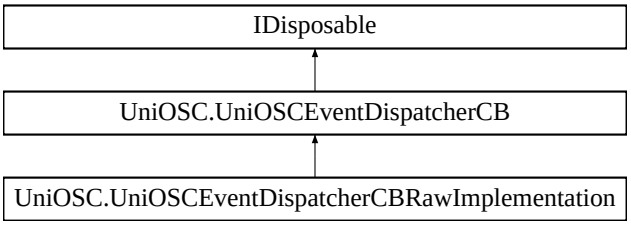
Parameters

val	If set to `true` value.

16.17 UniOSC.UniOSCEventDispatcherCBRawImplementation Class Reference

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherCBRawImplementation:



Public Member Functions

- UniOSCEventDispatcherCBRawImplementation (string _oscOutAddress, string _oscOutIPAddress, int _↩ oscPort)

  You have to override the constructors you want to use from the base class UniOSC.UniOSCEventDispatcherCode↩ Based class.
- UniOSCEventDispatcherCBRawImplementation (string _oscOutAddress, UniOSCConnection _explicit↩ Connection)
- override void Awake ()
- override void Enable ()

  Enable this instance.
- override void Disable ()

  Disable this instance.

Additional Inherited Members

16.17.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.

//Don't forget the base callings !!!!

16.17.2 Constructor & Destructor Documentation

16.17.2.1 UniOSC.UniOSCEventDispatcherCBRawImplementation.UniOSCEventDispatcherCBRawImplementation (string *_oscOutAddress*, string *_oscOutIPAddress*, int *_oscPort*)

You have to override the constructors you want to use from the base class UniOSC.UniOSCEventDispatcherCode↩ Based class.

16.17.2.2 UniOSC.UniOSCEventDispatcherCBRawImplementation.UniOSCEventDispatcherCBRawImplementation (string *_oscOutAddress,* UniOSCConnection *_explicitConnection*)

16.17.3 Member Function Documentation

16.17.3.1 override void UniOSC.UniOSCEventDispatcherCBRawImplementation.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.17.3.2 override void UniOSC.UniOSCEventDispatcherCBRawImplementation.Disable ( ) [virtual]

Disable this instance.

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.17.3.3 override void UniOSC.UniOSCEventDispatcherCBRawImplementation.Enable ( ) [virtual]

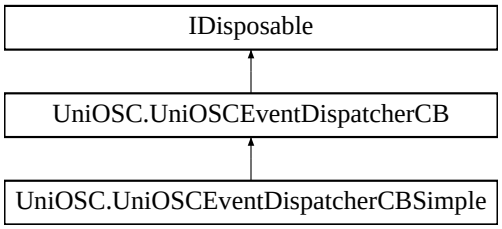
Enable this instance.

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.18 UniOSC.UniOSCEventDispatcherCBSimple Class Reference

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherCBSimple:



Public Member Functions

- UniOSCEventDispatcherCBSimple (string __oscOutAddress, string __oscOutIPAddress, int __oscPort)

  You have to override the constructors you want to use from the base class UniOSC.UniOSCEventDispatcherCode↩ Based class.
- UniOSCEventDispatcherCBSimple (string __oscOutAddress, UniOSCConnection __explicitConnection)
- override void Awake ()
- override void Enable ()

  Enable this instance.
- override void Disable ()

  Disable this instance.
- void SetDataAtIndex0 (object val)

  Just a demo method to show how you can change the data of your OSC Message

Additional Inherited Members

16.18.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcherCodeBased Dispatcher forces a OSCConnection to send a OSC Message.

//Don't forget the base callings !!!!

16.18.2 Constructor & Destructor Documentation

16.18.2.1 UniOSC.UniOSCEventDispatcherCBSimple.UniOSCEventDispatcherCBSimple (string *__oscOutAddress,* string *__oscOutIPAddress,* int *__oscPort*)

You have to override the constructors you want to use from the base class UniOSC.UniOSCEventDispatcherCode↩ Based class.

16.18.2.2 UniOSC.UniOSCEventDispatcherCBSimple.UniOSCEventDispatcherCBSimple (string *__oscOutAddress,* UniOSCConnection *__explicitConnection*)

16.18.3 Member Function Documentation

16.18.3.1 override void UniOSC.UniOSCEventDispatcherCBSimple.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.18.3.2 override void UniOSC.UniOSCEventDispatcherCBSimple.Disable ( ) [virtual]

Disable this instance.

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.18.3.3 override void UniOSC.UniOSCEventDispatcherCBSimple.Enable ( ) [virtual]

Enable this instance.

Reimplemented from UniOSC.UniOSCEventDispatcherCB.

16.18.3.4 void UniOSC.UniOSCEventDispatcherCBSimple.SetDataAtIndex0 (object *val*)

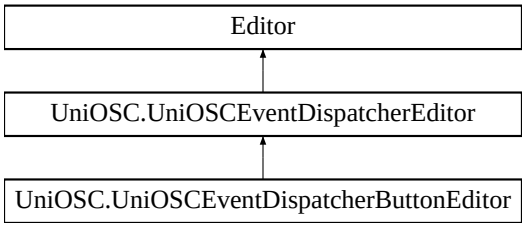
Just a demo method to show how you can change the data of your OSC Message

Parameters

val	If set to `true` value.

16.19 UniOSC.UniOSCEventDispatcherEditor Class Reference

Inheritance diagram for UniOSC.UniOSCEventDispatcherEditor:



Public Member Functions

- virtual void OnEnable ()
- override void OnInspectorGUI ()

Static Public Member Functions

- static void Show (string label, SerializedProperty list)

Protected Member Functions

- void DrawPort ()
- void DrawIPAddress ()
- void DrawConnectionInfo ()

Protected Attributes

- UniOSCEventDispatcher _target
- SerializedProperty _myOSCConnectionsProp
- SerializedProperty OSCConnectionsProp
- SerializedProperty OSCOutPortProp
- SerializedProperty OSCOutAddressProp
- SerializedProperty OSCOutIPAddressProp
- SerializedProperty OSCOutProp
- SerializedProperty UseExplicitConnectionProp
- SerializedProperty ExplicitConnectionProp
- SerializedProperty drawDefaultInspectorProp
- int _portIndex = 0
- string[] _options
- Texture2D _tex_logo

16.19.1 Member Function Documentation

16.19.1.1 void UniOSC.UniOSCEventDispatcherEditor.DrawConnectionInfo ( ) [protected]

16.19.1.2 void UniOSC.UniOSCEventDispatcherEditor.DrawIPAddress ( ) [protected]

16.19.1.3 void UniOSC.UniOSCEventDispatcherEditor.DrawPort ( ) [protected]

16.19.1.4 virtual void UniOSC.UniOSCEventDispatcherEditor.OnEnable ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventDispatcherButtonEditor.

16.19.1.5 override void UniOSC.UniOSCEventDispatcherEditor.OnInspectorGUI ( )

16.19.1.6 static void UniOSC.UniOSCEventDispatcherEditor.Show (string *label,* SerializedProperty *list*) [static]

16.19.2 Member Data Documentation

16.19.2.1 SerializedProperty UniOSC.UniOSCEventDispatcherEditor._myOSCConnectionsProp [protected]

16.19.2.2 string [] UniOSC.UniOSCEventDispatcherEditor._options [protected]

16.19.2.3 int UniOSC.UniOSCEventDispatcherEditor._portIndex = 0 [protected]

16.19.2.4 UniOSCEventDispatcher UniOSC.UniOSCEventDispatcherEditor._target [protected]

16.19.2.5 Texture2D UniOSC.UniOSCEventDispatcherEditor._tex_logo [protected]

16.19.2.6 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.drawDefaultInspectorProp [protected]

16.19.2.7 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.ExplicitConnectionProp [protected]

16.19.2.8 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.OSCConnectionsProp [protected]

16.19.2.9 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.OSCOutAddressProp [protected]

16.19.2.10 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.OSCOutIPAddressProp [protected]

16.19.2.11 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.OSCOutPortProp [protected]

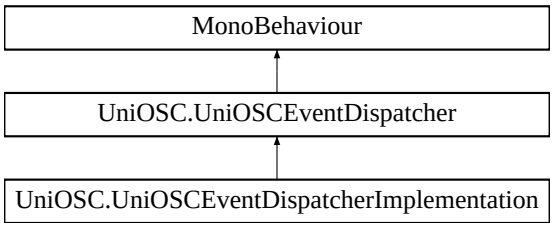
16.19.2.12 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.OSCOutProp [protected]

16.19.2.13 SerializedProperty UniOSC.UniOSCEventDispatcherEditor.UseExplicitConnectionProp [protected]

16.20 UniOSC.UniOSCEventDispatcherImplementation Class Reference

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherImplementation:



Public Member Functions

- override void Awake ()
- override void OnEnable ()
- override void OnDisable ()
- void MySendOSCMessageTriggerMethod ()

  Just a dummy method that shows how you trigger the OSC sending and how you could change the data of the OSC Message

Public Attributes

- int dynamicIntValue = 1000
- float dynamicFloatValue = 1000f
- string dynamicStringValue = "Test"

Additional Inherited Members

16.20.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.

//Don't forget the base callings !!!!

16.20.2 Member Function Documentation

16.20.2.1 override void UniOSC.UniOSCEventDispatcherImplementation.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.20.2.2 void UniOSC.UniOSCEventDispatcherImplementation.MySendOSCMessageTriggerMethod ( )

Just a dummy method that shows how you trigger the OSC sending and how you could change the data of the OSC Message

16.20.2.3 override void UniOSC.UniOSCEventDispatcherImplementation.OnDisable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.20.2.4 override void UniOSC.UniOSCEventDispatcherImplementation.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.20.3 Member Data Documentation

16.20.3.1 float UniOSC.UniOSCEventDispatcherImplementation.dynamicFloatValue = 1000f

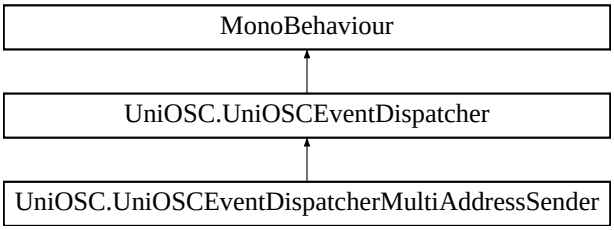
16.20.3.2 int UniOSC.UniOSCEventDispatcherImplementation.dynamicIntValue = 1000

16.20.3.3 string UniOSC.UniOSCEventDispatcherImplementation.dynamicStringValue = "Test"

16.21 UniOSC.UniOSCEventDispatcherMultiAddressSender Class Reference

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherMultiAddressSender:



Public Member Functions

- override void Awake ()
- override void OnEnable ()
- override void OnDisable ()
- void MySendOSCMessageTriggerMethod ()

  Just a dummy method that shows how you trigger the OSC sending and how you could change the data of the OSC Message
- void OnGUI ()

Public Attributes

- bool bundleMode
- float data = 1000f
- string[] addressArray

Additional Inherited Members

16.21.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.

//Don't forget the base callings !!!!

16.21.2 Member Function Documentation

16.21.2.1 override void UniOSC.UniOSCEventDispatcherMultiAddressSender.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.21.2.2 void UniOSC.UniOSCEventDispatcherMultiAddressSender.MySendOSCMessageTriggerMethod ( )

Just a dummy method that shows how you trigger the OSC sending and how you could change the data of the OSC Message

16.21.2.3 override void UniOSC.UniOSCEventDispatcherMultiAddressSender.OnDisable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.21.2.4 override void UniOSC.UniOSCEventDispatcherMultiAddressSender.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.21.2.5 void UniOSC.UniOSCEventDispatcherMultiAddressSender.OnGUI ( )

16.21.3 Member Data Documentation

16.21.3.1 string [] UniOSC.UniOSCEventDispatcherMultiAddressSender.addressArray

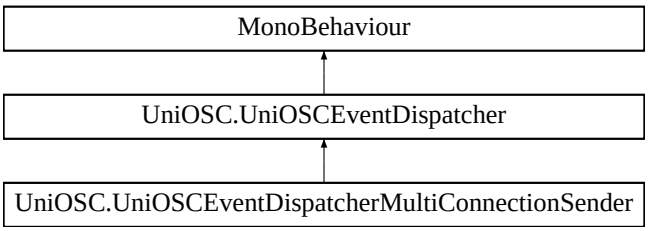
16.21.3.2 bool UniOSC.UniOSCEventDispatcherMultiAddressSender.bundleMode

16.21.3.3 float UniOSC.UniOSCEventDispatcherMultiAddressSender.data = 1000f

16.22 UniOSC.UniOSCEventDispatcherMultiConnectionSender Class Reference

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherMultiConnectionSender:



Public Member Functions

- override void Awake ()
- override void OnEnable ()
- override void OnDisable ()
- void MySendOSCMessageTrigerMethod ()

  Just a dummy method that shows how you trigger the OSC sending and how you could change the data of the OSC Message
- void OnGUI ()

Public Attributes

- float data = 1000f
- UniOSCConnection[] connectionArray

Additional Inherited Members

16.22.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class UniOSCEventDispatcher Dispatcher forces a OSCConnection to send a OSC Message.

//Don't forget the base callings !!!!

16.22.2 Member Function Documentation

16.22.2.1 override void UniOSC.UniOSCEventDispatcherMultiConnectionSender.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.22.2.2 void UniOSC.UniOSCEventDispatcherMultiConnectionSender.MySendOSCMessageTrigerMethod ( )

Just a dummy method that shows how you trigger the OSC sending and how you could change the data of the OSC Message

16.22.2.3 override void UniOSC.UniOSCEventDispatcherMultiConnectionSender.OnDisable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.22.2.4 override void UniOSC.UniOSCEventDispatcherMultiConnectionSender.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.22.2.5 void UniOSC.UniOSCEventDispatcherMultiConnectionSender.OnGUI ( )

16.22.3 Member Data Documentation

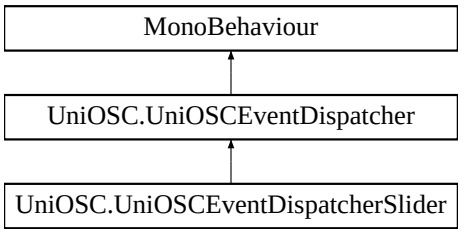
16.22.3.1 UniOSCConnection [] UniOSC.UniOSCEventDispatcherMultiConnectionSender.connectionArray

16.22.3.2 float UniOSC.UniOSCEventDispatcherMultiConnectionSender.data = 1000f

16.23 UniOSC.UniOSCEventDispatcherSlider Class Reference

Dispatcher toggle that forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherSlider:



Public Types

- enum SliderMode { SliderMode.Horizontal, SliderMode.Vertical }

Public Member Functions

- override void Awake ()
- override void OnEnable ()
- override void OnDisable ()
- override void SendOSCMessage ()

  Sends the OSC message with the sliderValue.

Public Attributes

- SliderMode sliderMode
- float minOSCDataValue = 0
- float maxOSCDataValue = 1
- bool showGUI
- float xPos
- float yPos
- float sliderSize = 100f

Additional Inherited Members

16.23.1 Detailed Description

Dispatcher toggle that forces a OSCConnection to send a OSC Message.

Two separate states: On and Off

16.23.2 Member Enumeration Documentation

16.23.2.1 enum UniOSC.UniOSCEventDispatcherSlider.SliderMode

Enumerator

Horizontal

Vertical

16.23.3 Member Function Documentation

16.23.3.1 override void UniOSC.UniOSCEventDispatcherSlider.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.23.3.2 override void UniOSC.UniOSCEventDispatcherSlider.OnDisable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.23.3.3 override void UniOSC.UniOSCEventDispatcherSlider.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.23.3.4 override void UniOSC.UniOSCEventDispatcherSlider.SendOSCMessage ( ) [virtual]

Sends the OSC message with the sliderValue.

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.23.4 Member Data Documentation

16.23.4.1 float UniOSC.UniOSCEventDispatcherSlider.maxOSCDataValue = 1

16.23.4.2 float UniOSC.UniOSCEventDispatcherSlider.minOSCDataValue = 0

16.23.4.3 bool UniOSC.UniOSCEventDispatcherSlider.showGUI

16.23.4.4 SliderMode UniOSC.UniOSCEventDispatcherSlider.sliderMode

16.23.4.5 float UniOSC.UniOSCEventDispatcherSlider.sliderSize = 100f

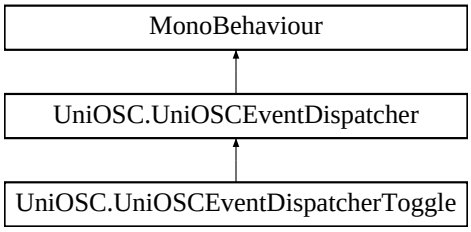
16.23.4.6 float UniOSC.UniOSCEventDispatcherSlider.xPos

16.23.4.7 float UniOSC.UniOSCEventDispatcherSlider.yPos

16.24 UniOSC.UniOSCEventDispatcherToggle Class Reference

Dispatcher toggle that forces a OSCConnection to send a OSC Message.

Inheritance diagram for UniOSC.UniOSCEventDispatcherToggle:



Public Member Functions

- override void Awake ()
- override void OnEnable ()
- override void OnDisable ()
- void SendOSCMessageOn ()

  Sends the OSC message with the downOSCDataValue.
- void SendOSCMessageOff ()

  Sends the OSC message with the upOSCDataValue.

Public Attributes

- float onOSCDataValue =1
- float offOSCDataValue =0
- bool showGUI
- float xPos
- float yPos

Additional Inherited Members

16.24.1 Detailed Description

Dispatcher toggle that forces a OSCConnection to send a OSC Message.

Two separate states: On and Off

16.24.2 Member Function Documentation

16.24.2.1 override void UniOSC.UniOSCEventDispatcherToggle.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.24.2.2 override void UniOSC.UniOSCEventDispatcherToggle.OnDisable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.24.2.3 override void UniOSC.UniOSCEventDispatcherToggle.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.24.2.4 void UniOSC.UniOSCEventDispatcherToggle.SendOSCMessageOff ( )

Sends the OSC message with the upOSCDataValue.

16.24.2.5 void UniOSC.UniOSCEventDispatcherToggle.SendOSCMessageOn ( )

Sends the OSC message with the downOSCDataValue.

16.24.3 Member Data Documentation

16.24.3.1 float UniOSC.UniOSCEventDispatcherToggle.offOSCDataValue =0

16.24.3.2 float UniOSC.UniOSCEventDispatcherToggle.onOSCDataValue =1

16.24.3.3 bool UniOSC.UniOSCEventDispatcherToggle.showGUI

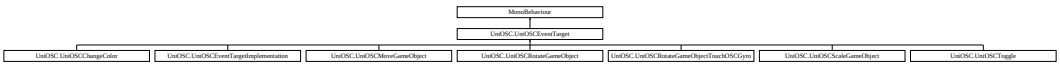
16.24.3.4 float UniOSC.UniOSCEventDispatcherToggle.xPos

16.24.3.5 float UniOSC.UniOSCEventDispatcherToggle.yPos

16.25 UniOSC.UniOSCEventTarget Class Reference

UniOSC event target.

Inheritance diagram for UniOSC.UniOSCEventTarget:



Public Member Functions

- virtual void Start ()
- virtual void Update ()
- virtual void OnEnable ()

  Enable this component and reinitialize.
- void ForceSetupChange ()

  This method is mainly for the EventTargetEditor to force an update of the internal message setup

- virtual void OnDestroy ()
- virtual void OnDisable ()

  When the component is disabled we disconnect from all OSCConnections and clear some internal data.
- abstract void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- Dictionary< UniOSCConnection, List< UniOSCMappingItem > > ConnectToDict = new Dictionary<UniOSCConnection,List<UniOSCMappingItem>>()

Protected Member Functions

- void _OnConnectionInStatusChanged (UniOSCConnection con)
- void _ConnectToDispatchers ()
- void _DisconnectFromDispatchers ()

Protected Attributes

- string _oscAddress = "/"
- List< string > _oscAddresses = new List<string>()
- bool _receiveAllAddresses
- bool _useExplicitConnection
- UniOSCConnection _explicitConnection
- int _oscPort
- bool _receiveAllPorts
- bool _redrawFlag
- List< UnityEngine.Object > _foldoutList = new List<UnityEngine.Object>()

Properties

- List< string > GetOSCAddresses `[get]`
- string oscAddress `[get, set]`
- bool receiveAllAddresses `[get, set]`
- bool useExplicitConnection `[get, set]`
- UniOSCConnection explicitConnection `[get, set]`
- int oscPort `[get, set]`
- bool receiveAllPorts `[get, set]`

Events

- EventHandler< UniOSCEventArgs > OSCMessageReceived

  Occurs when the OnOSCMessageReceived method is called.

16.25.1 Detailed Description

UniOSC event target.

This is the abstract class you should subclass from when you want to receive OSC data

16.25.2 Member Function Documentation

16.25.2.1 void UniOSC.UniOSCEventTarget._ConnectToDispatchers ( ) [protected]

16.25.2.2 void UniOSC.UniOSCEventTarget._DisconnectFromDispatchers ( ) [protected]

16.25.2.3 void UniOSC.UniOSCEventTarget._OnConnectionInStatusChanged (UniOSCConnection *con*) [protected]

16.25.2.4 void UniOSC.UniOSCEventTarget.ForceSetupChange ( )

This method is mainly for the EventTargetEditor to force an update of the internal message setup

16.25.2.5 virtual void UniOSC.UniOSCEventTarget.OnDestroy ( ) [virtual]

16.25.2.6 virtual void UniOSC.UniOSCEventTarget.OnDisable ( ) [virtual]

When the component is disabled we disconnect from all OSCConnections and clear some internal data.

Reimplemented in UniOSC.UniOSCEventTargetImplementation.

16.25.2.7 virtual void UniOSC.UniOSCEventTarget.OnEnable ( ) [virtual]

Enable this component and reinitialize.

If a property of the component is changed via the inspector we force a OnEnable to update the status of the component. In general the component disconnects from all OSCConnections and try to find a new OSCConnection to connect to with a matching port. If you change properties via code you should call this explicit.

Reimplemented in UniOSC.UniOSCToggle, UniOSC.UniOSCChangeColor, UniOSC.UniOSCRotateGameObject, UniOSC.UniOSCEventTargetImplementation, UniOSC.UniOSCScaleGameObject, UniOSC.UniOSCRotateGameObjectTouchOSCGyro, and UniOSC.UniOSCMoveGameObject.

16.25.2.8 abstract void UniOSC.UniOSCEventTarget.OnOSCMessageReceived (UniOSCEventArgs *args*) [pure virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implemented in UniOSC.UniOSCEventTargetImplementation, UniOSC.UniOSCToggle, UniOSC.UniOSCRotateGameObject, UniOSC.UniOSCChangeColor, UniOSC.UniOSCRotateGameObjectTouchOSCGyro, UniOSC.UniOSCMoveGameObject, and UniOSC.UniOSCScaleGameObject.

16.25.2.9 virtual void UniOSC.UniOSCEventTarget.Start ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventTargetImplementation.

16.25.2.10 virtual void UniOSC.UniOSCEventTarget.Update ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventTargetImplementation.

16.25.3 Member Data Documentation

16.25.3.1 UniOSCConnection UniOSC.UniOSCEventTarget._explicitConnection [protected]

16.25.3.2 List<UnityEngine.Object> UniOSC.UniOSCEventTarget._foldoutList = new List<UnityEngine.Object>() [protected]

16.25.3.3 string UniOSC.UniOSCEventTarget._oscAddress = "/" [protected]

16.25.3.4 List<string> UniOSC.UniOSCEventTarget._oscAddresses = new List<string>() [protected]

16.25.3.5 int UniOSC.UniOSCEventTarget._oscPort [protected]

16.25.3.6 bool UniOSC.UniOSCEventTarget._receiveAllAddresses [protected]

16.25.3.7 bool UniOSC.UniOSCEventTarget._receiveAllPorts [protected]

16.25.3.8 bool UniOSC.UniOSCEventTarget._redrawFlag [protected]

16.25.3.9 bool UniOSC.UniOSCEventTarget._useExplicitConnection [protected]

16.25.3.10 Dictionary<UniOSCConnection,List<UniOSCMappingItem> > UniOSC.UniOSCEventTarget.ConnectToDict = new Dictionary<UniOSCConnection,List<UniOSCMappingItem>>()

16.25.4 Property Documentation

16.25.4.1 UniOSCConnection UniOSC.UniOSCEventTarget.explicitConnection [get], [set]

16.25.4.2 List<string> UniOSC.UniOSCEventTarget.GetOSCAddresses [get]

16.25.4.3 string UniOSC.UniOSCEventTarget.oscAddress [get], [set]

16.25.4.4 int UniOSC.UniOSCEventTarget.oscPort [get], [set]

16.25.4.5 bool UniOSC.UniOSCEventTarget.receiveAllAddresses [get], [set]

16.25.4.6 bool UniOSC.UniOSCEventTarget.receiveAllPorts [get], [set]

16.25.4.7 bool UniOSC.UniOSCEventTarget.useExplicitConnection [get], [set]

16.25.5 Event Documentation

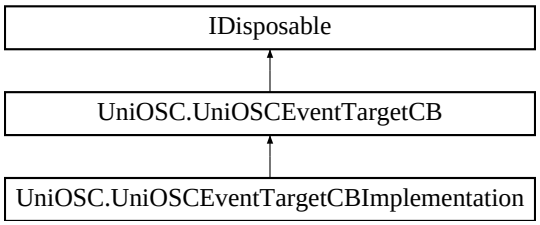
16.25.5.1 EventHandler<UniOSCEventArgs> UniOSC.UniOSCEventTarget.OSCMessageReceived

Occurs when the OnOSCMessageReceived method is called.

16.26 UniOSC.UniOSCEventTargetCB Class Reference

UniOSC event target for class based scripting.

Inheritance diagram for UniOSC.UniOSCEventTargetCB:



Public Member Functions

- string oscAddressAt (int index)

  If you call this method you will remove all addresses that you added before!
- bool AddAddress (string __oscAddress)
- UniOSCEventTargetCB (int __oscPort)
- UniOSCEventTargetCB (string _oscAddress)
- UniOSCEventTargetCB (string _oscAddress, int __oscPort)
- UniOSCEventTargetCB (UniOSCConnection con)
- UniOSCEventTargetCB (string _oscAddress, UniOSCConnection con)
- virtual void Awake ()
- virtual void Enable ()

  Enable this instance.
- virtual void Disable ()

  Disable this instance.
- void Dispose ()

  Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.
- void ForceSetupChange ()

  This method forces a reconnection.
- abstract void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- Dictionary< UniOSCConnection, List< UniOSCMappingItem > > ConnectToDict = new Dictionary<UniOSCConnection,List<UniOSCMappingItem>>()

Protected Member Functions

- void _ConnectToDispatchers ()

Protected Attributes

- List< string > _oscAddresses = new List<string>()
- string _oscAddress
- bool _receiveAllAddresses
- bool _useExplicitConnection
- UniOSCConnection _explicitConnection
- int _oscPort
- bool _receiveAllPorts

Properties

- bool isEnabled [get]
- string oscAddress [get, set]
- bool receiveAllAddresses [get, set]
- bool useExplicitConnection [get, set]
- UniOSCConnection explicitConnection [get, set]
- int oscPort [get, set]
- bool receiveAllPorts [get, set]

Events

- EventHandler< UniOSCEventArgs > OSCMessageReceived

16.26.1 Detailed Description

UniOSC event target for class based scripting.

This is the abstract class you should subclass from

16.26.2 Constructor & Destructor Documentation

16.26.2.1 UniOSC.UniOSCEventTargetCB.UniOSCEventTargetCB (int *__oscPort*)

16.26.2.2 UniOSC.UniOSCEventTargetCB.UniOSCEventTargetCB (string *_oscAddress*)

16.26.2.3 UniOSC.UniOSCEventTargetCB.UniOSCEventTargetCB (string *_oscAddress,* int *__oscPort*)

16.26.2.4 UniOSC.UniOSCEventTargetCB.UniOSCEventTargetCB (UniOSCConnection *con*)

16.26.2.5 UniOSC.UniOSCEventTargetCB.UniOSCEventTargetCB (string *_oscAddress,* UniOSCConnection *con*)

16.26.3 Member Function Documentation

16.26.3.1 void UniOSC.UniOSCEventTargetCB._ConnectToDispatchers ( ) [protected]

16.26.3.2 bool UniOSC.UniOSCEventTargetCB.AddAddress (string *__oscAddress*)

16.26.3.3 virtual void UniOSC.UniOSCEventTargetCB.Awake ( ) [virtual]

Reimplemented in UniOSC.UniOSCEventTargetCBImplementation.

16.26.3.4 virtual void UniOSC.UniOSCEventTargetCB.Disable ( ) [virtual]

Disable this instance.

Reimplemented in UniOSC.UniOSCEventTargetCBImplementation.

16.26.3.5 void UniOSC.UniOSCEventTargetCB.Dispose ( )

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Call Dispose when you are finished using the UniOSC.UniOSCEventTargetCB. The Dispose method leaves the UniOSC.UniOSCEventTargetCB in an unusable state. After calling Dispose, you must release all references to the

UniOSC.UniOSCEventTargetCB so the garbage collector can reclaim the memory that the UniOSC.UniOSCEvent↩ TargetCB was occupying.

16.26.3.6 virtual void UniOSC.UniOSCEventTargetCB.Enable ( ) [virtual]

Enable this instance.

Reimplemented in UniOSC.UniOSCEventTargetCBImplementation.

16.26.3.7 void UniOSC.UniOSCEventTargetCB.ForceSetupChange ( )

This method forces a reconnection.

Normally you don't need to call this method explicit but when you use this class for editor scripting you have to call this method when the playmode has changed.

16.26.3.8 abstract void UniOSC.UniOSCEventTargetCB.OnOSCMessageReceived (UniOSCEventArgs *args*) [pure virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implemented in UniOSC.UniOSCEventTargetCBImplementation.

16.26.3.9 string UniOSC.UniOSCEventTargetCB.oscAddressAt (int *index*)

If you call this method you will remove all addresses that you added before!

16.26.4 Member Data Documentation

16.26.4.1 UniOSCConnection UniOSC.UniOSCEventTargetCB._explicitConnection [protected]

16.26.4.2 string UniOSC.UniOSCEventTargetCB._oscAddress [protected]

16.26.4.3 List<string> UniOSC.UniOSCEventTargetCB._oscAddresses = new List<string>() [protected]

16.26.4.4 int UniOSC.UniOSCEventTargetCB._oscPort [protected]

16.26.4.5 bool UniOSC.UniOSCEventTargetCB._receiveAllAddresses [protected]

16.26.4.6 bool UniOSC.UniOSCEventTargetCB._receiveAllPorts [protected]

16.26.4.7 bool UniOSC.UniOSCEventTargetCB._useExplicitConnection [protected]

16.26.4.8 Dictionary<UniOSCConnection,List<UniOSCMappingItem> > UniOSC.UniOSCEventTargetCB.Connect↩ ToDict = new Dictionary<UniOSCConnection,List<UniOSCMappingItem>>()

16.26.5 Property Documentation

16.26.5.1 UniOSCConnection UniOSC.UniOSCEventTargetCB.explicitConnection [get], [set]

16.26.5.2 bool UniOSC.UniOSCEventTargetCB.isEnabled [get]

16.26.5.3 string UniOSC.UniOSCEventTargetCB.oscAddress [get], [set]

16.26.5.4 int UniOSC.UniOSCEventTargetCB.oscPort [get], [set]

16.26.5.5 bool UniOSC.UniOSCEventTargetCB.receiveAllAddresses [get], [set]

16.26.5.6 bool UniOSC.UniOSCEventTargetCB.receiveAllPorts [get], [set]

16.26.5.7 bool UniOSC.UniOSCEventTargetCB.useExplicitConnection [get], [set]

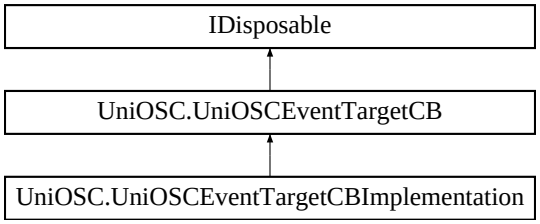
16.26.6 Event Documentation

16.26.6.1 EventHandler<UniOSCEventArgs> UniOSC.UniOSCEventTargetCB.OSCMessageReceived

16.27 UniOSC.UniOSCEventTargetCBImplementation Class Reference

This class is a blueprint for your own implementations of the abstract class OSCDispatcherTargetCB //Don't forget the base callings !!!! The OnOSCMessageReceived method is where you should parse the OSC data

Inheritance diagram for UniOSC.UniOSCEventTargetCBImplementation:



Public Member Functions

- UniOSCEventTargetCBImplementation (int oscPort)

  You have to override the constructors you want to use from the base class UniOSC.UniOSCEventTargetCB class.
- UniOSCEventTargetCBImplementation (string oscAddress)
- UniOSCEventTargetCBImplementation (UniOSCConnection con)
- UniOSCEventTargetCBImplementation (string oscAddress, int oscPort)
- UniOSCEventTargetCBImplementation (string oscAddress, UniOSCConnection con)
- override void Awake ()
- override void Enable ()

  Enable this instance.
- override void Disable ()

  Disable this instance.
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  Method is called from a OSCConnection when a OSC message arrives.

Additional Inherited Members

16.27.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class OSCDispatcherTargetCB //Don't forget the base callings !!!! The OnOSCMessageReceived method is where you should parse the OSC data

16.27.2 Constructor & Destructor Documentation

16.27.2.1 UniOSC.UniOSCEventTargetCBImplementation.UniOSCEventTargetCBImplementation (int *oscPort*)

You have to override the constructors you want to use from the base class UniOSC.UniOSCEventTargetCB class.

16.27.2.2 UniOSC.UniOSCEventTargetCBImplementation.UniOSCEventTargetCBImplementation (string *oscAddress*)

16.27.2.3 UniOSC.UniOSCEventTargetCBImplementation.UniOSCEventTargetCBImplementation (UniOSCConnection *con*)

16.27.2.4 UniOSC.UniOSCEventTargetCBImplementation.UniOSCEventTargetCBImplementation (string *oscAddress,* int *oscPort*)

16.27.2.5 UniOSC.UniOSCEventTargetCBImplementation.UniOSCEventTargetCBImplementation (string *oscAddress,* UniOSCConnection *con*)

16.27.3 Member Function Documentation

16.27.3.1 override void UniOSC.UniOSCEventTargetCBImplementation.Awake ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventTargetCB.

16.27.3.2 override void UniOSC.UniOSCEventTargetCBImplementation.Disable ( ) [virtual]

Disable this instance.

Reimplemented from UniOSC.UniOSCEventTargetCB.

16.27.3.3 override void UniOSC.UniOSCEventTargetCBImplementation.Enable ( ) [virtual]

Enable this instance.

Reimplemented from UniOSC.UniOSCEventTargetCB.

16.27.3.4 override void UniOSC.UniOSCEventTargetCBImplementation.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

Method is called from a OSCConnection when a OSC message arrives.

The argument is a UniOSCEventArgs object where all the related data is enclosed

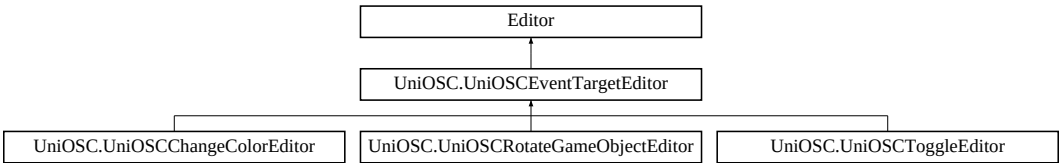
Parameters

args	OSCEventArgs

Implements UniOSC.UniOSCEventTargetCB.

16.28 UniOSC.UniOSCEventTargetEditor Class Reference

Inheritance diagram for UniOSC.UniOSCEventTargetEditor:



Public Member Functions

- virtual void OnEnable ()
- override void OnInspectorGUI ()

Protected Member Functions

- void DrawConnectionSetup ()
- void DrawPort ()
- void DrawConnectionInfo ()
- void ShowFoldoutConnectionStatus (string label, SerializedProperty list, IDictionary dict)

Protected Attributes

- UniOSCEventTarget _target
- SerializedProperty ReceiveAllAddressesProp
- SerializedProperty ReceiveAllPortsProp
- SerializedProperty OSCAddressProp
- SerializedProperty OSCPortProp
- SerializedProperty FoldoutListProp
- SerializedProperty AvailableINPortsProp
- SerializedProperty UseExplicitConnectionProp
- SerializedProperty ExplicitConnectionProp
- Texture2D _tex_logo
- int _portIndex = 0
- string[] _options

16.28.1 Member Function Documentation

16.28.1.1 void UniOSC.UniOSCEventTargetEditor.DrawConnectionInfo ( ) [protected]

16.28.1.2 void UniOSC.UniOSCEventTargetEditor.DrawConnectionSetup ( ) [protected]

16.28.1.3 void UniOSC.UniOSCEventTargetEditor.DrawPort ( ) [protected]

16.28.1.4 virtual void UniOSC.UniOSCEventTargetEditor.OnEnable ( ) [virtual]

Reimplemented in UniOSC.UniOSCToggleEditor.

16.28.1.5 override void UniOSC.UniOSCEventTargetEditor.OnInspectorGUI ( )

16.28.1.6 void UniOSC.UniOSCEventTargetEditor.ShowFoldoutConnectionStatus (string *label*, SerializedProperty *list*, IDictionary *dict*) [protected]

16.28.2 Member Data Documentation

16.28.2.1 string [] UniOSC.UniOSCEventTargetEditor._options [protected]

16.28.2.2 int UniOSC.UniOSCEventTargetEditor._portIndex = 0 [protected]

16.28.2.3 UniOSCEventTarget UniOSC.UniOSCEventTargetEditor._target [protected]

16.28.2.4 Texture2D UniOSC.UniOSCEventTargetEditor._tex_logo [protected]

16.28.2.5 SerializedProperty UniOSC.UniOSCEventTargetEditor.AvailableINPortsProp [protected]

16.28.2.6 SerializedProperty UniOSC.UniOSCEventTargetEditor.ExplicitConnectionProp [protected]

16.28.2.7 SerializedProperty UniOSC.UniOSCEventTargetEditor.FoldoutListProp [protected]

16.28.2.8 SerializedProperty UniOSC.UniOSCEventTargetEditor.OSCAddressProp [protected]

16.28.2.9 SerializedProperty UniOSC.UniOSCEventTargetEditor.OSCPortProp [protected]

16.28.2.10 SerializedProperty UniOSC.UniOSCEventTargetEditor.ReceiveAllAddressesProp [protected]

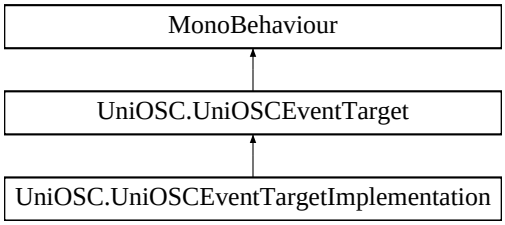
16.28.2.11 SerializedProperty UniOSC.UniOSCEventTargetEditor.ReceiveAllPortsProp [protected]

16.28.2.12 SerializedProperty UniOSC.UniOSCEventTargetEditor.UseExplicitConnectionProp [protected]

16.29 UniOSC.UniOSCEventTargetImplementation Class Reference

This class is a blueprint for your own implementations of the abstract class OSCDispatcherTarget //Don't forget the base callings !!!! The OnOSCMessageReceived method is where you should parse the OSC data

Inheritance diagram for UniOSC.UniOSCEventTargetImplementation:



Public Member Functions

- override void Start ()

  Start this instance.
- override void OnEnable ()

  Raises the enable event.
- override void OnDisable ()

  Raises the disable event.
- override void Update ()
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  Method is called from a OSCConnection when a OSC message arrives.

Additional Inherited Members

16.29.1 Detailed Description

This class is a blueprint for your own implementations of the abstract class OSCDispatcherTarget //Don't forget the base callings !!!! The OnOSCMessageReceived method is where you should parse the OSC data

16.29.2 Member Function Documentation

16.29.2.1 override void UniOSC.UniOSCEventTargetImplementation.OnDisable ( ) [virtual]

Raises the disable event.

Reimplemented from UniOSC.UniOSCEventTarget.

16.29.2.2 override void UniOSC.UniOSCEventTargetImplementation.OnEnable ( ) [virtual]

Raises the enable event.

If you want to listen to several OSC messages you have to set the OSCAddresses property before you call base.↩ OnEnable() OSCAddresses.Clear(); OSCAddresses.Add(...);

Reimplemented from UniOSC.UniOSCEventTarget.

16.29.2.3 override void UniOSC.UniOSCEventTargetImplementation.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

Method is called from a OSCConnection when a OSC message arrives.

The argument is a UniOSCEventArgs object where all the related data is enclosed

Parameters

args	OSCEventArgs

Implements UniOSC.UniOSCEventTarget.

16.29.2.4 override void UniOSC.UniOSCEventTargetImplementation.Start ( ) [virtual]

Start this instance.

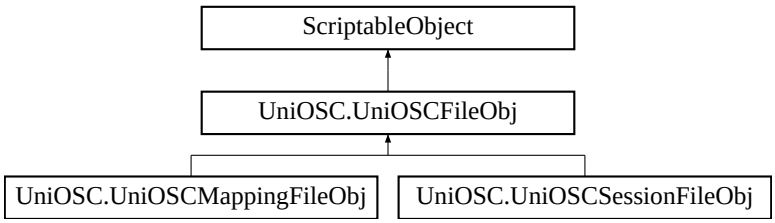
Reimplemented from UniOSC.UniOSCEventTarget.

16.29.2.5 override void UniOSC.UniOSCEventTargetImplementation.Update ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventTarget.

16.30 UniOSC.UniOSCFileObj Class Reference

Inheritance diagram for UniOSC.UniOSCFileObj:



Public Attributes

- string my_guid
- bool IsLearning
- Vector2 scrollpos = new Vector2()
- Vector2 scrollposInspector = new Vector2()

Events

- EventHandler< UniOSCEventArgs > OSCMessageSend

16.30.1 Detailed Description

16.30.2 Member Data Documentation

16.30.2.1 bool UniOSC.UniOSCFileObj.IsLearning

16.30.2.2 string UniOSC.UniOSCFileObj.my_guid

16.30.2.3 Vector2 UniOSC.UniOSCFileObj.scrollpos = new Vector2()

16.30.2.4 Vector2 UniOSC.UniOSCFileObj.scrollposInspector = new Vector2()

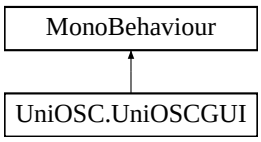
16.30.3 Event Documentation

16.30.3.1 EventHandler<UniOSCEventArgs> UniOSC.UniOSCFileObj.OSCMessageSend

16.31 UniOSC.UniOSCGUI Class Reference

GUI class that mimics the UniOSC editor interface for runtime use You can start/stop the OSCConnections and trace OSC data messages

Inheritance diagram for UniOSC.UniOSCGUI:



Public Attributes

- bool ShowInEditMode
- bool traceMessages

16.31.1 Detailed Description

GUI class that mimics the UniOSC editor interface for runtime use You can start/stop the OSCConnections and trace OSC data messages

16.31.2 Member Data Documentation

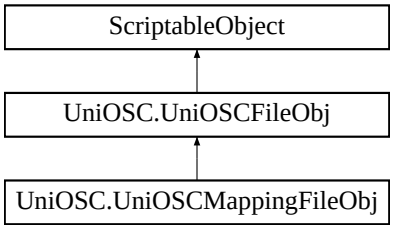
16.31.2.1 bool UniOSC.UniOSCGUI.ShowInEditMode

16.31.2.2 bool UniOSC.UniOSCGUI.traceMessages

16.32 UniOSC.UniOSCMappingFileObj Class Reference

Mapping file class .

Inheritance diagram for UniOSC.UniOSCMappingFileObj:



Public Member Functions

- void OnEnable ()
- void AddOSCMappingItem ()

  Adds a new OSC Mapping item.
- void RemoveOSCMappingItem (UniOSCMappingItem obj)

  Removes the OSC Mapping item from the list and destroys the item instance.
- void OnOSCMessageReceived (object sender, UniOSCEventArgs args)

  Checks if we are in learning mode an writes the OSC message address into the address property of a mapping item that is in learn mode(when user hold down the 'learn' button in the editor.

Public Attributes

- List< UniOSCMappingItem > oscMappingItemList

Additional Inherited Members

16.32.1 Detailed Description

Mapping file class .

Every mapping file get stored as a .asset file. You can copy & paste a mapping file to another Unity project but you have to aware that sometimes Unity changes the serialization format. If you have any trouble go to 'Edit/Project Settings/Editor' and change the seriaization mode to 'Force Text' and then switch back to 'Force Binary'

16.32.2 Member Function Documentation

16.32.2.1 void UniOSC.UniOSCMappingFileObj.AddOSCMappingItem ( )

Adds a new OSC Mapping item.

16.32.2.2 void UniOSC.UniOSCMappingFileObj.OnEnable ( )

16.32.2.3 void UniOSC.UniOSCMappingFileObj.OnOSCMessageReceived (object *sender*, **UniOSCEventArgs** *args*)

Checks if we are in learning mode an writes the OSC message address into the address property of a mapping item that is in learn mode(when user hold down the 'learn' button in the editor.

Parameters

sender	Sender.
args	UniOSCEventArgs that contains the OSC message

16.32.2.4 void UniOSC.UniOSCMappingFileObj.RemoveOSCMappingItem (**UniOSCMappingItem** *obj*)

Removes the OSC Mapping item from the list and destroys the item instance.

UniOSC.UniOSCMappingItem.OnOSCDataDispatcherDelete

Parameters

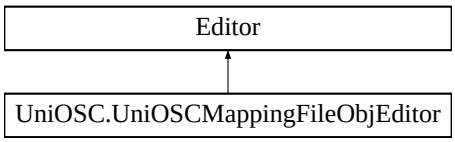
obj	Object to remove.

16.32.3 Member Data Documentation

16.32.3.1 List<**UniOSCMappingItem**> UniOSC.UniOSCMappingFileObj.oscMappingItemList

16.33 UniOSC.UniOSCMappingFileObjEditor Class Reference

Inheritance diagram for UniOSC.UniOSCMappingFileObjEditor:



Public Member Functions

- override void OnInspectorGUI ()

Static Public Member Functions

- static void Init ()
- static void OnGUI_OSCMappingData_Editor (UniOSCMappingFileObj obj, float screenWidth, float screen↩ Height)
- static void OnGUI_OSCMappingData_Inspector (UniOSCMappingFileObj obj, float screenWidth, float screenHeight)

Static Public Attributes

- static GUIStyle style

16.33.1 Member Function Documentation

16.33.1.1 static void UniOSC.UniOSCMappingFileObjEditor.Init ( ) [static]

16.33.1.2 static void UniOSC.UniOSCMappingFileObjEditor.OnGUI_OSCMappingData_Editor (UniOSCMappingFileObj *obj,* float *screenWidth,* float *screenHeight*) [static]

16.33.1.3 static void UniOSC.UniOSCMappingFileObjEditor.OnGUI_OSCMappingData_Inspector (UniOSCMappingFileObj *obj,* float *screenWidth,* float *screenHeight*) [static]

16.33.1.4 override void UniOSC.UniOSCMappingFileObjEditor.OnInspectorGUI ( )

16.33.2 Member Data Documentation

16.33.2.1 GUIStyle UniOSC.UniOSCMappingFileObjEditor.style [static]

16.34 UniOSC.UniOSCMappingItem Class Reference

Uni OSC mapping item.

Public Member Functions

- UniOSCMappingItem ()
- UniOSCMappingItem (UniOSCMappingFileObj _hostObj)

  Initializes a new instance of the UniOSC.UniOSCMappingItem class.
- void OnOSCMappingItemDelete ()

  Removes this item from the UniOSCMappingFileObj host object.Afterwards it gets destroyed.
- void MapData (UniOSCEventArgs args)

  Maps the incoming OSC data.

Public Attributes

- UniOSCMappingFileObj hostObj
- string address = ""
- float min =0f
- float max =1f
- float mappingMIN =0f
- float mappingMAX =1f
- bool isLearning
- const int MAXWIDTH = 250
- const int MAXHEIGTH = 150
- bool collapsed = true

16.34.1 Detailed Description

Uni OSC mapping item.

<author> Stefan Schlupek </author>

16.34.2 Constructor & Destructor Documentation

16.34.2.1 UniOSC.UniOSCMappingItem.UniOSCMappingItem ( )

16.34.2.2 UniOSC.UniOSCMappingItem.UniOSCMappingItem (**UniOSCMappingFileObj** *_hostObj*)

Initializes a new instance of the UniOSC.UniOSCMappingItem class.

Parameters

_hostObj	The UniOSCMappingFileObj object that host the item

16.34.3 Member Function Documentation

16.34.3.1 void UniOSC.UniOSCMappingItem.MapData (**UniOSCEventArgs** *args*)

Maps the incoming OSC data.

Parameters

args	Arguments.

16.34.3.2 void UniOSC.UniOSCMappingItem.OnOSCMappingItemDelete ( )

Removes this item from the UniOSCMappingFileObj host object.Afterwards it gets destroyed.

16.34.4 Member Data Documentation

16.34.4.1 string UniOSC.UniOSCMappingItem.address = ""

16.34.4.2 bool UniOSC.UniOSCMappingItem.collapsed = true

16.34.4.3 **UniOSCMappingFileObj** UniOSC.UniOSCMappingItem.hostObj

16.34.4.4 bool UniOSC.UniOSCMappingItem.isLearning

16.34.4.5 float UniOSC.UniOSCMappingItem.mappingMAX =1f

16.34.4.6 float UniOSC.UniOSCMappingItem.mappingMIN =0f

16.34.4.7 float UniOSC.UniOSCMappingItem.max =1f

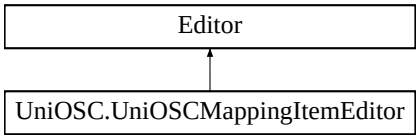
16.34.4.8 const int UniOSC.UniOSCMappingItem.MAXHEIGTH = 150

16.34.4.9 const int UniOSC.UniOSCMappingItem.MAXWIDTH = 250

16.34.4.10 float UniOSC.UniOSCMappingItem.min =0f

16.35 UniOSC.UniOSCMappingItemEditor Class Reference

Inheritance diagram for UniOSC.UniOSCMappingItemEditor:



Public Member Functions

- void OnEnable ()

Static Public Member Functions

- static void OnGUI_Editor (UniOSCMappingItem obj)
- static void OnGUI_Inspector (UniOSCMappingItem obj)

16.35.1 Member Function Documentation

16.35.1.1 void UniOSC.UniOSCMappingItemEditor.OnEnable ( )

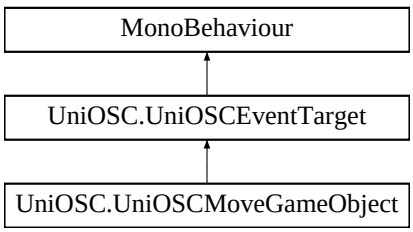
16.35.1.2 static void UniOSC.UniOSCMappingItemEditor.OnGUI_Editor (UniOSCMappingItem *obj*) [static]

16.35.1.3 static void UniOSC.UniOSCMappingItemEditor.OnGUI_Inspector (UniOSCMappingItem *obj*) [static]

16.36 UniOSC.UniOSCMoveGameObject Class Reference

Moves a GameObject in normalized coordinates (ScreenToWorldPoint)

Inheritance diagram for UniOSC.UniOSCMoveGameObject:



Public Types

- enum Mode { Mode.Screen, Mode.Relative }

Public Member Functions

- override void OnEnable ()

  Enable this component and reinitialize.
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- Transform transformToMove
- float nearClipPlaneOffset = 1
- Mode movementMode

Additional Inherited Members

16.36.1 Detailed Description

Moves a GameObject in normalized coordinates (ScreenToWorldPoint)

16.36.2 Member Enumeration Documentation

16.36.2.1 enum UniOSC.UniOSCMoveGameObject.Mode

Enumerator

Screen

Relative

16.36.3 Member Function Documentation

16.36.3.1 override void UniOSC.UniOSCMoveGameObject.OnEnable ( ) [virtual]

Enable this component and reinitialize.

If a property of the component is changed via the inspector we force a OnEnable to update the status of the component. In general the component disconnects from all OSCConnections and try to find a new OSCConnection to connect to with a matching port. If you change properties via code you should call this explicit.

Reimplemented from UniOSC.UniOSCEventTarget.

16.36.3.2 override void UniOSC.UniOSCMoveGameObject.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implements UniOSC.UniOSCEventTarget.

16.36.4 Member Data Documentation

16.36.4.1 Mode UniOSC.UniOSCMoveGameObject.movementMode

16.36.4.2 float UniOSC.UniOSCMoveGameObject.nearClipPlaneOffset = 1

16.36.4.3 Transform UniOSC.UniOSCMoveGameObject.transformToMove

16.37 UniOSC.UniOSCReceiver Class Reference

Uni OSC receiver.

Public Member Functions

- UniOSCReceiver ()
- UniOSCReceiver (int port, string MulticastAddress)
- UniOSCReceiver (int port, TransmissionType ttype, IPAddress MulticastAddress)
- bool Connect ()

  Connect this instance.
- void Disconnect ()

  Disconnect this instance.

Properties

- int Port [get]
- int FrameNumber [get]

Events

- EventHandler< UniOSCEventArgs > OSCMessageReceived
- EventHandler< ExceptionEventArgs > OSCErrorOccured

16.37.1 Detailed Description

Uni OSC receiver.

16.37.2 Constructor & Destructor Documentation

16.37.2.1 UniOSC.UniOSCReceiver.UniOSCReceiver ( )

16.37.2.2 UniOSC.UniOSCReceiver.UniOSCReceiver (int *port*, string *MulticastAddress*)

16.37.2.3 UniOSC.UniOSCReceiver.UniOSCReceiver (int *port*, TransmissionType *ttype*, IPAddress *MulticastAddress*)

16.37.3 Member Function Documentation

16.37.3.1 bool UniOSC.UniOSCReceiver.Connect ( )

Connect this instance.

16.37.3.2 void UniOSC.UniOSCReceiver.Disconnect ( )

Disconnect this instance.

16.37.4 Property Documentation

16.37.4.1 int UniOSC.UniOSCReceiver.FrameNumber [get]

16.37.4.2 int UniOSC.UniOSCReceiver.Port [get]

16.37.5 Event Documentation

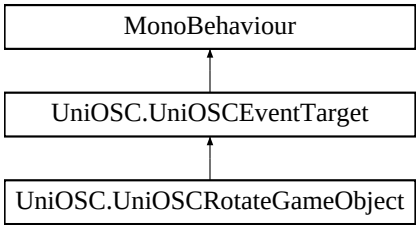
16.37.5.1 EventHandler<ExceptionEventArgs> UniOSC.UniOSCReceiver.OSCErrorOccured

16.37.5.2 EventHandler<UniOSCEventArgs> UniOSC.UniOSCReceiver.OSCMessageReceived

16.38 UniOSC.UniOSCRotateGameObject Class Reference

Rotates (localRotation) the hosting game object.

Inheritance diagram for UniOSC.UniOSCRotateGameObject:



Public Member Functions

- override void OnEnable ()

  Enable this component and reinitialize.
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- Transform transformToRotate
- string X_Address
- string Y_Address
- string Z_Address
- float x_RotationFactor
- float y_RotationFactor
- float z_RotationFactor

Additional Inherited Members

16.38.1 Detailed Description

Rotates (localRotation) the hosting game object.

For every axis you have a separate OSC address to specify

16.38.2 Member Function Documentation

16.38.2.1 override void UniOSC.UniOSCRotateGameObject.OnEnable ( ) [virtual]

Enable this component and reinitialize.

If a property of the component is changed via the inspector we force a OnEnable to update the status of the component. In general the component disconnects from all OSCConnections and try to find a new OSCConnection to connect to with a matching port. If you change properties via code you should call this explicit.

Reimplemented from UniOSC.UniOSCEventTarget.

16.38.2.2 override void UniOSC.UniOSCRotateGameObject.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implements UniOSC.UniOSCEventTarget.

16.38.3 Member Data Documentation

16.38.3.1 Transform UniOSC.UniOSCRotateGameObject.transformToRotate

16.38.3.2 string UniOSC.UniOSCRotateGameObject.X_Address

16.38.3.3 float UniOSC.UniOSCRotateGameObject.x_RotationFactor

16.38.3.4 string UniOSC.UniOSCRotateGameObject.Y_Address

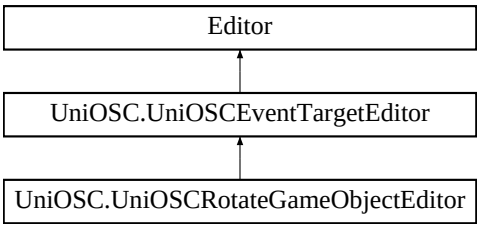
16.38.3.5 float UniOSC.UniOSCRotateGameObject.y_RotationFactor

16.38.3.6 string UniOSC.UniOSCRotateGameObject.Z_Address

16.38.3.7 float UniOSC.UniOSCRotateGameObject.z_RotationFactor

16.39 UniOSC.UniOSCRotateGameObjectEditor Class Reference

Inheritance diagram for UniOSC.UniOSCRotateGameObjectEditor:



Public Member Functions

- override void OnInspectorGUI ()

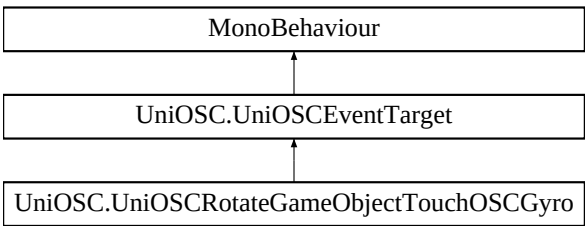
Additional Inherited Members

16.39.1 Member Function Documentation

16.39.1.1 override void UniOSC.UniOSCRotateGameObjectEditor.OnInspectorGUI ( )

16.40 UniOSC.UniOSCRotateGameObjectTouchOSCGyro Class Reference

Inheritance diagram for UniOSC.UniOSCRotateGameObjectTouchOSCGyro:



Public Member Functions

- override void OnEnable ()

  Enable this component and reinitialize.
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- Transform transformToRotate
- float x_RotationFactor =90
- float y_RotationFactor =90
- float z_RotationFactor =90
- float damping =1

Additional Inherited Members

16.40.1 Member Function Documentation

16.40.1.1 override void UniOSC.UniOSCRotateGameObjectTouchOSCGyro.OnEnable ( ) [virtual]

Enable this component and reinitialize.

If a property of the component is changed via the inspector we force a OnEnable to update the status of the component. In general the component disconnects from all OSCConnections and try to find a new OSCConnection to connect to with a matching port. If you change properties via code you should call this explicit.

Reimplemented from UniOSC.UniOSCEventTarget.

16.40.1.2 override void UniOSC.UniOSCRotateGameObjectTouchOSCGyro.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implements UniOSC.UniOSCEventTarget.

16.40.2 Member Data Documentation

16.40.2.1 float UniOSC.UniOSCRotateGameObjectTouchOSCGyro.damping =1

16.40.2.2 Transform UniOSC.UniOSCRotateGameObjectTouchOSCGyro.transformToRotate

16.40.2.3 float UniOSC.UniOSCRotateGameObjectTouchOSCGyro.x_RotationFactor =90

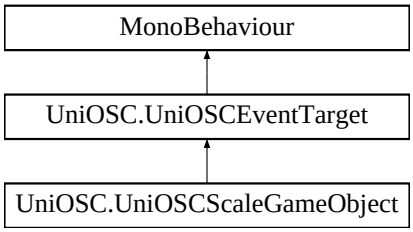
16.40.2.4 float UniOSC.UniOSCRotateGameObjectTouchOSCGyro.y_RotationFactor =90

16.40.2.5 float UniOSC.UniOSCRotateGameObjectTouchOSCGyro.z_RotationFactor =90

16.41 UniOSC.UniOSCScaleGameObject Class Reference

Uni OSC scale game object.

Inheritance diagram for UniOSC.UniOSCScaleGameObject:



Public Member Functions

- override void OnEnable ()

  Enable this component and reinitialize.
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- Transform transformToScale
- float scaleFactor = 1

Additional Inherited Members

16.41.1 Detailed Description

Uni OSC scale game object.

16.41.2 Member Function Documentation

16.41.2.1 override void UniOSC.UniOSCScaleGameObject.OnEnable ( ) [virtual]

Enable this component and reinitialize.

If a property of the component is changed via the inspector we force a OnEnable to update the status of the component. In general the component disconnects from all OSCConnections and try to find a new OSCConnection to connect to with a matching port. If you change properties via code you should call this explicit.

Reimplemented from UniOSC.UniOSCEventTarget.

16.41.2.2 override void UniOSC.UniOSCScaleGameObject.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implements UniOSC.UniOSCEventTarget.

16.41.3 Member Data Documentation

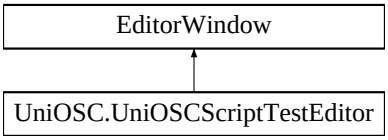
16.41.3.1 float UniOSC.UniOSCScaleGameObject.scaleFactor = 1

16.41.3.2 Transform UniOSC.UniOSCScaleGameObject.transformToScale

16.42 UniOSC.UniOSCScriptTestEditor Class Reference

Editor for the administration of OSCconnections, mapping files.

Inheritance diagram for UniOSC.UniOSCScriptTestEditor:



Public Member Functions

- void OnEnable ()
- void OnDisable ()

Properties

- static UniOSCScriptTestEditor Instance `[get]`
- static bool IsOpen `[get]`

16.42.1 Detailed Description

Editor for the administration of OSCconnections, mapping files.

You can also trace the OSC data flow .

16.42.2 Member Function Documentation

16.42.2.1 void UniOSC.UniOSCScriptTestEditor.OnDisable ( )

16.42.2.2 void UniOSC.UniOSCScriptTestEditor.OnEnable ( )

16.42.3 Property Documentation

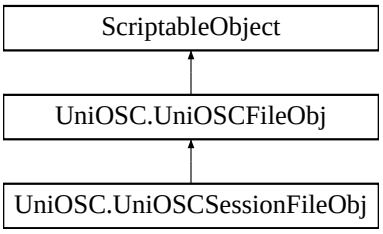
16.42.3.1 UniOSCScriptTestEditor UniOSC.UniOSCScriptTestEditor.Instance `[static],[get]`

16.42.3.2 bool UniOSC.UniOSCScriptTestEditor.IsOpen `[static],[get]`

16.43 UniOSC.UniOSCSessionFileObj Class Reference

OSC Session file class .

Inheritance diagram for UniOSC.UniOSCSessionFileObj:



Public Member Functions

- void OnEnable ()
- void AddOSCSessionItem ()

  Adds a new OSC Session item.
- void RemoveOSCSessionItem (UniOSCSessionItem obj)

  Removes the OSC Session item from the list and destroys the item instance.
- void OnOSCMessageReceived (object sender, UniOSCEventArgs args)

  Checks if we are in learning mode an writes the OSC message address into the address property of a session item that is in learn mode(when user hold down the 'learn' button in the editor.

Public Attributes

- List< UniOSCSessionItem > oscSessionItemList

Additional Inherited Members

16.43.1 Detailed Description

OSC Session file class .

Every Session file get stored as a .asset file. You can copy & paste a Session file to another Unity project but you have to aware that sometimes Unity changes the serialization format. If you have any trouble go to 'Edit/Project Settings/Editor' and change the seriaization mode to 'Force Text' and then switch back to 'Force Binary'

16.43.2 Member Function Documentation

16.43.2.1 void UniOSC.UniOSCSessionFileObj.AddOSCSessionItem ( )

Adds a new OSC Session item.

16.43.2.2 void UniOSC.UniOSCSessionFileObj.OnEnable ( )

16.43.2.3 void UniOSC.UniOSCSessionFileObj.OnOSCMessageReceived (object *sender,* UniOSCEventArgs *args*)

Checks if we are in learning mode an writes the OSC message address into the address property of a session item that is in learn mode(when user hold down the 'learn' button in the editor.

Parameters

sender	Sender.
args	UniOSCEventArgs that contains the OSC message

16.43.2.4 void UniOSC.UniOSCSessionFileObj.RemoveOSCSessionItem (UniOSCSessionItem *obj*)

Removes the OSC Session item from the list and destroys the item instance.

UniOSC.UniOSCSessionItem.OnOSCSessionItemDelete

Parameters

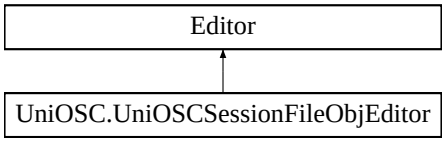
obj	UniOSCSessionItem to remove.

16.43.3 Member Data Documentation

16.43.3.1 List<UniOSCSessionItem> UniOSC.UniOSCSessionFileObj.oscSessionItemList

16.44 UniOSC.UniOSCSessionFileObjEditor Class Reference

Inheritance diagram for UniOSC.UniOSCSessionFileObjEditor:



Public Member Functions

- override void OnInspectorGUI ()

Static Public Member Functions

- static void Init ()
- static void OnGUI_OSCSessionData_Editor (UniOSCSessionFileObj obj, float screenWidth, float screenHeight)
- static void OnGUI_OSCSessionData_Inspector (UniOSCSessionFileObj obj, float screenWidth, float screenHeight)

Static Public Attributes

- static GUIStyle style

16.44.1 Member Function Documentation

16.44.1.1 static void UniOSC.UniOSCSessionFileObjEditor.Init ( ) [static]

16.44.1.2 static void UniOSC.UniOSCSessionFileObjEditor.OnGUI_OSCSessionData_Editor (UniOSCSessionFileObj *obj*, float *screenWidth*, float *screenHeight*) [static]

16.44.1.3 static void UniOSC.UniOSCSessionFileObjEditor.OnGUI_OSCSessionData_Inspector (UniOSCSessionFileObj *obj*, float *screenWidth*, float *screenHeight*) [static]

16.44.1.4 override void UniOSC.UniOSCSessionFileObjEditor.OnInspectorGUI ( )

16.44.2 Member Data Documentation

16.44.2.1 GUIStyle UniOSC.UniOSCSessionFileObjEditor.style [static]

16.45 UniOSC.UniOSCSessionItem Class Reference

Uni OSC mapping item.

Public Member Functions

- UniOSCSessionItem ()
- UniOSCSessionItem (UniOSCSessionFileObj _hostObj)

  Initializes a new instance of the UniOSC.UniOSCSessionItem class.
- void OnOSCSessionItemDelete ()

  Removes this item from the UniOSCSessionFileObj host object.Afterwards it gets destroyed.

Public Attributes

- UniOSCSessionFileObj hostObj
- string address = ""
- List< string > data = new List<string>()
- List< string > dataTypeList = new List<string>()
- bool isLearning
- const int MAXWIDTH = 250
- const int MAXHEIGTH = 150
- bool collapsed = true

16.45.1 Detailed Description

Uni OSC mapping item.

<author> Stefan Schlupek </author>

16.45.2 Constructor & Destructor Documentation

16.45.2.1 UniOSC.UniOSCSessionItem.UniOSCSessionItem ( )

16.45.2.2 UniOSC.UniOSCSessionItem.UniOSCSessionItem (UniOSCSessionFileObj *_hostObj*)

Initializes a new instance of the UniOSC.UniOSCSessionItem class.

Parameters

_hostObj	The UniOSCSessionFileObj object that host the item

16.45.3 Member Function Documentation

16.45.3.1 void UniOSC.UniOSCSessionItem.OnOSCSessionItemDelete ( )

Removes this item from the UniOSCSessionFileObj host object.Afterwards it gets destroyed.

16.45.4 Member Data Documentation

16.45.4.1 string UniOSC.UniOSCSessionItem.address = ""

16.45.4.2 bool UniOSC.UniOSCSessionItem.collapsed = true

16.45.4.3 List<string> UniOSC.UniOSCSessionItem.data = new List<string>()

16.45.4.4 List<string> UniOSC.UniOSCSessionItem.dataTypeList = new List<string>()

16.45.4.5 UniOSCSessionFileObj UniOSC.UniOSCSessionItem.hostObj

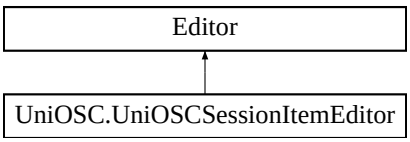
16.45.4.6 bool UniOSC.UniOSCSessionItem.isLearning

16.45.4.7 const int UniOSC.UniOSCSessionItem.MAXHEIGTH = 150

16.45.4.8 const int UniOSC.UniOSCSessionItem.MAXWIDTH = 250

16.46 UniOSC.UniOSCSessionItemEditor Class Reference

Inheritance diagram for UniOSC.UniOSCSessionItemEditor:



Public Member Functions

- void OnEnable ()

Static Public Member Functions

- static void OnGUI_Editor (UniOSCSessionItem obj)
- static void OnGUI_Inspector (UniOSCSessionItem obj)

16.46.1 Member Function Documentation

16.46.1.1 void UniOSC.UniOSCSessionItemEditor.OnEnable ( )

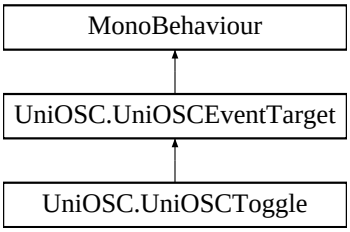
16.46.1.2 static void UniOSC.UniOSCSessionItemEditor.OnGUI_Editor (UniOSCSessionItem *obj*) [static]

16.46.1.3 static void UniOSC.UniOSCSessionItemEditor.OnGUI_Inspector (UniOSCSessionItem *obj*) [static]

16.47 UniOSC.UniOSCToggle Class Reference

With this class you can toggle most of the Unity Components on/off The data of the OSC message should be only 0(off) or 1(on)

Inheritance diagram for UniOSC.UniOSCToggle:



Public Member Functions

- void UpdateComponentState ()

  Updates the state of the component.
- override void OnEnable ()

  Enable this component and reinitialize.
- override void OnOSCMessageReceived (UniOSCEventArgs args)

  You should override this method in a subclass to handle the OSC data.

Public Attributes

- Component componentToToggle
- bool toggleState

Additional Inherited Members

16.47.1 Detailed Description

With this class you can toggle most of the Unity Components on/off The data of the OSC message should be only 0(off) or 1(on)

16.47.2 Member Function Documentation

16.47.2.1 override void UniOSC.UniOSCToggle.OnEnable ( ) [virtual]

Enable this component and reinitialize.

If a property of the component is changed via the inspector we force a OnEnable to update the status of the component. In general the component disconnects from all OSCConnections and try to find a new OSCConnection to connect to with a matching port. If you change properties via code you should call this explicit.

Reimplemented from UniOSC.UniOSCEventTarget.

16.47.2.2 override void UniOSC.UniOSCToggle.OnOSCMessageReceived (UniOSCEventArgs *args*) [virtual]

You should override this method in a subclass to handle the OSC data.

Parameters

args	The current OSCEventArgs object

Implements UniOSC.UniOSCEventTarget.

16.47.2.3 void UniOSC.UniOSCToggle.UpdateComponentState ( )

Updates the state of the component.

(enabled)

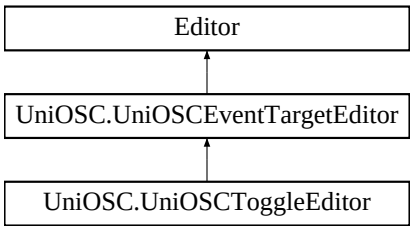
16.47.3 Member Data Documentation

16.47.3.1 Component UniOSC.UniOSCToggle.componentToToggle

16.47.3.2 bool UniOSC.UniOSCToggle.toggleState

16.48 UniOSC.UniOSCToggleEditor Class Reference

Inheritance diagram for UniOSC.UniOSCToggleEditor:



Public Member Functions

- override void OnEnable ()
- override void OnInspectorGUI ()

Protected Member Functions

- void ForceUpdate ()

Protected Attributes

- UniOSCToggle _targetToggle
- SerializedProperty ComponentToToggleProp
- SerializedProperty ToggleStateProp
- int _componentIndex = 0
- bool _updateFlag

16.48.1 Member Function Documentation

16.48.1.1 void UniOSC.UniOSCToggleEditor.ForceUpdate ( ) [protected]

16.48.1.2 override void UniOSC.UniOSCToggleEditor.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventTargetEditor.

16.48.1.3 override void UniOSC.UniOSCToggleEditor.OnInspectorGUI ( )

16.48.2 Member Data Documentation

16.48.2.1 int UniOSC.UniOSCToggleEditor._componentIndex = 0 [protected]

16.48.2.2 UniOSCToggle UniOSC.UniOSCToggleEditor._targetToggle [protected]

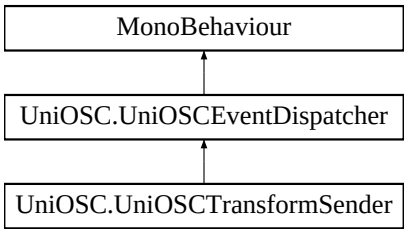
16.48.2.3 bool UniOSC.UniOSCToggleEditor._updateFlag [protected]

16.48.2.4 SerializedProperty UniOSC.UniOSCToggleEditor.ComponentToToggleProp [protected]

16.48.2.5 SerializedProperty UniOSC.UniOSCToggleEditor.ToggleStateProp [protected]

16.49 UniOSC.UniOSCTransformSender Class Reference

Inheritance diagram for UniOSC.UniOSCTransformSender:



Public Member Functions

- override void OnEnable ()
- override void OnDisable ()

Public Attributes

- GameObject trackedGameObject

Protected Member Functions

- override void _Update ()

Additional Inherited Members

16.49.1 Member Function Documentation

16.49.1.1 override void UniOSC.UniOSCTransformSender._Update ( ) [protected], [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.49.1.2 override void UniOSC.UniOSCTransformSender.OnDisable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.49.1.3 override void UniOSC.UniOSCTransformSender.OnEnable ( ) [virtual]

Reimplemented from UniOSC.UniOSCEventDispatcher.

16.49.2 Member Data Documentation

16.49.2.1 GameObject UniOSC.UniOSCTransformSender.trackedGameObject

16.50 UniOSC.UniOSCTransmitter Class Reference

Public Member Functions

- UniOSCTransmitter ()
- UniOSCTransmitter (string ipAddress, int port)
- UniOSCTransmitter (IPAddress ipAddress, int port)
- UniOSCTransmitter (IPAddress ipAddress, TransmissionType ttype, int port)
- bool Connect ()
- void Close ()
- bool SendOSCMessage (object sender, UniOSCEventArgs args)

Properties

- IPAddress IPAddress [get]
- int Port [get]
- TransmissionType transmissionType [get]

Events

- EventHandler< ExceptionEventArgs > OSCErrorOccured

16.50.1 Constructor & Destructor Documentation

16.50.1.1 UniOSC.UniOSCTransmitter.UniOSCTransmitter ( )

16.50.1.2 UniOSC.UniOSCTransmitter.UniOSCTransmitter (string *ipAddress,* int *port*)

16.50.1.3 UniOSC.UniOSCTransmitter.UniOSCTransmitter (IPAddress *ipAddress,* int *port*)

16.50.1.4 UniOSC.UniOSCTransmitter.UniOSCTransmitter (IPAddress *ipAddress,* TransmissionType *ttype,* int *port*)

16.50.2 Member Function Documentation

16.50.2.1 void UniOSC.UniOSCTransmitter.Close ( )

16.50.2.2 bool UniOSC.UniOSCTransmitter.Connect ( )

16.50.2.3 bool UniOSC.UniOSCTransmitter.SendOSCMessage (object *sender,* UniOSCEventArgs *args*)

16.50.3 Property Documentation

16.50.3.1 IPAddress UniOSC.UniOSCTransmitter.IPAddress [get]

16.50.3.2 int UniOSC.UniOSCTransmitter.Port [get]

16.50.3.3 TransmissionType UniOSC.UniOSCTransmitter.transmissionType [get]

16.50.4 Event Documentation

16.50.4.1 EventHandler<ExceptionEventArgs> UniOSC.UniOSCTransmitter.OSCErrorOccured

Index