OSC simpl Reference

Version 1.1

MonoBehaviours

- OscIn
- OscOut

Classes

- OscBundle
- OscImpulse
- OscMessage
- OscTimeTag

Enums

- OscReceiveMode
- OscRemoteStatus
- OscSendMode

OscIn

MonoBehaviour for receiving OscMessage objects.

int port
Gets the local port that this application is set to listen to. (read only). To set, call the Open method.

OscReceiveMode mode
Gets the transmission mode (read only). Can either be UnicastBroadcast or Multicast. The mode is automatically derived from arguments passed to the Open method.

string multicastAddress
Gets the remote address to the multicast group that this application is set to listen to (read only). To set, call the Open method and provide a valid multicast address.

static string ipAddress
Gets the local network IP address for this device (read only). Returns an empty string if the address is not available. In that case ensure that your device is connected to a network. Using a VPN may block you from getting the local IP.

bool isOpen
Indicates whether the Open method has been called and the object is ready to receive.

bool filterDuplicates
When enabled, only one message per OSC address will be forwarded every Update call. The last (newest) message received will be used. Default is true.

bool addTimeTagsToBundledMessages
When enabled, timetags from bundles are added to contained messages as last argument. Incoming bundles are never exposed, so if you want to access a time tag from a incoming bundle then enable this. Default is false.

int messageCount
Gets the number of messages received since last update.

OscMessageEvent onAnyMessage
Add listener to this event to receive a call when any OscMessage is received on the specified port. The event is influenced by the state of 'filterDuplicates' property.

bool Open(int port, string multicastAddress = "")
Open to receive messages on specified port and (optionally) from specified multicast IP address. Returns success status.

void Close()
Close and stop receiving messages.

void Map(string address, UnityAction method)
Request that specified method is invoked when a message with specified OSC address is received. Methods with one argument of any of the following types are supported: float, double, int, long, string, char, bool, Color32, byte[] and OscMessage (for entire message).

void Unmap(UnityAction method)
Request that specified method is no longer invoked. Note that only mappings made at runtime can be unmapped.

void Unmap(string address)
Request that all methods that are mapped to specified OSC address will no longer invoked. This is useful for unmapping delegates.

OscOut

MonoBehaviour for sending OscMessage and OscBundle objects.

int port
Gets the port to be send to on the target remote device (read only). To set, call the Open method.

OscSendMode mode
Gets the transmission mode (read only). Can either be UnicastToSelf, Unicast, Broadcast or Multicast. The mode is automatically derived from the IP address passed to the Open method.

string ipAddress
Gets the IP address of the target remote device (read only). To set, call the 'Open' method.

bool isOpen
Indicates whether the Open method has been called and the object is ready to send.

OscRemoteStatus remoteStatus
Gets the remote connection status (read only). Can either be Connected, Disconnected or Unknown.

int messageCount
Gets the number of messages send since last update.

bool multicastLoopback
Indicates whether outgoing multicast messages are also delivered to the sending application. Default is true.

bool bundleMessagesOnEndOfFrame
When enabled, messages will automatically be buffered in a single OscBundle and send at the end of the frame (i.e. Unity's WaitForEndOfFrame). Default is false.

bool Open(int port, string ipAddress = "")
Open to send messages to specified port and (optional) IP address. If no IP address is given, messages will be send locally on this device. Returns success status.

void Close()
Close and stop sending messages.

bool Send(string address, params object[] args)
Send a OscMessage with specified address and arguments. Returns success status.

bool Send(OscPacket packet)
Send an OscMessage or OscBundle. Returns success status.

OscBundle

Class representing an OSC bundle. Bundles have a OscTimeTag and can contain OscMessage and OscBundle objects.

OscTimeTag timeTag
Gets or sets the timetag for this bundle.

List<OscPacket> packets
Gets the list of OscMessage and OscBundle objects.

OscBundle()
Constructor for creating a bundle with a timetag containing the current time.

OscBundle(OscTimeTag timeTag)
Constructor for creating a bundle with specified timetag.

void Add(OscPacket packet)
Add a OscMessage or OscBundle to this bundle. Shorthand for bundle.packets.Add.

void Clear()
Remove all OscMessage and OscBundle object in this bundle, and do so recursively for all contained bundles.

OscImpulse

Class representing the OSC 1.1 argument type Impulse. In OSC 1.0 this was called 'Infinitum'.

OscMessage

Class representing an OSC message. Messages have an OSC address an a number of OSC arguments.

Supported argument types and their Unity translations:

- float (f) <-> float
- integer (i) <-> int
- string (s), symbol (S) <-> string
- blob (b) <-> byte[]
- long (h) <-> long
- double (d) <-> double
- boolean (T,F) <-> bool
- character (c) <-> char
- color (r) <-> Color32, Color
- timetag (t) <-> OscTimeTag, DateTime
- impulse (I) <-> OscImpulse
- nil (N)<-> null

string address
Gets or sets the address of the message. Must start with '/'.

List<object> args
Gets or sets the OSC arguments. Manipulate the list directly.

OscMessage(string address, params object[] argParams)
Constructor taking an address and an arbitrary number of OSC type arguments.

void Add(float value)
Add argument. Supports all other OSC argument types. Shorthand for message.args.Add.

void Clear()
Clear all arguments. Shorthand for message.args.Clear.

bool TryGet(int index, out float value)
Tries to get argument at index.

OscTimeTag

Class representing a OSC timetag. OscTimeTag objects are send implicitly with bundles and explicitly as message arguments. Incoming bundles are never exposed to the user. Instead, the contained messages are unwrapped automatically and send to mapped methods. If you want to receive timetags from bundles, then enable 'addTimeTagsToBundledMessages' on OscIn and grab the timetag from the last argument of your incoming bundled message. Timed scheduling of received bundled messages is not supported.

DateTime time
Gets or sets the time with DateTime tick precision.

bool immediately
Gets or sets the 'immediately' flag. Some OSC implementations may interpret the flag as "process immediately on receipt" - as opposed to "schedule the recieved bundle for processing at time" - while other implementations may ignore it. Default is true.

ulong oscNtp
Get or sets the OSC flavoured NTP encoded value in which a time and a 'immediately' flag is stored. Don't manipulate this property directly, unless you have read the OSC 1.0 specification.

OscReceiveMode

Enum representing the mode of transmission for OscIn. Can either be UnicastBroadcast or UnicastBroadcastMulticast.

OscRemoteStatus

Enum representing the connection status to a remote device. Can be either Connected, Disconnected or Unknown. For broadcast and multicast mode the status will always be Unknown.

OscSendMode

Enum representing the mode of transmission for OscOut. Can either be UnicastToSelf, Unicast, Broadcast or Multicast.