

# OSCXtra

The OSCXtra is developed at the Interactive Institute research studio Sonic in Piteå, Sweden (<http://www.tii.se/sonic>). The OSCXtra works on both Windows XP and Mac OS X, is freeware and delivered on an “as is” basis.

## Method details

OSCXtra has only four methods, three of which are calls from Lingo and the fourth is a callback from OSCXtra to Lingo.

### **new object me**

new creates a new instance of the OSCXtra.

Example:

```
gOSCXtra = new(xtra "OSCXtra")
```

### **oscInit object me, string aRemoteIPAddress, integer aRemotePort, integer aLocalPort**

oscInit initializes an instance of OSCXtra with an IP-address to send messages to, a port number to send messages to and a local port to listen for incoming messages on. When an incoming OSC message is received, OSCXtra will look for a Lingo handler called oscInWaiting defined in a movie script. If OSCXtra finds such a handler it is called (see below for details.)

Example:

```
gOSCXtra.oscInit("192.168.0.1", 10000, 10001)
-- oscSend will hereafter send OSC messages to IP address
-- 192.168.0.1 on port 10000
-- OSCXtra will listen for incoming OSC messages on port
-- 10001
```

### **oscSend object me, string anOSCAddress, \***

oscSend sends OSC messages to the address and port stated in the call to oscInit (described above). The OSC address is a string that must start with the character “/”.

After the OSC address follows a variable number of parameters. The value types of these parameters must only be strings, integers, floats or linear Lingo lists. In the case of a linear Lingo list, the entries in the list must only be of types strings, integers or floats.

Example:

```
gOSCExtra.oscSend("/interactionSet", "MusicPlease", 1)

argList = []
argList.add("Velvet Underground") -- String parameter
argList.add(3.1415)                -- float parameter
argList.add(42)                    -- int parameter
gOSCExtra.oscSend("/soundInstallation, argList)
```

## oscInWaiting

If the OSCXtra receives an OSC message on its local port, it looks for a handler called `oscInWaiting` in a movie script scope. If a handler called `oscInWaiting` is found, that handler is called with a variable number of parameters. The first parameter is the OSC address as a string. The rest of the parameters are OSC parameters of types strings, integers or floats.

Example:

```
on oscInWaiting
-----
-- Function: An OSC-message arrived
-- Input:    An OSC-address + a number of OSC parameters
-- Output:
-----

-- Is it a proper OSC message
if not stringP(param(1)) then return

-- Assemble an OSC message in the form of a Lingo
-- property list
OSCMes = [#oscAddress:param(1), #oscParams:[]]
repeat with i = 2 to the paramCount
    OSCMes.oscParams.append(param(i))
end repeat

-- Add the message to the list of messages to handle
gOSCList.append(OSCMes)
end
```