BufferDef / BufferFunc

BufferDef maintains the registry of BufferFunc and provides a global interface to all nodes

BufferDef creates instances of client side Buffer while allocating, streaming or reading data to the server depending on the arguments. This is a conveneint way to access often used methods and to keep buffers organized. EventDef, can reference BufferDefs by synbol/name in the process of instancing synths.

BufferDef.new(name, server, arg, arg1, ..., argN)
name - a symbol
server - a Server
arg - a string, a SimpleNumber or an Array.
when arg is
a String - .read is performed with the path to a sound file
a SimpleNumber - .alloc is performed with arg1 as numFrames and arg2 is numChannels
an Array - .sendCollection is performed with arg1 the collection to stream into the buffer. AoA results in a multichannel Buffer

.write(...args)
write the contents of the buffer to a file using Buffer.write method
args - from Buffer.write: .write(path, headerFormat: "aiff", sampleFormat: "int24", numFrames: -1, startFrame: 0, leaveOpen: false, completionMessage)

.writeMsg(...args)
write the contents of the buffer to a file using Buffer.writeMsg method
args - from Buffer.write: .write(path, headerFormat: "aiff", sampleFormat: "int24", numFrames: -1, startFrame: 0, leaveOpen: false, completionMessage)

.perform (selector ...args)
selector - a Symbol

.zero
zero the buffer

.bufnum
return the buffer number

s.options.sampleRate_(44100); s.reboot;

//
// load a buffer from file
//
BufferDef(\asdf, s, "/Users/jeffcarey/Documents/000_work/000_projects/2023/000_field_recording/20230331/230330_001.WAV");

BufferDef(\asdf)

```
BufferDef(\asdf).buffer
BufferDef(\asdf).bufnum
BufferDef(\asdf).perform(\plot)

//
// create and empty buffer
//
Konnect
BufferDef(\onesecond, s, s.options.sampleRate, 2)

BufferDef(\onesecond)
BufferDef(\onesecond).buffer
BufferDef(\onesecond).bufnum
BufferDef(\onesecond).buffer.plot

//
//build a buffer from an array of samples
//

a = Array.fill(4410, { 1.0.rand2 }); b = Array.fill(4410, { 1.0.rand2 });
BufferDef(\noises, s, [a,b]);

BufferDef(\noises).buffer
BufferDef(\noises).perform(\plot)
```