

OSCmessageDecoder

General

OSCmessageDecoder decodes (deserializes) a single OSC message. This channel requires an OSCpackageDecoder plug-in for pre-processing, if the OSC client is sending bundled messages. Supported OSC argument types here are integer, float and text. The Quest3D partner channel for encoding messages is OSCmessageEncoder.

Children

1. IN OSC Message (Buffer) -- required
2. OUT OSC Address Pattern (Text) -- required
3. OUT OSC Argument (Text) -- growing link list
4. OUT OSC Argument (Value) -- growing link list
5. OUT OSC Type Tag (Text) -- optional

Description

Briefly speaking, OSC Messages are defined as a sequence of an Address Pattern followed by none to many arguments. Incoming integers and floats are both mapped to Quest3D value channels. OSCdecoder maps all supported incoming arguments to a growing child list of text and value channels in their local order of appearance. For example, an OSC sequence of text/integer/text/float/float is mapped to the text/text/value/value/value child sequence. The OSC Type Tag text output is provided for additional information; it can be used to clarify any ambiguity.

Single message use:

→ See Sample scenario below.

Bundle use:

→ See OSCmessageDecoder

The channel behaves as a value channel, return values are:

- >0 Number of OSC arguments decoded
- 0 ready / idle
- 1 improper child assignment found
- 2 license restrictions apply
- 3 OSC bundle received (please use additional OSCpacketDecoder for bundles)

Volume, runtime duration, and expiration restrictions are enforced with trial and lite licenses. The applicable restrictions are listed in the editor debug window.

Tips

- Use OSCDUMP.exe for test purposes.

- Inspect OSC Type Tag output for test purposes.
- Check Debug Window for error explanations.
- On slow systems UDP packages may queue up before the UDPreader.
- The output sequence of arguments is kept, however, sorted into the corresponding channel type list.

Legal note: Permission granted for evaluation and educational purposes only (trial version), commercial use requires an explicit license agreement

Contact: quest3d.godbersen.eu

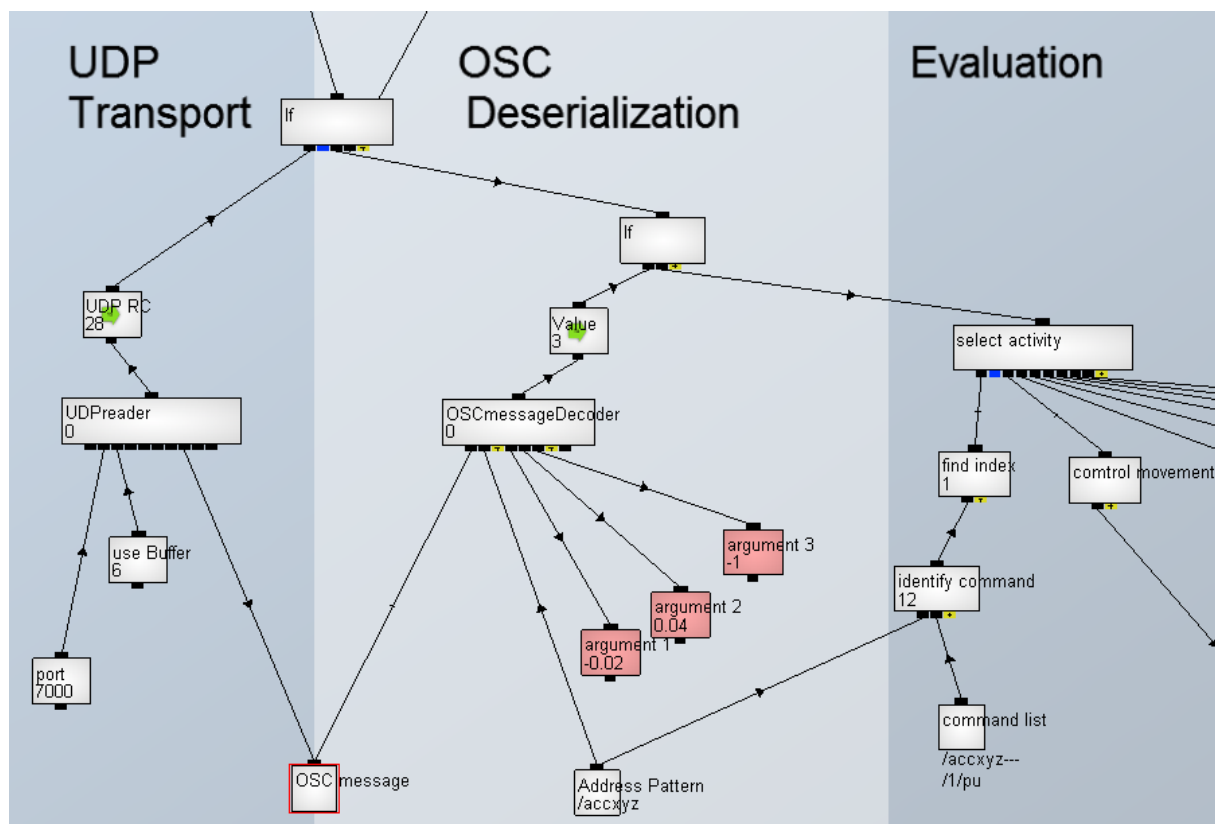
Known Problems:

-

Revision History:

20110530 initial version

Sample scenario:



The application is calling UDPreader each frame. UDPreader listens at default IP (0.0.0.0) and port 9000. UDPreader stores the Byte stream in the OSCmessage Buffer channel, if new content has arrived. And returns with the number of Bytes received.

OSCmessageDecoder is now called. If it succeeded decoding the message, it will post the extracted Texts and Values in child channels and return with the number of arguments found.

Deserialization: This positive feedback will start the channel switch. "select activity" will invoke the specific processing sub-tree associated with a particular Address Pattern. Identifying the commands is done here by text search: All expected patterns are listed in lines of equal length in the "command list" text channel. Identifying command delivers the raw position of a text match, which finally has to be divided by the line length to get the index. An alternative would be to use LUA or OO.

See also:

- OSCreader for simple messages

Sample application:

1. OSC Dump
2. Walk in the Woods