

UDPReader

General

Reads a single UDP packet. Binds initially to a local socket. Reads packet and decodes it according to the selector value. Data type choices are Text, Value, Vector, Matrix, and Buffer.

Children

0. IN local listening IP (text, default 0.0.0.0)
1. IN local listening port (value, default 5000)
2. IN data type selector (value): 0 text (default) , 1 value, 2 vector, 3 matrix, 4 buffer, 5 Hexdump
3. OUT data (text)
4. OUT data (value)
5. OUT data (vector)
6. OUT data (matrix)
7. OUT Buffer (buffer)
8. OUT sender IP address (text)
9. OUT sender IP port (value)

Description

Reads local IP, local port number and target data type selector with the first call only. This settings will be valid during the remaining application live time. Tries to read UDP packets send to this recipient. Returns with 0 if no packet present. Decodes the payload according to the selector setting if a UDP packet is available and its size is appropriate. Returns with number of bytes successfully interpreted or an error code <0. Trial time expiration returns a -2.

Children 8 and 9 provide the actual sender IP address and its port number. This allows to identify different source, if needed. Hexdump is delivered on child 3 (Text) and is only intended for testing purposes of short payloads.

Tips

- Multiple UDPReader channels in one application are OK. Use different port numbers to avoid binding conflicts.
- The usage of the default network adapter address 0.0.0.0 is sufficient for most scenarios, including broadcast receipt.
- Value, vector and matrix data structures follow DirectX conventions, little endian. Make sure your peer UDP station acts appropriately.
- Handle firewalls beforehand (application may crash during firewall dialogs).
- Length mismatch with value, vector and matrix results in a discard.
- Use hexdump for an insight into any payload content.
- Text length 0 acceptable.
- Content preset in the OUT channels may be outdated. Ensure immediate processing after arrival.
- Some socket errors: 10048 Address already in use, 10049 Can't assign requested address. A list of all Socket error codes: <http://msdn.microsoft.com/en-us/library/ms740668>.
- Buffer usage is required for OSCmessageDecoder and OSCpackageDecoder. However, almost any channel content can be transported between Quest3D applications with SaveChannelToBuffer / LoadBufferIntoChannel.

Legal note: Permission granted for evaluation and educational purposes only (trial version), commercial use requires an explicit license agreement (full version). Acknowledgements: Contains code fragments from Jeremy Friesner under GNU LGPL license.

Volume, runtime duration, and expiration restrictions are enforced with trial licenses. Details can be found by reading the output of the Debug Window, and on my web page. Be prepared to cope with this situation in your application.

Supported Quest3D Versions:

- 4.3.2
- 5.0 (x64 only) Win32 on request

Known Problems:

- none

Revision History:

2012 10 12 minor changes, V 5.0 support
2011 02 01 initial release

Sample Scenario:

See UDPwriter_Notes.pdf

Contact: quest3d.godbensen.eu

Tutorial:

For testing purposes I recommend this small utility
http://www.hw-group.com/products/hercules/index_en.html