

QUALISYS MATLAB CLIENT v1.9

Qualisys MATLAB Client (QMC) is a mex-function for the software suit MATLAB from MathWorks. The purpose of the mex-function is to collect data from Qualisys Track Manager via the built in real-time server and output this to MATLAB.

The QMC package consists of 3 files.

- QMC.mexw32 (32 bit version)
- QMC.mexw64 (64 bit version)
- QMC_conf.txt
- Qualisys MATLAB Client.pdf

The QMC.mex-files are 32bit and 64bit Windows versions of the Qualisys MATLAB Client. QMC_conf.txt is the default configuration file and contains configuration info for the client. Qualisys MATLAB Client.pdf is this documentation.

INSTALLATION

Run the setup file “Qualisys MATLAB Client Setup.exe” and follow the on screen instructions. The setup program will install the mex files, configuration file and documentation file. It will also add the path to these files to the MATLAB search path.

USAGE

Connect

Next step is to initiate connection with the QTM RT server. There are two ways to connect to the QTM RT server.

Alt. 1

Connect using selected configuration file. If no path is included in the filename, MATLAB will search for the file in the current MATLAB folder. If the connection succeeded, the filename is stored as last configuration file.

```
>> q = QMC('c:\my_files\QMC_conf.txt')
```

Alt 2.

Connect using last configuration file used. See Alt 1, above. If no last configuration file is stored, QMC will attempt to load default configuration file QMC_conf.txt from MATLABs working directory.

```
>> q = QMC
```

Depending on where you install the Qualisys MATLAB Client files and which Windows version you use,

you might need Administrator rights to edit the default configuration file. In this case Alt2 will enable you to change the default configuration file so it can be edited by a standard user.

Alt1 and Alt2 above, initiates connection with QTM and adds the object handle 'q'. 'q' must be initiated in order for QMC to function. QMC uses the settings for IP address and port in the configuration file to connect to the QTM RT server.

The <STREAM> setting in the configuration file tells the client if it shall poll the data from the RT server via TCP or stream the data via UDP or TCP.

<STREAM:0> Polling over TCP.

<STREAM:1> Streaming over UDP.

<STREAM:2> Streaming over TCP.

When using streaming over UDP, <UDPPORT:n> can be used to set the port number used for streaming, where n is the port number. Set n to 0 to use a random port.

In case of polling, the data is updated each time QMC is called. When using streaming, the <FREQ> setting will decide how often the data will be sent from QTM by the RT server.

<FREQ:AllFrames> Send RT frames with the camera frequency.

<FREQ:Frequency:n> Stream data in n Hz.

<FREQ:FrequencyDivisor:n> Stream data with the camera frequency divided by n.

When streaming over UDP, the last frame in the in-buffer will be retrieved with QMC, see below. If there are older frames in the buffer, they will be discarded. When streaming over TCP, the first frame in the in-buffer will be retrieved with QMC, see below. No frames will be discarded and they can be retrieved from the in buffer with another QMC call.

The configuration file also tells QMC which data components to retrieve from QTM. See example configuration file in the end of the document. Supported components are:

- 2D
- 2D linearized
- 3D
- 3D no labels
- 3D with residual
- 3D no labels with residual
- AnalogSingle (Only returns the latest analog value.)
- ForceSingle (Only returns the latest force value.)
- 6DOF
- 6DOF Euler
- 6DOF with residual
- 6DOF Euler with residual

Retrieve data

```
>> data = QMC(q)
```

Retrieves data frame from QTM. Each time the function is called, one sample is extracted.

```
>> data = QMC(q, camera_no)
```

Retrieves 2D data from QTM. 'camera_no' selects which camera to retrieve data from.

Frame info

```
>> frameinfo = QMC(q, 'frameinfo')
```

Retrieves frame number and timestamp from the last data frame read.

Events

```
>> QMC(q, 'event')
```

Prints last QTM event.

Disconnect

```
>> QMC(q, 'disconnect')
```

Disconnects from the QTM RT server and destroys the object handle 'q'.

ANALOG DATA

Analog data works similar like all other data types, with the exception that another setting must be setup. Because there can be several analog devices with several channels, both of these must be written in the configuration file.

<Analog:1> represents how many devices that should be used when gathering data and <CHANNEL:1> is used to determine how many channels should be used on each device.

EXAMPLES

There is a QTM file called 6dofbodies_and_analog.qtm included in the MATLAB client packet files. This file can be used to simulate a real time connection in QTM. Open 6dofbodies_and_analog.qtm in QTM and select "Run real time processing on file..." under the Capture menu in QTM. QTM will now stream data from the file, as if it was connected to a camera system in real time.

If you don't wish to simulate your real time, you only have to select New in QTM to connect to your camera system. The QTM RT server is always running. All data that is available in QTM is also available on the RT interface.

EXAMPLE, retrieving 3 6DOF bodies

To retrieve 6DOF data from QTM running on the same computer as MATLAB, the QMC_config.txt shall look like this:

```
# IP address of the QTM_RT server
<IP:127.0.0.1>    # Local host

# Port used
<PORT:22222>
<UDPPORT:0>      # Use random port

<STREAM:0>

<3D:0>
<3D-NoLabels:0>
<AnalogSingle:0>
<ForceSingle:0>
<6DOF:0>
<6DOF-Euler:3>
<2D:0>
<2D-Lin:0>
<3D-Residual:0>
<3D-NoLabels-Residual:0>
<6DOF-Residual:0>
<6DOF-Euler-Residual:0>
```

With the configuration set, you may initiate the client by entering:

```
>> q = QMC
```

In MATLAB, where q is a variable of your choice, q contains an ObjectHandle, which makes sure that the socket-connection to QTM is held open throughout your data-transfer.

To start extracting data, simply start QMC again, using q as in input variable as such:

```
>> data = QMC(q)
```

The variable data should contain 3 rows with 6 columns (there is one row for each 6DOF body).

To disconnect, issue the following command:

```
>> QMC(q, 'disconnect')
```

EXAMPLE, retrieving several data types

To retrieve unlabeled 3D data and 6DOF-Euler data from QTM running on the same computer as MATLAB, QMC_config.txt shall look like this:

```
# IP address of the QTM_RT server
<IP:127.0.0.1>    # Local host

# Port used
<PORT:22222>
<UDPPORT:0>      # Use random port

<STREAM:0>

<3D:15>
<3D-NoLabels:0>
<AnalogSingle:0>
<ForceSingle:0>
<6DOF:0>
<6DOF-Euler:3>
<2D:0>
<2D-Lin:0>
<3D-Residual:0>
<3D-NoLabels-Residual:0>
<6DOF-Residual:0>
<6DOF-Euler-Residual:0>
```

With the configuration set, you may initiate the client by entering:

```
>> q = QMC
```

In MATLAB, where q is a variable of your choice, q contains an ObjectHandle, which makes sure that the socket-connection to QTM is held open throughout your data-transfer.

To start extracting data, simply start QMC again, using q as in input variable as such:

```
>> [data_3d data_6dof] = QMC(q)
```

The output is put in different variables when sending several data types, starting with the one listed highest in the configuration file.

To disconnect, issue the following command:

```
>> QMC(q, 'disconnect')
```

EXAMPLE, retrieving analog data

To retrieve data from 3 analog channels on one analog device from QTM running on the same computer as MATLAB, the QMC_config.txt shall look like this:

```
# IP address of the QTM_RT server
<IP:127.0.0.1>    # Local host

# Port used
<PORT:22222>
<UDPPORT:0>      # Use random port

<STREAM:0>

<3D:0>
<3D-NoLabels:0>
<AnalogSingle:1>
<ForceSingle:0>
<6DOF:0>
<6DOF-Euler:0>
<2D:0>
<2D-Lin:0>
<3D-Residual:0>
<3D-NoLabels-Residual:0>
<6DOF-Residual:0>
<6DOF-Euler-Residual:0>

# Max number of analog channels to receive data from.
<CHANNEL:3>
```

With the configuration set, you may initiate the client by entering:

```
>> q = QMC
```

In MATLAB, where q is a variable of your choice, q contains an ObjectHandle, which makes sure that the socket-connection to QTM is held open throughout your data-transfer.

To start extracting data, simply start QMC again, using q as in input variable as such:

```
>> data = QMC(q)
```

The variable data should contain analog data from 3 channels.

To disconnect, issue the following command:

```
>> QMC(q, 'disconnect')
```

CONFIGURATION FILE

Here is an example of a configuration file, with descriptions.

```
# IP address of the QTM_RT server
<IP:127.0.0.1>    # Local host

# Port used
<PORT:22222>
<UDPPORT:0>      # Port used for UDP data streaming. Set to 0 to use random port.

# Stream data
# 0 : Request (poll) data over standard TCP connection.
# 1 : Stream the data over UDP.
# 2 : Stream the data over TCP.
<STREAM:0>

# Frequency to fetch the data from the QTM RT server. This is only used when
# streaming data (STREAM set to 1), se below.
# <FREQ:FrequencyDivisor:n>    # The camera frequency divided by n.
# <FREQ:Frequency:n>           # Stream data in n Hz
# <FREQ:AllFrames>             # Stream data with camera frequency.
<FREQ:AllFrames>

# Output data size to MATLAB. Amount of objects sent to output for each
# component. Enter a value for each component. Use 0 to disable a component.
<3D:1>                # Max number of markers to receive.
<3D-NoLabels:0>       # Max number of unlabeled markers to receive.
<AnalogSingle:0>      # Max number of analog devices to receive data from.
<ForceSingle:0>       # Max number of forces to receive.
<6DOF:0>              # Max number of 6DOF bodies to receive.
<6DOF-Euler:1>        # Max number of 6DOF-Euler bodies to receive.
<2D:0>                # Max number of 2D points to receive from one camera.
<2D-Lin:0>            # Max number of 2D-Lin points to receive from one camera.
<3D-Residual:0>       # Max number of markers with residual to receive.
<3D-NoLabels-Residual:0> # Max number of unlabeled markers with residual to receive.
<6DOF-Residual:0>     # Max number of 6DOF bodies to receive.
<6DOF-Euler-Residual:0> # Max number of 6DOF-Euler bodies to receive.

# Max number of analog channels to receive data from.
<CHANNEL:4>
```

Changes from version 1.8 to 1.9

Fixed bug that reduced max number of 6DOF bodies to 6. Added UDP streaming port to configuration file.

Changes from version 1.7 to 1.8

- Added function to read frame number and timestamp for last data frame read.
- Changed name of Analog and force data components to AnalogSingle and ForceSingle.
- Fixed Force data bug.

