

# Project GICAR - Application Programming Interface

Generated by Doxygen 1.8.3

Wed Jan 9 2013 13:19:56



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Client Class Reference . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Constructor & Destructor Documentation . . . . .	8
4.1.2.1	Client . . . . .	8
4.1.2.2	~Client . . . . .	8
4.1.3	Member Function Documentation . . . . .	8
4.1.3.1	ConnectToHost . . . . .	8
4.1.3.2	ConnectToHosts . . . . .	8
4.1.3.3	DataReady . . . . .	8
4.1.3.4	GetHttpClient . . . . .	8
4.1.3.5	GetMultiCamClient . . . . .	8
4.1.3.6	IsMultiCam . . . . .	8
4.1.3.7	IsRunning . . . . .	8
4.1.3.8	ReadData . . . . .	9
4.1.3.9	Stop . . . . .	9
4.2	Logger::EndOfLine Class Reference . . . . .	9
4.3	GenericHttpClient Class Reference . . . . .	9
4.3.1	Detailed Description . . . . .	10
4.3.2	Constructor & Destructor Documentation . . . . .	10
4.3.2.1	GenericHttpClient . . . . .	10
4.3.2.2	~GenericHttpClient . . . . .	10
4.3.3	Member Function Documentation . . . . .	10

4.3.3.1	Connect	10
4.3.3.2	ConnectToHost	11
4.3.3.3	DataReady	11
4.3.3.4	GetGUID	11
4.3.3.5	IsRunning	11
4.3.3.6	ParseResponse	11
4.3.3.7	ReadData	11
4.3.3.8	SetGUID	11
4.3.3.9	SetHost	11
4.3.3.10	Stop	11
4.3.3.11	SwapBuffers	12
4.3.4	Member Data Documentation	12
4.3.4.1	logger	12
4.3.4.2	m_guid	12
4.3.4.3	response	12
4.4	HdrDownloader Class Reference	12
4.4.1	Detailed Description	13
4.4.2	Constructor & Destructor Documentation	13
4.4.2.1	HdrDownloader	13
4.4.3	Member Function Documentation	13
4.4.3.1	ParseResponse	13
4.4.3.2	ReadData	13
4.4.3.3	ReadData	13
4.4.3.4	SwapBuffers	13
4.5	HdrWriter Class Reference	13
4.5.1	Detailed Description	14
4.5.2	Member Function Documentation	14
4.5.2.1	ConvertRGBFToRGB888	14
4.5.2.2	FloatToRGBE	14
4.5.2.3	GrayscaleFloatToRGBE	14
4.5.2.4	Read	14
4.5.2.5	RGBEToFloat	15
4.5.2.6	Write	15
4.6	Host Struct Reference	15
4.6.1	Detailed Description	15
4.6.2	Constructor & Destructor Documentation	15
4.6.2.1	Host	15
4.6.3	Member Data Documentation	15
4.6.3.1	m_hostName	15
4.6.3.2	m_port	15

4.7	HttpDataClient Class Reference	16
4.7.1	Detailed Description	16
4.7.2	Constructor & Destructor Documentation	16
4.7.2.1	HttpDataClient	16
4.7.3	Member Function Documentation	16
4.7.3.1	ParseResponse	16
4.7.3.2	ReadData	17
4.7.3.3	SwapBuffers	17
4.8	HttpQ2TreeClient Class Reference	17
4.8.1	Detailed Description	17
4.8.2	Constructor & Destructor Documentation	18
4.8.2.1	HttpQ2TreeClient	18
4.8.3	Member Function Documentation	18
4.8.3.1	ParseResponse	18
4.8.3.2	ReadData	18
4.8.3.3	ReadQ2Tree	18
4.8.3.4	SwapBuffers	18
4.9	IClient Class Reference	18
4.9.1	Detailed Description	19
4.9.2	Constructor & Destructor Documentation	19
4.9.2.1	~IClient	19
4.9.3	Member Function Documentation	19
4.9.3.1	DataReady	19
4.9.3.2	IsRunning	19
4.9.3.3	ReadData	19
4.9.3.4	Stop	19
4.10	Logger Class Reference	20
4.10.1	Detailed Description	20
4.10.2	Member Enumeration Documentation	20
4.10.2.1	anonymous enum	20
4.10.3	Constructor & Destructor Documentation	21
4.10.3.1	Logger	21
4.10.4	Member Function Documentation	21
4.10.4.1	GetDebugLevel	21
4.10.4.2	operator()	21
4.10.4.3	operator<<	21
4.10.4.4	operator<<	21
4.10.5	Member Data Documentation	21
4.10.5.1	endl	21
4.11	MultiCamClient Class Reference	21

4.11.1	Detailed Description	22
4.11.2	Member Typedef Documentation	22
4.11.2.1	SamplesMap	22
4.11.3	Member Enumeration Documentation	22
4.11.3.1	anonymous enum	23
4.11.4	Constructor & Destructor Documentation	23
4.11.4.1	MultiCamClient	23
4.11.4.2	~MultiCamClient	23
4.11.5	Member Function Documentation	23
4.11.5.1	Connect	23
4.11.5.2	ConnectToHosts	23
4.11.5.3	DataReady	23
4.11.5.4	IsRunning	23
4.11.5.5	MergeDeviceData	23
4.11.5.6	MergePDF	24
4.11.5.7	MergeQ2Tree	24
4.11.5.8	NormalizePDF	24
4.11.5.9	NormalizeQ2Tree	24
4.11.5.10	ReadData	24
4.11.5.11	SetNumberOfSamples	24
4.11.5.12	Stop	24
4.11.5.13	SwapBuffers	24
4.12	Q2TreeData Class Reference	24
4.12.1	Detailed Description	26
4.12.2	Constructor & Destructor Documentation	26
4.12.2.1	Q2TreeData	26
4.12.3	Member Function Documentation	26
4.12.3.1	BuildTree	26
4.12.3.2	CutOff	26
4.12.3.3	DebugPrint	26
4.12.3.4	Deserialize	26
4.12.3.5	GetLastUpdate	26
4.12.3.6	GetLeafCount	26
4.12.3.7	GetLength	26
4.12.3.8	Merge	26
4.12.3.9	Normalize	26
4.12.3.10	PlaceSamples	26
4.12.3.11	ReadSerializedData	27
4.12.3.12	RecursiveMerge	27
4.12.3.13	RecursiveNormalize	27

4.12.3.14 Reduce . . . . .	27
4.12.3.15 SortSubtree . . . . .	27
4.13 Q2TreeData::Quad Struct Reference . . . . .	27
4.13.1 Detailed Description . . . . .	28
4.13.2 Member Function Documentation . . . . .	28
4.13.2.1 ComparePosition . . . . .	28
4.13.2.2 EvaluateImportance . . . . .	28
4.13.2.3 IsLeaf . . . . .	28
4.13.3 Member Data Documentation . . . . .	28
4.13.3.1 children . . . . .	28
4.13.3.2 face . . . . .	28
4.13.3.3 id . . . . .	28
4.13.3.4 importance . . . . .	28
4.13.3.5 luminance . . . . .	29
4.13.3.6 parent . . . . .	29
4.13.3.7 solidAngle . . . . .	29
4.13.3.8 x . . . . .	29
4.13.3.9 y . . . . .	29
4.14 Q2TreeData::QuadIdComparer Class Reference . . . . .	29
4.14.1 Detailed Description . . . . .	29
4.14.2 Constructor & Destructor Documentation . . . . .	29
4.14.2.1 QuadIdComparer . . . . .	29
4.14.3 Member Function Documentation . . . . .	29
4.14.3.1 operator() . . . . .	29
4.15 Q2TreeData::QuadImportanceComparer Class Reference . . . . .	30
4.15.1 Detailed Description . . . . .	30
4.15.2 Constructor & Destructor Documentation . . . . .	30
4.15.2.1 QuadImportanceComparer . . . . .	30
4.15.3 Member Function Documentation . . . . .	30
4.15.3.1 operator() . . . . .	30
4.16 Q2TreeData::QuadPositionComparer Class Reference . . . . .	30
4.16.1 Detailed Description . . . . .	30
4.16.2 Constructor & Destructor Documentation . . . . .	30
4.16.2.1 QuadPositionComparer . . . . .	30
4.16.3 Member Function Documentation . . . . .	30
4.16.3.1 operator() . . . . .	30
4.17 MultiCamClient::SampleComparer Struct Reference . . . . .	31
4.17.1 Detailed Description . . . . .	31
4.17.2 Member Function Documentation . . . . .	31
4.17.2.1 operator() . . . . .	31

<b>5</b>	<b>File Documentation</b>	<b>33</b>
5.1	C:/DATA/Repo/MasterThesis/src/Client/Client.cpp File Reference . . . . .	33
5.2	C:/DATA/Repo/MasterThesis/src/Client/Client.h File Reference . . . . .	33
5.3	C:/DATA/Repo/MasterThesis/src/Client/Host.h File Reference . . . . .	33
5.4	C:/DATA/Repo/MasterThesis/src/Client/IClient.h File Reference . . . . .	33
5.5	C:/DATA/Repo/MasterThesis/src/HttpClient/GenericHttpClient.cpp File Reference . . . . .	34
5.6	C:/DATA/Repo/MasterThesis/src/HttpClient/GenericHttpClient.h File Reference . . . . .	34
5.7	C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.cpp File Reference . . . . .	34
5.8	C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.h File Reference . . . . .	34
5.9	C:/DATA/Repo/MasterThesis/src/HttpClient/HttpDataClient.cpp File Reference . . . . .	35
5.10	C:/DATA/Repo/MasterThesis/src/HttpClient/HttpDataClient.h File Reference . . . . .	35
5.11	C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2TreeClient.cpp File Reference . . . . .	35
5.12	C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2TreeClient.h File Reference . . . . .	35
5.13	C:/DATA/Repo/MasterThesis/src/HttpClient/main.cpp File Reference . . . . .	36
5.13.1	Function Documentation . . . . .	36
5.13.1.1	main . . . . .	36
5.14	C:/DATA/Repo/MasterThesis/src/MultiCamClient/main.cpp File Reference . . . . .	36
5.14.1	Function Documentation . . . . .	36
5.14.1.1	main . . . . .	36
5.15	C:/DATA/Repo/MasterThesis/src/MultiCamClient/MultiCamClient.cpp File Reference . . . . .	36
5.16	C:/DATA/Repo/MasterThesis/src/MultiCamClient/MultiCamClient.h File Reference . . . . .	37
5.17	C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.cpp File Reference . . . . .	37
5.18	C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.h File Reference . . . . .	37
5.19	C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.cpp File Reference . . . . .	38
5.20	C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.h File Reference . . . . .	38
5.21	C:/DATA/Repo/MasterThesis/src/shared/Logger.cpp File Reference . . . . .	38
5.22	C:/DATA/Repo/MasterThesis/src/shared/Logger.h File Reference . . . . .	38
<b>Index</b>		<b>38</b>



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Logger::EndOfLine . . . . .	9
HdrWriter . . . . .	13
Host . . . . .	15
IClient . . . . .	18
Client . . . . .	7
GenericHttpClient . . . . .	9
HdrDownloader . . . . .	12
HttpDataClient . . . . .	16
HttpQ2TreeClient . . . . .	17
MultiCamClient . . . . .	21
Logger . . . . .	20
Q2TreeData . . . . .	24
QObject	
Client . . . . .	7
GenericHttpClient . . . . .	9
MultiCamClient . . . . .	21
Q2TreeData::Quad . . . . .	27
Q2TreeData::QuadIdComparer . . . . .	29
Q2TreeData::QuadImportanceComparer . . . . .	30
Q2TreeData::QuadPositionComparer . . . . .	30
MultiCamClient::SampleComparer . . . . .	31



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Client</a>	A wrapper that provides access to both single and multi camera clients . . . . .	7
<a href="#">Logger::EndOfLine</a>	. . . . .	9
<a href="#">GenericHttpClient</a>	This abstract class implements routines for initializing and maintaining the connection to the server over HTTP, as well as error handling . . . . .	9
<a href="#">HdrDownloader</a>	This class implements parsing of .hdr images from a network stream . . . . .	12
<a href="#">HdrWriter</a>	This class implements reading and writing of HDR images in .hdr format . . . . .	13
<a href="#">Host</a>	This structure represents the address and port of a TCP host . . . . .	15
<a href="#">HttpDataClient</a>	This class implements parsing of sampling data (set of directional light sources) from a network stream . . . . .	16
<a href="#">HttpQ2TreeClient</a>	This class implements parsing of Q2-Trees from a network stream . . . . .	17
<a href="#">IClient</a>	This abstract class represents an interface of all the API client classes . . . . .	18
<a href="#">Logger</a>	A logger class . . . . .	20
<a href="#">MultiCamClient</a>	This class allows for simultaneous connection to multiple servers and implements merging of received data . . . . .	21
<a href="#">Q2TreeData</a>	This class implments merging of Q2-Trees . . . . .	24
<a href="#">Q2TreeData::Quad</a>	Represents a quad in the HEALPix mapping . . . . .	27
<a href="#">Q2TreeData::QuadIdComparer</a>	Comparison class for sorting . . . . .	29
<a href="#">Q2TreeData::QuadImportanceComparer</a>	Comparison class for sorting . . . . .	30
<a href="#">Q2TreeData::QuadPositionComparer</a>	Comparison class for sorting . . . . .	30
<a href="#">MultiCamClient::SampleComparer</a>	A comparer function object for ordering of samples . . . . .	31



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

C:/DATA/Repo/MasterThesis/src/Client/Client.cpp	33
C:/DATA/Repo/MasterThesis/src/Client/Client.h	33
C:/DATA/Repo/MasterThesis/src/Client/Host.h	33
C:/DATA/Repo/MasterThesis/src/Client/IClient.h	33
C:/DATA/Repo/MasterThesis/src/HttpClient/GenericHttpClient.cpp	34
C:/DATA/Repo/MasterThesis/src/HttpClient/GenericHttpClient.h	34
C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.cpp	34
C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.h	34
C:/DATA/Repo/MasterThesis/src/HttpClient/HttpDataClient.cpp	35
C:/DATA/Repo/MasterThesis/src/HttpClient/HttpDataClient.h	35
C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2TreeClient.cpp	35
C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2TreeClient.h	35
C:/DATA/Repo/MasterThesis/src/HttpClient/main.cpp	36
C:/DATA/Repo/MasterThesis/src/MultiCamClient/main.cpp	36
C:/DATA/Repo/MasterThesis/src/MultiCamClient/MultiCamClient.cpp	36
C:/DATA/Repo/MasterThesis/src/MultiCamClient/MultiCamClient.h	37
C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.cpp	37
C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.h	37
C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.cpp	38
C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.h	38
C:/DATA/Repo/MasterThesis/src/shared/Logger.cpp	38
C:/DATA/Repo/MasterThesis/src/shared/Logger.h	38



## Chapter 4

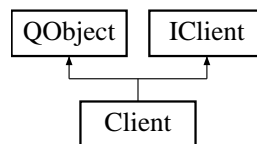
# Class Documentation

### 4.1 Client Class Reference

A wrapper that provides access to both single and multi camera clients.

```
#include <Client.h>
```

Inheritance diagram for Client:



### Signals

- void `DataReady` ()

*This signal is emitted when the content of the front buffer gets updated.*

### Public Member Functions

- `Client` (`Logger` &logger, bool multiCam, int timeout=3, `QObject` \*parent=NULL)
- virtual `~Client` ()
- void `ConnectToHost` (QString hostName, quint16 port)  
*Connects to the server and sends a request for the data.*
- void `ConnectToHosts` (`Host` \*hosts, int count, int timeout, int mode, int numSamples)  
*Establishes connection with multiple remote devices.*
- void `Stop` ()  
*Stops the communication with the remote devices.*
- bool `IsRunning` ()  
*Returns true if the client is communicating with remote devices.*
- bool `IsMultiCam` ()  
*Returns true if this instance supports simultaneous connection to multiple remote hosts.*
- void `ReadData` (`SamplingData` &data)  
*Reads the sampling data.*

## Protected Member Functions

- [HttpDataClient](#) \* [GetHttpClient](#) ()
- [MultiCamClient](#) \* [GetMultiCamClient](#) ()

### 4.1.1 Detailed Description

A wrapper that provides access to both single and multi camera clients.

### 4.1.2 Constructor & Destructor Documentation

4.1.2.1 `Client::Client ( Logger & logger, bool multiCam, int timeout = 3, QObject * parent = NULL )`

4.1.2.2 `Client::~~Client ( )` [[virtual](#)]

### 4.1.3 Member Function Documentation

4.1.3.1 `void Client::ConnectToHost ( QString hostName, quint16 port )`

Connects to the server and sends a request for the data.

When the transfer is completed and the client either receives the data or encounters an error, the request is automatically repeated.

4.1.3.2 `void Client::ConnectToHosts ( Host * hosts, int count, int timeout, int mode, int numSamples = -1 )`

Establishes connection with multiple remote devices.

The client continues to receive updated data from all the remote devices until stopped by a call to the [Stop\(\)](#) method.

#### Parameters

<i>timeout</i>	Resend request if the device does not respond withing the specified timeout interval.
----------------	---

#### Remarks

This method is available only if the class was created with the multiCam flag set to true.

4.1.3.3 `void Client::DataReady ( )` [[signal](#)]

This signal is emitted when the content of the front buffer gets updated.

4.1.3.4 `HttpDataClient * Client::GetHttpClient ( )` [[protected](#)]

4.1.3.5 `MultiCamClient * Client::GetMultiCamClient ( )` [[protected](#)]

4.1.3.6 `bool Client::IsMultiCam ( )` [[inline](#)]

Returns true if this instance supports simultaneous connection to multiple remote hosts.

4.1.3.7 `bool Client::IsRunning ( )` [[virtual](#)]

Returns true if the client is communicating with remote devices.

Implements [IClient](#).



4.1.3.8 void Client::ReadData ( SamplingData & data ) [virtual]

Reads the sampling data.

Implements [IClient](#).

4.1.3.9 void Client::Stop ( ) [virtual]

Stops the communication with the remote devices.

Implements [IClient](#).

The documentation for this class was generated from the following files:

- C:/DATA/Repo/MasterThesis/src/Client/[Client.h](#)
- C:/DATA/Repo/MasterThesis/src/Client/[Client.cpp](#)

## 4.2 Logger::EndOfLine Class Reference

```
#include <Logger.h>
```

The documentation for this class was generated from the following file:

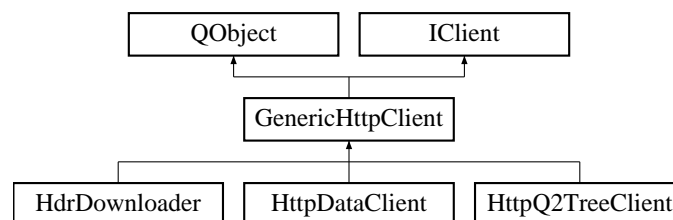
- C:/DATA/Repo/MasterThesis/src/shared/[Logger.h](#)

## 4.3 GenericHttpClient Class Reference

This abstract class implements routines for initializing and maintaining the connection to the server over HTTP, as well as error handling.

```
#include <GenericHttpClient.h>
```

Inheritance diagram for GenericHttpClient:



### Signals

- void [DataReady](#) ()  
*This signal is emitted when the content of the front buffer gets updated.*

### Public Member Functions

- [GenericHttpClient](#) ([Logger](#) &[logger](#), const char \*httpRequest, int resendIntervalMsec, QObject \*parent=0)
- virtual [~GenericHttpClient](#) ()
- void [ConnectToHost](#) (QString hostName, quint16 port)  
*Connects to the server and sends a request for the data.*

- void [Stop](#) ()  
*Stops the communication with the server. (The client exits the infinite loop)*
- bool [IsRunning](#) ()  
*Returns true if the client is communicating with a server.*
- void [SetHost](#) (QString hostName, quint16 port)  
*Sets the address and port of the remote server. Thread safe.*
- virtual void [ReadData](#) (SamplingData &data)=0  
*Reads the data from the front buffer.*
- void [SetGUID](#) (int guid)  
*Sets the Global Unique ID of this instance.*
- int [GetGUID](#) ()  
*Gets the Global Unique ID of this instance.*

### Protected Member Functions

- void [Connect](#) ()  
*Connects to the server and sends a request for the data.*
- virtual bool [ParseResponse](#) (QByteArray response)=0  
*This method is invoked when all the data is successfully transfer over the network and it parses the response and fills the back buffer with data.*
- virtual void [SwapBuffers](#) ()=0  
*Copies the contents of the back buffer into the front buffer.*

### Protected Attributes

- [Logger](#) & [logger](#)
- QByteArray [response](#)
- int [m\\_guid](#)

#### 4.3.1 Detailed Description

This abstract class implements routines for initializing and maintaining the connection to the server over HTTP, as well as error handling.

#### 4.3.2 Constructor & Destructor Documentation

**4.3.2.1** `GenericHttpClient::GenericHttpClient ( Logger & logger, const char * httpRequest, int resendIntervalMsec, QObject * parent = 0 )`

**4.3.2.2** `GenericHttpClient::~GenericHttpClient ( )` [virtual]

#### 4.3.3 Member Function Documentation

**4.3.3.1** `void GenericHttpClient::Connect ( )` [protected]

Connects to the server and sends a request for the data.

When the transfer is completed and the client either receives the data or encounters an error, the request is automatically repeated. So calling this method starts an infinite loop of requests to the server, which are in turn parsed and the results moved into the front data buffer to be used by other threads.

#### 4.3.3.2 void GenericHttpClient::ConnectToHost ( QString *hostName*, quint16 *port* )

Connects to the server and sends a request for the data.

When the transfer is completed and the client either receives the data or encounters an error, the request is automatically repeated. So calling this method starts an infinite loop of requests to the server, which are in turn parsed and the results moved into the front data buffer to be used by other threads.

This method is equivalent to calling [SetHost\(\)](#) followed by [Connect\(\)](#).

#### 4.3.3.3 void GenericHttpClient::DataReady ( ) [signal]

This signal is emitted when the content of the front buffer gets updated.

#### 4.3.3.4 int GenericHttpClient::GetGUID ( ) [inline]

Gets the Global Unique ID of this instance.

#### 4.3.3.5 bool GenericHttpClient::IsRunning ( ) [inline],[virtual]

Returns true if the client is communicating with a server.

Implements [IClient](#).

#### 4.3.3.6 virtual bool GenericHttpClient::ParseResponse ( QByteArray *response* ) [protected],[pure virtual]

This method is invoked when all the data is successfully transfer over the network and it parses the response and fills the back buffer with data.

Implemented in [HdrDownloader](#), [HttpQ2TreeClient](#), and [HttpDataClient](#).

#### 4.3.3.7 virtual void GenericHttpClient::ReadData ( SamplingData & *data* ) [pure virtual]

Reads the data from the front buffer.

Implements [IClient](#).

Implemented in [HdrDownloader](#), [HttpQ2TreeClient](#), and [HttpDataClient](#).

#### 4.3.3.8 void GenericHttpClient::SetGUID ( int *guid* ) [inline]

Sets the Global Unique ID of this instance.

#### 4.3.3.9 void GenericHttpClient::SetHost ( QString *hostName*, quint16 *port* )

Sets the address and port of the remote server. Thread safe.

#### 4.3.3.10 void GenericHttpClient::Stop ( ) [inline],[virtual]

Stops the communication with the server. (The client exits the infinite loop)

Implements [IClient](#).

#### 4.3.3.11 virtual void GenericHttpClient::SwapBuffers ( ) [protected],[pure virtual]

Copies the contents of the back buffer into the front buffer.

Implemented in [HttpQ2TreeClient](#), [HdrDownloader](#), and [HttpDataClient](#).

### 4.3.4 Member Data Documentation

#### 4.3.4.1 Logger& GenericHttpClient::logger [protected]

#### 4.3.4.2 int GenericHttpClient::m\_guid [protected]

#### 4.3.4.3 QByteArray GenericHttpClient::response [protected]

The documentation for this class was generated from the following files:

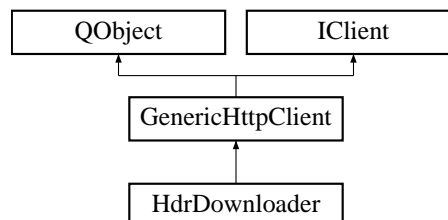
- C:/DATA/Repo/MasterThesis/src/HttpClient/[GenericHttpClient.h](#)
- C:/DATA/Repo/MasterThesis/src/HttpClient/[GenericHttpClient.cpp](#)

## 4.4 HdrDownloader Class Reference

This class implements parsing of .hdr images from a network stream.

```
#include <HdrDownloader.h>
```

Inheritance diagram for HdrDownloader:



### Public Member Functions

- [HdrDownloader](#) ([Logger](#) &[logger](#), int resendIntervalMsec, QObject \*parent=0)
- virtual const QByteArray [ReadData](#) () const  
*Reads the data from the front buffer.*
- virtual void [ReadData](#) (SamplingData &data)  
*Reads the data from the front buffer.*

### Protected Member Functions

- virtual bool [ParseResponse](#) (QByteArray [response](#))  
*This method is invoked when all the data is successfully transfer over the network; it parses the response and fills the back buffer with data.*
- virtual void [SwapBuffers](#) ()  
*Copies the contents of the back buffer into the front buffer and signals DataReady.*

## Additional Inherited Members

### 4.4.1 Detailed Description

This class implements parsing of .hdr images from a network stream.

### 4.4.2 Constructor & Destructor Documentation

4.4.2.1 `HdrDownloader::HdrDownloader ( Logger & logger, int resendIntervalMsec, QObject * parent = 0 )`

### 4.4.3 Member Function Documentation

4.4.3.1 `bool HdrDownloader::ParseResponse ( QByteArray response )` [protected], [virtual]

This method is invoked when all the data is successfully transfer over the network; it parses the response and fills the back buffer with data.

Implements [GenericHttpClient](#).

4.4.3.2 `const QByteArray HdrDownloader::ReadData ( ) const` [virtual]

Reads the data from the front buffer.

4.4.3.3 `virtual void HdrDownloader::ReadData ( SamplingData & data )` [inline], [virtual]

Reads the data from the front buffer.

Implements [GenericHttpClient](#).

4.4.3.4 `void HdrDownloader::SwapBuffers ( )` [protected], [virtual]

Copies the contents of the back buffer into the front buffer and signals DataReady.

Implements [GenericHttpClient](#).

The documentation for this class was generated from the following files:

- C:/DATA/Repo/MasterThesis/src/HttpClient/[HdrDownloader.h](#)
- C:/DATA/Repo/MasterThesis/src/HttpClient/[HdrDownloader.cpp](#)

## 4.5 HdrWriter Class Reference

This class implements reading and writing of HDR images in .hdr format.

```
#include <HdrWriter.h>
```

### Static Public Member Functions

- static bool [Write](#) (std::ostream &output, unsigned width, unsigned height, const float \*rgbf, bool is-Grayscale=false)  
*Converts floating point image data into .hdr file format.*
- static bool [Read](#) (const char \*header, const char \*resolution, const char \*data, int dataLength, unsigned &width, unsigned &height, float \*\*rgbf)  
*Imports a .hdr file.*

- static void [ConvertRGBFToRGB888](#) (unsigned width, unsigned height, float \*rgbf, unsigned char \*rgb888)  
*Converts floating point RGBF to 24-bit RGB.*

### Static Protected Member Functions

- static void [FloatToRGBE](#) (unsigned char rgbe[4], const float \*rgb)  
*Conversion from float pixes to RGBE.*
- static void [GrayscaleFloatToRGBE](#) (unsigned char rgbe[4], const float \*grayscale)  
*Conversion from float pixes to RGBE.*
- static void [RGBEToFloat](#) (unsigned char rgbe[4], float \*rgb)  
*Conversion from RGBE to float pixels.*

#### 4.5.1 Detailed Description

This class implements reading and writing of HDR images in .hdr format.

#### 4.5.2 Member Function Documentation

**4.5.2.1** void HdrWriter::ConvertRGBFToRGB888 ( unsigned width, unsigned height, float \* rgbf, unsigned char \* rgb888 )  
[static]

Converts floating point RGBF to 24-bit RGB.

##### Remarks

Floating point values are tone-mapped to the range 0-255.

**4.5.2.2** static void HdrWriter::FloatToRGBE ( unsigned char rgbe[4], const float \* rgb ) [inline], [static],  
[protected]

Conversion from float pixes to RGBE.

**4.5.2.3** static void HdrWriter::GrayscaleFloatToRGBE ( unsigned char rgbe[4], const float \* grayscale ) [inline],  
[static], [protected]

Conversion from float pixes to RGBE.

**4.5.2.4** bool HdrWriter::Read ( const char \* header, const char \* resolution, const char \* data, int dataLenght, unsigned & width, unsigned & height, float \*\* rgbf ) [static]

Imports a .hdr file.

##### Parameters

<i>rgbf</i>	Image data (3 floats per pixel - RGB) - the space is dynamically allocated by this method
<i>width</i>	Image width
<i>height</i>	Image height
<i>header</i>	.hdr file header
<i>resolution</i>	.hdr file resolution section
<i>data</i>	.hdr file data section

4.5.2.5 `static void HdrWriter::RGBToFloat ( unsigned char rgbe[4], float * rgb ) [inline], [static], [protected]`

Conversion from RGBE to float pixels.

4.5.2.6 `bool HdrWriter::Write ( std::ostream & output, unsigned width, unsigned height, const float * rgbf, bool isGrayscale = false ) [static]`

Converts floating point image data into .hdr file format.

#### Parameters

<i>rgbf</i>	Image data (either RGB 3 floats per pixel or grayscale 1 float per pixel)
<i>width</i>	Image width
<i>height</i>	Image height
<i>isGrayscale</i>	true if <i>rgbf</i> points to grayscale image data (1 float per pixel), false for RGB (3 floats per pixel)
<i>output</i>	Reference to an output stream where the .hdr data are written

The documentation for this class was generated from the following files:

- C:/DATA/Repo/MasterThesis/src/shared/[HdrWriter.h](#)
- C:/DATA/Repo/MasterThesis/src/shared/[HdrWriter.cpp](#)

## 4.6 Host Struct Reference

This structure represents the address and port of a TCP host.

```
#include <Host.h>
```

### Public Member Functions

- [Host](#) (QString *hostName*, quint16 *port*)

### Public Attributes

- QString [m\\_hostName](#)
- quint16 [m\\_port](#)

#### 4.6.1 Detailed Description

This structure represents the address and port of a TCP host.

#### 4.6.2 Constructor & Destructor Documentation

4.6.2.1 `Host::Host ( QString hostName, quint16 port ) [inline]`

#### 4.6.3 Member Data Documentation

4.6.3.1 `QString Host::m_hostName`

4.6.3.2 `quint16 Host::m_port`

The documentation for this struct was generated from the following file:

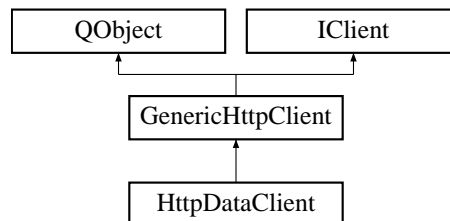
- [C:/DATA/Repo/MasterThesis/src/Client/Host.h](#)

## 4.7 HttpDataClient Class Reference

This class implements parsing of sampling data (set of directional light sources) from a network stream.

```
#include <HttpDataClient.h>
```

Inheritance diagram for HttpDataClient:



### Public Member Functions

- [HttpDataClient](#) ([Logger](#) &[logger](#), int [resendIntervalMsec](#), [QObject](#) \*[parent](#)=0)
- virtual void [ReadData](#) ([SamplingData](#) &[data](#))

*Reads the data from the front buffer.*

### Protected Member Functions

- virtual bool [ParseResponse](#) ([QByteArray](#) [response](#))

*This method is invoked when all the data is successfully transfer over the network; it parses the response and fills the back buffer with data.*

- virtual void [SwapBuffers](#) ()

*Copies the contents of the back buffer into the front buffer and signals DataReady.*

### Additional Inherited Members

#### 4.7.1 Detailed Description

This class implements parsing of sampling data (set of directional light sources) from a network stream.

#### 4.7.2 Constructor & Destructor Documentation

4.7.2.1 [HttpDataClient::HttpDataClient](#) ( [Logger](#) & [logger](#), int [resendIntervalMsec](#), [QObject](#) \* [parent](#) = 0 )

#### 4.7.3 Member Function Documentation

4.7.3.1 [bool HttpDataClient::ParseResponse](#) ( [QByteArray](#) [response](#) ) [protected], [virtual]

This method is invoked when all the data is successfully transfer over the network; it parses the response and fills the back buffer with data.

Implements [GenericHttpClient](#).



4.7.3.2 void HttpDataClient::ReadData ( SamplingData & data ) [virtual]

Reads the data from the front buffer.

Implements [GenericHttpClient](#).

4.7.3.3 void HttpDataClient::SwapBuffers ( ) [protected],[virtual]

Copies the contents of the back buffer into the front buffer and signals DataReady.

Implements [GenericHttpClient](#).

The documentation for this class was generated from the following files:

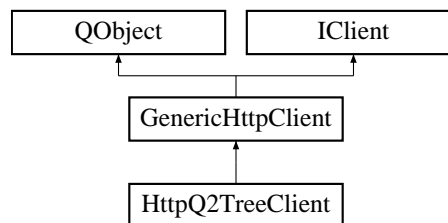
- C:/DATA/Repo/MasterThesis/src/HttpClient/[HttpDataClient.h](#)
- C:/DATA/Repo/MasterThesis/src/HttpClient/[HttpDataClient.cpp](#)

## 4.8 HttpQ2TreeClient Class Reference

This class implements parsing of Q2-Trees from a network stream.

```
#include <HttpQ2TreeClient.h>
```

Inheritance diagram for HttpQ2TreeClient:



### Public Member Functions

- [HttpQ2TreeClient](#) ([Logger](#) &[logger](#), int resendIntervalMsec, QObject \*parent=0)
- virtual void [ReadData](#) (SamplingData &data)  
*Reads the data from the front buffer.*
- [Q2TreeData](#) & [ReadQ2Tree](#) ()

### Protected Member Functions

- virtual bool [ParseResponse](#) (QByteArray response)  
*This method is invoked when all the data is successfully transfer over the network; it parses the response and fills the back buffer with data.*
- virtual void [SwapBuffers](#) ()  
*Copies the contents of the back buffer into the front buffer.*

### Additional Inherited Members

#### 4.8.1 Detailed Description

This class implements parsing of Q2-Trees from a network stream.

## 4.8.2 Constructor & Destructor Documentation

4.8.2.1 `HttpQ2TreeClient::HttpQ2TreeClient ( Logger & logger, int resendIntervalMsec, QObject * parent = 0 )`

## 4.8.3 Member Function Documentation

4.8.3.1 `bool HttpQ2TreeClient::ParseResponse ( QByteArray response )` [protected],[virtual]

This method is invoked when all the data is successfully transfer over the network; it parses the response and fills the back buffer with data.

Implements [GenericHttpClient](#).

4.8.3.2 `virtual void HttpQ2TreeClient::ReadData ( SamplingData & data )` [inline],[virtual]

Reads the data from the front buffer.

Implements [GenericHttpClient](#).

4.8.3.3 `Q2TreeData& HttpQ2TreeClient::ReadQ2Tree ( )` [inline]

4.8.3.4 `virtual void HttpQ2TreeClient::SwapBuffers ( )` [inline],[protected],[virtual]

Copies the contents of the back buffer into the front buffer.

Implements [GenericHttpClient](#).

The documentation for this class was generated from the following files:

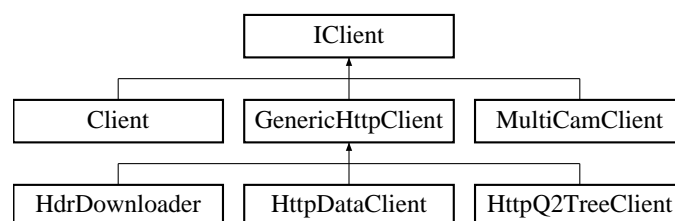
- C:/DATA/Repo/MasterThesis/src/HttpClient/[HttpQ2TreeClient.h](#)
- C:/DATA/Repo/MasterThesis/src/HttpClient/[HttpQ2TreeClient.cpp](#)

## 4.9 IClient Class Reference

This abstract class represents an interface of all the API client classes.

```
#include <IClient.h>
```

Inheritance diagram for IClient:



## Signals

- virtual void [DataReady](#) ()=0

*This signal is emitted when the content of the front buffer gets updated.*

## Public Member Functions

- virtual `~IClient()`  
*Empty virtual destructor. Needs to be defined explicitly to support multiple inheritance.*
- virtual bool `IsRunning()`=0  
*Returns true if the client is communicating with a server.*
- virtual void `Stop()`=0  
*Stops the communication with the remote devices.*
- virtual void `ReadData(SamplingData &data)`=0  
*Reads the data from the front buffer.*

### 4.9.1 Detailed Description

This abstract class represents an interface of all the API client classes.

#### Author

Tomas Nikodym

### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 virtual IClient::~~IClient( ) [inline],[virtual]

Empty virtual destructor. Needs to be defined explicitly to support multiple inheritance.

### 4.9.3 Member Function Documentation

#### 4.9.3.1 virtual void IClient::DataReady( ) [pure virtual],[signal]

This signal is emitted when the content of the front buffer gets updated.

#### 4.9.3.2 virtual bool IClient::IsRunning( ) [pure virtual]

Returns true if the client is communicating with a server.

Implemented in [MultiCamClient](#), [Client](#), and [GenericHttpClient](#).

#### 4.9.3.3 virtual void IClient::ReadData( SamplingData & data ) [pure virtual]

Reads the data from the front buffer.

Implemented in [MultiCamClient](#), [Client](#), [GenericHttpClient](#), [HdrDownloader](#), [HttpQ2TreeClient](#), and [HttpDataClient](#).

#### 4.9.3.4 virtual void IClient::Stop( ) [pure virtual]

Stops the communication with the remote devices.

Implemented in [MultiCamClient](#), [Client](#), and [GenericHttpClient](#).

The documentation for this class was generated from the following file:

- C:/DATA/Repo/MasterThesis/src/Client/[IClient.h](#)

## 4.10 Logger Class Reference

A logger class.

```
#include <Logger.h>
```

### Classes

- class [EndOfLine](#)

### Public Types

- enum {  
[ERROR](#), [WARNING](#), [INFO](#), [DEBUG](#),  
[DEBUG2](#), [DEBUG3](#) }

*Specifies the urgency of a message. Only messages that are more urgent than the value specified in constructor are pass to the output stream.*

### Public Member Functions

- [Logger](#) (ostream &os, ostream &err, int debugLevel)
- [Logger](#) & [operator\(\)](#) (int msgLevel)
- template<class T >  
[Logger](#) & [operator<<](#) (T msg)
- [Logger](#) & [operator<<](#) ([EndOfLine](#))
- int [GetDebugLevel](#) ()

### Static Public Attributes

- static [EndOfLine endl](#)

#### 4.10.1 Detailed Description

A logger class.

Example: `logger(Logger::Warning) << "Failed to open file: " << filename << Logger::endl;`

#### 4.10.2 Member Enumeration Documentation

##### 4.10.2.1 anonymous enum

Specifies the urgency of a message. Only messages that are more urgent than the value specified in constructor are pass to the output stream.

##### Enumerator

**ERROR** Serious error, execution cannot continue.

**WARNING** Local error, does not impair global functioning of the application.

**INFO** Information, used to report that a major task was executed successfully.

**DEBUG** Information of minor importance, but should not be reported more than once in a while.

**DEBUG2** Information of minor importance, can be reported several times per second (for debugging purposes only)

**DEBUG3** Information of minor importance, can be reported several times per second (for debugging purposes only)

### 4.10.3 Constructor & Destructor Documentation

4.10.3.1 `Logger::Logger ( ostream & os, ostream & err, int debugLevel )` `[inline]`

### 4.10.4 Member Function Documentation

4.10.4.1 `int Logger::GetDebugLevel ( )` `[inline]`

4.10.4.2 `Logger& Logger::operator() ( int msgLevel )` `[inline]`

4.10.4.3 `template<class T > Logger& Logger::operator<< ( T msg )` `[inline]`

4.10.4.4 `Logger& Logger::operator<< ( EndOfLine )` `[inline]`

### 4.10.5 Member Data Documentation

4.10.5.1 `Logger::EndOfLine Logger::endl` `[static]`

The documentation for this class was generated from the following files:

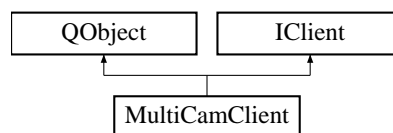
- [C:/DATA/Repo/MasterThesis/src/shared/Logger.h](#)
- [C:/DATA/Repo/MasterThesis/src/shared/Logger.cpp](#)

## 4.11 MultiCamClient Class Reference

This class allows for simultaneous connection to multiple servers and implements merging of received data.

```
#include <MultiCamClient.h>
```

Inheritance diagram for MultiCamClient:



### Classes

- struct [SampleComparer](#)  
*A comparer function object for ordering of samples.*

### Public Types

- enum { [StandardMergeMode](#), [AdvancedQ2TreeMergeMode](#) }

### Signals

- void [DataReady](#) ()  
*This signal is emitted when the content of the front buffer gets updated.*

## Public Member Functions

- [MultiCamClient](#) ([Logger](#) &logger, int dataValidPeriod=5000, [QObject](#) \*parent=0)
- [~MultiCamClient](#) ()
- void [ConnectToHosts](#) ([Host](#) \*hosts, int count, int timeout, int mode=[StandardMergeMode](#))  
*Establishes connection with the specified remote devices.*
- void [Stop](#) ()  
*Stops the communication with the remote devices.*
- bool [IsRunning](#) ()  
*Returns true if the client is communicating with remote devices.*
- void [ReadData](#) ([SamplingData](#) &data)  
*Reads the data from the front buffer.*
- void [SetNumberOfSamples](#) (int numSamples)  
*Sets the requested number of samples.*

## Protected Types

- typedef std::map  
  < [SamplingData::Sample](#),  
  [SamplingData::Sample](#),  
  [SampleComparer](#) > [SamplesMap](#)

## Protected Member Functions

- void [Connect](#) ()  
*Establishes connection with the remote devices.*
- void [MergeDeviceData](#) ()  
*Merges data from all connected devices.*
- void [MergePDF](#) ([SamplingData](#) &resultData, const [SamplingData](#) &samplingData)  
*Merge samples (specific to PDF-based sampling algorithms)*
- void [NormalizePDF](#) ([SamplingData](#) &resultData, int numDevicesMerged)  
*Normalize results (specific to PDF-based sampling algorithms)*
- void [MergeQ2Tree](#) ([SamplesMap](#) &samples, const [SamplingData](#) &samplingData)  
*Merge samples (specific to Q2-Tree sampling algorithms)*
- void [NormalizeQ2Tree](#) ([SamplingData](#) &resultData, int numDevicesMerged)  
*Normalize results (specific to Q2-Tree sampling algorithms)*
- void [SwapBuffers](#) ()  
*Copies the contents of the back buffer into the front buffer.*

### 4.11.1 Detailed Description

This class allows for simultaneous connection to multiple servers and implements merging of received data.

### 4.11.2 Member Typedef Documentation

- 4.11.2.1 `typedef std::map<SamplingData::Sample, SamplingData::Sample, SampleComparer>  
MultiCamClient::SamplesMap [protected]`

### 4.11.3 Member Enumeration Documentation

## 4.11.3.1 anonymous enum

Enumerator

**StandardMergeMode** Download sampling data, apply general merging.

**AdvancedQ2TreeMergeMode** Download Q2Tree, apply advanced Q2Tree specific algorithm.

## 4.11.4 Constructor &amp; Destructor Documentation

4.11.4.1 MultiCamClient::MultiCamClient ( **Logger & logger**, int **dataValidPeriod** = 5000, **QObject \* parent** = 0 )

Parameters

<i>dataValidPeriod</i>	The time in milliseconds, for which the samples received from a device are considered valid. When this period expires, the samples from this device are ignored in the Merge process.
------------------------	---

4.11.4.2 MultiCamClient::~~MultiCamClient ( )

## 4.11.5 Member Function Documentation

4.11.5.1 void MultiCamClient::Connect ( ) [protected]

Establishes connection with the remote devices.

The client continues to receive updated data from all the remote devices until stopped by a call to the [Stop\(\)](#) method.

4.11.5.2 void MultiCamClient::ConnectToHosts ( **Host \* hosts**, int **count**, int **timeout**, int **mode** = **StandardMergeMode** )

Establishes connection with the specified remote devices.

The client continues to receive updated data from all the remote devices until stopped by a call to the [Stop\(\)](#) method.

Parameters

<i>timeout</i>	Resend request if the device does not respond withing the specified timeout interval
----------------	--

4.11.5.3 void MultiCamClient::DataReady ( ) [signal]

This signal is emitted when the content of the front buffer gets updated.

4.11.5.4 bool MultiCamClient::IsRunning ( ) [inline],[virtual]

Returns true if the client is communitating with remote devices.

Implements [IClient](#).

4.11.5.5 void MultiCamClient::MergeDeviceData ( ) [protected]

Merges data from all connected devices.

4.11.5.6 `void MultiCamClient::MergePDF ( SamplingData & resultData, const SamplingData & samplingData )`  
`[protected]`

Merge samples (specific to PDF-based sampling algorithms)

4.11.5.7 `void MultiCamClient::MergeQ2Tree ( SamplesMap & samples, const SamplingData & samplingData )`  
`[protected]`

Merge samples (specific to Q2-Tree sampling algorithms)

4.11.5.8 `void MultiCamClient::NormalizePDF ( SamplingData & resultData, int numDevicesMerged )` `[protected]`

Normalize results (specific to PDF-based sampling algorithms)

4.11.5.9 `void MultiCamClient::NormalizeQ2Tree ( SamplingData & resultData, int numDevicesMerged )` `[protected]`

Normalize results (specific to Q2-Tree sampling algorithms)

4.11.5.10 `void MultiCamClient::ReadData ( SamplingData & data )` `[virtual]`

Reads the data from the front buffer.

Implements [IClient](#).

4.11.5.11 `void MultiCamClient::SetNumberOfSamples ( int numSamples )` `[inline]`

Sets the requested number of samples.

4.11.5.12 `void MultiCamClient::Stop ( )` `[inline], [virtual]`

Stops the communication with the remote devices.

Implements [IClient](#).

4.11.5.13 `void MultiCamClient::SwapBuffers ( )` `[protected]`

Copies the contents of the back buffer into the front buffer.

The documentation for this class was generated from the following files:

- C:/DATA/Repo/MasterThesis/src/MultiCamClient/[MultiCamClient.h](#)
- C:/DATA/Repo/MasterThesis/src/MultiCamClient/[MultiCamClient.cpp](#)

## 4.12 Q2TreeData Class Reference

This class implments merging of Q2-Trees.

```
#include <Q2TreeData.h>
```



## Classes

- struct [Quad](#)  
*Represents a quad in the HEALPix mapping.*
- class [QuadIdComparer](#)  
*comparison class for sorting*
- class [QuadImportanceComparer](#)  
*comparison class for sorting*
- class [QuadPositionComparer](#)  
*comparison class for sorting*

## Public Member Functions

- [Q2TreeData](#) ()
- bool [Deserialize](#) (std::istream &in)  
*Reads a list of quads and builds a tree on top of it.*
- void [Reduce](#) (int numSamples)  
*Reduces the number of leaf nodes to 'numSamples'.*
- void [PlaceSamples](#) (SamplingData &samplingData)  
*Places one sample in each leaf node.*
- void [DebugPrint](#) ()  
*For debugging purposes.*
- int [GetLength](#) ()
- clock\_t [GetLastUpdate](#) ()
- int [GetLeafCount](#) ()

## Static Public Member Functions

- static void [Merge](#) ([Q2TreeData](#) \*result, [Q2TreeData](#) \*\*input, bool \*ignore, int inputCount)  
*Merges input Q2Trees into a single normalized Q2Tree.*

## Protected Member Functions

- bool [ReadSerializedData](#) (std::istream &in)  
*Deserialization subroutine.*
- bool [BuildTree](#) ()  
*Deserialization subroutine.*
- void [SortSubtree](#) (int root)  
*Deserialization subroutine.*
- void [Normalize](#) (int nTreesMerged)  
*Normalizes the Q2Tree, so that the luminance of each node equals the sum of luminances of its four children. The luminance of the base quads is preserved.*
- void [RecursiveNormalize](#) (int index)  
*Normalize subroutine.*
- int [CutOff](#) (int index)  
*Removes the subtree at the specified index, leaving only the root.*

## Static Protected Member Functions

- static void [RecursiveMerge](#) ([Q2TreeData](#) \*result, int resultIndex, int parentIndex, int &childIndex, [Q2TreeData](#) \*\*input, int \*inputIndices, int inputCount)  
*Merge subroutine.*

### 4.12.1 Detailed Description

This class implments merging of Q2-Trees.

### 4.12.2 Constructor & Destructor Documentation

#### 4.12.2.1 Q2TreeData::Q2TreeData ( )

### 4.12.3 Member Function Documentation

#### 4.12.3.1 bool Q2TreeData::BuildTree ( ) [protected]

Deserialization subroutine.

#### 4.12.3.2 int Q2TreeData::CutOff ( int *index* ) [protected]

Removes the subtree at the specified index, leaving only the root.

#### Returns

Number of leaf nodes removed.

#### 4.12.3.3 void Q2TreeData::DebugPrint ( )

For debugging purposes.

#### 4.12.3.4 bool Q2TreeData::Deserialize ( std::istream & *in* )

Reads a list of quads and builds a tree on top of it.

#### 4.12.3.5 clock\_t Q2TreeData::GetLastUpdate ( ) [inline]

#### 4.12.3.6 int Q2TreeData::GetLeafCount ( )

#### 4.12.3.7 int Q2TreeData::GetLength ( ) [inline]

#### 4.12.3.8 void Q2TreeData::Merge ( Q2TreeData \* *result*, Q2TreeData \*\* *input*, bool \* *ignore*, int *inputCount* ) [static]

Merges input Q2Trees into a single normalized Q2Tree.

#### 4.12.3.9 void Q2TreeData::Normalize ( int *nTreesMerged* ) [protected]

Normalizes the Q2Tree, so that the luminance of each node equals the sum of luminances of its four children. The luminance of the base quads is preserved.

#### 4.12.3.10 void Q2TreeData::PlaceSamples ( SamplingData & *samplingData* )

Places one sample in each leaf node.

4.12.3.11 `bool Q2TreeData::ReadSerializedData ( std::istream & in )` `[protected]`

Deserialization subroutine.

4.12.3.12 `void Q2TreeData::RecursiveMerge ( Q2TreeData * result, int resultIndex, int parentIndex, int & childIndex, Q2TreeData ** input, int * inputIndices, int inputCount )` `[static]`, `[protected]`

Merge subroutine.

4.12.3.13 `void Q2TreeData::RecursiveNormalize ( int index )` `[protected]`

Normalize subroutine.

4.12.3.14 `void Q2TreeData::Reduce ( int numSamples )`

Reduces the number of leaf nodes to 'numSamples'.

At each iteration, the lowest importance interior node is selected and its children are cut-off, making the interior node a leaf. One iteration reduces the number of leaves by 3. This operation is repeated until the number of leaf nodes is less than 'numSamples'. Note: for merged trees, the lowest importance interior node's children might not be all leaf nodes. In such cases, the entire subtree is cut off and so the number of leaf nodes is reduced by more than 3.

4.12.3.15 `void Q2TreeData::SortSubtree ( int root )` `[protected]`

Deserialization subroutine.

The documentation for this class was generated from the following files:

- [C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.h](#)
- [C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.cpp](#)

## 4.13 Q2TreeData::Quad Struct Reference

Represents a quad in the HEALPix mapping.

```
#include <Q2TreeData.h>
```

### Public Member Functions

- `bool IsLeaf ()`
- `void EvaluateImportance ()`

### Static Public Member Functions

- `static bool ComparePosition (Quad &q1, Quad &q2)`  
*comparison function for sorting*

## Public Attributes

- long `id`  
*this quad's id*
- long `parent`  
*parent's id*
- int `children` [4]  
*children's indexes (not IDs)*
- float `solidAngle`  
*solid angle of the quad*
- float `luminance`  
*luminance of the quad*
- int `x`
- int `y`  
*position*
- int `face`  
*q2-tree face*
- float `importance`  
*importance of the quad (based on luminance and solid angle,  $I = L^a \cdot W^b$ )*

### 4.13.1 Detailed Description

Represents a quad in the HEALPix mapping.

### 4.13.2 Member Function Documentation

4.13.2.1 `static bool Q2TreeData::Quad::ComparePosition ( Quad & q1, Quad & q2 )` [inline], [static]

comparison function for sorting

4.13.2.2 `void Q2TreeData::Quad::EvaluateImportance ( )` [inline]

4.13.2.3 `bool Q2TreeData::Quad::IsLeaf ( )` [inline]

### 4.13.3 Member Data Documentation

4.13.3.1 `int Q2TreeData::Quad::children[4]`

children's indexes (not IDs)

4.13.3.2 `int Q2TreeData::Quad::face`

q2-tree face

4.13.3.3 `long Q2TreeData::Quad::id`

this quad's id

4.13.3.4 `float Q2TreeData::Quad::importance`

importance of the quad (based on luminance and solid angle,  $I = L^a \cdot W^b$ )

## 4.13.3.5 float Q2TreeData::Quad::luminance

luminance of the quad

## 4.13.3.6 long Q2TreeData::Quad::parent

parent's id

## 4.13.3.7 float Q2TreeData::Quad::solidAngle

solid angle of the quad

## 4.13.3.8 int Q2TreeData::Quad::x

## 4.13.3.9 int Q2TreeData::Quad::y

position

The documentation for this struct was generated from the following file:

- <C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.h>

## 4.14 Q2TreeData::QuadIdComparer Class Reference

comparison class for sorting

```
#include <Q2TreeData.h>
```

### Public Member Functions

- [QuadIdComparer](#) ([Quad](#) \*quadArray)
- bool [operator\(\)](#) (int index1, int index2)

#### 4.14.1 Detailed Description

comparison class for sorting

#### 4.14.2 Constructor & Destructor Documentation

4.14.2.1 Q2TreeData::QuadIdComparer::QuadIdComparer ( [Quad](#) \* *quadArray* ) [inline]

#### 4.14.3 Member Function Documentation

4.14.3.1 bool Q2TreeData::QuadIdComparer::operator() ( int *index1*, int *index2* ) [inline]

The documentation for this class was generated from the following file:

- <C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.h>

## 4.15 Q2TreeData::QuadImportanceComparer Class Reference

comparison class for sorting

```
#include <Q2TreeData.h>
```

### Public Member Functions

- [QuadImportanceComparer](#) ([Quad](#) \*quadArray)
- bool [operator\(\)](#) (int index1, int index2)

#### 4.15.1 Detailed Description

comparison class for sorting

#### 4.15.2 Constructor & Destructor Documentation

4.15.2.1 [Q2TreeData::QuadImportanceComparer::QuadImportanceComparer \( \[Quad\]\(#\) \\* \*quadArray\* \)](#) `[inline]`

#### 4.15.3 Member Function Documentation

4.15.3.1 [bool Q2TreeData::QuadImportanceComparer::operator\(\) \( \[int\]\(#\) \*index1\*, \[int\]\(#\) \*index2\* \)](#) `[inline]`

The documentation for this class was generated from the following file:

- C:/DATA/Repo/MasterThesis/src/MultiCamClient/[Q2TreeData.h](#)

## 4.16 Q2TreeData::QuadPositionComparer Class Reference

comparison class for sorting

```
#include <Q2TreeData.h>
```

### Public Member Functions

- [QuadPositionComparer](#) ([Quad](#) \*quadArray)
- bool [operator\(\)](#) (int index1, int index2)

#### 4.16.1 Detailed Description

comparison class for sorting

#### 4.16.2 Constructor & Destructor Documentation

4.16.2.1 [Q2TreeData::QuadPositionComparer::QuadPositionComparer \( \[Quad\]\(#\) \\* \*quadArray\* \)](#) `[inline]`

#### 4.16.3 Member Function Documentation

4.16.3.1 [bool Q2TreeData::QuadPositionComparer::operator\(\) \( \[int\]\(#\) \*index1\*, \[int\]\(#\) \*index2\* \)](#) `[inline]`

The documentation for this class was generated from the following file:

- [C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.h](#)

## 4.17 MultiCamClient::SampleComparer Struct Reference

A comparer function object for ordering of samples.

```
#include <MultiCamClient.h>
```

### Public Member Functions

- `bool operator() (const SamplingData::Sample &a, const SamplingData::Sample &b) const`

#### 4.17.1 Detailed Description

A comparer function object for ordering of samples.

#### 4.17.2 Member Function Documentation

4.17.2.1 `bool MultiCamClient::SampleComparer::operator() ( const SamplingData::Sample & a, const SamplingData::Sample & b ) const` `[inline]`

The documentation for this struct was generated from the following file:

- [C:/DATA/Repo/MasterThesis/src/MultiCamClient/MultiCamClient.h](#)





## Chapter 5

# File Documentation

### 5.1 C:/DATA/Repo/MasterThesis/src/Client/Client.cpp File Reference

```
#include "Client.h"  
#include <typeinfo>  
#include "MultiCamClient.h"  
#include "HttpDataClient.h"
```

### 5.2 C:/DATA/Repo/MasterThesis/src/Client/Client.h File Reference

```
#include <QObject>  
#include "IClient.h"  
#include "Logger.h"
```

#### Classes

- class [Client](#)

*A wrapper that provides access to both single and multi camera clients.*

### 5.3 C:/DATA/Repo/MasterThesis/src/Client/Host.h File Reference

```
#include <QString>
```

#### Classes

- struct [Host](#)

*This structure represents the address and port of a TCP host.*

### 5.4 C:/DATA/Repo/MasterThesis/src/Client/IClient.h File Reference

```
#include <QObject>  
#include "SamplingData.h"  
#include "Host.h"
```

## Classes

- class [IClient](#)

*This abstract class represents an interface of all the API client classes.*

## 5.5 C:/DATA/Repo/MasterThesis/src/HttpClient/GenericHttpClient.cpp File Reference

```
#include "GenericHttpClient.h"  
#include "Logger.h"  
#include <QStringList>  
#include <iostream>  
#include <sstream>
```

## 5.6 C:/DATA/Repo/MasterThesis/src/HttpClient/GenericHttpClient.h File Reference

```
#include <QTcpSocket>  
#include <QMutex>  
#include <QTimer>  
#include "Logger.h"  
#include "SamplingData.h"  
#include "IClient.h"
```

## Classes

- class [GenericHttpClient](#)

*This abstract class implements routines for initializing and maintaining the connection to the server over HTTP, as well as error handling.*

## 5.7 C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.cpp File Reference

```
#include "HdrDownloader.h"  
#include "Logger.h"  
#include <QStringList>  
#include <iostream>  
#include <sstream>
```

## 5.8 C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.h File Reference

```
#include <string>  
#include <QTcpSocket>  
#include <QMutex>  
#include <QTimer>  
#include "Logger.h"  
#include "SamplingData.h"  
#include "GenericHttpClient.h"
```

## Classes

- class [HdrDownloader](#)

*This class implements parsing of .hdr images from a network stream.*

## 5.9 C:/DATA/Repo/MasterThesis/src/HttpClient/HttpDataClient.cpp File Reference

```
#include "HttpDataClient.h"
#include "Logger.h"
#include <QStringList>
#include <iostream>
#include <sstream>
```

## 5.10 C:/DATA/Repo/MasterThesis/src/HttpClient/HttpDataClient.h File Reference

```
#include <QTcpSocket>
#include <QMutex>
#include <QTimer>
#include "Logger.h"
#include "SamplingData.h"
#include "GenericHttpClient.h"
```

## Classes

- class [HttpDataClient](#)

*This class implements parsing of sampling data (set of directional light sources) from a network stream.*

## 5.11 C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2TreeClient.cpp File Reference

```
#include "HttpQ2TreeClient.h"
#include "Logger.h"
#include <QStringList>
#include <iostream>
#include <sstream>
```

## 5.12 C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2TreeClient.h File Reference

```
#include <QTcpSocket>
#include <QMutex>
#include <QTimer>
#include "Logger.h"
#include "Q2TreeData.h"
#include "GenericHttpClient.h"
```

## Classes

- class [HttpQ2TreeClient](#)

*This class implements parsing of Q2-Trees from a network stream.*

## 5.13 C:/DATA/Repo/MasterThesis/src/HttpClient/main.cpp File Reference

```
#include <QtCore/QCoreApplication>
#include "HttpClient.h"
```

## Functions

- int [main](#) (int argc, char \*argv[])

*Sample of GICAR API usage.*

### 5.13.1 Function Documentation

5.13.1.1 int main ( int argc, char \* argv[] )

Sample of GICAR API usage.

## 5.14 C:/DATA/Repo/MasterThesis/src/MultiCamClient/main.cpp File Reference

```
#include <QtCore/QCoreApplication>
#include "MultiCamClient.h"
```

## Functions

- int [main](#) (int argc, char \*argv[])

*Sample of GICAR API usage.*

### 5.14.1 Function Documentation

5.14.1.1 int main ( int argc, char \* argv[] )

Sample of GICAR API usage.

## 5.15 C:/DATA/Repo/MasterThesis/src/MultiCamClient/MultiCamClient.cpp File Reference

```
#include "MultiCamClient.h"
#include "Logger.h"
#include "Settings.h"
#include <QStringList>
#include <iostream>
#include <sstream>
#include <map>
```

## 5.16 C:/DATA/Repo/MasterThesis/src/MultiCamClient/MultiCamClient.h File Reference

```
#include <ctime>
#include <map>
#include <QMutex>
#include <QSharedPointer>
#include "Logger.h"
#include "SamplingData.h"
#include "HttpDataClient.h"
#include "HttpQ2TreeClient.h"
#include "IClient.h"
#include "Q2TreeData.h"
```

### Classes

- class [MultiCamClient](#)  
*This class allows for simultaneous connection to multiple servers and implements merging of received data.*
- struct [MultiCamClient::SampleComparer](#)  
*A comparer function object for ordering of samples.*

## 5.17 C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.cpp File Reference

```
#include "Q2TreeData.h"
#include <map>
#include <algorithm>
#include <list>
#include <assert.h>
#include "healpix.h"
#include "Settings.h"
```

## 5.18 C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2TreeData.h File Reference

```
#include <iostream>
#include <ctime>
#include <cmath>
#include "SamplingData.h"
```

### Classes

- class [Q2TreeData](#)  
*This class implements merging of Q2-Trees.*
- struct [Q2TreeData::Quad](#)  
*Represents a quad in the HEALPix mapping.*
- class [Q2TreeData::QuadPositionComparer](#)  
*comparison class for sorting*
- class [Q2TreeData::QuadIdComparer](#)  
*comparison class for sorting*
- class [Q2TreeData::QuadImportanceComparer](#)  
*comparison class for sorting*

### 5.19 C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.cpp File Reference

```
#include "HdrWriter.h"  
#include <cstring>  
#include <cstdio>  
#include <sstream>
```

### 5.20 C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.h File Reference

```
#include <iostream>  
#include <cmath>
```

#### Classes

- class [HdrWriter](#)

*This class implements reading and writing of HDR images in .hdr format.*

### 5.21 C:/DATA/Repo/MasterThesis/src/shared/Logger.cpp File Reference

```
#include "Logger.h"
```

### 5.22 C:/DATA/Repo/MasterThesis/src/shared/Logger.h File Reference

```
#include <iostream>  
#include <fstream>
```

#### Classes

- class [Logger](#)  
*A logger class.*
- class [Logger::EndOfLine](#)

# Index

- ~Client
  - Client, 8
- ~GenericHttpClient
  - GenericHttpClient, 10
- ~IClient
  - IClient, 19
- ~MultiCamClient
  - MultiCamClient, 23
- AdvancedQ2TreeMergeMode
  - MultiCamClient, 23
- BuildTree
  - Q2TreeData, 26
- C:/DATA/Repo/MasterThesis/src/Client/Client.cpp, 33
- C:/DATA/Repo/MasterThesis/src/Client/Client.h, 33
- C:/DATA/Repo/MasterThesis/src/Client/Host.h, 33
- C:/DATA/Repo/MasterThesis/src/Client/IClient.h, 33
- C:/DATA/Repo/MasterThesis/src/HttpClient/Generic-  
HttpClient.cpp, 34
- C:/DATA/Repo/MasterThesis/src/HttpClient/Generic-  
HttpClient.h, 34
- C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.  
cpp, 34
- C:/DATA/Repo/MasterThesis/src/HttpClient/HdrDownloader.  
h, 34
- C:/DATA/Repo/MasterThesis/src/HttpClient/HttpData-  
Client.cpp, 35
- C:/DATA/Repo/MasterThesis/src/HttpClient/HttpData-  
Client.h, 35
- C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2-  
TreeClient.cpp, 35
- C:/DATA/Repo/MasterThesis/src/HttpClient/HttpQ2-  
TreeClient.h, 35
- C:/DATA/Repo/MasterThesis/src/HttpClient/main.cpp,  
36
- C:/DATA/Repo/MasterThesis/src/MultiCamClient/Multi-  
CamClient.cpp, 36
- C:/DATA/Repo/MasterThesis/src/MultiCamClient/Multi-  
CamClient.h, 37
- C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2-  
TreeData.cpp, 37
- C:/DATA/Repo/MasterThesis/src/MultiCamClient/Q2-  
TreeData.h, 37
- C:/DATA/Repo/MasterThesis/src/MultiCamClient/main.-  
cpp, 36
- C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.cpp,  
38
- C:/DATA/Repo/MasterThesis/src/shared/HdrWriter.h,  
38
- C:/DATA/Repo/MasterThesis/src/shared/Logger.cpp, 38
- C:/DATA/Repo/MasterThesis/src/shared/Logger.h, 38
- children
  - Q2TreeData::Quad, 28
- Client, 7
  - ~Client, 8
  - Client, 8
  - ConnectToHost, 8
  - ConnectToHosts, 8
  - DataReady, 8
  - GetHttpClient, 8
  - GetMultiCamClient, 8
  - IsMultiCam, 8
  - IsRunning, 8
  - ReadData, 8
  - Stop, 9
- ComparePosition
  - Q2TreeData::Quad, 28
- Connect
  - GenericHttpClient, 10
  - MultiCamClient, 23
- ConnectToHost
  - Client, 8
  - GenericHttpClient, 10
- ConnectToHosts
  - Client, 8
  - MultiCamClient, 23
- ConvertRGBFToRGB888
  - HdrWriter, 14
- CutOff
  - Q2TreeData, 26
- DEBUG
  - Logger, 20
- DEBUG2
  - Logger, 20
- DEBUG3
  - Logger, 20
- DataReady
  - Client, 8
  - GenericHttpClient, 11
  - IClient, 19
  - MultiCamClient, 23
- DebugPrint
  - Q2TreeData, 26
- Deserialize
  - Q2TreeData, 26

ERROR  
     Logger, 20  
 endl  
     Logger, 21  
 EvaluateImportance  
     Q2TreeData::Quad, 28  
  
 face  
     Q2TreeData::Quad, 28  
 FloatToRGBE  
     HdrWriter, 14  
  
 GenericHttpClient, 9  
     ~GenericHttpClient, 10  
     Connect, 10  
     ConnectToHost, 10  
     DataReady, 11  
     GenericHttpClient, 10  
     GenericHttpClient, 10  
     GetGUID, 11  
     IsRunning, 11  
     logger, 12  
     m\_guid, 12  
     ParseResponse, 11  
     ReadData, 11  
     response, 12  
     SetGUID, 11  
     SetHost, 11  
     Stop, 11  
     SwapBuffers, 11  
 GetDebugLevel  
     Logger, 21  
 GetGUID  
     GenericHttpClient, 11  
 GetHttpClient  
     Client, 8  
 GetLastUpdate  
     Q2TreeData, 26  
 GetLeafCount  
     Q2TreeData, 26  
 GetLength  
     Q2TreeData, 26  
 GetMultiCamClient  
     Client, 8  
 GrayscaleFloatToRGBE  
     HdrWriter, 14  
  
 HdrDownloader, 12  
     HdrDownloader, 13  
     HdrDownloader, 13  
     ParseResponse, 13  
     ReadData, 13  
     SwapBuffers, 13  
 HdrWriter, 13  
     ConvertRGBFToRGB888, 14  
     FloatToRGBE, 14  
     GrayscaleFloatToRGBE, 14  
     RGBEToFloat, 14  
     Read, 14  
     Write, 15  
 Host, 15  
     Host, 15  
     m\_hostName, 15  
     m\_port, 15  
 HttpClient/main.cpp  
     main, 36  
 HttpDataClient, 16  
     HttpDataClient, 16  
     HttpDataClient, 16  
     ParseResponse, 16  
     ReadData, 16  
     SwapBuffers, 17  
 HttpQ2TreeClient, 17  
     HttpQ2TreeClient, 18  
     HttpQ2TreeClient, 18  
     ParseResponse, 18  
     ReadData, 18  
     ReadQ2Tree, 18  
     SwapBuffers, 18  
  
 INFO  
     Logger, 20  
 IClient, 18  
     ~IClient, 19  
     DataReady, 19  
     IsRunning, 19  
     ReadData, 19  
     Stop, 19  
 id  
     Q2TreeData::Quad, 28  
 importance  
     Q2TreeData::Quad, 28  
 IsLeaf  
     Q2TreeData::Quad, 28  
 IsMultiCam  
     Client, 8  
 IsRunning  
     Client, 8  
     GenericHttpClient, 11  
     IClient, 19  
     MultiCamClient, 23  
  
 Logger, 20  
     DEBUG, 20  
     DEBUG2, 20  
     DEBUG3, 20  
     ERROR, 20  
     endl, 21  
     GetDebugLevel, 21  
     INFO, 20  
     Logger, 21  
     operator<<, 21  
     operator(), 21  
     WARNING, 20  
 logger  
     GenericHttpClient, 12  
 Logger::EndOfLine, 9  
 luminance



- Q2TreeData::Quad, [28](#)
- m\_guid
  - GenericHttpClient, [12](#)
- m\_hostName
  - Host, [15](#)
- m\_port
  - Host, [15](#)
- main
  - HttpClient/main.cpp, [36](#)
  - MultiCamClient/main.cpp, [36](#)
- Merge
  - Q2TreeData, [26](#)
- MergeDeviceData
  - MultiCamClient, [23](#)
- MergePDF
  - MultiCamClient, [23](#)
- MergeQ2Tree
  - MultiCamClient, [24](#)
- MultiCamClient
  - AdvancedQ2TreeMergeMode, [23](#)
  - StandardMergeMode, [23](#)
- MultiCamClient, [21](#)
  - ~MultiCamClient, [23](#)
  - Connect, [23](#)
  - ConnectToHosts, [23](#)
  - DataReady, [23](#)
  - IsRunning, [23](#)
  - MergeDeviceData, [23](#)
  - MergePDF, [23](#)
  - MergeQ2Tree, [24](#)
  - MultiCamClient, [23](#)
  - MultiCamClient, [23](#)
  - NormalizePDF, [24](#)
  - NormalizeQ2Tree, [24](#)
  - ReadData, [24](#)
  - SamplesMap, [22](#)
  - SetNumberOfSamples, [24](#)
  - Stop, [24](#)
  - SwapBuffers, [24](#)
- MultiCamClient/main.cpp
  - main, [36](#)
- MultiCamClient::SampleComparer, [31](#)
  - operator(), [31](#)
- Normalize
  - Q2TreeData, [26](#)
- NormalizePDF
  - MultiCamClient, [24](#)
- NormalizeQ2Tree
  - MultiCamClient, [24](#)
- operator<<
  - Logger, [21](#)
- operator()
  - Logger, [21](#)
  - MultiCamClient::SampleComparer, [31](#)
  - Q2TreeData::QuadIdComparer, [29](#)
  - Q2TreeData::QuadImportanceComparer, [30](#)
- Q2TreeData::QuadPositionComparer, [30](#)
- parent
  - Q2TreeData::Quad, [29](#)
- ParseResponse
  - GenericHttpClient, [11](#)
  - HdrDownloader, [13](#)
  - HttpDataClient, [16](#)
  - HttpQ2TreeClient, [18](#)
- PlaceSamples
  - Q2TreeData, [26](#)
- Q2TreeData, [24](#)
  - BuildTree, [26](#)
  - CutOff, [26](#)
  - DebugPrint, [26](#)
  - Deserialize, [26](#)
  - GetLastUpdate, [26](#)
  - GetLeafCount, [26](#)
  - GetLength, [26](#)
  - Merge, [26](#)
  - Normalize, [26](#)
  - PlaceSamples, [26](#)
  - Q2TreeData, [26](#)
  - Q2TreeData, [26](#)
  - ReadSerializedData, [26](#)
  - RecursiveMerge, [27](#)
  - RecursiveNormalize, [27](#)
  - Reduce, [27](#)
  - SortSubtree, [27](#)
- Q2TreeData::Quad, [27](#)
  - children, [28](#)
  - ComparePosition, [28](#)
  - EvaluateImportance, [28](#)
  - face, [28](#)
  - id, [28](#)
  - importance, [28](#)
  - IsLeaf, [28](#)
  - luminance, [28](#)
  - parent, [29](#)
  - solidAngle, [29](#)
  - x, [29](#)
  - y, [29](#)
- Q2TreeData::QuadIdComparer, [29](#)
  - operator(), [29](#)
  - QuadIdComparer, [29](#)
- Q2TreeData::QuadImportanceComparer, [30](#)
  - operator(), [30](#)
  - QuadImportanceComparer, [30](#)
- Q2TreeData::QuadPositionComparer, [30](#)
  - operator(), [30](#)
  - QuadPositionComparer, [30](#)
- QuadIdComparer
  - Q2TreeData::QuadIdComparer, [29](#)
- QuadImportanceComparer
  - Q2TreeData::QuadImportanceComparer, [30](#)
- QuadPositionComparer
  - Q2TreeData::QuadPositionComparer, [30](#)

RGBToFloat  
     HdrWriter, [14](#)  
 Read  
     HdrWriter, [14](#)  
 ReadData  
     Client, [8](#)  
     GenericHttpClient, [11](#)  
     HdrDownloader, [13](#)  
     HttpDataClient, [16](#)  
     HttpQ2TreeClient, [18](#)  
     IClient, [19](#)  
     MultiCamClient, [24](#)  
 ReadQ2Tree  
     HttpQ2TreeClient, [18](#)  
 ReadSerializedData  
     Q2TreeData, [26](#)  
 RecursiveMerge  
     Q2TreeData, [27](#)  
 RecursiveNormalize  
     Q2TreeData, [27](#)  
 Reduce  
     Q2TreeData, [27](#)  
 response  
     GenericHttpClient, [12](#)  
  
 SamplesMap  
     MultiCamClient, [22](#)  
 SetGUID  
     GenericHttpClient, [11](#)  
 SetHost  
     GenericHttpClient, [11](#)  
 SetNumberOfSamples  
     MultiCamClient, [24](#)  
 solidAngle  
     Q2TreeData::Quad, [29](#)  
 SortSubtree  
     Q2TreeData, [27](#)  
 StandardMergeMode  
     MultiCamClient, [23](#)  
 Stop  
     Client, [9](#)  
     GenericHttpClient, [11](#)  
     IClient, [19](#)  
     MultiCamClient, [24](#)  
 SwapBuffers  
     GenericHttpClient, [11](#)  
     HdrDownloader, [13](#)  
     HttpDataClient, [17](#)  
     HttpQ2TreeClient, [18](#)  
     MultiCamClient, [24](#)  
  
 WARNING  
     Logger, [20](#)  
 Write  
     HdrWriter, [15](#)  
  
 x  
     Q2TreeData::Quad, [29](#)

y

Q2TreeData::Quad, [29](#)