2015 Test beam Run Control

Generated by Doxygen 1.8.9.1

Fri Apr 17 2015 22:53:20

Contents

1 Hierarchical Index **1**

1.1 Class Hierarchy . . . 1

2 Data Structure Index **3**

2.1 Data Structures . . . 3

3 Data Structure Documentation **5**

3.1 Client Class Reference . . . 5

3.1.1 Detailed Description . . . 6

3.1.2 Constructor & Destructor Documentation . . . 6

3.1.2.1 Client . . . 6

3.1.2.2 Client . . . 6

3.1.2.3 ~Client . . . 6

3.1.3 Member Function Documentation . . . 6

3.1.3.1 Connect . . . 6

3.1.3.2 Disconnect . . . 6

3.1.3.3 GetType . . . 6

3.1.3.4 ParseMessage . . . 6

3.1.3.5 Receive . . . 6

3.1.3.6 Send . . . 6

3.2 Exception Class Reference . . . 7

3.2.1 Detailed Description . . . 7

3.2.2 Constructor & Destructor Documentation . . . 7

3.2.2.1 Exception . . . 7

3.2.2.2 Exception . . . 7

3.2.2.3 ~Exception . . . 7

3.2.3 Member Function Documentation . . . 8

3.2.3.1 Description . . . 8

3.2.3.2 Dump . . . 8

3.2.3.3 ErrorNumber . . . 8

3.2.3.4 From . . . 9

3.2.3.5 Type . . . 9

3.2.3.6 TypeString . . . 9
3.3 file_header_t Struct Reference . . . 10
3.3.1 Field Documentation . . . 10
3.3.1.1 magic . . . 10
3.3.1.2 run_id . . . 10
3.3.1.3 spill_id . . . 10
3.4 FPGAHandler Class Reference . . . 10
3.4.1 Detailed Description . . . 12
3.4.2 Constructor & Destructor Documentation . . . 12
3.4.2.1 FPGAHandler . . . 12
3.4.2.2 ∼FPGAHandler . . . 12
3.4.3 Member Function Documentation . . . 12
3.4.3.1 GetFilename . . . 12
3.4.3.2 GetType . . . 12
3.4.3.3 OpenFile . . . 12
3.4.3.4 ReadBuffer . . . 12
3.4.3.5 ReadConfiguration . . . 12
3.4.3.6 SendConfiguration . . . 12
3.5 HTTPMessage Class Reference . . . 12
3.5.1 Constructor & Destructor Documentation . . . 13
3.5.1.1 HTTPMessage . . . 13
3.5.1.2 HTTPMessage . . . 14
3.5.2 Member Function Documentation . . . 14
3.5.2.1 Decode . . . 14
3.5.2.2 Dump . . . 14
3.5.2.3 Encode . . . 14
3.5.2.4 GetKey . . . 14
3.6 ListenerInfo Struct Reference . . . 15
3.6.1 Field Documentation . . . 15
3.6.1.1 name . . . 15
3.6.1.2 type . . . 15
3.7 Message Class Reference . . . 15
3.7.1 Detailed Description . . . 16
3.7.2 Constructor & Destructor Documentation . . . 16
3.7.2.1 Message . . . 16
3.7.2.2 Message . . . 16
3.7.2.3 Message . . . 16
3.7.2.4 ∼Message . . . 16
3.7.3 Member Function Documentation . . . 16
3.7.3.1 Dump . . . 16

3.7.3.2 GetKey . . . 16
3.7.3.3 GetString . . . 16
3.7.3.4 IsFromWeb . . . 16
3.7.4 Field Documentation . . . 16
3.7.4.1 fString . . . 16
3.8 Messenger Class Reference . . . 17
3.8.1 Detailed Description . . . 18
3.8.2 Constructor & Destructor Documentation . . . 18
3.8.2.1 Messenger . . . 18
3.8.2.2 Messenger . . . 18
3.8.2.3 ∼Messenger . . . 18
3.8.3 Member Function Documentation . . . 18
3.8.3.1 AddClient . . . 18
3.8.3.2 Broadcast . . . 18
3.8.3.3 Connect . . . 18
3.8.3.4 Disconnect . . . 18
3.8.3.5 DisconnectClient . . . 18
3.8.3.6 ProcessMessage . . . 18
3.8.3.7 Receive . . . 19
3.8.3.8 Send . . . 19
3.9 Socket Class Reference . . . 19
3.9.1 Detailed Description . . . 20
3.9.2 Constructor & Destructor Documentation . . . 21
3.9.2.1 Socket . . . 21
3.9.2.2 Socket . . . 21
3.9.2.3 ∼Socket . . . 21
3.9.3 Member Function Documentation . . . 21
3.9.3.1 AcceptConnections . . . 21
3.9.3.2 Bind . . . 21
3.9.3.3 DumpConnected . . . 21
3.9.3.4 FetchMessage . . . 21
3.9.3.5 GetPort . . . 21
3.9.3.6 GetSocketId . . . 21
3.9.3.7 GetSocketType . . . 22
3.9.3.8 IsWebSocket . . . 22
3.9.3.9 Listen . . . 22
3.9.3.10 PrepareConnection . . . 22
3.9.3.11 SelectConnections . . . 22
3.9.3.12 SendMessage . . . 22
3.9.3.13 SetPort . . . 22

3.9.3.14 SetSocketId . . . 22
3.9.3.15 Start . . . 22
3.9.3.16 Stop . . . 23
3.9.4 Field Documentation . . . 23
3.9.4.1 fBuffer . . . 23
3.9.4.2 fMaster . . . 23
3.9.4.3 fPort . . . 23
3.9.4.4 fReadFds . . . 23
3.9.4.5 fSocketsConnected . . . 23
3.10 SocketMessage Class Reference . . . 23
3.10.1 Detailed Description . . . 24
3.10.2 Constructor & Destructor Documentation . . . 25
3.10.2.1 SocketMessage . . . 25
3.10.2.2 SocketMessage . . . 25
3.10.2.3 SocketMessage . . . 25
3.10.2.4 SocketMessage . . . 25
3.10.2.5 SocketMessage . . . 25
3.10.2.6 SocketMessage . . . 25
3.10.2.7 SocketMessage . . . 25
3.10.2.8 SocketMessage . . . 26
3.10.2.9 SocketMessage . . . 26
3.10.2.10 SocketMessage . . . 26
3.10.2.11 SocketMessage . . . 26
3.10.2.12 ∼SocketMessage . . . 26
3.10.3 Member Function Documentation . . . 26
3.10.3.1 Dump . . . 27
3.10.3.2 GetIntValue . . . 27
3.10.3.3 GetKey . . . 27
3.10.3.4 GetString . . . 27
3.10.3.5 GetValue . . . 27
3.10.3.6 GetVectorValue . . . 28
3.10.3.7 SetKeyValue . . . 28
3.10.3.8 SetKeyValue . . . 28
3.10.3.9 SetKeyValue . . . 28
3.10.3.10 SetKeyValue . . . 29
3.10.3.11 SetKeyValue . . . 29
3.11 TDCConfiguration Class Reference . . . 29
3.11.1 Detailed Description . . . 30
3.11.2 Constructor & Destructor Documentation . . . 30
3.11.2.1 TDCConfiguration . . . 30

3.11.2.2 ~TDCConfiguration 30
3.11.3 Member Function Documentation 30
3.11.3.1 Dump 30
3.11.3.2 GetChannelOffset 30
3.11.3.3 SetAllChannelsOffset 30
3.11.3.4 SetChannelOffset 30

Index **33**

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

- Exception . . . 7
- file_header_t . . . 10
- ListenerInfo . . . 15
- Message . . . 15
  - HTTPMessage . . . 12
  - SocketMessage . . . 23
- Socket . . . 19
  - Client . . . 5
    - FPGAHandler . . . 10
  - Messenger . . . 17
- TDCConfiguration . . . 29

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Client 5
Exception
A simple exception handler 7
file_header_t 10
FPGAHandler 10
HTTPMessage 12
ListenerInfo 15
Message
Base message type 15
Messenger 17
Socket 19
SocketMessage
Socket-passed message type 23
TDCConfiguration 29

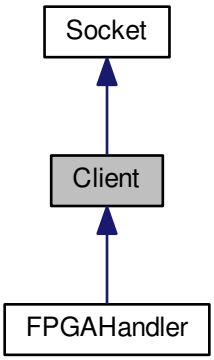
Chapter 3

Data Structure Documentation

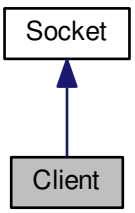
3.1 Client Class Reference

`#include <Client.h>`

Inheritance diagram for Client:



Collaboration diagram for Client:



Public Member Functions

- Client ()
- Client (int port)
- ~Client ()
- bool Connect ()
- void Disconnect ()
- void Send (const Message &m) const
- void Receive ()
- virtual void ParseMessage (const SocketMessage &m)
- virtual SocketType GetType () const

Additional Inherited Members

3.1.1 Detailed Description

Client object used by the server to send/receive commands from the messenger/broadcaster.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

24 Mar 2015

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Client::Client ( ) [inline]

3.1.2.2 Client::Client (int *port*)

3.1.2.3 Client::~Client ( )

3.1.3 Member Function Documentation

3.1.3.1 bool Client::Connect ( )

3.1.3.2 void Client::Disconnect ( )

3.1.3.3 virtual SocketType Client::GetType ( ) const [inline], [virtual]

Reimplemented in FPGAHandler.

3.1.3.4 virtual void Client::ParseMessage (const SocketMessage & *m*) [inline], [virtual]

3.1.3.5 void Client::Receive ( )

3.1.3.6 void Client::Send (const Message & *m*) const

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

- include/Client.h

3.2 Exception Class Reference

A simple exception handler.

`#include <Exception.h>`

Public Member Functions

- Exception (const char *from, std::string desc, ExceptionType type=Undefined, const int id=0)
- Exception (const char *from, const char *desc, ExceptionType type=Undefined, const int id=0)
- ~Exception ()
- std::string From () const
- int ErrorNumber () const
- std::string Description () const
- ExceptionType Type () const
- std::string TypeString () const
- void Dump (std::ostream &os=std::cerr) const

3.2.1 Detailed Description

A simple exception handler.

Author

Laurent Forthomme `laurent.forthomme@cern.ch`

Date

24 Mar 2015

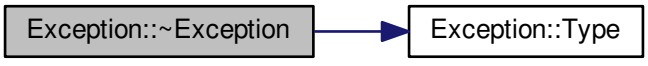
3.2.2 Constructor & Destructor Documentation

3.2.2.1 Exception::Exception (const char * *from*, std::string *desc*, ExceptionType *type* = `Undefined`, const int *id* = `0`) `[inline]`

3.2.2.2 Exception::Exception (const char * *from*, const char * *desc*, ExceptionType *type* = `Undefined`, const int *id* = `0`) `[inline]`

3.2.2.3 Exception::~Exception ( ) `[inline]`

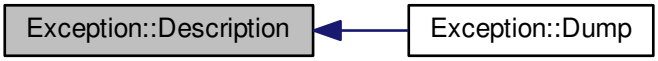
Here is the call graph for this function:



3.2.3 Member Function Documentation

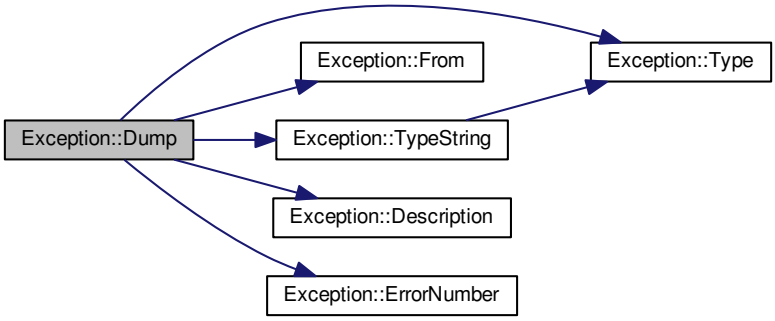
3.2.3.1 std::string Exception::Description ( ) const [inline]

Here is the caller graph for this function:



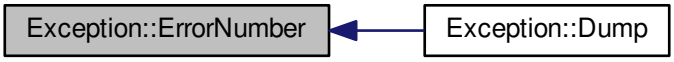
3.2.3.2 void Exception::Dump (std::ostream & *os* = std::cerr) const [inline]

Here is the call graph for this function:



3.2.3.3 int Exception::ErrorNumber ( ) const [inline]

Here is the caller graph for this function:

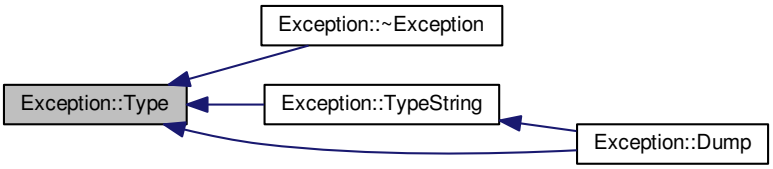


3.2.3.4 std::string Exception::From ( ) const [inline]

Here is the caller graph for this function:

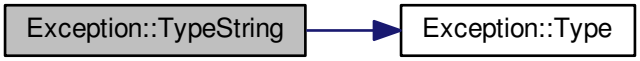
3.2.3.5 ExceptionType Exception::Type ( ) const [inline]

Here is the caller graph for this function:

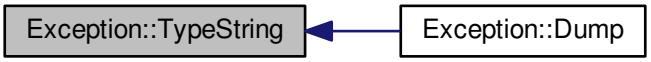


3.2.3.6 std::string Exception::TypeString ( ) const [inline]

Here is the call graph for this function:



Here is the caller graph for this function:



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

- include/Exception.h

3.3 file_header_t Struct Reference

`#include <FPGAHandler.h>`

Data Fields

- uint32_t magic
- uint32_t run_id
- uint32_t spill_id

3.3.1 Field Documentation

3.3.1.1 uint32_t file_header_t::magic

3.3.1.2 uint32_t file_header_t::run_id

3.3.1.3 uint32_t file_header_t::spill_id

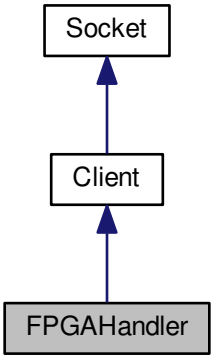
The documentation for this struct was generated from the following file:

- include/FPGAHandler.h

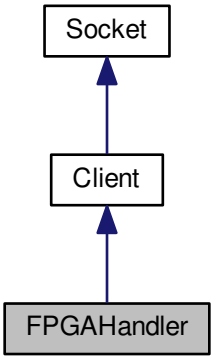
3.4 FPGAHandler Class Reference

`#include <FPGAHandler.h>`

Inheritance diagram for FPGAHandler:



Collaboration diagram for FPGAHandler:



Public Member Functions

- FPGAHandler (int port, const char ∗dev)
- virtual ∼FPGAHandler ()
- void OpenFile ()
- std::string GetFilename () const
- void SendConfiguration (const TDCConfiguration &c)
- TDCConfiguration ReadConfiguration ()
- void ReadBuffer ()
- SocketType GetType () const

Additional Inherited Members

3.4.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

14 Apr 2015

3.4.2 Constructor & Destructor Documentation

3.4.2.1 FPGAHandler::FPGAHandler (int *port*, const char * *dev*)

3.4.2.2 virtual FPGAHandler::~FPGAHandler ( ) [virtual]

3.4.3 Member Function Documentation

3.4.3.1 std::string FPGAHandler::GetFilename ( ) const [inline]

3.4.3.2 SocketType FPGAHandler::GetType ( ) const [inline], [virtual]

Reimplemented from Client.

3.4.3.3 void FPGAHandler::OpenFile ( )

3.4.3.4 void FPGAHandler::ReadBuffer ( )

3.4.3.5 TDCConfiguration FPGAHandler::ReadConfiguration ( )

3.4.3.6 void FPGAHandler::SendConfiguration (const TDCConfiguration & *c*)

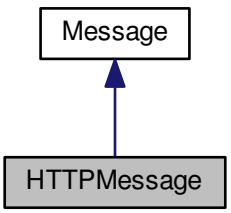
The documentation for this class was generated from the following file:

- include/FPGAHandler.h

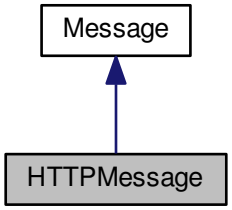
3.5 HTTPMessage Class Reference

`#include <HTTPMessage.h>`

Inheritance diagram for HTTPMessage:



Collaboration diagram for HTTPMessage:



Public Member Functions

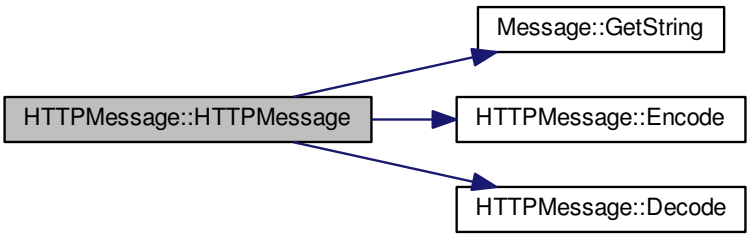
- HTTPMessage (WebSocket ∗ws, Message m, MessageAction a)
- HTTPMessage (WebSocket ∗ws, const char ∗msg, MessageAction a)
- void Decode ()
- void Encode ()
- MessageKey GetKey () const
- void Dump (std::ostream &os=std::cout) const

Additional Inherited Members

3.5.1 Constructor & Destructor Documentation

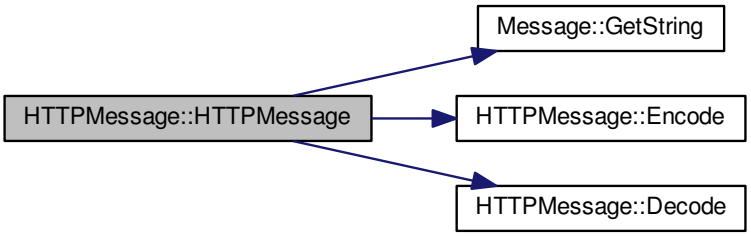
3.5.1.1 HTTPMessage::HTTPMessage (WebSocket ∗ *ws,* Message *m,* MessageAction *a*) [inline]

Here is the call graph for this function:



3.5.1.2 HTTPMessage::HTTPMessage (WebSocket * *ws*, const char * *msg*, MessageAction *a*) [inline]

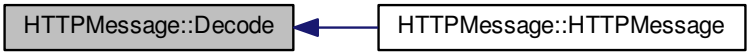
Here is the call graph for this function:



3.5.2 Member Function Documentation

3.5.2.1 void HTTPMessage::Decode ( ) [inline]

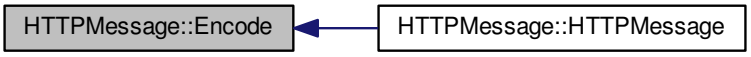
Here is the caller graph for this function:



3.5.2.2 void HTTPMessage::Dump (std::ostream & *os* = std::cout) const [inline]

3.5.2.3 void HTTPMessage::Encode ( ) [inline]

Here is the caller graph for this function:



3.5.2.4 MessageKey HTTPMessage::GetKey ( ) const [inline]

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

- include/HTTPMessage.h

3.6 ListenerInfo Struct Reference

`#include <Messenger.h>`

Data Fields

- std::string name
- int type

3.6.1 Field Documentation

3.6.1.1 std::string ListenerInfo::name

3.6.1.2 int ListenerInfo::type

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

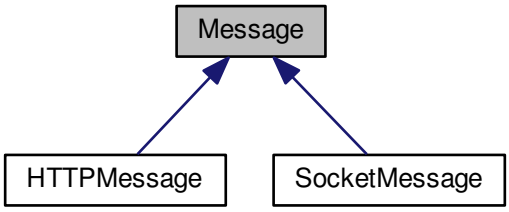
- include/Messenger.h

3.7 Message Class Reference

Base message type.

`#include <Message.h>`

Inheritance diagram for Message:



Public Member Functions

- Message ()
- Message (const char *msg)
- Message (std::string msg)
- ~Message ()
- MessageKey GetKey () const
- std::string GetString () const
- bool IsFromWeb () const
- void Dump (std::ostream &os=std::cout) const

Protected Attributes

- std::string fString

3.7.1 Detailed Description

Base message type.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

6 Apr 2015

3.7.2 Constructor & Destructor Documentation

3.7.2.1 Message::Message ( ) [inline]

3.7.2.2 Message::Message (const char * *msg*) [inline]

3.7.2.3 Message::Message (std::string *msg*) [inline]

3.7.2.4 Message::~Message ( ) [inline]

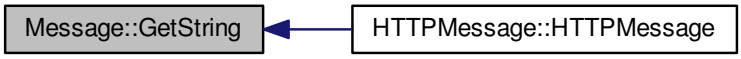
3.7.3 Member Function Documentation

3.7.3.1 void Message::Dump (std::ostream & *os* = std::cout) const [inline]

3.7.3.2 MessageKey Message::GetKey ( ) const [inline]

3.7.3.3 std::string Message::GetString ( ) const [inline]

Here is the caller graph for this function:



3.7.3.4 bool Message::IsFromWeb ( ) const [inline]

3.7.4 Field Documentation

3.7.4.1 std::string Message::fString [protected]

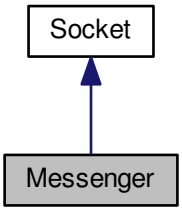
The documentation for this class was generated from the following file:

- include/Message.h

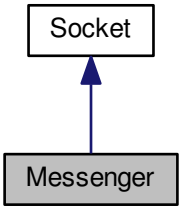
3.8 Messenger Class Reference

`#include <Messenger.h>`

Inheritance diagram for Messenger:



Collaboration diagram for Messenger:



Public Member Functions

- Messenger ()
- Messenger (int port)
- ~Messenger ()
- bool Connect ()
- void Disconnect ()
- void AddClient ()
- void DisconnectClient (int sid, MessageKey key, bool force=false)

  Disconnect a client.
- void Send (const Message &m, int sid) const

  Send any type of message to any client.
- MessageKey Receive ()

  Handle a message reception from a client.
- void ProcessMessage (SocketMessage m, int sid)

  Process a message received from the socket.
- void Broadcast (const Message &m) const

  Emit a message to all clients connected through the socket.

Additional Inherited Members

3.8.1 Detailed Description

Messenger/broadcaster object used by the server to send/receive commands from the clients/listeners.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

23 Mar 2015

3.8.2 Constructor & Destructor Documentation

3.8.2.1 Messenger::Messenger ( )

3.8.2.2 Messenger::Messenger (int *port*)

3.8.2.3 Messenger::~Messenger ( )

3.8.3 Member Function Documentation

3.8.3.1 void Messenger::AddClient ( )

3.8.3.2 void Messenger::Broadcast (const Message & *m*) const

Emit a message to all clients connected through the socket.

Parameters

in	*m*	Message to transmit

3.8.3.3 bool Messenger::Connect ( )

3.8.3.4 void Messenger::Disconnect ( )

3.8.3.5 void Messenger::DisconnectClient (int *sid*, MessageKey *key*, bool *force* = false)

Disconnect a client.

Ask to a client to disconnect from this socket

Parameters

in	*sid*	Unique identifier of the client to disconnect
in	*key*	Key to the message to transmit for disconnection
in	*force*	Do we need to force the client out of this socket ?

3.8.3.6 void Messenger::ProcessMessage (SocketMessage *m*, int *sid*)

Process a message received from the socket.

Parameters

in	*Unique*	identifier of the client sending the message

3.8.3.7 MessageKey Messenger::Receive ( )

Handle a message reception from a client.

Returns

The key to the message received if successfully parsed

3.8.3.8 void Messenger::Send (const Message & *m*, int *sid*) const [inline]

Send any type of message to any client.

Parameters

in	*m*	Message to transmit
in	*sid*	Unique identifier of the client on this socket

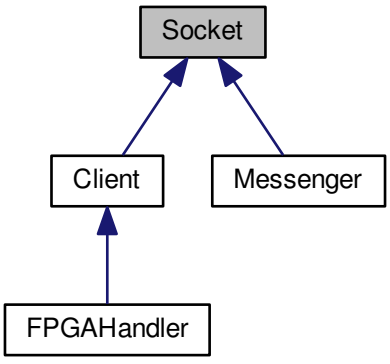
The documentation for this class was generated from the following file:

- include/Messenger.h

3.9 Socket Class Reference

`#include <Socket.h>`

Inheritance diagram for Socket:



Public Member Functions

- Socket ()
- Socket (int port)

- virtual ~Socket ()
- void SetPort (int port)
- int GetPort () const

  Retrieve the port used for this socket.
- void AcceptConnections (Socket &socket)

  Accept connection from a client.
- void SelectConnections ()
- void SetSocketId (int sid)
- int GetSocketId () const
- SocketType GetSocketType (int sid) const
- bool IsWebSocket (int sid) const
- void DumpConnected () const

Protected Member Functions

- bool Start ()

  Start the socket.
- void Stop ()

  Terminates the socket and all attached communications.
- void Bind ()

  Bind a name to a socket.
- void PrepareConnection ()
- void Listen (int maxconn)

  Listen to incoming messages.
- void SendMessage (Message message, int id=-1) const

  Send a message on a socket.
- Message FetchMessage (int id=-1) const

  Receive a message from a socket.

Protected Attributes

- int fPort
- char fBuffer [MAX_WORD_LENGTH]
- SocketCollection fSocketsConnected
- fd_set fMaster

  Master file descriptor list.
- fd_set fReadFds

  Temp file descriptor list for select()

3.9.1 Detailed Description

General object providing all useful method to connect/bind/send/receive information through system sockets.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

23 Mar 2015

3.9.2 Constructor & Destructor Documentation

3.9.2.1 Socket::Socket ( ) [inline]

3.9.2.2 Socket::Socket (int *port*)

3.9.2.3 virtual Socket::~Socket ( ) [virtual]

3.9.3 Member Function Documentation

3.9.3.1 void Socket::AcceptConnections (Socket & *socket*)

Accept connection from a client.

Set the socket to accept connections any client transmitting through the socket

Parameters

in,out	*socket*	Master/client object to enable on the socket

3.9.3.2 void Socket::Bind ( ) [protected]

Bind a name to a socket.

Returns

Success of the operation

3.9.3.3 void Socket::DumpConnected ( ) const

3.9.3.4 Message Socket::FetchMessage (int *id* = -1) const [protected]

Receive a message from a socket.

Returns

Received message as a std::string

3.9.3.5 int Socket::GetPort ( ) const [inline]

Retrieve the port used for this socket.

3.9.3.6 int Socket::GetSocketId ( ) const [inline]

3.9.3.7 SocketType Socket::GetSocketType (int *sid*) const [inline]

Here is the caller graph for this function:

3.9.3.8 bool Socket::IsWebSocket (int *sid*) const [inline]

Here is the call graph for this function:

3.9.3.9 void Socket::Listen (int *maxconn*) [protected]

Listen to incoming messages.

Set the socket to listen to any message coming from outside

3.9.3.10 void Socket::PrepareConnection ( ) [protected]

3.9.3.11 void Socket::SelectConnections ( )

Register all open file descriptors to read their communication through the socket

3.9.3.12 void Socket::SendMessage (Message *message,* int *id* = −1) const [protected]

Send a message on a socket.

3.9.3.13 void Socket::SetPort (int *port*) [inline]

3.9.3.14 void Socket::SetSocketId (int *sid*) [inline]

3.9.3.15 bool Socket::Start ( ) [protected]

Start the socket.

Launch all mandatory operations to set the socket to be used

Returns

Success of the operation

3.9.3.16 void Socket::Stop ( ) [protected]

Terminates the socket and all attached communications.

3.9.4 Field Documentation

3.9.4.1 char Socket::fBuffer[MAX_WORD_LENGTH] [protected]

3.9.4.2 fd_set Socket::fMaster [protected]

Master file descriptor list.

3.9.4.3 int Socket::fPort [protected]

3.9.4.4 fd_set Socket::fReadFds [protected]

Temp file descriptor list for select()

3.9.4.5 SocketCollection Socket::fSocketsConnected [protected]

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

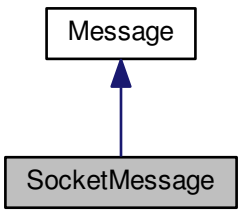
- include/Socket.h

3.10 SocketMessage Class Reference

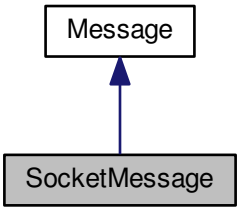
Socket-passed message type.

`#include <SocketMessage.h>`

Inheritance diagram for SocketMessage:



Collaboration diagram for SocketMessage:



Public Member Functions

- SocketMessage ()
- SocketMessage (const Message &msg)
- SocketMessage (const char ∗msg_s)
- SocketMessage (std::string msg_s)
- SocketMessage (MessageKey key)
- SocketMessage (MessageKey key, const char ∗value)
- SocketMessage (MessageKey key, std::string value)
- SocketMessage (MessageKey key, const int value)
- SocketMessage (MessageKey key, const float value)
- SocketMessage (MessageKey key, const double value)
- SocketMessage (MessageMap msg_m)
- ∼SocketMessage ()
- void SetKeyValue (MessageKey key, std::string value)

  Send a string-valued message.
- void SetKeyValue (MessageKey key, const char ∗value)
- void SetKeyValue (MessageKey key, int int_value)

  Send an integer-valued message.
- void SetKeyValue (MessageKey key, float float_value)

  Send an float-valued message.
- void SetKeyValue (MessageKey key, double double_value)

  Send an double-valued message.
- std::string GetString () const
- MessageKey GetKey () const
- std::string GetValue () const
- int GetIntValue () const
- VectorValue GetVectorValue () const
- void Dump (std::ostream &os=std::cout) const

Additional Inherited Members

3.10.1 Detailed Description

Socket-passed message type.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

26 Mar 2015

3.10.2 Constructor & Destructor Documentation

3.10.2.1 SocketMessage::SocketMessage ( ) [inline]

3.10.2.2 SocketMessage::SocketMessage (const Message & *msg*) [inline]

3.10.2.3 SocketMessage::SocketMessage (const char * *msg_s*) [inline]

3.10.2.4 SocketMessage::SocketMessage (std::string *msg_s*) [inline]

3.10.2.5 SocketMessage::SocketMessage (MessageKey *key*) [inline]

Here is the call graph for this function:

3.10.2.6 SocketMessage::SocketMessage (MessageKey *key*, const char * *value*) [inline]

Here is the call graph for this function:

3.10.2.7 SocketMessage::SocketMessage (MessageKey *key*, std::string *value*) [inline]

Here is the call graph for this function:

3.10.2.8 SocketMessage::SocketMessage (MessageKey *key*, const int *value*) [inline]

Here is the call graph for this function:

3.10.2.9 SocketMessage::SocketMessage (MessageKey *key*, const float *value*) [inline]

Here is the call graph for this function:

3.10.2.10 SocketMessage::SocketMessage (MessageKey *key*, const double *value*) [inline]

Here is the call graph for this function:

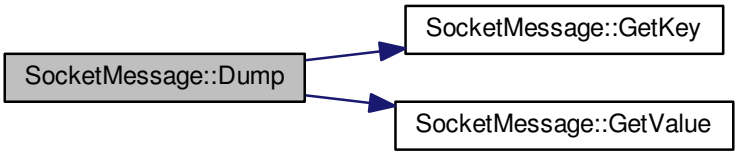
3.10.2.11 SocketMessage::SocketMessage (MessageMap *msg_m*) [inline]

3.10.2.12 SocketMessage::~SocketMessage ( ) [inline]

3.10.3 Member Function Documentation

3.10.3.1 void SocketMessage::Dump (std::ostream & *os* = `std::cout`) const [inline]

Here is the call graph for this function:



3.10.3.2 int SocketMessage::GetIntValue ( ) const [inline]

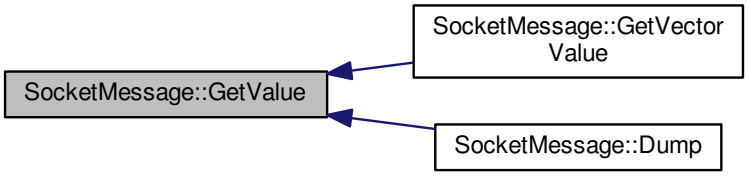
3.10.3.3 MessageKey SocketMessage::GetKey ( ) const [inline]

Here is the caller graph for this function:

3.10.3.4 std::string SocketMessage::GetString ( ) const [inline]

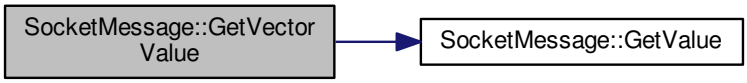
3.10.3.5 std::string SocketMessage::GetValue ( ) const [inline]

Here is the caller graph for this function:



3.10.3.6 VectorValue SocketMessage::GetVectorValue ( ) const [inline]

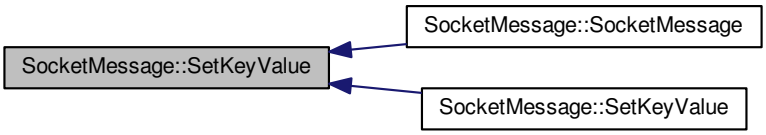
Here is the call graph for this function:



3.10.3.7 void SocketMessage::SetKeyValue (MessageKey *key*, std::string *value*) [inline]

Send a string-valued message.

Here is the caller graph for this function:



3.10.3.8 void SocketMessage::SetKeyValue (MessageKey *key*, const char * *value*) [inline]

Here is the call graph for this function:

3.10.3.9 void SocketMessage::SetKeyValue (MessageKey *key*, int *int_value*) [inline]

Send an integer-valued message.

Here is the call graph for this function:

3.10.3.10 void SocketMessage::SetKeyValue (MessageKey *key,* float *float_value*) [inline]

Send an float-valued message.

Here is the call graph for this function:

3.10.3.11 void SocketMessage::SetKeyValue (MessageKey *key,* double *double_value*) [inline]

Send an double-valued message.

Here is the call graph for this function:

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

- include/SocketMessage.h

3.11 TDCConfiguration Class Reference

`#include <TDCConfiguration.h>`

Public Member Functions

- TDCConfiguration ()
- virtual ~TDCConfiguration ()

- void SetChannelOffset (int channel, short offset)
- short GetChannelOffset (int channel)
- void SetAllChannelsOffset (short offset)
- void Dump () const

3.11.1 Detailed Description

Object handling the configuration word provided by/to the HPTDC chip

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

16 Apr 2015

3.11.2 Constructor & Destructor Documentation

3.11.2.1 TDCConfiguration::TDCConfiguration ( )

3.11.2.2 virtual TDCConfiguration::~TDCConfiguration ( ) [inline], [virtual]

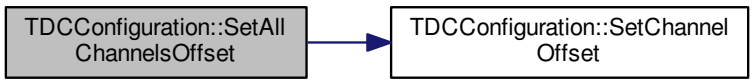
3.11.3 Member Function Documentation

3.11.3.1 void TDCConfiguration::Dump ( ) const

3.11.3.2 short TDCConfiguration::GetChannelOffset (int *channel*)

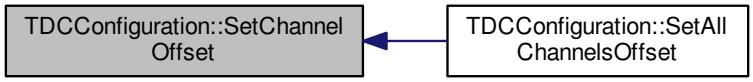
3.11.3.3 void TDCConfiguration::SetAllChannelsOffset (short *offset*) [inline]

Here is the call graph for this function:



3.11.3.4 void TDCConfiguration::SetChannelOffset (int *channel,* short *offset*)

Here is the caller graph for this function:



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

- include/TDCConfiguration.h

Index

~Client
  Client, 6
~Exception
  Exception, 7
~FPGAHandler
  FPGAHandler, 12
~Message
  Message, 16
~Messenger
  Messenger, 18
~Socket
  Socket, 21
~SocketMessage
  SocketMessage, 26
~TDCConfiguration
  TDCConfiguration, 30

AcceptConnections
  Socket, 21
AddClient
  Messenger, 18

Bind
  Socket, 21
Broadcast
  Messenger, 18

Client, 5
  ~Client, 6
  Client, 6
  Connect, 6
  Disconnect, 6
  GetType, 6
  ParseMessage, 6
  Receive, 6
  Send, 6
Connect
  Client, 6
  Messenger, 18

Decode
  HTTPMessage, 14
Description
  Exception, 8
Disconnect
  Client, 6
  Messenger, 18
DisconnectClient
  Messenger, 18
Dump
  Exception, 8
  HTTPMessage, 14
  Message, 16
  SocketMessage, 26
  TDCConfiguration, 30
DumpConnected
  Socket, 21

Encode
  HTTPMessage, 14
ErrorNumber
  Exception, 8
Exception, 7
  ~Exception, 7
  Description, 8
  Dump, 8
  ErrorNumber, 8
  Exception, 7
  From, 8
  Type, 9
  TypeString, 9

fBuffer
  Socket, 23
fMaster
  Socket, 23
FPGAHandler, 10
  ~FPGAHandler, 12
  FPGAHandler, 12
  GetFilename, 12
  GetType, 12
  OpenFile, 12
  ReadBuffer, 12
  ReadConfiguration, 12
  SendConfiguration, 12
fPort
  Socket, 23
fReadFds
  Socket, 23
fSocketsConnected
  Socket, 23
fString
  Message, 16
FetchMessage
  Socket, 21
file_header_t, 10
  magic, 10
  run_id, 10
  spill_id, 10
From

Exception, 8

GetChannelOffset
TDCConfiguration, 30
GetFilename
FPGAHandler, 12
GetIntValue
SocketMessage, 27
GetKey
HTTPMessage, 14
Message, 16
SocketMessage, 27
GetPort
Socket, 21
GetSocketId
Socket, 21
GetSocketType
Socket, 21
GetString
Message, 16
SocketMessage, 27
GetType
Client, 6
FPGAHandler, 12
GetValue
SocketMessage, 27
GetVectorValue
SocketMessage, 27

HTTPMessage, 12
Decode, 14
Dump, 14
Encode, 14
GetKey, 14
HTTPMessage, 13

IsFromWeb
Message, 16
IsWebSocket
Socket, 22

Listen
Socket, 22
ListenerInfo, 15
name, 15
type, 15

magic
file_header_t, 10
Message, 15
∼Message, 16
Dump, 16
fString, 16
GetKey, 16
GetString, 16
IsFromWeb, 16
Message, 16
Messenger, 17
∼Messenger, 18
AddClient, 18
Broadcast, 18
Connect, 18
Disconnect, 18
DisconnectClient, 18
Messenger, 18
ProcessMessage, 18
Receive, 19
Send, 19

name
ListenerInfo, 15

OpenFile
FPGAHandler, 12

ParseMessage
Client, 6
PrepareConnection
Socket, 22
ProcessMessage
Messenger, 18

ReadBuffer
FPGAHandler, 12
ReadConfiguration
FPGAHandler, 12
Receive
Client, 6
Messenger, 19
run_id
file_header_t, 10

SelectConnections
Socket, 22
Send
Client, 6
Messenger, 19
SendConfiguration
FPGAHandler, 12
SendMessage
Socket, 22
SetAllChannelsOffset
TDCConfiguration, 30
SetChannelOffset
TDCConfiguration, 30
SetKeyValue
SocketMessage, 28, 29
SetPort
Socket, 22
SetSocketId
Socket, 22
Socket, 19
∼Socket, 21
AcceptConnections, 21
Bind, 21
DumpConnected, 21
fBuffer, 23
fMaster, 23

fPort, 23
fReadFds, 23
fSocketsConnected, 23
FetchMessage, 21
GetPort, 21
GetSocketId, 21
GetSocketType, 21
IsWebSocket, 22
Listen, 22
PrepareConnection, 22
SelectConnections, 22
SendMessage, 22
SetPort, 22
SetSocketId, 22
Socket, 21
Start, 22
Stop, 23
SocketMessage, 23
~SocketMessage, 26
Dump, 26
GetIntValue, 27
GetKey, 27
GetString, 27
GetValue, 27
GetVectorValue, 27
SetKeyValue, 28, 29
SocketMessage, 25, 26
spill_id
file_header_t, 10
Start
Socket, 22
Stop
Socket, 23

TDCConfiguration, 29
~TDCConfiguration, 30
Dump, 30
GetChannelOffset, 30
SetAllChannelsOffset, 30
SetChannelOffset, 30
TDCConfiguration, 30
Type
Exception, 9
type
ListenerInfo, 15
TypeString
Exception, 9