

2015 Test beam Run Control

Generated by Doxygen 1.8.9.1

Fri Apr 17 2015 15:16:59

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	Exception Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	5
3.1.2.1	Exception	5
3.1.2.2	Exception	5
3.1.2.3	~Exception	6
3.1.3	Member Function Documentation	6
3.1.3.1	Description	6
3.1.3.2	Dump	6
3.1.3.3	ErrorNumber	7
3.1.3.4	From	7
3.1.3.5	Type	7
3.1.3.6	TypeString	8
3.2	file_header_t Struct Reference	8
3.2.1	Field Documentation	8
3.2.1.1	magic	8
3.2.1.2	run_id	8
3.2.1.3	spill_id	8
3.3	FPGAHandler Class Reference	9
3.3.1	Detailed Description	10
3.3.2	Constructor & Destructor Documentation	10
3.3.2.1	FPGAHandler	10
3.3.2.2	~FPGAHandler	10
3.3.3	Member Function Documentation	10
3.3.3.1	GetFilename	10

3.3.3.2	OpenFile	10
3.3.3.3	ReadBuffer	10
3.3.3.4	ReadConfiguration	10
3.3.3.5	SendConfiguration	10
3.4	HTTPMessage Class Reference	10
3.4.1	Constructor & Destructor Documentation	11
3.4.1.1	HTTPMessage	11
3.4.1.2	HTTPMessage	12
3.4.2	Member Function Documentation	12
3.4.2.1	Decode	12
3.4.2.2	Dump	12
3.4.2.3	Encode	12
3.4.2.4	GetKey	12
3.5	Listener Class Reference	13
3.5.1	Detailed Description	14
3.5.2	Constructor & Destructor Documentation	14
3.5.2.1	Listener	14
3.5.2.2	Listener	14
3.5.2.3	~Listener	14
3.5.3	Member Function Documentation	14
3.5.3.1	Connect	14
3.5.3.2	Disconnect	14
3.5.3.3	ParseMessage	14
3.5.3.4	Receive	14
3.5.3.5	Send	14
3.6	ListenerInfo Struct Reference	14
3.6.1	Field Documentation	14
3.6.1.1	name	14
3.6.1.2	type	14
3.7	Message Class Reference	15
3.7.1	Detailed Description	15
3.7.2	Constructor & Destructor Documentation	15
3.7.2.1	Message	15
3.7.2.2	Message	15
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	16
3.8.1	Detailed Description	17
3.8.2	Constructor & Destructor Documentation	17
3.8.2.1	Messenger	17
3.8.2.2	Messenger	18
3.8.2.3	~Messenger	18
3.8.3	Member Function Documentation	18
3.8.3.1	Broadcast	18
3.8.3.2	Connect	18
3.8.3.3	Disconnect	18
3.8.3.4	DisconnectClient	18
3.8.3.5	ProcessMessage	18
3.8.3.6	Receive	18
3.8.3.7	Send	18
3.9	Socket Class Reference	19
3.9.1	Detailed Description	20
3.9.2	Constructor & Destructor Documentation	20
3.9.2.1	Socket	20
3.9.2.2	Socket	20
3.9.2.3	~Socket	20
3.9.3	Member Function Documentation	20
3.9.3.1	AcceptConnections	20
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	IsWebsocket	21
3.9.3.8	Listen	21
3.9.3.9	PrepareConnection	21
3.9.3.10	SelectConnections	21
3.9.3.11	SendMessage	21
3.9.3.12	SetPort	21
3.9.3.13	SetSocketId	21
3.9.3.14	Start	21
3.9.3.15	Stop	22
3.9.4	Field Documentation	22

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

3.11.3.3	SetAllChannelsOffset	29
3.11.3.4	SetChannelOffset	29
Index		31

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Exception	5
file_header_t	8
ListenerInfo	14
Message	15
HTTPMessage	10
SocketMessage	22
Socket	19
Listener	13
FPGAHandler	9
Messenger	16
TDCConfiguration	28

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Exception	
A simple exception handler	5
file_header_t	8
FPGAHandler	9
HTTPMessage	10
Listener	13
ListenerInfo	14
Message	
Base message type	15
Messenger	16
Socket	19
SocketMessage	
Socket-passed message type	22
TDCCConfiguration	28

Chapter 3

Data Structure Documentation

3.1 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

3.1.1 Detailed Description

A simple exception handler.

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

24 Mar 2015

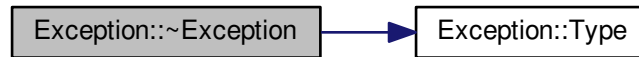
3.1.2 Constructor & Destructor Documentation

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

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

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

Here is the call graph for this function:



3.1.3 Member Function Documentation

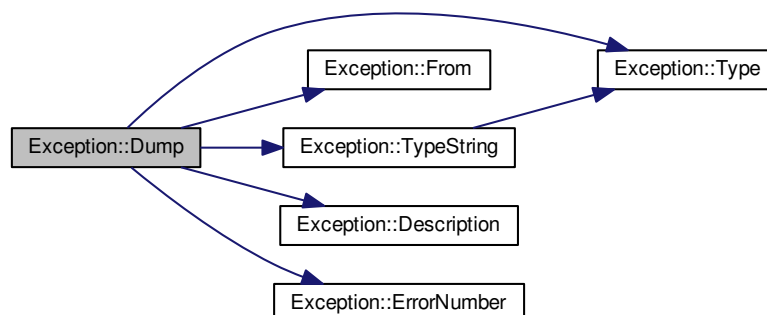
3.1.3.1 `std::string Exception::Description () const [inline]`

Here is the caller graph for this function:



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

Here is the call graph for this function:



3.1.3.3 `int Exception::ErrorNumber () const [inline]`

Here is the caller graph for this function:



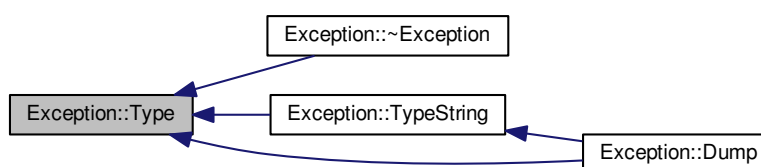
3.1.3.4 `std::string Exception::From () const [inline]`

Here is the caller graph for this function:



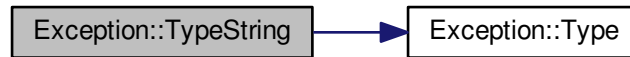
3.1.3.5 `ExceptionType Exception::Type () const [inline]`

Here is the caller graph for this function:



3.1.3.6 `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`

3.2 `file_header_t` Struct Reference

```
#include <FPGAHandler.h>
```

Data Fields

- `uint32_t magic`
- `uint32_t run_id`
- `uint32_t spill_id`

3.2.1 Field Documentation

3.2.1.1 `uint32_t file_header_t::magic`

3.2.1.2 `uint32_t file_header_t::run_id`

3.2.1.3 `uint32_t file_header_t::spill_id`

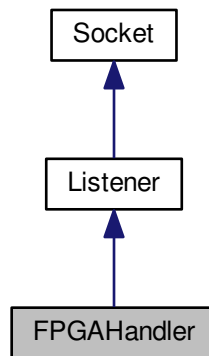
The documentation for this struct was generated from the following file:

- `include/FPGAHandler.h`

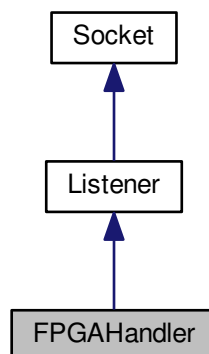
3.3 FPGAHandler Class Reference

```
#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 [TDCCConfiguration](#) &c)
- [TDCCConfiguration](#) [ReadConfiguration](#) ()
- void [ReadBuffer](#) ()

Additional Inherited Members

3.3.1 Detailed Description

Author

Laurent Forthomme laurent.forthomme@cern.ch

Date

14 Apr 2015

3.3.2 Constructor & Destructor Documentation

3.3.2.1 `FPGAHandler::FPGAHandler (int port, const char * dev)`

3.3.2.2 `virtual FPGAHandler::~~FPGAHandler ()` `[virtual]`

3.3.3 Member Function Documentation

3.3.3.1 `std::string FPGAHandler::GetFilename ()` `const` `[inline]`

3.3.3.2 `void FPGAHandler::OpenFile ()`

3.3.3.3 `void FPGAHandler::ReadBuffer ()`

3.3.3.4 `TDCCConfiguration FPGAHandler::ReadConfiguration ()`

3.3.3.5 `void FPGAHandler::SendConfiguration (const TDCCConfiguration & c)`

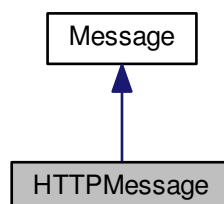
The documentation for this class was generated from the following file:

- `include/FPGAHandler.h`

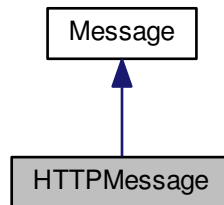
3.4 HTTPMessage Class Reference

```
#include <HTTPMessage.h>
```

Inheritance diagram for HTTPMessage:



Collaboration diagram for HTTPMessage:



Public Member Functions

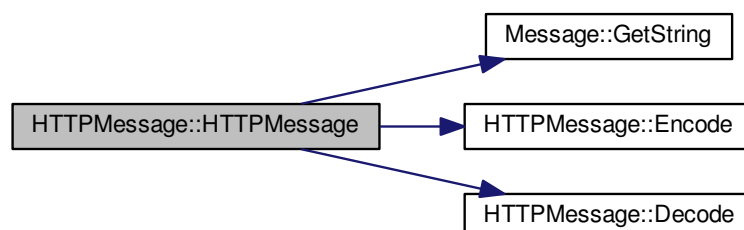
- [HTTPMessage](#) (WebSocket *ws, [Message](#) m, MessageAction a)
- [HTTPMessage](#) (WebSocket *ws, const char *msg, MessageAction a)
- void [Decode](#) ()
- void [Encode](#) ()
- MessageKey [GetKey](#) () const
- void [Dump](#) (std::ostream &os=std::cout) const

Additional Inherited Members

3.4.1 Constructor & Destructor Documentation

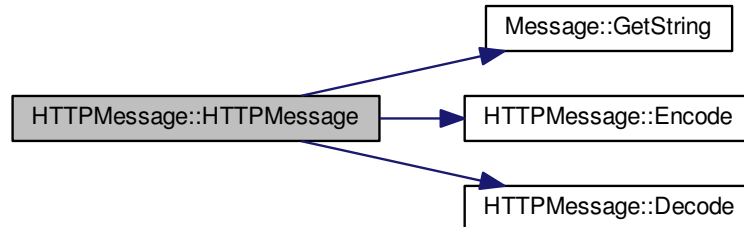
3.4.1.1 `HTTPMessage::HTTPMessage (WebSocket * ws, Message m, MessageAction a) [inline]`

Here is the call graph for this function:



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

Here is the call graph for this function:



3.4.2 Member Function Documentation

3.4.2.1 void HTTPMessage::Decode () [inline]

Here is the caller graph for this function:



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

3.4.2.3 void HTTPMessage::Encode () [inline]

Here is the caller graph for this function:



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

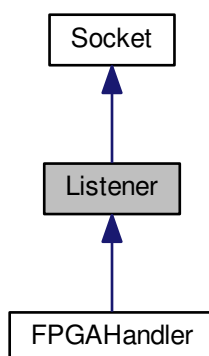
The documentation for this class was generated from the following file:

- include/HTTPMessage.h

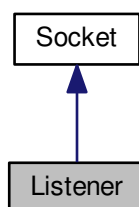
3.5 Listener Class Reference

```
#include <Listener.h>
```

Inheritance diagram for Listener:



Collaboration diagram for Listener:



Public Member Functions

- [Listener](#) ()
- [Listener](#) (int port)
- [~Listener](#) ()
- bool [Connect](#) ()
- void [Disconnect](#) ()
- void [Send](#) (const [Message](#) &m) const
- void [Receive](#) ()
- virtual void [ParseMessage](#) (const [SocketMessage](#) &m)

Additional Inherited Members

3.5.1 Detailed Description

Listener/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

3.5.2 Constructor & Destructor Documentation

3.5.2.1 Listener::Listener () `[inline]`

3.5.2.2 Listener::Listener (int *port*)

3.5.2.3 Listener::~Listener ()

3.5.3 Member Function Documentation

3.5.3.1 bool Listener::Connect ()

3.5.3.2 void Listener::Disconnect ()

3.5.3.3 virtual void Listener::ParseMessage (const SocketMessage & *m*) `[inline]`, `[virtual]`

3.5.3.4 void Listener::Receive ()

3.5.3.5 void Listener::Send (const Message & *m*) const

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

- include/Listener.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:

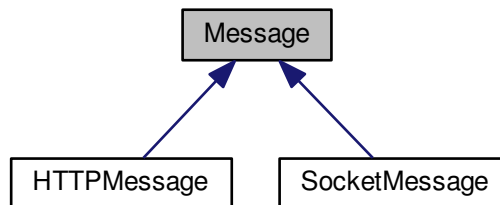
- include/Messenger.h

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]

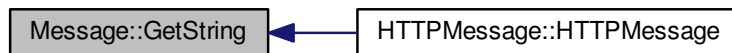
3.7.3 Member Function Documentation

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]

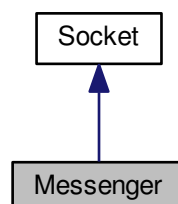
The documentation for this class was generated from the following file:

- include/Message.h

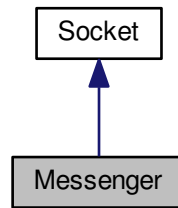
3.8 Messenger Class Reference

```
#include <Messenger.h>
```

Inheritance diagram for Messenger:



Collaboration diagram for Messenger:



Public Member Functions

- [Messenger](#) ()
- [Messenger](#) (int port)
- [~Messenger](#) ()
- bool [Connect](#) ()
- void [Disconnect](#) ()
- void [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::Broadcast** (const **Message** & *m*) const

Emit a message to all clients connected through the socket.

Parameters

in	<i>m</i>	Message to transmit
----	----------	-------------------------------------

3.8.3.2 **bool Messenger::Connect** ()

3.8.3.3 **void Messenger::Disconnect** ()

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

Disconnect a client.

Ask to a client to disconnect from this socket

Parameters

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

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

Process a message received from the socket.

Parameters

in	<i>Unique</i>	identifier of the client sending the message
----	---------------	--

3.8.3.6 **MessageKey Messenger::Receive** ()

Handle a message reception from a client.

Returns

The key to the message received if successfully parsed

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

Send any type of message to any client.

Parameters

in	<i>m</i>	Message to transmit
in	<i>sid</i>	Unique identifier of the client on this socket

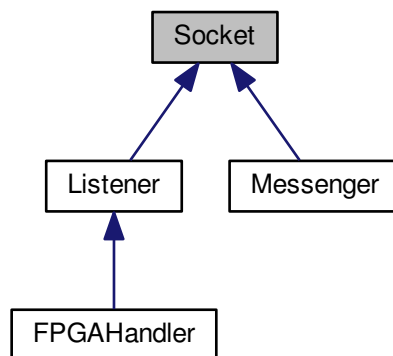
The documentation for this class was generated from the following file:

- 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
- 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

<code>in, out</code>	<code>socket</code>	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 `bool Socket::IsWebSocket (int sid)` `const` `[inline]`

3.9.3.8 `void Socket::Listen (int maxconn)` `[protected]`

Listen to incoming messages.

Set the socket to listen to any message coming from outside

3.9.3.9 `void Socket::PrepareConnection ()` `[protected]`

3.9.3.10 `void Socket::SelectConnections ()`

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

3.9.3.11 `void Socket::SendMessage (Message message, int id = -1)` `const` `[protected]`

Send a message on a socket.

3.9.3.12 `void Socket::SetPort (int port)` `[inline]`

3.9.3.13 `void Socket::SetSocketId (int sid)` `[inline]`

3.9.3.14 `bool Socket::Start ()` `[protected]`

Start the socket.

Launch all mandatory operations to set the socket to be used

Returns

Success of the operation

3.9.3.15 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:

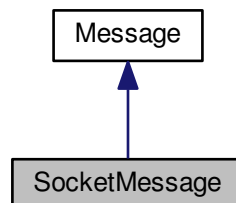
- include/Socket.h

3.10 SocketMessage Class Reference

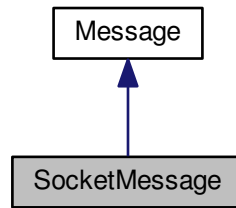
Socket-passed message type.

```
#include <SocketMessage.h>
```

Inheritance diagram for SocketMessage:



Collaboration diagram for SocketMessage:



Public Member Functions

- [SocketMessage](#) ()
- [SocketMessage](#) ([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 (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:



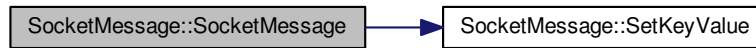
3.10.2.7 `SocketMessage::SocketMessage (MessageKey key, std::string value)` [inline]

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:



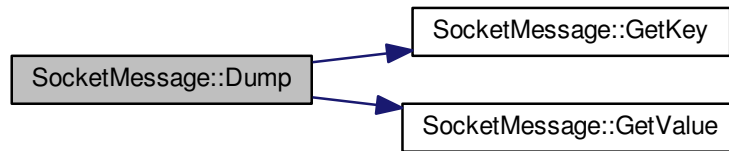
3.10.2.11 SocketMessage::SocketMessage (MessageMap *msg_m*) [inline]

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]`

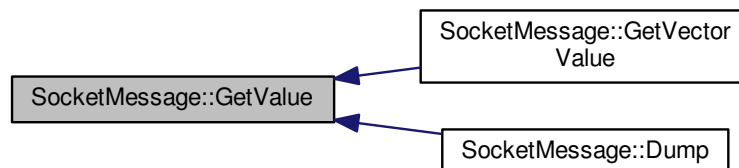
Here is the caller graph for this function:



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

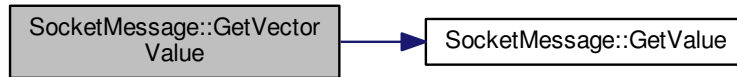
3.10.3.5 `std::string SocketMessage::GetValue () const [inline]`

Here is the caller graph for this function:



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

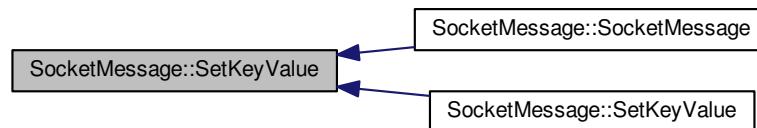
Here is the call graph for this function:



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 TDCCConfiguration::TDCCConfiguration ()

3.11.2.2 virtual TDCCConfiguration::~~TDCCConfiguration () `[inline]`, `[virtual]`

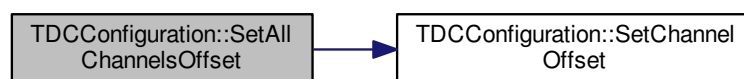
3.11.3 Member Function Documentation

3.11.3.1 void TDCCConfiguration::Dump () const

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

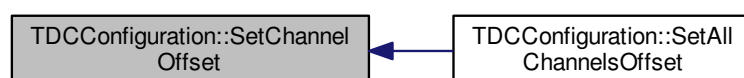
3.11.3.3 void TDCCConfiguration::SetAllChannelsOffset (short *offset*) `[inline]`

Here is the call graph for this function:



3.11.3.4 void TDCCConfiguration::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

- ~Exception
 - Exception, [5](#)
- ~FPGAHandler
 - FPGAHandler, [10](#)
- ~Listener
 - Listener, [14](#)
- ~Message
 - Message, [16](#)
- ~Messenger
 - Messenger, [18](#)
- ~Socket
 - Socket, [20](#)
- ~SocketMessage
 - SocketMessage, [25](#)
- ~TDCConfiguration
 - TDCConfiguration, [29](#)
- AcceptConnections
 - Socket, [20](#)
- Bind
 - Socket, [21](#)
- Broadcast
 - Messenger, [18](#)
- Connect
 - Listener, [14](#)
 - Messenger, [18](#)
- Decode
 - HTTPMessage, [12](#)
- Description
 - Exception, [6](#)
- Disconnect
 - Listener, [14](#)
 - Messenger, [18](#)
- DisconnectClient
 - Messenger, [18](#)
- Dump
 - Exception, [6](#)
 - HTTPMessage, [12](#)
 - Message, [16](#)
 - SocketMessage, [25](#)
 - TDCConfiguration, [29](#)
- DumpConnected
 - Socket, [21](#)
- Encode
 - HTTPMessage, [12](#)
- ErrorNumber
 - Exception, [6](#)

- Exception, [5](#)
 - ~Exception, [5](#)
 - Description, [6](#)
 - Dump, [6](#)
 - ErrorNumber, [6](#)
 - Exception, [5](#)
 - From, [7](#)
 - Type, [7](#)
 - TypeString, [7](#)
- fBuffer
 - Socket, [22](#)
- fMaster
 - Socket, [22](#)
- FPGAHandler, [9](#)
 - ~FPGAHandler, [10](#)
 - FPGAHandler, [10](#)
 - GetFilename, [10](#)
 - OpenFile, [10](#)
 - ReadBuffer, [10](#)
 - ReadConfiguration, [10](#)
 - SendConfiguration, [10](#)
- fPort
 - Socket, [22](#)
- fReadFds
 - Socket, [22](#)
- fSocketsConnected
 - Socket, [22](#)
- fString
 - Message, [16](#)
- FetchMessage
 - Socket, [21](#)
- file_header_t, [8](#)
 - magic, [8](#)
 - run_id, [8](#)
 - spill_id, [8](#)
- From
 - Exception, [7](#)
- GetChannelOffset
 - TDCConfiguration, [29](#)
- GetFilename
 - FPGAHandler, [10](#)
- GetIntValue
 - SocketMessage, [26](#)
- GetKey
 - HTTPMessage, [12](#)
 - Message, [16](#)
 - SocketMessage, [26](#)
- GetPort

- Socket, 21
- GetSocketId
 - Socket, 21
- GetString
 - Message, 16
 - SocketMessage, 26
- GetValue
 - SocketMessage, 26
- GetVectorValue
 - SocketMessage, 26
- HTTPMessage, 10
 - Decode, 12
 - Dump, 12
 - Encode, 12
 - GetKey, 12
 - HTTPMessage, 11
- IsFromWeb
 - Message, 16
- IsWebSocket
 - Socket, 21
- Listen
 - Socket, 21
- Listener, 13
 - ~Listener, 14
 - Connect, 14
 - Disconnect, 14
 - Listener, 14
 - ParseMessage, 14
 - Receive, 14
 - Send, 14
- ListenerInfo, 14
 - name, 14
 - type, 14
- magic
 - file_header_t, 8
- Message, 15
 - ~Message, 16
 - Dump, 16
 - fString, 16
 - GetKey, 16
 - GetString, 16
 - IsFromWeb, 16
 - Message, 15
- Messenger, 16
 - ~Messenger, 18
 - Broadcast, 18
 - Connect, 18
 - Disconnect, 18
 - DisconnectClient, 18
 - Messenger, 17
 - ProcessMessage, 18
 - Receive, 18
 - Send, 18
- name
 - ListenerInfo, 14
- OpenFile
 - FPGAHandler, 10
- ParseMessage
 - Listener, 14
- PrepareConnection
 - Socket, 21
- ProcessMessage
 - Messenger, 18
- ReadBuffer
 - FPGAHandler, 10
- ReadConfiguration
 - FPGAHandler, 10
- Receive
 - Listener, 14
 - Messenger, 18
- run_id
 - file_header_t, 8
- SelectConnections
 - Socket, 21
- Send
 - Listener, 14
 - Messenger, 18
- SendConfiguration
 - FPGAHandler, 10
- SendMessage
 - Socket, 21
- SetAllChannelsOffset
 - TDCConfiguration, 29
- SetChannelOffset
 - TDCConfiguration, 29
- SetKeyValue
 - SocketMessage, 27, 28
- SetPort
 - Socket, 21
- SetSocketId
 - Socket, 21
- Socket, 19
 - ~Socket, 20
 - AcceptConnections, 20
 - Bind, 21
 - DumpConnected, 21
 - fBuffer, 22
 - fMaster, 22
 - fPort, 22
 - fReadFds, 22
 - fSocketsConnected, 22
 - FetchMessage, 21
 - GetPort, 21
 - GetSocketId, 21
 - IsWebSocket, 21
 - Listen, 21
 - PrepareConnection, 21
 - SelectConnections, 21
 - SendMessage, 21

- SetPort, [21](#)
- SetSocketId, [21](#)
- Socket, [20](#)
- Start, [21](#)
- Stop, [22](#)
- SocketMessage, [22](#)
 - ~SocketMessage, [25](#)
 - Dump, [25](#)
 - GetIntValue, [26](#)
 - GetKey, [26](#)
 - GetString, [26](#)
 - GetValue, [26](#)
 - GetVectorValue, [26](#)
 - SetKeyValue, [27](#), [28](#)
 - SocketMessage, [24](#), [25](#)
- spill_id
 - file_header_t, [8](#)
- Start
 - Socket, [21](#)
- Stop
 - Socket, [22](#)
- TDCCConfiguration, [28](#)
 - ~TDCCConfiguration, [29](#)
 - Dump, [29](#)
 - GetChannelOffset, [29](#)
 - SetAllChannelsOffset, [29](#)
 - SetChannelOffset, [29](#)
 - TDCCConfiguration, [29](#)
- Type
 - Exception, [7](#)
- type
 - ListenerInfo, [14](#)
- TypeString
 - Exception, [7](#)