

DOCUMENT TYPE:	TECHNICAL NOTE		
TITLE:	PacketLib 1.2.3 Detailed Design Report		
DOCUMENT Ref. No.:	AGILE-ITE-SD-002	N° OF PAGES:	i-iii, 348
	IASF section of Bologna Report 350/02		
ISSUE No.:	02	DATE:	May 2003
PREPARED BY:	A. BULGARELLI, F. GIANOTTI, M. TRIFOGLIO		
CHECKED BY:	M. TRIFOGLIO		
SUBSYSTEM MANAGER:	M. TRIFOGLIO		

APPROVED BY:

SUBSYSTEM LEADER:	G. DI COCCO	DATE:
PROJECT LEADER:	M. TAVANI	DATE:
PAYLOAD MANAGER:	A. ZAMBRA	DATE:
PAPM:	A. BERNABEO	DATE:
CONFIGURATION:	C. MANGILI	DATE:

AGILE

Ref: AGILE-ITE-SD-002
Project Ref.: AGILE
Issue: 1 Page: iii
Date: 7-May-03

DISTRIBUTION LIST

CHANGE RECORD

Contents

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Compound Index	5
3.1 Compound List	5
4 File Index	9
4.1 File List	9
5 Namespace Documentation	13
5.1 PacketLib Namespace Reference	13
5.2 std Namespace Reference	19
6 Class Documentation	21
6.1 PacketLib::ByteStream Class Reference	21
6.2 PacketLib::ConfigurationFile Class Reference	30
6.3 PacketLib::DataFieldHeader Class Reference	33
6.4 PacketLib::Device Class Reference	34
6.5 PacketLib::Field Class Reference	37
6.6 PacketLib::File Class Reference	40
6.7 PacketLib::FileStream Class Reference	47
6.8 PacketLib::FileStreamPointer Class Reference	50
6.9 PacketLib::Input Class Reference	52
6.10 PacketLib::InputFile Class Reference	55

6.11	PacketLib::InputPacketStream Class Reference	58
6.12	PacketLib::InputPacketStreamFile Class Reference	62
6.13	PacketLib::InputSocketServer Class Reference	68
6.14	PacketLib::InputText Class Reference	72
6.15	PacketLib::InputTextFile Class Reference	75
6.16	PacketLib::MemoryBuffer Class Reference	79
6.17	PacketLib::mesgbuf Struct Reference	83
6.18	PacketLib::MSGQ Class Reference	84
6.19	PacketLib::Output Class Reference	88
6.20	PacketLib::OutputFile Class Reference	91
6.21	PacketLib::OutputPacketStream Class Reference	94
6.22	PacketLib::OutputSocketClient Class Reference	97
6.23	PacketLib::Packet Class Reference	100
6.24	PacketLib::PacketDataField Class Reference	113
6.25	PacketLib::PacketException Class Reference	117
6.26	PacketLib::PacketExceptionFormat Class Reference	120
6.27	PacketLib::PacketExceptionIO Class Reference	122
6.28	PacketLib::PacketHeader Class Reference	124
6.29	PacketLib::PacketIdentifier Class Reference	127
6.30	PacketLib::PacketNotRecognized Class Reference	129
6.31	PacketLib::PacketStream Class Reference	132
6.32	PacketLib::PartOfPacket Class Reference	138
6.33	PacketLib::SDFBFBlock Class Reference	145
6.34	PacketLib::SDFBFixed Class Reference	146
6.35	PacketLib::SDFBVariable Class Reference	154
6.36	PacketLib::SDFBVBlock Class Reference	163
6.37	PacketLib::SDFBVBlockFixed Class Reference	167
6.38	PacketLib::SDFBVBlockVariable Class Reference	170
6.39	PacketLib::SDFNBVariable Class Reference	172
6.40	PacketLib::SDFNBFixed Class Reference	174
6.41	PacketLib::SDFNoBlockFixed Class Reference	175
6.42	PacketLib::SDFNoBlockVariable Class Reference	179
6.43	PacketLib::SHM Class Reference	184
6.44	PacketLib::Socket Class Reference	188

6.45	PacketLib::SocketClient Class Reference	191
6.46	PacketLib::SocketServer Class Reference	193
6.47	PacketLib::SourceDataField Class Reference	196
6.48	PacketLib::Utility Class Reference	207
7	File Documentation	211
7.1	ByteStream.cpp File Reference	211
7.2	ByteStream.h File Reference	212
7.3	ConfigurationFile.cpp File Reference	213
7.4	ConfigurationFile.h File Reference	214
7.5	DataFieldHeader.cpp File Reference	216
7.6	DataFieldHeader.h File Reference	217
7.7	Device.cpp File Reference	219
7.8	Device.h File Reference	220
7.9	Field.cpp File Reference	221
7.10	Field.h File Reference	222
7.11	File.cpp File Reference	223
7.12	File.h File Reference	224
7.13	FileStream.cpp File Reference	225
7.14	FileStream.h File Reference	226
7.15	FileStreamPointer.cpp File Reference	227
7.16	FileStreamPointer.h File Reference	228
7.17	Input.cpp File Reference	229
7.18	Input.h File Reference	230
7.19	InputFile.cpp File Reference	231
7.20	InputFile.h File Reference	232
7.21	InputPacketStream.cpp File Reference	233
7.22	InputPacketStream.h File Reference	234
7.23	InputPacketStreamFile.cpp File Reference	235
7.24	InputPacketStreamFile.h File Reference	236
7.25	InputSocketServer.cpp File Reference	237
7.26	InputSocketServer.h File Reference	238
7.27	InputText.cpp File Reference	239
7.28	InputText.h File Reference	240

7.29 InputTextFile.cpp File Reference	241
7.30 InputTextFile.h File Reference	242
7.31 MemoryBuffer.cpp File Reference	243
7.32 MemoryBuffer.h File Reference	244
7.33 MSGQ.cpp File Reference	245
7.34 MSGQ.h File Reference	246
7.35 Output.cpp File Reference	247
7.36 Output.h File Reference	248
7.37 OutputFile.cpp File Reference	249
7.38 OutputFile.h File Reference	250
7.39 OutputPacketStream.cpp File Reference	251
7.40 OutputPacketStream.h File Reference	252
7.41 OutputSocketClient.cpp File Reference	253
7.42 OutputSocketClient.h File Reference	254
7.43 Packet.cpp File Reference	255
7.44 Packet.h File Reference	256
7.45 PacketDataField.cpp File Reference	258
7.46 PacketDataField.h File Reference	259
7.47 PacketException.cpp File Reference	261
7.48 PacketException.h File Reference	262
7.49 PacketExceptionFormatException.cpp File Reference	263
7.50 PacketExceptionFormatException.h File Reference	264
7.51 PacketExceptionIO.cpp File Reference	265
7.52 PacketExceptionIO.h File Reference	266
7.53 PacketHeader.cpp File Reference	268
7.54 PacketHeader.h File Reference	269
7.55 PacketIdentifier.cpp File Reference	270
7.56 PacketIdentifier.h File Reference	271
7.57 PacketLibDefinition.h File Reference	272
7.58 PacketNotRecognized.cpp File Reference	275
7.59 PacketNotRecognized.h File Reference	276
7.60 PacketStream.cpp File Reference	277
7.61 PacketStream.h File Reference	278
7.62 PartOfPacket.cpp File Reference	279

7.63	PartOfPacket.h File Reference	280
7.64	SDFBFBlock.cpp File Reference	282
7.65	SDFBFBlock.h File Reference	283
7.66	SDFBlockFixed.cpp File Reference	285
7.67	SDFBlockFixed.h File Reference	286
7.68	SDFBlockVariable.cpp File Reference	288
7.69	SDFBlockVariable.h File Reference	289
7.70	SDFBVBlock.cpp File Reference	291
7.71	SDFBVBlock.h File Reference	292
7.72	SDFBVBlockFixed.cpp File Reference	294
7.73	SDFBVBlockFixed.h File Reference	295
7.74	SDFBVBlockVariable.cpp File Reference	297
7.75	SDFBVBlockVariable.h File Reference	298
7.76	SDFNBVariable.cpp File Reference	300
7.77	SDFNBVariable.h File Reference	301
7.78	SDFNBVFixed.cpp File Reference	303
7.79	SDFNBVFixed.h File Reference	304
7.80	SDFNoBlockFixed.cpp File Reference	306
7.81	SDFNoBlockFixed.h File Reference	307
7.82	SDFNoBlockVariable.cpp File Reference	309
7.83	SDFNoBlockVariable.h File Reference	310
7.84	SHM.cpp File Reference	312
7.85	SHM.h File Reference	313
7.86	Socket.cpp File Reference	314
7.87	Socket.h File Reference	315
7.88	SocketClient.cpp File Reference	317
7.89	SocketClient.h File Reference	318
7.90	SocketServer.cpp File Reference	319
7.91	SocketServer.h File Reference	320
7.92	SourceDataField.cpp File Reference	321
7.93	SourceDataField.h File Reference	322
7.94	Utility.cpp File Reference	323
7.95	Utility.h File Reference	324

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

PacketLib	13
std	19

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

PacketLib::ByteStream	21
PacketLib::Device	34
PacketLib::File	40
PacketLib::FileStream	47
PacketLib::MSGQ	84
PacketLib::SHM	184
PacketLib::Socket	188
PacketLib::SocketClient	191
PacketLib::SocketServer	193
PacketLib::Field	37
PacketLib::FileStreamPointer	50
PacketLib::Input	52
PacketLib::InputFile	55
PacketLib::InputSocketServer	68
PacketLib::InputText	72
PacketLib::InputTextFile	75
PacketLib::ConfigurationFile	30
PacketLib::MemoryBuffer	79
PacketLib::mesgbuf	83
PacketLib::Output	88
PacketLib::OutputFile	91
PacketLib::OutputSocketClient	97
PacketLib::Packet	100
PacketLib::PacketNotRecognized	129
PacketLib::PacketException	117
PacketLib::PacketExceptionFormat	120
PacketLib::PacketExceptionIO	122

PacketLib::PacketIdentifier	127
PacketLib::PacketStream	132
PacketLib::InputPacketStream	58
PacketLib::InputPacketStreamFile	62
PacketLib::OutputPacketStream	94
PacketLib::PartOfPacket	138
PacketLib::DataFieldHeader	33
PacketLib::PacketDataField	113
PacketLib::PacketHeader	124
PacketLib::SDFBFBlock	145
PacketLib::SDFBVBlock	163
PacketLib::SDFBVBlockFixed	167
PacketLib::SDFBVBlockVariable	170
PacketLib::SDFNBVariable	172
PacketLib::SDFNBVFixed	174
PacketLib::SourceDataField	196
PacketLib::SDFBlockFixed	146
PacketLib::SDFBlockVariable	154
PacketLib::SDFNoBlockFixed	175
PacketLib::SDFNoBlockVariable	179
PacketLib::Utility	207

Chapter 3

Compound Index

3.1 Compound List

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

PacketLib::ByteStream (Represent a stream of byte)	21
PacketLib::ConfigurationFile (This class represent the text file which contains the configurations)	30
PacketLib::DataFieldHeader (This class represent the data field header of a packet)	33
PacketLib::Device (This class represent a Device (p. 34) for IO sys- tem)	34
PacketLib::Field (This is a single field of a packet)	37
PacketLib::File (This class represent a device file for IO system) . .	40
PacketLib::FileStream (This class represent a stream of byte into a file)	47
PacketLib::FileStreamPointer (This class represent a pointer into a file of byte)	50
PacketLib::Input (This class represent a generic input for IO system)	52
PacketLib::InputFile (This class represent a File (p. 40) input for IO system)	55
PacketLib::InputPacketStream (This class represents the stream of byte in input that contains the packets. The input object must be deleted by its creator)	58
PacketLib::InputPacketStreamFile (This class represents the stream of byte into a file that contains the packets)	62
PacketLib::InputSocketServer (This class represent a Socket (p. 188) Server input for IO system)	68
PacketLib::InputText (This class represents generic type of input. The input must be a text input. This class is used for the configuration files)	72
PacketLib::InputTextFile (This class represents generic type of in- put. The input must be a input text file)	75

PacketLib::MemoryBuffer (Class that represent an FIFO structure of char*)	79
PacketLib::mesgbuf	83
PacketLib::MSGQ (Represent a shared queue)	84
PacketLib::Output (This class represents generic type of output)	88
PacketLib::OutputFile (This class represents generic type of output. The output must be a file)	91
PacketLib::OutputPacketStream (This class represents generic output of stream if packet)	94
PacketLib::OutputSocketClient (This class represents generic type of output. The output must be a file)	97
PacketLib::Packet (Represent a single packet)	100
PacketLib::PacketDataField (Represent the data field of the packet)	113
PacketLib::PacketException (Exception class of the PacketLib (p. 13))	117
PacketLib::PacketExceptionFileFormat (Exception class of the PacketLib (p. 13). The error is the bad file format of the configuration files)	120
PacketLib::PacketExceptionIO (Exception class of the PacketLib (p. 13). The error is an IO error)	122
PacketLib::PacketHeader (Header of packet)	124
PacketLib::PacketIdentifier (Identifier of packet)	127
PacketLib::PacketNotRecognized (Packet (p. 100) not recognized)	129
PacketLib::PacketStream (Stream of byte with packets)	132
PacketLib::PartOfPacket (Single part of packet)	138
PacketLib::SDFBFBlock (Represent a single block with the number of fields fixed)	145
PacketLib::SDFBFixed (Represents a source data field structured with blocks but the number of fields for each block are fixed)	146
PacketLib::SDFBVariable (Represents a source data field structered with blocks and with the number of fields for each block variable)	154
PacketLib::SDFBVBlock (Class wich represents a single block with a fixed part and with a variable part)	163
PacketLib::SDFBVBlockFixed (Class wich represent the part of block with the number of field fixed)	167
PacketLib::SDFBVBlockVariable (Class which represents the part of block repeated because the number of fields are variable) .	170
PacketLib::SDFNBVariable (Class which represents the variable part (the fields (or elements) wich are repeated) of source data field)	172
PacketLib::SDFNBFixed (Represents the fixed part of source data field)	174
PacketLib::SDFNoBlockFixed (Represents a source data field of packet where the structure is noblock and with a fixed dimension)	175
PacketLib::SDFNoBlockVariable (Represents a source data field with no block but with the number of fields variable)	179
PacketLib::SHM (This class represents generic SHM (p.184))	184

PacketLib::Socket (This class represents generic socket)	188
PacketLib::SocketClient (This class represents generic client socket)	191
PacketLib::SocketServer (This class represents generic client socket)	193
PacketLib::SourceDataField (A generic source data field)	196
PacketLib::Utility (A class with static method with common functionality)	207

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

ByteStream.cpp	211
ByteStream.h	212
ConfigurationFile.cpp	213
ConfigurationFile.h	214
DataFieldHeader.cpp	216
DataFieldHeader.h	217
Device.cpp	219
Device.h	220
Field.cpp	221
Field.h	222
File.cpp	223
File.h	224
FileStream.cpp	225
FileStream.h	226
FileStreamPointer.cpp	227
FileStreamPointer.h	228
Input.cpp	229
Input.h	230
InputFile.cpp	231
InputFile.h	232
InputPacketStream.cpp	233
InputPacketStream.h	234
InputPacketStreamFile.cpp	235
InputPacketStreamFile.h	236
InputSocketServer.cpp	237
InputSocketServer.h	238
InputText.cpp	239
InputText.h	240
InputTextFile.cpp	241

InputTextFile.h	242
MemoryBuffer.cpp	243
MemoryBuffer.h	244
MSGQ.cpp	245
MSGQ.h	246
Output.cpp	247
Output.h	248
OutputFile.cpp	249
OutputFile.h	250
OutputPacketStream.cpp	251
OutputPacketStream.h	252
OutputSocketClient.cpp	253
OutputSocketClient.h	254
Packet.cpp	255
Packet.h	256
PacketDataField.cpp	258
PacketDataField.h	259
PacketException.cpp	261
PacketException.h	262
PacketExceptionFormatException.cpp	263
PacketExceptionFormatException.h	264
PacketExceptionIO.cpp	265
PacketExceptionIO.h	266
PacketHeader.cpp	268
PacketHeader.h	269
PacketIdentifier.cpp	270
PacketIdentifier.h	271
PacketLibDefinition.h	272
PacketNotRecognized.cpp	275
PacketNotRecognized.h	276
PacketStream.cpp	277
PacketStream.h	278
PartOfPacket.cpp	279
PartOfPacket.h	280
SDFBFBlock.cpp	282
SDFBFBlock.h	283
SDFBlockFixed.cpp	285
SDFBlockFixed.h	286
SDFBlockVariable.cpp	288
SDFBlockVariable.h	289
SDFBVBlock.cpp	291
SDFBVBlock.h	292
SDFBVBlockFixed.cpp	294
SDFBVBlockFixed.h	295
SDFBVBlockVariable.cpp	297
SDFBVBlockVariable.h	298
SDFNBVariable.cpp	300
SDFNBVariable.h	301
SDFNBFixed.cpp	303
SDFNBFixed.h	304

SDFNoBlockFixed.cpp	306
SDFNoBlockFixed.h	307
SDFNoBlockVariable.cpp	309
SDFNoBlockVariable.h	310
SHM.cpp	312
SHM.h	313
Socket.cpp	314
Socket.h	315
SocketClient.cpp	317
SocketClient.h	318
SocketServer.cpp	319
SocketServer.h	320
SourceDataField.cpp	321
SourceDataField.h	322
Utility.cpp	323
Utility.h	324

Chapter 5

Namespace Documentation

5.1 PacketLib Namespace Reference

Compounds

- class **PacketLib::ByteStream**

Represent a stream of byte.
 - class **PacketLib::ConfigurationFile**

This class represent the text file which contains the configurations.
 - class **PacketLib::DataFieldHeader**

This class represent the data field header of a packet.
 - class **PacketLib::Device**

This class represent a Device (p. 34) for IO system.
 - class **PacketLib::Field**

This is a single field of a packet.
 - class **PacketLib::File**

This class represent a device file for IO system.
 - class **PacketLib::FileStream**

This class represent a stream of byte into a file.
 - class **PacketLib::FileStreamPointer**

This class represent a pointer into a file of byte.
 - class **PacketLib::Input**

This class represent a generic input for IO system.
-

- class **PacketLib::InputFile**

This class represent a File (p. 40) input for IO system.
- class **PacketLib::InputPacketStream**

This class represents the stream of byte in input that contains the packets. The input object must be deleted by its creator.
- class **PacketLib::InputPacketStreamFile**

This class represents the stream of byte into a file that contains the packets.
- class **PacketLib::InputSocketServer**

This class represent a Socket (p. 188) Server input for IO system.
- class **PacketLib::InputText**

This class represents generic type of input. The input must be a text input. This class is used for the configuration files.
- class **PacketLib::InputTextFile**

This class represents generic type of input. The input must be a input text file.
- class **PacketLib::MemoryBuffer**

Class that represent an FIFO structure of char.*
- struct **PacketLib::mesgbuf**
- class **PacketLib::MSGQ**

Represent a shared queue.
- class **PacketLib::Output**

This class represents generic type of output.
- class **PacketLib::OutputFile**

This class represents generic type of output. The output must be a file.
- class **PacketLib::OutputPacketStream**

This class represents generic output of stream if packet.
- class **PacketLib::OutputSocketClient**

This class represents generic type of output. The output must be a file.
- class **PacketLib::Packet**

Represent a single packet.
- class **PacketLib::PacketDataField**

Represent the data field of the packet.
- class **PacketLib::PacketException**

*Exception class of the **PacketLib** (p. 13).*

- class **PacketLib::PacketExceptionFormat**

*Exception class of the **PacketLib** (p. 13). The error is the bad file format of the configuration files.*

- class **PacketLib::PacketExceptionIO**

*Exception class of the **PacketLib** (p. 13). The error is an IO error.*

- class **PacketLib::PacketHeader**

Header of packet.

- class **PacketLib::PacketIdentifier**

Identifier of packet.

- class **PacketLib::PacketNotRecognized**

Packet (p. 100) not recognized.

- class **PacketLib::PacketStream**

Stream of byte with packets.

- class **PacketLib::PartOfPacket**

Single part of packet.

- class **PacketLib::SDFBFBlock**

Represent a single block with the number of fields fixed.

- class **PacketLib::SDFBlockFixed**

Represents a source data field structured with blocks but the number of fields for each block are fixed.

- class **PacketLib::SDFBlockVariable**

Represents a source data field structured with blocks and with the number of fields for each block variable.

- class **PacketLib::SDFBVBlock**

Class which represents a single block with a fixed part and with a variable part.

- class **PacketLib::SDFBVBlockFixed**

Class which represent the part of block with the number of field fixed.

- class **PacketLib::SDFBVBlockVariable**

Class which represents the part of block repeated because the number of fields are variable.

- class **PacketLib::SDFNBVariable**

Class which represents the variable part (the fields (or elements) which are repeated) of source data field.

- class **PacketLib::SDFNBVFixed**

Represents the fixed part of source data field.

- class **PacketLib::SDFNoBlockFixed**

Represents a source data field of packet where the structure is noblock and with a fixed dimension.

- class **PacketLib::SDFNoBlockVariable**

Represents a source data field with no block but with the number of fields variable.

- class **PacketLib::SHM**

This class represents generic SHM (p. 184).

- class **PacketLib::Socket**

This class represents generic socket.

- class **PacketLib::SocketClient**

This class represents generic client socket.

- class **PacketLib::SocketServer**

This class represents generic client socket.

- class **PacketLib::SourceDataField**

A generic source data field.

- class **PacketLib::Utility**

A class with static method with common functionality.

Typedefs

- typedef **PacketLib::mesgbuf msgbuf**
- typedef unsigned char **byte**
- typedef unsigned short **word**
- typedef unsigned long **dword**

5.1.1 Type Definition Documentation

5.1.1.1 **typedef unsigned char PacketLib::byte**

Definition at line 45 of file PacketLibDefinition.h.

Referenced by PacketLib::ByteStream::ByteStream, PacketLib::Utility::byteToHexadecimal, PacketLib::File::find, PacketLib::PartOfPacket::generateStream, PacketLib::ByteStream::getByte, PacketLib::File::getNByte, PacketLib::ByteStream::getOutputStream, PacketLib::Packet::getPacketID, PacketLib::ByteStream::getStream, PacketLib::ByteStream::getSubByteStreamCopy, PacketLib::ByteStream::getValue, PacketLib::Packet::loadIdentifiers, PacketLib::SHM::open, PacketLib::PacketIdentifier::PacketIdentifier, PacketLib::Socket::recv, PacketLib::Socket::send, PacketLib::PartOfPacket::setByteStream, PacketLib::Packet::setPacketID, PacketLib::ByteStream::setStream, PacketLib::ByteStream::setStreamCopy, PacketLib::ByteStream::setWord, PacketLib::Utility::stringToHexadecimal, PacketLib::ByteStream::swap, PacketLib::Utility::wordToBinary, PacketLib::File::writeByteStream, and PacketLib::SHM::writeSlot.

5.1.1.2 **typedef unsigned long PacketLib::dword**

Definition at line 51 of file PacketLibDefinition.h.

Referenced by PacketLib::SHM::create, PacketLib::Field::Field, PacketLib::Utility::format_output, PacketLib::MemoryBuffer::getbuffer, PacketLib::MemoryBuffer::getBufferDimension, PacketLib::MemoryBuffer::MemoryBuffer, PacketLib::SHM::open, PacketLib::SHM::readSlot, PacketLib::MemoryBuffer::setbuffer, PacketLib::Utility::wordToBinary2, PacketLib::SHM::writeSlot, and PacketLib::MemoryBuffer::~MemoryBuffer.

5.1.1.3 **typedef struct PacketLib::mesgbuf PacketLib::msgbuf**

5.1.1.4 **typedef unsigned short PacketLib::word**

Definition at line 48 of file PacketLibDefinition.h.

Referenced by PacketLib::PacketNotRecognized::createPacketType, PacketLib::Packet::createPacketType, PacketLib::FileStream::FileStream, PacketLib::InputPacketStreamFile::freeRun, PacketLib::Utility::ftimeToChar, PacketLib::SDFBVBlock::generateStream, PacketLib::SDFBlockVariable::generateStream, PacketLib::SDFBlockFixed::generateStream, PacketLib::PartOfPacket::generateStream, PacketLib::Packet::generateStream, PacketLib::Utility::getbits, PacketLib::Utility::getbits2, PacketLib::SourceDataField::getDimension, PacketLib::SDFNoBlockVariable::getDimension, PacketLib::SDFBVBlock::getDimension, PacketLib::SDFBlockVariable::getDimension, PacketLib::SDFBlockFixed::getDimension, PacketLib::PartOfPacket::getDimension, PacketLib::PacketDataField::getDimension, PacketLib::Packet::getDimension, PacketLib::SourceDataField::getFields, PacketLib::SDFNoBlockVariable::getFields, PacketLib::SDFBVBlock::getFields, PacketLib::SDFBlockVariable::getFields, PacketLib::SDFBlockFixed::getFields, PacketLib::PartOfPacket::getFields, PacketLib::SourceDataField::getFieldValue, PacketLib::SDFNoBlockVariable::getFieldValue, PacketLib::SDFNoBlockFixed::getFieldValue, PacketLib::SDFBVBlock::getFieldValue, PacketLib::SDFBlockVariable::get-

FieldValue, PacketLib::SDFBlockFixed::getFieldValue, PacketLib::PartOfPacket::getFieldValue, PacketLib::PacketStream::getHeaderDimension, PacketLib::SourceDataField::getIndexOfNBlock, PacketLib::SourceDataField::getMaxDimension, PacketLib::SDFNoBlockVariable::getMaxDimension, PacketLib::SDFNoBlockFixed::getMaxDimension, PacketLib::SDFBVBlock::getMaxDimension, PacketLib::SDFBlockVariable::getMaxDimension, PacketLib::SDFBlockFixed::getMaxDimension, PacketLib::PacketDataField::getMaxDimension, PacketLib::Packet::getMaxDimension, PacketLib::SourceDataField::getMaxNumberOfBlock, PacketLib::SDFBVBlockFixed::getMaxNumberOfElement, PacketLib::SourceDataField::getMaxNumberOfElements, PacketLib::SDFBlockVariable::getMaxNumberOfElements, PacketLib::SourceDataField::getNumberOfFields, PacketLib::SDFNoBlockVariable::getNumberOfFields, PacketLib::SDFBVBlock::getNumberOfFields, PacketLib::SDFBlockVariable::getNumberOfFields, PacketLib::SDFBlockFixed::getNumberOfFields, PacketLib::PartOfPacket::getNumberOfFields, PacketLib::PacketStream::getNumberOfPacketType, PacketLib::SourceDataField::getNumberOfRealDataBlock, PacketLib::PacketDataField::getNumberOfRealDataBlock, PacketLib::SourceDataField::getNumberOfRealElement, PacketLib::SDFBVBlockFixed::getNumberOfRealElement, PacketLib::SDFBlockVariable::getNumberOfRealElement, PacketLib::InputPacketStreamFile::getPacketFromFileStreamPointer, PacketLib::InputPacketStreamFile::getPacketFromStream, PacketLib::PacketHeader::getPacketLength, PacketLib::PacketStream::getPrefixDimension, PacketLib::ByteStream::getSubByteStream, PacketLib::ByteStream::getSubByteStreamCopy, PacketLib::SourceDataField::getSubFromNBlock, PacketLib::SDFBlockVariable::loadFields, PacketLib::Packet::loadIdentifiers, PacketLib::PacketIdentifier::PacketIdentifier, PacketLib::SDFBlockVariable::printValue, PacketLib::SDFBlockFixed::printValue, PacketLib::InputPacketStream::readPacket, PacketLib::Socket::recv, PacketLib::Packet::setAndVerifyPacketValue, PacketLib::ByteStream::setByte, PacketLib::SDFBlockVariable::setByteStream, PacketLib::SDFBlockFixed::setByteStream, PacketLib::PartOfPacket::setByteStream, PacketLib::SourceDataField::setFieldValue, PacketLib::SDFNoBlockVariable::setFieldValue, PacketLib::SDFBlockVariable::setFieldValue, PacketLib::SDFBlockFixed::setFieldValue, PacketLib::PartOfPacket::setFieldValue, PacketLib::SDFBVBlockFixed::setIndexOfNElement, PacketLib::SDFBVBlockFixed::setMaxNumberOfElement, PacketLib::SourceDataField::setNumberOfRealDataBlock, PacketLib::SourceDataField::setNumberOfRealElement, PacketLib::SDFBVBlockFixed::setNumberOfRealElement, PacketLib::SDFBlockVariable::setNumberOfRealElement, PacketLib::SDFBVBlock::setOutputStream, PacketLib::SDFBlockVariable::setOutputStream, PacketLib::PartOfPacket::setOutputStream, PacketLib::PacketDataField::setOutputStream, PacketLib::Packet::setOutputStream, PacketLib::PacketValueDataFieldHeader::setPacketValue, PacketLib::Packet::setPacketValueSourceDataField, PacketLib::ByteStream::setStream, PacketLib::ByteStream::setWord, PacketLib::ByteStream::swap, PacketLib::Packet::verifyPacketValue, and PacketLib::Utility::wordToBinary.

5.2 std Namespace Reference

Chapter 6

Class Documentation

6.1 PacketLib::ByteStream Class Reference

Represent a stream of byte.

```
#include <ByteStream.h>
```

Public Methods

- **ByteStream** (bool **bigEndian**=false)
 - **ByteStream** (long **size**, bool **bigEndian**)
 - **ByteStream** (**byte** ***stream**, long **dim**, bool **bigEndian**, bool **memorySharing**=true)
 - **ByteStream** (ByteStream ***b0**, ByteStream ***b1**, ByteStream ***b2**)
 - **~ByteStream** ()
 - **byte** * **getStream** ()
 - **byte** * **getOutputStream** ()
 - void **endOutputStream** ()
 - ByteStream * **getSubByteStream** (**word** **first**, **word** **last**)
 - ByteStream * **getSubByteStreamCopy** (**word** **first**, **word** **last**)
 - bool **setStream** (**byte** ***b**, unsigned **dim**, bool **bigEndian**, bool **memorySharing**=true)
 - bool **setStream** (ByteStream ***b**, **word** **first**, **word** **last**)
 - void **setStreamCopy** (**byte** ***b**, unsigned **dim**)
 - bool **setWord** (unsigned **start**, **word** **value**)
 - void **setByte** (unsigned **start**, **word** **value**)
 - **byte** **getByte** (unsigned **byteNumber**)
 - long **getValue** (unsigned **start**, unsigned **dim**)
 - unsigned **getDimension** ()
 - char * **printStreamInHexadecimal** ()
 - bool **getMemAllocation** ()
 - bool **isBigEndian** () const
 - void **swap** ()
-

Public Attributes

- **byte * stream**

Static Public Attributes

- **dword count_object = 0**
- **dword count_object2 = 0**
- **long count_object_deleted = 0**
- **long count_object_deleted2 = 0**

Protected Methods

- **void setMemoryAllocated (bool allocated)**
- **void deleteStreamMemory ()**

Private Attributes

- **word byteInTheStream**
- **bool bigendian**
- **bool mem_allocation**
- **bool mem_allocation_constructor**

6.1.1 Detailed Description

Represent a stream of byte.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/ByteStream.h,v 1.6 2003/05/02
08:18:00 agile Exp

Id:

ByteStream.h,v 1.6 2003/05/02 08:18:00 agile Exp

Revision:

1.6

Definition at line 33 of file ByteStream.h.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 PacketLib::ByteStream::ByteStream (*bool bigEndian = false*)

The memory of byte* is not allocated

Definition at line 31 of file ByteStream.cpp.

References bigEndian, byteInTheStream, mem_allocation_constructor, setMemoryAllocated, and stream.

Referenced by getSubByteStream, and getSubByteStreamCopy.

6.1.2.2 PacketLib::ByteStream::ByteStream (*long size, bool bigEndian*)

The memory of byte* is allocated

Definition at line 48 of file ByteStream.cpp.

References bigEndian, PacketLib::byte, byteInTheStream, mem_allocation_constructor, setMemoryAllocated, and stream.

6.1.2.3 PacketLib::ByteStream::ByteStream (*byte * stream, long dim, bool bigEndian, bool memory_sharing = true*)

The memory of byte* is not allocated

Definition at line 66 of file ByteStream.cpp.

References bigEndian, PacketLib::byte, byteInTheStream, mem_allocation_constructor, setMemoryAllocated, stream, and swap.

6.1.2.4 PacketLib::ByteStream::ByteStream (*ByteStream * b0, ByteStream * b1, ByteStream * b2*)

Creates a new **ByteStream** (p. 21) that it is the sum of one or two or three **ByteStream** (p. 21) passed as input. It's possible to pass 0 as pointer. The memory of byte* is allocated.

Definition at line 86 of file ByteStream.cpp.

References bigEndian, PacketLib::byte, byteInTheStream, setMemoryAllocated, and stream.

6.1.2.5 PacketLib::ByteStream::~ByteStream ()

Definition at line 137 of file ByteStream.cpp.

References stream.

6.1.3 Member Function Documentation

6.1.3.1 void PacketLib::ByteStream::deleteStreamMemory () [protected]

Internal use only for memory management

Definition at line 484 of file ByteStream.cpp.

References mem_allocation, mem_allocation_constructor, and stream.

Referenced by setStream, and setStreamCopy.

6.1.3.2 void PacketLib::ByteStream::endOutputStream ()

This method must be call after the output operation.

Definition at line 268 of file ByteStream.cpp.

References swap.

6.1.3.3 byte PacketLib::ByteStream::getByte (unsigned byteNumber)

This method return a byte of the stram in the position byteNumber

Definition at line 154 of file ByteStream.cpp.

References PacketLib::byte, byteInTheStream, and stream.

6.1.3.4 unsigned PacketLib::ByteStream::getDimension ()

Gets the dimension of the stream

Definition at line 278 of file ByteStream.cpp.

References byteInTheStream.

6.1.3.5 bool PacketLib::ByteStream::getMemAllocation ()

Get type of allocations.

Returns:

True if this **ByteStream** (p. 21) contains the allocated memory area for the stream; false if this **ByteStream** (p. 21) haven't allocated the memory area but it's only a pointer of this area

Definition at line 356 of file ByteStream.cpp.

References mem_allocation.

6.1.3.6 byte * PacketLib::ByteStream::getOutputStream ()

Returns a pointer of the stream for output purpose. If there is problemas return NULL

Remarks:

In little endian architecture it is necessary to call **endOutputStream()** (p. 24) after the use of the byte* stream in output operations.

Definition at line 257 of file ByteStream.cpp.

References PacketLib::byte, stream, and swap.

6.1.3.7 byte * PacketLib::ByteStream::getStream ()

Definition at line 247 of file ByteStream.cpp.

References PacketLib::byte, and stream.

**6.1.3.8 ByteStream * PacketLib::ByteStream::getSubByteStream
(word *first*, word *last*)**

Returns a subset of the current stream. If there is problemas return NULL

Remarks:

This method don't allocate a new stream structure in memory, but create only a new **ByteStream** (p. 21) object that points in the same memory area.

Definition at line 214 of file ByteStream.cpp.

References bigEndian, byteInTheStream, ByteStream, stream, and PacketLib::word.

**6.1.3.9 ByteStream * PacketLib::ByteStream::get-
SubByteStreamCopy (word *first*, word
last)**

Returns a subset of the current stream. If there is problemas return NULL

Remarks:

This method allocate a new stream structure in memory.

Definition at line 229 of file ByteStream.cpp.

References bigEndian, PacketLib::byte, byteInTheStream, ByteStream, stream, and PacketLib::word.

**6.1.3.10 long PacketLib::ByteStream::getValue (unsigned *start*,
unsigned *dim*)**

Get the value in the stream starting from position start and of dim dimension

Parameters:

start Start position in the stream of byte

dim number of byte (1 oe 2)

Definition at line 167 of file ByteStream.cpp.

References PacketLib::byte, byteInTheStream, and stream.

6.1.3.11 bool PacketLib::ByteStream::isBigEndian () const

Definition at line 366 of file ByteStream.cpp.

References bigEndian.

**6.1.3.12 char * PacketLib::ByteStream::printStreamInHexadecimal
()**

Definition at line 288 of file ByteStream.cpp.

References byteInTheStream, and stream.

**6.1.3.13 void PacketLib::ByteStream::setByte (unsigned *start*,
word *value*)**

Sets a byte of the stream with the value of parameter.

Definition at line 376 of file ByteStream.cpp.

References stream, and PacketLib::word.

**6.1.3.14 void PacketLib::ByteStream::setMemoryAllocated (bool
allocated) [protected]**

Internal use only for memory management

Definition at line 442 of file ByteStream.cpp.

References mem_allocation.

Referenced by ByteStream, setStream, and setStreamCopy.

**6.1.3.15 bool PacketLib::ByteStream::setStream (ByteStream * *b*,
word *first*, word *last*)**

Get the stream from another object of the same type. Don't allocate new memory

Definition at line 333 of file ByteStream.cpp.

References bigEndian, byteInTheStream, deleteStreamMemory, setMemoryAllocated, stream, and PacketLib::word.

**6.1.3.16 bool PacketLib::ByteStream::setStream (byte * *b*,
unsigned *dim*, bool *big endian*, bool *memory_sharing* =
true)**

Sets the stream from arguments.

Definition at line 315 of file ByteStream.cpp.

References bigEndian, PacketLib::byte, byteInTheStream, deleteStreamMemory, setMemoryAllocated, stream, and swap.

**6.1.3.17 void PacketLib::ByteStream::setStreamCopy (byte * *b*,
unsigned *dim*)**

Copy the stream in argument of dimension dim.

Remarks:

This method delete old stream and creates a new stream in memory

Definition at line 298 of file ByteStream.cpp.

References PacketLib::byte, byteInTheStream, deleteStreamMemory, setMemoryAllocated, stream, and swap.

**6.1.3.18 bool PacketLib::ByteStream::setWord (unsigned *start*,
word *value*)**

Sets a word of the stream with the value of parameter.

Definition at line 386 of file ByteStream.cpp.

References ARCH_BIGENDIAN, bigEndian, PacketLib::byte, byteInTheStream, stream, and PacketLib::word.

6.1.3.19 void PacketLib::ByteStream::swap ()

Swap of the stream if the architecture is little endian

Definition at line 419 of file ByteStream.cpp.

References bigEndian, PacketLib::byte, byteInTheStream, stream, and PacketLib::word.

Referenced by ByteStream, endOutputStream, getOutputStream, setStream, and setStreamCopy.

6.1.4 Member Data Documentation

6.1.4.1 **bool PacketLib::ByteStream::bigEndian [private]**

Specifies if the stream is big or little endian

Definition at line 224 of file ByteStream.h.

Referenced by **ByteStream**, **getSubByteStream**, **getSubByteStreamCopy**, **isBigEndian**, **setStream**, **setWord**, and **swap**.

6.1.4.2 **word PacketLib::ByteStream::byteInTheStream [private]**

This attribute specifies the number of byte in the stream

Definition at line 218 of file ByteStream.h.

Referenced by **ByteStream**, **getByte**, **getDimension**, **getSubByteStream**, **getSubByteStreamCopy**, **getValue**, **printStreamInHexadecimal**, **setStream**, **setStreamCopy**, **setWord**, and **swap**.

6.1.4.3 **dword ByteStream::count_object = 0 [static]**

Definition at line 9 of file ByteStream.cpp.

6.1.4.4 **dword ByteStream::count_object2 = 0 [static]**

Definition at line 11 of file ByteStream.cpp.

6.1.4.5 **long ByteStream::count_object_deleted = 0 [static]**

Definition at line 14 of file ByteStream.cpp.

6.1.4.6 **long ByteStream::count_object_deleted2 = 0 [static]**

Definition at line 17 of file ByteStream.cpp.

6.1.4.7 **bool PacketLib::ByteStream::mem_allocation [private]**

Indicates if there is memory allocated for stream. It could be indicate that the **byte*** stream memory area is shared between many **ByteStream** (p. 21). In this case the memory must not be deallocated (when **memory_sharing** attribute is true).

Definition at line 233 of file ByteStream.h.

Referenced by **deleteStreamMemory**, **getMemAllocation**, and **setMemoryAllocated**.

6.1.4.8 **bool PacketLib::ByteStream::mem_allocation_constructor [private]**

Definition at line 237 of file ByteStream.h.

Referenced by ByteStream, and deleteStreamMemory.

6.1.4.9 **byte* PacketLib::ByteStream::stream**

Pointer of the stream

Definition at line 180 of file ByteStream.h.

Referenced by ByteStream, deleteStreamMemory, getByte, getOutputStream, getStream, getSubByteStream, getSubByteStreamCopy, getValue, printStreamInHexadecimal, setByte, setStream, setStreamCopy, setWord, swap, and ~ByteStream.

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

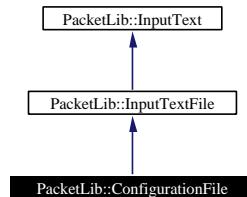
- **ByteStream.h**
- **ByteStream.cpp**

6.2 PacketLib::ConfigurationFile Class Reference

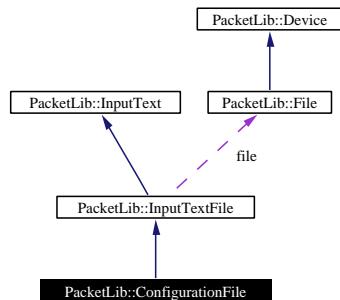
This class represent the text file which contains the configurations.

```
#include <ConfigurationFile.h>
```

Inheritance diagram for PacketLib::ConfigurationFile:



Collaboration diagram for PacketLib::ConfigurationFile:



Public Methods

- **ConfigurationFile ()**
- **~ConfigurationFile ()**
- virtual bool **open** (char **parameters) throw (PacketExceptionIO*)
- virtual char * **getLine** () throw (PacketExceptionIO*)
- virtual char * **getLine** (char *s) throw (PacketExceptionIO*)
- virtual void **close** () throw (PacketExceptionIO*)

Private Attributes

- char * **currentpwd**

6.2.1 Detailed Description

This class represent the text file which contains the configurations.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/ConfigurationFile.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

ConfigurationFile.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 35 of file ConfigurationFile.h.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 ConfigurationFile::ConfigurationFile ()

Definition at line 26 of file ConfigurationFile.cpp.

References currentpwd.

6.2.2.2 ConfigurationFile::~ConfigurationFile ()

Definition at line 36 of file ConfigurationFile.cpp.

References currentpwd.

6.2.3 Member Function Documentation

6.2.3.1 void ConfigurationFile::close () throw (PacketExceptionIO*) [virtual]

Reimplemented from **PacketLib::InputTextFile** (p. 76).

Definition at line 61 of file ConfigurationFile.cpp.

6.2.3.2 char * ConfigurationFile::getLine (char * s) throw (PacketExceptionIO*) [virtual]

Reimplemented from **PacketLib::InputTextFile** (p. 77).

Definition at line 97 of file ConfigurationFile.cpp.

6.2.3.3 `char * ConfigurationFile::getLine () throw
(PacketExceptionIO*) [virtual]`

Reimplemented from **PacketLib::InputTextFile** (p. 77).

Definition at line 72 of file ConfigurationFile.cpp.

6.2.3.4 `bool ConfigurationFile::open (char ** parameters) throw
(PacketExceptionIO*) [virtual]`

Reimplemented from **PacketLib::InputTextFile** (p. 77).

Definition at line 46 of file ConfigurationFile.cpp.

6.2.4 Member Data Documentation

6.2.4.1 `char* PacketLib::ConfigurationFile::currentpwd [private]`

Definition at line 67 of file ConfigurationFile.h.

Referenced by ConfigurationFile, and ~ConfigurationFile.

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

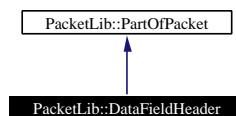
- **ConfigurationFile.h**
- **ConfigurationFile.cpp**

6.3 PacketLib::DataFieldHeader Class Reference

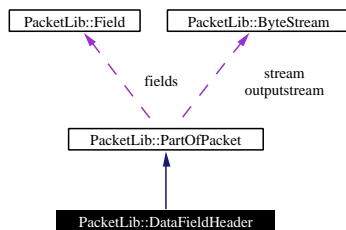
This class represent the data field header of a packet.

```
#include <DataFieldHeader.h>
```

Inheritance diagram for PacketLib::DataFieldHeader:



Collaboration diagram for PacketLib::DataFieldHeader:



6.3.1 Detailed Description

This class represent the data field header of a packet.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/DataFieldHeader.h,v	1.3
2002/11/13 15:16:35 andrea Exp	

Id:

DataFieldHeader.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 34 of file DataFieldHeader.h.

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

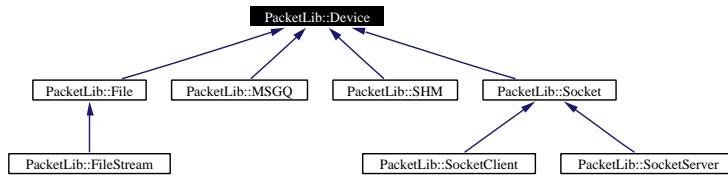
- **DataFieldHeader.h**

6.4 PacketLib::Device Class Reference

This class represent a **Device** (p. 34) for IO system.

```
#include <Device.h>
```

Inheritance diagram for PacketLib::Device:



Public Methods

- **Device (bool bigEndian)**
- virtual bool **isClosed () const**
- virtual bool **isEOF () const**
- virtual void **setEndSession (const bool &_newVal)**
- virtual const bool & **getEndSession () const**
- virtual bool **isBigEndian () const**

Protected Attributes

- bool **eof**
- bool **closed**
- bool **bigEndian**
- bool **endsession**

6.4.1 Detailed Description

This class represent a **Device** (p. 34) for IO system.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/Device.h,v 1.4 2003/05/02
08:18:00 agile Exp

Id:

Device.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 31 of file Device.h.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Device::Device (bool *bigendian*)

Definition at line 22 of file Device.cpp.

References bigendian, closed, and eof.

6.4.3 Member Function Documentation

6.4.3.1 const bool & Device::getEndSession () const [virtual]

Read property of bool endsession.

Definition at line 71 of file Device.cpp.

References endsession.

6.4.3.2 bool Device::isBigEndian () const [virtual]

Definition at line 34 of file Device.cpp.

References bigendian.

6.4.3.3 bool Device::isClosed () const [virtual]

Definition at line 47 of file Device.cpp.

References closed.

6.4.3.4 bool Device::isEOF () const [virtual]

Definition at line 57 of file Device.cpp.

References closed, and eof.

6.4.3.5 void Device::setEndSession (const bool & *_newVal*) [virtual]

Write property of bool endsession.

Definition at line 82 of file Device.cpp.

References endsession.

6.4.4 Member Data Documentation

6.4.4.1 bool PacketLib::Device::bigendian [protected]

Definition at line 73 of file Device.h.

Referenced by Device, PacketLib::File::getNByte, and isBigEndian.

6.4.4.2 bool PacketLib::Device::closed [protected]

Definition at line 69 of file Device.h.

Referenced by Device, PacketLib::File::getByte, PacketLib::File::isClosed, isClosed, PacketLib::File::isEOF, and isEOF.

6.4.4.3 bool PacketLib::Device::endsession [protected]

This variable indicates when a measurement session is terminated. The device class have this attribute because this information is passed with the device.

Definition at line 81 of file Device.h.

Referenced by getEndSession, and setEndSession.

6.4.4.4 bool PacketLib::Device::eof [protected]

Definition at line 65 of file Device.h.

Referenced by Device, PacketLib::File::getByte, PacketLib::File::isEOF, and isEOF.

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

- **Device.h**
- **Device.cpp**

6.5 PacketLib::Field Class Reference

This is a single field of a packet.

```
#include <Field.h>
```

Public Methods

- **Field** (char ***name**, char ***dimension**, char ***predefinedValue**, int **progressiv**)
- **~Field** ()

Public Attributes

- **word dimension**
- **bool thereIsPredefinedValue**
- **int predefinedValue**
- **int progressiv**
- **char * name**
- **word value**

6.5.1 Detailed Description

This is a single field of a packet.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/Field.h,v 1.3 2002/11/13 15:16:35
andrea Exp

Id:

Field.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 33 of file Field.h.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 **Field::Field** (char * *name*, char * *dimension*, char * *predefinedValue*, int *progressiv*)

Constructor of class.

Definition at line 12 of file Field.cpp.

References dimension, PacketLib::dword, name, predefinedValue, and thereIsPredefinedValue.

6.5.2.2 Field::~Field ()

Destructor of class.

Definition at line 40 of file Field.cpp.

References name.

6.5.3 Member Data Documentation

6.5.3.1 word PacketLib::Field::dimension

Dimension of field in bit.

Definition at line 52 of file Field.h.

Referenced by Field.

6.5.3.2 char* PacketLib::Field::name

Name of the field.

Definition at line 76 of file Field.h.

Referenced by Field, and ~Field.

6.5.3.3 int PacketLib::Field::predefinedValue

Predefined value of field (for the generation of packet).

Definition at line 64 of file Field.h.

Referenced by Field.

6.5.3.4 int PacketLib::Field::progressiv

Index of packet in the list of packet.

Definition at line 70 of file Field.h.

6.5.3.5 bool PacketLib::Field::thereIsPredefinedValue

It indicates that there's a predefined value for this field.

Definition at line 58 of file Field.h.

Referenced by Field.

6.5.3.6 word PacketLib::Field::value

Value reads from stream of byte (from packet).

Definition at line 81 of file Field.h.

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

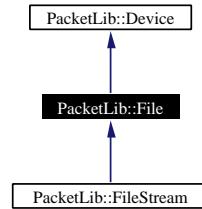
- [Field.h](#)
- [Field.cpp](#)

6.6 PacketLib::File Class Reference

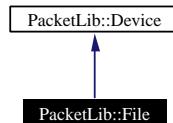
This class represent a device file for IO system.

```
#include <File.h>
```

Inheritance diagram for PacketLib::File:



Collaboration diagram for PacketLib::File:



Public Methods

- **File ()**
- **File (bool b, long startP=0)**
- **virtual bool open (char *filename, char *mode="r") throw (PacketExceptionIO*)**
- **virtual int getByte ()**
- **virtual ByteStream * getNByte (int N=1)**
- **virtual char * getLine () throw (PacketExceptionIO*)**
- **virtual void close () throw (PacketExceptionIO*)**
- **virtual bool isClosed ()**
- **char * getLastLineRead ()**
- **virtual long setpos (long offset) throw (PacketExceptionIO*)**
- **virtual long getpos ()**
- **virtual bool memBookmarkPos ()**
- **virtual bool setLastBookmarkPos ()**
- **virtual bool isEOF ()**
- **int setFirstPos () throw (PacketExceptionIO*)**
- **bool fchdir () throw (PacketExceptionIO*)**
- **long find (byte b) throw (PacketExceptionIO*)**
- **bool writeString (char *str) throw (PacketExceptionIO*)**
- **bool writeByteStream (ByteStream *b) throw (PacketExceptionIO*)**

Static Public Attributes

- unsigned long **byte_read** = 0
- unsigned long **char_read** = 0

Protected Attributes

- char * **lastLineRead**
- FILE * **fp**
- long **bookmarkPos**
- long **startPosition**
- char * **filename**

6.6.1 Detailed Description

This class represent a device file for IO system.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/File.h,v 1.2 2003/05/02 08:18:00
agile Exp

Id:

File.h,v 1.2 2003/05/02 08:18:00 agile Exp

Revision:

1.2

Definition at line 36 of file File.h.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 File::File ()

Definition at line 33 of file File.cpp.

References filename, and startPosition.

6.6.2.2 File::File (bool *b*, long *startP* = 0)

Parameters:

- b* Indicates if the machine is big or little endian. If little endian, this class swap the byte of a word
- startP* Starting position whitin file

Definition at line 45 of file File.cpp.
References filename, and startPosition.

6.6.3 Member Function Documentation

6.6.3.1 void File::close () throw (PacketExceptionIO*) [virtual]

Close the file.

Definition at line 209 of file File.cpp.

6.6.3.2 bool File::fchdir () throw (PacketExceptionIO*)

Changes directory. The path is in the filename of file opened.

Precondition:

The file name must be setted.

Definition at line 320 of file File.cpp.

6.6.3.3 long File::find (byte b) throw (PacketExceptionIO*)

Definition at line 351 of file File.cpp.

References PacketLib::byte.

6.6.3.4 int File::getByte () [virtual]

Get a single byte from current position of file opened.

Precondition:

The file must be opened.

Definition at line 86 of file File.cpp.

References PacketLib::Device::closed, PacketLib::Device::eof, EOI, and fp.

Referenced by getNByte.

6.6.3.5 char * File::getLastLineRead ()

Returns:

Returns last line read from file.

Definition at line 234 of file File.cpp.

References lastLineRead.

6.6.3.6 char * File::getLine () throw (PacketExceptionIO*) [virtual]

Get the current line.

Precondition:

The file must be opened.

Definition at line 164 of file File.cpp.

6.6.3.7 ByteStream * File::getNByte (int *N* = 1) [virtual]

Reads a pair number of byte from opened file and manages the big or little endian format. If the format of machine is little endian, a pair of byte is swapped.

Precondition:

The file must be opened.

Definition at line 112 of file File.cpp.

References PacketLib::Device::bigEndian, PacketLib::byte, EOI, and getByte.

Referenced by PacketLib::FileStream::readDataField, PacketLib::FileStream::readHeader, and PacketLib::FileStream::readPrefix.

6.6.3.8 long File::getpos () [virtual]

Gets the current file position.

Precondition:

The file must be opened.

Definition at line 261 of file File.cpp.

References fp.

6.6.3.9 bool File::isClosed () [virtual]**Returns:**

true if file is closed, false if file is opened.

Definition at line 224 of file File.cpp.

References PacketLib::Device::closed.

6.6.3.10 bool File::isEOF () [virtual]**Returns:**

Returns true is the position into file is at end of file

Definition at line 297 of file File.cpp.

References PacketLib::Device::closed, and PacketLib::Device::eof.

6.6.3.11 bool File::memBookmarkPos () [virtual]

Stores the current position

Precondition:

The file must be opened.

Definition at line 271 of file File.cpp.

References bookmarkPos, and fp.

6.6.3.12 bool File::open (char * *filename*, char * *mode* = "r") throw (PacketExceptionIO*) [virtual]

Open the file

Parameters:

filename Name of file

mode Mode of file opened. r for read, w for writing

Definition at line 57 of file File.cpp.

6.6.3.13 int File::setFirstPos () throw (PacketExceptionIO*)

Precondition:

The file must be opened.

Definition at line 310 of file File.cpp.

6.6.3.14 bool File::setLastBookmarkPos () [virtual]

Sets the position stored with **memBookmarkPos()** (p. 44) method.

Precondition:

The file must be opened.

Definition at line 284 of file File.cpp.

References bookmarkPos, and fp.

6.6.3.15 long File::setpos (long *offset*) throw (PacketExceptionIO*) [virtual]

Sets the position into file.

Precondition:

The file must be opened.

Definition at line 244 of file File.cpp.

**6.6.3.16 bool File::writeByteStream (ByteStream * *b*) throw
(PacketExceptionIO*)**

Writes a stream of byte into opend file.

Precondition:

The file must be opened in w mode

Definition at line 382 of file File.cpp.

References PacketLib::byte.

**6.6.3.17 bool File::writeString (char * *str*) throw
(PacketExceptionIO*)**

Writes a string into opend file.

Precondition:

The file must be opened in w mode

Definition at line 368 of file File.cpp.

6.6.4 Member Data Documentation

6.6.4.1 long PacketLib::File::bookmarkPos [protected]

Definition at line 181 of file File.h.

Referenced by memBookmarkPos, and setLastBookmarkPos.

6.6.4.2 unsigned long File::byte_read = 0 [static]

Definition at line 26 of file File.cpp.

6.6.4.3 unsigned long File::char_read = 0 [static]

Definition at line 29 of file File.cpp.

6.6.4.4 char* PacketLib::File::filename [protected]

Definition at line 189 of file File.h.

Referenced by File.

6.6.4.5 FILE* PacketLib::File::fp [protected]

Definition at line 177 of file File.h.

Referenced by getByte, getpos, memBookmarkPos, and setLastBookmarkPos.

6.6.4.6 char* PacketLib::File::lastLineRead [protected]

Definition at line 173 of file File.h.

Referenced by getLastLineRead.

6.6.4.7 long PacketLib::File::startPosition [protected]

Definition at line 185 of file File.h.

Referenced by File.

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

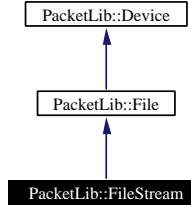
- [File.h](#)
- [File.cpp](#)

6.7 PacketLib::FileStream Class Reference

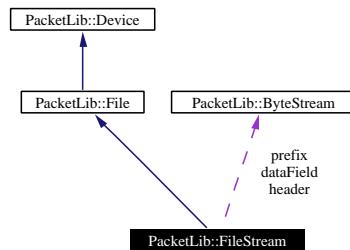
This class represent a stream of byte into a file.

```
#include <FileStream.h>
```

Inheritance diagram for PacketLib::FileStream:



Collaboration diagram for PacketLib::FileStream:



Public Methods

- **FileStream** (bool **prefix**, bool **bigen**, **word** **dimprefix**, long **startposition**=0)
- **ByteStream * readPrefix ()**
- **ByteStream * readHeader (unsigned int dimHeader)**
- **ByteStream * readDataField (unsigned int dimDataField)**

Private Attributes

- bool **thereIsPrefix**
- **ByteStream * header**
- **ByteStream * dataField**
- **ByteStream * prefix**
- **word dimPrefix**

6.7.1 Detailed Description

This class represent a stream of byte into a file.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/FileStream.h,v 1.3 2002/11/13
15:16:35 andrea Exp

Id:

FileStream.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 33 of file FileStream.h.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 **FileStream::FileStream (bool *prefix*, bool *bigen*, word *dimprefix*, long *startposition* = 0)**

Definition at line 7 of file FileStream.cpp.

References dimPrefix, prefix, thereIsPrefix, and PacketLib::word.

6.7.3 Member Function Documentation

6.7.3.1 **ByteStream * FileStream::readDataField (unsigned int *dimDataField*)**

Definition at line 30 of file FileStream.cpp.

References dataField, and PacketLib::File::getNByte.

6.7.3.2 **ByteStream * FileStream::readHeader (unsigned int *dimHeader*)**

Definition at line 19 of file FileStream.cpp.

References PacketLib::File::getNByte, and header.

6.7.3.3 **ByteStream * FileStream::readPrefix ()**

Definition at line 42 of file FileStream.cpp.

References dimPrefix, PacketLib::File::getNByte, and prefix.

6.7.4 Member Data Documentation

6.7.4.1 `ByteStream* PacketLib::FileStream::dataField [private]`

Definition at line 65 of file FileStream.h.

Referenced by `readDataField`.

6.7.4.2 `word PacketLib::FileStream::dimPrefix [private]`

Definition at line 69 of file FileStream.h.

Referenced by `FileStream`, and `readPrefix`.

6.7.4.3 `ByteStream* PacketLib::FileStream::header [private]`

Definition at line 63 of file FileStream.h.

Referenced by `readHeader`.

6.7.4.4 `ByteStream* PacketLib::FileStream::prefix [private]`

Definition at line 67 of file FileStream.h.

Referenced by `FileStream`, and `readPrefix`.

6.7.4.5 `bool PacketLib::FileStream::thereIsPrefix [private]`

Indicates if it's present a prefix for each packet

- true if it's present
- false if it isn't present

Definition at line 61 of file FileStream.h.

Referenced by `FileStream`.

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

- `FileStream.h`
- `FileStream.cpp`

6.8 PacketLib::FileStreamPointer Class Reference

This class represent a pointer into a file of byte.

```
#include <FileStreamPointer.h>
```

Public Methods

- **FileStreamPointer ()**
- **~FileStreamPointer ()**

Public Attributes

- int **typeOfPacket**
- long **pointerStart**
- long **index**
- char * **nameOfPacket**
- long **pointerEnd**

6.8.1 Detailed Description

This class represent a pointer into a file of byte.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/FileStreamPointer.h,v
2002/11/13 15:16:35 andrea Exp

1.3

Id:

FileStreamPointer.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 31 of file FileStreamPointer.h.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 FileStreamPointer::FileStreamPointer ()

Definition at line 24 of file FileStreamPointer.cpp.

6.8.2.2 FileStreamPointer::~FileStreamPointer ()

Definition at line 33 of file FileStreamPointer.cpp.

6.8.3 Member Data Documentation

6.8.3.1 long PacketLib::FileStreamPointer::index

Definition at line 59 of file FileStreamPointer.h.

6.8.3.2 char* PacketLib::FileStreamPointer::nameOfPacket

Definition at line 63 of file FileStreamPointer.h.

6.8.3.3 long PacketLib::FileStreamPointer::pointerEnd

Definition at line 67 of file FileStreamPointer.h.

6.8.3.4 long PacketLib::FileStreamPointer::pointerStart

pointer in the file (**FileStream** (p. 47))

Definition at line 55 of file FileStreamPointer.h.

6.8.3.5 int PacketLib::FileStreamPointer::typeOfPacket

Type of packet. This is the index in the list of packetType attribute of **Input-
PacketStream** (p. 58) class

Definition at line 49 of file FileStreamPointer.h.

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

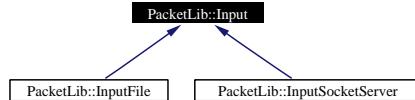
- **FileStreamPointer.h**
- **FileStreamPointer.cpp**

6.9 PacketLib::Input Class Reference

This class represent a generic input for IO system.

```
#include <Input.h>
```

Inheritance diagram for PacketLib::Input:



Public Methods

- **Input (bool bigEndian)**
- virtual **~Input ()**
- virtual bool **open (char **parameters)=0** throw (PacketExceptionIO*)
- virtual **ByteStream * readByteStream (int n_byte)=0** throw (PacketExceptionIO*)
- virtual char * **readString ()=0** throw (PacketExceptionIO*)
- virtual void **close ()=0** throw (PacketExceptionIO*)
- virtual int **getType ()=0**
- virtual bool **isClosed ()**
- virtual bool **isEOF ()**
- virtual bool **isBigEndian ()**

Protected Attributes

- bool **eof**
- bool **closed**
- bool **bigEndian**

6.9.1 Detailed Description

This class represent a generic input for IO system.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/Input.h,v 1.4 2003/05/02 08:18:00
agile Exp

Id:

Input.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 34 of file Input.h.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 Input::Input (*bool bigEndian*)

Definition at line 24 of file Input.cpp.

References bigEndian, closed, and eof.

6.9.2.2 Input::~Input () [virtual]

Definition at line 36 of file Input.cpp.

6.9.3 Member Function Documentation

6.9.3.1 virtual void PacketLib::Input::close () throw (PacketExceptionIO*) [pure virtual]

Implemented in **PacketLib::InputFile** (p. 56).

6.9.3.2 virtual int PacketLib::Input::getType () [pure virtual]

Implemented in **PacketLib::InputFile** (p. 56).

6.9.3.3 bool Input::isBigEndian () [virtual]

Definition at line 69 of file Input.cpp.

References bigEndian.

6.9.3.4 bool Input::isClosed () [virtual]

Definition at line 46 of file Input.cpp.

References closed.

6.9.3.5 bool Input::isEOF () [virtual]

Definition at line 56 of file Input.cpp.

References eof.

**6.9.3.6 virtual bool PacketLib::Input::open (char ** *parameters*)
throw (PacketExceptionIO*) [pure virtual]**

Implemented in **PacketLib::InputFile** (p. 56).

**6.9.3.7 virtual ByteStream* PacketLib::Input::readByteStream (int
n_byte) throw (PacketExceptionIO*) [pure virtual]**

Implemented in **PacketLib::InputFile** (p. 57).

**6.9.3.8 virtual char* PacketLib::Input::readString () throw
(PacketExceptionIO*) [pure virtual]**

Implemented in **PacketLib::InputFile** (p. 57).

6.9.4 Member Data Documentation

6.9.4.1 bool PacketLib::Input::bigEndian [protected]

Definition at line 90 of file Input.h.

Referenced by Input, and isBigEndian.

6.9.4.2 bool PacketLib::Input::closed [protected]

Definition at line 86 of file Input.h.

Referenced by Input, and isClosed.

6.9.4.3 bool PacketLib::Input::eof [protected]

Definition at line 82 of file Input.h.

Referenced by Input, and isEOF.

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

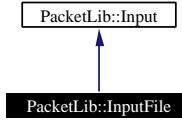
- **Input.h**
- **Input.cpp**

6.10 PacketLib::InputFile Class Reference

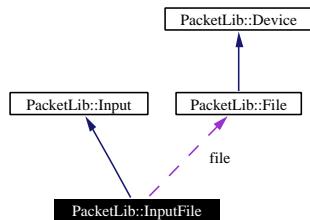
This class represent a **File** (p. 40) input for IO system.

```
#include <InputFile.h>
```

Inheritance diagram for PacketLib::InputFile:



Collaboration diagram for PacketLib::InputFile:



Public Methods

- **InputFile (bool bigEndian)**
- **~InputFile ()**
- virtual bool **open** (char **parameters) throw (PacketExceptionIO*)
- virtual void **close ()** throw (PacketExceptionIO*)
- virtual **ByteStream * readByteStream** (int n_byte) throw (PacketExceptionIO*)
- virtual char * **readString ()** throw (PacketExceptionIO*)
- virtual int **getType ()**

Protected Attributes

- **File * file**
- **char * filename**

6.10.1 Detailed Description

This class represent a **File** (p. 40) input for IO system.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/InputFile.h,v 1.2 2003/05/02
08:18:00 agile Exp

Id:

InputFile.h,v 1.2 2003/05/02 08:18:00 agile Exp

Revision:

1.2

Definition at line 35 of file InputFile.h.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 InputFile::InputFile (bool *bg*)

Definition at line 24 of file InputFile.cpp.

References file.

6.10.2.2 InputFile::~InputFile ()

Definition at line 34 of file InputFile.cpp.

References file.

6.10.3 Member Function Documentation

6.10.3.1 void InputFile::close () throw (PacketExceptionIO*) [virtual]

Implements **PacketLib::Input** (p. 53).

Definition at line 57 of file InputFile.cpp.

6.10.3.2 virtual int PacketLib::InputFile::getType () [inline, virtual]

Implements **PacketLib::Input** (p. 53).

Definition at line 64 of file InputFile.h.

6.10.3.3 bool InputFile::open (char ** *parameters*) throw (PacketExceptionIO*) [virtual]

Implements **PacketLib::Input** (p. 54).

Definition at line 44 of file InputFile.cpp.

6.10.3.4 `ByteStream * InputFile::readByteStream (int n-byte)
throw (PacketExceptionIO*) [virtual]`

Implements **PacketLib::Input** (p. 54).

Definition at line 68 of file InputFile.cpp.

6.10.3.5 `char * InputFile::readString () throw
(PacketExceptionIO*) [virtual]`

Implements **PacketLib::Input** (p. 54).

Definition at line 86 of file InputFile.cpp.

6.10.4 Member Data Documentation

6.10.4.1 `File* PacketLib::InputFile::file [protected]`

Definition at line 70 of file InputFile.h.

Referenced by InputFile, and ~InputFile.

6.10.4.2 `char* PacketLib::InputFile::filename [protected]`

Definition at line 74 of file InputFile.h.

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

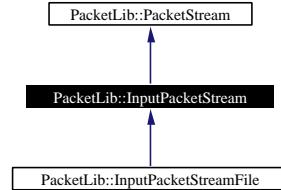
- `InputFile.h`
- `InputFile.cpp`

6.11 PacketLib::InputPacketStream Class Reference

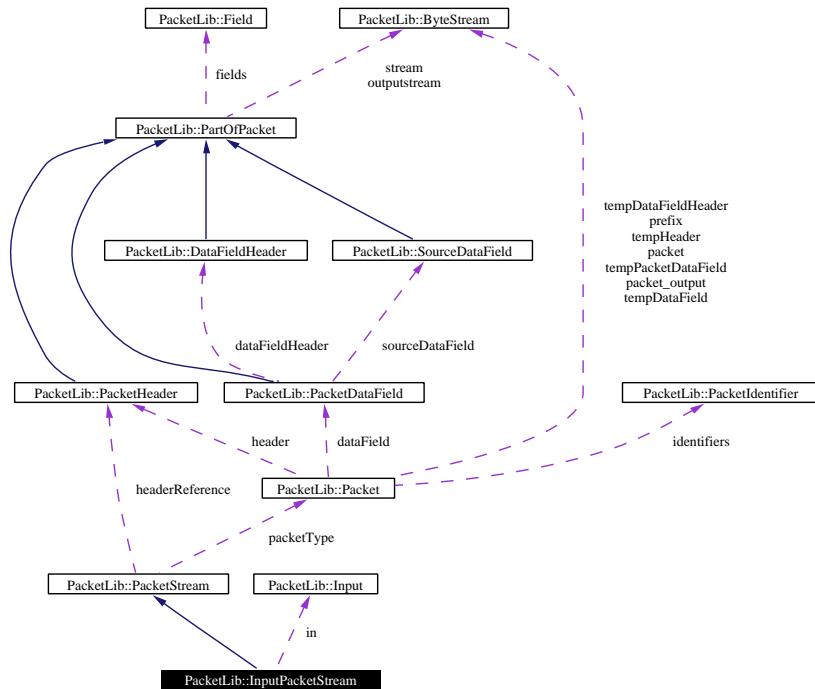
This class represents the stream of byte in input that contains the packets. The input object must be deleted by its creator.

```
#include <InputPacketStream.h>
```

Inheritance diagram for PacketLib::InputPacketStream:



Collaboration diagram for PacketLib::InputPacketStream:



Public Methods

- **InputPacketStream ()**
- **InputPacketStream (char *fileNameConfig)**

- `~InputPacketStream ()`
- `int detPacketType (ByteStream *prefix, ByteStream *packetHeader, ByteStream *packetDataField)`
- `int detPacketType (ByteStream *prefix, ByteStream *packet)`
- `void setInput (Input *in)`
- `Packet * readPacket () throw (PacketExceptionIO*)`

Protected Attributes

- `Input * in`

6.11.1 Detailed Description

This class represents the stream of byte in input that contains the packets. The input object must be deleted by its creator.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/InputPacketStream.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

InputPacketStream.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 35 of file InputPacketStream.h.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 InputPacketStream::InputPacketStream ()

Definition at line 24 of file InputPacketStream.cpp.

References in.

6.11.2.2 InputPacketStream::InputPacketStream (char * *fileNameConfig*)

Definition at line 34 of file InputPacketStream.cpp.

References in.

6.11.2.3 InputPacketStream::~InputPacketStream ()

Definition at line 44 of file InputPacketStream.cpp.

6.11.3 Member Function Documentation

6.11.3.1 int InputPacketStream::detPacketType (ByteStream * *prefix*, ByteStream * *packet*)

Returns:

The index of packet type if it's recognized. 0 if packet isn't recognized.

Parameters:

prefix A **ByteStream** (p.21) that contains the prefix of packet (if present).

packet A **ByteStream** (p.21) that contains the packet.

Definition at line 72 of file InputPacketStream.cpp.

References PacketLib::PacketStream::getPacketType, PacketLib::PacketStream::numberOfPacketType, and PacketLib::PacketStream::prefix.

6.11.3.2 int InputPacketStream::detPacketType (ByteStream * *prefix*, ByteStream * *packetHeader*, ByteStream * *packetDataField*)

Returns:

The index of packet type if it's recognized. 0 if packet isn't recognized.

Parameters:

prefix A **ByteStream** (p.21) that contains the prefix of packet (if present).

packetHeader A **ByteStream** (p.21) that contains the packet header.

packetDataField A **ByteStream** (p.21) that contains the packet data field.

Definition at line 54 of file InputPacketStream.cpp.

References PacketLib::PacketStream::getPacketType, PacketLib::PacketStream::numberOfPacketType, and PacketLib::PacketStream::prefix.

6.11.3.3 Packet * InputPacketStream::readPacket () throw (PacketExceptionIO*)

This method read a telemetry packet

Precondition:

The setInput method must be invoked

Returns:

A pointer telemetry packet. Make attention: the object returned is one of the TM packet object of the array of this object. Don't delete it!

Definition at line 100 of file InputPacketStream.cpp.

References PacketLib::word.

6.11.3.4 void InputPacketStream::setInput (Input * *in*)

This method sets the generic input of the stream

Parameters:

in An input.

Definition at line 90 of file InputPacketStream.cpp.

References in.

6.11.4 Member Data Documentation**6.11.4.1 Input* PacketLib::InputPacketStream::in [protected]**

Definition at line 87 of file InputPacketStream.h.

Referenced by InputPacketStream, and setInput.

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

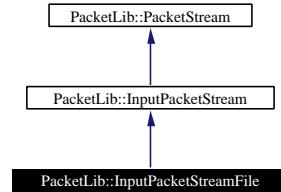
- **InputPacketStream.h**
- **InputPacketStream.cpp**

6.12 PacketLib::InputPacketStreamFile Class Reference

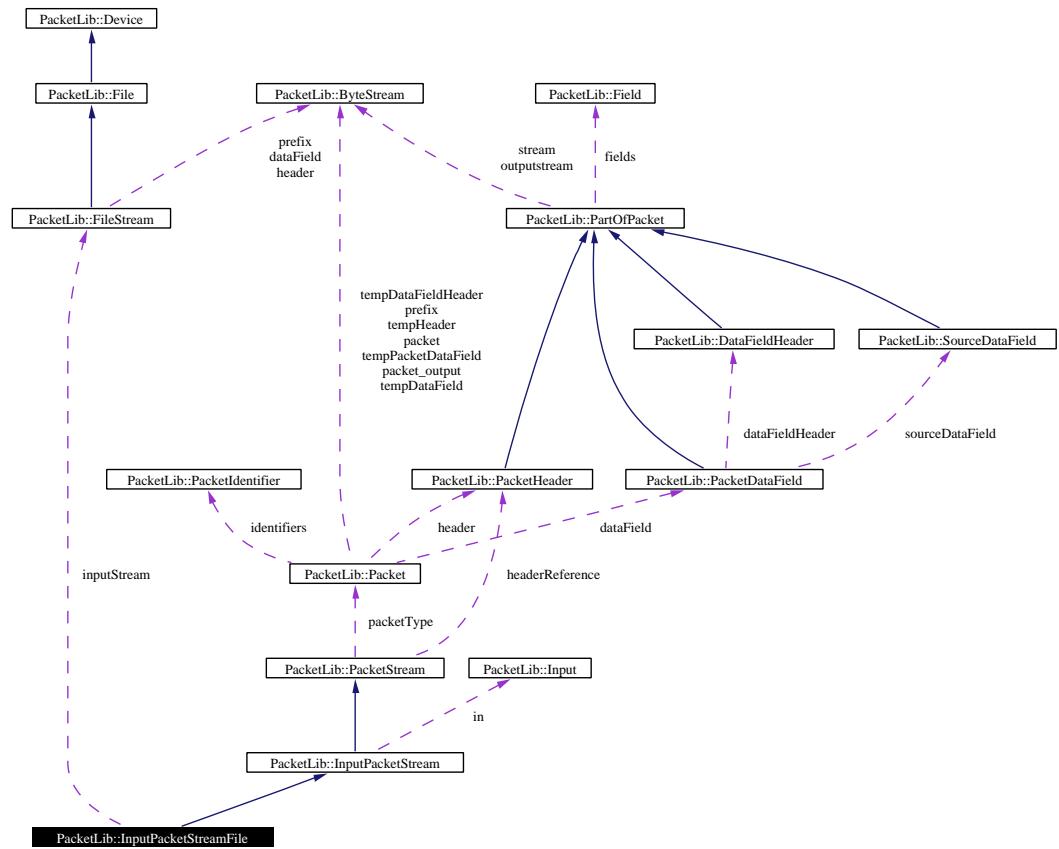
This class represents the stream of byte into a file that contains the packets.

```
#include <InputPacketStreamFile.h>
```

Inheritance diagram for PacketLib::InputPacketStreamFile:



Collaboration diagram for PacketLib::InputPacketStreamFile:



Public Methods

- **InputPacketStreamFile** (char *fileNameConfig, char *fileNameStream, long initialPosition)
- **InputPacketStreamFile** ()
- **~InputPacketStreamFile** ()
- void **setFileNameStream** (char *)
- bool **openInputStream** () throw (PacketExceptionIO*)
- void **closeInputStream** () throw (PacketExceptionIO*)
- bool **isInputStreamEOF** () throw (PacketExceptionIO*)
- **Packet * getPacketFromStream** () throw (PacketExceptionIO *)
- long **getNumberOfFileStreamPointer** ()
- bool **freeRun** () throw (PacketExceptionIO*)
- **Packet * getFileStreamPointer** (int index) throw (PacketExceptionIO*)
- **FileStreamPointer * getFileStreamPointer** (int index)
- void **setInitialPosition** (long p)

Public Attributes

- list< **FileStreamPointer** > **listOfFileStreamPointer**
- **FileStream * inputStream**

Private Attributes

- char * **fileNameStream**
- long **initialPosition**
- long **numberOfFileStreamPointer**

6.12.1 Detailed Description

This class represents the stream of byte into a file that contains the packets.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/InputPacketStreamFile.h,v 1.2
2003/05/02 08:18:00 agile Exp

Id:

InputPacketStreamFile.h,v 1.2 2003/05/02 08:18:00 agile Exp

Revision:

1.2

Definition at line 40 of file InputPacketStreamFile.h.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 InputPacketStreamFile::InputPacketStreamFile (char * *fileNameConfig*, char * *fileNameStream*, long *initialPosition*)

Definition at line 15 of file InputPacketStreamFile.cpp.

References *fileNameStream*, *initialPosition*, *inputStream*, and *numberOfFileStreamPointer*.

6.12.2.2 InputPacketStreamFile::InputPacketStreamFile ()

Definition at line 29 of file InputPacketStreamFile.cpp.

References *fileNameStream*, *initialPosition*, *inputStream*, and *numberOfFileStreamPointer*.

6.12.2.3 InputPacketStreamFile::~InputPacketStreamFile ()

Definition at line 42 of file InputPacketStreamFile.cpp.

References *inputStream*, *listOfFileStreamPointer*, *PacketLib::PacketStream::numberOfPacketType*, and *PacketLib::PacketStream::packetType*.

6.12.3 Member Function Documentation

6.12.3.1 void InputPacketStreamFile::closeInputStream () throw (PacketExceptionIO*)

Closes the input stream

Definition at line 190 of file InputPacketStreamFile.cpp.

6.12.3.2 bool InputPacketStreamFile::freeRun () throw (PacketExceptionIO*)

This method builds a list of **FileStreamPointer** (p. 50)

Definition at line 81 of file InputPacketStreamFile.cpp.

References *PacketLib::word*.

6.12.3.3 FileStreamPointer * InputPacketStreamFile::getFileStreamPointer (int *index*)

Get a file stream pointer

Precondition:

freeRun() (p. 64) must be invoked.

Definition at line 200 of file InputPacketStreamFile.cpp.

References listOfFileStreamPointer, and numberOfFileStreamPointer.

6.12.3.4 long InputPacketStreamFile::getNumberOfFileStream- Pointer ()

Definition at line 242 of file InputPacketStreamFile.cpp.

References numberOfFileStreamPointer.

6.12.3.5 Packet * InputPacketStreamFile::getPacket- FromFileStreamPointer (int *index*) throw (PacketExceptionIO*)

Get a packet form the file stream pointer

Precondition:

freeRun() (p. 64) must be invocated.

Definition at line 150 of file InputPacketStreamFile.cpp.

References PacketLib::word.

6.12.3.6 Packet * InputPacketStreamFile::getPacketFromStream () throw (PacketExceptionIO *)

Definition at line 252 of file InputPacketStreamFile.cpp.

References PacketLib::word.

6.12.3.7 bool InputPacketStreamFile::isInputStreamEOF () throw (PacketExceptionIO*)

Returns:

true if stream is at the end of file (input)

Definition at line 305 of file InputPacketStreamFile.cpp.

6.12.3.8 bool InputPacketStreamFile::openInputStream () throw (PacketExceptionIO*)

Opens the input stream.

Definition at line 57 of file InputPacketStreamFile.cpp.

6.12.3.9 void InputPacketStreamFile::setFileNameStream (char * f)

Definition at line 232 of file InputPacketStreamFile.cpp.

References fileNameStream.

6.12.3.10 void InputPacketStreamFile::setInitialPosition (long p)

Definition at line 219 of file InputPacketStreamFile.cpp.

References initialPosition.

6.12.4 Member Data Documentation

6.12.4.1 char* PacketLib::InputPacketStreamFile::fileNameStream [private]

File (p. 40) name of file that contains the packet

Definition at line 129 of file InputPacketStreamFile.h.

Referenced by InputPacketStreamFile, and setFileNameStream.

6.12.4.2 long PacketLib::InputPacketStreamFile::initialPosition [private]

Initial position into file

Definition at line 135 of file InputPacketStreamFile.h.

Referenced by InputPacketStreamFile, and setInitialPosition.

6.12.4.3 FileStream* PacketLib::InputPacketStreamFile::inputStream

File (p. 40) that contains byte stream

Definition at line 121 of file InputPacketStreamFile.h.

Referenced by InputPacketStreamFile, and ~InputPacketStreamFile.

6.12.4.4 list<FileStreamPointer> PacketLib::InputPacketStreamFile::listOfFileStreamPointer

List of **FileStreamPointer** (p. 50) in the inputStream

Definition at line 115 of file InputPacketStreamFile.h.

Referenced by getFileStreamPointer, and ~InputPacketStreamFile.

6.12.4.5 long PacketLib::InputPacketStreamFile::numberOfFileStreamPointer [private]

Number of packet in the listOfFileStreamPointer

Definition at line 141 of file InputPacketStreamFile.h.

Referenced by getFileStreamPointer, getNumberOfFileStreamPointer, and InputPacketStreamFile.

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

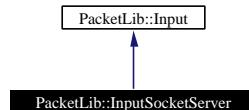
- **InputPacketStreamFile.h**
- **InputPacketStreamFile.cpp**

6.13 PacketLib::InputSocketServer Class Reference

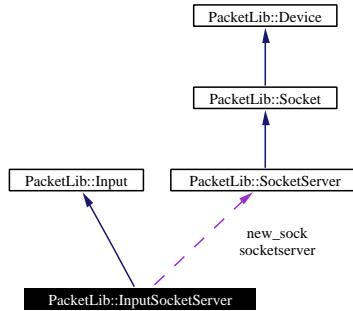
This class represent a **Socket** (p.188) Server input for IO system.

```
#include <InputSocketServer.h>
```

Inheritance diagram for PacketLib::InputSocketServer:



Collaboration diagram for PacketLib::InputSocketServer:



Public Methods

- **InputSocketServer (bool bigendian)**
- **~InputSocketServer ()**
- **virtual bool open (char **parameters) throw (PacketExceptionIO*)**
- **virtual void close () throw (PacketExceptionIO*)**
- **virtual ByteStream * readByteStream (int n_byte) throw (PacketExceptionIO*)**
- **virtual char * readString () throw (PacketExceptionIO*)**
- **virtual int getType ()**

Protected Attributes

- **SocketServer * socketserver**
- **int port**
- **bool accepted**

Private Attributes

- `SocketServer * new_sock`

6.13.1 Detailed Description

This class represent a `Socket` (p. 188) Server input for IO system.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/InputSocketServer.h,v	1.2
2003/05/02 08:18:00 agile Exp	

Id:

`InputSocketServer.h,v 1.2 2003/05/02 08:18:00 agile Exp`

Revision:

1.2

Definition at line 35 of file `InputSocketServer.h`.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 InputSocketServer::InputSocketServer (bool *bg*)

Definition at line 24 of file `InputSocketServer.cpp`.

References `new_sock`, and `socketserver`.

6.13.2.2 InputSocketServer::~InputSocketServer ()

Definition at line 35 of file `InputSocketServer.cpp`.

References `socketserver`.

6.13.3 Member Function Documentation

6.13.3.1 void InputSocketServer::close () throw (`PacketExceptionIO*`) [virtual]

Implements `PacketLib::Input` (p. 53).

Definition at line 62 of file `InputSocketServer.cpp`.

6.13.3.2 `virtual int PacketLib::InputSocketServer::getType () [inline, virtual]`

Implements **PacketLib::Input** (p. 53).

Definition at line 64 of file InputSocketServer.h.

6.13.3.3 `bool InputSocketServer::open (char ** parameters) throw (PacketExceptionIO*) [virtual]`

Implements **PacketLib::Input** (p. 54).

Definition at line 45 of file InputSocketServer.cpp.

6.13.3.4 `ByteStream * InputSocketServer::readByteStream (int n_byte) throw (PacketExceptionIO*) [virtual]`

Implements **PacketLib::Input** (p. 54).

Definition at line 74 of file InputSocketServer.cpp.

6.13.3.5 `char * InputSocketServer::readString () throw (PacketExceptionIO*) [virtual]`

Implements **PacketLib::Input** (p. 54).

Definition at line 101 of file InputSocketServer.cpp.

6.13.4 Member Data Documentation

6.13.4.1 `bool PacketLib::InputSocketServer::accepted [protected]`

Definition at line 78 of file InputSocketServer.h.

6.13.4.2 `SocketServer* PacketLib::InputSocketServer::new_sock [private]`

Definition at line 84 of file InputSocketServer.h.

Referenced by InputSocketServer.

6.13.4.3 `int PacketLib::InputSocketServer::port [protected]`

Definition at line 74 of file InputSocketServer.h.

6.13.4.4 **SocketServer* PacketLib::InputSocketServer::socketserver** [protected]

Definition at line 70 of file InputSocketServer.h.

Referenced by InputSocketServer, and ~InputSocketServer.

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

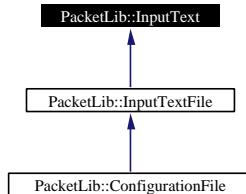
- **InputSocketServer.h**
- **InputSocketServer.cpp**

6.14 PacketLib::InputText Class Reference

This class represents generic type of input. The input must be a text input. This class is used for the configuration files.

```
#include <InputText.h>
```

Inheritance diagram for PacketLib::InputText:



Public Methods

- **InputText ()**
- virtual **~InputText ()**
- virtual char * **getLine ()=0** throw (PacketExceptionIO*)
- virtual char * **getLine (char *s)=0** throw (PacketExceptionIO*)
- virtual char * **getLastLineRead ()=0**
- virtual bool **isClosed ()**
- virtual bool **isEOF ()**

Protected Attributes

- bool **eof**
- bool **closed**

6.14.1 Detailed Description

This class represents generic type of input. The input must be a text input. This class is used for the configuration files.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/InputText.h,v 1.4 2003/05/02
08:18:00 agile Exp

Id:

InputText.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 36 of file InputText.h.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 InputText::InputText ()

Definition at line 24 of file InputText.cpp.

References closed, and eof.

6.14.2.2 InputText::~InputText () [virtual]

Definition at line 35 of file InputText.cpp.

6.14.3 Member Function Documentation

6.14.3.1 virtual char* PacketLib::InputText::getLastLineRead () [pure virtual]

Implemented in **PacketLib::InputTextFile** (p. 76).

6.14.3.2 virtual char* PacketLib::InputText::getLine (char * s) throw (PacketExceptionIO*) [pure virtual]

Implemented in **PacketLib::ConfigurationFile** (p. 31).

6.14.3.3 virtual char* PacketLib::InputText::getLine () throw (PacketExceptionIO*) [pure virtual]

Implemented in **PacketLib::ConfigurationFile** (p. 32).

6.14.3.4 bool InputText::isClosed () [virtual]

Definition at line 44 of file InputText.cpp.

References closed.

6.14.3.5 bool InputText::isEOF () [virtual]

Definition at line 54 of file InputText.cpp.

References eof.

6.14.4 Member Data Documentation

6.14.4.1 bool PacketLib::InputText::closed [protected]

Definition at line 76 of file InputText.h.

Referenced by InputText, and isClosed.

6.14.4.2 bool PacketLib::InputText::eof [protected]

Definition at line 72 of file InputText.h.

Referenced by InputText, and isEOF.

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

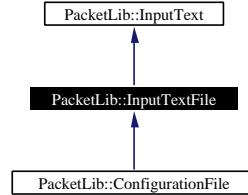
- **InputText.h**
- **InputText.cpp**

6.15 PacketLib::InputTextFile Class Reference

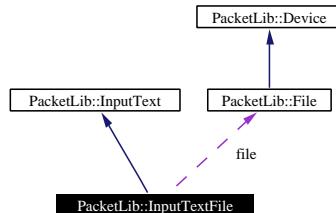
This class represents generic type of input. The input must be a input text file.

```
#include <InputTextFile.h>
```

Inheritance diagram for PacketLib::InputTextFile:



Collaboration diagram for PacketLib::InputTextFile:



Public Methods

- **InputTextFile ()**
- virtual ~**InputTextFile ()**
- virtual bool **open** (char **parameters) throw (PacketExceptionIO*)
- virtual void **close** () throw (PacketExceptionIO*)
- virtual char * **getLine** () throw (PacketExceptionIO*)
- virtual char * **getLine** (char *s) throw (PacketExceptionIO*)
- virtual char * **getLastLineRead** ()
- virtual long **getpos** ()
- virtual bool **memBookmarkPos** ()
- virtual int **setFirstPos** ()
- virtual bool **setLastBookmarkPos** ()
- virtual long **setpos** (long offset) throw (PacketExceptionIO*)
- virtual bool **fchdir** () throw (PacketExceptionIO*)

Protected Attributes

- **File file**

6.15.1 Detailed Description

This class represents generic type of input. The input must be a input text file.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/InputTextFile.h,v 1.5 2003/05/02
08:18:00 agile Exp

Id:

InputTextFile.h,v 1.5 2003/05/02 08:18:00 agile Exp

Revision:

1.5

Definition at line 36 of file InputTextFile.h.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 InputTextFile::InputTextFile ()

Definition at line 24 of file InputTextFile.cpp.

6.15.2.2 InputTextFile::~InputTextFile () [virtual]

Definition at line 34 of file InputTextFile.cpp.

6.15.3 Member Function Documentation

6.15.3.1 void InputTextFile::close () throw (PacketExceptionIO*) [virtual]

Reimplemented in **PacketLib::ConfigurationFile** (p. 31).

Definition at line 80 of file InputTextFile.cpp.

6.15.3.2 bool InputTextFile::fchdir () throw (PacketExceptionIO*) [virtual]

Definition at line 44 of file InputTextFile.cpp.

6.15.3.3 char * InputTextFile::getLastLineRead () [virtual]

Implements **PacketLib::InputText** (p. 73).

Definition at line 126 of file InputTextFile.cpp.

References file.

6.15.3.4 `char * InputTextFile::getLine (char * s) throw
(PacketExceptionIO*) [virtual]`

Implements **PacketLib::InputText** (p. 73).

Reimplemented in **PacketLib::ConfigurationFile** (p. 31).

Definition at line 109 of file InputTextFile.cpp.

6.15.3.5 `char * InputTextFile::getLine () throw
(PacketExceptionIO*) [virtual]`

Implements **PacketLib::InputText** (p. 73).

Reimplemented in **PacketLib::ConfigurationFile** (p. 32).

Definition at line 89 of file InputTextFile.cpp.

6.15.3.6 `long InputTextFile::getpos () [virtual]`

Definition at line 136 of file InputTextFile.cpp.

References file.

6.15.3.7 `bool InputTextFile::memBookmarkPos () [virtual]`

Definition at line 146 of file InputTextFile.cpp.

References file.

6.15.3.8 `bool InputTextFile::open (char ** parameters) throw
(PacketExceptionIO*) [virtual]`

Reimplemented in **PacketLib::ConfigurationFile** (p. 32).

Definition at line 62 of file InputTextFile.cpp.

6.15.3.9 `int InputTextFile::setFirstPos () [virtual]`

Definition at line 156 of file InputTextFile.cpp.

References file.

6.15.3.10 `bool InputTextFile::setLastBookmarkPos () [virtual]`

Definition at line 166 of file InputTextFile.cpp.

References file.

**6.15.3.11 long InputTextFile::setpos (long *offset*) throw
(PacketExceptionIO*) [virtual]**

Definition at line 176 of file InputTextFile.cpp.

6.15.4 Member Data Documentation

6.15.4.1 File PacketLib::InputTextFile::file [protected]

Definition at line 95 of file InputTextFile.h.

Referenced by getLastLineRead, getpos, memBookmarkPos, setFirstPos, and setLastBookmarkPos.

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

- **InputTextFile.h**
- **InputTextFile.cpp**

6.16 PacketLib::MemoryBuffer Class Reference

Class that represent an FIFO structure of char*.

```
#include <MemoryBuffer.h>
```

Public Methods

- **MemoryBuffer ()**
- **~MemoryBuffer ()**
- **void setbuffer (char *_newVal)**
- **void setbuffer (char *_newVal, dword index)**
- **char * getbuffer ()**
- **char * getbuffer (dword index)**
- **void readRewind ()**
- **void writeRewind ()**
- **dword getBufferDimension ()**
- **bool loadBuffer (char *filename) throw (PacketExceptionIO*)**
- **bool saveBuffer (char *filename) throw (PacketExceptionIO*)**

Private Attributes

- **dword dim**
- **char ** buffer**
- **dword indexwrite**
- **dword indexread**
- **char * currentpwd**

6.16.1 Detailed Description

Class that represent an FIFO structure of char*.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/MemoryBuffer.h,v 1.5 2003/05/02
08:18:00 agile Exp

Id:

MemoryBuffer.h,v 1.5 2003/05/02 08:18:00 agile Exp

Revision:

1.5

Definition at line 34 of file MemoryBuffer.h.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 `MemoryBuffer::MemoryBuffer ()`

Definition at line 26 of file `MemoryBuffer.cpp`.

References `buffer`, `dim`, `PacketLib::dword`, `indexread`, and `indexwrite`.

6.16.2.2 `MemoryBuffer::~MemoryBuffer ()`

Definition at line 41 of file `MemoryBuffer.cpp`.

References `buffer`, `dim`, and `PacketLib::dword`.

6.16.3 Member Function Documentation

6.16.3.1 `char * MemoryBuffer::getbuffer (dword index)`

Definition at line 70 of file `MemoryBuffer.cpp`.

References `buffer`, `dim`, and `PacketLib::dword`.

6.16.3.2 `char * MemoryBuffer::getbuffer ()`

Read property of `char** buffer`.

Definition at line 54 of file `MemoryBuffer.cpp`.

References `buffer`, `dim`, and `indexread`.

6.16.3.3 `dword MemoryBuffer::getBufferDimension ()`

Definition at line 153 of file `MemoryBuffer.cpp`.

References `PacketLib::dword`, and `indexwrite`.

6.16.3.4 `bool MemoryBuffer::loadBuffer (char * filename) throw (PacketExceptionIO*)`

Definition at line 163 of file `MemoryBuffer.cpp`.

6.16.3.5 `void MemoryBuffer::readRewind ()`

Definition at line 132 of file `MemoryBuffer.cpp`.

References `indexread`.

**6.16.3.6 bool MemoryBuffer::saveBuffer (char * *filename*) throw
(PacketExceptionIO*)**

Definition at line 197 of file MemoryBuffer.cpp.

**6.16.3.7 void MemoryBuffer::setbuffer (char * *_newVal*, dword
index)**

Definition at line 102 of file MemoryBuffer.cpp.

References buffer, dim, PacketLib::dword, and indexwrite.

6.16.3.8 void MemoryBuffer::setbuffer (char * *_newVal*)

Write property of char** buffer.

Definition at line 87 of file MemoryBuffer.cpp.

References buffer, dim, and indexwrite.

6.16.3.9 void MemoryBuffer::writeRewind ()

Definition at line 142 of file MemoryBuffer.cpp.

References indexwrite.

6.16.4 Member Data Documentation

6.16.4.1 char PacketLib::MemoryBuffer::buffer [private]**

Definition at line 92 of file MemoryBuffer.h.

Referenced by getbuffer, MemoryBuffer, setbuffer, and ~MemoryBuffer.

6.16.4.2 char* PacketLib::MemoryBuffer::currentpwd [private]

Definition at line 104 of file MemoryBuffer.h.

6.16.4.3 dword PacketLib::MemoryBuffer::dim [private]

Definition at line 88 of file MemoryBuffer.h.

Referenced by getbuffer, MemoryBuffer, setbuffer, and ~MemoryBuffer.

6.16.4.4 dword PacketLib::MemoryBuffer::indexread [private]

Definition at line 100 of file MemoryBuffer.h.

Referenced by getbuffer, MemoryBuffer, and readRewind.

6.16.4.5 dword PacketLib::MemoryBuffer::indexwrite [private]

Definition at line 96 of file MemoryBuffer.h.

Referenced by getBufferDimension, MemoryBuffer, setbuffer, and writeRewind.

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

- **MemoryBuffer.h**
- **MemoryBuffer.cpp**

6.17 PacketLib::mesgbuf Struct Reference

```
#include <MSGQ.h>
```

Public Attributes

- long **mtype**
- char **mtext** [2048]

6.17.1 Member Data Documentation

6.17.1.1 char PacketLib::mesgbuf::mtext[2048]

Definition at line 36 of file MSGQ.h.

6.17.1.2 long PacketLib::mesgbuf::mtype

Definition at line 35 of file MSGQ.h.

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

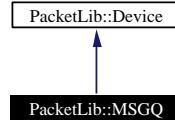
- [MSGQ.h](#)

6.18 PacketLib::MSGQ Class Reference

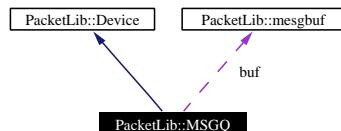
Represent a shared queue.

```
#include <MSGQ.h>
```

Inheritance diagram for PacketLib::MSGQ:



Collaboration diagram for PacketLib::MSGQ:



Public Methods

- **MSGQ (bool bigEndian)**
- virtual ~**MSGQ ()**
- virtual bool **create** (long shqkey, long dim) throw (PacketExceptionIO*)
- virtual bool **open** (long shqkey, long dim) throw (PacketExceptionIO*)
- virtual bool **close ()** throw (PacketExceptionIO*)
- virtual bool **destroy ()** throw (PacketExceptionIO*)
- virtual bool **isCreated ()** const
- virtual void **writeMessage** (char *msg, long mtype) throw (PacketExceptionIO*)
- virtual char * **readMessage** (long mtype) throw (PacketExceptionIO*)

Public Attributes

- int **msgqid**
- long **msgqkey**
- bool **created**
- long **dimmsg**
- **msgbuf buf**

6.18.1 Detailed Description

Represent a shared queue.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/MSGQ.h,v 1.3 2003/05/02
08:18:00 agile Exp

Id:

MSGQ.h,v 1.3 2003/05/02 08:18:00 agile Exp

Revision:

1.3

Definition at line 46 of file MSGQ.h.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 MSGQ::MSGQ (*bool big endian*)

Definition at line 24 of file MSGQ.cpp.

References created.

6.18.2.2 MSGQ::~MSGQ () [virtual]

Definition at line 32 of file MSGQ.cpp.

6.18.3 Member Function Documentation

6.18.3.1 bool MSGQ::close () throw (PacketExceptionIO*) [virtual]

Definition at line 108 of file MSGQ.cpp.

6.18.3.2 bool MSGQ::create (long *shqkey*, long *dim*) throw (PacketExceptionIO*) [virtual]

Definition at line 40 of file MSGQ.cpp.

References PERMMSGQ.

6.18.3.3 bool MSGQ::destroy () throw (PacketExceptionIO*) [virtual]

Definition at line 118 of file MSGQ.cpp.

6.18.3.4 bool MSGQ::isCreated () const [virtual]

Definition at line 138 of file MSGQ.cpp.

References created.

6.18.3.5 bool MSGQ::open (long *shqkey*, long *dim*) throw (PacketExceptionIO*) [virtual]

Definition at line 75 of file MSGQ.cpp.

References PERMMSGQ.

6.18.3.6 char * MSGQ::readMessage (long *mtype*) throw (PacketExceptionIO*) [virtual]

Definition at line 169 of file MSGQ.cpp.

6.18.3.7 void MSGQ::writeMessage (char * *msg*, long *mtype*) throw (PacketExceptionIO*) [virtual]

Definition at line 146 of file MSGQ.cpp.

6.18.4 Member Data Documentation

6.18.4.1 msgbuf PacketLib::MSGQ::buf

Definition at line 106 of file MSGQ.h.

6.18.4.2 bool PacketLib::MSGQ::created

Definition at line 98 of file MSGQ.h.

Referenced by isCreated, and MSGQ.

6.18.4.3 long PacketLib::MSGQ::dimmsg

Definition at line 102 of file MSGQ.h.

6.18.4.4 int PacketLib::MSGQ::msgqid

Definition at line 90 of file MSGQ.h.

6.18.4.5 long PacketLib::MSGQ::msgqkey

Definition at line 94 of file MSGQ.h.

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

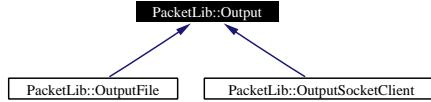
- **MSGQ.h**
- **MSGQ.cpp**

6.19 PacketLib::Output Class Reference

This class represents generic type of output.

```
#include <Output.h>
```

Inheritance diagram for PacketLib::Output:



Public Methods

- **Output (bool bigEndian)**
- virtual void **close ()=0** throw (PacketExceptionIO*)
- virtual bool **open (char **parameters)=0** throw (PacketExceptionIO*)
- virtual bool **isClosed ()**
- virtual bool **isBigEndian ()**
- virtual bool **writeByteStream (ByteStream *b)=0** throw (PacketExceptionIO*)
- virtual bool **writeString (char *str)=0** throw (PacketExceptionIO*)
- virtual int **getType ()=0**

Protected Attributes

- bool **isclosed**
- bool **bigEndian**

6.19.1 Detailed Description

This class represents generic type of output.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/Output.h,v 1.4 2003/05/02 08:18:00 agile Exp

Id:

Output.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 35 of file Output.h.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 Output::Output (bool *bigendian*)

Definition at line 7 of file Output.cpp.

References bigendian, and isclosed.

6.19.3 Member Function Documentation

6.19.3.1 virtual void PacketLib::Output::close () throw (PacketExceptionIO*) [pure virtual]

Implemented in **PacketLib::OutputFile** (p. 92).

6.19.3.2 virtual int PacketLib::Output::getType () [pure virtual]

Implemented in **PacketLib::OutputFile** (p. 92).

6.19.3.3 bool Output::isBigEndian () [virtual]

Definition at line 26 of file Output.cpp.

References bigendian.

6.19.3.4 bool Output::isClosed () [virtual]

Definition at line 16 of file Output.cpp.

References isclosed.

6.19.3.5 virtual bool PacketLib::Output::open (char ** *parameters*) throw (PacketExceptionIO*) [pure virtual]

Implemented in **PacketLib::OutputFile** (p. 92).

6.19.3.6 virtual bool PacketLib::Output::writeByteStream (ByteStream * *b*) throw (PacketExceptionIO*) [pure virtual]

Implemented in **PacketLib::OutputFile** (p. 93).

6.19.3.7 virtual bool PacketLib::Output::writeString (char * *str*) throw (PacketExceptionIO*) [pure virtual]

Implemented in **PacketLib::OutputFile** (p. 93).

6.19.4 Member Data Documentation

6.19.4.1 bool PacketLib::Output::bigEndian [protected]

Definition at line 78 of file Output.h.

Referenced by isBigEndian, Output, and PacketLib::OutputFile::OutputFile.

6.19.4.2 bool PacketLib::Output::isClosed [protected]

Definition at line 74 of file Output.h.

Referenced by isClosed, and Output.

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

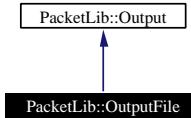
- [Output.h](#)
- [Output.cpp](#)

6.20 PacketLib::OutputFile Class Reference

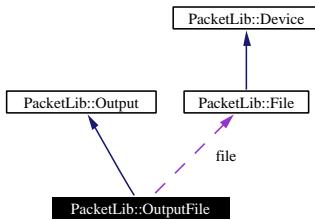
This class represents generic type of output. The output must be a file.

```
#include < outputFile.h >
```

Inheritance diagram for PacketLib::OutputFile:



Collaboration diagram for PacketLib::OutputFile:



Public Methods

- **OutputFile (bool bigEndian)**
- virtual ~**OutputFile ()**
- virtual void **close ()** throw (PacketExceptionIO*)
- virtual bool **open (char **parameters)** throw (PacketExceptionIO*)
- virtual bool **writeByteStream (ByteStream *b)** throw (PacketExceptionIO*)
- virtual bool **writeString (char *str)** throw (PacketExceptionIO*)
- virtual int **getType ()**

Protected Attributes

- **File * file**
- **char * filename**

6.20.1 Detailed Description

This class represents generic type of output. The output must be a file.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/OutputFile.h,v 1.4 2003/05/02
08:18:00 agile Exp

Id:

OutputFile.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 35 of file OutputFile.h.

6.20.2 Constructor & Destructor Documentation

6.20.2.1 OutputFile::OutputFile (bool *bigendian*)

Definition at line 24 of file OutputFile.cpp.

References PacketLib::Output::bigEndian, and file.

6.20.2.2 OutputFile::~OutputFile () [virtual]

Definition at line 34 of file OutputFile.cpp.

References file.

6.20.3 Member Function Documentation

6.20.3.1 void OutputFile::close () throw (PacketExceptionIO*) [virtual]

Implements **PacketLib::Output** (p. 89).

Definition at line 44 of file OutputFile.cpp.

6.20.3.2 virtual int PacketLib::OutputFile::getType () [inline, virtual]

Implements **PacketLib::Output** (p. 89).

Definition at line 66 of file OutputFile.h.

6.20.3.3 bool OutputFile::open (char ** *parameters*) throw (PacketExceptionIO*) [virtual]

Implements **PacketLib::Output** (p. 89).

Definition at line 55 of file OutputFile.cpp.

**6.20.3.4 bool OutputFile::writeByteStream (ByteStream * *b*)
throw (PacketExceptionIO*) [virtual]**

Implements **PacketLib::Output** (p. 89).

Definition at line 68 of file OutputFile.cpp.

**6.20.3.5 bool OutputFile::writeString (char * *str*) throw
(PacketExceptionIO*) [virtual]**

Implements **PacketLib::Output** (p. 89).

Definition at line 83 of file OutputFile.cpp.

6.20.4 Member Data Documentation

6.20.4.1 File* PacketLib::OutputFile::file [protected]

Definition at line 72 of file OutputFile.h.

Referenced by OutputFile, and ~OutputFile.

6.20.4.2 char* PacketLib::OutputFile::filename [protected]

Definition at line 76 of file OutputFile.h.

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

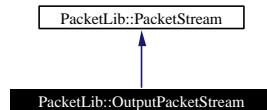
- **OutputFile.h**
- **OutputFile.cpp**

6.21 PacketLib::OutputPacketStream Class Reference

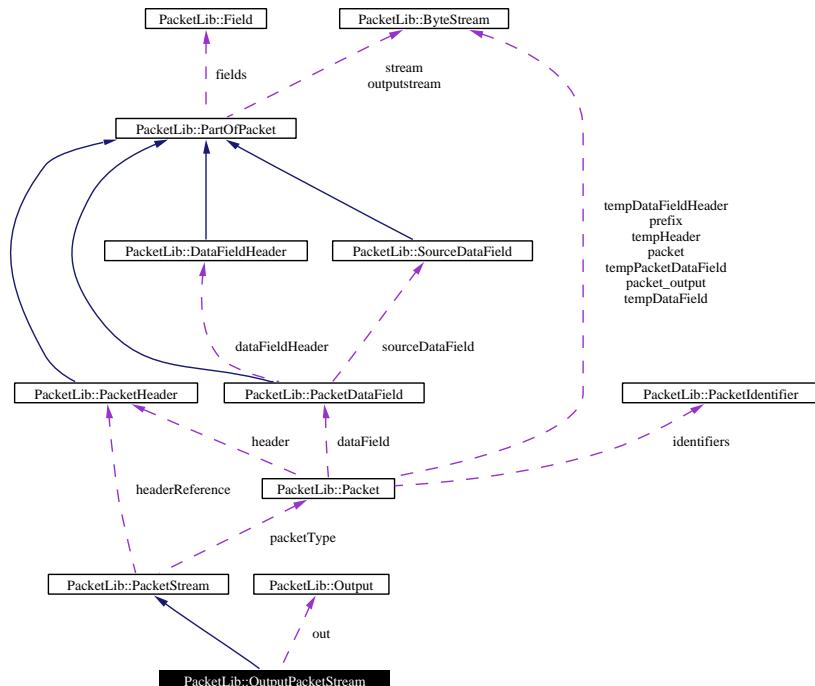
This class represents generic output of stream if packet.

```
#include <OutputPacketStream.h>
```

Inheritance diagram for PacketLib::OutputPacketStream:



Collaboration diagram for PacketLib::OutputPacketStream:



Public Methods

- **OutputPacketStream ()**
 - **OutputPacketStream (char *fileNameConfig)**
 - **bool writePacket (Packet *p) throw (PacketExceptionIO*)**
 - **void setOutput (Output *out)**

Protected Attributes

- **Output * out**

6.21.1 Detailed Description

This class represents generic output of stream if packet.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/OutputPacketStream.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

OutputPacketStream.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 35 of file OutputPacketStream.h.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 OutputPacketStream::OutputPacketStream ()

Definition at line 7 of file OutputPacketStream.cpp.

References out.

6.21.2.2 OutputPacketStream::OutputPacketStream (char * *fileNameConfig*)

Definition at line 17 of file OutputPacketStream.cpp.

References out.

6.21.3 Member Function Documentation

6.21.3.1 void OutputPacketStream::setOutput (Output * *out*)

Definition at line 52 of file OutputPacketStream.cpp.

References out.

**6.21.3.2 bool OutputPacketStream::writePacket (Packet * *p*) throw
(PacketExceptionIO*)**

Definition at line 27 of file OutputPacketStream.cpp.

6.21.4 Member Data Documentation**6.21.4.1 Output* PacketLib::OutputPacketStream::out
[protected]**

Definition at line 59 of file OutputPacketStream.h.

Referenced by OutputPacketStream, and setOutput.

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

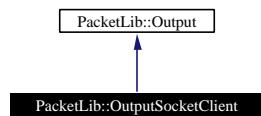
- **OutputPacketStream.h**
- **OutputPacketStream.cpp**

6.22 PacketLib::OutputSocketClient Class Reference

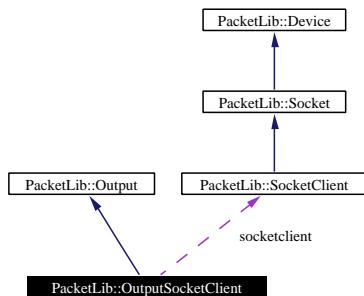
This class represents generic type of output. The output must be a file.

```
#include <OutputSocketClient.h>
```

Inheritance diagram for PacketLib::OutputSocketClient:



Collaboration diagram for PacketLib::OutputSocketClient:



Public Methods

- **OutputSocketClient** (bool bigendian)
- virtual ~**OutputSocketClient** ()
- virtual void **close** () throw (PacketExceptionIO*)
- virtual bool **open** (char **parameters) throw (PacketExceptionIO*)
- virtual bool **writeByteStream** (ByteStream *b) throw (PacketExceptionIO*)
- virtual bool **writeString** (char *str) throw (PacketExceptionIO*)
- virtual int **getType** ()

Protected Attributes

- **SocketClient * socketclient**
- **char * host**
- **int port**

6.22.1 Detailed Description

This class represents generic type of output. The output must be a file.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/OutputSocketClient.h,v 1.2
2003/05/02 08:18:00 agile Exp

Id:

OutputSocketClient.h,v 1.2 2003/05/02 08:18:00 agile Exp

Revision:

1.2

Definition at line 35 of file OutputSocketClient.h.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 OutputSocketClient::OutputSocketClient (bool *big endian*)

Definition at line 24 of file OutputSocketClient.cpp.

References host, and socketclient.

6.22.2.2 OutputSocketClient::~OutputSocketClient () [virtual]

Definition at line 35 of file OutputSocketClient.cpp.

References host, and socketclient.

6.22.3 Member Function Documentation

6.22.3.1 void OutputSocketClient::close () throw (PacketExceptionIO*) [virtual]

Implements **PacketLib::Output** (p. 89).

Definition at line 46 of file OutputSocketClient.cpp.

6.22.3.2 virtual int PacketLib::OutputSocketClient::getType () [inline, virtual]

Implements **PacketLib::Output** (p. 89).

Definition at line 65 of file OutputSocketClient.h.

**6.22.3.3 bool OutputSocketClient::open (char ** *argv*) throw
(PacketExceptionIO*) [virtual]**

Implements **PacketLib::Output** (p. 89).

Definition at line 56 of file OutputSocketClient.cpp.

**6.22.3.4 bool OutputSocketClient::writeByteStream (ByteStream *
b) throw (PacketExceptionIO*) [virtual]**

Implements **PacketLib::Output** (p. 89).

Definition at line 70 of file OutputSocketClient.cpp.

**6.22.3.5 bool OutputSocketClient::writeString (char * *str*) throw
(PacketExceptionIO*) [virtual]**

Implements **PacketLib::Output** (p. 89).

Definition at line 85 of file OutputSocketClient.cpp.

6.22.4 Member Data Documentation

6.22.4.1 char* PacketLib::OutputSocketClient::host [protected]

Definition at line 75 of file OutputSocketClient.h.

Referenced by OutputSocketClient, and ~OutputSocketClient.

6.22.4.2 int PacketLib::OutputSocketClient::port [protected]

Definition at line 79 of file OutputSocketClient.h.

**6.22.4.3 SocketClient* PacketLib::OutputSocketClient::socketclient
[protected]**

Definition at line 71 of file OutputSocketClient.h.

Referenced by OutputSocketClient, and ~OutputSocketClient.

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

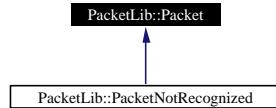
- **OutputSocketClient.h**
- **OutputSocketClient.cpp**

6.23 PacketLib::Packet Class Reference

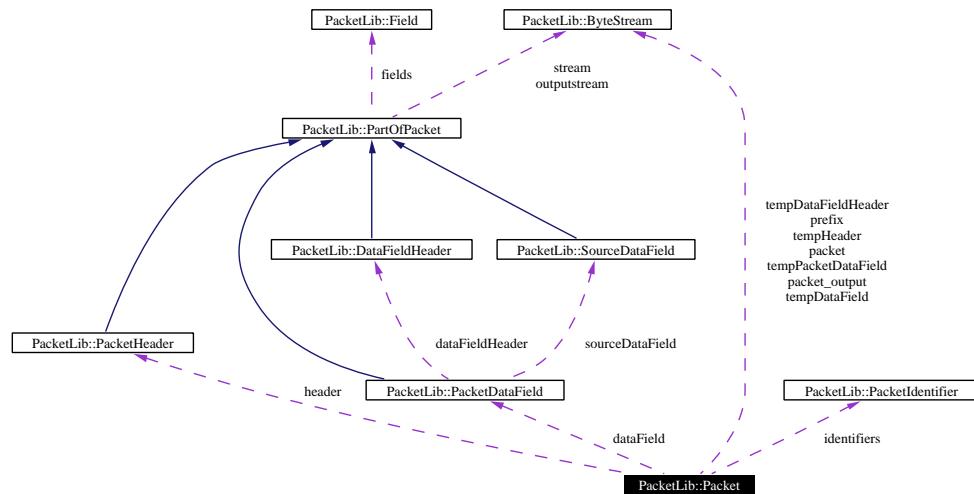
Represent a single packet.

```
#include <Packet.h>
```

Inheritance diagram for PacketLib::Packet:



Collaboration diagram for PacketLib::Packet:



Public Methods

- **Packet (bool bigEndian)**
- virtual ~**Packet ()**
- virtual bool **createPacketType** (char *fileName, bool prefix, word dimprefix) throw (PacketException*)
- virtual bool **setAndVerifyPacketValue** (ByteStream *prefix, ByteStream *packetHeader, ByteStream *packetDataField)
- virtual bool **setAndVerifyPacketValue** (ByteStream *prefix, ByteStream *packet)
- virtual bool **setPacketValue** (ByteStream *prefix, ByteStream *packetHeader, ByteStream *packetDataField)
- virtual bool **setPacketValue** (ByteStream *prefix, ByteStream *packet)

- `bool verifyPacketValue (ByteStream *prefix, ByteStream *packet)`
- `bool verifyPacketValue (ByteStream *prefix, ByteStream *packetHeader, ByteStream *packetDataField)`
- `virtual void printPacketValue ()`
- `word getDimension ()`
- `word getMaxDimension ()`
- `virtual char * getName ()`
- `virtual void printIdentifiers ()`
- `virtual void deleteExternalByteStream ()`
- `ByteStream * getOutputStream ()`
- `virtual void setPacketID (const byte &_newVal)`
- `virtual const byte & getPacketID ()`
- `bool thereIsPrefix ()`

Public Attributes

- `PacketHeader * header`
- `PacketDataField * dataField`
- `ByteStream * prefix`
- `ByteStream * packet`
- `ByteStream * packet_output`

Protected Methods

- `virtual void generateStream ()`
- `virtual bool loadIdentifiers (ConfigurationFile &file)`
- `bool setPacketValuePrefix (ByteStream *prefix)`
- `bool setPacketValueVerify (ByteStream *prefix, ByteStream *packetHeader, ByteStream *packetDataField)`
- `bool setPacketValueDataFieldHeader (ByteStream *packetDataField)`
- `bool setPacketValueSourceDataField (ByteStream *packetDataField)`
- `bool setPacketValueHeader (ByteStream *header)`
- `void memByteStream (ByteStream *prefix, ByteStream *packetHeader, ByteStream *packetDataField)`
- `void memByteStream (ByteStream *prefix, ByteStream *packet)`

Protected Attributes

- `byte packetID`
- `char * name`
- `PacketIdentifier ** identifiers`
- `bool type_of_identifier [3]`
- `unsigned number_of_identifier`

- bool **bigEndian**
- bool **thereIsPrefix**
- word **dimPrefix**

Private Attributes

- ByteStream * **tempHeader**
- ByteStream * **tempDataField**
- ByteStream * **tempDataFieldHeader**
- ByteStream * **tempPacketDataField**
- bool **first_output_stream_setted**

6.23.1 Detailed Description

Represent a single packet.

Memory management: the ByteStrem passed to an object of this class isn't delete to this class. The responsibility of delete the **ByteStream** (p. 21) is of the caller. For deleting the external **ByteStream** (p. 21) of this class it's possible to call the **deleteExternalByteStream()** (p. 103) method.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/	Packet.h,v	1.6	2003/05/02
08:18:00 agile Exp			

Id:

Packet.h,v 1.6 2003/05/02 08:18:00 agile Exp

Revision:

1.6

Definition at line 50 of file Packet.h.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 Packet::Packet (bool *bigEndian*)

Constructor

Definition at line 8 of file Packet.cpp.

References bigEndian, dataField, first_output_stream_setted, header, identifiers, name, number_of_identifier, packet, packet_output, prefix, tempDataField, tempDataFieldHeader, tempHeader, and tempPacketDataField.

6.23.2.2 Packet::~Packet () [virtual]

Distruttore

Definition at line 34 of file Packet.cpp.

References dataField, header, identifiers, name, number_of_identifier, packet_output, tempDataField, tempDataFieldHeader, tempHeader, and tempPacketDataField.

6.23.3 Member Function Documentation

6.23.3.1 bool Packet::createPacketType (char * *fileName*, bool *isprefix*, word *dimprefix*) throw (PacketException*) [virtual]

Reimplemented in **PacketLib::PacketNotRecognized** (p. 130).

Definition at line 55 of file Packet.cpp.

References PRINTDEBUG, and PacketLib::word.

6.23.3.2 void Packet::deleteExternalByteStream () [virtual]

Deletes the **ByteStream** (p. 21) passed as arguments.

Definition at line 601 of file Packet.cpp.

References dataField, header, packet, and prefix.

6.23.3.3 void Packet::generateStream () [protected, virtual]

Generates the stream for output.

Precondition:

the number of block and the number of element for each block must be setted (if applicable)

Definition at line 473 of file Packet.cpp.

References big endian, dataField, first_output_stream_setted, header, packet_output, thereisprefix, and PacketLib::word.

Referenced by getOutputStream.

6.23.3.4 word Packet::getDimension ()

Gets the dimension in byte of the current packet.

Definition at line 453 of file Packet.cpp.

References dataField, header, and PacketLib::word.

Referenced by `getOutputStream`.

6.23.3.5 word Packet::getMaxDimension ()

Gets the max dimension in byte of the packet.

Definition at line 463 of file `Packet.cpp`.

References `dataField`, `header`, and `PacketLib::word`.

6.23.3.6 virtual char* PacketLib::Packet::getName () [inline, virtual]

Gets the name of packet.

Definition at line 165 of file `Packet.h`.

References `name`.

6.23.3.7 ByteStream * Packet::getOutputStream ()

Gets the packet generated with the `generateStream()` (p. 103) method.

Definition at line 653 of file `Packet.cpp`.

References `bigEndian`, `generateStream`, `getDimension`, `packet_output`, and `thereisprefix`.

6.23.3.8 const byte & Packet::getPacketID () [virtual]

Read property of byte `packetID`.

Definition at line 666 of file `Packet.cpp`.

References `PacketLib::byte`, and `packetID`.

6.23.3.9 bool Packet::loadIdentifiers (ConfigurationFile & *file*) [protected, virtual]

Loads the configuration of identifiers from file `.packet`.

Definition at line 206 of file `Packet.cpp`.

References `PacketLib::byte`, `identifiers`, `number_of_identifier`, `type_of_identifier`, and `PacketLib::word`.

6.23.3.10 void Packet::memByteStream (ByteStream * *prefix*, ByteStream * *packet*) [protected]

Definition at line 630 of file `Packet.cpp`.

References `packet`, and `prefix`.

**6.23.3.11 void Packet::memByteStream (ByteStream *
prefix, ByteStream * *packetHeader*, ByteStream *
packetDataField) [protected]**

Definition at line 641 of file Packet.cpp.

References dataField, header, and prefix.

Referenced by setAndVerifyPacketValue, setPacketValue, and verifyPacketValue.

6.23.3.12 void Packet::printIdentifiers () [virtual]

Prints to stdout the identifiers of the packet.

Definition at line 246 of file Packet.cpp.

References number_of_identifier.

6.23.3.13 void Packet::printPacketValue () [virtual]

Prints to stdout the value of packet data field in a formatted mode.

Definition at line 432 of file Packet.cpp.

References dataField, header, and PRINTDEBUG.

**6.23.3.14 bool Packet::setAndVerifyPacketValue (ByteStream *
prefix, ByteStream * *packet*) [virtual]**

This method verifies if the **ByteStream** (p. 21) on argument contains the correct value in the identifiers. If this is true, the method returns true and the stream contains a packet of this type. This method overloads another method.

Postcondition:

A side effect of this method is that the values of fields of packet are setted with correct value

Definition at line 385 of file Packet.cpp.

References header, memByteStream, packet, prefix, setAndVerifyPacketValue, tempDataField, tempHeader, and PacketLib::word.

**6.23.3.15 bool Packet::setAndVerifyPacketValue (ByteStream
* *prefix*, ByteStream * *packetHeader*, ByteStream *
packetDataField) [virtual]**

This method verifies if the **ByteStream** (p. 21) on argument contains the correct value in the identifiers. If this is true, the method return true and the stream contains a packet of this type. This method overloads another method.

Postcondition:

A side effects of this method is that the value of fields of packet are setted with correct value.

Definition at line 399 of file Packet.cpp.

References memByteStream, prefix, setPacketValueDataFieldHeader, setPacketValueHeader, setPacketValueSourceDataField, type_of_identifier, and verifyPacketValue.

Referenced by setAndVerifyPacketValue.

6.23.3.16 void Packet::setPacketID (const byte & *_newVal*) [virtual]

Writes property of byte packetID.

Definition at line 677 of file Packet.cpp.

References PacketLib::byte, and packetID.

6.23.3.17 bool Packet::setPacketValue (ByteStream * *prefix*, ByteStream * *packet*) [virtual]

Sets all the fields of the packet with correct value contained into the input **ByteStream** (p. 21).

Precondition:

The structure of the stream must be loaded.

Parameters:

prefix This is the prefix of the packet

packet This is the packet

Postcondition:

If return is true all the fields are setted with the correct value.

Definition at line 295 of file Packet.cpp.

References header, memByteStream, packet, prefix, setPacketValue, tempDataField, tempHeader, and PacketLib::word.

6.23.3.18 bool Packet::setPacketValue (ByteStream * *prefix*, ByteStream * *packetHeader*, ByteStream * *packetDataField*) [virtual]

Sets all the fields of the packet with correct value contained into the input **ByteStream** (p. 21).

Precondition:

The structure of the stream must be loaded.

Parameters:

prefix This is the prefix of the packet

packetHeader This is the header of the packet

packetDataField This is the data field of the packet

Postcondition:

If return is true all the fields are setted with the correct value.

Reimplemented in **PacketLib::PacketNotRecognized** (p. 131).

Definition at line 265 of file Packet.cpp.

References memByteStream, prefix, setPacketValueDataFieldHeader, setPacketValueHeader, setPacketValuePrefix, setPacketValueSourceDataField, and setPacketValueVerify.

Referenced by setPacketValue, and verifyPacketValue.

**6.23.3.19 bool Packet::setPacketValueDataFieldHeader
(ByteStream * *packetDataField*) [protected]**

Definition at line 534 of file Packet.cpp.

References dataField, tempDataFieldHeader, and PacketLib::word.

Referenced by setAndVerifyPacketValue, setPacketValue, and verifyPacketValue.

**6.23.3.20 bool Packet::setPacketValueHeader (ByteStream *
packetHeader) [protected]**

Definition at line 586 of file Packet.cpp.

References header.

Referenced by setAndVerifyPacketValue, setPacketValue, and verifyPacketValue.

**6.23.3.21 bool Packet::setPacketValuePrefix (ByteStream * *prefix*)
[protected]**

Definition at line 521 of file Packet.cpp.

References prefix.

Referenced by setPacketValue, and verifyPacketValue.

**6.23.3.22 bool Packet::setPacketValueSourceDataField
(ByteStream * *packetDataField*) [protected]**

Definition at line 556 of file Packet.cpp.

References `dataField`, `header`, `tempPacketDataField`, and `PacketLib::word`.

Referenced by `setAndVerifyPacketValue`, and `setPacketValue`.

6.23.3.23 `bool Packet::setPacketValueVerify (ByteStream * prefix, ByteStream * packetHeader, ByteStream * packetDataField) [protected]`

Definition at line 504 of file `Packet.cpp`.

References `header`.

Referenced by `setPacketValue`, and `verifyPacketValue`.

6.23.3.24 `bool Packet::thereIsPrefix ()`

Definition at line 687 of file `Packet.cpp`.

References `thereisprefix`.

6.23.3.25 `bool Packet::verifyPacketValue (ByteStream * prefix, ByteStream * packetHeader, ByteStream * packetDataField)`

Verifies if within the **ByteStream** (p. 21) passed with arguments it's present a correct packet.

Precondition:

The structure of the stream must be loaded.

Parameters:

prefix This is the prefix of the packet.

packetHeader This is the header of the packet.

packetDataField This is the data field of the packet.

Returns:

True if the **ByteStream** (p. 21) contains a packet.

Definition at line 323 of file `Packet.cpp`.

References `dataField`, `header`, `identifiers`, `memByteStream`, `number_of_identifier`, `prefix`, `setPacketValue`, `setPacketValueDataFieldHeader`, `setPacketValueHeader`, `setPacketValuePrefix`, `setPacketValueVerify`, and `type_of_identifier`.

6.23.3.26 `bool Packet::verifyPacketValue (ByteStream * prefix, ByteStream * packet)`

Verifies if within the **ByteStream** (p. 21) passed with arguments it's present a correct packet.

Precondition:

The structure of the stream must be loaded.

Parameters:

prefix This is the prefix of the packet

packet This is the packet

Returns:

True if the **ByteStream** (p. 21) contains a packet

Definition at line 309 of file Packet.cpp.

References header, memByteStream, packet, prefix, tempDataField, tempHeader, and PacketLib::word.

Referenced by setAndVerifyPacketValue.

6.23.4 Member Data Documentation

6.23.4.1 bool PacketLib::Packet::bigEndian [protected]

Definition at line 303 of file Packet.h.

Referenced by generateStream, getOutputStream, and Packet.

6.23.4.2 PacketDataField* PacketLib::Packet::dataField

This attribute represent the packet data field.

Definition at line 209 of file Packet.h.

Referenced by deleteExternalByteStream, generateStream, getDimension, getMaxDimension, memByteStream, Packet, PacketLib::PacketNotRecognized::PacketNotRecognized, printPacketValue, PacketLib::PacketNotRecognized::setPacketValue, setPacketValueDataFieldHeader, setPacketValueSourceDataField, verifyPacketValue, and ~Packet.

6.23.4.3 word PacketLib::Packet::dimPrefix [protected]

Definition at line 311 of file Packet.h.

6.23.4.4 bool PacketLib::Packet::first_output_stream_setted [private]

Definition at line 333 of file Packet.h.

Referenced by generateStream, and Packet.

6.23.4.5 PacketHeader* PacketLib::Packet::header

This attribute represents the packet header.

Definition at line 204 of file Packet.h.

Referenced by deleteExternalByteStream, generateStream, getDimension, getMaxDimension, memByteStream, Packet, printPacketValue, setAndVerifyPacketValue, PacketLib::PacketNotRecognized::setPacketValue, setPacketValue, setPacketValueHeader, setPacketValueSourceDataField, setPacketValueVerify, verifyPacketValue, and ~Packet.

6.23.4.6 PacketIdentifier PacketLib::Packet::identifiers [protected]**

List of identifiers. This identifiers permits to identify if the stream contains a particular type of packet

Definition at line 260 of file Packet.h.

Referenced by loadIdentifiers, Packet, verifyPacketValue, and ~Packet.

6.23.4.7 char* PacketLib::Packet::name [protected]

The name of packet

Definition at line 253 of file Packet.h.

Referenced by getName, Packet, and ~Packet.

6.23.4.8 unsigned PacketLib::Packet::number_of_identifier [protected]

Definition at line 271 of file Packet.h.

Referenced by loadIdentifiers, Packet, printIdentifiers, verifyPacketValue, and ~Packet.

6.23.4.9 ByteStream* PacketLib::Packet::packet

Whitin this ByteStrem is present the packet.

Definition at line 221 of file Packet.h.

Referenced by deleteExternalByteStream, memByteStream, Packet, setAndVerifyPacketValue, setPacketValue, and verifyPacketValue.

6.23.4.10 ByteStream* PacketLib::Packet::packet_output

Definition at line 225 of file Packet.h.

Referenced by generateStream, getOutputStream, Packet, and ~Packet.

6.23.4.11 byte PacketLib::Packet::packetID [protected]

This is the number that identifies the packet in the .stream file

Definition at line 247 of file Packet.h.

Referenced by getPacketID, and setPacketID.

6.23.4.12 ByteStream* PacketLib::Packet::prefix

Within this ByteStream is present prefix of packet.

Definition at line 215 of file Packet.h.

Referenced by deleteExternalByteStream, memByteStream, Packet, setAndVerifyPacketValue, PacketLib::PacketNotRecognized::setPacketValue, setPacketValue, setPacketValuePrefix, and verifyPacketValue.

6.23.4.13 ByteStream* PacketLib::Packet::tempDataField [private]

Definition at line 321 of file Packet.h.

Referenced by Packet, setAndVerifyPacketValue, setPacketValue, verifyPacketValue, and ~Packet.

6.23.4.14 ByteStream* PacketLib::Packet::tempDataFieldHeader [private]

Definition at line 325 of file Packet.h.

Referenced by Packet, setPacketValueDataFieldHeader, and ~Packet.

6.23.4.15 ByteStream* PacketLib::Packet::tempHeader [private]

Definition at line 317 of file Packet.h.

Referenced by Packet, setAndVerifyPacketValue, setPacketValue, verifyPacketValue, and ~Packet.

6.23.4.16 ByteStream* PacketLib::Packet::tempPacketDataField [private]

Definition at line 329 of file Packet.h.

Referenced by Packet, setPacketValueSourceDataField, and ~Packet.

6.23.4.17 bool PacketLib::Packet::thereisprefix [protected]

Definition at line 307 of file Packet.h.

Referenced by generateStream, getOutputStream, and thereIsPrefix.

6.23.4.18 bool PacketLib::Packet::type_of_identifier[3] [protected]

List of bool that indicates which part of packet are presents into one or more identifiers. 0 is the header, 1 is the data field header, 2 is the source data field
Definition at line 267 of file Packet.h.

Referenced by loadIdentifiers, setAndVerifyPacketValue, and verifyPacketValue.

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

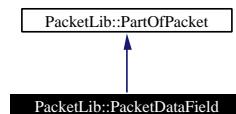
- **Packet.h**
- **Packet.cpp**

6.24 PacketLib::PacketDataField Class Reference

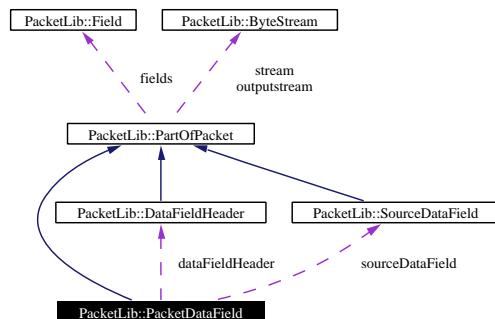
Represent the data field of the packet.

```
#include <PacketDataField.h>
```

Inheritance diagram for PacketLib::PacketDataField:



Collaboration diagram for PacketLib::PacketDataField:



Public Methods

- **PacketDataField ()**
- **~PacketDataField ()**
- **word getNumberOfRealDataBlock ()**
- **void setNumberOfRealDataBlock (word number)**
- **word getDimension ()**
- **word getMaxDimension ()**
- **virtual bool setOutputStream (ByteStream *os, word first)**
- **virtual ByteStream * generateStream (bool bigendian)**

Public Attributes

- **DataFieldHeader * dataFieldHeader**
- **SourceDataField * sourceDataField**

6.24.1 Detailed Description

Represent the data field of the packet.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/PacketDataField.h,v
2003/05/02 08:18:00 agile Exp

1.4

Id:

PacketDataField.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 36 of file PacketDataField.h.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 PacketDataField::PacketDataField ()

Constructor

Definition at line 7 of file PacketDataField.cpp.

References dataFieldHeader, and sourceDataField.

6.24.2.2 PacketDataField::~PacketDataField ()

Definition at line 15 of file PacketDataField.cpp.

References dataFieldHeader, and sourceDataField.

6.24.3 Member Function Documentation

6.24.3.1 *ByteStream * PacketDataField::generateStream (bool bigEndian) [virtual]*

Generates the stream.

Reimplemented from **PacketLib::PartOfPacket** (p. 140).

Definition at line 105 of file PacketDataField.cpp.

References dataFieldHeader, PacketLib::PartOfPacket::outputstream, and sourceDataField.

6.24.3.2 word PacketDataField::getDimension () [virtual]

Total dimension in bytes of data field.

Reimplemented from **PacketLib::PartOfPacket** (p. 140).

Definition at line 26 of file PacketDataField.cpp.

References dataFieldHeader, sourceDataField, and PacketLib::word.

Referenced by setOutputStream.

6.24.3.3 word PacketDataField::getMaxDimension ()

Total max dimension in bytes of data field.

Definition at line 39 of file PacketDataField.cpp.

References dataFieldHeader, sourceDataField, and PacketLib::word.

6.24.3.4 word PacketDataField::getNumberOfRealDataBlock ()

Returns the number of events data block in the source data field. Returns 0 if there aren't blocks.

Definition at line 52 of file PacketDataField.cpp.

References dataFieldHeader, sourceDataField, and PacketLib::word.

6.24.3.5 void PacketDataField::setNumberOfRealDataBlock (word *number*)

Sets the number of data block presents in the source data field. If the number is up of the max number of data block allowed, the number of real data blocks is setted with the max number of blocks.

Definition at line 71 of file PacketDataField.cpp.

References dataFieldHeader, sourceDataField, and PacketLib::word.

6.24.3.6 bool PacketDataField::setOutputStream (ByteStream * *os*, word *first*) [virtual]

Creates the outputstream **ByteStream** (p. 21) for the generation of the output stream

Reimplemented from **PacketLib::PartOfPacket** (p. 143).

Definition at line 91 of file PacketDataField.cpp.

References dataFieldHeader, getDimension, PacketLib::PartOfPacket::outputstream, sourceDataField, and PacketLib::word.

6.24.4 Member Data Documentation

6.24.4.1 **DataFieldHeader* PacketLib::PacketDataField::dataField-Header**

Represents the data field header.

Definition at line 90 of file PacketDataField.h.

Referenced by generateStream, getDimension, getMaxDimension, getNumberOfRealDataBlock, PacketDataField, setNumberOfRealDataBlock, setOutputStream, and ~PacketDataField.

6.24.4.2 **SourceDataField* PacketLib::PacketDataField::source-DataField**

Represents the source data field.

Definition at line 95 of file PacketDataField.h.

Referenced by generateStream, getDimension, getMaxDimension, getNumberOfRealDataBlock, PacketDataField, setNumberOfRealDataBlock, setOutputStream, and ~PacketDataField.

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

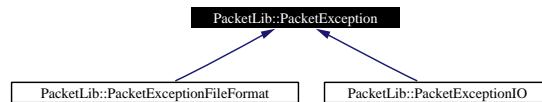
- **PacketDataField.h**
- **PacketDataField.cpp**

6.25 PacketLib::PacketException Class Reference

Exception class of the **PacketLib** (p. 13).

```
#include <PacketException.h>
```

Inheritance diagram for PacketLib::PacketException:



Public Methods

- **PacketException** (char *error, int code=0)
- virtual ~**PacketException** ()
- virtual const char * **geterror** ()
- void **add** (char *error)
- int **geterrorcode** () const
- void **seterrorcode** (int errorcode)

Protected Attributes

- string * **error**
- int **errorcode**

6.25.1 Detailed Description

Exception class of the **PacketLib** (p. 13).

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/	PacketException.h,v	1.5
2003/05/02 08:18:00	agile Exp	

Id:

PacketException.h,v 1.5 2003/05/02 08:18:00 agile Exp

Revision:

1.5

Definition at line 33 of file PacketException.h.

6.25.2 Constructor & Destructor Documentation

6.25.2.1 **PacketException::PacketException** (*char * error, int code = 0*)

Definition at line 24 of file PacketException.cpp.

References error, and errorcode.

6.25.2.2 **PacketException::~PacketException** () [virtual]

Definition at line 35 of file PacketException.cpp.

References error.

6.25.3 Member Function Documentation

6.25.3.1 **void PacketException::add** (*char * err*)

Definition at line 57 of file PacketException.cpp.

References error.

6.25.3.2 **const char * PacketException::geterror** () [virtual]

Definition at line 46 of file PacketException.cpp.

References error.

6.25.3.3 **int PacketException::geterrorcode** () const

Definition at line 65 of file PacketException.cpp.

References errorcode.

6.25.3.4 **void PacketException::seterrorcode** (*int errorcode*)

Definition at line 75 of file PacketException.cpp.

References errorcode.

6.25.4 Member Data Documentation

6.25.4.1 **string* PacketLib::PacketException::error** [protected]

Definition at line 64 of file PacketException.h.

Referenced by add, geterror, PacketException, ~PacketException, PacketLib::PacketExceptionFormatException::~PacketExceptionFormatException, and PacketLib::PacketExceptionIO::~PacketExceptionIO.

6.25.4.2 int PacketLib::PacketException::errorcode [protected]

Definition at line 67 of file PacketException.h.

Referenced by geterrorcode, PacketException, and seterrorcode.

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

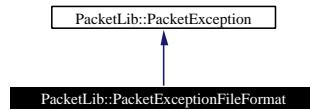
- **PacketException.h**
- **PacketException.cpp**

6.26 PacketLib::PacketExceptionFormatException Class Reference

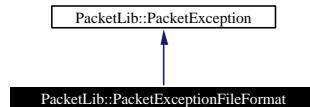
Exception class of the **PacketLib** (p. 13). The error is the bad file format of the configuration files.

```
#include <PacketExceptionFormatException.h>
```

Inheritance diagram for PacketLib::PacketExceptionFormatException:



Collaboration diagram for PacketLib::PacketExceptionFormatException:



Public Methods

- **PacketExceptionFormatException** (char *error)
- virtual ~**PacketExceptionFormatException** ()

6.26.1 Detailed Description

Exception class of the **PacketLib** (p. 13). The error is the bad file format of the configuration files.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/PacketExceptionFormatException.h,v
1.3 2002/11/13 15:16:35 andrea Exp

Id:

PacketExceptionFormatException.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 34 of file PacketExceptionFormatException.h.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 PacketExceptionFormat::PacketExceptionFormat (char * *error*)

Definition at line 24 of file PacketExceptionFormat.cpp.

6.26.2.2 PacketExceptionFormat::~PacketExceptionFormat () [virtual]

Definition at line 33 of file PacketExceptionFormat.cpp.

References PacketLib::PacketException::error.

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

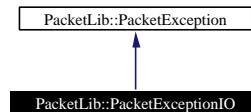
- **PacketExceptionFormat.h**
- **PacketExceptionFormat.cpp**

6.27 PacketLib::PacketExceptionIO Class Reference

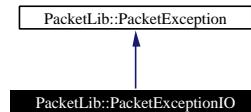
Exception class of the **PacketLib** (p. 13). The error is an IO error.

```
#include <PacketExceptionIO.h>
```

Inheritance diagram for PacketLib::PacketExceptionIO:



Collaboration diagram for PacketLib::PacketExceptionIO:



Public Methods

- **PacketExceptionIO** (char *error)
- virtual ~**PacketExceptionIO** ()

6.27.1 Detailed Description

Exception class of the **PacketLib** (p. 13). The error is an IO error.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/	PacketExceptionIO.h,v	1.3
2002/11/13 15:16:35	andrea Exp	

Id:

PacketExceptionIO.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 34 of file PacketExceptionIO.h.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 PacketExceptionIO::PacketExceptionIO (char * *error*)

Definition at line 24 of file PacketExceptionIO.cpp.

6.27.2.2 PacketExceptionIO::~PacketExceptionIO () [virtual]

Definition at line 34 of file PacketExceptionIO.cpp.

References PacketLib::PacketException::error.

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

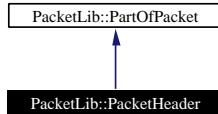
- **PacketExceptionIO.h**
- **PacketExceptionIO.cpp**

6.28 PacketLib::PacketHeader Class Reference

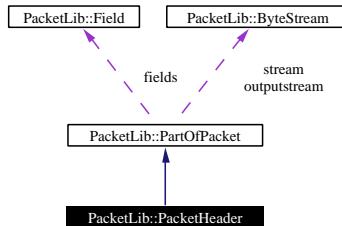
Header of packet.

```
#include <PacketHeader.h>
```

Inheritance diagram for PacketLib::PacketHeader:



Collaboration diagram for PacketLib::PacketHeader:



Public Methods

- **~PacketHeader ()**
- void **setName (char *n)**
- bool **loadHeader (char *fileName) throw (PacketException*)**
- word **getPacketLength ()**
- **PacketHeader ()**
- **Field * getFieldWithPacketDimension ()**

Private Attributes

- char * **name**
- word **numberOffFieldWithPacketDimension**

6.28.1 Detailed Description

Header of packet.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/PacketHeader.h,v 1.4 2003/05/02
08:18:00 agile Exp

Id:

PacketHeader.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 34 of file PacketHeader.h.

6.28.2 Constructor & Destructor Documentation

6.28.2.1 PacketHeader::~PacketHeader ()

Definition at line 8 of file PacketHeader.cpp.

References name.

6.28.2.2 PacketHeader::PacketHeader ()

Definition at line 67 of file PacketHeader.cpp.

References name.

6.28.3 Member Function Documentation

6.28.3.1 Field * PacketHeader::getFieldWithPacketDimension ()

Definition at line 77 of file PacketHeader.cpp.

References PacketLib::PartOfPacket::getFields, and numberFieldWithPacketDimension.

6.28.3.2 word PacketHeader::getPacketLength ()

Returns the total lenght of packet data field (data field header plus source data field)

Definition at line 55 of file PacketHeader.cpp.

References PacketLib::PartOfPacket::getFields, numberFieldWithPacketDimension, and PacketLib::word.

6.28.3.3 bool PacketHeader::loadHeader (char * *fileName*) throw (PacketException*)

Loads data header from configuration file.

Definition at line 18 of file PacketHeader.cpp.

**6.28.3.4 void PacketLib::PacketHeader::setName (char * *n*)
[inline]**

Sets the name of packet header

Definition at line 46 of file PacketHeader.h.

References name.

6.28.4 Member Data Documentation

6.28.4.1 char* PacketLib::PacketHeader::name [private]

Packet (p.100) name

Definition at line 74 of file PacketHeader.h.

Referenced by PacketHeader, setName, and ~PacketHeader.

**6.28.4.2 word PacketLib::PacketHeader::numberOffieldWith-
PacketDimension [private]**

Numbers of fields into the header that contains the dimension of packet.

Definition at line 79 of file PacketHeader.h.

Referenced by getFieldWithPacketDimension, and getPacketLength.

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

- **PacketHeader.h**
- **PacketHeader.cpp**

6.29 PacketLib::PacketIdentifier Class Reference

Identifier of packet.

```
#include <PacketIdentifier.h>
```

Public Methods

- **PacketIdentifier** (int fn, byte t, word v)
- **~PacketIdentifier** ()

Public Attributes

- **word fieldNumber**
- **byte type**
- **word value**

6.29.1 Detailed Description

Identifier of packet.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/PacketIdentifier.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

PacketIdentifier.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 33 of file PacketIdentifier.h.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 PacketIdentifier::PacketIdentifier (int *fn*, byte *t*, word *v*)

Definition at line 7 of file PacketIdentifier.cpp.

References PacketLib::byte, fieldNumber, type, value, and PacketLib::word.

6.29.2.2 **PacketIdentifier::~PacketIdentifier ()**

Definition at line 19 of file PacketIdentifier.cpp.

6.29.3 Member Data Documentation

6.29.3.1 word **PacketLib::PacketIdentifier::fieldNumber**

Definition at line 39 of file PacketIdentifier.h.

Referenced by PacketIdentifier.

6.29.3.2 byte **PacketLib::PacketIdentifier::type**

This field rappresent which list of fields read:

- 0 - List of fields of **PacketHeader** (p. 124);
- 1 - List of fields of **DataFieldHeader** (p. 33)
- 2 - List of fields of **SourceDataField** (p. 196)

Definition at line 48 of file PacketIdentifier.h.

Referenced by PacketIdentifier.

6.29.3.3 word **PacketLib::PacketIdentifier::value**

Definition at line 52 of file PacketIdentifier.h.

Referenced by PacketIdentifier.

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

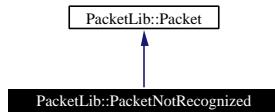
- **PacketIdentifier.h**
- **PacketIdentifier.cpp**

6.30 PacketLib::PacketNotRecognized Class Reference

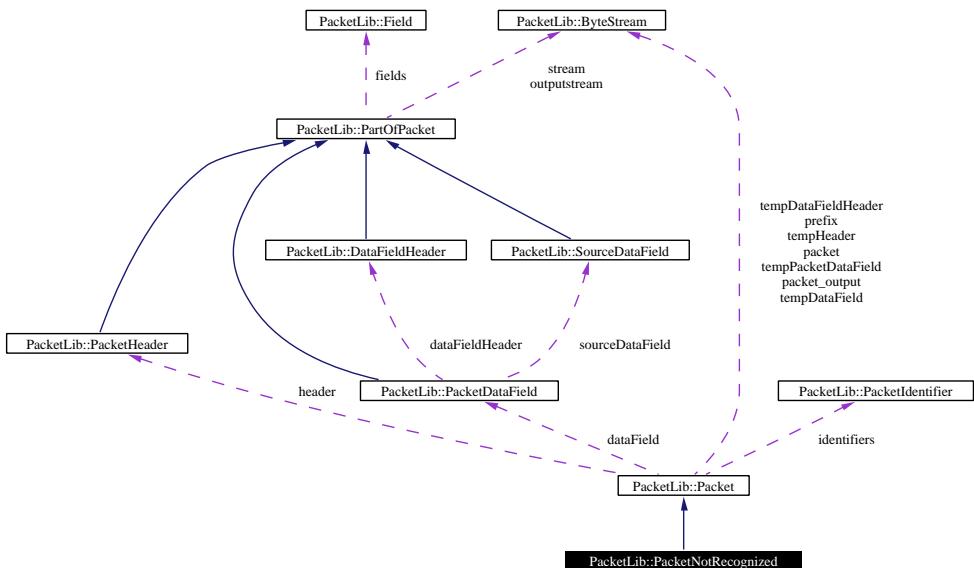
Packet (p. 100) not recognized.

```
#include <PacketNotRecognized.h>
```

Inheritance diagram for PacketLib::PacketNotRecognized:



Collaboration diagram for PacketLib::PacketNotRecognized:



Public Methods

- **PacketNotRecognized** (bool **bigEndian**)
- virtual ~**PacketNotRecognized** ()
- virtual bool **createPacketType** (char *fileName, bool **prefix**, word **dimprefix**) throw (PacketException*)
- virtual bool **setPacketValue** (**ByteStream** ***prefix**, **ByteStream** ***packetHeader**, **ByteStream** ***packetDataField**)

6.30.1 Detailed Description

Packet (p. 100) not recognized.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/PacketNotRecognized.h,v 1.4
2002/11/13 15:16:35 andrea Exp

Id:

PacketNotRecognized.h,v 1.4 2002/11/13 15:16:35 andrea Exp

Revision:

1.4

Definition at line 35 of file PacketNotRecognized.h.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 **PacketNotRecognized::PacketNotRecognized (bool *bigendian*)**

Definition at line 48 of file PacketNotRecognized.cpp.

References PacketLib::Packet::dataField.

6.30.2.2 **PacketNotRecognized::~PacketNotRecognized () [virtual]**

Definition at line 58 of file PacketNotRecognized.cpp.

6.30.3 Member Function Documentation

6.30.3.1 **bool PacketNotRecognized::createPacketType (char * *fileName*, bool *prefix*, word *dimprefix*) throw (PacketException*) [virtual]**

Reimplemented from **PacketLib::Packet** (p. 103).

Definition at line 25 of file PacketNotRecognized.cpp.

References PRINTDEBUG, and PacketLib::word.

**6.30.3.2 bool PacketNotRecognized::setPacketValue (ByteStream
* *prefix*, ByteStream * *packetHeader*, ByteStream *
 packetDataField) [virtual]**

Sets all the fields of the packet with correct value contained into the input
ByteStream (p. 21).

Precondition:

The structure of the stream must be loaded.

Parameters:

prefix This is the prefix of the packet
packetHeader This is the header of the packet
packetDataField This is the data field of the packet

Postcondition:

If return is true all the fields are setted with the correct value.

Reimplemented from **PacketLib::Packet** (p. 106).

Definition at line 68 of file PacketNotRecognized.cpp.

References **PacketLib::Packet::dataField**, **PacketLib::Packet::header**, and
PacketLib::Packet::prefix.

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

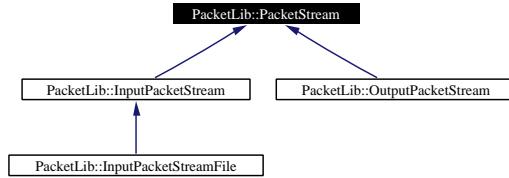
- **PacketNotRecognized.h**
- **PacketNotRecognized.cpp**

6.31 PacketLib::PacketStream Class Reference

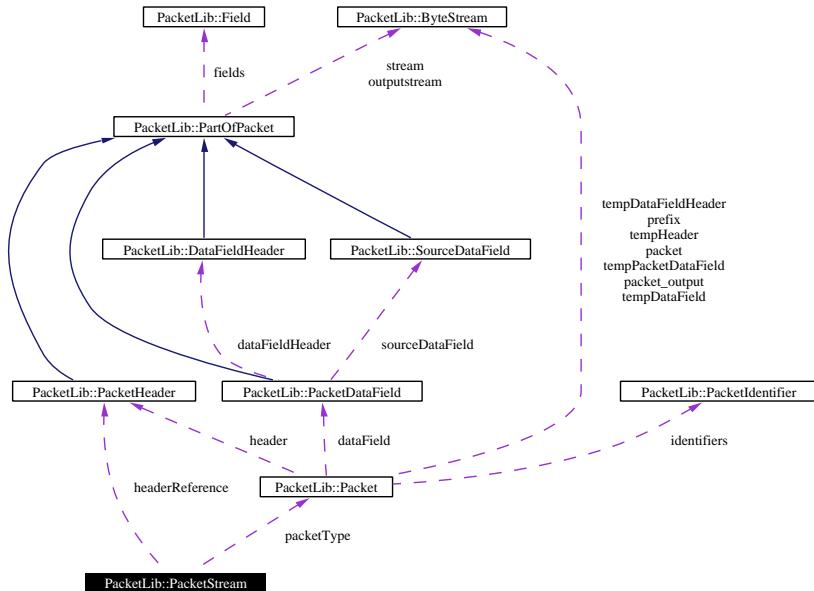
Stream of byte with packets.

```
#include <PacketStream.h>
```

Inheritance diagram for PacketLib::PacketStream:



Collaboration diagram for PacketLib::PacketStream:



Public Methods

- `PacketStream (char *fileNameConfig)`
- `PacketStream ()`
- `~PacketStream ()`
- `void setFileNameConfig (char *f)`
- `bool createStreamStructure () throw (PacketException*)`
- `Packet * getPacketType (int index)`
- `bool isBigEndian ()`

- **bool thereIsPrefix ()**
- **word getNumberOfPacketType ()**
- **word getPrefixDimension () const**
- **word getHeaderDimension () const**

Static Public Methods

- **char * packetLibVersion ()**

Public Attributes

- **PacketHeader * headerReference**
- **char * filenameConfig**

Protected Attributes

- **Packet ** packetType**
- **byte numberOfPacketType**
- **bool prefix**
- **bool bigEndian**
- **word dimPrefix**

6.31.1 Detailed Description

Stream of byte with packets.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/PacketStream.h,v 1.5 2003/05/02
08:18:00 agile Exp

Id:

PacketStream.h,v 1.5 2003/05/02 08:18:00 agile Exp

Revision:

1.5

Definition at line 35 of file PacketStream.h.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 PacketStream::PacketStream (char * *fileNameConfig*)

Definition at line 21 of file PacketStream.cpp.

References filenameConfig, headerReference, numberOfPacketType, and packetType.

6.31.2.2 PacketStream::PacketStream ()

Definition at line 37 of file PacketStream.cpp.

References filenameConfig, headerReference, numberOfPacketType, and packetType.

6.31.2.3 PacketStream::~PacketStream ()

Definition at line 53 of file PacketStream.cpp.

References headerReference, numberOfPacketType, and packetType.

6.31.3 Member Function Documentation

6.31.3.1 bool PacketStream::createStreamStructure () throw (PacketException*)

This method creates the structure of the stream. The structure of the stream is represented with the pachetReference and the collection of type of packet. Each packet has the collection of field and the collection of identifier

Definition at line 66 of file PacketStream.cpp.

6.31.3.2 word PacketStream::getHeaderDimension () const

Gets the dimension (in bytes) of the header of all the packets of the stream

Definition at line 210 of file PacketStream.cpp.

References headerReference, and PacketLib::word.

6.31.3.3 word PacketStream::getNumberOfPacketType ()

Definition at line 223 of file PacketStream.cpp.

References numberOfPacketType, and PacketLib::word.

6.31.3.4 Packet * PacketStream::getPacketType (int *index*)

Return a packet of index passed as parameters.

Definition at line 233 of file PacketStream.cpp.

References packetType.

Referenced by PacketLib::InputPacketStream::detPacketType.

6.31.3.5 word PacketStream::getPrefixDimension () const

Gets the dimension (in bytes) of the prefix of all the packets of the stream

Definition at line 197 of file PacketStream.cpp.

References dimPrefix, and PacketLib::word.

6.31.3.6 bool PacketStream::isBigEndian ()

Definition at line 243 of file PacketStream.cpp.

References bigendian.

6.31.3.7 char * PacketStream::packetLibVersion () [static]

Definition at line 11 of file PacketStream.cpp.

6.31.3.8 void PacketStream::setFileNameConfig (char * *f*)

Definition at line 253 of file PacketStream.cpp.

References filenameConfig.

6.31.3.9 bool PacketStream::thereIsPrefix ()

Definition at line 263 of file PacketStream.cpp.

References prefix.

6.31.4 Member Data Documentation**6.31.4.1 bool PacketLib::PacketStream::bigEndian [protected]**

Indicates if the machine is big endian.

Definition at line 129 of file PacketStream.h.

Referenced by isBigEndian.

6.31.4.2 word PacketLib::PacketStream::dimPrefix [protected]

Indicates the dimension of prefix.

Definition at line 135 of file PacketStream.h.

Referenced by getPrefixDimension.

6.31.4.3 char* PacketLib::PacketStream::filenameConfig

Definition at line 103 of file PacketStream.h.

Referenced by PacketStream, and setFileNameConfig.

6.31.4.4 PacketHeader* PacketLib::PacketStream::headerReference

Definition at line 99 of file PacketStream.h.

Referenced by getHeaderDimension, PacketStream, and ~PacketStream.

6.31.4.5 byte PacketLib::PacketStream::numberOfPacketType [protected]

Numbers of type of packet presents

Definition at line 117 of file PacketStream.h.

Referenced by PacketLib::InputPacketStream::detPacketType, getNumberOfPacketType, PacketStream, PacketLib::InputPacketStreamFile::~InputPacketStreamFile, and ~PacketStream.

6.31.4.6 Packet PacketLib::PacketStream::packetType [protected]**

List of packet.

Definition at line 111 of file PacketStream.h.

Referenced by getPacketType, PacketStream, PacketLib::InputPacketStreamFile::~InputPacketStreamFile, and ~PacketStream.

6.31.4.7 bool PacketLib::PacketStream::prefix [protected]

Indicates if there is a prefix.

Definition at line 123 of file PacketStream.h.

Referenced by PacketLib::InputPacketStream::detPacketType, and thereIsPrefix.

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

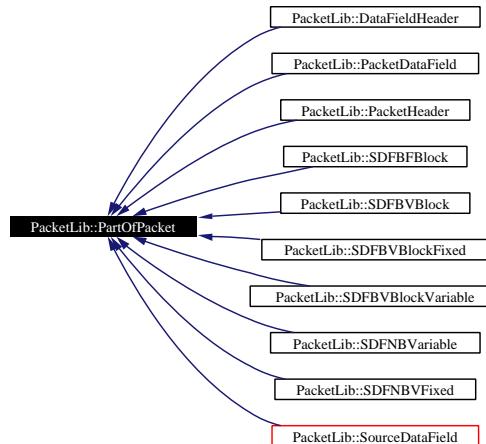
- **PacketStream.h**
- **PacketStream.cpp**

6.32 PacketLib::PartOfPacket Class Reference

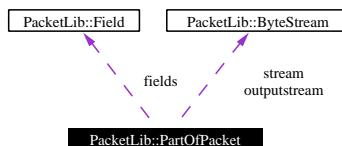
Single part of packet.

```
#include <PartOfPacket.h>
```

Inheritance diagram for PacketLib::PartOfPacket:



Collaboration diagram for PacketLib::PartOfPacket:



Public Methods

- **PartOfPacket ()**
- virtual ~**PartOfPacket ()**
- virtual string * **printStructure ()**
- virtual char ** **printValue (char *addString=(""))**
- virtual bool **loadFields (InputText &fp) throw (PacketException*)**
- virtual bool **loadFields (MemoryBuffer *mb) throw (PacketException*)**
- virtual MemoryBuffer * **loadFieldsInBuffer (InputText &fp)**
- virtual word **getDimension ()**
- virtual Field * **getFields (word index)**
- virtual word **getFieldValue (word index)**
- virtual void **setFieldValue (word index, word value)**
- virtual word **getNumberOffFields ()**

- virtual bool **setByteStream** (ByteStream *s)
- virtual ByteStream * **generateStream** (bool bigendian)
- virtual bool **setOutputStream** (ByteStream *os, word first)

Public Attributes

- **ByteStream * stream**
- **ByteStream * outputstream**

Protected Attributes

- **Field ** fields**
- **word fieldsDimension**
- **word numberOffFields**

Private Methods

- void **deleteFields** ()

6.32.1 Detailed Description

Single part of packet.

This class represent a subset of the packet. This class has been created for grouping the common behaviours of **PacketHeader** (p. 124), **DataFieldHeader** (p. 33) and **SourceDataField** (p. 196)

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/PartOfPacket.h,v 1.4 2003/05/02
08:18:00 agile Exp

Id:

PartOfPacket.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 42 of file PartOfPacket.h.

6.32.2 Constructor & Destructor Documentation

6.32.2.1 PartOfPacket::PartOfPacket ()

Constructor

Definition at line 26 of file PartOfPacket.cpp.

References fields, fieldsDimension, numberOffFields, outputStream, and stream.

6.32.2.2 PartOfPacket::~PartOfPacket () [virtual]

Virtual destructor

Definition at line 40 of file PartOfPacket.cpp.

References deleteFields, and stream.

6.32.3 Member Function Documentation

6.32.3.1 void PartOfPacket::deleteFields () [private]

Definition at line 349 of file PartOfPacket.cpp.

References fields, and numberOffFields.

Referenced by ~PartOfPacket.

6.32.3.2 ByteStream * PartOfPacket::generateStream (bool *bigEndian*) [virtual]

Generates the stream.

Reimplemented in **PacketLib::PacketDataField** (p. 114).

Definition at line 361 of file PartOfPacket.cpp.

References PacketLib::byte, fields, getDimension, numberOffFields, outputStream, and PacketLib::word.

6.32.3.3 word PartOfPacket::getDimension () [inline, virtual]

Returns the dimension (in byte) of this part of packet.

Reimplemented in **PacketLib::PacketDataField** (p. 115).

Definition at line 174 of file PartOfPacket.cpp.

References fieldsDimension, and PacketLib::word.

Referenced by generateStream, setByteStream, and setOutputStream.

6.32.3.4 Field * PartOfPacket::getFields (word *index*) [inline, virtual]

Returns a pointer of a field in the list of fields of this part of packet.

Parameters:

index Represent the index in the list.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 149).

Definition at line 184 of file PartOfPacket.cpp.

References fields, numberOfFields, and PacketLib::word.

Referenced by PacketLib::PacketHeader::getFieldWithPacketDimension, PacketLib::SDFBVBlockFixed::getNumberOfRealElement, PacketLib::PacketHeader::getPacketLength, and PacketLib::SDFBVBlockFixed::setNumberOfRealElement.

6.32.3.5 word PartOfPacket::getFieldValue (word *index*) [inline, virtual]

Returns the value of a field in the list of fields of this part of packet.

Parameters:

index Represent the index in the list.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 149).

Definition at line 197 of file PartOfPacket.cpp.

References fields, numberOfFields, and PacketLib::word.

6.32.3.6 word PartOfPacket::getNumberOfFields () [inline, virtual]

Returns the number of fields.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 150).

Definition at line 221 of file PartOfPacket.cpp.

References numberOfFields, and PacketLib::word.

**6.32.3.7 bool PartOfPacket::loadFields (MemoryBuffer * *mb*)
throw (PacketException*) [virtual]**

This method loads the field present into the **MemoryBuffer** (p. 79) (passed with the parameter).

Definition at line 96 of file PartOfPacket.cpp.

**6.32.3.8 bool PartOfPacket::loadFields (InputText & *fp*) throw
(PacketException*) [virtual]**

This method loads the field present into the **InputText** (p. 72) (passed with the parameter). The **InputText** (p. 72) must be open and the internal pointer of the file must be in the first line that describes the fields.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 151).

Definition at line 80 of file PartOfPacket.cpp.

**6.32.3.9 MemoryBuffer * PartOfPacket::loadFieldsInBuffer
(InputText & *fp*) [virtual]**

Definition at line 138 of file PartOfPacket.cpp.

6.32.3.10 string * PartOfPacket::printStructure () [virtual]

Prints the structure of this part of packet.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 151).

Definition at line 55 of file PartOfPacket.cpp.

References fields, and numberOfFields.

**6.32.3.11 char ** PartOfPacket::printValue (char * *addString* = "")
[virtual]**

Prints the value of each field of this part of packet.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 151).

Definition at line 293 of file PartOfPacket.cpp.

References fields, numberOfFields, and stream.

**6.32.3.12 bool PartOfPacket::setByteStream (ByteStream * *s*)
[virtual]**

Sets the stream of byte. This method assigns the value of stream for each field of part of packet

Reimplemented in **PacketLib::SDFBlockFixed** (p. 151).

Definition at line 231 of file PartOfPacket.cpp.

References PacketLib::byte, fields, getDimension, numberOfFields, pattern, stream, and PacketLib::word.

6.32.3.13 void PartOfPacket::setFieldValue (word *index*, word *value*) [inline, virtual]

Sets the value of a field in the list of fields of this part of packet.

Parameters:

index Represent the index in the list.

value The value must be setted.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 152).

Definition at line 210 of file PartOfPacket.cpp.

References fields, numberOfFields, and PacketLib::word.

6.32.3.14 bool PartOfPacket::setOutputStream (ByteStream * *os*, word *first*) [virtual]

Creates the outputstream **ByteStream** (p. 21) for the generation of the output stream

Reimplemented in **PacketLib::PacketDataField** (p. 115).

Definition at line 413 of file PartOfPacket.cpp.

References getDimension, outputStream, and PacketLib::word.

6.32.4 Member Data Documentation

6.32.4.1 Field PacketLib::PartOfPacket::fields [protected]**

List of field of part of packet.

Definition at line 158 of file PartOfPacket.h.

Referenced by deleteFields, generateStream, getFields, getFieldValue, PartOfPacket, printStructure, printValue, setByteStream, and setFieldValue.

6.32.4.2 word PacketLib::PartOfPacket::fieldsDimension [protected]

Dimension in bit of fields.

Definition at line 163 of file PartOfPacket.h.

Referenced by getDimension, and PartOfPacket.

6.32.4.3 word PacketLib::PartOfPacket::numberOfFields [protected]

Number of fields.

Definition at line 169 of file PartOfPacket.h.

Referenced by deleteFields, generateStream, getFields, getFieldValue, getNumberOfFields, PartOfPacket, printStructure, printValue, setByteStream, and setFieldValue.

6.32.4.4 ByteStream* PacketLib::PartOfPacket::outputstream

Represent current stream writes to output.

Definition at line 139 of file PartOfPacket.h.

Referenced by PacketLib::SDFBVBlock::generateStream, PacketLib::SDFBlockVariable::generateStream, PacketLib::SDFBlockFixed::generateStream, generateStream, PacketLib::PacketDataField::generateStream, PartOfPacket, PacketLib::SDFBlockVariable::setOutputStream, PacketLib::SDFBlockFixed::setOutputStream, setOutputStream, and PacketLib::PacketDataField::setOutputStream.

6.32.4.5 ByteStream* PacketLib::PartOfPacket::stream

Represent current stream reads from input.

Definition at line 133 of file PartOfPacket.h.

Referenced by PartOfPacket, printValue, PacketLib::SDFBlockVariable::setByteStream, PacketLib::SDFBlockFixed::setByteStream, setByteStream, and ~PartOfPacket.

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

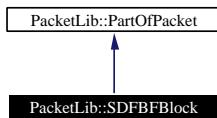
- **PartOfPacket.h**
- **PartOfPacket.cpp**

6.33 PacketLib::SDFBFBlock Class Reference

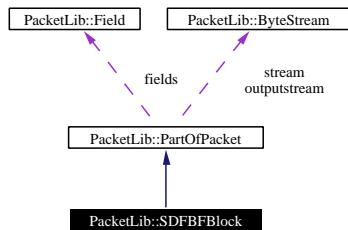
Represent a single block with the number of fields fixed.

```
#include <SDFBFBlock.h>
```

Inheritance diagram for PacketLib::SDFBFBlock:



Collaboration diagram for PacketLib::SDFBFBlock:



6.33.1 Detailed Description

Represent a single block with the number of fields fixed.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/SDFBFBlock.h,v 1.4 2002/11/13
15:16:35 andrea Exp

Id:

SDFBFBlock.h,v 1.4 2002/11/13 15:16:35 andrea Exp

Revision:

1.4

Definition at line 18 of file SDFBFBlock.h.

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

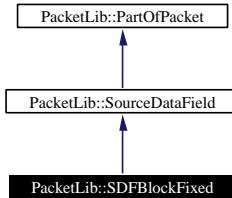
- **SDFBFBlock.h**

6.34 PacketLib::SDFBlockFixed Class Reference

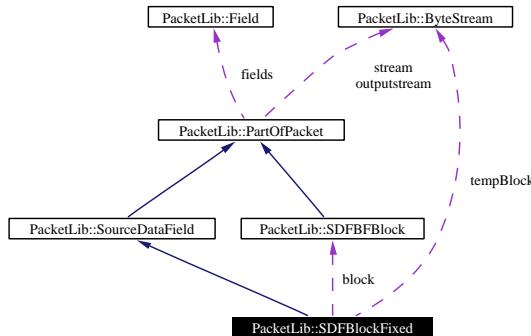
Represents a source data field structured with blocks but the number of fields for each block are fixed.

```
#include <SDFBlockFixed.h>
```

Inheritance diagram for PacketLib::SDFBlockFixed:



Collaboration diagram for PacketLib::SDFBlockFixed:



Public Methods

- virtual bool **loadFields** (InputText &fp) throw (PacketException*)
- **SDFBlockFixed ()**
- virtual ~**SDFBlockFixed ()**
- virtual Field * **getFields** (word index)
- virtual word **getNumberOfFields ()**
- virtual bool **setByteStream** (ByteStream *s)
- virtual char ** **printValue** (char *addString="")
- virtual word **getDimension ()**
- virtual word **getDimension (word block)**
- virtual word **getMaxDimension ()**
- virtual word **getMaxDimension (word nblock)**

- virtual string * **printStructure** ()
- virtual Field * **getFields** (word block, word index)
- virtual word **getFieldValue** (word index)
- virtual word **getFieldValue** (word block, word index)
- virtual void **setFieldValue** (word index, word value)
- virtual void **setFieldValue** (word block, word index, word value)
- virtual bool **setOutputStream** (ByteStream *os, word first)
- virtual ByteStream * **generateStream** (bool big endian)
- virtual word **getNumberOffFields** (word block)

Private Attributes

- SDFBFBlock * block
- ByteStream * tempBlock

6.34.1 Detailed Description

Represents a source data field structured with blocks but the number of fields for each block are fixed.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SDFBlockFixed.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

SDFBlockFixed.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 40 of file SDFBlockFixed.h.

6.34.2 Constructor & Destructor Documentation

6.34.2.1 SDFBlockFixed::SDFBlockFixed ()

Definition at line 69 of file SDFBlockFixed.cpp.

References block, PacketLib::SourceDataField::fixed, PacketLib::SourceDataField::isblock, PacketLib::SourceDataField::subFromNBlock, and tempBlock.

6.34.2.2 SDFBlockFixed::~SDFBlockFixed () [virtual]

Definition at line 83 of file SDFBlockFixed.cpp.

References block, and tempBlock.

6.34.3 Member Function Documentation

6.34.3.1 `ByteStream * SDFFBlockFixed::generateStream (bool bigEndian) [virtual]`

Generates the stream.

Reimplemented from **PacketLib::PartOfPacket** (p. 140).

Definition at line 350 of file SDFFBlockFixed.cpp.

References block, PacketLib::SourceDataField::getNumberOfRealDataBlock, PacketLib::PartOfPacket::outputstream, and PacketLib::word.

6.34.3.2 `word SDFFBlockFixed::getDimension (word block) [virtual]`

Gets the current dimension (in byte) for a single block.

Reimplemented from **PacketLib::SourceDataField** (p. 198).

Definition at line 278 of file SDFFBlockFixed.cpp.

References block, PacketLib::SourceDataField::numberOfRealDataBlock, and PacketLib::word.

6.34.3.3 `word SDFFBlockFixed::getDimension () [virtual]`

Gets the current dimension (in byte) of the source data field.

Reimplemented from **PacketLib::SourceDataField** (p. 198).

Definition at line 265 of file SDFFBlockFixed.cpp.

References block, PacketLib::SourceDataField::numberOfRealDataBlock, and PacketLib::word.

Referenced by setOutputStream.

6.34.3.4 `Field * SDFFBlockFixed::getFields (word block, word index) [virtual]`

Gets a field of the source data field.

Parameters:

block The block that contains the field

index The number of field in the source data field (into the block).

Returns:

The field.

Reimplemented from **PacketLib::SourceDataField** (p. 198).

Definition at line 117 of file SDFBlockFixed.cpp.

References block, PacketLib::SourceDataField::numberOfRealDataBlock, and PacketLib::word.

6.34.3.5 Field * SDFBlockFixed::getFields (word *index*) [virtual]

Gets a field of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 96 of file SDFBlockFixed.cpp.

References block, PacketLib::SourceDataField::getNumberOfRealDataBlock, and PacketLib::word.

Referenced by getFieldValue, and setFieldValue.

6.34.3.6 word SDFBlockFixed::getFieldValue (word *block*, word *index*) [virtual]

Gets a field value of the source data field.

Parameters:

block The block that contains the field.

index The number of field in the source data field (into the block).

Returns:

The value of field.

Reimplemented from **PacketLib::SourceDataField** (p. 199).

Definition at line 144 of file SDFBlockFixed.cpp.

References block, getFields, and PacketLib::word.

6.34.3.7 word SDFBlockFixed::getFieldValue (word *index*) [virtual]

Gets a field value of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The value of field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 130 of file SDFFixed.cpp.

References getFields, and PacketLib::word.

6.34.3.8 word SDFFixed::getMaxDimension (word *nblock*) [virtual]

Gets the total max dimension in bytes of source data field for each block

Parameters:

nblock Number of block

Reimplemented from **PacketLib::SourceDataField** (p. 200).

Definition at line 308 of file SDFFixed.cpp.

References block, PacketLib::SourceDataField::numberOfRealDataBlock, and PacketLib::word.

6.34.3.9 word SDFFixed::getMaxDimension () [virtual]

Gets the total max dimension in bytes of source data field

Implements **PacketLib::SourceDataField** (p. 200).

Definition at line 291 of file SDFFixed.cpp.

References block, PacketLib::SourceDataField::maxNumberOfBlock, and PacketLib::word.

6.34.3.10 word SDFFixed::getNumberOfFields (word *block*) [virtual]

Returns the number of fields for each block.

Reimplemented from **PacketLib::SourceDataField** (p. 201).

Definition at line 365 of file SDFFixed.cpp.

References block, PacketLib::SourceDataField::numberOfRealDataBlock, and PacketLib::word.

6.34.3.11 word SDFFixed::getNumberOfFields () [virtual]

Returns the total number of fields for the source data field.

Reimplemented from **PacketLib::SourceDataField** (p. 201).

Definition at line 182 of file SDFFixed.cpp.

References block, PacketLib::SourceDataField::getNumberOfRealDataBlock, and PacketLib::word.

Referenced by printValue.

6.34.3.12 bool SDFBlockFixed::loadFields (InputText & *fp*) throw (PacketException*) [virtual]

Loads the field form .packet file. See **PartOfPacket** (p. 138) class.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 11 of file SDFBlockFixed.cpp.

References PRINTDEBUG.

6.34.3.13 string * SDFBlockFixed::printStructure () [virtual]

Prints the structure of this part of packet.

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 321 of file SDFBlockFixed.cpp.

6.34.3.14 char ** SDFBlockFixed::printValue (char * *addString* = "") [virtual]

Prints the value of each field of this part of packet.

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 228 of file SDFBlockFixed.cpp.

References block, getNumberOfFields, PacketLib::SourceDataField::getNumberOfRealDataBlock, and PacketLib::word.

6.34.3.15 bool SDFBlockFixed::setByteStream (ByteStream * *s*) [virtual]

Sets the stream of byte. This method assigns the value of stream for each field of part of packet

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 193 of file SDFBlockFixed.cpp.

References block, PacketLib::SourceDataField::getMaxNumberOfBlock, PacketLib::SourceDataField::getNumberOfRealDataBlock, PacketLib::PartOfPacket::stream, tempBlock, and PacketLib::word.

6.34.3.16 void SDFBlockFixed::setFieldValue (word *block*, word *index*, word *value*) [virtual]

Sets a field value of the source data field.

Parameters:

block The block that contains the field.

index The number of field in the source data field (into the block).

value The value must be setted.

Reimplemented from **PacketLib::SourceDataField** (p. 203).

Definition at line 170 of file SDFFixed.cpp.

References block, getFields, and PacketLib::word.

6.34.3.17 void SDFFixed::setFieldValue (word *index*, word *value*) [virtual]

Sets a field value of the source data field.

Parameters:

index The number of field in the source data field.

value The value must be setted.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 158 of file SDFFixed.cpp.

References getFields, and PacketLib::word.

6.34.3.18 bool SDFFixed::setOutputStream (ByteStream * *os*, word *first*) [virtual]

Creates the outputstream **ByteStream** (p. 21) for the generation of the output stream

Reimplemented from **PacketLib::PartOfPacket** (p. 143).

Definition at line 331 of file SDFFixed.cpp.

References block, getDimension, PacketLib::SourceDataField::getNumberOfRealDataBlock, PacketLib::PartOfPacket::outputstream, and PacketLib::word.

6.34.4 Member Data Documentation

6.34.4.1 SDFBFixed* PacketLib::SDFFixed::block [private]

Definition at line 132 of file SDFFixed.h.

Referenced by generateStream, getDimension, getFields, getFieldValue, getMaxDimension, getNumberOfFields, printValue, SDFFixed, setByteStream, setFieldValue, setOutputStream, and ~SDFFixed.

6.34.4.2 `ByteStream* PacketLib::SDFBlockFixed::tempBlock` [private]

Definition at line 136 of file SDFBlockFixed.h.

Referenced by SDFBlockFixed, setByteStream, and ~SDFBlockFixed.

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

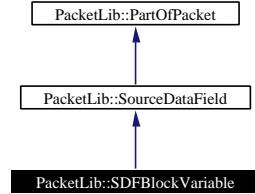
- **SDFBlockFixed.h**
- **SDFBlockFixed.cpp**

6.35 PacketLib::SDFBlockVariable Class Reference

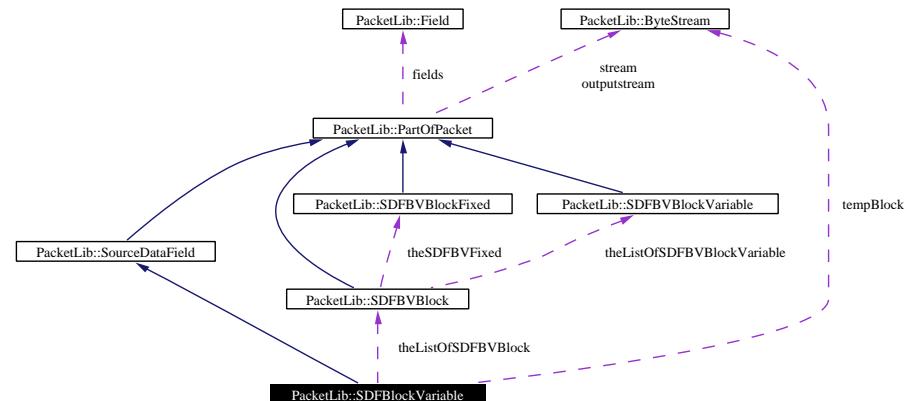
Represents a source data field structured with blocks and with the number of fields for each block variable.

```
#include <SDFBlockVariable.h>
```

Inheritance diagram for PacketLib::SDFBlockVariable:



Collaboration diagram for PacketLib::SDFBlockVariable:



Public Methods

- virtual bool **loadFields** (InputText &fp) throw (PacketException*)
- **SDFBlockVariable ()**
- virtual ~**SDFBlockVariable ()**
- virtual Field * **getFields** (word index)
- virtual Field * **getFields** (word block, word index)
- virtual word **getNumberOfFields ()**
- virtual bool **setByteStream** (ByteStream **s)
- virtual word **getDimension ()**
- virtual word **getDimension** (word nblock)
- virtual word **getMaxDimension ()**

- virtual **word getMaxDimension (word nblock)**
- virtual **char ** printValue (char *addString="")**
- virtual **string * printStructure ()**
- virtual **word getMaxNumberOfElements (word nblock)**
- virtual **word getNumberOfFields (word block)**
- virtual **word getFieldValue (word index)**
- virtual **word getFieldValue (word block, word index)**
- virtual **void setFieldValue (word index, word value)**
- virtual **void setFieldValue (word block, word index, word value)**
- virtual **bool setOutputStream (ByteStream *os, word first)**
- virtual **ByteStream * generateStream (bool bigendian)**
- virtual **word getNumberOfRealElement (word block)**
- virtual **void setNumberOfRealElement (word block, word value)**

Private Attributes

- **SDFBVBlock * theListOfSDFBVBlock**
- **ByteStream * tempBlock**

6.35.1 Detailed Description

Represents a source data field structured with blocks and with the number of fields for each block variable.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SDFBlockVariable.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

SDFBlockVariable.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 38 of file SDFBlockVariable.h.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 SDFBlockVariable::SDFBlockVariable ()

Definition at line 142 of file SDFBlockVariable.cpp.

References [PacketLib::SourceDataField::fixed](#), [PacketLib::SourceDataField::isblock](#), [tempBlock](#), and [theListOfSDFBVBlock](#).

6.35.2.2 SDFBlockVariable::~SDFBlockVariable () [virtual]

Definition at line 155 of file SDFFBlockVariable.cpp.

6.35.3 Member Function Documentation**6.35.3.1 ByteStream * SDFBlockVariable::generateStream (bool *bigr endian*) [virtual]**

Generates the stream.

Reimplemented from **PacketLib::PartOfPacket** (p. 140).

Definition at line 464 of file SDFFBlockVariable.cpp.

References **PacketLib::SourceDataField::getNumberOfRealDataBlock**, **PacketLib::PartOfPacket::outputstream**, **theListOfSDFBVBlock**, and **PacketLib::word**.

6.35.3.2 word SDFFBlockVariable::getDimension (word *nblock*) [virtual]

Gets the current dimension (in byte) for a single block.

Reimplemented from **PacketLib::SourceDataField** (p. 198).

Definition at line 247 of file SDFFBlockVariable.cpp.

References **PacketLib::SourceDataField::numberOfRealDataBlock**, **theListOfSDFBVBlock**, and **PacketLib::word**.

6.35.3.3 word SDFFBlockVariable::getDimension () [virtual]

Gets the current dimension (in byte) of the source data field.

Reimplemented from **PacketLib::SourceDataField** (p. 198).

Definition at line 220 of file SDFFBlockVariable.cpp.

References **PacketLib::SourceDataField::getNumberOfRealDataBlock**, **theListOfSDFBVBlock**, and **PacketLib::word**.

Referenced by **setOutputStream**.

6.35.3.4 Field * SDFFBlockVariable::getFields (word *block*, word *index*) [virtual]

Gets a field of the source data field.

Parameters:

block The block that contains the field

index The number of field in the source data field (into the block).

Returns:

The field.

Reimplemented from **PacketLib::SourceDataField** (p. 198).

Definition at line 324 of file SDFBlockVariable.cpp.

References `PacketLib::SourceDataField::numberOfRealDataBlock`, `theListOfSDFBVBlock`, and `PacketLib::word`.

6.35.3.5 Field * SDFBlockVariable::getFields (word *index*) [virtual]

Gets a field of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 332 of file SDFBlockVariable.cpp.

References `PacketLib::SourceDataField::getNumberOfRealDataBlock`, `theListOfSDFBVBlock`, and `PacketLib::word`.

Referenced by `getFieldValue`, and `setFieldValue`.

6.35.3.6 word SDFBlockVariable::getFieldValue (word *block*, word *index*) [virtual]

Gets a field value of the source data field.

Parameters:

block The block that contains the field.

index The number of field in the source data field (into the block).

Returns:

The value of field.

Reimplemented from **PacketLib::SourceDataField** (p. 199).

Definition at line 407 of file SDFBlockVariable.cpp.

References `getFields`, and `PacketLib::word`.

6.35.3.7 word SDFBlockVariable::getFieldValue (word *index*) [virtual]

Gets a field value of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The value of field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 393 of file SDFBLOCKVariable.cpp.

References getFields, and PacketLib::word.

6.35.3.8 word SDFBLOCKVariable::getMaxDimension (word *nblock*) [virtual]

Gets the total max dimension in bytes of source data field for each block

Parameters:

nblock Number of block

Reimplemented from **PacketLib::SourceDataField** (p. 200).

Definition at line 234 of file SDFBLOCKVariable.cpp.

References PacketLib::SourceDataField::numberOfRealDataBlock, theListOfSDFBVBlock, and PacketLib::word.

6.35.3.9 word SDFBLOCKVariable::getMaxDimension () [virtual]

Gets the total max dimension in bytes of source data field

Implements **PacketLib::SourceDataField** (p. 200).

Definition at line 260 of file SDFBLOCKVariable.cpp.

References PacketLib::SourceDataField::getMaxNumberOfBlock, theListOfSDFBVBlock, and PacketLib::word.

6.35.3.10 word SDFBLOCKVariable::getMaxNumberOfElements (word *nblock*) [virtual]

Returns:

Returns the max number of elements for each block.

Parameters:

nblock Number of block.

Reimplemented from **PacketLib::SourceDataField** (p. 201).

Definition at line 370 of file SDFBLOCKVariable.cpp.

References theListOfSDFBVBlock, and PacketLib::word.

6.35.3.11 word SDFBlockVariable::getNumberOfFields (word *block*) [virtual]

Returns the number of fields for each block

Reimplemented from **PacketLib::SourceDataField** (p. 201).

Definition at line 380 of file SDFBlockVariable.cpp.

References **PacketLib::SourceDataField::numberOfRealDataBlock**, **theListOfSDFBVBlock**, and **PacketLib::word**.

6.35.3.12 word SDFBlockVariable::getNumberOfFields () [virtual]

Returns the total number of fields for the source data field.

Reimplemented from **PacketLib::SourceDataField** (p. 201).

Definition at line 164 of file SDFBlockVariable.cpp.

References **PacketLib::SourceDataField::getNumberOfRealDataBlock**, **theListOfSDFBVBlock**, and **PacketLib::word**.

Referenced by **printValue**.

6.35.3.13 word SDFBlockVariable::getNumberOfRealElement (word *block*) [virtual]

Returns the number of real elements for each block.

Reimplemented from **PacketLib::SourceDataField** (p. 202).

Definition at line 352 of file SDFBlockVariable.cpp.

References **theListOfSDFBVBlock**, and **PacketLib::word**.

6.35.3.14 bool SDFBlockVariable::loadFields (InputText & *fp*) throw (PacketException*) [virtual]

Loads the field form .packet file. See **PartOfPacket** (p. 138) class.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 12 of file SDFBlockVariable.cpp.

References **PacketLib::word**.

6.35.3.15 string * SDFBlockVariable::printStructure () [virtual]

Prints the structure of this part of packet

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 314 of file SDFBlockVariable.cpp.

6.35.3.16 `char ** SDFBlockVariable::printValue (char * addString
= "") [virtual]`

Prints the value of each field of this part of packet

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 274 of file SDFBlockVariable.cpp.

References `getNumberOfFields`, `PacketLib::SourceDataField::getNumberOfRealDataBlock`, `theListOfSDFBVBlock`, and `PacketLib::word`.

6.35.3.17 `bool SDFBlockVariable::setByteStream (ByteStream * s)
[virtual]`

Sets the stream of byte. This method assigns the value of stream for each field of part of packet

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 184 of file SDFBlockVariable.cpp.

References `PacketLib::SourceDataField::getNumberOfRealDataBlock`, `PacketLib::PartOfPacket::stream`, `tempBlock`, `theListOfSDFBVBlock`, and `PacketLib::word`.

6.35.3.18 `void SDFBlockVariable::setFieldValue (word block, word
index, word value) [virtual]`

Sets a field value of the source data field.

Parameters:

block The block that contains the field.

index The number of field in the source data field (into the block).

value The value must be setted.

Reimplemented from **PacketLib::SourceDataField** (p. 203).

Definition at line 433 of file SDFBlockVariable.cpp.

References `getFields`, and `PacketLib::word`.

6.35.3.19 `void SDFBlockVariable::setFieldValue (word index, word
value) [virtual]`

Sets a field value of the source data field.

Parameters:

index The number of field in the source data field.

value The value must be setted.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 421 of file SDFBlockVariable.cpp.

References getFields, and PacketLib::word.

6.35.3.20 void SDFBlockVariable::setNumberOfRealElement (word *block*, word *value*) [virtual]

Sets the number of real elements for each block.

Parameters:

block Number of block. \value *value* The value must be setted.

Reimplemented from **PacketLib::SourceDataField** (p. 204).

Definition at line 359 of file SDFBlockVariable.cpp.

References PacketLib::SourceDataField::maxNumberOfBlock, PacketLib::SourceDataField::numberOfRealDataBlock, PacketLib::SourceDataField::reset_output_stream, theListOfSDFBVBlock, and PacketLib::word.

6.35.3.21 bool SDFBlockVariable::setOutputStream (ByteStream * *os*, word *first*) [virtual]

Creates the outputstream **ByteStream** (p. 21) for the generation of the output stream

Reimplemented from **PacketLib::PartOfPacket** (p. 143).

Definition at line 445 of file SDFBlockVariable.cpp.

References getDimension, PacketLib::SourceDataField::getNumberOfRealDataBlock, PacketLib::PartOfPacket::outputstream, theListOfSDFBVBlock, and PacketLib::word.

6.35.4 Member Data Documentation

6.35.4.1 ByteStream* PacketLib::SDFBlockVariable::tempBlock [private]

Definition at line 148 of file SDFBlockVariable.h.

Referenced by SDFBlockVariable, and setByteStream.

6.35.4.2 SDFBVBlock* PacketLib::SDFBlockVariable::theListOfSDFBVBlock [private]

Definition at line 144 of file SDFBlockVariable.h.

Referenced by generateStream, getDimension, getFields, getMaxDimension, getMaxNumberOfElements, getNumberOfFields, getNumberOfRealElement, printValue, SDFBlockVariable, setByteStream, setNumberOfRealElement, and setOutputStream.

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

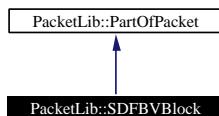
- **SDFBlockVariable.h**
- **SDFBlockVariable.cpp**

6.36 PacketLib::SDFBVBlock Class Reference

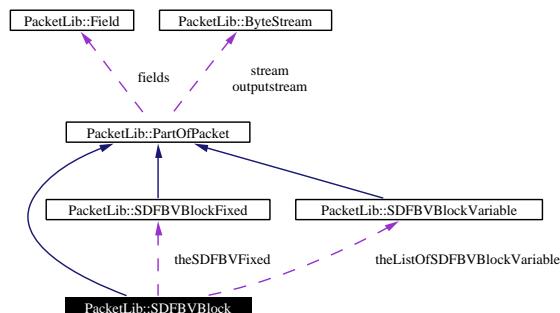
Class which represents a single block with a fixed part and with a variable part.

```
#include <SDFBVBlock.h>
```

Inheritance diagram for PacketLib::SDFBVBlock:



Collaboration diagram for PacketLib::SDFBVBlock:



Public Methods

- **word** **getNumberOfFields** ()
 - **Field *** **getFields** (**word** index)
 - **word** **getFieldValue** (**word** index)
 - **word** **getDimension** ()
 - **word** **getMaxDimension** ()
 - **bool** **setOutputStream** (**ByteStream** *os, **word** first)
 - **ByteStream *** **generateStream** (bool bigEndian)

Public Attributes

- **SDFBVBlockFixed** theSDFBVFixed
 - **SDFBVBlockVariable *** theListOfSDFBVBlockVariable

6.36.1 Detailed Description

Class which represents a single block with a fixed part and with a variable part.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SDFBVBlock.h,v 1.4 2003/05/02
08:18:00 agile Exp

Id:

SDFBVBlock.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 34 of file SDFBVBlock.h.

6.36.2 Member Function Documentation

6.36.2.1 `ByteStream * SDFBVBlock::generateStream (bool bigEndian) [virtual]`

Generates the stream.

Reimplemented from **PacketLib::PartOfPacket** (p. 140).

Definition at line 99 of file SDFBVBlock.cpp.

References `PacketLib::PartOfPacket::outputstream`, `theListOfSDFBVBlockVariable`, `theSDFBVFixed`, and `PacketLib::word`.

6.36.2.2 `word SDFBVBlock::getDimension () [virtual]`

Returns the dimension (in byte) of this part of packet.

Reimplemented from **PacketLib::PartOfPacket** (p. 140).

Definition at line 50 of file SDFBVBlock.cpp.

References `theListOfSDFBVBlockVariable`, `theSDFBVFixed`, and `PacketLib::word`.

6.36.2.3 `Field * SDFBVBlock::getFields (word index) [virtual]`

Returns a pointer of a field in the list of fields of this part of packet.

Parameters:

index Represent the index in the list.

Reimplemented from **PacketLib::PartOfPacket** (p. 141).

Definition at line 7 of file SDFBVBlock.cpp.

References `theListOfSDFBVBlockVariable`, `theSDFBVFixed`, and `PacketLib::word`.

Referenced by `getFieldValue`.

6.36.2.4 word SDFBVBlock::getFieldValue (word *index*) [virtual]

Returns the value of a field in the list of fields of this part of packet.

Parameters:

index Represent the index in the list.

Reimplemented from **PacketLib::PartOfPacket** (p. 141).

Definition at line 26 of file SDFBVBlock.cpp.

References `getFields`, and `PacketLib::word`.

6.36.2.5 word SDFBVBlock::getMaxDimension ()

Total max dimension in bytes of block

Definition at line 65 of file SDFBVBlock.cpp.

References `theListOfSDFBVBlockVariable`, `theSDFBVFixed`, and `PacketLib::word`.

6.36.2.6 word SDFBVBlock::getNumberOfFields () [virtual]

Returns the number of fields.

Reimplemented from **PacketLib::PartOfPacket** (p. 141).

Definition at line 36 of file SDFBVBlock.cpp.

References `theListOfSDFBVBlockVariable`, `theSDFBVFixed`, and `PacketLib::word`.

6.36.2.7 bool SDFBVBlock::setOutputStream (ByteStream * *os*, word *first*) [virtual]

Creates the outputstream **ByteStream** (p. 21) for the generation of the output stream

Reimplemented from **PacketLib::PartOfPacket** (p. 143).

Definition at line 81 of file SDFBVBlock.cpp.

References `theListOfSDFBVBlockVariable`, `theSDFBVFixed`, and `PacketLib::word`.

6.36.3 Member Data Documentation

6.36.3.1 SDFBVBlockVariable* PacketLib::SDFBVBlock::theListOfSDFBVBlockVariable

Definition at line 45 of file SDFBVBlock.h.

Referenced by generateStream, getDimension, getFields, getMaxDimension, getNumberOfFields, and setOutputStream.

6.36.3.2 SDFBVBlockFixed PacketLib::SDFBVBlock::theSDFBVFixed

D

Definition at line 41 of file SDFBVBlock.h.

Referenced by generateStream, getDimension, getFields, getMaxDimension, getNumberOfFields, and setOutputStream.

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

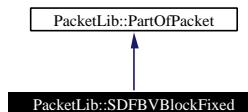
- **SDFBVBlock.h**
- **SDFBVBlock.cpp**

6.37 PacketLib::SDFBVBlockFixed Class Reference

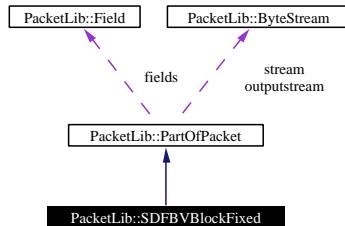
Class which represent the part of block with the number of field fixed.

```
#include <SDFBVBlockFixed.h>
```

Inheritance diagram for PacketLib::SDFBVBlockFixed:



Collaboration diagram for PacketLib::SDFBVBlockFixed:



Public Methods

- **word getNumberOfRealElement ()**
- **void setNumberOfRealElement (word value)**
- **void setIndexOfNElement (word index)**
- **void setAddToNElement (unsigned num)**
- **void setMaxNumberOfElement (word num)**
- **word getMaxNumberOfElement () const**

Private Attributes

- **word indexOfNElement**
- **unsigned addToNElement**
- **word maxNumberOfElement**

6.37.1 Detailed Description

Class which represent the part of block with the number of field fixed.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SDFBVBlockFixed.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

SDFBVBlockFixed.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 33 of file SDFBVBlockFixed.h.

6.37.2 Member Function Documentation

6.37.2.1 word SDFBVBlockFixed::getMaxNumberOfElement () const

Definition at line 60 of file SDFBVBlockFixed.cpp.

References maxNumberOfElement, and PacketLib::word.

6.37.2.2 word SDFBVBlockFixed::getNumberOfRealElement ()

Definition at line 7 of file SDFBVBlockFixed.cpp.

References addToNElement, PacketLib::PartOfPacket::getFields, indexOfNElement, and PacketLib::word.

6.37.2.3 void SDFBVBlockFixed::setAddToNElement (unsigned *num*)

Sets the number to sum for obtaining the numbers of real elements.

Definition at line 40 of file SDFBVBlockFixed.cpp.

References addToNElement.

6.37.2.4 void SDFBVBlockFixed::setIndexOfNElement (word *index*)

Definition at line 30 of file SDFBVBlockFixed.cpp.

References indexOfNElement, and PacketLib::word.

6.37.2.5 void SDFBVBlockFixed::setMaxNumberOfElement (word num)

Definition at line 50 of file SDFBVBlockFixed.cpp.

References maxNumberOfElement, and PacketLib::word.

6.37.2.6 void SDFBVBlockFixed::setNumberOfRealElement (word value)

Definition at line 18 of file SDFBVBlockFixed.cpp.

References addToNElement, PacketLib::PartOfPacket::getFields, indexOfNElement, and PacketLib::word.

6.37.3 Member Data Documentation

6.37.3.1 unsigned PacketLib::SDFBVBlockFixed::addToNElement [private]

Number to sum for obtaining the numbers of real elements.

Definition at line 48 of file SDFBVBlockFixed.h.

Referenced by getNumberOfRealElement, setAddToNElement, and setNumberOfRealElement.

6.37.3.2 word PacketLib::SDFBVBlockFixed::indexOfNElement [private]

Represents the index of field which have the number of element in a source data field variable.

Definition at line 42 of file SDFBVBlockFixed.h.

Referenced by getNumberOfRealElement, setIndexOfNElement, and setNumberOfRealElement.

6.37.3.3 word PacketLib::SDFBVBlockFixed::maxNumberOfElement [private]

Represents the max number of elements in a source data field variable.

Definition at line 54 of file SDFBVBlockFixed.h.

Referenced by getMaxNumberOfElement, and setMaxNumberOfElement.

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

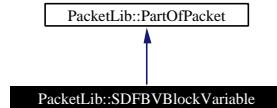
- **SDFBVBlockFixed.h**
- **SDFBVBlockFixed.cpp**

6.38 PacketLib::SDFBVBlockVariable Class Reference

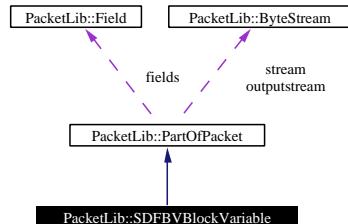
Class which represents the part of block repeated because the number of fields are variable.

```
#include <SDFBVBlockVariable.h>
```

Inheritance diagram for PacketLib::SDFBVBlockVariable:



Collaboration diagram for PacketLib::SDFBVBlockVariable:



6.38.1 Detailed Description

Class which represents the part of block repeated because the number of fields are variable.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/SDFBVBlockVariable.h,v	1.3
2002/11/13 15:16:35 andrea Exp	

Id:

SDFBVBlockVariable.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 35 of file SDFBVBlockVariable.h.

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

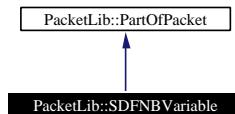
- **SDFBVBlockVariable.h**

6.39 PacketLib::SDFNBVariable Class Reference

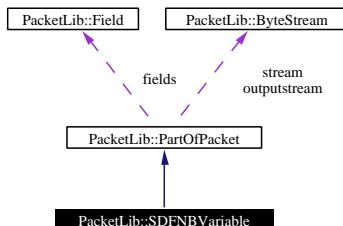
Class which represents the variable part (the fields (or elements) which are repeated) of source data field.

```
#include <SDFNBVariable.h>
```

Inheritance diagram for PacketLib::SDFNBVariable:



Collaboration diagram for PacketLib::SDFNBVariable:



6.39.1 Detailed Description

Class which represents the variable part (the fields (or elements) which are repeated) of source data field.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/SDFNBVariable.h,v	1.3
2002/11/13 15:16:35 andrea Exp	

Id:

SDFNBVariable.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 35 of file SDFNBVariable.h.

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

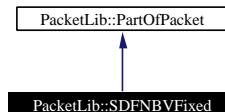
- **SDFNBVariable.h**

6.40 PacketLib::SDFNBVFixed Class Reference

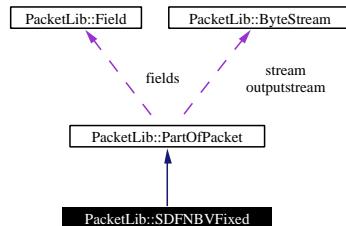
Represents the fixed part of source data field.

```
#include <SDFNBVFixed.h>
```

Inheritance diagram for PacketLib::SDFNBVFixed:



Collaboration diagram for PacketLib::SDFNBVFixed:



6.40.1 Detailed Description

Represents the fixed part of source data field.

Date:

2002/11/13 15:16:35

Header:

/home/cvs/PacketLib/packetlib/packet/SDFNBVFixed.h,v	1.3
2002/11/13 15:16:35 andrea Exp	

Id:

SDFNBVFixed.h,v 1.3 2002/11/13 15:16:35 andrea Exp

Revision:

1.3

Definition at line 34 of file SDFNBVFixed.h.

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

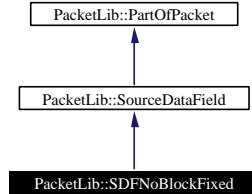
- **SDFNBVFixed.h**

6.41 PacketLib::SDFNoBlockFixed Class Reference

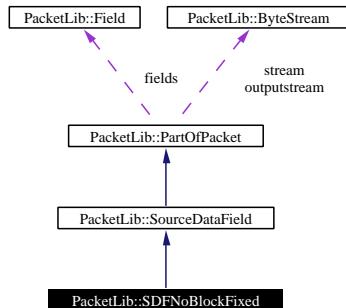
Represents a source data field of packet where the structure is noblock and with a fixed dimension.

```
#include <SDFNoBlockFixed.h>
```

Inheritance diagram for PacketLib::SDFNoBlockFixed:



Collaboration diagram for PacketLib::SDFNoBlockFixed:



Public Methods

- **SDFNoBlockFixed ()**
- virtual ~**SDFNoBlockFixed ()**
- virtual bool **loadFields** (**InputText** &fp) throw (**PacketException***)
- virtual **Field** * **getFields** (**word** index)
- virtual **word** **getFieldValue** (**word** index)
- virtual void **setFieldValue** (**word** index, **word** value)
- virtual **word** **getMaxDimension** ()

6.41.1 Detailed Description

Represents a source data field of packet where the structure is noblock and with a fixed dimension.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SDFNoBlockFixed.h,v 1.4
2003/05/02 08:18:00 agile Exp

Id:

SDFNoBlockFixed.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 35 of file SDFNoBlockFixed.h.

6.41.2 Constructor & Destructor Documentation

6.41.2.1 SDFNoBlockFixed::SDFNoBlockFixed ()

Definition at line 8 of file SDFNoBlockFixed.cpp.

References PacketLib::SourceDataField::fixed, and PacketLib::SourceDataField::isblock.

6.41.2.2 SDFNoBlockFixed::~SDFNoBlockFixed () [virtual]

Definition at line 19 of file SDFNoBlockFixed.cpp.

6.41.3 Member Function Documentation

6.41.3.1 Field * SDFNoBlockFixed::getFields (word *index*) [virtual]

Gets a field of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 45 of file SDFNoBlockFixed.cpp.

References PacketLib::word.

**6.41.3.2 word SDFNoBlockFixed::getFieldValue (word *index*)
[virtual]**

Gets a field value of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The value of field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 52 of file SDFNoBlockFixed.cpp.

References PacketLib::word.

6.41.3.3 word SDFNoBlockFixed::getMaxDimension () [virtual]

Gets the total max dimension in bytes of source data field

Implements **PacketLib::SourceDataField** (p. 200).

Definition at line 69 of file SDFNoBlockFixed.cpp.

References PacketLib::SourceDataField::getDimension, and PacketLib::word.

**6.41.3.4 bool SDFNoBlockFixed::loadFields (InputText & *fp*)
throw (PacketException*) [virtual]**

Loads the field form .packet file. See **PartOfPacket** (p. 138) class.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 28 of file SDFNoBlockFixed.cpp.

**6.41.3.5 void SDFNoBlockFixed::setFieldValue (word *index*, word
value) [virtual]**

Sets a field value of the source data field.

Parameters:

index The number of field in the source data field.

value The value must be setted.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 62 of file SDFNoBlockFixed.cpp.

References PacketLib::word.

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

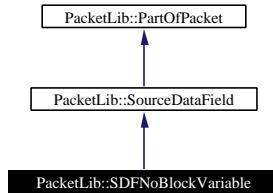
- **SDFNoBlockFixed.h**
- **SDFNoBlockFixed.cpp**

6.42 PacketLib::SDFNoBlockVariable Class Reference

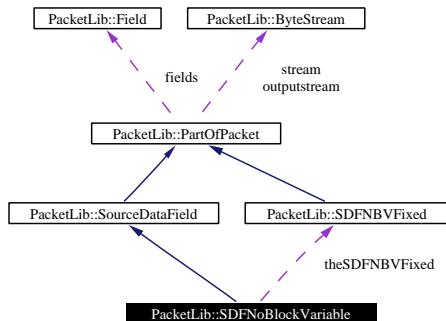
Represents a source data field with no block but with the number of fields variable.

```
#include <SDFNoBlockVariable.h>
```

Inheritance diagram for PacketLib::SDFNoBlockVariable:



Collaboration diagram for PacketLib::SDFNoBlockVariable:



Public Methods

- virtual bool **loadFields** (InputText &fp) throw (PacketException*)
- **SDFNoBlockVariable** ()
- virtual ~**SDFNoBlockVariable** ()
- virtual **Field** * **getFields** (word index)
- virtual word **getNumberOffFields** ()
- virtual bool **setByteStream** (ByteStream *s)
- virtual word **getDimension** ()
- virtual word **getMaxDimension** ()
- virtual char ** **printValue** (char *addString="")
- virtual string * **printStructure** ()
- virtual word **getFieldValue** (word index)
- virtual void **setFieldValue** (word index, word value)

Private Attributes

- **SDFNBVFixed** theSDFNBVFixed
- **list< SDFNBVariable >** theListOfSDFNBVariable

6.42.1 Detailed Description

Represents a source data field with no block but with the number of fields variable.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SDFNoBlockVariable.h,v	1.4
2003/05/02 08:18:00 agile Exp	

Id:

SDFNoBlockVariable.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 41 of file SDFNoBlockVariable.h.

6.42.2 Constructor & Destructor Documentation

6.42.2.1 SDFNoBlockVariable::SDFNoBlockVariable ()

Definition at line 20 of file SDFNoBlockVariable.cpp.

References PacketLib::SourceDataField::fixed, and PacketLib::SourceDataField::isblock.

6.42.2.2 SDFNoBlockVariable::~SDFNoBlockVariable () [virtual]

Definition at line 31 of file SDFNoBlockVariable.cpp.

6.42.3 Member Function Documentation

6.42.3.1 word SDFNoBlockVariable::getDimension () [virtual]

Gets the current dimension (in byte) of the source data field.

Reimplemented from **PacketLib::SourceDataField** (p. 198).

Definition at line 81 of file SDFNoBlockVariable.cpp.

References PacketLib::word.

**6.42.3.2 Field * SDFNoBlockVariable::getFields (word *index*)
[virtual]**

Gets a field of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 40 of file SDFNoBlockVariable.cpp.

References PacketLib::word.

Referenced by getFieldValue.

**6.42.3.3 word SDFNoBlockVariable::getFieldValue (word *index*)
[virtual]**

Gets a field value of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The value of field.

Implements **PacketLib::SourceDataField** (p. 199).

Definition at line 108 of file SDFNoBlockVariable.cpp.

References getFields, and PacketLib::word.

**6.42.3.4 word SDFNoBlockVariable::getMaxDimension ()
[virtual]**

Total max dimension in bytes of source data field

Implements **PacketLib::SourceDataField** (p. 200).

Definition at line 91 of file SDFNoBlockVariable.cpp.

References PacketLib::word.

**6.42.3.5 word SDFNoBlockVariable::getNumberOfFields ()
[virtual]**

Returns the total number of fields for the source data field.

Reimplemented from **PacketLib::SourceDataField** (p. 201).

Definition at line 50 of file SDFNoBlockVariable.cpp.

References PacketLib::word.

6.42.3.6 bool SDFNoBlockVariable::loadFields (InputText & *fp*) throw (PacketException*) [virtual]

Loads the field form .packet file. See **PartOfPacket** (p. 138) class.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 10 of file SDFNoBlockVariable.cpp.

6.42.3.7 string * SDFNoBlockVariable::printStructure () [virtual]

Print the structure of this part of packet

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 98 of file SDFNoBlockVariable.cpp.

6.42.3.8 char ** SDFNoBlockVariable::printValue (char * *addString* = "") [virtual]

Print the value of each fields of this part of packet

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 70 of file SDFNoBlockVariable.cpp.

6.42.3.9 bool SDFNoBlockVariable::setByteStream (ByteStream * *s*) [virtual]

Sets the stream of byte. This method assigns the value of stream for each field of part of packet

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Definition at line 60 of file SDFNoBlockVariable.cpp.

6.42.3.10 void SDFNoBlockVariable::setFieldValue (word *index*, word *value*) [virtual]

Sets a field value of the source data field.

Parameters:

index The number of field in the source data field.

value The value must be setted.

Implements **PacketLib::SourceDataField** (p. 203).

Definition at line 118 of file SDFNoBlockVariable.cpp.

References PacketLib::word.

6.42.4 Member Data Documentation

6.42.4.1 `list<SDFNBVariable> PacketLib::SDFNoBlockVariable::theListOfSDFNBVariable [private]`

Definition at line 107 of file SDFNoBlockVariable.h.

6.42.4.2 `SDFNBVFixed PacketLib::SDFNoBlockVariable::theSDFNBVFixed [private]`

Definition at line 103 of file SDFNoBlockVariable.h.

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

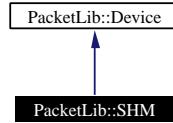
- `SDFNoBlockVariable.h`
- `SDFNoBlockVariable.cpp`

6.43 PacketLib::SHM Class Reference

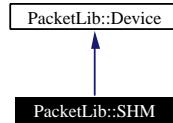
This class represents generic **SHM** (p. 184).

```
#include <SHM.h>
```

Inheritance diagram for PacketLib::SHM:



Collaboration diagram for PacketLib::SHM:



Public Methods

- **SHM (bool bigendian)**
- virtual **~SHM ()**
- virtual bool **create (long shmkey, dword num_slot, dword slot_dim)**
throw (PacketExceptionIO*)
- virtual bool **open (long shmkey=0, dword num_slot=0, dword slot_dim=0)** throw (PacketExceptionIO*)
- virtual bool **close ()** throw (PacketExceptionIO*)
- virtual bool **destroy ()** throw (PacketExceptionIO*)
- virtual bool **isCreated ()** const
- virtual void **writeSlot (dword n_slot, void *element)** throw (PacketExceptionIO*)
- virtual void * **readSlot (dword n_slot)** throw (PacketExceptionIO*)

Protected Attributes

- int **shmid**
- long **shmkey**
- **dword num_slot**
- **dword slot_dim**
- bool **created**
- **byte * shm**

6.43.1 Detailed Description

This class represents generic **SHM** (p. 184).

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SHM.h,v 1.3 2003/05/02 08:18:00
agile Exp

Id:

SHM.h,v 1.3 2003/05/02 08:18:00 agile Exp

Revision:

1.3

Definition at line 38 of file SHM.h.

6.43.2 Constructor & Destructor Documentation

6.43.2.1 SHM::SHM (*bool bigendian*)

Definition at line 25 of file SHM.cpp.

References created.

6.43.2.2 SHM::~SHM () [virtual]

Definition at line 35 of file SHM.cpp.

References shmid.

6.43.3 Member Function Documentation

6.43.3.1 bool SHM::close () throw (PacketExceptionIO*) [virtual]

Definition at line 124 of file SHM.cpp.

6.43.3.2 bool SHM::create (long *shmkey*, dword *num_slot*, dword *slot_dim*) throw (PacketExceptionIO*) [virtual]

Definition at line 46 of file SHM.cpp.

References PacketLib::dword, and PERM.

6.43.3.3 bool SHM::destroy () throw (PacketExceptionIO*) [virtual]

Definition at line 146 of file SHM.cpp.

6.43.3.4 bool SHM::isCreated () const [virtual]

Definition at line 168 of file SHM.cpp.

References created.

6.43.3.5 bool SHM::open (long *shmkey* = 0, dword *num_slot* = 0, dword *slot_dim* = 0) throw (PacketExceptionIO*) [virtual]

Definition at line 83 of file SHM.cpp.

References PacketLib::byte, and PacketLib::dword.

6.43.3.6 void * SHM::readSlot (dword *n_slot*) throw (PacketExceptionIO*) [virtual]

Definition at line 201 of file SHM.cpp.

References PacketLib::dword.

6.43.3.7 void SHM::writeSlot (dword *n_slot*, void * *element*) throw (PacketExceptionIO*) [virtual]

Definition at line 178 of file SHM.cpp.

References PacketLib::byte, and PacketLib::dword.

6.43.4 Member Data Documentation

6.43.4.1 bool PacketLib::SHM::created [protected]

Definition at line 97 of file SHM.h.

Referenced by isCreated, and SHM.

6.43.4.2 dword PacketLib::SHM::num_slot [protected]

Definition at line 89 of file SHM.h.

6.43.4.3 byte* PacketLib::SHM::shm [protected]

Definition at line 101 of file SHM.h.

6.43.4.4 int PacketLib::SHM::shm_id [protected]

Definition at line 81 of file SHM.h.

Referenced by ~SHM.

6.43.4.5 long PacketLib::SHM::shmkey [protected]

Definition at line 85 of file SHM.h.

6.43.4.6 dword PacketLib::SHM::slot_dim [protected]

Definition at line 93 of file SHM.h.

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

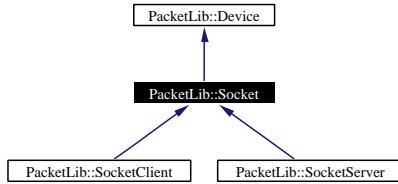
- **SHM.h**
- **SHM.cpp**

6.44 PacketLib::Socket Class Reference

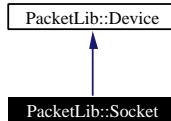
This class represents generic socket.

```
#include <Socket.h>
```

Inheritance diagram for PacketLib::Socket:



Collaboration diagram for PacketLib::Socket:



Public Methods

- **Socket (bool bigendian)**
- virtual **~Socket ()**
- virtual bool **is_valid () const**
- virtual bool **send (ByteStream *b) const throw (PacketExceptionIO*)**
- virtual **ByteStream * recv (word dim, int &status) throw (PacketExceptionIO*)**
- virtual bool **connect (const std::string host, const int port) throw (PacketExceptionIO*)**
- virtual bool **create () throw (PacketExceptionIO*)**
- virtual bool **close () throw (PacketExceptionIO*)**

Protected Attributes

- int **m_sock**
- sockaddr_in **m_addr**

6.44.1 Detailed Description

This class represents generic socket.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/Socket.h,v 1.4 2003/05/02
08:18:00 agile Exp

Id:

Socket.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 47 of file Socket.h.

6.44.2 Constructor & Destructor Documentation

6.44.2.1 **Socket::Socket (bool *bigendian*)**

Definition at line 30 of file Socket.cpp.

References m_addr.

6.44.2.2 **Socket::~Socket () [virtual]**

Definition at line 42 of file Socket.cpp.

References is_valid.

6.44.3 Member Function Documentation

6.44.3.1 **bool Socket::close () throw (PacketExceptionIO*) [virtual]**

Definition at line 76 of file Socket.cpp.

6.44.3.2 **bool Socket::connect (const std::string *host*, const int *port*) throw (PacketExceptionIO*) [virtual]**

Definition at line 158 of file Socket.cpp.

6.44.3.3 **bool Socket::create () throw (PacketExceptionIO*) [virtual]**

Definition at line 53 of file Socket.cpp.

Referenced by PacketLib::SocketServer::SocketServer.

6.44.3.4 virtual bool PacketLib::Socket::is_valid () const [inline, virtual]

Definition at line 61 of file Socket.h.

References m_sock.

Referenced by ~Socket.

6.44.3.5 ByteStream * Socket::recv (word *dim*, int & *status*) throw (PacketExceptionIO*) [virtual]

Definition at line 110 of file Socket.cpp.

References PacketLib::byte, and PacketLib::word.

6.44.3.6 bool Socket::send (ByteStream * *b*) const throw (PacketExceptionIO*) [virtual]

Data Transimission

Definition at line 90 of file Socket.cpp.

References PacketLib::byte.

6.44.4 Member Data Documentation

6.44.4.1 sockaddr_in PacketLib::Socket::m_addr [protected]

Definition at line 93 of file Socket.h.

Referenced by Socket.

6.44.4.2 int PacketLib::Socket::m_sock [protected]

Definition at line 89 of file Socket.h.

Referenced by is_valid, and PacketLib::SocketServer::set_non_blocking.

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

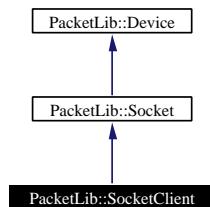
- [Socket.h](#)
- [Socket.cpp](#)

6.45 PacketLib::SocketClient Class Reference

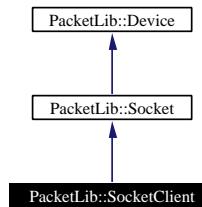
This class represents generic client socket.

```
#include <SocketClient.h>
```

Inheritance diagram for PacketLib::SocketClient:



Collaboration diagram for PacketLib::SocketClient:



Public Methods

- **SocketClient** (bool **bigEndian**, std::string host, int port)

6.45.1 Detailed Description

This class represents generic client socket.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SocketClient.h,v 1.3 2003/05/02
08:18:00 agile Exp

Id:

SocketClient.h,v 1.3 2003/05/02 08:18:00 agile Exp

Revision:

1.3

Definition at line 37 of file SocketClient.h.

6.45.2 Constructor & Destructor Documentation

6.45.2.1 `SocketClient::SocketClient (bool bigendian, std::string host, int port)`

Definition at line 27 of file SocketClient.cpp.

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

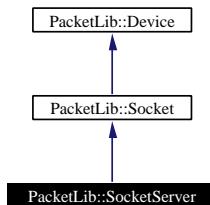
- `SocketClient.h`
- `SocketClient.cpp`

6.46 PacketLib::SocketServer Class Reference

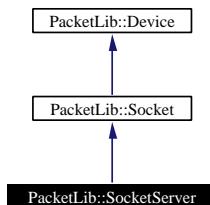
This class represents generic client socket.

```
#include <SocketServer.h>
```

Inheritance diagram for PacketLib::SocketServer:



Collaboration diagram for PacketLib::SocketServer:



Public Methods

- **SocketServer (bool bigEndian)**
- **SocketServer (bool bigEndian, int port)**
- **virtual ~SocketServer ()**
- **bool bind (const int port) throw (PacketExceptionIO*)**
- **bool listen () const throw (PacketExceptionIO*)**
- **void set_non_blocking (const bool)**
- **bool accept (SocketServer &) throw (PacketExceptionIO*)**

6.46.1 Detailed Description

This class represents generic client socket.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SocketServer.h,v 1.3 2003/05/02
08:18:00 agile Exp

Id:

SocketServer.h,v 1.3 2003/05/02 08:18:00 agile Exp

Revision:

1.3

Definition at line 41 of file SocketServer.h.

6.46.2 Constructor & Destructor Documentation

6.46.2.1 `SocketServer::SocketServer (bool bigendian)`

Definition at line 44 of file SocketServer.cpp.

6.46.2.2 `SocketServer::SocketServer (bool bigendian, int port)`

Definition at line 26 of file SocketServer.cpp.

References bind, and PacketLib::Socket::create.

6.46.2.3 `SocketServer::~SocketServer () [virtual]`

Definition at line 51 of file SocketServer.cpp.

6.46.3 Member Function Documentation

6.46.3.1 `bool SocketServer::accept (SocketServer & new_socket) throw (PacketExceptionIO*)`

Definition at line 112 of file SocketServer.cpp.

6.46.3.2 `bool SocketServer::bind (const int port) throw (PacketExceptionIO*)`

Definition at line 61 of file SocketServer.cpp.

Referenced by SocketServer.

6.46.3.3 `bool SocketServer::listen () const throw (PacketExceptionIO*)`

Definition at line 90 of file SocketServer.cpp.

References MAXCONNECTIONS.

6.46.3.4 void SocketServer::set_non_blocking (const *bool*)

Definition at line 128 of file SocketServer.cpp.

References PacketLib::Socket::m_sock.

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

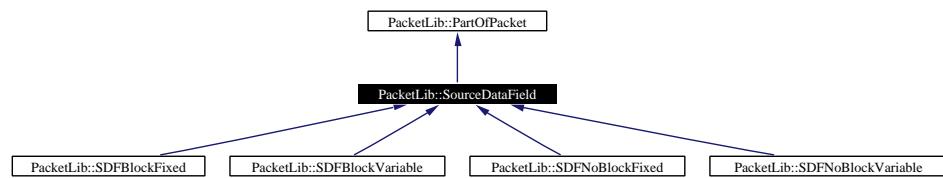
- **SocketServer.h**
- **SocketServer.cpp**

6.47 PacketLib::SourceDataField Class Reference

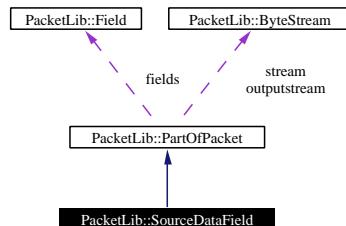
A generic source data field.

```
#include <SourceDataField.h>
```

Inheritance diagram for PacketLib::SourceDataField:



Collaboration diagram for PacketLib::SourceDataField:



Public Methods

- `SourceDataField ()`
- `virtual word getMaxDimension ()=0`
- `virtual word getMaxDimension (word nblock)`
- `virtual bool loadFields (InputText &fp)=0 throw (PacketException*)`
- `virtual bool isBlock ()`
- `virtual bool isFixed ()`
- `virtual bool isNumberOfBlockFixed ()`
- `virtual word getMaxNumberOfBlock ()`
- `virtual word getMaxNumberOfElements (word nblock)`
- `virtual word getDimension ()`
- `virtual word getDimension (word block)`
- `virtual word getIndexOfNBlock ()`
- `virtual void setNumberOfRealDataBlock (word value)`
- `virtual word getSubFromNBlock ()`
- `virtual word getNumberOfRealDataBlock ()`
- `virtual Field * getFields (word index)=0`

- virtual **Field** * **getFields** (**word** block, **word** index)
- virtual **word** **getFieldValue** (**word** index)=0
- virtual **word** **getFieldValue** (**word** block, **word** index)
- virtual void **setFieldValue** (**word** index, **word** value)=0
- virtual void **setFieldValue** (**word** block, **word** index, **word** value)
- virtual **word** **getNumberOffFields** (**word** block)
- virtual **word** **getNumberOffFields** ()
- virtual void **setNumberOfRealElement** (**word** block, **word** value)
- virtual **word** **getNumberOfRealElement** (**word** block)
- bool **get_reset_output_stream** () const
- void **set_reset_output_stream** (bool value)

Protected Attributes

- bool **isblock**
- bool **fixed**
- bool **numberOfBlockFixed**
- **word** **maxNumberOfBlock**
- **word** **indexOfNBlock**
- **word** **numberOfRealDataBlock**
- **word** **subFromNBlock**
- bool **reset_output_stream**

6.47.1 Detailed Description

A generic source data field.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/SourceDataField.h,v
2003/05/02 08:18:00 agile Exp

1.4

Id:

SourceDataField.h,v 1.4 2003/05/02 08:18:00 agile Exp

Revision:

1.4

Definition at line 35 of file SourceDataField.h.

6.47.2 Constructor & Destructor Documentation

6.47.2.1 `SourceDataField::SourceDataField ()`

Definition at line 7 of file SourceDataField.cpp.

References fixed, indexOfNBlock, isblock, maxNumberOfBlock, number_of_BlockFixed, number_of_RealDataBlock, reset_output_stream, and subFromNBlock.

6.47.3 Member Function Documentation

6.47.3.1 `bool SourceDataField::get_reset_output_stream () const`

Definition at line 170 of file SourceDataField.cpp.

References reset_output_stream.

6.47.3.2 `word SourceDataField::getDimension (word block) [virtual]`

Gets the current dimension (in byte) for a single block.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 148).

Definition at line 157 of file SourceDataField.cpp.

References PacketLib::word.

6.47.3.3 `word SourceDataField::getDimension () [virtual]`

Gets the current dimension (in byte) of the source data field.

Reimplemented from **PacketLib::PartOfPacket** (p. 140).

Reimplemented in **PacketLib::SDFBlockFixed** (p. 148).

Definition at line 127 of file SourceDataField.cpp.

References PacketLib::word.

Referenced by PacketLib::SDFNoBlockFixed::getMaxDimension.

6.47.3.4 `Field * SourceDataField::getFields (word block, word index) [virtual]`

Gets a field of the source data field.

Parameters:

block The block that contains the field

index The number of field in the source data field (into the block).

Returns:

The field.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 148).

Definition at line 148 of file SourceDataField.cpp.

References PacketLib::word.

**6.47.3.5 virtual Field* PacketLib::SourceDataField::getFields
(word *index*) [pure virtual]**

Gets a field of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The field.

Reimplemented from **PacketLib::PartOfPacket** (p. 141).

Implemented in **PacketLib::SDFBlockFixed** (p. 149).

**6.47.3.6 word SourceDataField::getFieldValue (word *block*, word
index) [virtual]**

Gets a field value of the source data field.

Parameters:

block The block that contains the field.

index The number of field in the source data field (into the block).

Returns:

The value of field.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 149).

Definition at line 153 of file SourceDataField.cpp.

References PacketLib::word.

**6.47.3.7 virtual word PacketLib::SourceDataField::getFieldValue
(word *index*) [pure virtual]**

Gets a field value of the source data field.

Parameters:

index The number of field in the source data field.

Returns:

The value of field.

Reimplemented from **PacketLib::PartOfPacket** (p. 141).

Implemented in **PacketLib::SDFBLOCKFixed** (p. 149).

6.47.3.8 word SourceDataField::getIndexOfNBlock () [virtual]

Gets the index of field in the data field header that contains the number of block in the source data field.

Definition at line 80 of file SourceDataField.cpp.

References indexOfNBlock, and PacketLib::word.

6.47.3.9 word SourceDataField::getMaxDimension (word *nblock*) [virtual]

Gets the total max dimension in bytes of source data field for each block

Parameters:

nblock Number of block

Reimplemented in **PacketLib::SDFBLOCKFixed** (p. 150).

Definition at line 162 of file SourceDataField.cpp.

References PacketLib::word.

6.47.3.10 virtual word PacketLib::SourceDataField::getMaxDimension () [pure virtual]

Gets the total max dimension in bytes of source data field

Implemented in **PacketLib::SDFBLOCKFixed** (p. 150).

6.47.3.11 word SourceDataField::getMaxNumberOfBlock () [virtual]**Returns:**

Returns the max number of block for this source data field.

Definition at line 54 of file SourceDataField.cpp.

References maxNumberOfBlock, and PacketLib::word.

Referenced by **PacketLib::SDFBLOCKVariable::getMaxDimension**, and **PacketLib::SDFBLOCKFixed::setByteStream**.

6.47.3.12 word SourceDataField::getMaxNumberOfElements (word *nblock*) [virtual]**Returns:**

Returns the max number of elements for each block.

Parameters:

nblock Number of block.

Reimplemented in **PacketLib::SDFBlockVariable** (p. 158).

Definition at line 166 of file SourceDataField.cpp.

References PacketLib::word.

6.47.3.13 word SourceDataField::getNumberOfFields () [virtual]

Returns the total number of fields for the source data field.

Reimplemented from **PacketLib::PartOfPacket** (p. 141).

Reimplemented in **PacketLib::SDFBlockFixed** (p. 150).

Definition at line 113 of file SourceDataField.cpp.

References PacketLib::word.

6.47.3.14 word SourceDataField::getNumberOfFields (word *block*) [virtual]

Returns the number of fields for each block.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 150).

Definition at line 120 of file SourceDataField.cpp.

References PacketLib::word.

6.47.3.15 word SourceDataField::getNumberOfRealDataBlock () [virtual]**Returns:**

Returns the number of real data block for the current packet.

Definition at line 102 of file SourceDataField.cpp.

References *numberOfRealDataBlock*, and PacketLib::word.

Referenced by PacketLib::SDFBlockVariable::generateStream, PacketLib::SDFBlockFixed::generateStream, PacketLib::SDFBlockVariable::getDimension, PacketLib::SDFBlockVariable::getFields, PacketLib::SDFBlockFixed::getFields, PacketLib::SDFBlockVariable::getNumberOfFields,

PacketLib::SDFBlockFixed::getNumberOfFields, PacketLib::SDFBlockVariable::printValue, PacketLib::SDFBlockFixed::printValue, PacketLib::SDFBlockVariable::setByteStream, PacketLib::SDFBlockFixed::setByteStream, PacketLib::SDFBlockVariable::setOutputStream, and PacketLib::SDFBlockFixed::setOutputStream.

6.47.3.16 word SourceDataField::getNumberOfRealElement (word block) [virtual]

Returns the number of real elements for each block.

Reimplemented in **PacketLib::SDFBlockVariable** (p. 159).

Definition at line 137 of file SourceDataField.cpp.

References PacketLib::word.

6.47.3.17 word SourceDataField::getSubFromNBlock () [virtual]

Definition at line 64 of file SourceDataField.cpp.

References subFromNBlock, and PacketLib::word.

6.47.3.18 bool SourceDataField::isBlock () [virtual]

Returns:

true if this source data field is compounded of blocks.

Definition at line 24 of file SourceDataField.cpp.

References isblock.

6.47.3.19 bool SourceDataField::isFixed () [virtual]

Returns:

true if the dimension of each block is fixed.

Definition at line 34 of file SourceDataField.cpp.

References fixed.

6.47.3.20 bool SourceDataField::isNumberOfBlockFixed () [virtual]

Returns:

true if the number of block is fixed.

Definition at line 44 of file SourceDataField.cpp.

References numberOfBlockFixed.

```
6.47.3.21 virtual bool PacketLib::SourceDataField::loadFields  
          (InputText & fp) throw (PacketException*) [pure  
          virtual]
```

Loads the field form .packet file. See **PartOfPacket** (p. 138) class.

Reimplemented from **PacketLib::PartOfPacket** (p. 142).

Implemented in **PacketLib::SDFBlockFixed** (p. 151).

```
6.47.3.22 void SourceDataField::set_reset_output_stream (bool  
              value)
```

Definition at line 180 of file SourceDataField.cpp.

References reset_output_stream.

```
6.47.3.23 void SourceDataField::setFieldValue (word block, word  
          index, word value) [virtual]
```

Sets a field value of the source data field.

Parameters:

block The block that contains the field.

index The number of field in the source data field (into the block).

value The value must be setted.

Reimplemented in **PacketLib::SDFBlockFixed** (p. 151).

Definition at line 146 of file SourceDataField.cpp.

References PacketLib::word.

```
6.47.3.24 virtual void PacketLib::SourceDataField::setFieldValue  
          (word index, word value) [pure virtual]
```

Sets a field value of the source data field.

Parameters:

index The number of field in the source data field.

value The value must be setted.

Reimplemented from **PacketLib::PartOfPacket** (p. 143).

Implemented in **PacketLib::SDFBlockFixed** (p. 152).

```
6.47.3.25 void SourceDataField::setNumberOfRealDataBlock  
          (word value) [virtual]
```

Sets the number of real data block for the current packet.

Definition at line 90 of file SourceDataField.cpp.

References `numberOfRealDataBlock`, `reset_output_stream`, and `PacketLib::word`.

6.47.3.26 void SourceDataField::setNumberOfRealElement (word *block*, word *value*) [virtual]

Sets the number of real elements for each block.

Parameters:

block Number of block. \value value The value must be setted.

Reimplemented in **PacketLib::SDFBlockVariable** (p. 161).

Definition at line 141 of file SourceDataField.cpp.

References `PacketLib::word`.

6.47.4 Member Data Documentation

6.47.4.1 bool PacketLib::SourceDataField::fixed [protected]

Indicates if the dimension of block is fixed or variable

Definition at line 225 of file SourceDataField.h.

Referenced by `isFixed`, `PacketLib::SDFBlockFixed::SDFBlockFixed`, `PacketLib::SDFBlockVariable::SDFBlockVariable`, `PacketLib::SDFNoBlockFixed::SDFNoBlockFixed`, `PacketLib::SDFNoBlockVariable::SDFNoBlockVariable`, and `SourceDataField`.

6.47.4.2 word PacketLib::SourceDataField::indexOfNBlock [protected]

Definition at line 247 of file SourceDataField.h.

Referenced by `getIndexOfNBlock`, and `SourceDataField`.

6.47.4.3 bool PacketLib::SourceDataField::isblock [protected]

This attribute indicates if the source data field is structured in block

Definition at line 219 of file SourceDataField.h.

Referenced by `isBlock`, `PacketLib::SDFBlockFixed::SDFBlockFixed`, `PacketLib::SDFBlockVariable::SDFBlockVariable`, `PacketLib::SDFNoBlockFixed::SDFNoBlockFixed`, `PacketLib::SDFNoBlockVariable::SDFNoBlockVariable`, and `SourceDataField`.

**6.47.4.4 word PacketLib::SourceDataField::maxNumberOfBlock
[protected]**

Represents the number of blocks (if `numberOfBlockFixed` is true) or max number of blocks (if `numberOfBlockFixed` is false) in the structure of source data field. This variable not represent the effective number of block into the stream of data.

Definition at line 243 of file `SourceDataField.h`.

Referenced by `PacketLib::SDFBlockFixed::getMaxDimension`, `getMaxNumberOfBlock`, `PacketLib::SDFBlockVariable::setNumberOfRealElement`, and `SourceDataField`.

**6.47.4.5 bool PacketLib::SourceDataField::numberOfBlockFixed
[protected]**

Represents the type of number of block. If true the number of blocks are fixed, if false the number of blocks are variable

Definition at line 234 of file `SourceDataField.h`.

Referenced by `isNumberOfBlockFixed`, and `SourceDataField`.

**6.47.4.6 word PacketLib::SourceDataField::numberOfRealData-
Block [protected]**

Real number of blocks.

Definition at line 253 of file `SourceDataField.h`.

Referenced by `PacketLib::SDFBlockVariable::getDimension`, `PacketLib::SDFBlockFixed::getDimension`, `PacketLib::SDFBlockVariable::getFields`, `PacketLib::SDFBlockFixed::getFields`, `PacketLib::SDFBlockVariable::getMaxDimension`, `PacketLib::SDFBlockFixed::getMaxDimension`, `PacketLib::SDFBlockVariable::getNumberOfFields`, `PacketLib::SDFBlockFixed::getNumberOfFields`, `getNumberOfRealDataBlock`, `setNumberOfRealDataBlock`, `PacketLib::SDFBlockVariable::setNumberOfRealElement`, and `SourceDataField`.

**6.47.4.7 bool PacketLib::SourceDataField::reset_output_stream
[protected]**

Definition at line 259 of file `SourceDataField.h`.

Referenced by `get_reset_output_stream`, `set_reset_output_stream`, `setNumberOfRealDataBlock`, `PacketLib::SDFBlockVariable::setNumberOfRealElement`, and `SourceDataField`.

6.47.4.8 word PacketLib::SourceDataField::subFromNBlock [protected]

Definition at line 255 of file SourceDataField.h.

Referenced by getSubFromNBlock, PacketLib::SDFBlockFixed::SDFBlockFixed, and SourceDataField.

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

- **SourceDataField.h**
- **SourceDataField.cpp**

6.48 PacketLib::Utility Class Reference

A class with static method with common functionality.

```
#include <Utility.h>
```

Static Public Methods

- int **binaryToInteger** (string bin)
- int **convertToInteger** (string bin)
- int **convertToInteger** (char *bin)
- *unsigned **getbits** (word x, int p, int n)
- unsigned **getbits2** (word x, int p, int n)
- string * **wordToBinary** (word w, byte dim)
- char * **wordToBinary2** (dword w, unsigned int dim)
- char * **format_output** (dword data, bool dec, bool hex, bool bin)
- string * **integerToString** (int n)
- char * **stringToHexadecimal** (byte *s, unsigned dim, bool bigendian=true)
- long **strlen** (char *s)
- char * **byteToHexadecimal** (byte b)
- int **hexadecimalToInteger** (string hex)
- char * **ftimeToChar** (word MSB, word LSB, word ms)
- char * **ftimeToChar** (struct tm *res, word ms)
- char * **integerToString2** (int n, char *str)
- char * **extractFileName** (char *path)
- char * **charcat** (char *, char)
- int **getchildpid** (int startpid, char *procname, int maxsearch=1000)
- bool **getchildpidlive** (int pid)

6.48.1 Detailed Description

A class with static method with common functionality.

Date:

2003/05/02 08:18:00

Header:

/home/cvs/PacketLib/packetlib/packet/Utility.h,v 1.5 2003/05/02
08:18:00 agile Exp

Id:

Utility.h,v 1.5 2003/05/02 08:18:00 agile Exp

Revision:

1.5

Definition at line 34 of file Utility.h.

6.48.2 Member Function Documentation

6.48.2.1 int Utility::binaryToInteger (string *bin*) [static]

Definition at line 62 of file Utility.cpp.

Referenced by convertToInteger.

6.48.2.2 char * Utility::byteToHexadecimal (byte *b*) [static]

Definition at line 319 of file Utility.cpp.

References PacketLib::byte.

6.48.2.3 char * Utility::charcat (char * *c1*, char *c2*) [static]

Definition at line 426 of file Utility.cpp.

References strlen.

6.48.2.4 int Utility::convertToInteger (char * *bin*) [static]

Definition at line 49 of file Utility.cpp.

6.48.2.5 int Utility::convertToInteger (string *bin*) [static]

Definition at line 27 of file Utility.cpp.

References binaryToInteger.

6.48.2.6 char * Utility::extractFileName (char * *path*) [static]

Definition at line 404 of file Utility.cpp.

6.48.2.7 char * Utility::format_output (dword *data*, bool *dec*, bool *hex*, bool *bin*) [static]

Definition at line 135 of file Utility.cpp.

References PacketLib::dword, and wordToBinary2.

6.48.2.8 char * Utility::ftimeToChar (struct tm * *res*, word *ms*) [static]

Definition at line 371 of file Utility.cpp.

References PacketLib::word.

6.48.2.9 `char * Utility::ftimeToChar (word MSB, word LSB, word ms) [static]`

Definition at line 353 of file Utility.cpp.

References PacketLib::word.

6.48.2.10 `unsigned Utility::getbits (word x, int p, int n) [static]`

Restituisce n bit (allineati a destra di x a partire dalla posizione p). Esempio:
getbits(x,4,3) restituisce bit 4, 3, 2 (allineamento a destra) 7 6 5 4 3 2 1 0

Definition at line 170 of file Utility.cpp.

References PacketLib::word.

Referenced by getbits2, wordToBinary, and wordToBinary2.

6.48.2.11 `unsigned Utility::getbits2 (word x, int p, int n) [static]`

Definition at line 183 of file Utility.cpp.

References getbits, and PacketLib::word.

6.48.2.12 `int Utility::getchildpid (int startpid, char * procname, int maxsearch = 1000) [static]`

Definition at line 437 of file Utility.cpp.

6.48.2.13 `bool Utility::getchildpidlive (int pid) [static]`

Definition at line 476 of file Utility.cpp.

6.48.2.14 `int Utility::hexadecimalToInteger (string hex) [static]`

Definition at line 94 of file Utility.cpp.

6.48.2.15 `string * Utility::integerToString (int n) [static]`

Definition at line 262 of file Utility.cpp.

6.48.2.16 `char * Utility::integerToString2 (int n, char * str) [static]`

Definition at line 384 of file Utility.cpp.

6.48.2.17 `char * Utility::stringToHexadecimal (byte * b, unsigned dim, bool bigEndian = true) [static]`

Definition at line 276 of file Utility.cpp.

References PacketLib::byte.

6.48.2.18 `long Utility::strlen (char * s) [static]`

Definition at line 339 of file Utility.cpp.

Referenced by charcat.

6.48.2.19 `string * Utility::wordToBinary (word w, byte dim) [static]`

Definition at line 194 of file Utility.cpp.

References PacketLib::byte, getbits, and PacketLib::word.

6.48.2.20 `char * Utility::wordToBinary2 (dword w, unsigned int dim) [static]`

Definition at line 230 of file Utility.cpp.

References PacketLib::dword, and getbits.

Referenced by format_output.

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

- `Utility.h`
- `Utility.cpp`

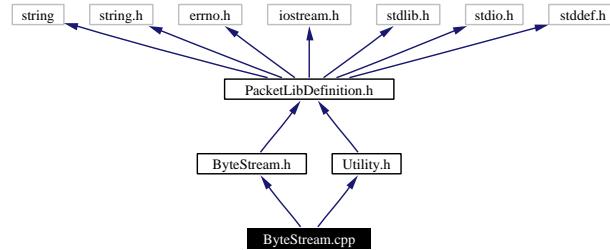
Chapter 7

File Documentation

7.1 ByteStream.cpp File Reference

```
#include "ByteStream.h"  
#include "Utility.h"
```

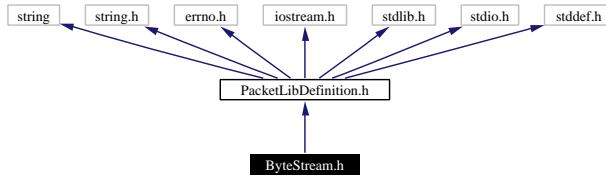
Include dependency graph for ByteStream.cpp:



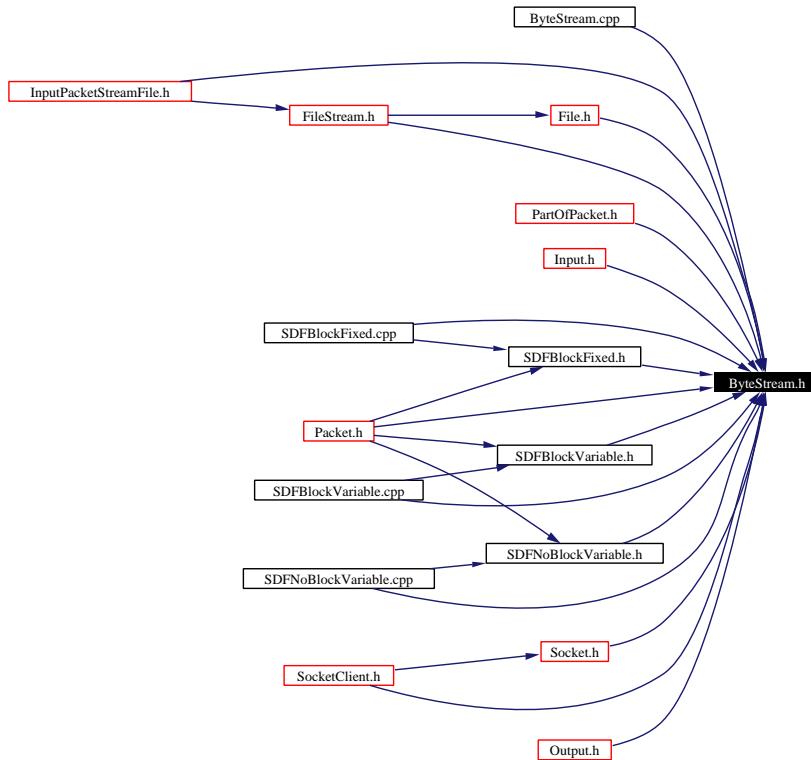
7.2 ByteStream.h File Reference

```
#include "PacketLibDefinition.h"
```

Include dependency graph for ByteStream.h:



This graph shows which files directly or indirectly include this file:



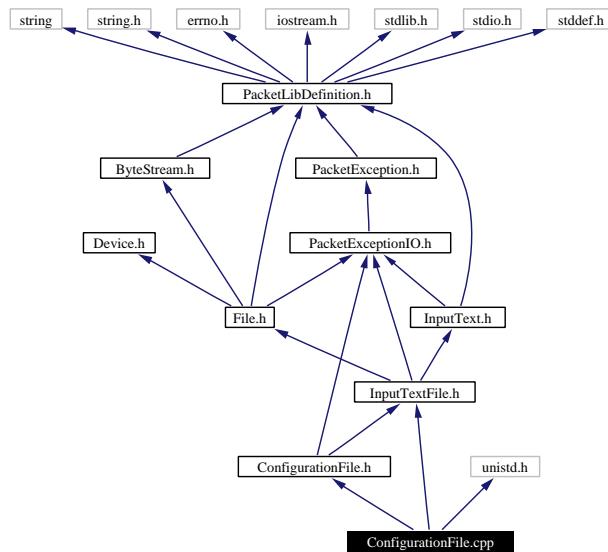
Namespaces

- namespace **PacketLib**

7.3 ConfigurationFile.cpp File Reference

```
#include "ConfigurationFile.h"  
#include "InputTextFile.h"  
#include <unistd.h>
```

Include dependency graph for ConfigurationFile.cpp:

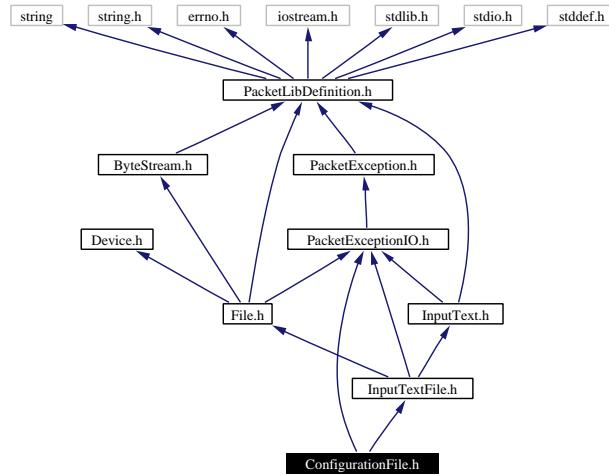


7.4 ConfigurationFile.h File Reference

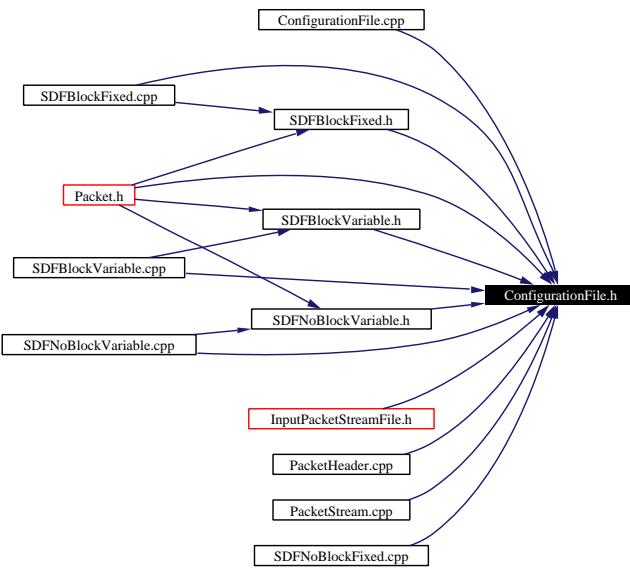
```
#include "PacketExceptionIO.h"
```

```
#include "InputTextFile.h"
```

Include dependency graph for ConfigurationFile.h:



This graph shows which files directly or indirectly include this file:



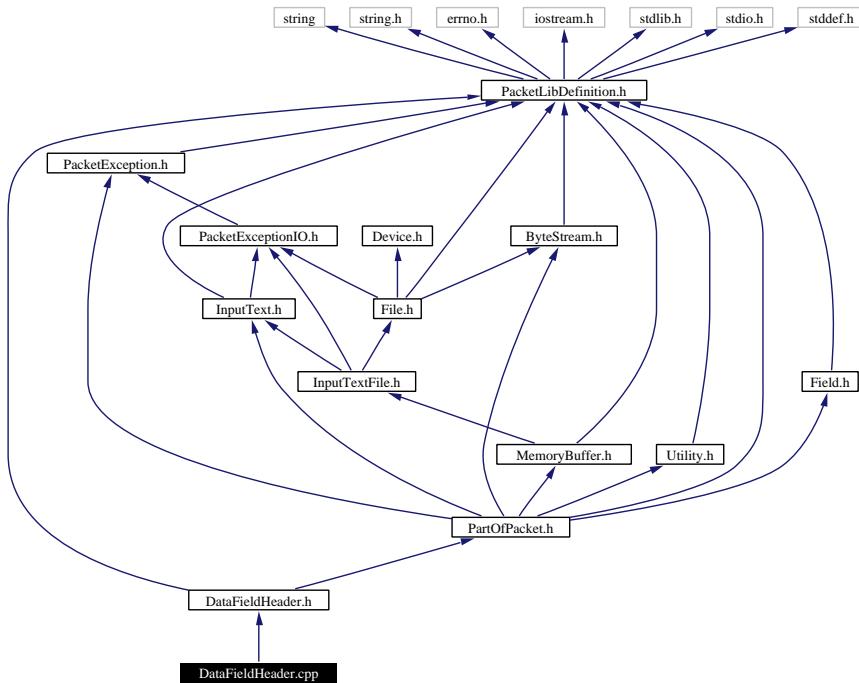
Namespaces

- namespace **PacketLib**

7.5 DataFieldHeader.cpp File Reference

```
#include "DataFieldHeader.h"
```

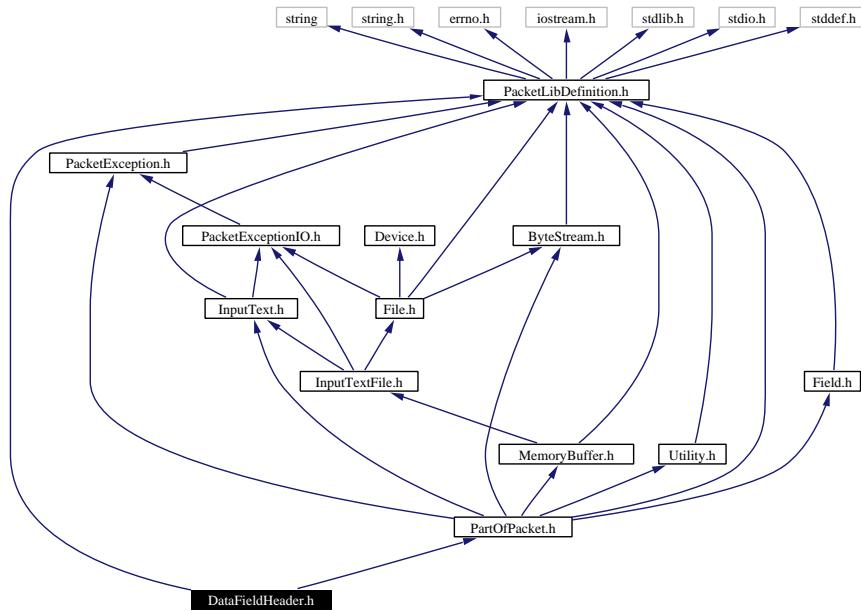
Include dependency graph for DataFieldHeader.cpp:



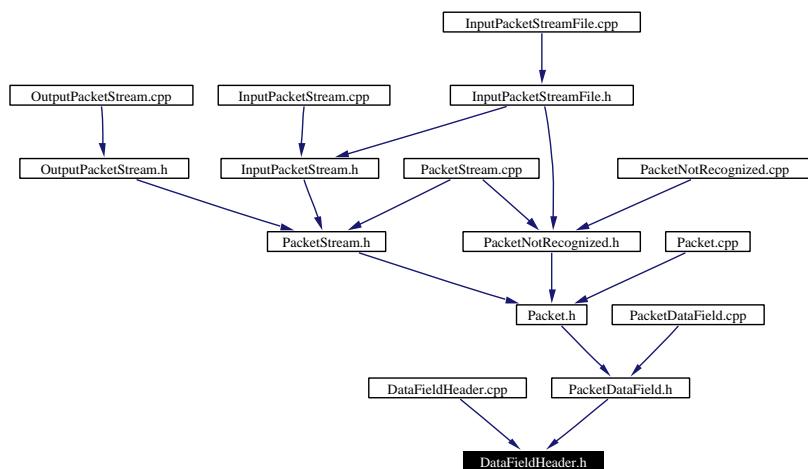
7.6 DataFieldHeader.h File Reference

```
#include "PacketLibDefinition.h"
#include "PartOfPacket.h"
```

Include dependency graph for DataFieldHeader.h:



This graph shows which files directly or indirectly include this file:



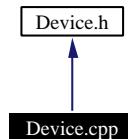
Namespaces

- namespace **PacketLib**

7.7 Device.cpp File Reference

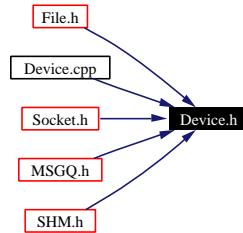
```
#include "Device.h"
```

Include dependency graph for Device.cpp:



7.8 Device.h File Reference

This graph shows which files directly or indirectly include this file:



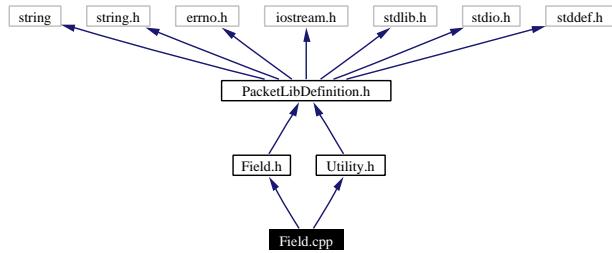
Namespaces

- namespace **PacketLib**

7.9 Field.cpp File Reference

```
#include "Field.h"  
#include "Utility.h"
```

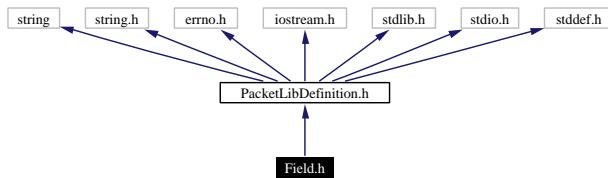
Include dependency graph for Field.cpp:



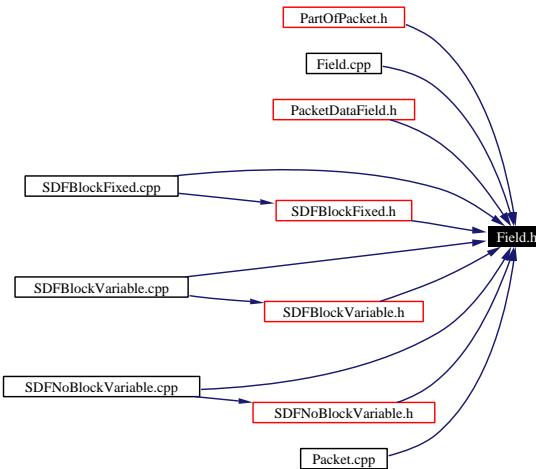
7.10 Field.h File Reference

```
#include "PacketLibDefinition.h"
```

Include dependency graph for Field.h:



This graph shows which files directly or indirectly include this file:



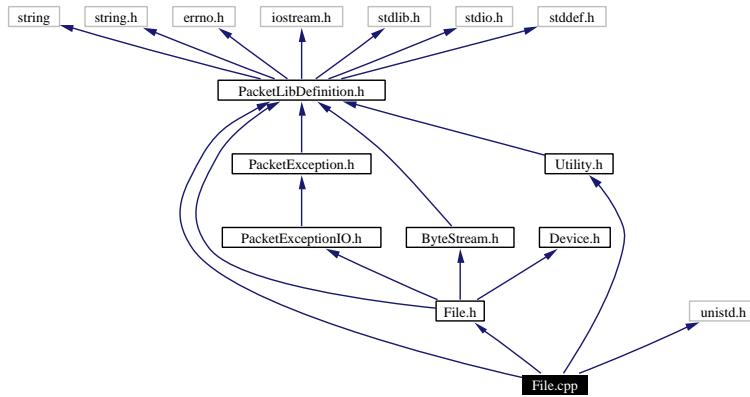
Namespaces

- namespace **PacketLib**

7.11 File.cpp File Reference

```
#include "File.h"  
#include "PacketLibDefinition.h"  
#include <unistd.h>  
#include "Utility.h"
```

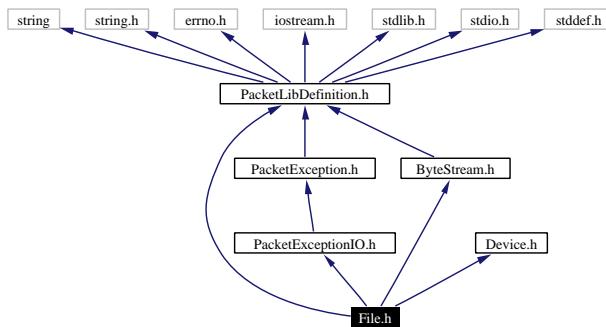
Include dependency graph for File.cpp:



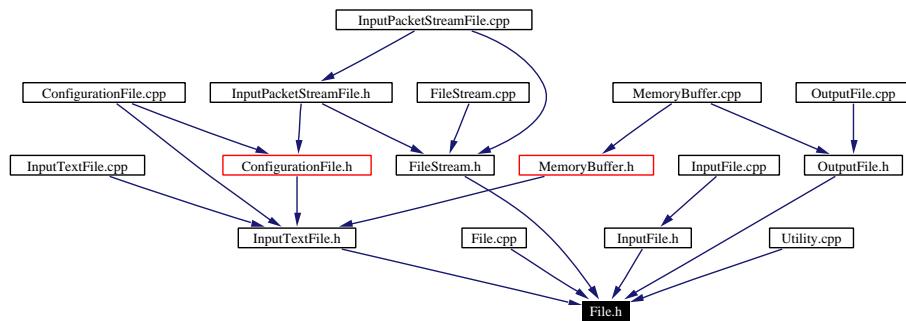
7.12 File.h File Reference

```
#include "PacketLibDefinition.h"
#include "Device.h"
#include "PacketExceptionIO.h"
#include "ByteStream.h"
```

Include dependency graph for File.h:



This graph shows which files directly or indirectly include this file:



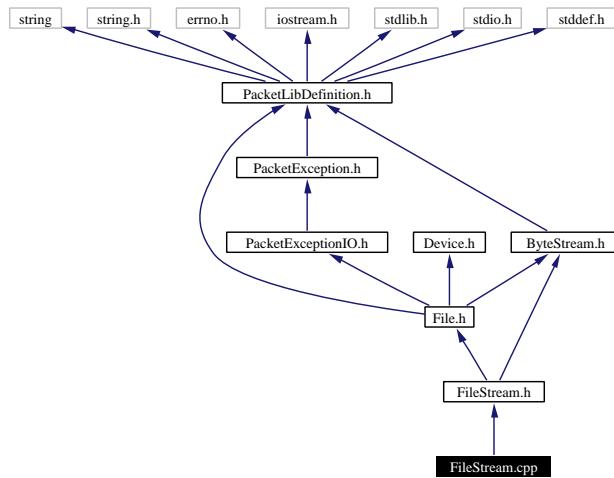
Namespaces

- namespace **PacketLib**

7.13 FileStream.cpp File Reference

```
#include "FileStream.h"
```

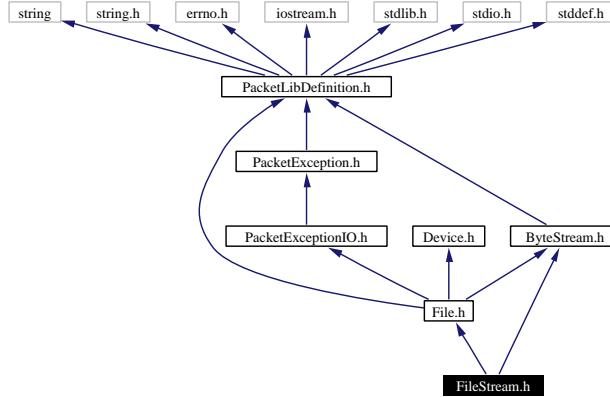
Include dependency graph for FileStream.cpp:



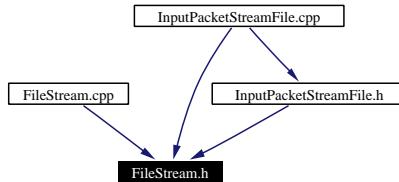
7.14 FileStream.h File Reference

```
#include "File.h"
#include "ByteStream.h"
```

Include dependency graph for FileStream.h:



This graph shows which files directly or indirectly include this file:



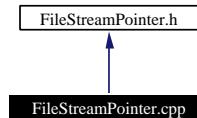
Namespaces

- namespace **PacketLib**

7.15 FileStreamPointer.cpp File Reference

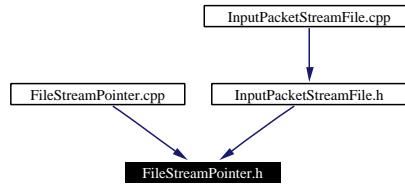
```
#include "FileStreamPointer.h"
```

Include dependency graph for FileStreamPointer.cpp:



7.16 FileStreamPointer.h File Reference

This graph shows which files directly or indirectly include this file:



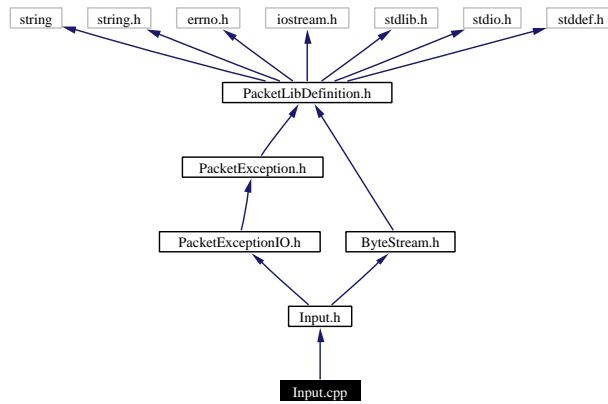
Namespaces

- namespace **PacketLib**

7.17 Input.cpp File Reference

```
#include "Input.h"
```

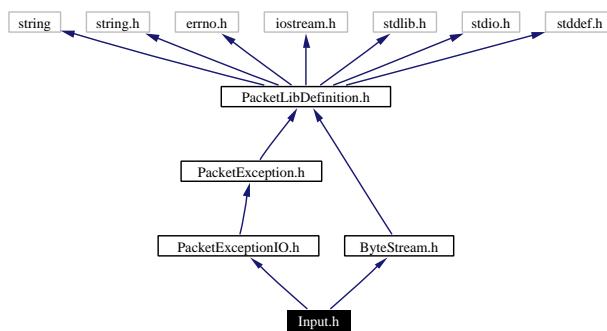
Include dependency graph for Input.cpp:



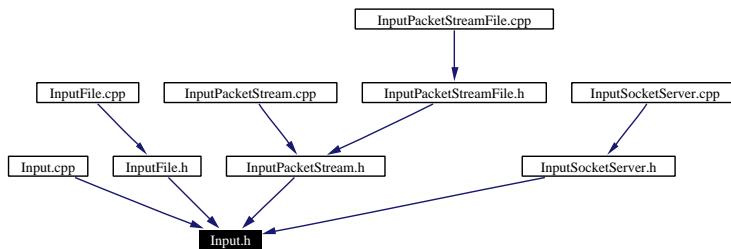
7.18 Input.h File Reference

```
#include "PacketExceptionIO.h"
#include "ByteStream.h"
```

Include dependency graph for Input.h:



This graph shows which files directly or indirectly include this file:



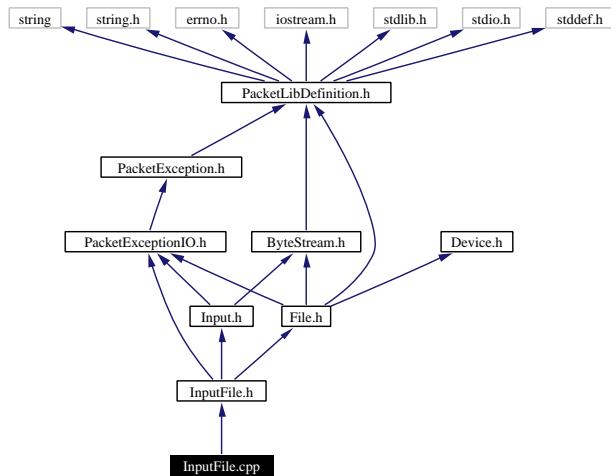
Namespaces

- namespace **PacketLib**

7.19 InputFile.cpp File Reference

```
#include "InputFile.h"
```

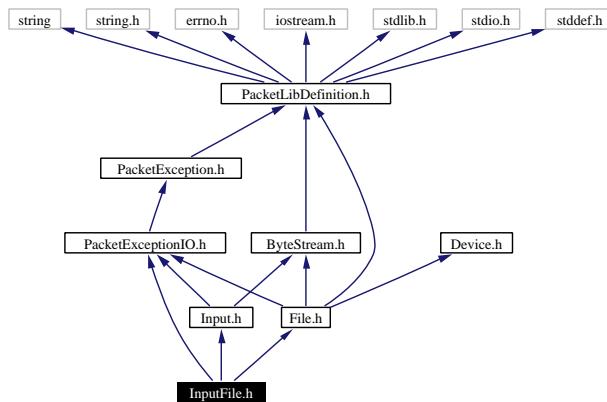
Include dependency graph for InputFile.cpp:



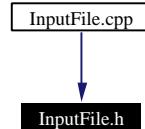
7.20 InputFile.h File Reference

```
#include "Input.h"
#include "PacketExceptionIO.h"
#include "File.h"
```

Include dependency graph for InputFile.h:



This graph shows which files directly or indirectly include this file:



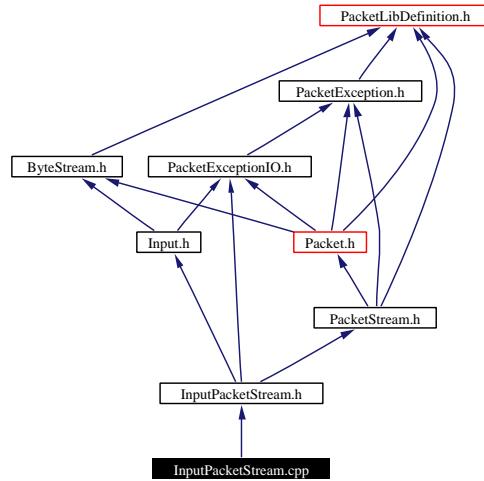
Namespaces

- namespace **PacketLib**

7.21 InputPacketStream.cpp File Reference

```
#include "InputPacketStream.h"
```

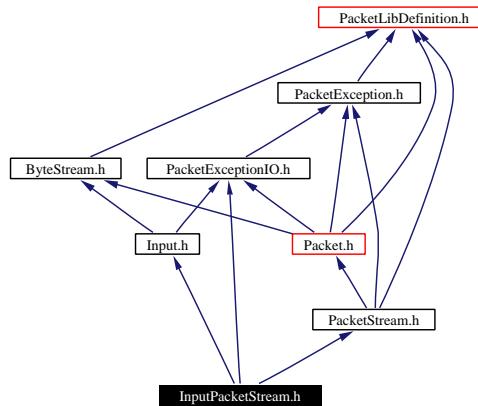
Include dependency graph for InputPacketStream.cpp:



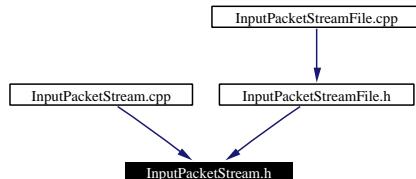
7.22 InputPacketStream.h File Reference

```
#include "PacketStream.h"
#include "Input.h"
#include "PacketExceptionIO.h"
```

Include dependency graph for InputPacketStream.h:



This graph shows which files directly or indirectly include this file:



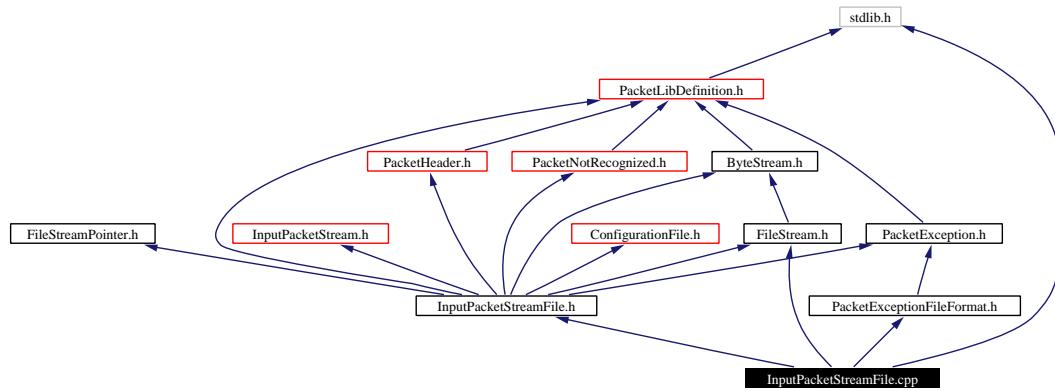
Namespaces

- namespace **PacketLib**

7.23 InputPacketStreamFile.cpp File Reference

```
#include "InputPacketStreamFile.h"
#include "FileStream.h"
#include <stdlib.h>
#include "PacketExceptionFileFormat.h"
```

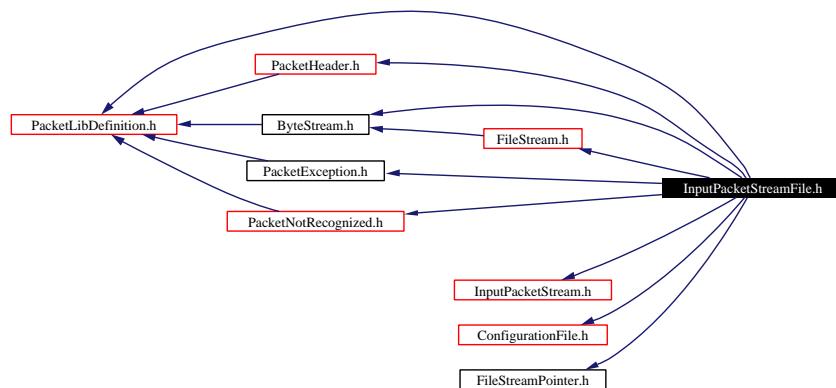
Include dependency graph for InputPacketStreamFile.cpp:



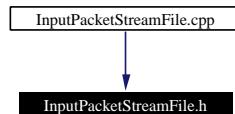
7.24 InputPacketStreamFile.h File Reference

```
#include "PacketLibDefinition.h"
#include "PacketHeader.h"
#include "InputPacketStream.h"
#include "ByteStream.h"
#include "ConfigurationFile.h"
#include "FileStreamPointer.h"
#include "PacketException.h"
#include "PacketNotRecognized.h"
#include "FileStream.h"
```

Include dependency graph for InputPacketStreamFile.h:



This graph shows which files directly or indirectly include this file:



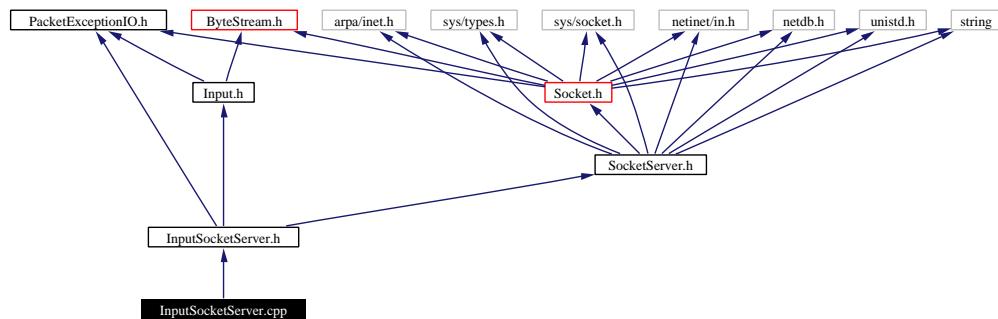
Namespaces

- namespace **PacketLib**

7.25 InputSocketServer.cpp File Reference

```
#include "InputSocketServer.h"
```

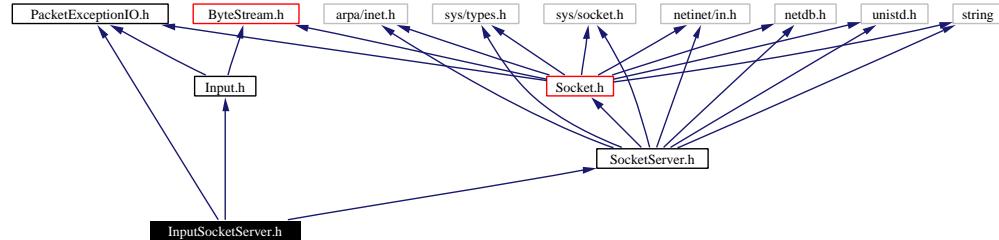
Include dependency graph for InputSocketServer.cpp:



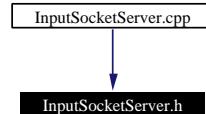
7.26 InputSocketServer.h File Reference

```
#include "Input.h"
#include "PacketExceptionIO.h"
#include "SocketServer.h"
```

Include dependency graph for InputSocketServer.h:



This graph shows which files directly or indirectly include this file:



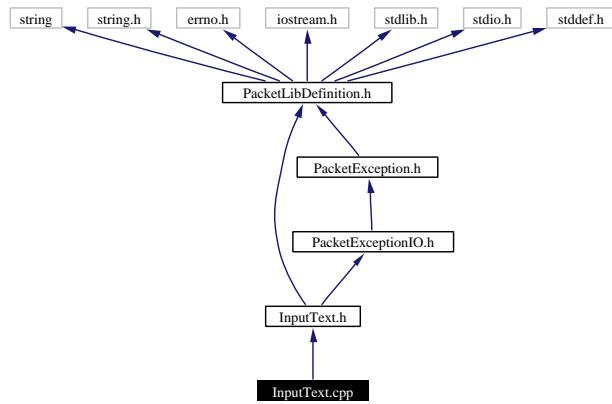
Namespaces

- namespace **PacketLib**

7.27 InputText.cpp File Reference

```
#include "InputText.h"
```

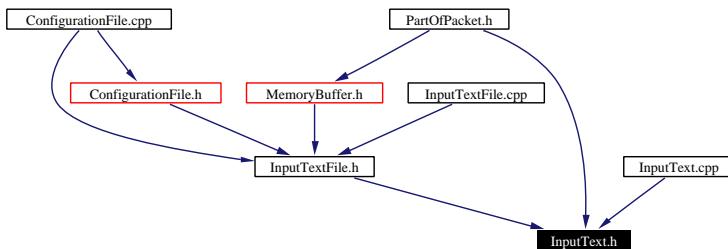
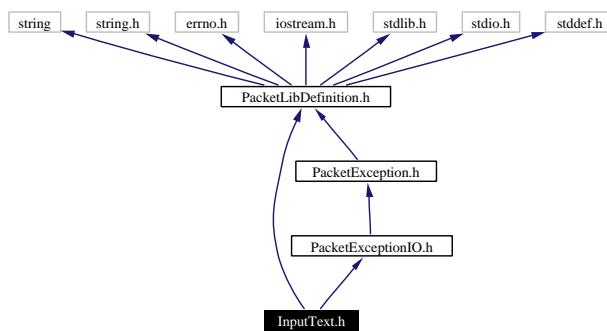
Include dependency graph for InputText.cpp:



7.28 InputText.h File Reference

```
#include "PacketLibDefinition.h"
#include "PacketExceptionI0.h"
```

Include dependency graph for InputText.h:



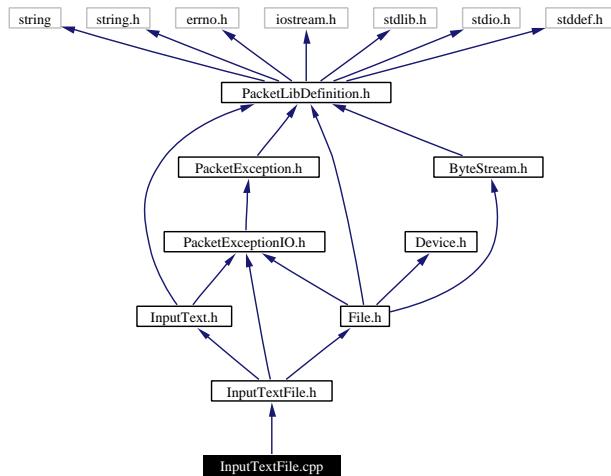
Namespaces

- namespace **PacketLib**

7.29 InputTextFile.cpp File Reference

```
#include "InputTextFile.h"
```

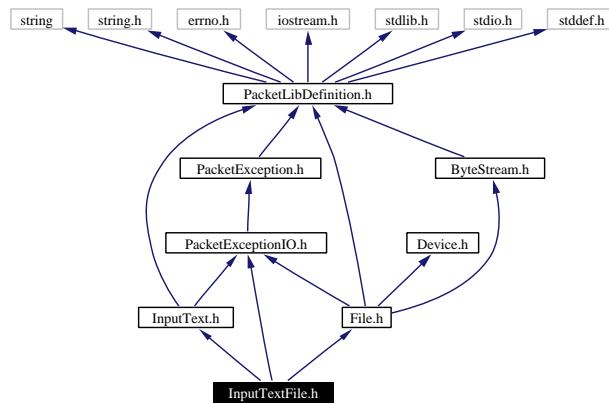
Include dependency graph for InputTextFile.cpp:



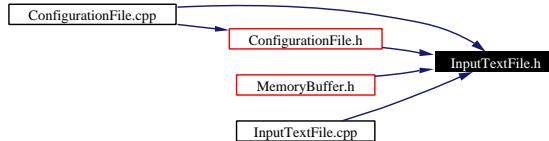
7.30 InputTextFile.h File Reference

```
#include "InputText.h"
#include "File.h"
#include "PacketExceptionIO.h"

Include dependency graph for InputTextFile.h:
```



This graph shows which files directly or indirectly include this file:



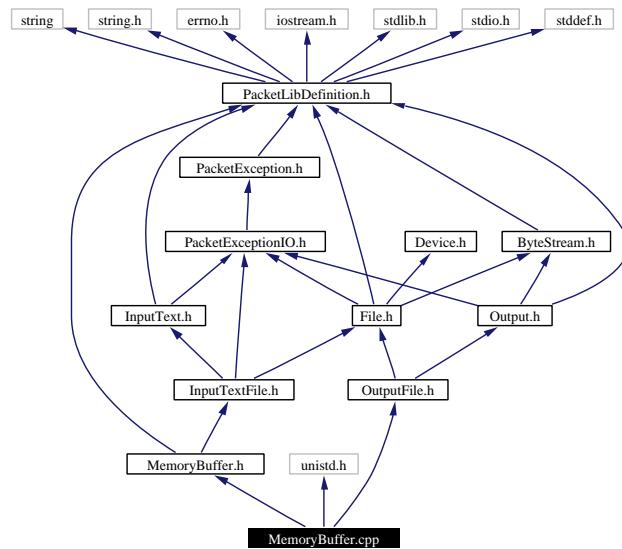
Namespaces

- namespace **PacketLib**

7.31 MemoryBuffer.cpp File Reference

```
#include "MemoryBuffer.h"  
#include <unistd.h>  
#include "OutputFile.h"
```

Include dependency graph for MemoryBuffer.cpp:

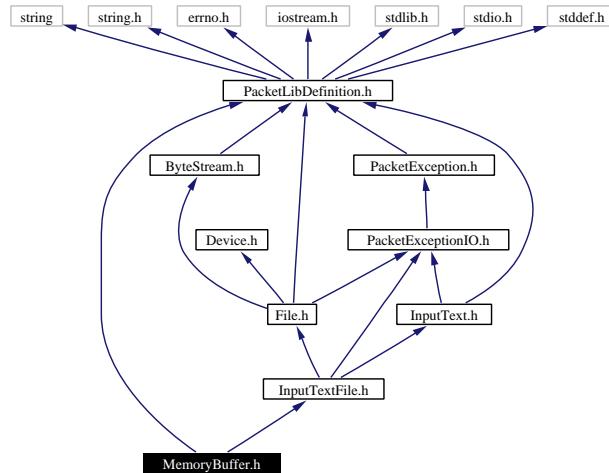


7.32 MemoryBuffer.h File Reference

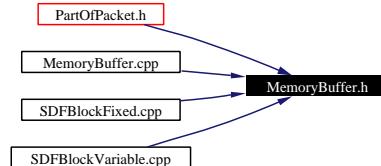
```
#include "PacketLibDefinition.h"
```

```
#include "InputTextFile.h"
```

Include dependency graph for MemoryBuffer.h:



This graph shows which files directly or indirectly include this file:



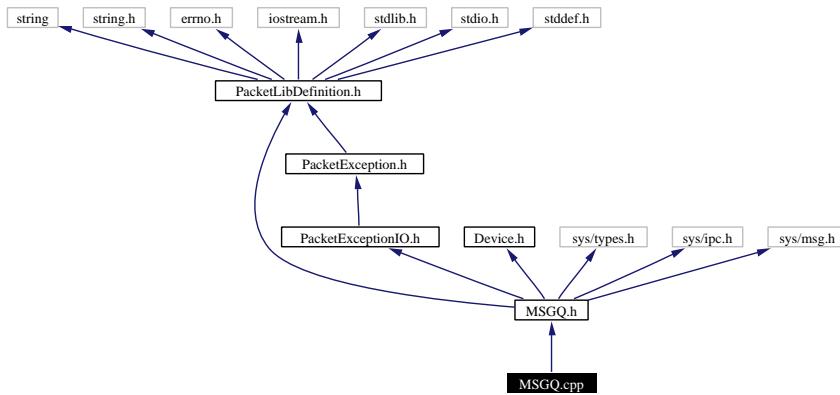
Namespaces

- namespace **PacketLib**

7.33 MSGQ.cpp File Reference

```
#include "MSGQ.h"
```

Include dependency graph for MSGQ.cpp:



Defines

- #define **PERMMSGQ** 0666
- #define **MSG_TYPE** 10

7.33.1 Define Documentation

7.33.1.1 #define MSG_TYPE 10

Definition at line 20 of file MSGQ.cpp.

7.33.1.2 #define PERMMSGQ 0666

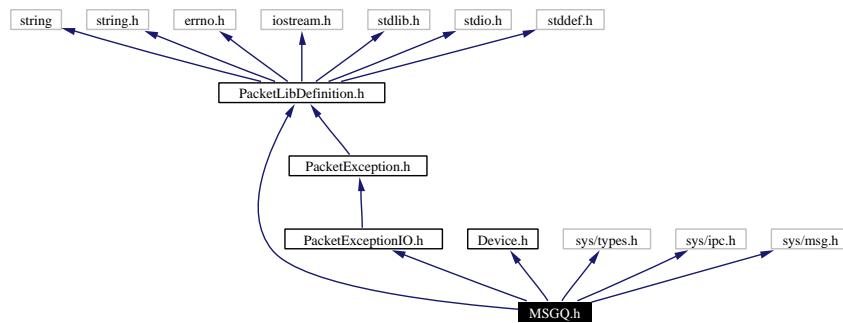
Definition at line 19 of file MSGQ.cpp.

Referenced by PacketLib::MSGQ::create, and PacketLib::MSGQ::open.

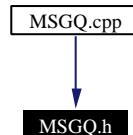
7.34 MSGQ.h File Reference

```
#include "PacketLibDefinition.h"
#include "Device.h"
#include "PacketExceptionIO.h"
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
```

Include dependency graph for MSGQ.h:



This graph shows which files directly or indirectly include this file:



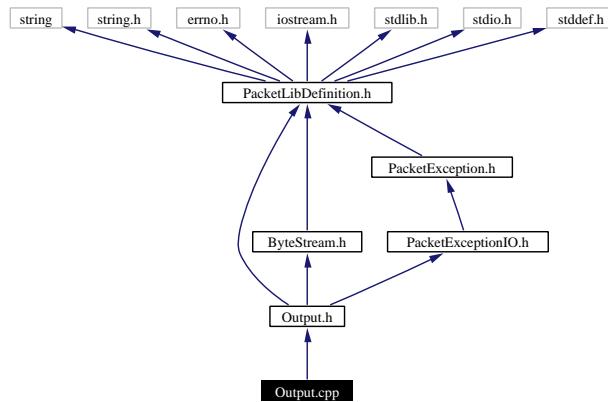
Namespaces

- namespace **PacketLib**

7.35 Output.cpp File Reference

```
#include "Output.h"
```

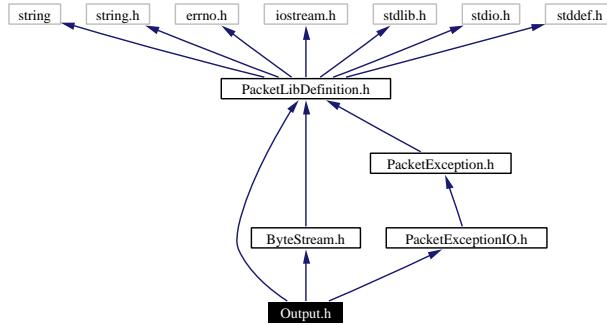
Include dependency graph for Output.cpp:



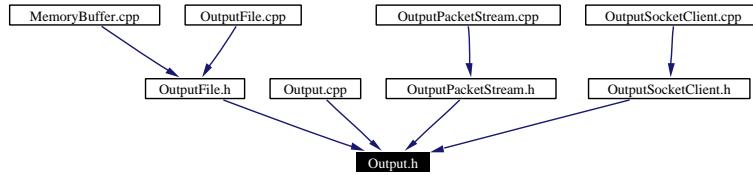
7.36 Output.h File Reference

```
#include "PacketLibDefinition.h"
#include "ByteStream.h"
#include "PacketExceptionIO.h"
```

Include dependency graph for Output.h:



This graph shows which files directly or indirectly include this file:



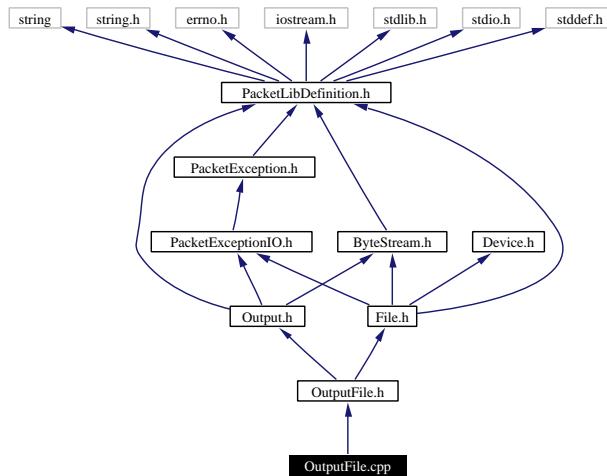
Namespaces

- namespace **PacketLib**

7.37 outputFile.cpp File Reference

```
#include "OutputFile.h"
```

Include dependency graph for outputFile.cpp:

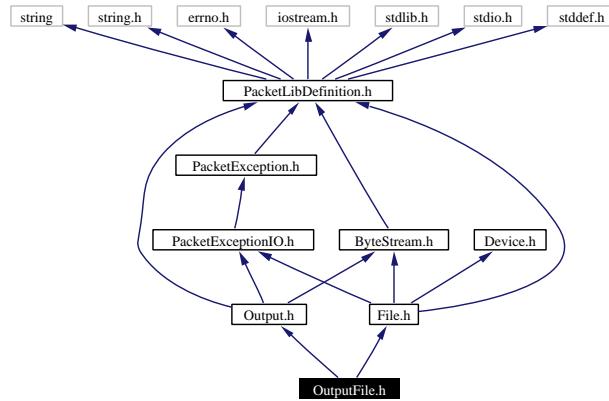


7.38 outputFile.h File Reference

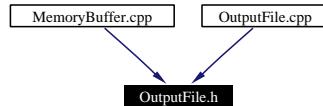
```
#include "Output.h"
```

```
#include "File.h"
```

Include dependency graph for outputFile.h:



This graph shows which files directly or indirectly include this file:



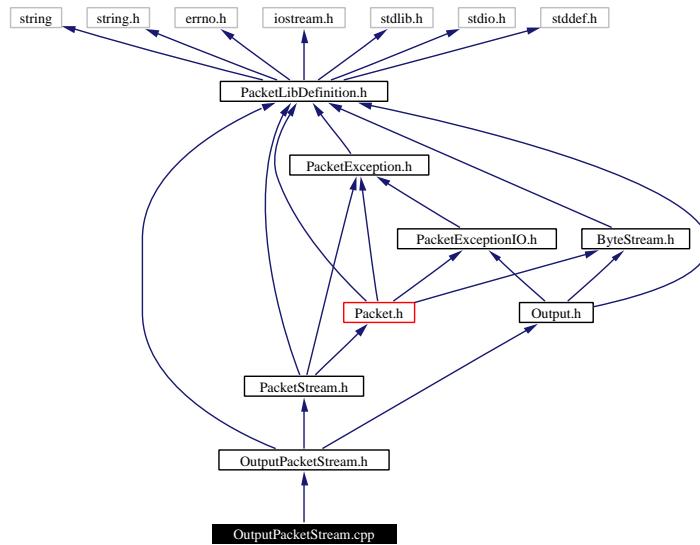
Namespaces

- namespace **PacketLib**

7.39 OutputPacketStream.cpp File Reference

```
#include "OutputPacketStream.h"
```

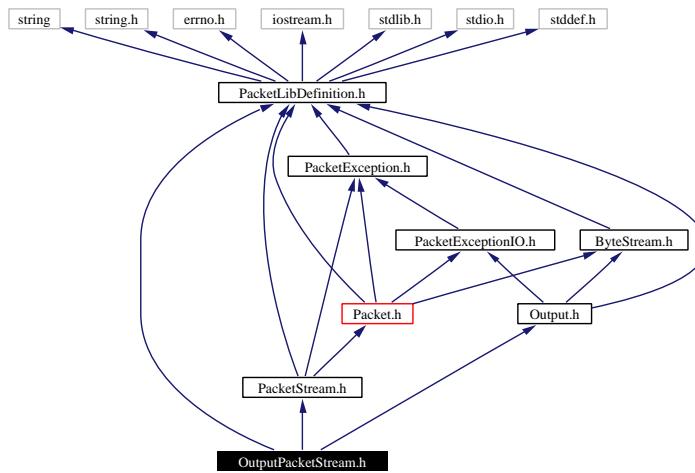
Include dependency graph for OutputPacketStream.cpp:



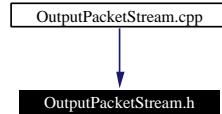
7.40 OutputPacketStream.h File Reference

```
#include "PacketLibDefinition.h"
#include "PacketStream.h"
#include "Output.h"
```

Include dependency graph for OutputPacketStream.h:



This graph shows which files directly or indirectly include this file:



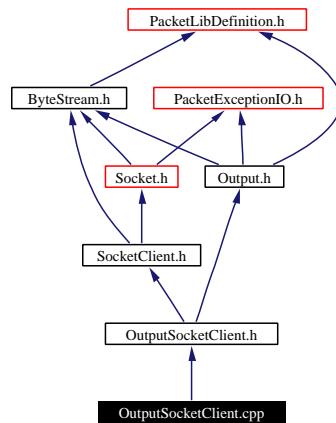
Namespaces

- namespace **PacketLib**

7.41 OutputSocketClient.cpp File Reference

```
#include "OutputSocketClient.h"
```

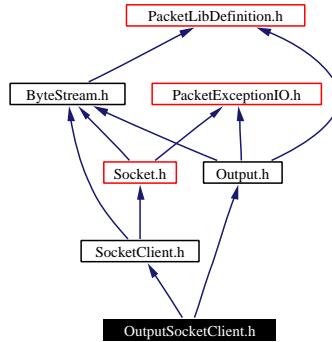
Include dependency graph for OutputSocketClient.cpp:



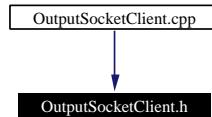
7.42 OutputSocketClient.h File Reference

```
#include "Output.h"
#include "SocketClient.h"

Include dependency graph for OutputSocketClient.h:
```



This graph shows which files directly or indirectly include this file:



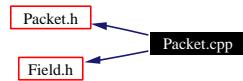
Namespaces

- namespace **PacketLib**

7.43 Packet.cpp File Reference

```
#include "Packet.h"  
#include "Field.h"
```

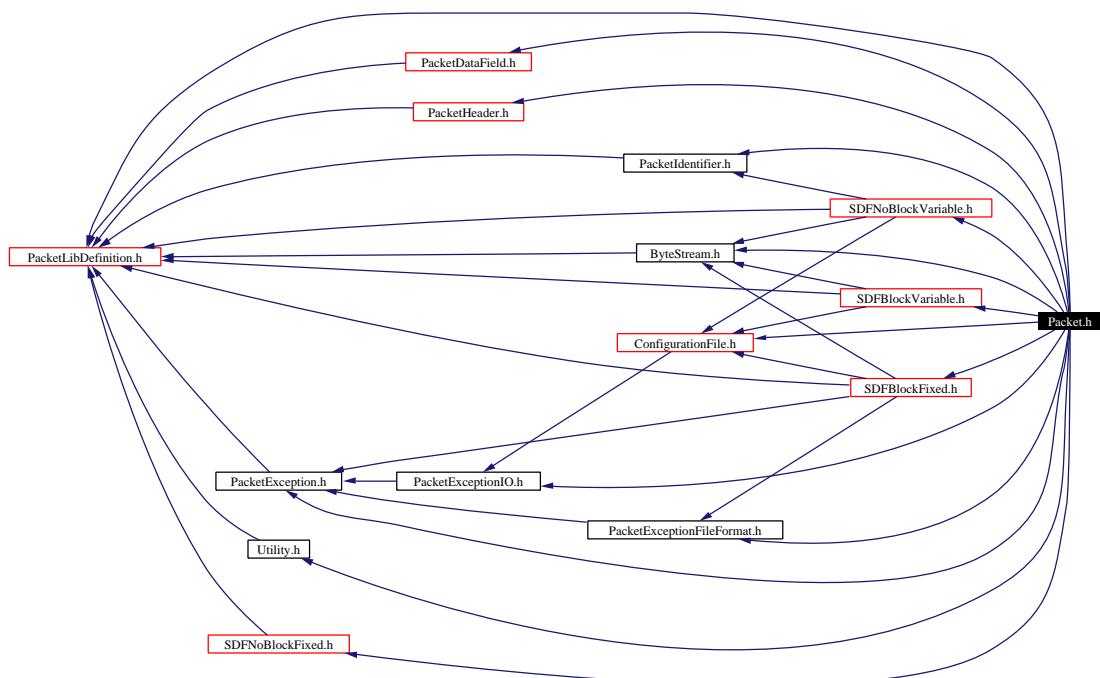
Include dependency graph for Packet.cpp:



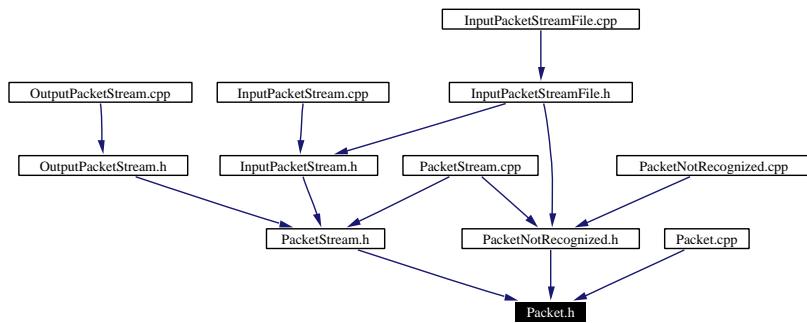
7.44 Packet.h File Reference

```
#include "PacketLibDefinition.h"
#include "PacketDataField.h"
#include "PacketHeader.h"
#include "PacketIdentifier.h"
#include "ByteStream.h"
#include "ConfigurationFile.h"
#include "Utility.h"
#include "SDFBlockFixed.h"
#include "SDFBlockVariable.h"
#include "SDFNoBlockFixed.h"
#include "SDFNoBlockVariable.h"
#include "PacketException.h"
#include "PacketExceptionIO.h"
#include "PacketExceptionFormatException.h"
```

Include dependency graph for Packet.h:



This graph shows which files directly or indirectly include this file:



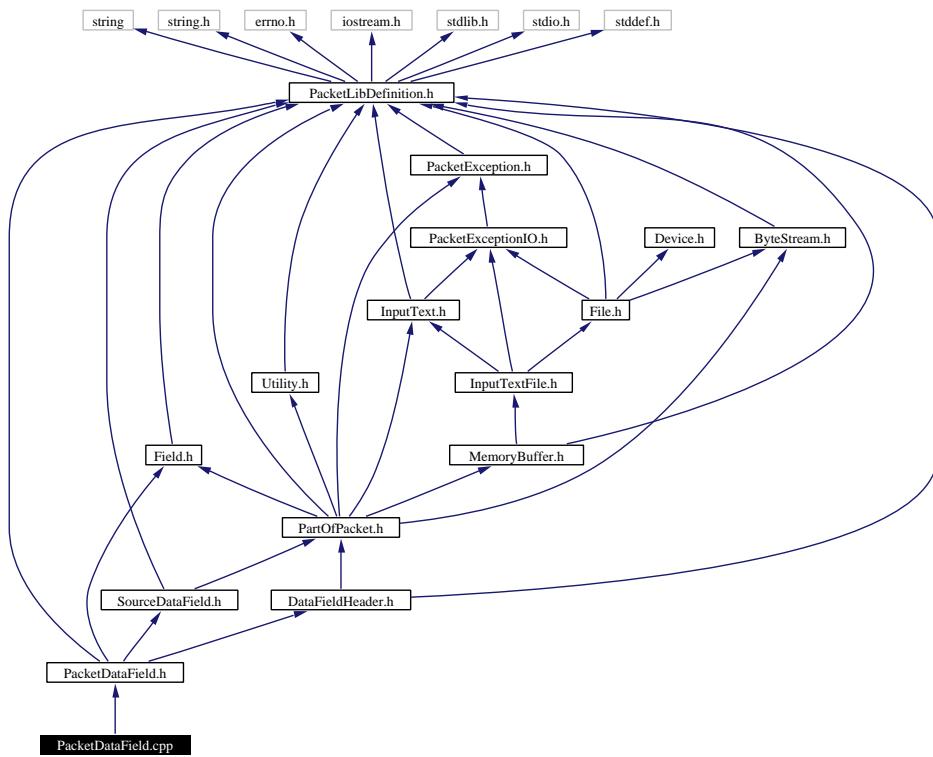
Namespaces

- namespace **PacketLib**

7.45 PacketDataField.cpp File Reference

```
#include "PacketDataField.h"
```

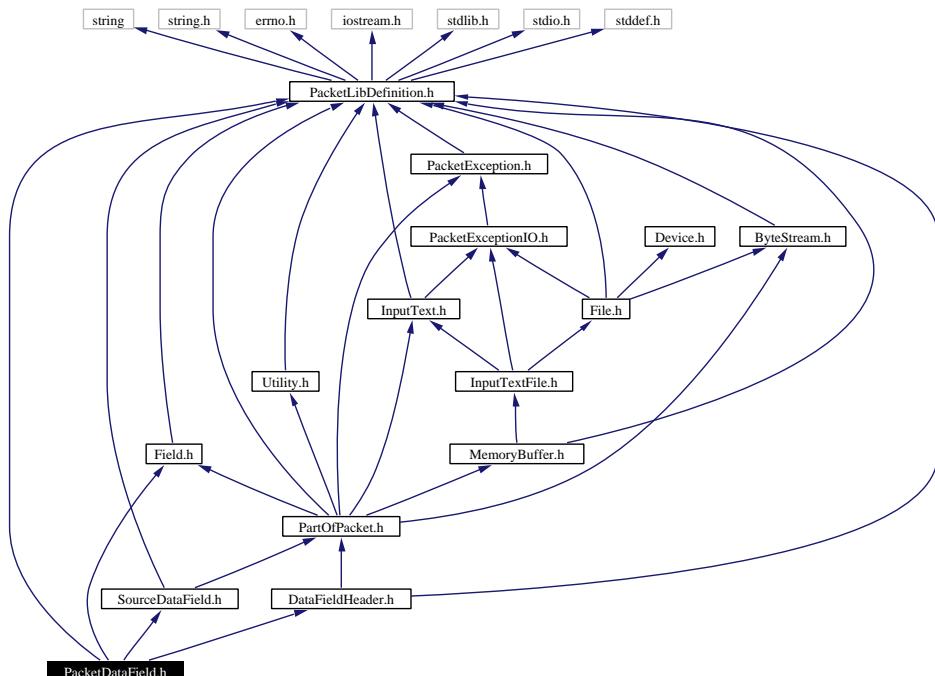
Include dependency graph for PacketDataField.cpp:



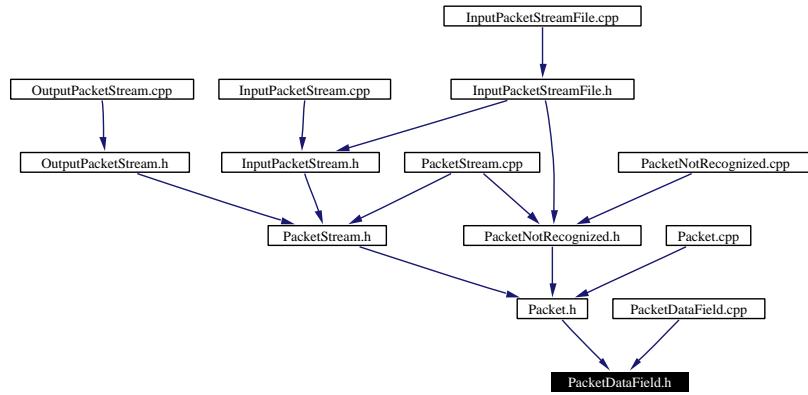
7.46 PacketDataField.h File Reference

```
#include "PacketLibDefinition.h"
#include "SourceDataField.h"
#include "DataFieldHeader.h"
#include "Field.h"
```

Include dependency graph for PacketDataField.h:



This graph shows which files directly or indirectly include this file:



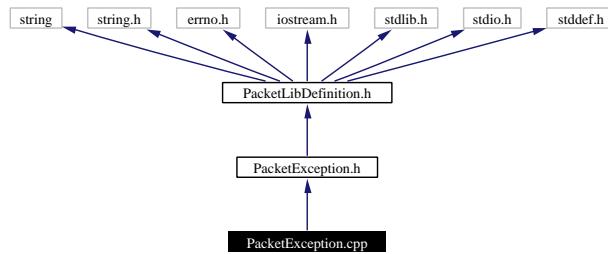
Namespaces

- namespace **PacketLib**

7.47 PacketException.cpp File Reference

```
#include "PacketException.h"
```

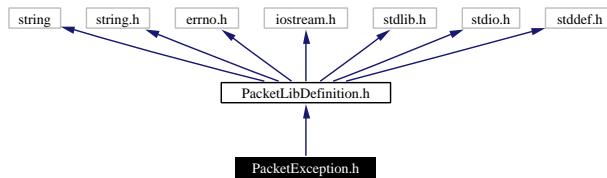
Include dependency graph for PacketException.cpp:



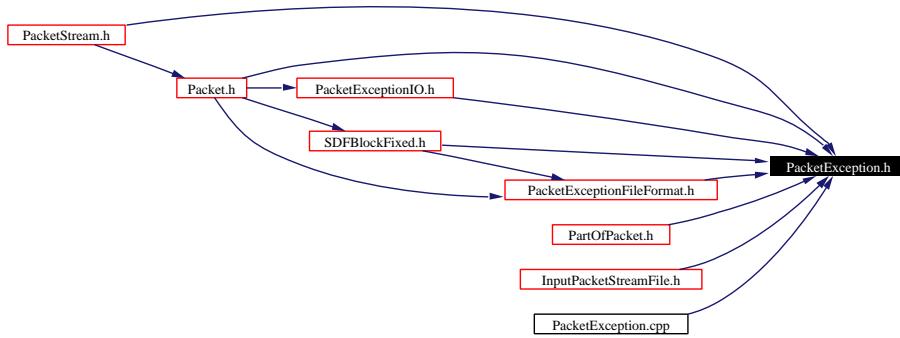
7.48 PacketException.h File Reference

```
#include "PacketLibDefinition.h"
```

Include dependency graph for PacketException.h:



This graph shows which files directly or indirectly include this file:



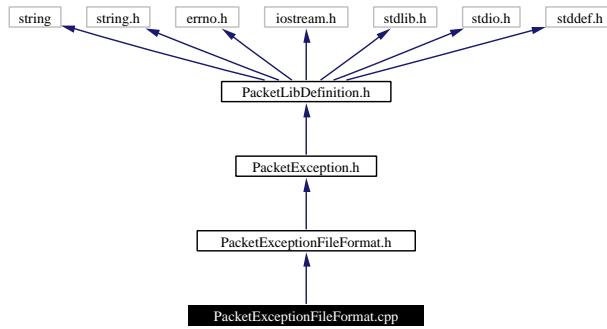
Namespaces

- namespace **PacketLib**

7.49 PacketExceptionFileFormat.cpp File Reference

```
#include "PacketExceptionFileFormat.h"
```

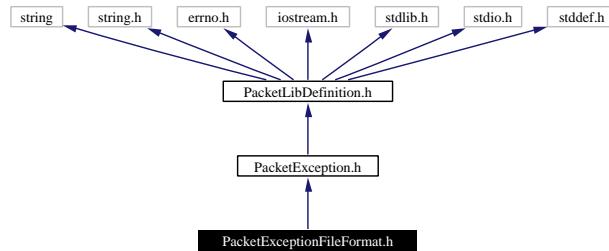
Include dependency graph for PacketExceptionFileFormat.cpp:



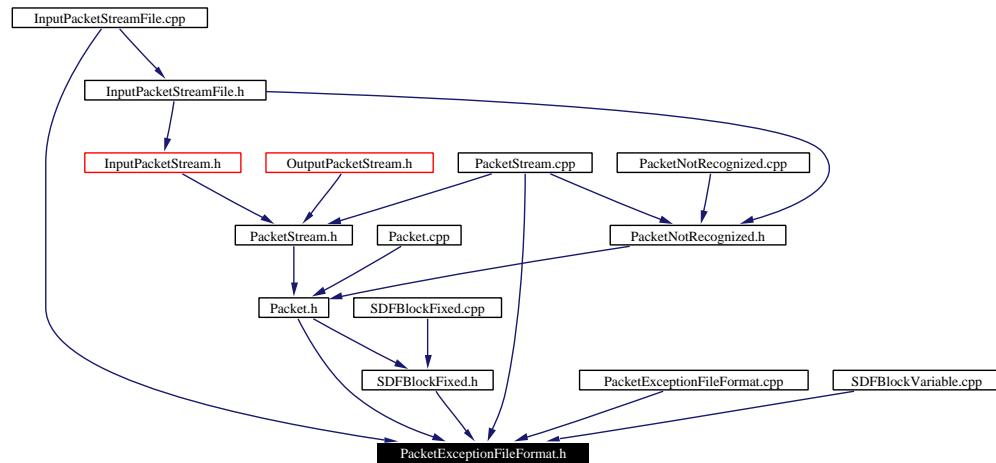
7.50 PacketExceptionFileFormat.h File Reference

```
#include "PacketException.h"
```

Include dependency graph for PacketExceptionFileFormat.h:



This graph shows which files directly or indirectly include this file:



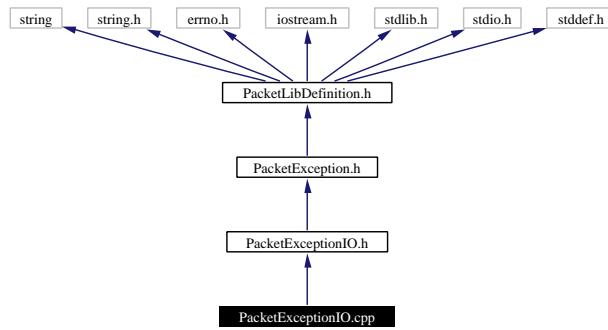
Namespaces

- namespace **PacketLib**

7.51 PacketExceptionIO.cpp File Reference

```
#include "PacketExceptionIO.h"
```

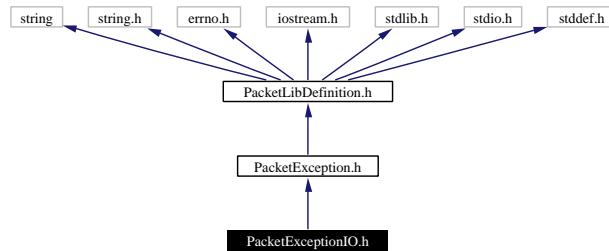
Include dependency graph for PacketExceptionIO.cpp:



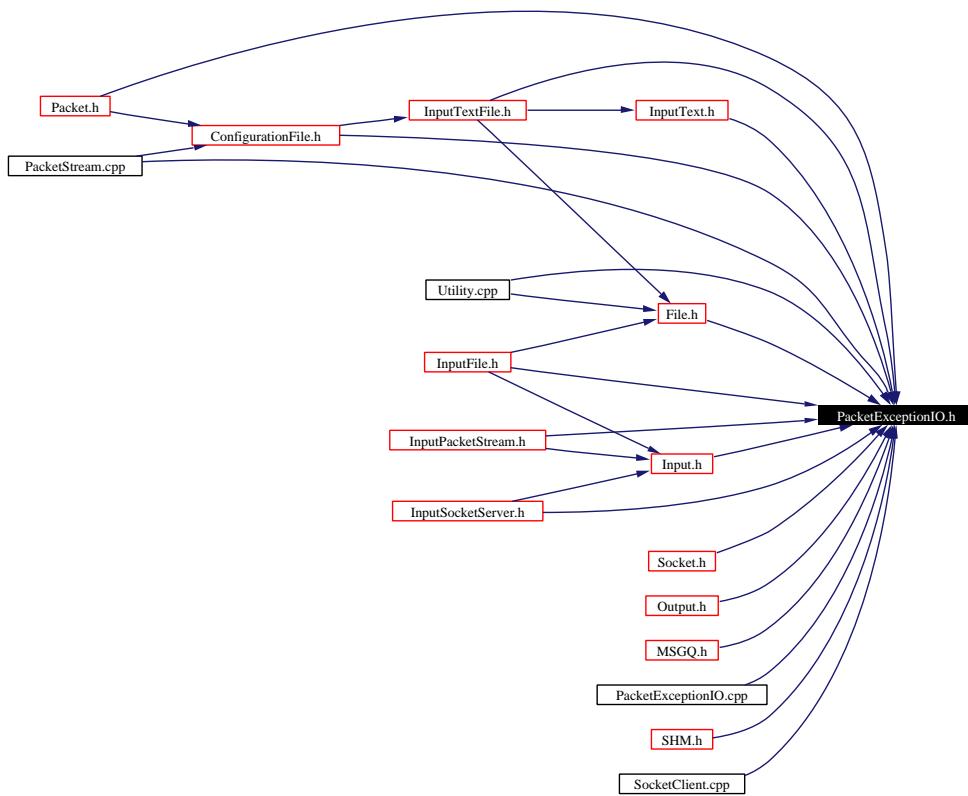
7.52 PacketExceptionIO.h File Reference

```
#include "PacketException.h"
```

Include dependency graph for PacketExceptionIO.h:



This graph shows which files directly or indirectly include this file:



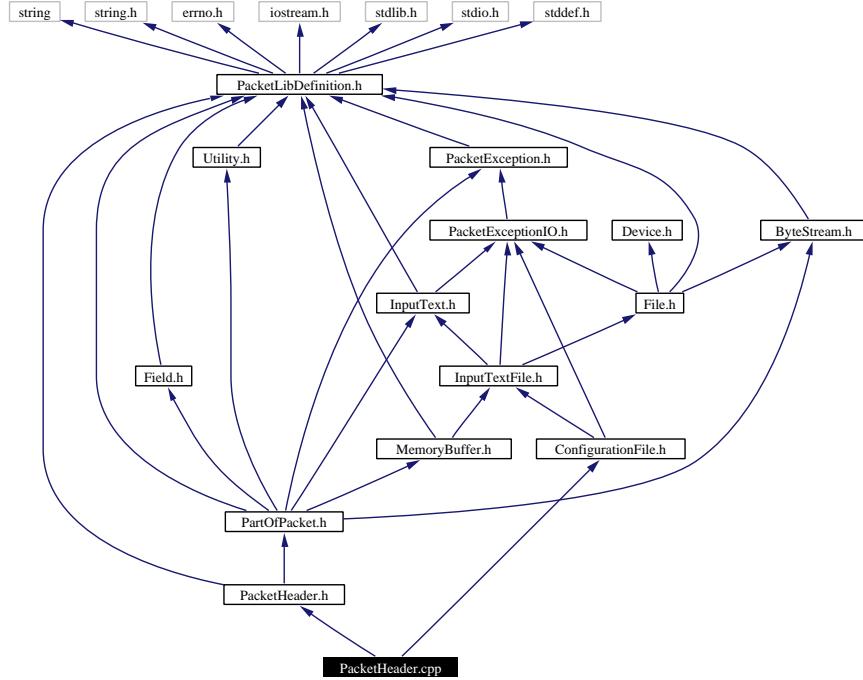
Namespaces

- namespace **PacketLib**

7.53 PacketHeader.cpp File Reference

```
#include "PacketHeader.h"  
#include "ConfigurationFile.h"
```

Include dependency graph for PacketHeader.cpp:

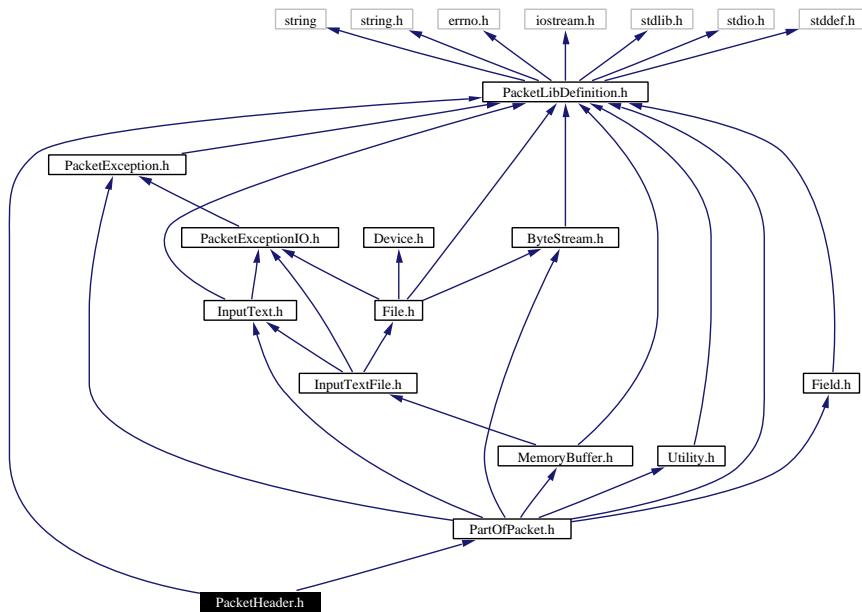


7.54 PacketHeader.h File Reference

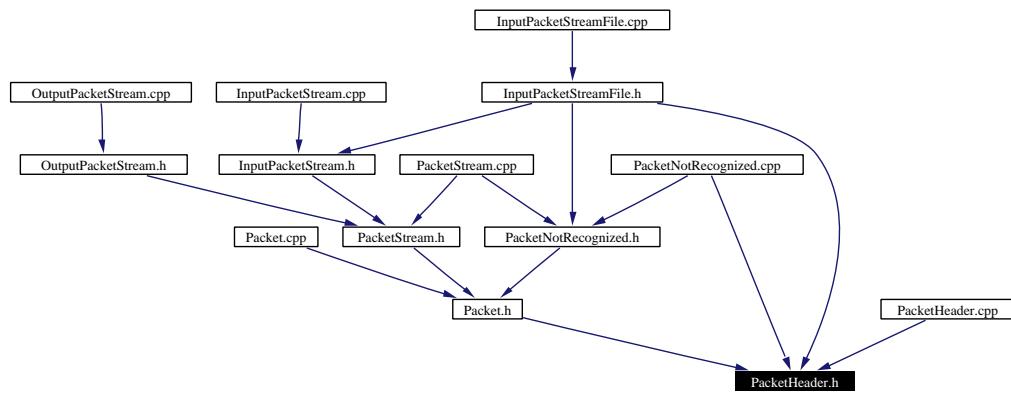
```
#include "PacketLibDefinition.h"
```

```
#include "PartOfPacket.h"
```

Include dependency graph for PacketHeader.h:



This graph shows which files directly or indirectly include this file:



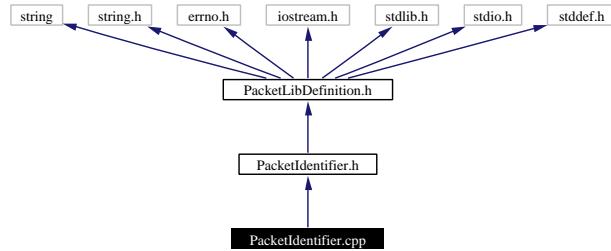
Namespaces

- namespace **PacketLib**

7.55 PacketIdentifier.cpp File Reference

```
#include "PacketIdentifier.h"
```

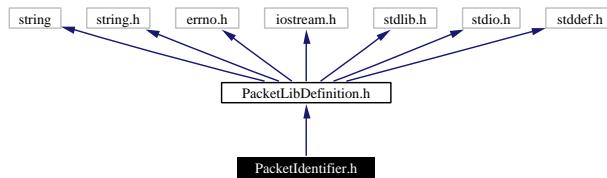
Include dependency graph for PacketIdentifier.cpp:



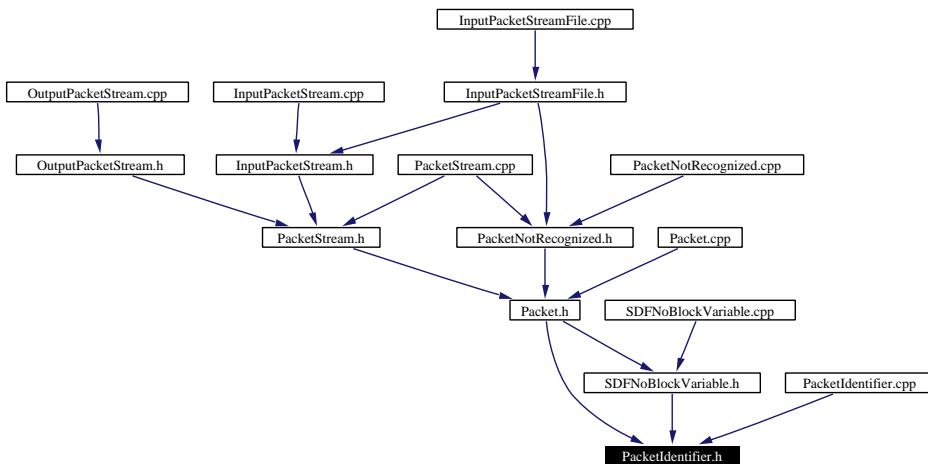
7.56 PacketIdentifier.h File Reference

```
#include "PacketLibDefinition.h"
```

Include dependency graph for PacketIdentifier.h:



This graph shows which files directly or indirectly include this file:



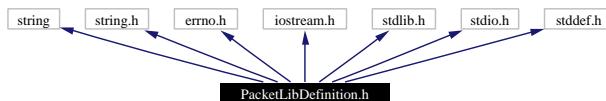
Namespaces

- namespace **PacketLib**

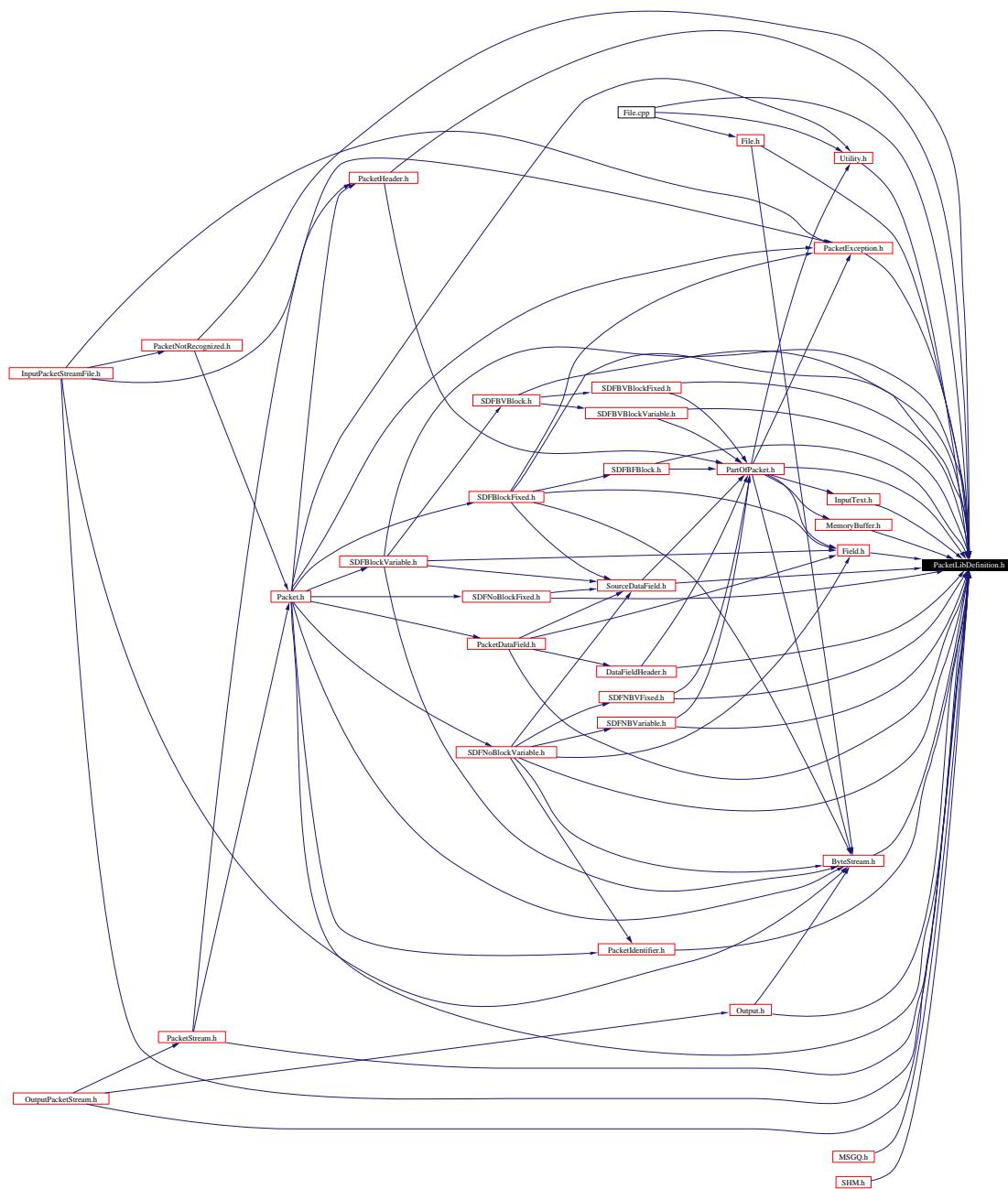
7.57 PacketLibDefinition.h File Reference

```
#include <string>
#include <string.h>
#include <errno.h>
#include <iostream.h>
#include <stdlib.h>
#include <stdio.h>
#include <stddef.h>
```

Include dependency graph for PacketLibDefinition.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace **PacketLib**
- namespace **std**

Defines

- #define **TRUE** 1
- #define **FALSE** 0
- #define **EOI** -1
- #define **DEBUGMODE** 0
- #define **ARCH_BIGENDIAN** 0
- #define **PRINTDEBUG**(strprint) if(DEBUGMODE) printf("%s\n", strprint);

7.57.1 Define Documentation

7.57.1.1 #define **ARCH_BIGENDIAN** 0

Definition at line 33 of file PacketLibDefinition.h.

Referenced by PacketLib::ByteStream::setWord.

7.57.1.2 #define **DEBUGMODE** 0

Definition at line 30 of file PacketLibDefinition.h.

7.57.1.3 #define **EOI** -1

Definition at line 29 of file PacketLibDefinition.h.

Referenced by PacketLib::File::getByte, and PacketLib::File::getNByte.

7.57.1.4 #define **FALSE** 0

Definition at line 28 of file PacketLibDefinition.h.

7.57.1.5 #define **PRINTDEBUG**(strprint) if(DEBUGMODE) printf("%s\n", strprint);

Definition at line 36 of file PacketLibDefinition.h.

Referenced by PacketLib::PacketNotRecognized::createPacketType, PacketLib::Packet::createPacketType, PacketLib::SDFBlockFixed::loadFields, and PacketLib::Packet::printPacketValue.

7.57.1.6 #define **TRUE** 1

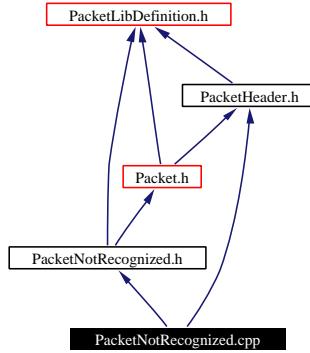
Definition at line 27 of file PacketLibDefinition.h.

7.58 PacketNotRecognized.cpp File Reference

```
#include "PacketNotRecognized.h"
```

```
#include "PacketHeader.h"
```

Include dependency graph for PacketNotRecognized.cpp:

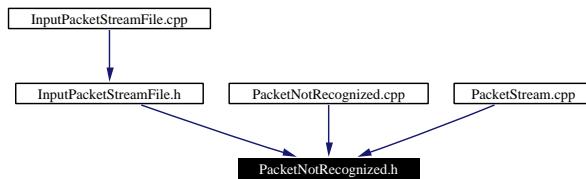


7.59 PacketNotRecognized.h File Reference

```
#include "Packet.h"  
#include "PacketLibDefinition.h"  
Include dependency graph for PacketNotRecognized.h:
```



This graph shows which files directly or indirectly include this file:



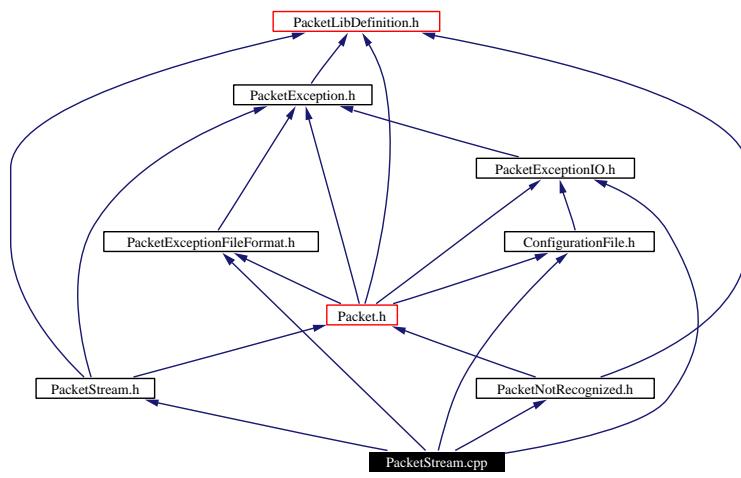
Namespaces

- namespace **PacketLib**

7.60 PacketStream.cpp File Reference

```
#include "PacketStream.h"
#include "ConfigurationFile.h"
#include "PacketExceptionFormat.h"
#include "PacketExceptionIO.h"
#include "PacketNotRecognized.h"
```

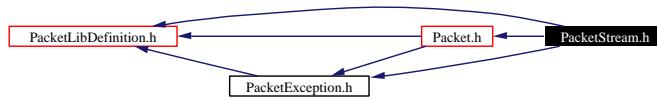
Include dependency graph for PacketStream.cpp:



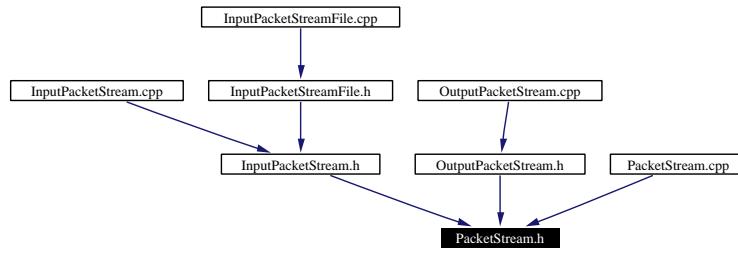
7.61 PacketStream.h File Reference

```
#include "PacketLibDefinition.h"
#include "Packet.h"
#include "PacketException.h"
```

Include dependency graph for PacketStream.h:



This graph shows which files directly or indirectly include this file:



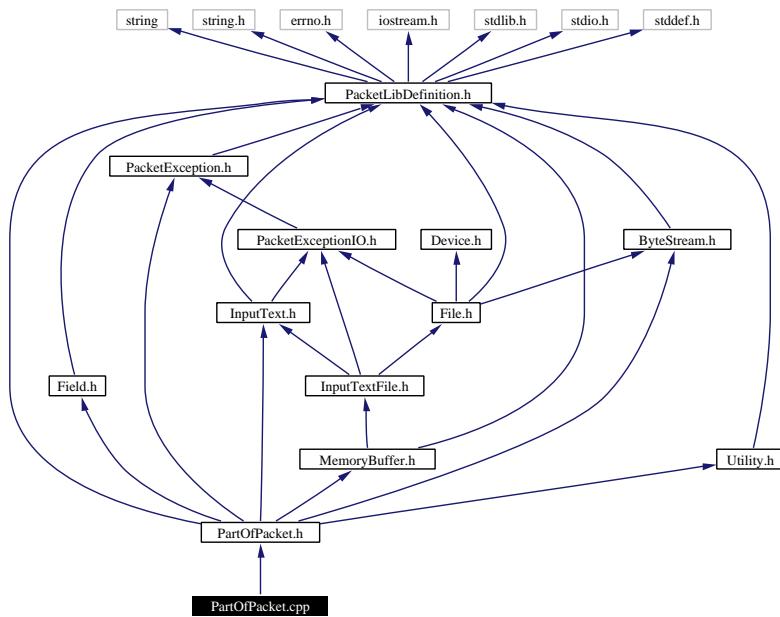
Namespaces

- namespace **PacketLib**

7.62 PartOfPacket.cpp File Reference

```
#include "PartOfPacket.h"
```

Include dependency graph for PartOfPacket.cpp:



Variables

- word **pattern** [] = {0,1,3,7,15,31,63,127,255,511,1023,2047,4095,8191,16383,32767,65535}

7.62.1 Variable Documentation

7.62.1.1 word pattern[] = {0,1,3,7,15,31,63,127,255,511,1023,2047,4095,8191,16383,32767,65535}

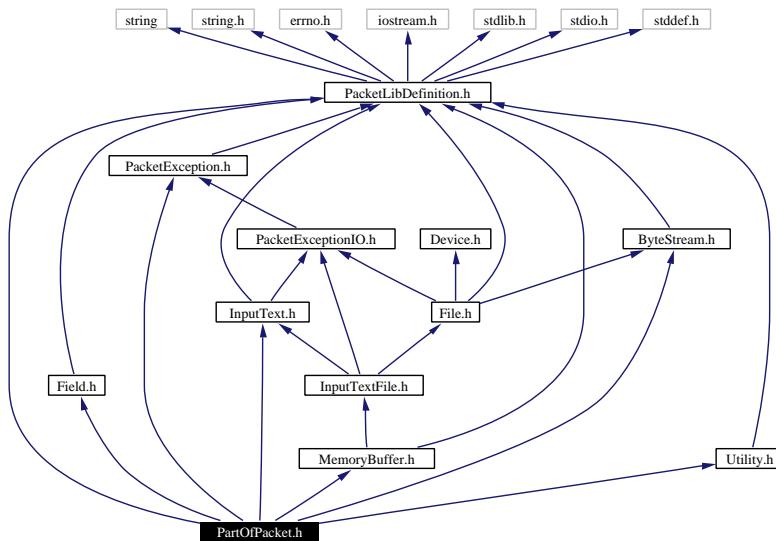
Definition at line 22 of file PartOfPacket.cpp.

Referenced by PacketLib::PartOfPacket::setByteStream.

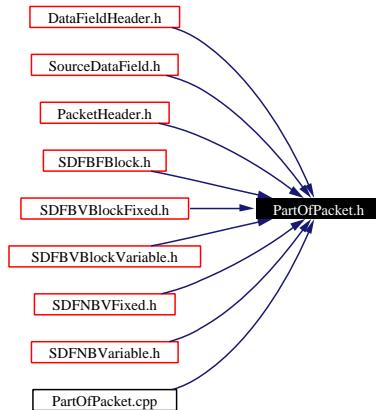
7.63 PartOfPacket.h File Reference

```
#include "InputText.h"
#include "Field.h"
#include "PacketLibDefinition.h"
#include "Utility.h"
#include "ByteStream.h"
#include "PacketException.h"
#include "MemoryBuffer.h"
```

Include dependency graph for PartOfPacket.h:



This graph shows which files directly or indirectly include this file:



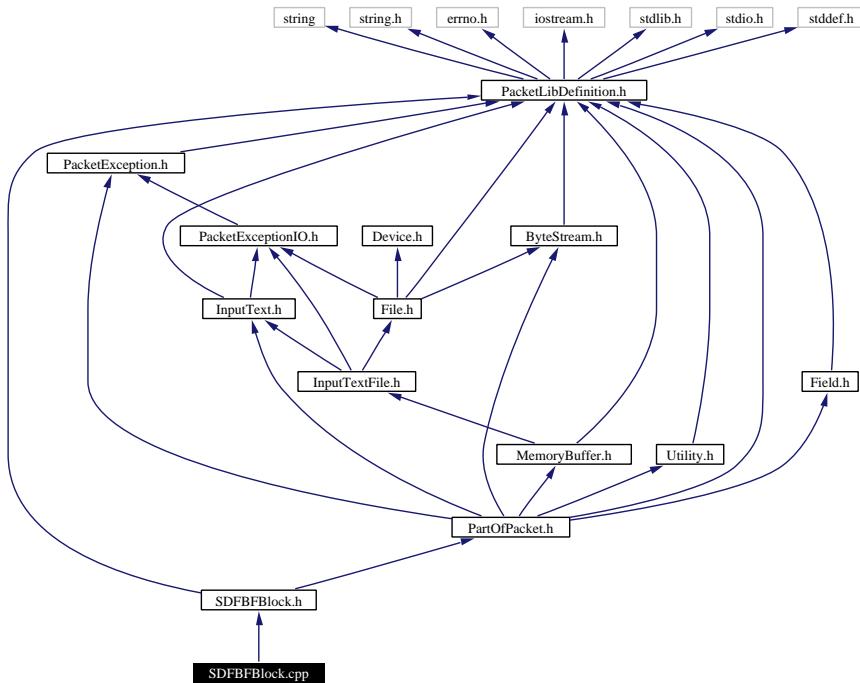
Namespaces

- namespace **PacketLib**

7.64 SDFBFBlock.cpp File Reference

```
#include "SDFBFBlock.h"
```

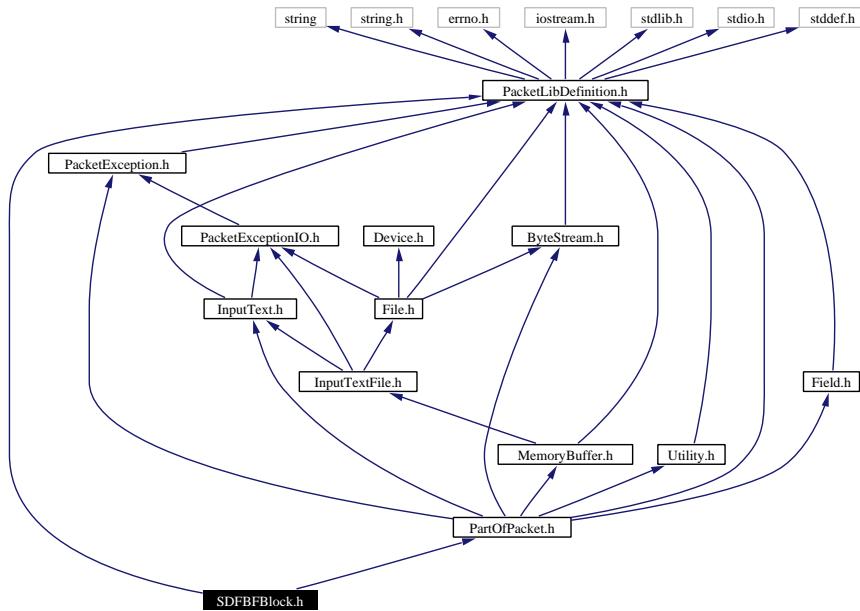
Include dependency graph for SDFBFBlock.cpp:



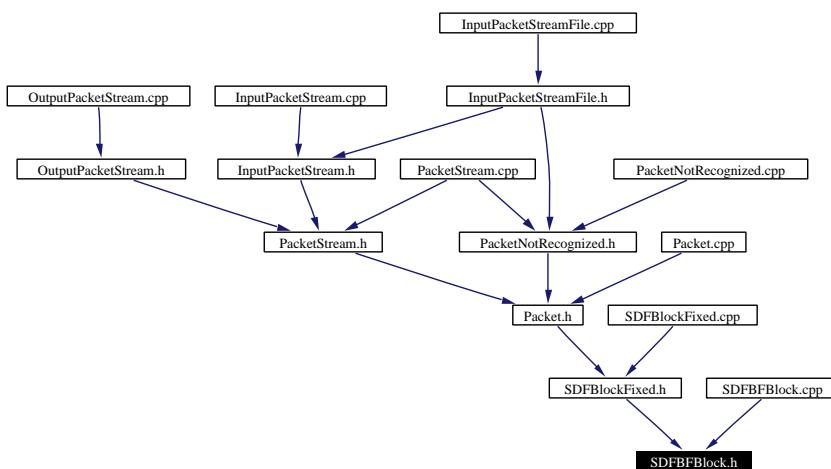
7.65 SDFBFBlock.h File Reference

```
#include "PacketLibDefinition.h"
#include "PartOfPacket.h"
```

Include dependency graph for SDFBFBlock.h:



This graph shows which files directly or indirectly include this file:



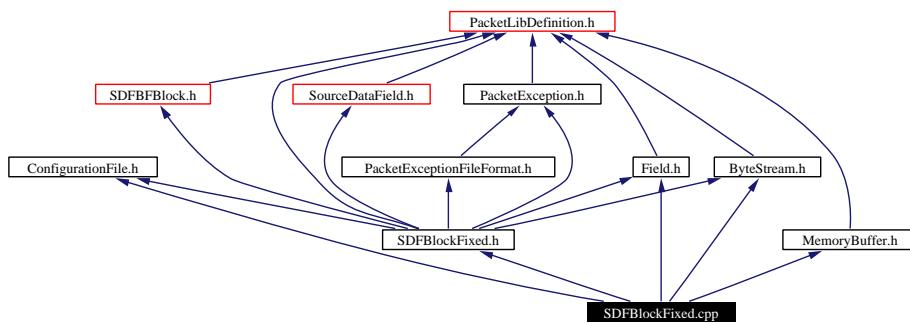
Namespaces

- namespace **PacketLib**

7.66 SDFBlockFixed.cpp File Reference

```
#include "SDFBlockFixed.h"
#include "ByteStream.h"
#include "Field.h"
#include "ConfigurationFile.h"
#include "MemoryBuffer.h"
```

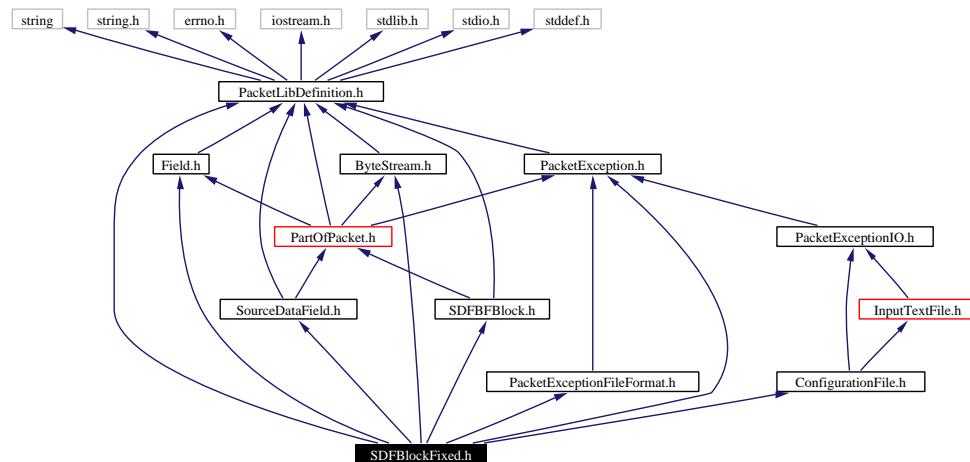
Include dependency graph for SDFBlockFixed.cpp:



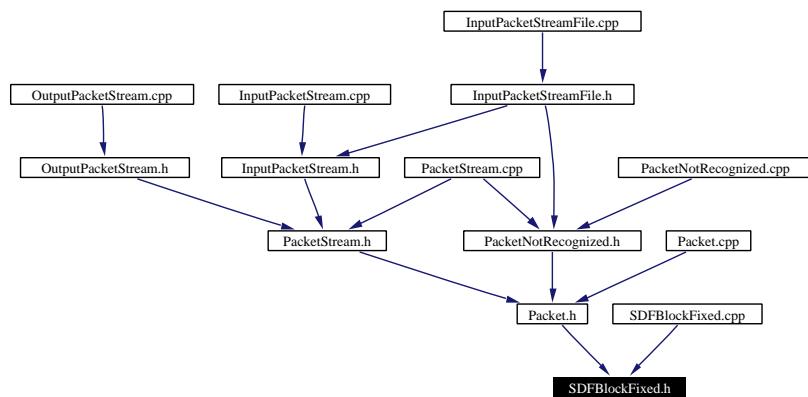
7.67 SDFBlockFixed.h File Reference

```
#include "PacketLibDefinition.h"
#include "SourceDataField.h"
#include "SDFBFBlock.h"
#include "ConfigurationFile.h"
#include "Field.h"
#include "ByteStream.h"
#include "PacketException.h"
#include "PacketExceptionFileFormat.h"

Include dependency graph for SDFBlockFixed.h:
```



This graph shows which files directly or indirectly include this file:



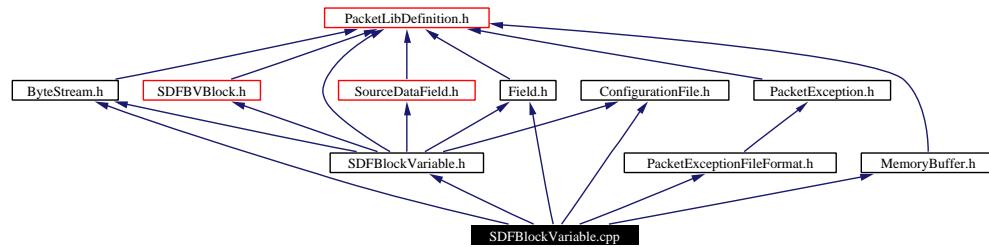
Namespaces

- namespace **PacketLib**

7.68 SDFBlockVariable.cpp File Reference

```
#include "SDFBlockVariable.h"
#include "ByteStream.h"
#include "Field.h"
#include "ConfigurationFile.h"
#include "PacketExceptionFileFormat.h"
#include "MemoryBuffer.h"
```

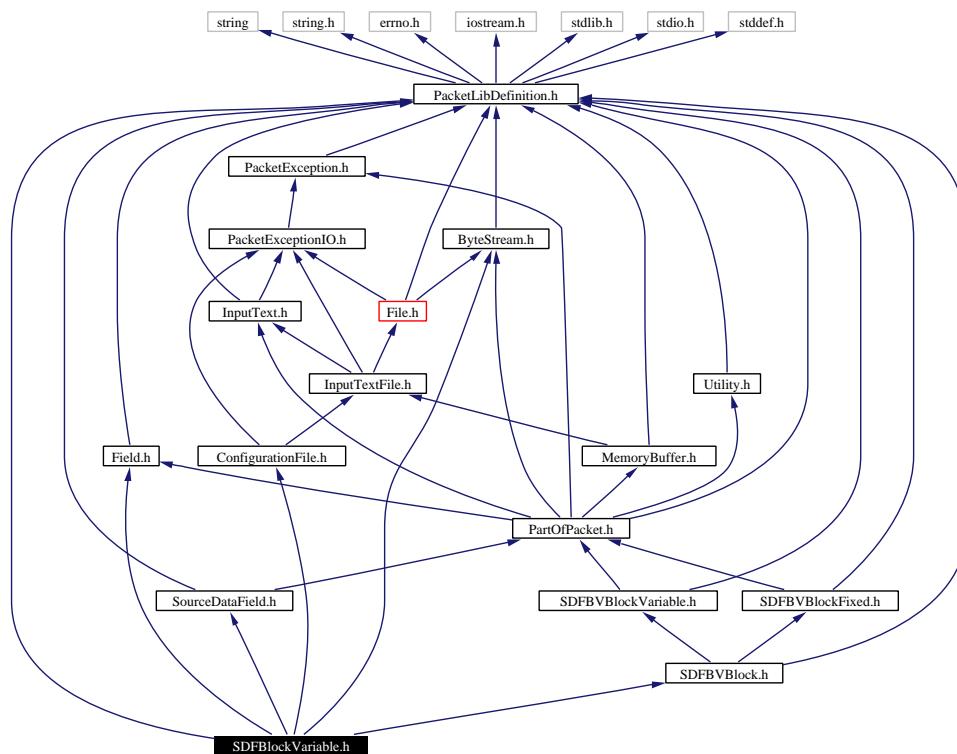
Include dependency graph for SDFBlockVariable.cpp:



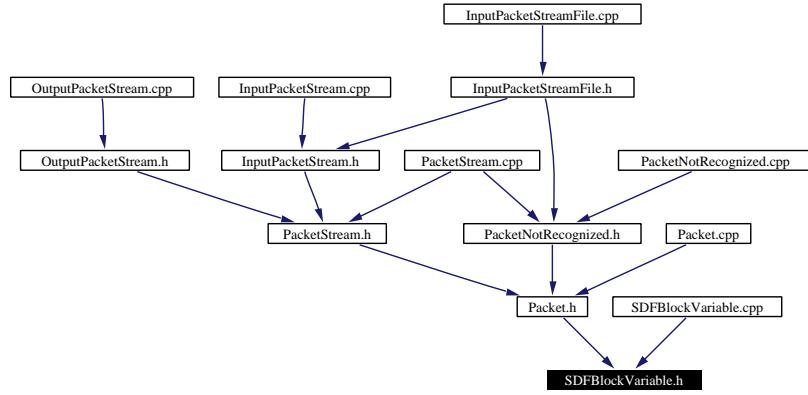
7.69 SDFBlockVariable.h File Reference

```
#include "PacketLibDefinition.h"
#include "SourceDataField.h"
#include "SDFBVBlock.h"
#include "ConfigurationFile.h"
#include "Field.h"
#include "ByteStream.h"
```

Include dependency graph for SDFBlockVariable.h:



This graph shows which files directly or indirectly include this file:



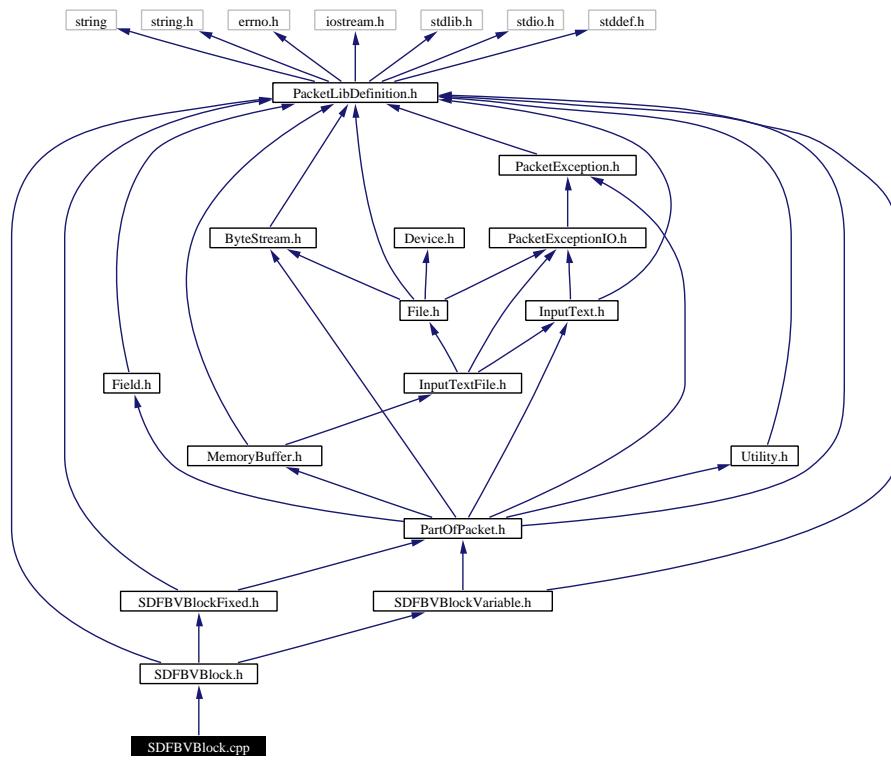
Namespaces

- namespace **PacketLib**

7.70 SDFBVBlock.cpp File Reference

```
#include "SDFBVBlock.h"
```

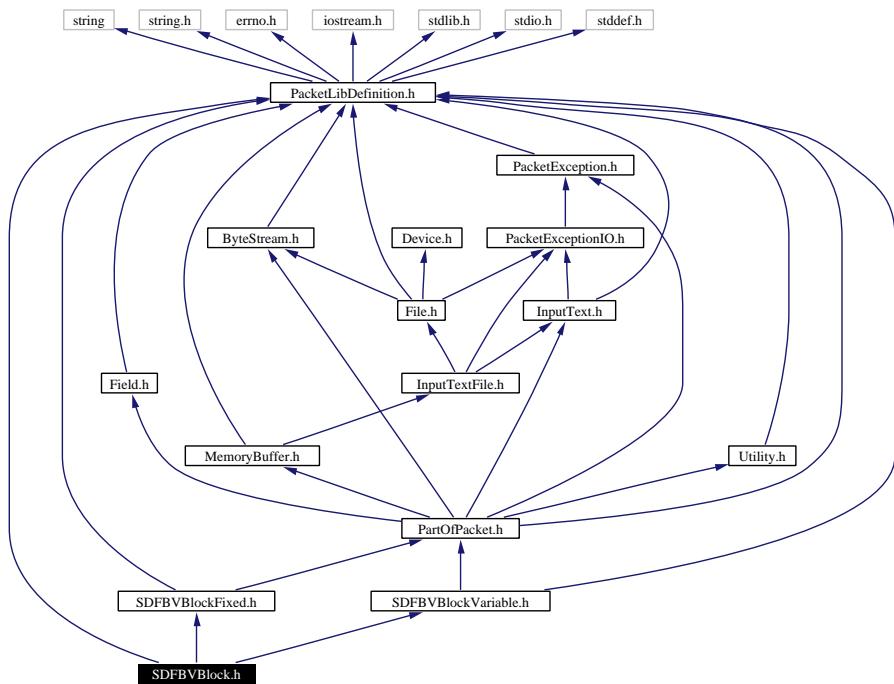
Include dependency graph for SDFBVBlock.cpp:



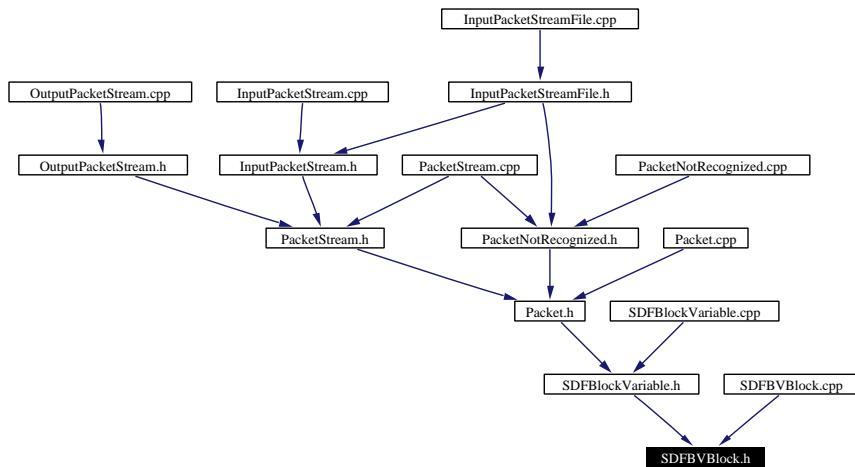
7.71 SDFBVBlock.h File Reference

```
#include "PacketLibDefinition.h"
#include "SDFBVBlockFixed.h"
#include "SDFBVBlockVariable.h"

Include dependency graph for SDFBVBlock.h:
```



This graph shows which files directly or indirectly include this file:



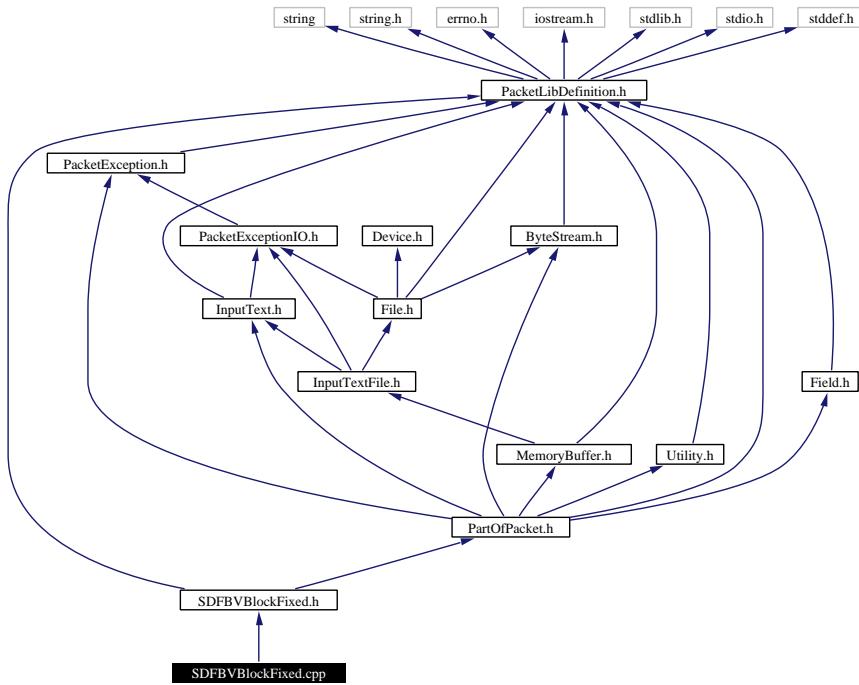
Namespaces

- namespace **PacketLib**

7.72 SDFBVBlockFixed.cpp File Reference

```
#include "SDFBVBlockFixed.h"
```

Include dependency graph for SDFBVBlockFixed.cpp:

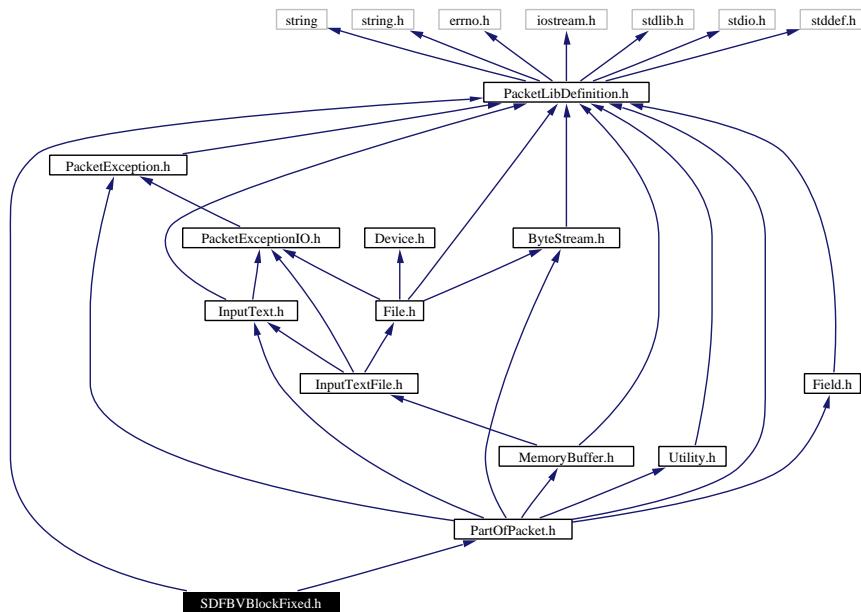


7.73 SDFBVBlockFixed.h File Reference

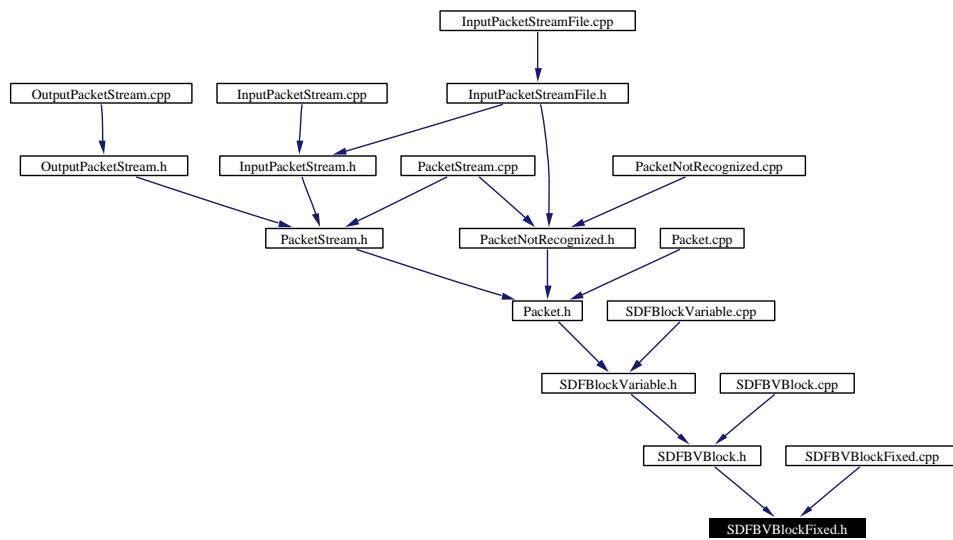
```
#include "PacketLibDefinition.h"
```

```
#include "PartOfPacket.h"
```

Include dependency graph for SDFBVBlockFixed.h:



This graph shows which files directly or indirectly include this file:



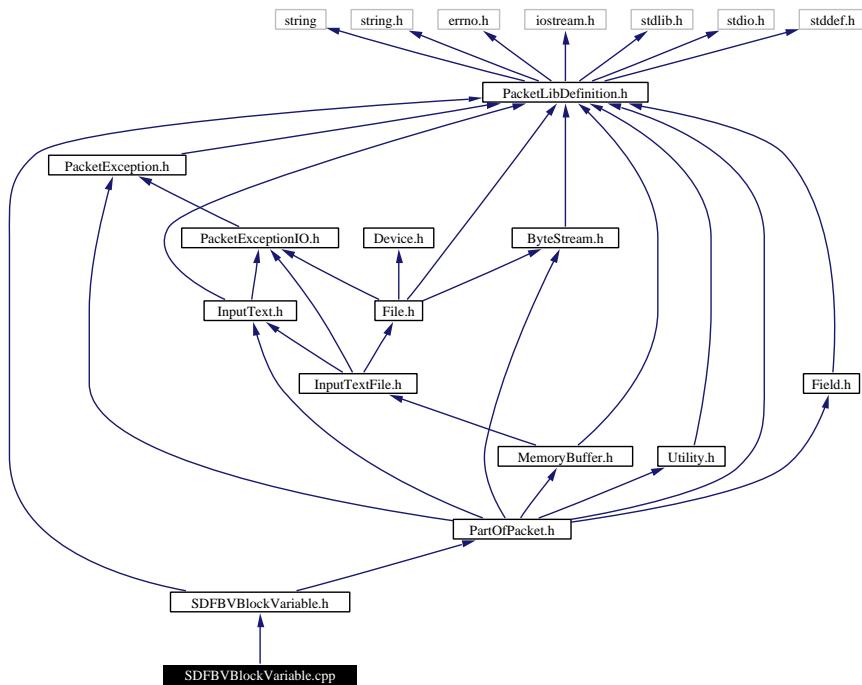
Namespaces

- namespace **PacketLib**

7.74 SDFBVBlockVariable.cpp File Reference

```
#include "SDFBVBlockVariable.h"
```

Include dependency graph for SDFBVBlockVariable.cpp:

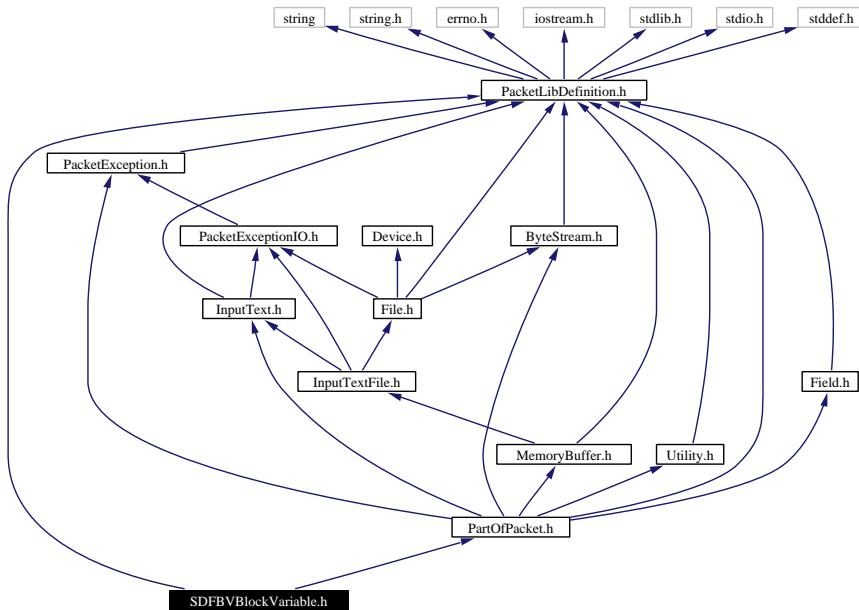


7.75 SDFBVBlockVariable.h File Reference

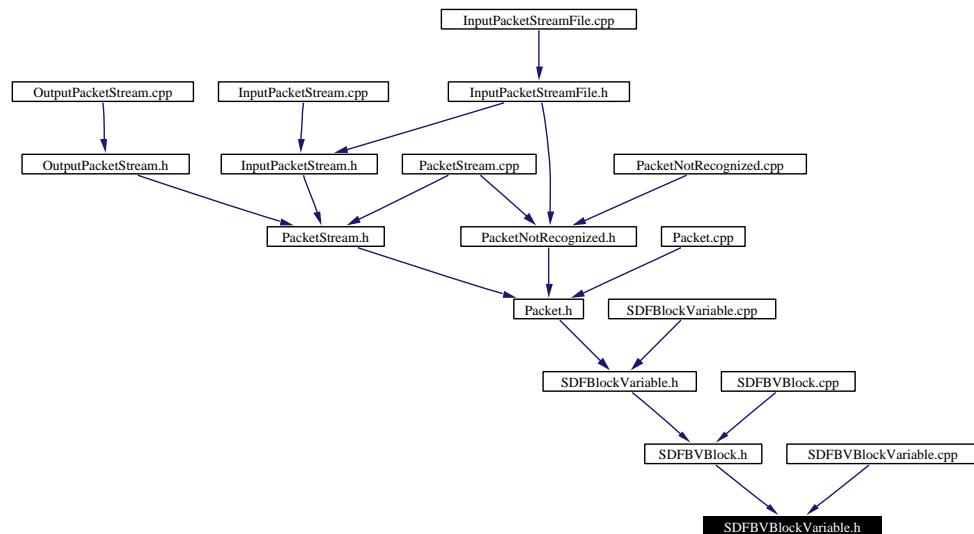
```
#include "PacketLibDefinition.h"
```

```
#include "PartOfPacket.h"
```

Include dependency graph for SDFBVBlockVariable.h:



This graph shows which files directly or indirectly include this file:



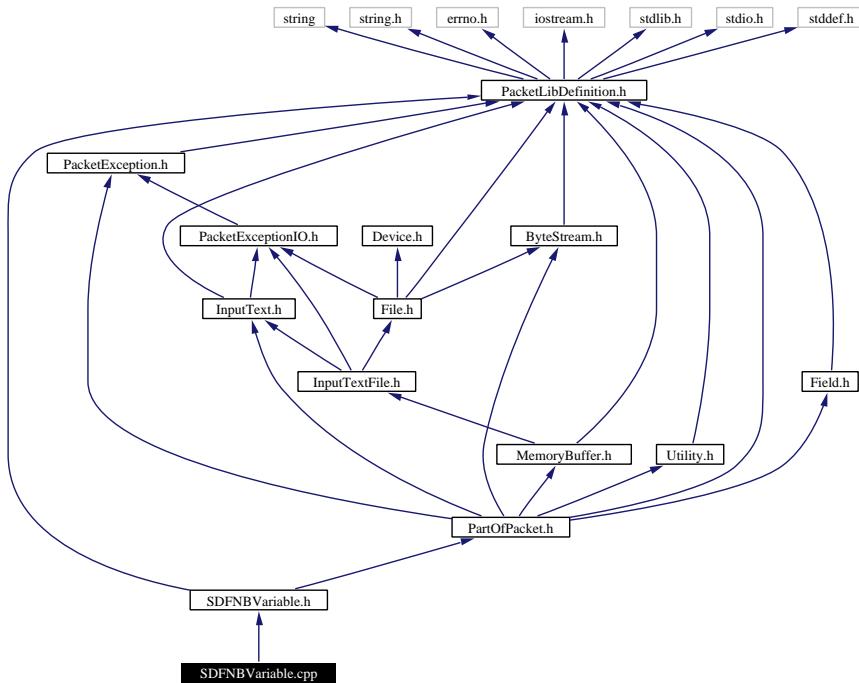
Namespaces

- namespace **PacketLib**

7.76 SDFNBVariable.cpp File Reference

```
#include "SDFNBVariable.h"
```

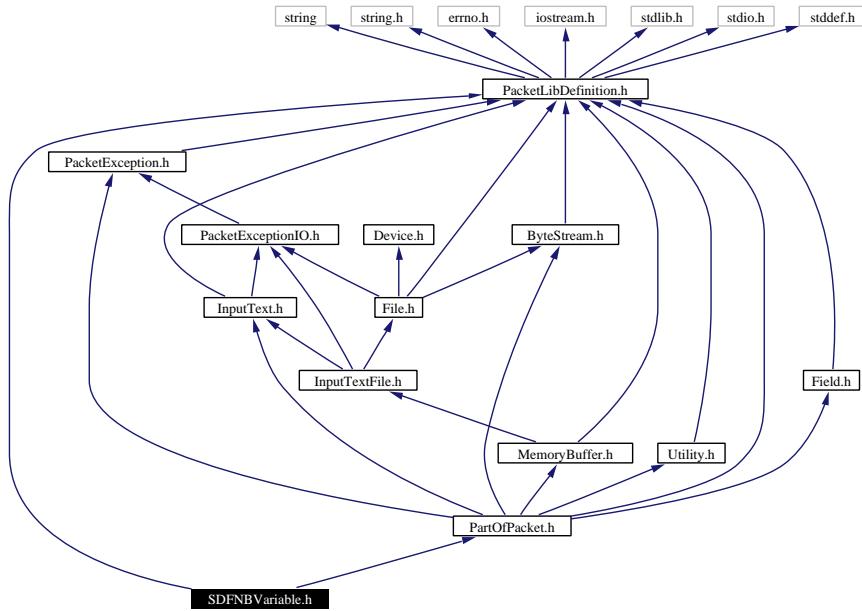
Include dependency graph for SDFNBVariable.cpp:



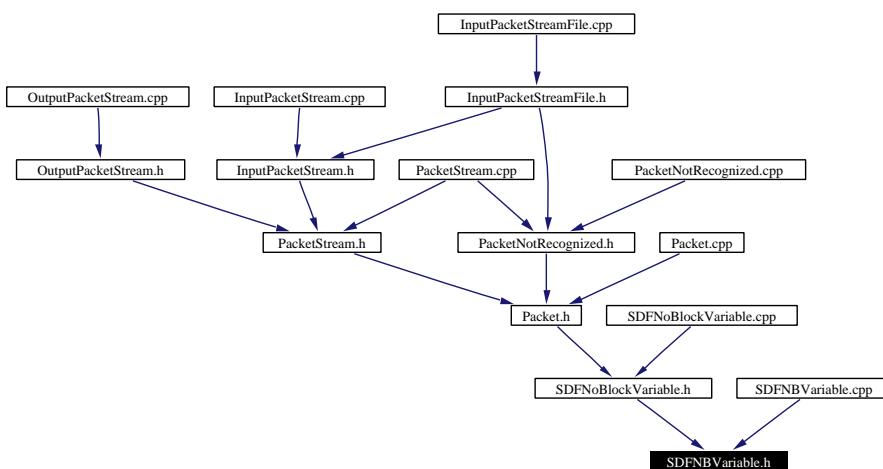
7.77 SDFNBVariable.h File Reference

```
#include "PacketLibDefinition.h"
#include "PartOfPacket.h"
```

Include dependency graph for SDFNBVariable.h:



This graph shows which files directly or indirectly include this file:



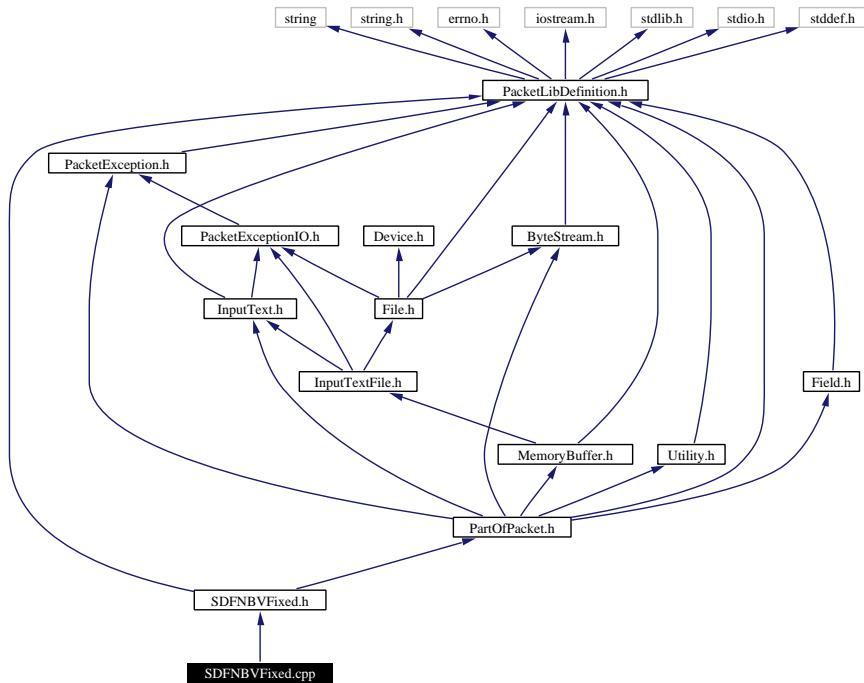
Namespaces

- namespace **PacketLib**

7.78 SDFNBVFixed.cpp File Reference

```
#include "SDFNBVFixed.h"
```

Include dependency graph for SDFNBVFixed.cpp:

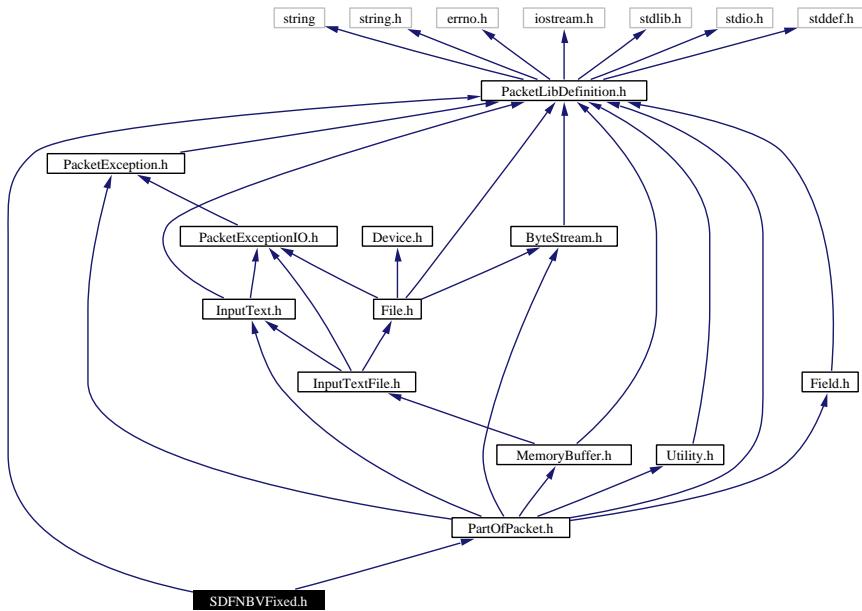


7.79 SDFNBVFixed.h File Reference

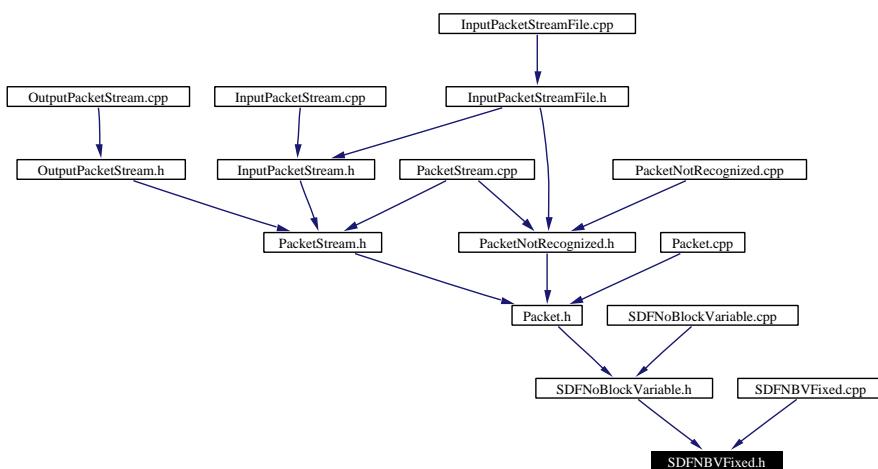
```
#include "PacketLibDefinition.h"
```

```
#include "PartOfPacket.h"
```

Include dependency graph for SDFNBVFixed.h:



This graph shows which files directly or indirectly include this file:



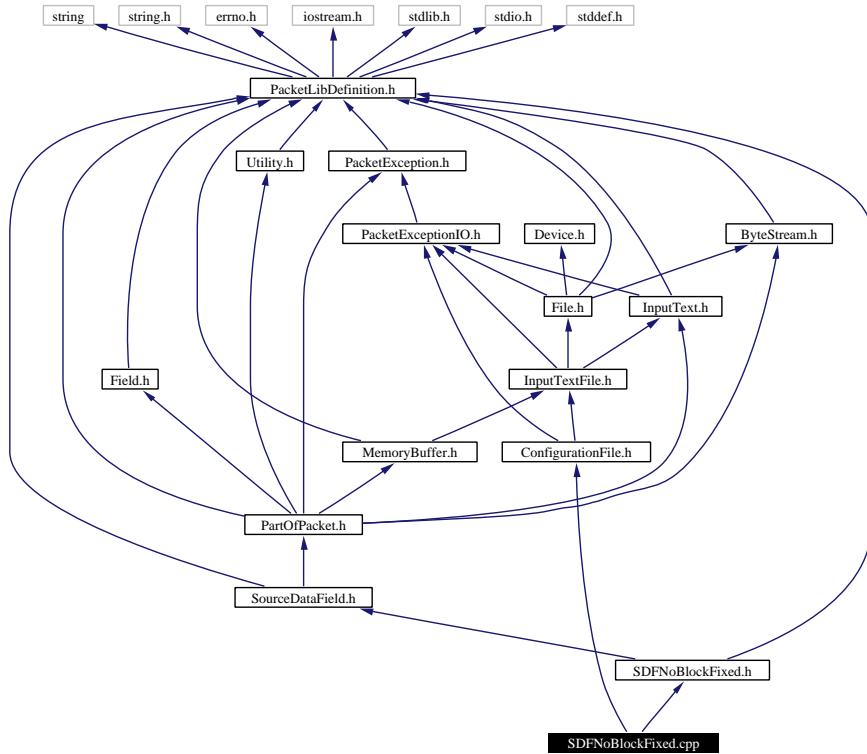
Namespaces

- namespace **PacketLib**

7.80 SDFNoBlockFixed.cpp File Reference

```
#include "SDFNoBlockFixed.h"
#include "ConfigurationFile.h"
```

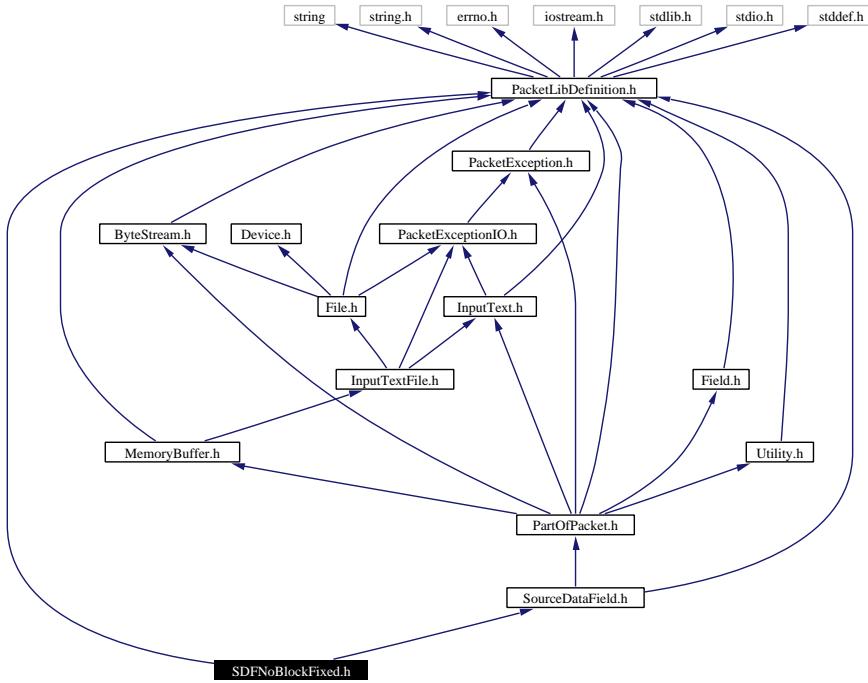
Include dependency graph for SDFNoBlockFixed.cpp:



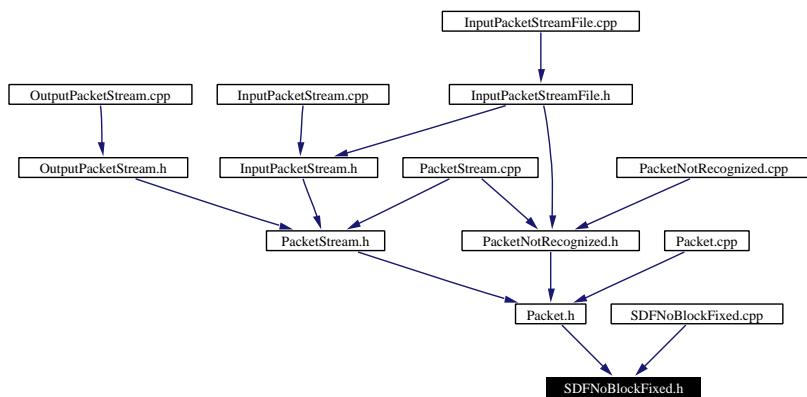
7.81 SDFNoBlockFixed.h File Reference

```
#include "PacketLibDefinition.h"
#include "SourceDataField.h"
```

Include dependency graph for SDFNoBlockFixed.h:



This graph shows which files directly or indirectly include this file:



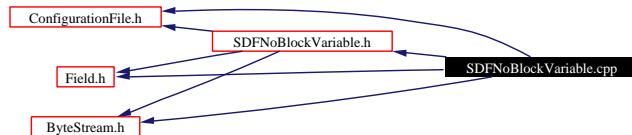
Namespaces

- namespace **PacketLib**

7.82 SDFNoBlockVariable.cpp File Reference

```
#include "SDFNoBlockVariable.h"  
#include "ByteStream.h"  
#include "Field.h"  
#include "ConfigurationFile.h"
```

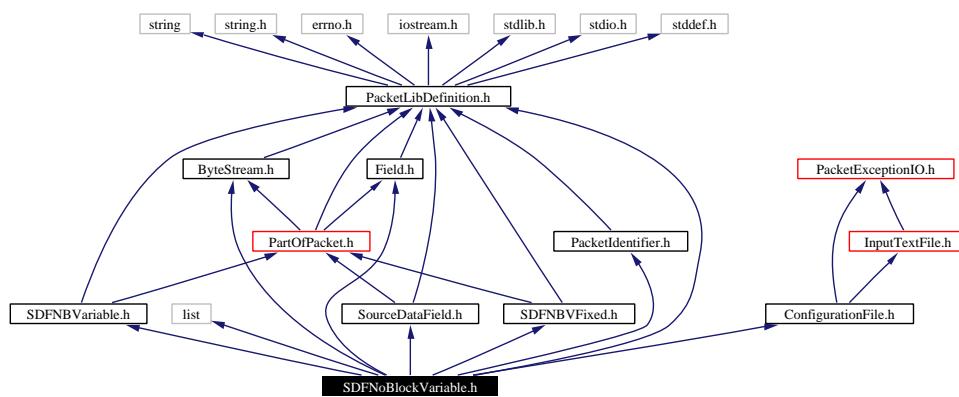
Include dependency graph for SDFNoBlockVariable.cpp:



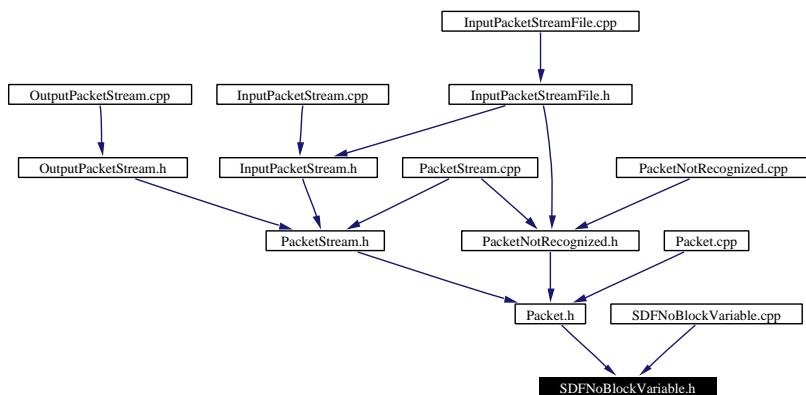
7.83 SDFNoBlockVariable.h File Reference

```
#include "PacketLibDefinition.h"
#include <list>
#include "SDFNBFixed.h"
#include "SDFNBVariable.h"
#include "SourceDataField.h"
#include "PacketIdentifier.h"
#include "ConfigurationFile.h"
#include "Field.h"
#include "ByteStream.h"

Include dependency graph for SDFNoBlockVariable.h:
```



This graph shows which files directly or indirectly include this file:



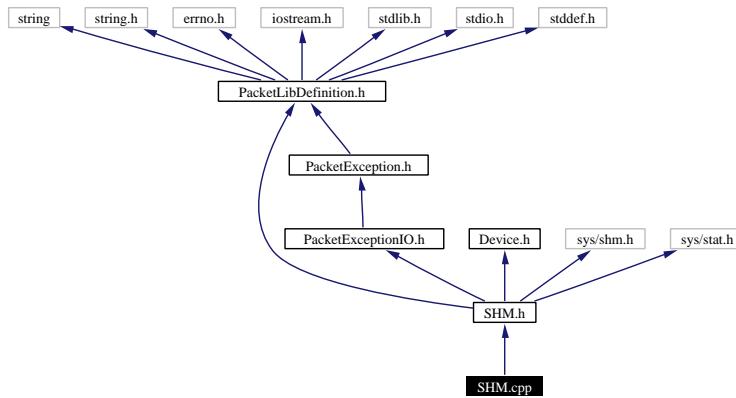
Namespaces

- namespace **PacketLib**

7.84 SHM.cpp File Reference

```
#include "SHM.h"
```

Include dependency graph for SHM.cpp:



Defines

- #define **PERM** 0777

7.84.1 Define Documentation

7.84.1.1 #define PERM 0777

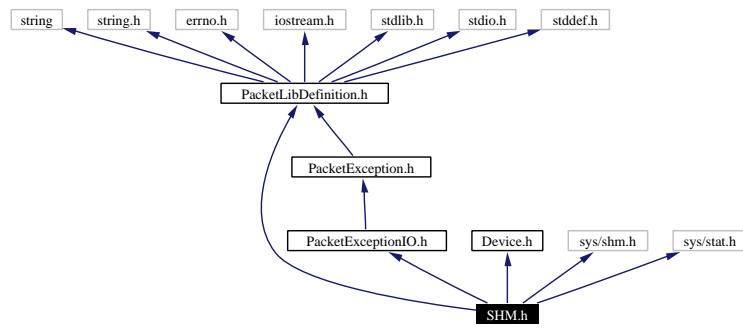
Definition at line 19 of file SHM.cpp.

Referenced by PacketLib::SHM::create.

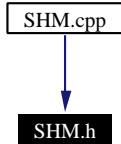
7.85 SHM.h File Reference

```
#include "PacketLibDefinition.h"
#include "Device.h"
#include "PacketExceptionIO.h"
#include <sys/shm.h>
#include <sys/stat.h>
```

Include dependency graph for SHM.h:



This graph shows which files directly or indirectly include this file:



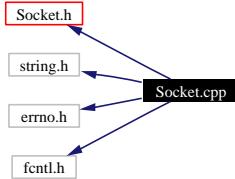
Namespaces

- namespace **PacketLib**

7.86 Socket.cpp File Reference

```
#include "Socket.h"  
#include "string.h"  
#include <errno.h>  
#include <fcntl.h>
```

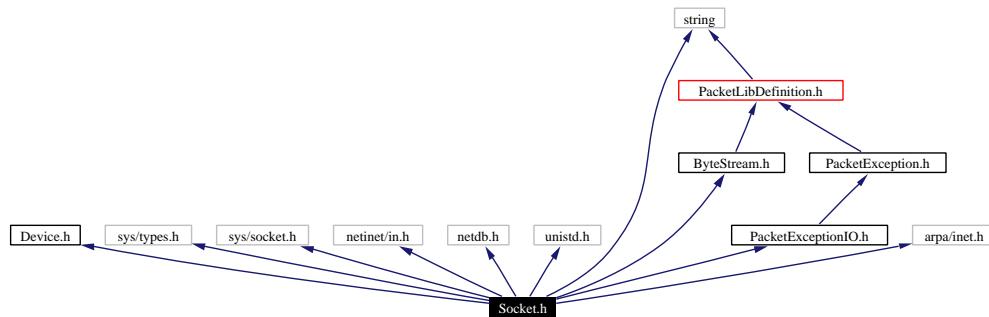
Include dependency graph for Socket.cpp:



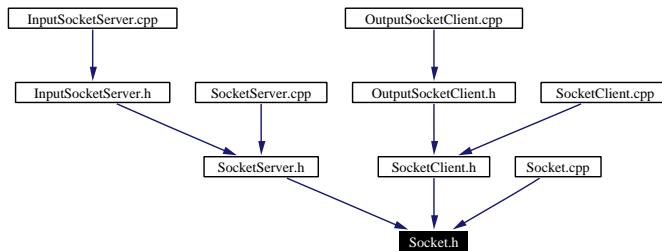
7.87 Socket.h File Reference

```
#include "Device.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <unistd.h>
#include <string>
#include <arpa/inet.h>
#include "ByteStream.h"
#include "PacketExceptionIO.h"
```

Include dependency graph for Socket.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace **PacketLib**

Variables

- const int **MAXHOSTNAME** = 200
- const int **MAXCONNECTIONS** = 5
- const int **MAXRECV** = 500

7.87.1 Variable Documentation

7.87.1.1 const int **MAXCONNECTIONS** = 5

Definition at line 33 of file Socket.h.

Referenced by PacketLib::SocketServer::listen.

7.87.1.2 const int **MAXHOSTNAME** = 200

Definition at line 32 of file Socket.h.

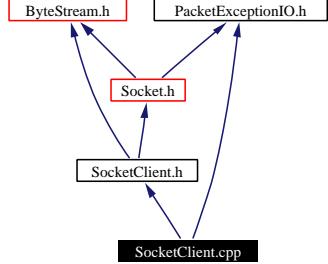
7.87.1.3 const int **MAXRECV** = 500

Definition at line 34 of file Socket.h.

7.88 SocketClient.cpp File Reference

```
#include "SocketClient.h"  
#include "PacketExceptionIO.h"
```

Include dependency graph for SocketClient.cpp:

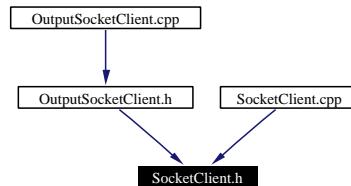


7.89 SocketClient.h File Reference

```
#include "Socket.h"  
#include "ByteStream.h"  
Include dependency graph for SocketClient.h:
```



This graph shows which files directly or indirectly include this file:



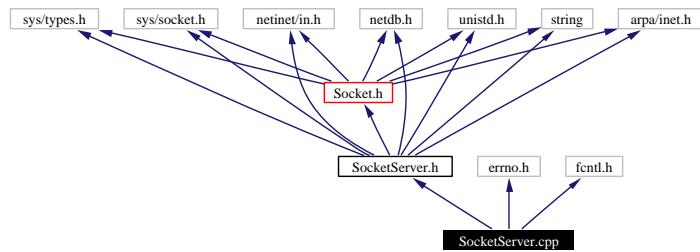
Namespaces

- namespace **PacketLib**

7.90 SocketServer.cpp File Reference

```
#include "SocketServer.h"  
#include <errno.h>  
#include <fcntl.h>
```

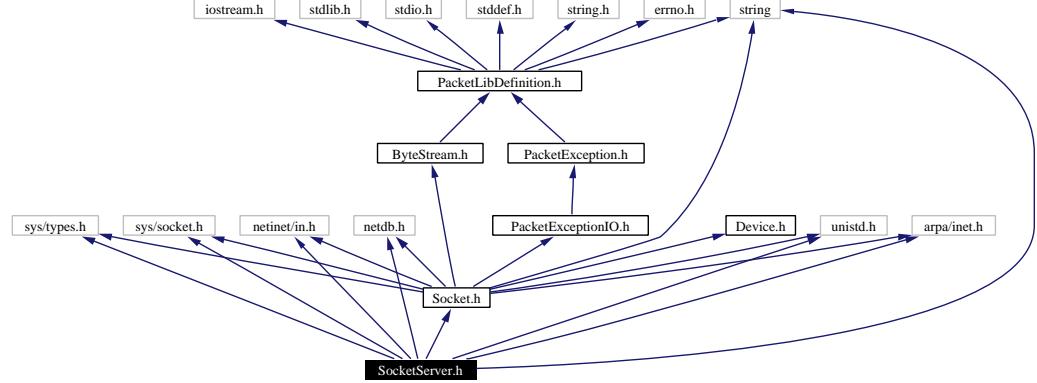
Include dependency graph for SocketServer.cpp:



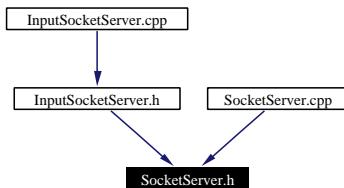
7.91 SocketServer.h File Reference

```
#include "Socket.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <unistd.h>
#include <string>
#include <arpa/inet.h>
```

Include dependency graph for SocketServer.h:



This graph shows which files directly or indirectly include this file:



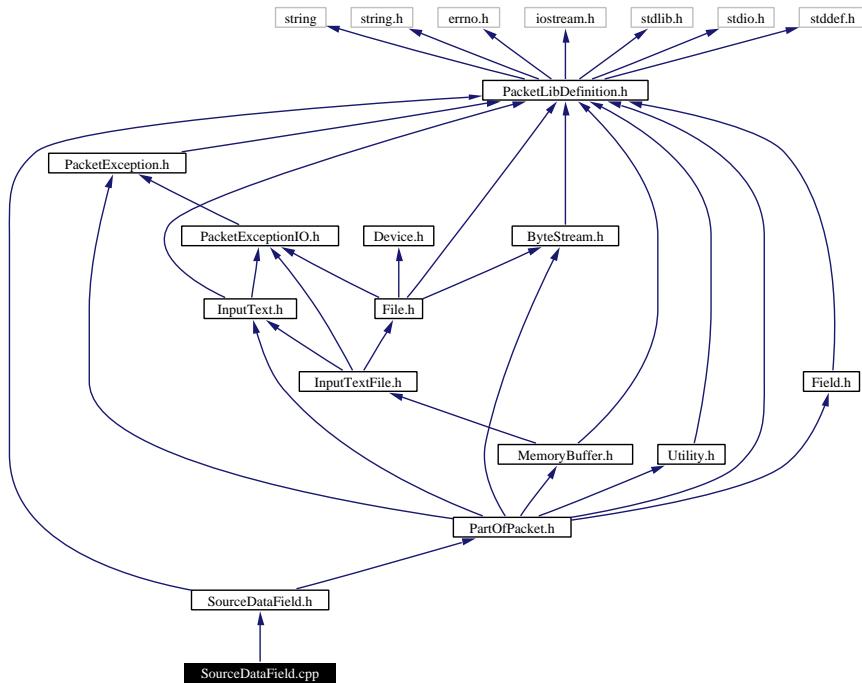
Namespaces

- namespace **PacketLib**

7.92 SourceDataField.cpp File Reference

```
#include "SourceDataField.h"
```

Include dependency graph for SourceDataField.cpp:

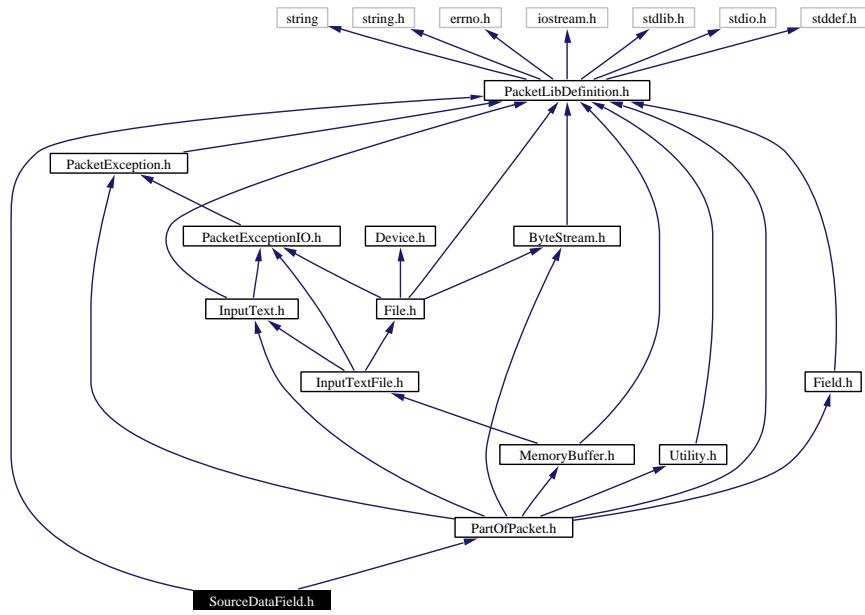


7.93 SourceDataField.h File Reference

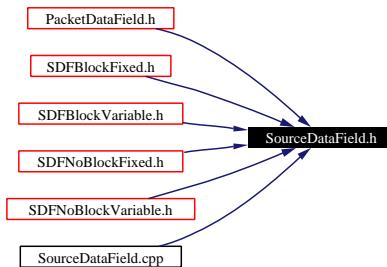
```
#include "PacketLibDefinition.h"
```

```
#include "PartOfPacket.h"
```

Include dependency graph for SourceDataField.h:



This graph shows which files directly or indirectly include this file:



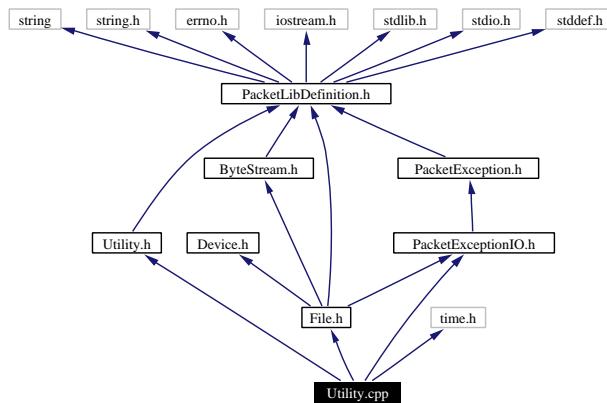
Namespaces

- namespace **PacketLib**

7.94 Utility.cpp File Reference

```
#include "Utility.h"
#include "File.h"
#include "PacketExceptionIO.h"
#include <time.h>
```

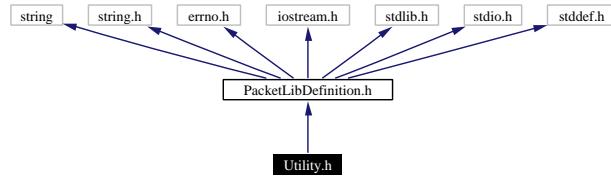
Include dependency graph for Utility.cpp:



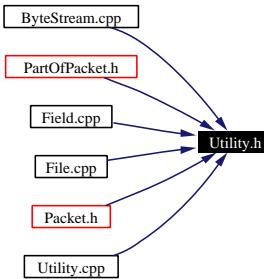
7.95 Utility.h File Reference

```
#include "PacketLibDefinition.h"
```

Include dependency graph for Utility.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- namespace **PacketLib**

Index

~ByteStream
 PacketLib::ByteStream, 23

~ConfigurationFile
 PacketLib::ConfigurationFile,
 31

~Field
 PacketLib::Field, 38

~FileStreamPointer
 PacketLib::FileStreamPointer,
 50

~Input
 PacketLib::Input, 53

~InputFile
 PacketLib::InputFile, 56

~InputPacketStream
 PacketLib::InputPacket-
 Stream, 59

~InputPacketStreamFile
 PacketLib::InputPacket-
 StreamFile, 64

~InputSocketServer
 PacketLib::InputSocketServer,
 69

~InputText
 PacketLib::InputText, 73

~InputTextFile
 PacketLib::InputTextFile, 76

~MSGQ
 PacketLib::MSGQ, 85

~MemoryBuffer
 PacketLib::MemoryBuffer, 80

~OutputFile
 PacketLib::OutputFile, 92

~OutputSocketClient
 PacketLib::OutputSocket-
 Client, 98

~Packet
 PacketLib::Packet, 102

~PacketDataField
 PacketLib::PacketDataField,

~PacketException
 PacketLib::PacketException,
 118

~PacketExceptionFormatException
 PacketLib::PacketException-
 FormatException, 121

~PacketExceptionIO
 PacketLib::PacketException-
 IO, 123

~PacketHeader
 PacketLib::PacketHeader, 125

~PacketIdentifier
 PacketLib::PacketIdentifier,
 127

~PacketNotRecognized
 PacketLib::PacketNot-
 Recognized, 130

~PacketStream
 PacketLib::PacketStream, 134

~PartOfPacket
 PacketLib::PartOfPacket, 140

~SDFBlockFixed
 PacketLib::SDFBlockFixed,
 147

~SDFBlockVariable
 PacketLib::SDFBlockVariable,
 155

~SDFNoBlockFixed
 PacketLib::SDFNoBlockFixed,
 176

~SDFNoBlockVariable
 PacketLib::SDFNoBlock-
 Variable, 180

~SHM
 PacketLib::SHM, 185

~Socket
 PacketLib::Socket, 189

~SocketServer
 PacketLib::SocketServer, 194

accept
 PacketLib::SocketServer, 194
 accepted
 PacketLib::InputSocketServer,
 70
 add
 PacketLib::PacketException,
 118
 addToNElement
 PacketLib::SDFBVBlock-
 Fixed, 169
 ARCH_BIGENDIAN
 PacketLibDefinition.h, 274

 bigEndian
 PacketLib::ByteStream, 28
 PacketLib::Device, 35
 PacketLib::Input, 54
 PacketLib::Output, 90
 PacketLib::Packet, 109
 PacketLib::PacketStream, 135
 binaryToInteger
 PacketLib::Utility, 208
 bind
 PacketLib::SocketServer, 194
 block
 PacketLib::SDFBlockFixed,
 152
 bookmarkPos
 PacketLib::File, 45
 buf
 PacketLib::MSGQ, 86
 buffer
 PacketLib::MemoryBuffer, 81
 byte
 PacketLib, 16
 byte_read
 PacketLib::File, 45
 byteInTheStream
 PacketLib::ByteStream, 28
 ByteStream
 PacketLib::ByteStream, 23
 ByteStream.cpp, 211
 ByteStream.h, 212
 byteToHexadecimal
 PacketLib::Utility, 208

 char_read
 PacketLib::File, 45
 charcat

PacketLib::Utility, 208
 close
 PacketLib::ConfigurationFile,
 31
 PacketLib::File, 42
 PacketLib::Input, 53
 PacketLib::InputFile, 56
 PacketLib::InputSocketServer,
 69
 PacketLib::InputTextFile, 76
 PacketLib::MSGQ, 85
 PacketLib::Output, 89
 PacketLib::OutputFile, 92
 PacketLib::OutputSocket-
 Client, 98
 PacketLib::SHM, 185
 PacketLib::Socket, 189
 closed
 PacketLib::Device, 36
 PacketLib::Input, 54
 PacketLib::InputText, 74
 closeInputStream
 PacketLib::InputPacket-
 StreamFile, 64
 ConfigurationFile
 PacketLib::ConfigurationFile,
 31
 ConfigurationFile.cpp, 213
 ConfigurationFile.h, 214
 connect
 PacketLib::Socket, 189
 convertToInteger
 PacketLib::Utility, 208
 count_object
 PacketLib::ByteStream, 28
 count_object2
 PacketLib::ByteStream, 28
 count_object_deleted
 PacketLib::ByteStream, 28
 count_object_deleted2
 PacketLib::ByteStream, 28
 create
 PacketLib::MSGQ, 85
 PacketLib::SHM, 185
 PacketLib::Socket, 189
 created
 PacketLib::MSGQ, 86
 PacketLib::SHM, 186
 createPacketType
 PacketLib::Packet, 103

PacketLib::PacketNotRecognized, 130
createStreamStructure
 PacketLib::PacketStream, 134
currentpwd
 PacketLib::ConfigurationFile,
 32
 PacketLib::MemoryBuffer, 81

dataField
 PacketLib::FileStream, 49
 PacketLib::Packet, 109
dataFieldHeader
 PacketLib::PacketDataField,
 116
DataFieldHeader.cpp, 216
DataFieldHeader.h, 217
DEBUGMODE
 PacketLibDefinition.h, 274
deleteExternalByteStream
 PacketLib::Packet, 103
deleteFields
 PacketLib::PartOfPacket, 140
deleteStreamMemory
 PacketLib::ByteStream, 24
destroy
 PacketLib::MSGQ, 85
 PacketLib::SHM, 185
detPacketType
 PacketLib::InputPacketStream, 60
Device
 PacketLib::Device, 35
Device.cpp, 219
Device.h, 220
dim
 PacketLib::MemoryBuffer, 81
dimension
 PacketLib::Field, 38
dimmsg
 PacketLib::MSGQ, 86
dimPrefix
 PacketLib::FileStream, 49
 PacketLib::Packet, 109
 PacketLib::PacketStream, 135
dword
 PacketLib, 17

endOutputStream
 PacketLib::ByteStream, 24

endsession
 PacketLib::Device, 36
eof
 PacketLib::Device, 36
 PacketLib::Input, 54
 PacketLib::InputText, 74
EOI
 PacketLibDefinition.h, 274
error
 PacketLib::PacketException,
 118
errorcode
 PacketLib::PacketException,
 119
extractFileName
 PacketLib::Utility, 208

FALSE
 PacketLibDefinition.h, 274
fchdir
 PacketLib::File, 42
 PacketLib::InputTextFile, 76
Field
 PacketLib::Field, 37
Field.cpp, 221
Field.h, 222
fieldNumber
 PacketLib::PacketIdentifier,
 128
fields
 PacketLib::PartOfPacket, 143
fieldsDimension
 PacketLib::PartOfPacket, 143
File
 PacketLib::File, 41
file
 PacketLib::InputFile, 57
 PacketLib::InputTextFile, 78
 PacketLib::OutputFile, 93
File.cpp, 223
File.h, 224
filename
 PacketLib::File, 45
 PacketLib::InputFile, 57
 PacketLib::OutputFile, 93
filenameConfig
 PacketLib::PacketStream, 136
fileNameStream
 PacketLib::InputPacketStreamFile, 66

FileStream
 PacketLib::FileStream, 48
 FileStream.cpp, 225
 FileStream.h, 226
 FileStreamPointer
 PacketLib::FileStreamPointer,
 50
 FileStreamPointer.cpp, 227
 FileStreamPointer.h, 228
 find
 PacketLib::File, 42
 first_output_stream_setted
 PacketLib::Packet, 109
 fixed
 PacketLib::SourceDataField,
 204
 format_output
 PacketLib::Utility, 208
 fp
 PacketLib::File, 45
 freeRun
 PacketLib::InputPacket-
 StreamFile, 64
 ftimeToChar
 PacketLib::Utility, 208
 generateStream
 PacketLib::Packet, 103
 PacketLib::PacketDataField,
 114
 PacketLib::PartOfPacket, 140
 PacketLib::SDFBlockFixed,
 148
 PacketLib::SDFBlockVariable,
 156
 PacketLib::SDFBVBlock, 164
 get_reset_output_stream
 PacketLib::SourceDataField,
 198
 getbits
 PacketLib::Utility, 209
 getbits2
 PacketLib::Utility, 209
 getbuffer
 PacketLib::MemoryBuffer, 80
 getBufferDimension
 PacketLib::MemoryBuffer, 80
 getByte
 PacketLib::ByteStream, 24
 PacketLib::File, 42
 getChildpid
 PacketLib::Utility, 209
 getChildpidlive
 PacketLib::Utility, 209
 getDimension
 PacketLib::ByteStream, 24
 PacketLib::Packet, 103
 PacketLib::PacketDataField,
 114
 PacketLib::PartOfPacket, 140
 PacketLib::SDFBlockFixed,
 148
 PacketLib::SDFBVBlock, 164
 PacketLib::SDFNoBlock-
 Variable, 180
 PacketLib::SourceDataField,
 198
 getEndSession
 PacketLib::Device, 35
 geterror
 PacketLib::PacketException,
 118
 getErrorCode
 PacketLib::PacketException,
 118
 getFields
 PacketLib::PartOfPacket, 140
 PacketLib::SDFBlockFixed,
 148, 149
 PacketLib::SDFBlockVariable,
 156, 157
 PacketLib::SDFBVBlock, 164
 PacketLib::SDFNoBlockFixed,
 176
 PacketLib::SDFNoBlock-
 Variable, 180
 PacketLib::SourceDataField,
 198, 199
 getFieldValue
 PacketLib::PartOfPacket, 141
 PacketLib::SDFBlockFixed,
 149
 PacketLib::SDFBlockVariable,
 157
 PacketLib::SDFBVBlock, 165
 PacketLib::SDFNoBlockFixed,
 176

PacketLib::SDFNoBlock-
Variable, 181
PacketLib::SourceDataField,
199
getFieldWithPacketDimension
PacketLib::PacketHeader, 125
getFileStreamPointer
PacketLib::InputPacket-
StreamFile, 64
getHeaderDimension
PacketLib::PacketStream, 134
getIndexOfNBlock
PacketLib::SourceDataField,
200
getLastLineRead
PacketLib::File, 42
PacketLib::InputText, 73
PacketLib::InputTextFile, 76
getLine
PacketLib::ConfigurationFile,
31
PacketLib::File, 42
PacketLib::InputText, 73
PacketLib::InputTextFile, 77
getMaxDimension
PacketLib::Packet, 104
PacketLib::PacketDataField,
115
PacketLib::SDFBlockFixed,
150
PacketLib::SDFBlockVariable,
158
PacketLib::SDFBVBlock, 165
PacketLib::SDFNoBlockFixed,
177
PacketLib::SDFNoBlock-
Variable, 181
PacketLib::SourceDataField,
200
getMaxNumberOfBlock
PacketLib::SourceDataField,
200
getMaxNumberOfElement
PacketLib::SDFBVBlock-
Fixed, 168
getMaxNumberOfElements
PacketLib::SDFBlockVariable,
158
PacketLib::SourceDataField,
200
getMemAllocation
PacketLib::ByteStream, 24
getName
PacketLib::Packet, 104
getNByte
PacketLib::File, 43
getNumberOfFields
PacketLib::PartOfPacket, 141
PacketLib::SDFBlockFixed,
150
PacketLib::SDFBlockVariable,
158, 159
PacketLib::SDFBVBlock, 165
PacketLib::SDFNoBlock-
Variable, 181
PacketLib::SourceDataField,
201
getNumberOfFileStreamPointer
PacketLib::InputPacket-
StreamFile, 65
getNumberOfPacketType
PacketLib::PacketStream, 134
getNumberOfRealDataBlock
PacketLib::PacketDataField,
115
PacketLib::SourceDataField,
201
getNumberOfRealElement
PacketLib::SDFBlockVariable,
159
PacketLib::SDFBVBlock-
Fixed, 168
PacketLib::SourceDataField,
202
getOutputStream
PacketLib::ByteStream, 24
PacketLib::Packet, 104
getPacketFromFileStreamPointer
PacketLib::InputPacket-
StreamFile, 65
getPacketFromStream
PacketLib::InputPacket-
StreamFile, 65
getPacketID
PacketLib::Packet, 104
getPacketLength
PacketLib::PacketHeader, 125
getPacketType
PacketLib::PacketStream, 134
getpos

PacketLib::File, 43
 PacketLib::InputTextFile, 77
 getPrefixDimension
 PacketLib::PacketStream, 135
 getStream
 PacketLib::ByteStream, 25
 getSubByteStream
 PacketLib::ByteStream, 25
 getSubByteStreamCopy
 PacketLib::ByteStream, 25
 getSubFromNBlock
 PacketLib::SourceDataField,
 202
 getType
 PacketLib::Input, 53
 PacketLib::InputFile, 56
 PacketLib::InputSocketServer,
 69
 PacketLib::Output, 89
 PacketLib::OutputFile, 92
 PacketLib::OutputSocket-
 Client, 98
 getValue
 PacketLib::ByteStream, 25

 header
 PacketLib::FileStream, 49
 PacketLib::Packet, 109
 headerReference
 PacketLib::PacketStream, 136
 hexadecimalToInteger
 PacketLib::Utility, 209
 host
 PacketLib::OutputSocket-
 Client, 99

 identifiers
 PacketLib::Packet, 110
 in
 PacketLib::InputPacket-
 Stream, 61
 index
 PacketLib::FileStreamPointer,
 51
 indexOfNBlock
 PacketLib::SourceDataField,
 204
 indexOfNElement
 PacketLib::SDFBVBlock-
 Fixed, 169

 indexread
 PacketLib::MemoryBuffer, 81
 indexwrite
 PacketLib::MemoryBuffer, 81
 initialPosition
 PacketLib::InputPacket-
 StreamFile, 66
 Input
 PacketLib::Input, 53
 Input.cpp, 229
 Input.h, 230
 InputFile
 PacketLib::InputFile, 56
 InputFile.cpp, 231
 InputFile.h, 232
 InputPacketStream
 PacketLib::InputPacket-
 Stream, 59
 InputPacketStream.cpp, 233
 InputPacketStream.h, 234
 InputPacketStreamFile
 PacketLib::InputPacket-
 StreamFile, 64
 InputPacketStreamFile.cpp, 235
 InputPacketStreamFile.h, 236
 InputSocketServer
 PacketLib::InputSocketServer,
 69
 InputSocketServer.cpp, 237
 InputSocketServer.h, 238
 inputStream
 PacketLib::InputPacket-
 StreamFile, 66
 InputText
 PacketLib::InputText, 73
 InputText.cpp, 239
 InputText.h, 240
 InputTextFile
 PacketLib::InputTextFile, 76
 InputTextFile.cpp, 241
 InputTextFile.h, 242
 integerToString
 PacketLib::Utility, 209
 integerToString2
 PacketLib::Utility, 209
 is_valid
 PacketLib::Socket, 189
 isBigEndian
 PacketLib::Device, 35
 PacketLib::PacketStream, 135

isBigEndian
 PacketLib::ByteStream, 26
 PacketLib::Input, 53
 PacketLib::Output, 89
isBlock
 PacketLib::SourceDataField,
 202
isblock
 PacketLib::SourceDataField,
 204
isClosed
 PacketLib::Device, 35
 PacketLib::File, 43
 PacketLib::Input, 53
 PacketLib::InputText, 73
 PacketLib::Output, 89
isclosed
 PacketLib::Output, 90
isCreated
 PacketLib::MSGQ, 86
 PacketLib::SHM, 186
isEOF
 PacketLib::Device, 35
 PacketLib::File, 43
 PacketLib::Input, 53
 PacketLib::InputText, 73
isFixed
 PacketLib::SourceDataField,
 202
isInputStreamEOF
 PacketLib::InputPacket-
 StreamFile, 65
isNumberOfBlockFixed
 PacketLib::SourceDataField,
 202

lastLineRead
 PacketLib::File, 46
listen
 PacketLib::SocketServer, 194
listOfFileStreamPointer
 PacketLib::InputPacket-
 StreamFile, 66
loadBuffer
 PacketLib::MemoryBuffer, 80
loadFields
 PacketLib::PartOfPacket, 141
 PacketLib::SDFBlockFixed,
 151

 PacketLib::SDFBlockVariable,
 159
 PacketLib::SDFNoBlockFixed,
 177
 PacketLib::SDFNoBlock-
 Variable, 182
 PacketLib::SourceDataField,
 202
loadFieldsInBuffer
 PacketLib::PartOfPacket, 142
loadHeader
 PacketLib::PacketHeader, 125
loadIdentifiers
 PacketLib::Packet, 104

m_addr
 PacketLib::Socket, 190
m_sock
 PacketLib::Socket, 190
MAXCONNECTIONS
 Socket.h, 316
MAXHOSTNAME
 Socket.h, 316
maxNumberOfBlock
 PacketLib::SourceDataField,
 204
maxNumberOfElement
 PacketLib::SDFBVBlock-
 Fixed, 169
MAXRECV
 Socket.h, 316
mem_allocation
 PacketLib::ByteStream, 28
mem_allocation_constructor
 PacketLib::ByteStream, 28
memBookmarkPos
 PacketLib::File, 44
 PacketLib::InputTextFile, 77
memByteStream
 PacketLib::Packet, 104
MemoryBuffer
 PacketLib::MemoryBuffer, 80
MemoryBuffer.cpp, 243
MemoryBuffer.h, 244
MSG_TYPE
 MSGQ.cpp, 245
msgbuf
 PacketLib, 17
MSGQ
 PacketLib::MSGQ, 85

MSGQ.cpp, 245
 MSG_TYPE, 245
 PERMMSGQ, 245
 MSGQ.h, 246
 msgqid
 PacketLib::MSGQ, 86
 msgqkey
 PacketLib::MSGQ, 87
 mtext
 PacketLib::mesgbuf, 83
 mtype
 PacketLib::mesgbuf, 83

 name
 PacketLib::Field, 38
 PacketLib::Packet, 110
 PacketLib::PacketHeader, 126
 nameOfPacket
 PacketLib::FileStreamPointer,
 51
 new_sock
 PacketLib::InputSocketServer,
 70
 num_slot
 PacketLib::SHM, 186
 number_of_identifier
 PacketLib::Packet, 110
 numberOfBlockFixed
 PacketLib::SourceDataField,
 205
 numberOffields
 PacketLib::PartOfPacket, 143
 numberOffieldWithPacketDimension
 PacketLib::PacketHeader, 126
 numberOffileStreamPointer
 PacketLib::InputPacket-
 StreamFile, 66
 numberOfPacketType
 PacketLib::PacketStream, 136
 numberOfRealDataBlock
 PacketLib::SourceDataField,
 205

 open
 PacketLib::ConfigurationFile,
 32
 PacketLib::File, 44
 PacketLib::Input, 53
 PacketLib::InputFile, 56

 PacketLib::InputSocketServer,
 70
 PacketLib::InputTextFile, 77
 PacketLib::MSGQ, 86
 PacketLib::Output, 89
 PacketLib::OutputFile, 92
 PacketLib::OutputSocket-
 Client, 98
 PacketLib::SHM, 186
 openInputStream
 PacketLib::InputPacket-
 StreamFile, 65
 out
 PacketLib::OutputPacket-
 Stream, 96
 Output
 PacketLib::Output, 89
 Output.cpp, 247
 Output.h, 248
 outputFile
 PacketLib::OutputFile, 92
 outputFile.cpp, 249
 outputFile.h, 250
 OutputPacketStream
 PacketLib::OutputPacket-
 Stream, 95
 OutputPacketStream.cpp, 251
 OutputPacketStream.h, 252
 OutputSocketClient
 PacketLib::OutputSocket-
 Client, 98
 OutputSocketClient.cpp, 253
 OutputSocketClient.h, 254
 outputStream
 PacketLib::PartOfPacket, 144

 Packet
 PacketLib::Packet, 102
 packet
 PacketLib::Packet, 110
 Packet.cpp, 255
 Packet.h, 256
 packet_output
 PacketLib::Packet, 110
 PacketDataField
 PacketLib::PacketDataField,
 114
 PacketDataField.cpp, 258
 PacketDataField.h, 259
 PacketException

PacketLib::PacketException,
 118
PacketException.cpp, 261
PacketException.h, 262
PacketExceptionFormatException
 PacketLib::PacketException-
 FormatException, 121
PacketExceptionFormatException.cpp,
 263
PacketExceptionFormatException.h, 264
PacketExceptionIO
 PacketLib::PacketException-
 IO, 123
PacketExceptionIO.cpp, 265
PacketExceptionIO.h, 266
PacketHeader
 PacketLib::PacketHeader, 125
PacketHeader.cpp, 268
PacketHeader.h, 269
packetID
 PacketLib::Packet, 110
PacketIdentifier
 PacketLib::PacketIdentifier,
 127
PacketIdentifier.cpp, 270
PacketIdentifier.h, 271
PacketLib, 13
 byte, 16
 dword, 17
 msgbuf, 17
 word, 17
PacketLib::ByteStream, 21
 ~ByteStream, 23
 bigEndian, 28
 byteInTheStream, 28
 ByteStream, 23
 count_object, 28
 count_object2, 28
 count_object_deleted, 28
 count_object_deleted2, 28
 deleteStreamMemory, 24
 endOutputStream, 24
 getByte, 24
 getDimension, 24
 getMemAllocation, 24
 getOutputStream, 24
 getStream, 25
 getSubByteStream, 25
 getSubByteStreamCopy, 25
 getValue, 25
 isBigEndian, 26
 mem_allocation, 28
 mem_allocation_constructor,
 28
 printStreamInHexadecimal, 26
 setByte, 26
 setMemoryAllocated, 26
 setStream, 26, 27
 setStreamCopy, 27
 setWord, 27
 stream, 29
 swap, 27
PacketLib::ConfigurationFile, 30
 ~ConfigurationFile, 31
 close, 31
 ConfigurationFile, 31
 currentpwd, 32
 getLine, 31
 open, 32
PacketLib::DataFieldHeader, 33
PacketLib::Device, 34
 bigEndian, 35
 closed, 36
 Device, 35
 endSession, 36
 eof, 36
 getEndSession, 35
 isBigEndian, 35
 isClosed, 35
 isEOF, 35
 setEndSession, 35
PacketLib::Field, 37
 ~Field, 38
 dimension, 38
 Field, 37
 name, 38
 predefinedValue, 38
 progressiv, 38
 thereIsPredefinedValue, 38
 value, 38
PacketLib::File, 40
 bookmarkPos, 45
 byte_read, 45
 char_read, 45
 close, 42
 fchdir, 42
 File, 41
 filename, 45
 find, 42
 fp, 45

```

getByte, 42
getLastLineRead, 42
getLine, 42
getNByte, 43
getpos, 43
isClosed, 43
isEOF, 43
lastLineRead, 46
memBookmarkPos, 44
open, 44
setFirstPos, 44
setLastBookmarkPos, 44
setpos, 44
startPosition, 46
writeByteStream, 45
writeString, 45
PacketLib::FileStream, 47
    dataField, 49
    dimPrefix, 49
    FileStream, 48
    header, 49
    prefix, 49
    readDataField, 48
    readHeader, 48
    readPrefix, 48
    thereIsPrefix, 49
PacketLib::FileStreamPointer, 50
    ~FileStreamPointer, 50
    FileStreamPointer, 50
    index, 51
    nameOfPacket, 51
    pointerEnd, 51
    pointerStart, 51
    typeOfPacket, 51
PacketLib::Input, 52
    ~Input, 53
    bigEndian, 54
    close, 53
    closed, 54
    eof, 54
    getType, 53
    Input, 53
    isBigEndian, 53
    isClosed, 53
    isEOF, 53
    open, 53
    readByteStream, 54
    readString, 54
PacketLib::InputFile, 55
    ~InputFile, 56
close, 56
file, 57
filename, 57
getType, 56
InputFile, 56
open, 56
readByteStream, 57
readString, 57
PacketLib::InputPacketStream, 58
    ~InputPacketStream, 59
    detPacketType, 60
    in, 61
    InputPacketStream, 59
    readPacket, 60
    setInput, 61
PacketLib::InputPacketStreamFile,
    62
    ~InputPacketStreamFile, 64
    closeInputStream, 64
    fileNameStream, 66
    freeRun, 64
    getFileStreamPointer, 64
    getNumberOfFileStream-
        Pointer, 65
    getPacketFromFileStream-
        Pointer, 65
    getPacketFromStream, 65
    initialPosition, 66
    InputPacketStreamFile, 64
    inputStream, 66
    isInputStreamEOF, 65
    listOfFileStreamPointer, 66
    numberOfFileStreamPointer,
        66
    openInputStream, 65
    setFileNameStream, 65
    setInitialPosition, 66
PacketLib::InputSocketServer, 68
    ~InputSocketServer, 69
    accepted, 70
    close, 69
    getType, 69
    InputSocketServer, 69
    new_sock, 70
    open, 70
    port, 70
    readByteStream, 70
    readString, 70
    socketserver, 70
PacketLib::InputText, 72

```

~InputText, 73
closed, 74
eof, 74
getLastLineRead, 73
getLine, 73
InputText, 73
isClosed, 73
isEOF, 73
PacketLib::InputTextFile, 75
 ~InputTextFile, 76
 close, 76
 fchdir, 76
 file, 78
 getLastLineRead, 76
 getLine, 77
 getpos, 77
 InputTextFile, 76
 memBookmarkPos, 77
 open, 77
 setFirstPos, 77
 setLastBookmarkPos, 77
 setpos, 78
PacketLib::MemoryBuffer, 79
 ~MemoryBuffer, 80
 buffer, 81
 currentpwd, 81
 dim, 81
 getbuffer, 80
 getBufferDimension, 80
 indexread, 81
 indexwrite, 81
 loadBuffer, 80
 MemoryBuffer, 80
 readRewind, 80
 saveBuffer, 80
 setbuffer, 81
 writeRewind, 81
PacketLib::mesgbuf, 83
 mtext, 83
 mtype, 83
PacketLib::MSGQ, 84
 ~MSGQ, 85
 buf, 86
 close, 85
 create, 85
 created, 86
 destroy, 85
 dimmsg, 86
 isCreated, 86
 MSGQ, 85
 msgid, 86
 msgqkey, 87
 open, 86
 readMessage, 86
 writeMessage, 86
 PacketLib::Output, 88
 bigEndian, 90
 close, 89
 getType, 89
 isBigEndian, 89
 isClosed, 89
 isclosed, 90
 open, 89
 Output, 89
 writeByteStream, 89
 writeString, 89
 PacketLib::OutputFile, 91
 ~OutputFile, 92
 close, 92
 file, 93
 filename, 93
 getType, 92
 open, 92
 OutputFile, 92
 writeByteStream, 92
 writeString, 93
 PacketLib::OutputPacketStream,
 94
 out, 96
 OutputPacketStream, 95
 setOutput, 95
 writePacket, 95
 PacketLib::OutputSocketClient, 97
 ~OutputSocketClient, 98
 close, 98
 getType, 98
 host, 99
 open, 98
 OutputSocketClient, 98
 port, 99
 socketclient, 99
 writeByteStream, 99
 writeString, 99
 PacketLib::Packet, 100
 ~Packet, 102
 bigEndian, 109
 createPacketType, 103
 dataField, 109
 deleteExternalByteStream,
 103

dimPrefix, 109
 first_output_stream_setted, 109
 generateStream, 103
 getDimension, 103
 getMaxDimension, 104
 getName, 104
 getOutputStream, 104
 getPacketID, 104
 header, 109
 identifiers, 110
 loadIdentifiers, 104
 memByteStream, 104
 name, 110
 number_of_identifier, 110
 Packet, 102
 packet, 110
 packet_output, 110
 packetID, 110
 prefix, 111
 printIdentifiers, 105
 printPacketValue, 105
 setAndVerifyPacketValue, 105
 setPacketID, 106
 setPacketValue, 106
 setPacketValueDataField-
 Header, 107
 setPacketValueHeader, 107
 setPacketValuePrefix, 107
 setPacketValueSource-
 DataField, 107
 setPacketValueVerify, 108
 tempDataField, 111
 tempDataFieldHeader, 111
 tempHeader, 111
 tempPacketDataField, 111
 thereIsPrefix, 108
 thereisprefix, 111
 type_of_identifier, 112
 verifyPacketValue, 108
 PacketLib::PacketDataField, 113
 ~PacketDataField, 114
 dataFieldHeader, 116
 generateStream, 114
 getDimension, 114
 getMaxDimension, 115
 getNumberOfRealDataBlock,
 115
 PacketDataField, 114
 setNumberOfRealDataBlock,
 115
 setOutputStream, 115
 sourceDataField, 116
 PacketLib::PacketException, 117
 ~PacketException, 118
 add, 118
 error, 118
 errorcode, 119
 geterror, 118
 geterrorcode, 118
 PacketException, 118
 seterrorcode, 118
 PacketLib::PacketExceptionFile-
 Format, 120
 ~PacketExceptionFileFormat,
 121
 PacketExceptionFileFormat,
 121
 PacketLib::PacketExceptionIO, 122
 ~PacketExceptionIO, 123
 PacketExceptionIO, 123
 PacketLib::PacketHeader, 124
 ~PacketHeader, 125
 getFieldWithPacketDimen-
 sion, 125
 getPacketLength, 125
 loadHeader, 125
 name, 126
 numberOfFieldWithPacketDi-
 mension, 126
 PacketHeader, 125
 setName, 126
 PacketLib::PacketIdentifier, 127
 ~PacketIdentifier, 127
 fieldNumber, 128
 PacketIdentifier, 127
 type, 128
 value, 128
 PacketLib::PacketNotRecognized,
 129
 ~PacketNotRecognized, 130
 createPacketType, 130
 PacketNotRecognized, 130
 setPacketValue, 130
 PacketLib::PacketStream, 132
 ~PacketStream, 134
 big endian, 135
 createStreamStructure, 134
 dimPrefix, 135
 filenameConfig, 136
 getHeaderDimension, 134

getNumberOfPacketType, 134
getPacketType, 134
getPrefixDimension, 135
headerReference, 136
isBigEndian, 135
numberOfPacketType, 136
packetLibVersion, 135
PacketStream, 134
packetType, 136
prefix, 136
setFileNameConfig, 135
thereIsPrefix, 135
PacketLib::PartOfPacket, 138
 ~PartOfPacket, 140
 deleteFields, 140
 fields, 143
 fieldsDimension, 143
 generateStream, 140
 getDimension, 140
 getFields, 140
 getFieldValue, 141
 getNumberOfFields, 141
 loadFields, 141
 loadFieldsInBuffer, 142
 numberOfFields, 143
 outputstream, 144
 PartOfPacket, 140
 printStructure, 142
 printValue, 142
 setByteStream, 142
 setFieldValue, 142
 setOutputStream, 143
 stream, 144
PacketLib::SDFBFBlock, 145
PacketLib::SDFBlockFixed, 146
 ~SDFBlockFixed, 147
 block, 152
 generateStream, 148
 getDimension, 148
 getFields, 148, 149
 getFieldValue, 149
 getMaxDimension, 150
 getNumberOfFields, 150
 loadFields, 151
 printStructure, 151
 printValue, 151
 SDFBlockFixed, 147
 setByteStream, 151
 setFieldValue, 151, 152
 setOutputStream, 152
 tempBlock, 152
PacketLib::SDFBlockVariable, 154
 ~SDFBlockVariable, 155
 generateStream, 156
 getDimension, 156
 getFields, 156, 157
 getFieldValue, 157
 getMaxDimension, 158
 getMaxNumberOfElements,
 158
 getNumberOfFields, 158, 159
 getNumberOfRealElement,
 159
 loadFields, 159
 printStructure, 159
 printValue, 159
 SDFBlockVariable, 155
 setByteStream, 160
 setFieldValue, 160
 setNumberOfRealElement, 161
 setOutputStream, 161
 tempBlock, 161
 theListOfSDFBVBlock, 161
PacketLib::SDFBVBlock, 163
 generateStream, 164
 getDimension, 164
 getFields, 164
 getFieldValue, 165
 getMaxDimension, 165
 getNumberOfFields, 165
 setOutputStream, 165
 theListOfSDFBVBlockVari-
 able, 166
theSDFBVFixed, 166
PacketLib::SDFBVBlockFixed, 167
 addToNElement, 169
 getMaxNumberOfElement,
 168
 getNumberOfRealElement,
 168
 indexOfNElement, 169
 maxNumberOfElement, 169
 setAddToNElement, 168
 setIndexOfNElement, 168
 setMaxNumberOfElement, 168
 setNumberOfRealElement, 169
PacketLib::SDFBVBlockVariable,
 170
PacketLib::SDFNBVariable, 172
PacketLib::SDFNBVFixed, 174

PacketLib::SDFNoBlockFixed, 175
 ~SDFNoBlockFixed, 176
 getFields, 176
 getFieldValue, 176
 getMaxDimension, 177
 loadFields, 177
SDFNoBlockFixed, 176
 setFieldValue, 177
PacketLib::SDFNoBlockVariable,
 179
 ~SDFNoBlockVariable, 180
 getDimension, 180
 getFields, 180
 getFieldValue, 181
 getMaxDimension, 181
 getNumberOfFields, 181
 loadFields, 182
 printStructure, 182
 printValue, 182
SDFNoBlockVariable, 180
 setByteStream, 182
 setFieldValue, 182
 theListOfSDFNBVariable, 183
 theSDFNBVFixed, 183
PacketLib::SHM, 184
 ~SHM, 185
 close, 185
 create, 185
 created, 186
 destroy, 185
 isCreated, 186
 num_slot, 186
 open, 186
 readSlot, 186
SHM, 185
 shm, 186
 shmid, 186
 shmkey, 187
 slot_dim, 187
 writeSlot, 186
PacketLib::Socket, 188
 ~Socket, 189
 close, 189
 connect, 189
 create, 189
 is_valid, 189
 m_addr, 190
 m_sock, 190
 recv, 190
 send, 190
Socket, 189
PacketLib::SocketClient, 191
 SocketClient, 192
PacketLib::SocketServer, 193
 ~SocketServer, 194
 accept, 194
 bind, 194
 listen, 194
 set_non_blocking, 194
 SocketServer, 194
PacketLib::SourceDataField, 196
 fixed, 204
 get_reset_output_stream, 198
 getDimension, 198
 getFields, 198, 199
 getFieldValue, 199
 getIndexOfNBlock, 200
 getMaxDimension, 200
 getMaxNumberOfBlock, 200
 getMaxNumberOfElements,
 200
 getNumberOfFields, 201
 getNumberOfRealDataBlock,
 201
 getNumberOfRealElement,
 202
 getSubFromNBlock, 202
 indexOfNBlock, 204
 isBlock, 202
 isblock, 204
 isFixed, 202
 isNumberOfBlockFixed, 202
 loadFields, 202
 maxNumberOfBlock, 204
 numberOfBlockFixed, 205
 numberOfRealDataBlock, 205
 reset_output_stream, 205
 set_reset_output_stream, 203
 setFieldValue, 203
 setNumberOfRealDataBlock,
 203
 setNumberOfRealElement, 204
SourceDataField, 198
 subFromNBlock, 205
PacketLib::Utility, 207
 binaryToInteger, 208
 byteToHexadecimal, 208
 charcat, 208
 convertToInteger, 208
 extractFileName, 208

format_output, 208
ftimeToChar, 208
getbits, 209
getbits2, 209
getchildpid, 209
getchildpidlive, 209
hexadecimalToInteger, 209
integerToString, 209
integerToString2, 209
stringToHexadecimal, 209
strlen, 210
wordToBinary, 210
wordToBinary2, 210
PacketLibDefinition.h, 272
 ARCH_BIGENDIAN, 274
 DEBUGMODE, 274
 EOI, 274
 FALSE, 274
 PRINTDEBUG, 274
 TRUE, 274
packetLibVersion
 PacketLib::PacketStream, 135
PacketNotRecognized
 PacketLib::PacketNot-
 Recognized, 130
PacketNotRecognized.cpp, 275
PacketNotRecognized.h, 276
PacketStream
 PacketLib::PacketStream, 134
PacketStream.cpp, 277
PacketStream.h, 278
packetType
 PacketLib::PacketStream, 136
PartOfPacket
 PacketLib::PartOfPacket, 140
PartOfPacket.cpp, 279
 pattern, 279
PartOfPacket.h, 280
pattern
 PartOfPacket.cpp, 279
PERM
 SHM.cpp, 312
PERMMMSGQ
 MSGQ.cpp, 245
pointerEnd
 PacketLib::FileStreamPointer,
 51
pointerStart
 PacketLib::FileStreamPointer,
 51
port
 PacketLib::InputSocketServer,
 70
 PacketLib::OutputSocket-
 Client, 99
predefinedValue
 PacketLib::Field, 38
prefix
 PacketLib::FileStream, 49
 PacketLib::Packet, 111
 PacketLib::PacketStream, 136
PRINTDEBUG
 PacketLibDefinition.h, 274
printIdentifiers
 PacketLib::Packet, 105
printPacketValue
 PacketLib::Packet, 105
printStreamInHexadecimal
 PacketLib::ByteStream, 26
printStructure
 PacketLib::PartOfPacket, 142
 PacketLib::SDFBlockFixed,
 151
 PacketLib::SDFBlockVariable,
 159
 PacketLib::SDFNoBlock-
 Variable, 182
printValue
 PacketLib::PartOfPacket, 142
 PacketLib::SDFBlockFixed,
 151
 PacketLib::SDFBlockVariable,
 159
 PacketLib::SDFNoBlock-
 Variable, 182
progressiv
 PacketLib::Field, 38
readByteStream
 PacketLib::Input, 54
 PacketLib::InputFile, 57
 PacketLib::InputSocketServer,
 70
readDataField
 PacketLib::FileStream, 48
readHeader
 PacketLib::FileStream, 48
readMessage
 PacketLib::MSGQ, 86
readPacket

PacketLib::InputPacketStream, 60
 readPrefix
 PacketLib::FileStream, 48
 readRewind
 PacketLib::MemoryBuffer, 80
 readSlot
 PacketLib::SHM, 186
 readString
 PacketLib::Input, 54
 PacketLib::InputFile, 57
 PacketLib::InputSocketServer,
 70
 recv
 PacketLib::Socket, 190
 reset_output_stream
 PacketLib::SourceDataField,
 205

 saveBuffer
 PacketLib::MemoryBuffer, 80
 SDFBFBlock.cpp, 282
 SDFBFBlock.h, 283
 SDFBlockFixed
 PacketLib::SDFBlockFixed,
 147
 SDFBlockFixed.cpp, 285
 SDFBlockFixed.h, 286
 SDFBlockVariable
 PacketLib::SDFBlockVariable,
 155
 SDFBlockVariable.cpp, 288
 SDFBlockVariable.h, 289
 SDFBVBlock.cpp, 291
 SDFBVBlock.h, 292
 SDFBVBlockFixed.cpp, 294
 SDFBVBlockFixed.h, 295
 SDFBVBlockVariable.cpp, 297
 SDFBVBlockVariable.h, 298
 SDFNBVariable.cpp, 300
 SDFNBVariable.h, 301
 SDFNBFixed.cpp, 303
 SDFNBFixed.h, 304
 SDFNoBlockFixed
 PacketLib::SDFNoBlockFixed,
 176
 SDFNoBlockFixed.cpp, 306
 SDFNoBlockFixed.h, 307
 SDFNoBlockVariable

PacketLib::SDFNoBlockVariable, 180
 SDFNoBlockVariable.cpp, 309
 SDFNoBlockVariable.h, 310
 send
 PacketLib::Socket, 190
 set_non_blocking
 PacketLib::SocketServer, 194
 set_reset_output_stream
 PacketLib::SourceDataField,
 203
 setAddToNElement
 PacketLib::SDFBVBlockFixed, 168
 setAndVerifyPacketValue
 PacketLib::Packet, 105
 setbuffer
 PacketLib::MemoryBuffer, 81
 setByte
 PacketLib::ByteStream, 26
 setByteStream
 PacketLib::PartOfPacket, 142
 PacketLib::SDFBlockFixed,
 151
 PacketLib::SDFBlockVariable,
 160
 PacketLib::SDFNoBlockVariable, 182
 setEndSession
 PacketLib::Device, 35
 setErrorCode
 PacketLib::PacketException,
 118
 setFieldValue
 PacketLib::PartOfPacket, 142
 PacketLib::SDFBlockFixed,
 151, 152
 PacketLib::SDFBlockVariable,
 160
 PacketLib::SDFNoBlockFixed,
 177
 PacketLib::SDFNoBlockVariable, 182
 PacketLib::SourceDataField,
 203
 setFileNameConfig
 PacketLib::PacketStream, 135
 setFileNameStream
 PacketLib::InputPacketStreamFile, 65

setFirstPos
 PacketLib::File, 44
 PacketLib::InputTextFile, 77
setIndexOfNElement
 PacketLib::SDFBVBlock-
 Fixed, 168
setInitialPosition
 PacketLib::InputPacket-
 StreamFile, 66
setInput
 PacketLib::InputPacket-
 Stream, 61
setLastBookmarkPos
 PacketLib::File, 44
 PacketLib::InputTextFile, 77
setMaxNumberOfElement
 PacketLib::SDFBVBlock-
 Fixed, 168
setMemoryAllocated
 PacketLib::ByteStream, 26
setName
 PacketLib::PacketHeader, 126
setNumberOfRealDataBlock
 PacketLib::PacketDataField,
 115
 PacketLib::SourceDataField,
 203
setNumberOfRealElement
 PacketLib::SDFBlockVariable,
 161
 PacketLib::SDFBVBlock-
 Fixed, 169
 PacketLib::SourceDataField,
 204
setOutput
 PacketLib::OutputPacket-
 Stream, 95
setOutputStream
 PacketLib::PacketDataField,
 115
 PacketLib::PartOfPacket, 143
 PacketLib::SDFBlockFixed,
 152
 PacketLib::SDFBlockVariable,
 161
 PacketLib::SDFBVBlock, 165
setPacketID
 PacketLib::Packet, 106
setPacketValue
 PacketLib::Packet, 106
 PacketLib::PacketNot-
 Recognized, 130
setPacketValueDataFieldHeader
 PacketLib::Packet, 107
setPacketValueHeader
 PacketLib::Packet, 107
setPacketValuePrefix
 PacketLib::Packet, 107
setPacketValueSourceDataField
 PacketLib::Packet, 107
setPacketValueVerify
 PacketLib::Packet, 108
setpos
 PacketLib::File, 44
 PacketLib::InputTextFile, 78
setStream
 PacketLib::ByteStream, 26, 27
setStreamCopy
 PacketLib::ByteStream, 27
setWord
 PacketLib::ByteStream, 27
SHM
 PacketLib::SHM, 185
shm
 PacketLib::SHM, 186
SHM.cpp, 312
 PERM, 312
SHM.h, 313
shmid
 PacketLib::SHM, 186
shmkey
 PacketLib::SHM, 187
slot_dim
 PacketLib::SHM, 187
Socket
 PacketLib::Socket, 189
Socket.cpp, 314
Socket.h, 315
 MAXCONNECTIONS, 316
 MAXHOSTNAME, 316
 MAXRECV, 316
SocketClient
 PacketLib::SocketClient, 192
socketclient
 PacketLib::OutputSocket-
 Client, 99
SocketClient.cpp, 317
SocketClient.h, 318
SocketServer
 PacketLib::SocketServer, 194

socketserver
 PacketLib::InputSocketServer,
 70
 SocketServer.cpp, 319
 SocketServer.h, 320
 SourceDataField
 PacketLib::SourceDataField,
 198
 sourceDataField
 PacketLib::PacketDataField,
 116
 SourceDataField.cpp, 321
 SourceDataField.h, 322
 startPosition
 PacketLib::File, 46
 std, 19
 stream
 PacketLib::ByteStream, 29
 PacketLib::PartOfPacket, 144
 stringToHexadecimal
 PacketLib::Utility, 209
 strlen
 PacketLib::Utility, 210
 subFromNBlock
 PacketLib::SourceDataField,
 205
 swap
 PacketLib::ByteStream, 27
 tempBlock
 PacketLib::SDFBlockFixed,
 152
 PacketLib::SDFBlockVariable,
 161
 tempDataField
 PacketLib::Packet, 111
 tempDataFieldHeader
 PacketLib::Packet, 111
 tempHeader
 PacketLib::Packet, 111
 tempPacketDataField
 PacketLib::Packet, 111
 theListOfSDFBVBlock
 PacketLib::SDFBlockVariable,
 161
 theListOfSDFBVBlockVariable
 PacketLib::SDFBVBlock, 166
 theListOfSDFNBVariable
 PacketLib::SDFNoBlock-
 Variable, 183
 thereIsPredefinedValue
 PacketLib::Field, 38
 thereIsPrefix
 PacketLib::FileStream, 49
 PacketLib::Packet, 108
 PacketLib::PacketStream, 135
 thereisprefix
 PacketLib::Packet, 111
 theSDFBVFixed
 PacketLib::SDFBVBlock, 166
 theSDFNBVFixed
 PacketLib::SDFNoBlock-
 Variable, 183
 TRUE
 PacketLibDefinition.h, 274
 type
 PacketLib::PacketIdentifier,
 128
 type_of_identifier
 PacketLib::Packet, 112
 typeOfPacket
 PacketLib::FileStreamPointer,
 51
 Utility.cpp, 323
 Utility.h, 324
 value
 PacketLib::Field, 38
 PacketLib::PacketIdentifier,
 128
 verifyPacketValue
 PacketLib::Packet, 108
 word
 PacketLib, 17
 wordToBinary
 PacketLib::Utility, 210
 wordToBinary2
 PacketLib::Utility, 210
 writeByteStream
 PacketLib::File, 45
 PacketLib::Output, 89
 PacketLib::OutputFile, 92
 PacketLib::OutputSocket-
 Client, 99
 writeMessage
 PacketLib::MSGQ, 86
 writePacket

PacketLib::OutputPacket-
Stream, 95
writeRewind
 PacketLib::MemoryBuffer, 81
writeSlot
 PacketLib::SHM, 186
writeString
 PacketLib::File, 45
 PacketLib::Output, 89
 PacketLib::OutputFile, 93
 PacketLib::OutputSocket-
 Client, 99