

qecc

0.8

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	Aad4 Class Reference . . . . .	7
4.1.1	Constructor & Destructor Documentation . . . . .	9
4.1.1.1	Aad4(bool b) . . . . .	9
4.1.2	Member Function Documentation . . . . .	9
4.1.2.1	encode(bool b) override . . . . .	9
4.1.2.2	getCS() override . . . . .	9
4.1.2.3	getDescriptor() override . . . . .	9
4.1.2.4	run() override . . . . .	10
4.1.3	Member Data Documentation . . . . .	10
4.1.3.1	CS . . . . .	10
4.2	BitFlip Class Reference . . . . .	10
4.2.1	Detailed Description . . . . .	12
4.2.2	Constructor & Destructor Documentation . . . . .	12
4.2.2.1	BitFlip(bool b) . . . . .	12
4.2.3	Member Function Documentation . . . . .	12

4.2.3.1	<code>encode(bool b)</code> override	12
4.2.3.2	<code>getCS()</code> override	12
4.2.3.3	<code>getDescriptor()</code> override	13
4.2.3.4	<code>run()</code> override	13
4.2.4	Member Data Documentation	13
4.2.4.1	CS	13
4.3	Channel Class Reference	13
4.3.1	Detailed Description	14
4.3.2	Constructor & Destructor Documentation	14
4.3.2.1	<code>Channel(int nodeid0, int nodeid1, Error *e)</code>	14
4.3.2.2	<code>~Channel()</code>	15
4.3.3	Member Function Documentation	15
4.3.3.1	<code>debugPrint()</code>	15
4.3.3.2	<code>getError()</code>	15
4.3.3.3	<code>getIds()</code>	15
4.3.3.4	<code>matchId(int nid0, int nid1)</code>	15
4.3.3.5	<code>matchOneGetOther(int id)</code>	16
4.4	Code Class Reference	16
4.4.1	Detailed Description	17
4.4.2	Constructor & Destructor Documentation	17
4.4.2.1	<code>Code(bool b)</code>	17
4.4.2.2	<code>~Code()</code>	18
4.4.3	Member Function Documentation	18
4.4.3.1	<code>addError(Error *e)</code>	18
4.4.3.2	<code>applyCGT(const cmat gate, const std::vector&lt; unsigned &gt; &amp;b)</code>	18
4.4.3.3	<code>applyGT(const cmat gate, unsigned b)</code>	18
4.4.3.4	<code>convertToMixed()</code>	18
4.4.3.5	<code>disableErrorDeletion()</code>	19
4.4.3.6	<code>encode(bool b)=0</code>	19
4.4.3.7	<code>error()</code>	19

4.4.3.8	<a href="#">getCS()=0</a>	19
4.4.3.9	<a href="#">getDescriptor()=0</a>	19
4.4.3.10	<a href="#">getMes(unsigned i)</a>	19
4.4.3.11	<a href="#">getOK()</a>	20
4.4.3.12	<a href="#">hadamardAllCodeBits()</a>	20
4.4.3.13	<a href="#">hadamardCodeBits(const std::vector&lt; unsigned &gt; &amp;b)</a>	20
4.4.3.14	<a href="#">run()=0</a>	20
4.4.3.15	<a href="#">setandmesAnc(const std::vector&lt; unsigned &gt; &amp;b, unsigned CS)</a>	20
4.4.3.16	<a href="#">setMixed()</a>	21
4.4.4	<a href="#">Friends And Related Function Documentation</a>	21
4.4.4.1	<a href="#">Error</a>	21
4.4.5	<a href="#">Member Data Documentation</a>	21
4.4.5.1	<a href="#">c</a>	21
4.4.5.2	<a href="#">d</a>	21
4.4.5.3	<a href="#">deleteError</a>	21
4.4.5.4	<a href="#">errorCounter</a>	21
4.4.5.5	<a href="#">errorlist</a>	21
4.4.5.6	<a href="#">input</a>	21
4.4.5.7	<a href="#">mixed</a>	22
4.4.5.8	<a href="#">ok</a>	22
4.4.5.9	<a href="#">result</a>	22
4.4.5.10	<a href="#">threadCounter</a>	22
4.5	<a href="#">Code5 Class Reference</a>	22
4.5.1	<a href="#">Detailed Description</a>	24
4.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	24
4.5.2.1	<a href="#">Code5(bool b)</a>	24
4.5.3	<a href="#">Member Function Documentation</a>	24
4.5.3.1	<a href="#">encode(bool b) override</a>	24
4.5.3.2	<a href="#">getCS() override</a>	24
4.5.3.3	<a href="#">getDescriptor() override</a>	25

4.5.3.4	<a href="#">run() override</a>	25
4.5.3.5	<a href="#">testError()</a>	25
4.5.4	<a href="#">Member Data Documentation</a>	25
4.5.4.1	<a href="#">CS</a>	25
4.6	<a href="#">Error Class Reference</a>	25
4.6.1	<a href="#">Detailed Description</a>	26
4.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	26
4.6.2.1	<a href="#">Error(int errt)</a>	26
4.6.2.2	<a href="#">Error(unsigned bit, int errt)</a>	26
4.6.3	<a href="#">Member Function Documentation</a>	26
4.6.3.1	<a href="#">getError()</a>	26
4.6.3.2	<a href="#">runError(Code *code)</a>	26
4.6.3.3	<a href="#">runErrorOneBit(Code *code, unsigned index)</a>	27
4.6.3.4	<a href="#">setError(unsigned x, unsigned y, unsigned z)</a>	27
4.6.3.5	<a href="#">setError(double adcg)</a>	27
4.6.4	<a href="#">Member Data Documentation</a>	27
4.6.4.1	<a href="#">ADC</a>	27
4.6.4.2	<a href="#">CONST</a>	27
4.6.4.3	<a href="#">RAND</a>	28
4.7	<a href="#">Network Class Reference</a>	28
4.7.1	<a href="#">Detailed Description</a>	29
4.7.2	<a href="#">Constructor &amp; Destructor Documentation</a>	29
4.7.2.1	<a href="#">Network(int type, unsigned maxnumthreads, std::string filename, int n)</a>	29
4.7.2.2	<a href="#">~Network()</a>	29
4.7.3	<a href="#">Member Function Documentation</a>	29
4.7.3.1	<a href="#">getResult() override</a>	29
4.7.3.2	<a href="#">inititalize() override</a>	29
4.7.3.3	<a href="#">runAll()</a>	30
4.8	<a href="#">Node Class Reference</a>	30
4.8.1	<a href="#">Detailed Description</a>	30

4.8.2	Constructor & Destructor Documentation . . . . .	30
4.8.2.1	Node(std::string n) . . . . .	30
4.8.3	Member Function Documentation . . . . .	30
4.8.3.1	getId() . . . . .	30
4.8.3.2	getName() . . . . .	31
4.9	None Class Reference . . . . .	31
4.9.1	Detailed Description . . . . .	33
4.9.2	Constructor & Destructor Documentation . . . . .	33
4.9.2.1	None(bool b) . . . . .	33
4.9.3	Member Function Documentation . . . . .	33
4.9.3.1	encode(bool b) override . . . . .	33
4.9.3.2	getCS() override . . . . .	33
4.9.3.3	getDescriptor() override . . . . .	34
4.9.3.4	run() override . . . . .	34
4.9.4	Member Data Documentation . . . . .	34
4.9.4.1	CS . . . . .	34
4.10	Runnable Class Reference . . . . .	34
4.10.1	Detailed Description . . . . .	36
4.10.2	Constructor & Destructor Documentation . . . . .	36
4.10.2.1	Runnable(int type, unsigned maxnumthreads) . . . . .	36
4.10.2.2	~Runnable() . . . . .	36
4.10.3	Member Function Documentation . . . . .	36
4.10.3.1	getCodeType() . . . . .	36
4.10.3.2	getResult()=0 . . . . .	36
4.10.3.3	inititalize()=0 . . . . .	37
4.10.3.4	run() . . . . .	37
4.10.3.5	setCodeType(int t) . . . . .	37
4.10.4	Member Data Documentation . . . . .	37
4.10.4.1	AAD4 . . . . .	37
4.10.4.2	BITFLIP . . . . .	37

4.10.4.3	CODE5	37
4.10.4.4	NONE	37
4.10.4.5	runner	37
4.10.4.6	SHOR	38
4.10.4.7	STEANE	38
4.11	Runner Class Reference	38
4.11.1	Detailed Description	38
4.11.2	Constructor & Destructor Documentation	38
4.11.2.1	Runner(unsigned maxnumthreads)	38
4.11.2.2	~Runner()	38
4.11.3	Member Function Documentation	39
4.11.3.1	addCode(Code *c)	39
4.11.3.2	getBER()	39
4.11.3.3	reset()	39
4.11.3.4	run()	39
4.12	Shor Class Reference	39
4.12.1	Detailed Description	41
4.12.2	Member Typedef Documentation	41
4.12.2.1	super	41
4.12.3	Constructor & Destructor Documentation	41
4.12.3.1	Shor(bool b)	41
4.12.4	Member Function Documentation	41
4.12.4.1	encode(bool b) override	41
4.12.4.2	getCS() override	41
4.12.4.3	getDescriptor() override	42
4.12.4.4	run() override	42
4.12.5	Member Data Documentation	42
4.12.5.1	CS	42
4.13	Single Class Reference	43
4.13.1	Detailed Description	43



4.13.2	Constructor & Destructor Documentation . . . . .	44
4.13.2.1	Single(int type, int maxnumthreads, Error *e, int n) . . . . .	44
4.13.3	Member Function Documentation . . . . .	45
4.13.3.1	getResult() override . . . . .	45
4.13.3.2	italize() override . . . . .	45
4.14	Steane Class Reference . . . . .	45
4.14.1	Detailed Description . . . . .	47
4.14.2	Constructor & Destructor Documentation . . . . .	47
4.14.2.1	Steane(bool b) . . . . .	47
4.14.3	Member Function Documentation . . . . .	47
4.14.3.1	encode(bool b) override . . . . .	47
4.14.3.2	getCS() override . . . . .	47
4.14.3.3	getDescriptor() override . . . . .	47
4.14.3.4	run() override . . . . .	48
4.14.4	Member Data Documentation . . . . .	48
4.14.4.1	CS . . . . .	48
4.15	Test Class Reference . . . . .	48
4.15.1	Detailed Description . . . . .	50
4.15.2	Constructor & Destructor Documentation . . . . .	50
4.15.2.1	Test(int type, int maxnumthreads) . . . . .	50
4.15.3	Member Function Documentation . . . . .	50
4.15.3.1	bitFlipTest(bool input) . . . . .	50
4.15.3.2	code5Test(bool input) . . . . .	50
4.15.3.3	getResult() override . . . . .	50
4.15.3.4	italize() override . . . . .	51
4.15.3.5	runAllTests() . . . . .	51
4.15.3.6	shorTest(bool input) . . . . .	51
4.15.3.7	steaneTest(bool input) . . . . .	51
4.15.3.8	testMixed() . . . . .	51
4.15.4	Member Data Documentation . . . . .	51
4.15.4.1	bitflipruns . . . . .	51
4.15.4.2	code5runs . . . . .	52
4.15.4.3	shorruns . . . . .	52
4.15.4.4	steaneruns . . . . .	52

<b>5 File Documentation</b>	<b>53</b>
5.1 /home/attila/src/qecc-netbeans/qecc/.dep.inc File Reference	53
5.2 /home/attila/src/qecc-netbeans/qecc/Aad4.cpp File Reference	53
5.3 /home/attila/src/qecc-netbeans/qecc/Aad4.h File Reference	53
5.4 /home/attila/src/qecc-netbeans/qecc/BitFlip.cpp File Reference	53
5.5 /home/attila/src/qecc-netbeans/qecc/BitFlip.h File Reference	53
5.6 /home/attila/src/qecc-netbeans/qecc/Channel.cpp File Reference	54
5.7 /home/attila/src/qecc-netbeans/qecc/Channel.h File Reference	54
5.8 /home/attila/src/qecc-netbeans/qecc/Code.cpp File Reference	54
5.9 /home/attila/src/qecc-netbeans/qecc/Code.h File Reference	54
5.10 /home/attila/src/qecc-netbeans/qecc/Code5.cpp File Reference	54
5.11 /home/attila/src/qecc-netbeans/qecc/Code5.h File Reference	55
5.12 /home/attila/src/qecc-netbeans/qecc/Error.cpp File Reference	55
5.13 /home/attila/src/qecc-netbeans/qecc/Error.h File Reference	55
5.14 /home/attila/src/qecc-netbeans/qecc/Network.cpp File Reference	55
5.15 /home/attila/src/qecc-netbeans/qecc/Network.h File Reference	55
5.16 /home/attila/src/qecc-netbeans/qecc/Node.cpp File Reference	56
5.17 /home/attila/src/qecc-netbeans/qecc/Node.h File Reference	56
5.18 /home/attila/src/qecc-netbeans/qecc/None.cpp File Reference	56
5.19 /home/attila/src/qecc-netbeans/qecc/None.h File Reference	56
5.20 /home/attila/src/qecc-netbeans/qecc/qecc.cpp File Reference	56
5.20.1 Function Documentation	57
5.20.1.1 main(int argc, char **argv)	57
5.20.1.2 printHelp()	57
5.20.2 Variable Documentation	57
5.20.2.1 errRun	57
5.20.2.2 numthreads	57
5.21 /home/attila/src/qecc-netbeans/qecc/Runnable.cpp File Reference	57
5.22 /home/attila/src/qecc-netbeans/qecc/Runnable.h File Reference	57
5.23 /home/attila/src/qecc-netbeans/qecc/Runner.cpp File Reference	58
5.23.1 Function Documentation	58
5.23.1.1 runCode(void *c)	58
5.24 /home/attila/src/qecc-netbeans/qecc/Runner.h File Reference	58
5.24.1 Function Documentation	58
5.24.1.1 runCode(void *c)	58
5.25 /home/attila/src/qecc-netbeans/qecc/Shor.cpp File Reference	59
5.26 /home/attila/src/qecc-netbeans/qecc/Shor.h File Reference	59
5.27 /home/attila/src/qecc-netbeans/qecc/Single.cpp File Reference	59
5.28 /home/attila/src/qecc-netbeans/qecc/Single.h File Reference	59
5.29 /home/attila/src/qecc-netbeans/qecc/Steane.cpp File Reference	60
5.30 /home/attila/src/qecc-netbeans/qecc/Steane.h File Reference	60
5.31 /home/attila/src/qecc-netbeans/qecc/Test.cpp File Reference	60
5.32 /home/attila/src/qecc-netbeans/qecc/Test.h File Reference	60

<a href="#">Index</a>	61
-----------------------	----



# Chapter 1

## qecc

Quantum error correcting code simulator



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Channel . . . . .	13
Code . . . . .	16
Aad4 . . . . .	7
BitFlip . . . . .	10
Code5 . . . . .	22
None . . . . .	31
Shor . . . . .	39
Steane . . . . .	45
Error . . . . .	25
Node . . . . .	30
Runnable . . . . .	34
Network . . . . .	28
Single . . . . .	43
Test . . . . .	48
Runner . . . . .	38





## Chapter 3

# Class Index

### 3.1 Class List

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

Aad4	7
BitFlip	10
Channel	13
Code	16
Code5	22
Error	25
Network	28
Node	30
None	31
Runnable	34
Runner	38
Shor	39
Single	43
Steane	45
Test	48



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

/home/attila/src/qecc-netbeans/qecc/.dep.inc	53
/home/attila/src/qecc-netbeans/qecc/Aad4.cpp	53
/home/attila/src/qecc-netbeans/qecc/Aad4.h	53
/home/attila/src/qecc-netbeans/qecc/BitFlip.cpp	53
/home/attila/src/qecc-netbeans/qecc/BitFlip.h	53
/home/attila/src/qecc-netbeans/qecc/Channel.cpp	54
/home/attila/src/qecc-netbeans/qecc/Channel.h	54
/home/attila/src/qecc-netbeans/qecc/Code.cpp	54
/home/attila/src/qecc-netbeans/qecc/Code.h	54
/home/attila/src/qecc-netbeans/qecc/Code5.cpp	54
/home/attila/src/qecc-netbeans/qecc/Code5.h	55
/home/attila/src/qecc-netbeans/qecc/Error.cpp	55
/home/attila/src/qecc-netbeans/qecc/Error.h	55
/home/attila/src/qecc-netbeans/qecc/Network.cpp	55
/home/attila/src/qecc-netbeans/qecc/Network.h	55
/home/attila/src/qecc-netbeans/qecc/Node.cpp	56
/home/attila/src/qecc-netbeans/qecc/Node.h	56
/home/attila/src/qecc-netbeans/qecc/None.cpp	56
/home/attila/src/qecc-netbeans/qecc/None.h	56
/home/attila/src/qecc-netbeans/qecc/qecc.cpp	56
/home/attila/src/qecc-netbeans/qecc/Runnable.cpp	57
/home/attila/src/qecc-netbeans/qecc/Runnable.h	57
/home/attila/src/qecc-netbeans/qecc/Runner.cpp	58
/home/attila/src/qecc-netbeans/qecc/Runner.h	58
/home/attila/src/qecc-netbeans/qecc/Shor.cpp	59
/home/attila/src/qecc-netbeans/qecc/Shor.h	59
/home/attila/src/qecc-netbeans/qecc/Single.cpp	59
/home/attila/src/qecc-netbeans/qecc/Single.h	59
/home/attila/src/qecc-netbeans/qecc/Steane.cpp	60
/home/attila/src/qecc-netbeans/qecc/Steane.h	60
/home/attila/src/qecc-netbeans/qecc/Test.cpp	60
/home/attila/src/qecc-netbeans/qecc/Test.h	60



## Chapter 5

# Class Documentation

### 5.1 Aad4 Class Reference

```
#include <Aad4.h>
```

Inheritance diagram for Aad4:



## Public Member Functions

- [Aad4](#) (bool b)
- bool [run](#) () override
- std::string [getDescriptor](#) () override

## Static Public Attributes

- static const unsigned [CS](#) = 4

## Protected Member Functions

- void [encode](#) (bool b) override
- void [decode](#) ()
- unsigned [getCS](#) () override
- bool [noErrorOp](#) ()
- bool [errorOp](#) (bool \*f)
- bool \* [setandmes2Anc](#) ()

## Additional Inherited Members

### 5.1.1 Constructor & Destructor Documentation

5.1.1.1 `Aad4::Aad4 ( bool b ) [inline]`

### 5.1.2 Member Function Documentation

5.1.2.1 `void Aad4::decode ( ) [protected]`

5.1.2.2 `void Aad4::encode ( bool b ) [override],[protected],[virtual]`

Encode input bit

Parameters

<i>b</i>	Classical input converted to $ 0\rangle$ or $ 1\rangle$
----------	---

Implements [Code](#).

5.1.2.3 `bool Aad4::errorOp ( bool * f ) [protected]`

5.1.2.4 `unsigned Aad4::getCS ( ) [inline],[override],[protected],[virtual]`

Return code size of a child

Returns

code size

Implements [Code](#).

5.1.2.5 `std::string Aad4::getDescriptor ( ) [override],[virtual]`

Log of run

To be defined in child class

Contains code type, input, output, ancilla bit measurements

Format:

*CODETYPE ancillas input->result OK/ERROR*

Returns

log string

Implements [Code](#).

5.1.2.6 `bool Aad4::noErrorOp ( ) [protected]`

5.1.2.7 `bool Aad4::run ( ) [override],[virtual]`

Function to simulate the transmission and get the result. To be defined in child classes.

Returns

Measured bit at Bob

Implements [Code](#).

5.1.2.8 `bool * Aad4::setandmes2Anc ( ) [protected]`

### 5.1.3 Member Data Documentation

5.1.3.1 `const unsigned Aad4::CS = 4 [static]`

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

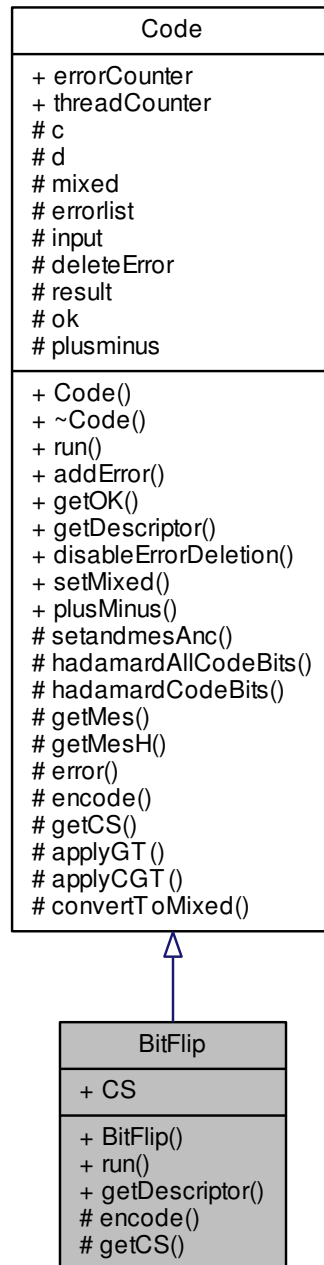
- `/home/attila/src/qecc-netbeans/qecc/Aad4.h`
- `/home/attila/src/qecc-netbeans/qecc/Aad4.cpp`



## 5.2 BitFlip Class Reference

```
#include <BitFlip.h>
```

Inheritance diagram for BitFlip:



### Public Member Functions

- [BitFlip](#) (bool b)
- bool [run](#) () override
- std::string [getDescriptor](#) () override

## Static Public Attributes

- static const unsigned `CS` = 3

## Protected Member Functions

- void `encode` (bool b) override
- unsigned `getCS` () override

## Additional Inherited Members

### 5.2.1 Detailed Description

Bit Flip code class

### 5.2.2 Constructor & Destructor Documentation

5.2.2.1 `BitFlip::BitFlip ( bool b )` `[inline]`

### 5.2.3 Member Function Documentation

5.2.3.1 `void BitFlip::encode ( bool b )` `[override]`, `[protected]`, `[virtual]`

Encode input bit

#### Parameters

<code>b</code>	Classical input converted to $ 0\rangle$ or $ 1\rangle$
----------------	---

Implements [Code](#).

5.2.3.2 `unsigned BitFlip::getCS ( )` `[inline]`, `[override]`, `[protected]`, `[virtual]`

Return code size of a child

#### Returns

code size

Implements [Code](#).

### 5.2.3.3 `std::string BitFlip::getDescriptor ( ) [override],[virtual]`

Returns descriptor string

Format:

BITFLIP aa i->o OK/ERROR

Where aa is flip array

Returns

Log string

Implements [Code](#).

### 5.2.3.4 `bool BitFlip::run ( ) [override],[virtual]`

Function to simulate the transmission and get the result. To be defined in child classes.

Returns

Measured bit at Bob

Implements [Code](#).

## 5.2.4 Member Data Documentation

### 5.2.4.1 `const unsigned BitFlip::CS = 3 [static]`

[Code](#) size for bit flip which is 3

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

- [/home/attila/src/qecc-netbeans/qecc/BitFlip.h](#)
- [/home/attila/src/qecc-netbeans/qecc/BitFlip.cpp](#)

## 5.3 Channel Class Reference

```
#include <Channel.h>
```

### Public Member Functions

- [Channel](#) (int nodeid0, int nodeid1, [Error](#) \*e)
- bool [matchId](#) (int nid0, int nid1)
- int [matchOneGetOther](#) (int id)
- int \* [getIds](#) ()
- std::string [debugPrint](#) ()
- [Error](#) \* [getError](#) ()
- [~Channel](#) ()

### 5.3.1 Detailed Description

Used by network mode

Stores two node ids and an error between them

### 5.3.2 Constructor & Destructor Documentation

5.3.2.1 `Channel::Channel ( int nodeid0, int nodeid1, Error * e )` `[inline]`

Constructor

## Parameters

<i>nodeid0</i>	Id of first node
<i>nodeid1</i>	Id of second node
<i>e</i>	<a href="#">Error</a> of the channel

5.3.2.2 `Channel::~~Channel( )` `[inline]`

### 5.3.3 Member Function Documentation

5.3.3.1 `std::string Channel::debugPrint( )` `[inline]`

Print data to string

## Returns

String which describes the instance

5.3.3.2 `Error* Channel::getError( )` `[inline]`

Get [Error](#)

## Returns

returns pointer to error

5.3.3.3 `int* Channel::getIds( )` `[inline]`

Get node ids

## Returns

Array of our the nodeids

5.3.3.4 `bool Channel::matchId( int nid0, int nid1 )` `[inline]`

Check if two ids given is what we have, order does not matter

## Parameters

<i>nid0</i>	First node id
<i>nid1</i>	Second node id

**Returns**

True if our ids match the input

**5.3.3.5** `int Channel::matchOneGetOther ( int id )` `[inline]`

Check if one of our ids is same as input and return the other or zero if no match

**Parameters**

<i>id</i>	<a href="#">Node id</a>
-----------	-------------------------

**Returns**

The id of our other node on match

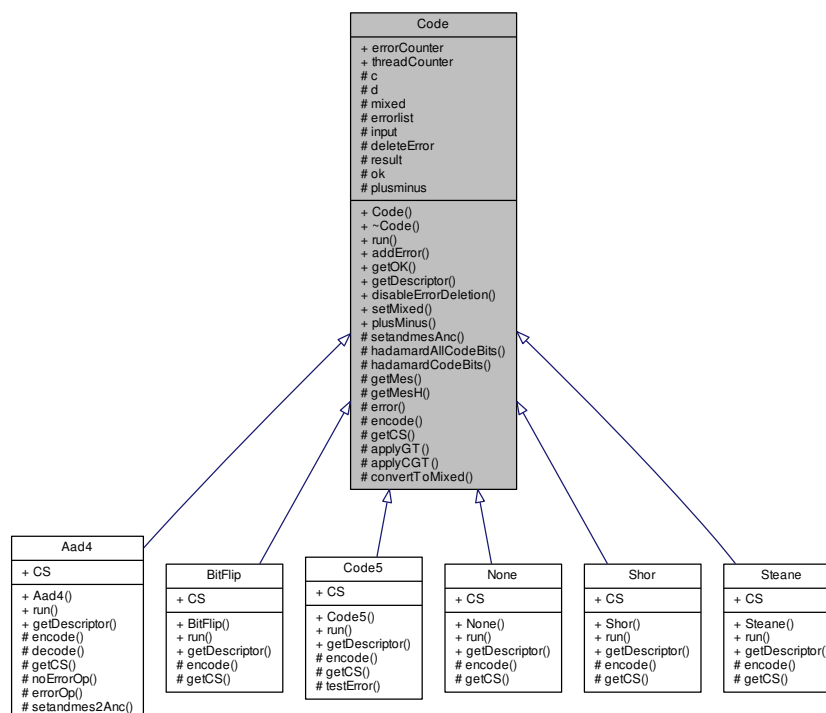
The documentation for this class was generated from the following file:

- [/home/attila/src/qecc-netbeans/qecc/Channel.h](#)

**5.4 Code Class Reference**

```
#include <Code.h>
```

Inheritance diagram for Code:



## Public Member Functions

- [Code](#) (bool b)
- virtual [~Code](#) ()
- virtual bool [run](#) ()=0
- void [addError](#) ([Error](#) \*e)
- bool [getOK](#) ()
- virtual std::string [getDescriptor](#) ()=0
- void [disableErrorDeletion](#) ()
- void [setMixed](#) ()
- void [plusMinus](#) ()

## Public Attributes

- std::atomic\_int \* [errorCounter](#)
- std::atomic\_int \* [threadCounter](#)

## Protected Member Functions

- bool [setandmesAnc](#) (const std::vector< unsigned > &b, unsigned CS)
- void [hadamardAllCodeBits](#) ()
- void [hadamardCodeBits](#) (const std::vector< unsigned > &b)
- bool [getMes](#) (unsigned i)
- bool [getMesH](#) (unsigned i)
- void [error](#) ()
- virtual void [encode](#) (bool b)=0
- virtual unsigned [getCS](#) ()=0
- void [applyGT](#) (const cmat gate, unsigned b)
- void [applyCGT](#) (const cmat gate, const std::vector< unsigned > &b)
- void [convertToMixed](#) ()

## Protected Attributes

- ket \* [c](#)
- cmat \* [d](#)
- bool [mixed](#)
- std::queue< [Error](#) \* > [errorlist](#)
- bool [input](#)
- bool [deleteError](#)
- bool [result](#)
- bool [ok](#)
- bool [plusminus](#)

## Friends

- class [Error](#)

### 5.4.1 Detailed Description

[Code](#) class provides a way to simulate a code on an erroneous channel  
This is the parent of all specific code classes

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 [Code::Code](#) ( bool b )

Constructor

## Parameters

<i>b</i>	Input bit 1 or 0
----------	------------------

5.4.2.2 `Code::~~Code ( )` [virtual]

Destructor

### 5.4.3 Member Function Documentation

5.4.3.1 `void Code::addError ( Error * e )` [inline]

Add an error to the error list

## Parameters

<i>e</i>	Pointer to an error to be added
----------	---------------------------------

5.4.3.2 `void Code::applyCGT ( const cmat gate, const std::vector< unsigned > & b )` [inline], [protected]

Applies a 2 qubit gate

## Parameters

<i>gate</i>	Matrix of the gate
<i>b</i>	array of two subsystem ids

5.4.3.3 `void Code::applyGT ( const cmat gate, unsigned b )` [inline], [protected]

Applies a single qubit gate

## Parameters

<i>gate</i>	Matrix of the gate
<i>b</i>	subsystem id

5.4.3.4 `void Code::convertToMixed ( )` [inline], [protected]

Convert vector notation to density matrix. **Must be called inside run!** From outside class stays the same but inside it will use density matrix. This has a performance drawback but needed for some errors for example amplitude damping. An amplitude damping [Error](#) does this conversion automatically.



#### 5.4.3.5 void Code::disableErrorDeletion ( ) [inline]

Normally Errors are deleted by run function. If the given [Error](#) instance is being used by multiple [Code](#) instances this will disable deletion of it .

#### 5.4.3.6 virtual void Code::encode ( bool b ) [protected], [pure virtual]

Encode input bit

##### Parameters

<i>b</i>	Classical input converted to $ 0\rangle$ or $ 1\rangle$
----------	---

Implemented in [Code5](#), [Shor](#), [None](#), [Steane](#), [BitFlip](#), and [Aad4](#).

#### 5.4.3.7 void Code::error ( ) [protected]

Applies errors on c or d

#### 5.4.3.8 virtual unsigned Code::getCS ( ) [protected], [pure virtual]

Return code size of a child

##### Returns

code size

Implemented in [Code5](#), [Shor](#), [None](#), [Steane](#), [BitFlip](#), and [Aad4](#).

#### 5.4.3.9 virtual std::string Code::getDescriptor ( ) [pure virtual]

Log of run

To be defined in child class

Contains code type, input, output, ancilla bit measurements

Format:

*CODETYPE ancillas input->result OK/ERROR*

##### Returns

log string

Implemented in [Code5](#), [Shor](#), [None](#), [Steane](#), [BitFlip](#), and [Aad4](#).

#### 5.4.3.10 bool Code::getMes ( unsigned i ) [protected]

Measures bit

**Parameters**

<i>i</i>	Subsystem id
----------	--------------

**Returns**

Measured classical value

**5.4.3.11** `bool Code::getMesH ( unsigned i )` `[protected]`

**5.4.3.12** `bool Code::getOK ( )` `[inline]`

Check whether transmission was successful

**Returns**

True if input is same as result, false if otherwise

**5.4.3.13** `void Code::hadamardAllCodeBits ( )` `[protected]`

Hadamard transforms all codebits

**5.4.3.14** `void Code::hadamardCodeBits ( const std::vector< unsigned > & b )` `[protected]`

Hadamard transforms a set of codebits

**Parameters**

<i>b</i>	Bits to transform
----------	-------------------

**5.4.3.15** `void Code::plusMinus ( )` `[inline]`

Run code in ket+, ket- basis

**5.4.3.16** `virtual bool Code::run ( )` `[pure virtual]`

Function to simulate the transmission and get the result. To be defined in child classes.

**Returns**

Measured bit at Bob

Implemented in [Code5](#), [Shor](#), [None](#), [Steane](#), [BitFlip](#), and [Aad4](#).

5.4.3.17 `bool Code::setandmesAnc ( const std::vector< unsigned > & b, unsigned CS )` [protected]

Applies CNOT gates controlled by bits in *b* on ancilla one-by-one. Than measures ancilla and returns its classical value.

#### Parameters

<i>b</i>	Bits controlling ancilla
<i>CS</i>	<a href="#">Code</a> size

#### Returns

Classical value of ancilla

5.4.3.18 `void Code::setMixed ( )` [inline]

Converts vector notation to density matrix. **Must be called before run!** Useful for testing.

## 5.4.4 Friends And Related Function Documentation

5.4.4.1 `friend class Error` [friend]

## 5.4.5 Member Data Documentation

5.4.5.1 `ket* Code::c` [protected]

Contains state in vector notation

5.4.5.2 `cmat* Code::d` [protected]

Contains state in density matrix

5.4.5.3 `bool Code::deleteError` [protected]

Delete Errors on run. True by default.

5.4.5.4 `std::atomic_int* Code::errorCounter`

External error counter provided by [Runner](#) class. If an error occurs this is incremented.

5.4.5.5 `std::queue<Error *> Code::errorlist` [protected]

List of errors to be applied

#### 5.4.5.6 `bool Code::input` [protected]

Input bit, converted to  $|0\rangle$  or  $|1\rangle$

#### 5.4.5.7 `bool Code::mixed` [protected]

True if density matrix is used

#### 5.4.5.8 `bool Code::ok` [protected]

True if result is same as input, false otherwise

#### 5.4.5.9 `bool Code::plusminus` [protected]

Run code in ket+, ket- basis

#### 5.4.5.10 `bool Code::result` [protected]

Measured bit, being set by run

#### 5.4.5.11 `std::atomic_int* Code::threadCounter`

Thread counter of [Runner](#) class. After run [Runner](#) decrements this.

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

- [/home/attila/src/qecc-netbeans/qecc/Code.h](#)
- [/home/attila/src/qecc-netbeans/qecc/Code.cpp](#)

## 5.5 Code5 Class Reference

```
#include <Code5.h>
```

Inheritance diagram for Code5:



## Public Member Functions

- [Code5](#) (bool b)
- bool [run](#) () override
- std::string [getDescriptor](#) () override

## Static Public Attributes

- static const unsigned [CS](#) = 5

## Protected Member Functions

- void [encode](#) (bool b) override
- unsigned [getCS](#) () override
- void [testError](#) ()

## Additional Inherited Members

### 5.5.1 Detailed Description

5Qubit code class

### 5.5.2 Constructor & Destructor Documentation

5.5.2.1 `Code5::Code5 ( bool b )` `[inline]`

### 5.5.3 Member Function Documentation

5.5.3.1 `void Code5::encode ( bool b )` `[override]`, `[protected]`, `[virtual]`

Encode 5qubit code

#### Parameters

<i>b</i>	Input bit
----------	-----------

Implements [Code](#).

5.5.3.2 `unsigned Code5::getCS ( )` `[inline]`, `[override]`, `[protected]`, `[virtual]`

Return code size of a child

#### Returns

code size

Implements [Code](#).

5.5.3.3 `std::string Code5::getDescriptor ( ) [override],[virtual]`

Returns descriptor string **Format: 5QUBIT aaaa i->o OK/ERROR** Where aaaa is flip array

Returns

**Log string**

Implements [Code](#).

5.5.3.4 `bool Code5::run ( ) [override],[virtual]`

Function to simulate the transmission and get the result. To be defined in child classes.

Returns

Measured bit at Bob

Implements [Code](#).

5.5.3.5 `void Code5::testError ( ) [protected]`

## 5.5.4 Member Data Documentation

5.5.4.1 `const unsigned Code5::CS = 5 [static]`

[Code](#) size for 5Qubit is 5

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

- `/home/attila/src/qecc-netbeans/qecc/Code5.h`
- `/home/attila/src/qecc-netbeans/qecc/Code5.cpp`

## 5.6 Error Class Reference

```
#include <Error.h>
```

### Public Member Functions

- [Error](#) (int errt)
- [Error](#) (unsigned bit, int errt)
- void [runError](#) ([Code](#) \*code)
- void [runErrorOneBit](#) ([Code](#) \*code, unsigned index)
- void [setError](#) (unsigned x, unsigned y, unsigned z)
- std::string [getError](#) ()
- void [setError](#) (double adcg)

## Static Public Attributes

- static const int [CONST](#) = 42
- static const int [RAND](#) = 43
- static const int [ADC](#) = 44

### 5.6.1 Detailed Description

Class to provide errors, friend of [Code](#)

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 `Error::Error ( int errt ) [inline]`

Constructor, runs on all bits

##### Parameters

<i>errt</i>	Type
-------------	------

#### 5.6.2.2 `Error::Error ( unsigned bit, int errt ) [inline]`

Constructor

##### Parameters

<i>bit</i>	Subsystem id to run on
<i>errt</i>	Type

### 5.6.3 Member Function Documentation

#### 5.6.3.1 `std::string Error::getError ( ) [inline]`

Returns descriptor string

##### Returns

Descriptor

#### 5.6.3.2 `void Error::runError ( Code * code )`

Run error on code



## Parameters

<i>code</i>	<a href="#">Code</a> to run on
-------------	--------------------------------

5.6.3.3 void Error::runErrorOneBit ( [Code](#) \* *code*, unsigned *index* )

Run one bit error on code

## Parameters

<i>code</i>	<a href="#">Code</a> instance
<i>index</i>	Bit to run on

5.6.3.4 void Error::setError ( unsigned *x*, unsigned *y*, unsigned *z* ) [\[inline\]](#)

Sets X, Y, Z effective if CONST or RAND

## Parameters

<i>x</i>	X error
<i>y</i>	Y error
<i>z</i>	Z error

5.6.3.5 void Error::setError ( double *adcg* ) [\[inline\]](#)

Sets gamma for amplitude damping effective if ADC

## Parameters

<i>adcg</i>	Gamma
-------------	-------

## 5.6.4 Member Data Documentation

5.6.4.1 const int Error::ADC = 44 [\[static\]](#)

ADC type int

5.6.4.2 const int Error::CONST = 42 [\[static\]](#)

CONST type int

5.6.4.3 `const int Error::RAND = 43` `[static]`

RAND type int

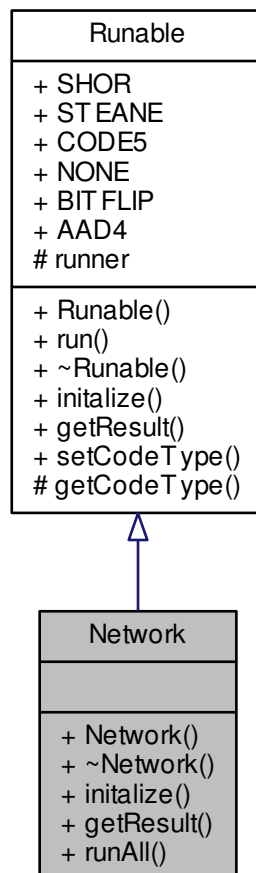
The documentation for this class was generated from the following files:

- [/home/attila/src/qecc-netbeans/qecc/Error.h](#)
- [/home/attila/src/qecc-netbeans/qecc/Error.cpp](#)

## 5.7 Network Class Reference

```
#include <Network.h>
```

Inheritance diagram for Network:



## Public Member Functions

- [Network](#) (int type, unsigned maxnumthreads, std::string filename, int n, std::string x, bool p)
- virtual [~Network](#) ()
- void [italize](#) () override
- std::string [getResult](#) () override
- void [runAll](#) ()

## Additional Inherited Members

### 5.7.1 Detailed Description

Provides parser and [Runner](#) for network mode

### 5.7.2 Constructor & Destructor Documentation

**5.7.2.1** `Network::Network ( int type, unsigned maxnumthreads, std::string filename, int n, std::string x, bool p )`  
`[inline]`

Constructor

Parameters

<i>type</i>	<a href="#">Code</a> type
<i>maxnumthreads</i>	Maximum number of threads running
<i>filename</i>	Input file name
<i>n</i>	Number of times to run one transmission

**5.7.2.2** `Network::~~Network ( )` `[virtual]`

### 5.7.3 Member Function Documentation

**5.7.3.1** `string Network::getResult ( )` `[override],[virtual]`

Returns result string for one transmission

Returns

Descriptor string

Implements [Runnable](#).

**5.7.3.2** `void Network::italize ( )` `[override],[virtual]`

Initializes [Runner](#), using get [Channel](#) for one pair. **currentPair must be set before!**

Implements [Runnable](#).

### 5.7.3.3 void Network::runAll ( )

Runs simulation for each pair of nodes

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

- /home/attila/src/qecc-netbeans/qecc/[Network.h](#)
- /home/attila/src/qecc-netbeans/qecc/[Network.cpp](#)

## 5.8 Node Class Reference

```
#include <Node.h>
```

### Public Member Functions

- [Node](#) (std::string n)
- int [getId](#) ()
- std::string [getName](#) ()
- void [makeDummy](#) ()
- bool [isDummy](#) ()

### 5.8.1 Detailed Description

Assigns a unique id to a name

### 5.8.2 Constructor & Destructor Documentation

#### 5.8.2.1 Node::Node ( std::string *n* ) `[inline]`

Constructor

Parameters

<i>n</i>	Name of the node
----------	------------------

### 5.8.3 Member Function Documentation

#### 5.8.3.1 int Node::getId ( ) `[inline]`

Returns Id

Returns

id

#### 5.8.3.2 `std::string Node::getName ( )` `[inline]`

Return name

Returns

name

#### 5.8.3.3 `bool Node::isDummy ( )` `[inline]`

Returns true if this is a dummy node

#### 5.8.3.4 `void Node::makeDummy ( )` `[inline]`

Makes this node a dummy node

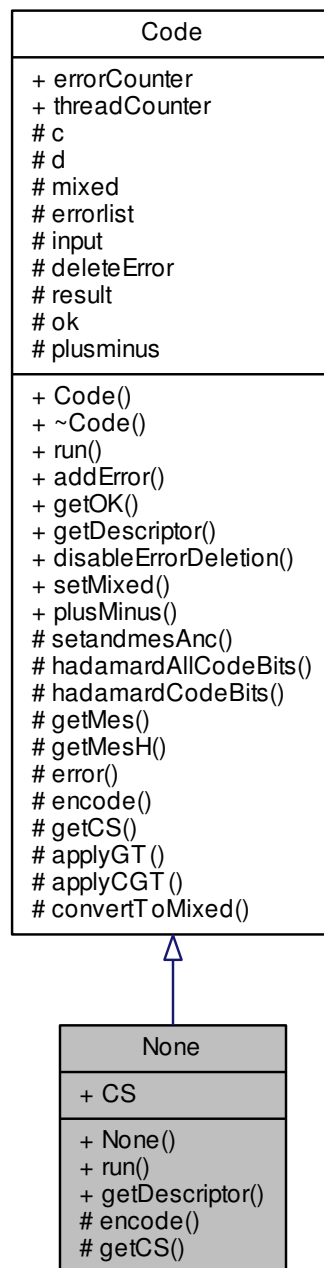
The documentation for this class was generated from the following files:

- [/home/attila/src/qecc-netbeans/qecc/Node.h](#)
- [/home/attila/src/qecc-netbeans/qecc/Node.cpp](#)

## 5.9 None Class Reference

```
#include <None.h>
```

Inheritance diagram for None:



## Public Member Functions

- [None](#) (bool b)
- bool [run](#) () override
- std::string [getDescriptor](#) () override

## Static Public Attributes

- static const unsigned [CS](#) = 1

## Protected Member Functions

- void [encode](#) (bool b) override
- unsigned [getCS](#) () override

## Additional Inherited Members

### 5.9.1 Detailed Description

[Code](#) class using no coding

### 5.9.2 Constructor & Destructor Documentation

5.9.2.1 `None::None ( bool b ) [inline]`

### 5.9.3 Member Function Documentation

5.9.3.1 `void None::encode ( bool b ) [override], [protected], [virtual]`

Encode input bit

#### Parameters

<i>b</i>	Classical input converted to $ 0\rangle$ or $ 1\rangle$
----------	---

Implements [Code](#).

5.9.3.2 `unsigned None::getCS ( ) [inline], [override], [protected], [virtual]`

Return code size of a child

#### Returns

code size

Implements [Code](#).

#### 5.9.3.3 `std::string None::getDescriptor ( )` `[override],[virtual]`

Returns a descriptor string

Format:

*NONE i->o ERROR/OK*

Returns

Log string

Implements [Code](#).

#### 5.9.3.4 `bool None::run ( )` `[override],[virtual]`

Function to simulate the transmission and get the result. To be defined in child classes.

Returns

Measured bit at Bob

Implements [Code](#).

### 5.9.4 Member Data Documentation

#### 5.9.4.1 `const unsigned None::CS = 1` `[static]`

[Code](#) size for [None](#) is 1

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

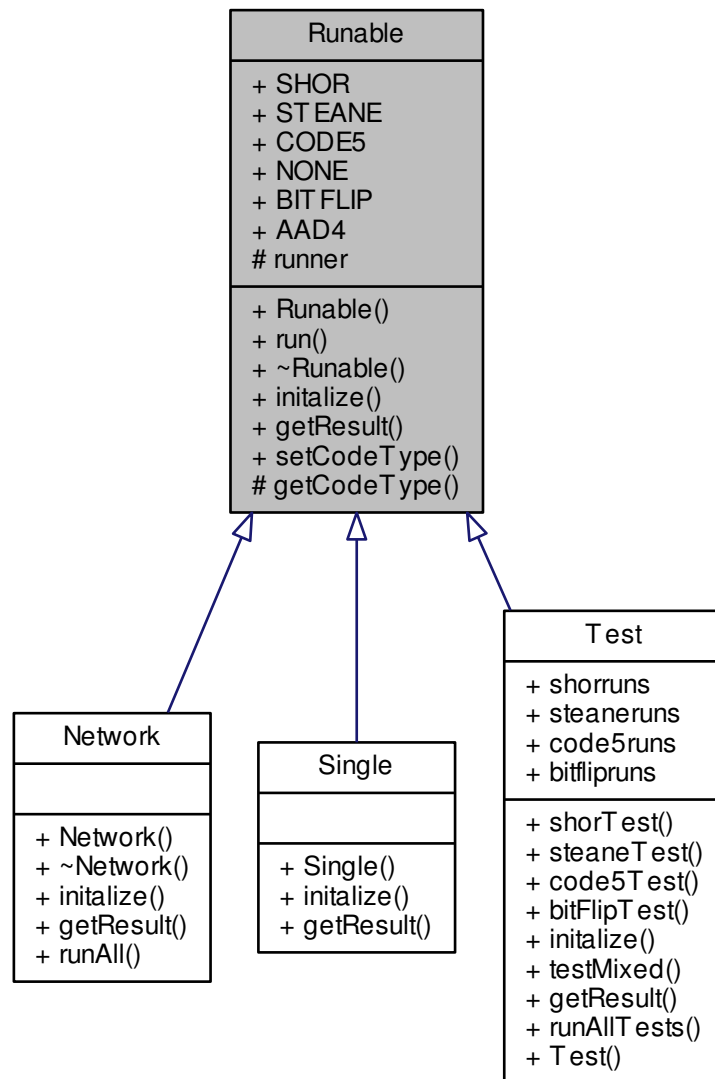
- [/home/attila/src/qecc-netbeans/qecc/None.h](#)
- [/home/attila/src/qecc-netbeans/qecc/None.cpp](#)

## 5.10 Runnable Class Reference

```
#include <Runnable.h>
```



Inheritance diagram for Runnable:



### Public Member Functions

- [Runnable](#) (int type, unsigned maxnumthreads, bool p)
- void [run](#) ()
- virtual [~Runnable](#) ()
- virtual void [initalize](#) ()=0
- virtual std::string [getResult](#) ()=0
- void [setCodeType](#) (int t)

### Static Public Attributes

- static const int [SHOR](#) = 100

- static const int [STEANE](#) = 101
- static const int [CODE5](#) = 102
- static const int [NONE](#) = 103
- static const int [BITFLIP](#) = 104
- static const int [AAD4](#) = 105

### Protected Member Functions

- int [getCodeType](#) ()

### Protected Attributes

- [Runner](#) \* [runner](#)

## 5.10.1 Detailed Description

[Runnable](#) provides container for [Runner](#)

## 5.10.2 Constructor & Destructor Documentation

5.10.2.1 [Runnable::Runnable](#) ( int *type*, unsigned *maxnumthreads*, bool *p* ) [inline]

Constructor

Parameters

<i>type</i>	<a href="#">Code</a> type
<i>maxnumthreads</i>	Maximum number of threads running

5.10.2.2 [Runnable::~~Runnable](#) ( ) [virtual]

## 5.10.3 Member Function Documentation

5.10.3.1 int [Runnable::getCodeType](#) ( ) [inline],[protected]

Returns

code type

5.10.3.2 virtual std::string [Runnable::getResult](#) ( ) [pure virtual]

Returns descriptor string. To be defined by child!

Returns

Log string

Implemented in [Test](#), [Network](#), and [Single](#).

#### 5.10.3.3 `virtual void Runnable::initialize ( ) [pure virtual]`

Initializes code list. To be defined by child!

Implemented in [Test](#), [Network](#), and [Single](#).

#### 5.10.3.4 `void Runnable::run ( ) [inline]`

Runs codes

#### 5.10.3.5 `void Runnable::setCodeType ( int t ) [inline]`

Sets code type to another

Parameters

<i>t</i>	code type
----------	-----------

### 5.10.4 Member Data Documentation

#### 5.10.4.1 `const int Runnable::AAD4 = 105 [static]`

4qubit amplitude damping code type int

#### 5.10.4.2 `const int Runnable::BITFLIP = 104 [static]`

Bit flip code type int

#### 5.10.4.3 `const int Runnable::CODE5 = 102 [static]`

5qubit code type int

#### 5.10.4.4 `const int Runnable::NONE = 103 [static]`

No coding type int

#### 5.10.4.5 `Runner* Runnable::runner [protected]`

[Runner](#) to provide multithreading

5.10.4.6 `const int Runnable::SHOR = 100` `[static]`

[Shor](#) code type int

5.10.4.7 `const int Runnable::STEANE = 101` `[static]`

[Steane](#) code type int

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

- [/home/attila/src/qecc-netbeans/qecc/Runnable.h](#)
- [/home/attila/src/qecc-netbeans/qecc/Runnable.cpp](#)

## 5.11 Runner Class Reference

```
#include <Runner.h>
```

### Public Member Functions

- [Runner](#) (unsigned maxnumthreads)
- virtual [~Runner](#) ()
- void [addCode](#) ([Code](#) \*c)
- void [run](#) ()
- void [reset](#) ()
- float [getBER](#) ()
- void [plusMinus](#) ()

### 5.11.1 Detailed Description

Runs codes multithreaded

### 5.11.2 Constructor & Destructor Documentation

5.11.2.1 `Runner::Runner ( unsigned maxnumthreads )`

Constructor

Parameters

<i>maxnumthreads</i>	Maximum number of thread running
----------------------	----------------------------------

5.11.2.2 `Runner::~~Runner ( )` `[virtual]`

### 5.11.3 Member Function Documentation

#### 5.11.3.1 void Runner::addCode ( Code \* c )

Adds code to code list

##### Parameters

c	Pointer to the code to be added
---	---------------------------------

#### 5.11.3.2 float Runner::getBER ( ) [inline]

Get bit error rate

##### Returns

Bit error rate

#### 5.11.3.3 void Runner::plusMinus ( ) [inline]

Run codes with ket+, ket- inputs

#### 5.11.3.4 void Runner::reset ( ) [inline]

Reset status normally run by [run\(\)](#)

#### 5.11.3.5 void Runner::run ( )

Runs codes in code list

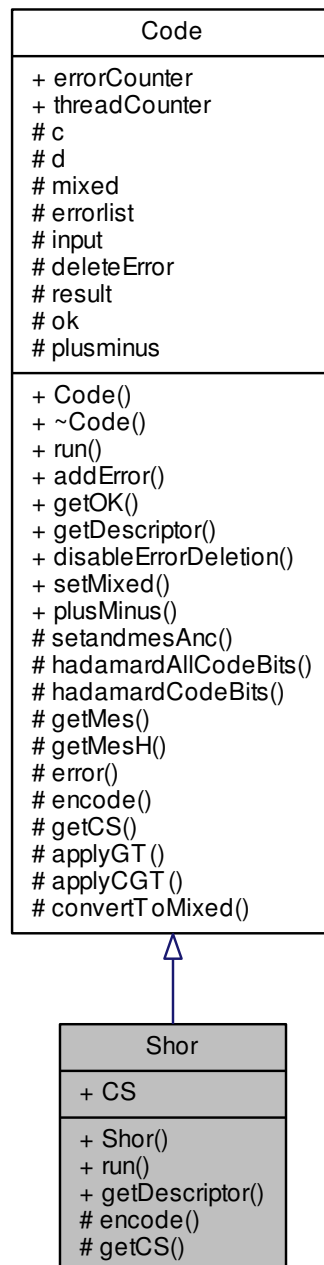
The documentation for this class was generated from the following files:

- [/home/attila/src/qecc-netbeans/qecc/Runner.h](#)
- [/home/attila/src/qecc-netbeans/qecc/Runner.cpp](#)

## 5.12 Shor Class Reference

```
#include <Shor.h>
```

Inheritance diagram for Shor:



### Public Types

- typedef [Code](#) super

## Public Member Functions

- [Shor](#) (bool b)
- bool [run](#) () override
- std::string [getDescriptor](#) () override

## Static Public Attributes

- static const unsigned [CS](#) = 9

## Protected Member Functions

- void [encode](#) (bool b) override
- unsigned [getCS](#) () override

## Additional Inherited Members

### 5.12.1 Detailed Description

[Shor](#) code class

### 5.12.2 Member Typedef Documentation

5.12.2.1 typedef Code [Shor::super](#)

### 5.12.3 Constructor & Destructor Documentation

5.12.3.1 [Shor::Shor](#) ( bool *b* ) [inline]

### 5.12.4 Member Function Documentation

5.12.4.1 void [Shor::encode](#) ( bool *b* ) [override],[protected],[virtual]

Encode input bit

#### Parameters

<i>b</i>	Classical input converted to $ 0\rangle$ or $ 1\rangle$
----------	---

Implements [Code](#).

5.12.4.2 unsigned [Shor::getCS](#) ( ) [inline],[override],[protected],[virtual]

Return code size of a child

**Returns**

code size

Implements [Code](#).

**5.12.4.3 `std::string Shor::getDescriptor ( )` `[override],[virtual]`**

Returns descriptor string

Format:

*SHOR aabbcc zz i->o OK/ERROR*

Where aabbcc is xflip, zz is zflip

**Returns**

Log string

Implements [Code](#).

**5.12.4.4 `bool Shor::run ( )` `[override],[virtual]`**

Function to simulate the transmission and get the result. To be defined in child classes.

**Returns**

Measured bit at Bob

Implements [Code](#).

**5.12.5 Member Data Documentation****5.12.5.1 `const unsigned Shor::CS = 9` `[static]`**

[Code](#) size for [Shor](#) is 9

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

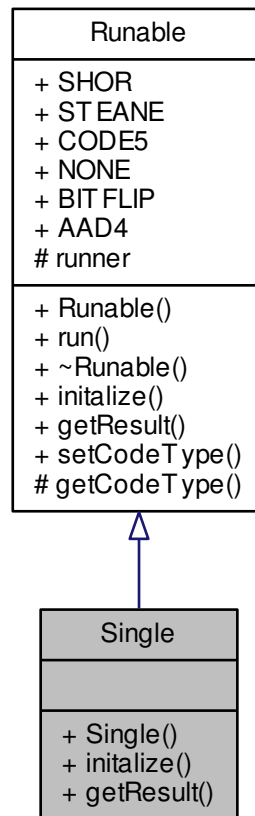
- [/home/attila/src/qecc-netbeans/qecc/Shor.h](#)
- [/home/attila/src/qecc-netbeans/qecc/Shor.cpp](#)



## 5.13 Single Class Reference

```
#include <Single.h>
```

Inheritance diagram for Single:



### Public Member Functions

- [Single](#) (int type, int maxnumthreads, [Error](#) \*e, int n, bool p)
- void [italize](#) () override
- std::string [getResult](#) () override

### Additional Inherited Members

#### 5.13.1 Detailed Description

[Single](#) class runs a single error channel n times

## 5.13.2 Constructor & Destructor Documentation

5.13.2.1 `Single::Single ( int type, int maxnumthreads, Error * e, int n, bool p )` `[inline]`

Constructor

## Parameters

<i>type</i>	<a href="#">Code</a> type
<i>maxnumthreads</i>	Maximum number of threads running
<i>e</i>	<a href="#">Error</a> to apply
<i>n</i>	Number of runs

### 5.13.3 Member Function Documentation

#### 5.13.3.1 `std::string Single::getResult ( ) [override],[virtual]`

Returns descriptor string. To be defined by child!

## Returns

Log string

Implements [Runnable](#).

#### 5.13.3.2 `void Single::inititalize ( ) [override],[virtual]`

Initializes code list. To be defined by child!

Implements [Runnable](#).

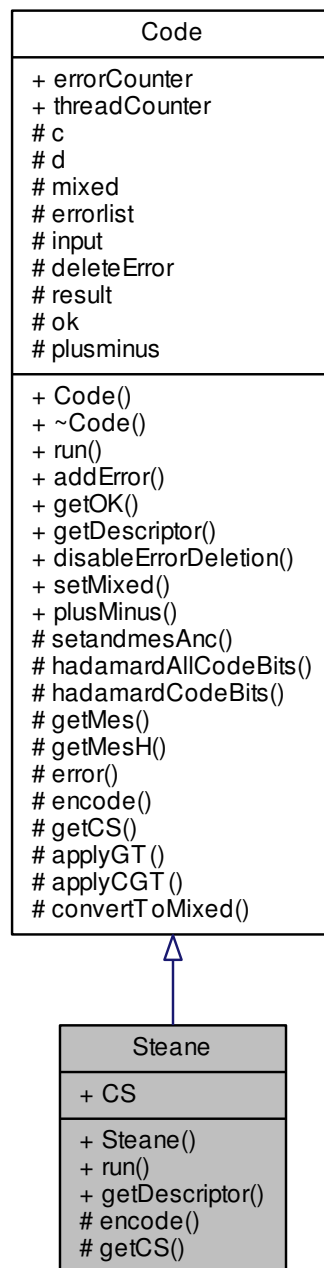
The documentation for this class was generated from the following files:

- `/home/attila/src/qecc-netbeans/qecc/Single.h`
- `/home/attila/src/qecc-netbeans/qecc/Single.cpp`

## 5.14 Steane Class Reference

```
#include <Steane.h>
```

Inheritance diagram for Steane:



## Public Member Functions

- [Steane](#) (bool b)
- bool [run](#) () override
- std::string [getDescriptor](#) () override

## Static Public Attributes

- static const int [CS](#) = 7

## Protected Member Functions

- void [encode](#) (bool b) override
- unsigned [getCS](#) () override

## Additional Inherited Members

### 5.14.1 Detailed Description

!Steane code class

### 5.14.2 Constructor & Destructor Documentation

5.14.2.1 `Steane::Steane ( bool b )` `[inline]`

### 5.14.3 Member Function Documentation

5.14.3.1 `void Steane::encode ( bool b )` `[override]`, `[protected]`, `[virtual]`

Encode input bit

#### Parameters

<i>b</i>	Classical input converted to $ 0\rangle$ or $ 1\rangle$
----------	---

Implements [Code](#).

5.14.3.2 `unsigned Steane::getCS ( )` `[inline]`, `[override]`, `[protected]`, `[virtual]`

Return code size of a child

#### Returns

code size

Implements [Code](#).

5.14.3.3 `std::string Steane::getDescriptor ( )` `[override]`, `[virtual]`

Returns a descriptor string

Format:

*STEANE xxx zzz i->o OK/ERROR*

Where xxx is xflip, zzz is zflip

Implements [Code](#).

#### 5.14.3.4 `bool Steane::run ( )` `[override],[virtual]`

Function to simulate the transmission and get the result. To be defined in child classes.

##### Returns

Measured bit at Bob

Implements [Code](#).

### 5.14.4 Member Data Documentation

#### 5.14.4.1 `const int Steane::CS = 7` `[static]`

[Code](#) size for steane is 7

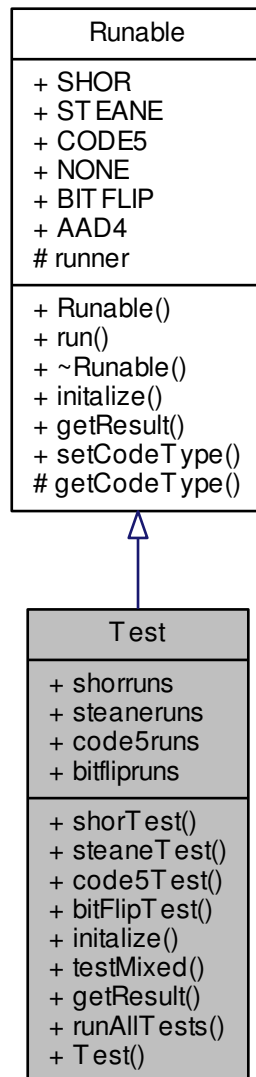
The documentation for this class was generated from the following files:

- `/home/attila/src/qecc-netbeans/qecc/Steane.h`
- `/home/attila/src/qecc-netbeans/qecc/Steane.cpp`

## 5.15 Test Class Reference

```
#include <Test.h>
```

Inheritance diagram for Test:



### Public Member Functions

- void [shorTest](#) (bool input)
- void [steaneTest](#) (bool input)
- void [code5Test](#) (bool input)
- void [bitFlipTest](#) (bool input)
- void [initalize](#) () override
- void [testMixed](#) ()
- std::string [getResult](#) () override
- void [runAllTests](#) ()
- [Test](#) (int type, int maxnumthreads, bool p)

## Static Public Attributes

- static const int [shorruns](#) = 4 \* [Shor::CS](#) + 6
- static const int [steaneruns](#) = 5 \* [Steane::CS](#)
- static const int [code5runs](#) = 4 \* [Code5::CS](#)
- static const int [bitflpruns](#) = [BitFlip::CS](#)

## Additional Inherited Members

### 5.15.1 Detailed Description

Runs tests for codes

### 5.15.2 Constructor & Destructor Documentation

5.15.2.1 `Test::Test ( int type, int maxnumthreads, bool p )` `[inline]`

Parameters

<i>type</i>	<a href="#">Code</a> type
<i>maxnumthreads</i>	Maximum number of threads

### 5.15.3 Member Function Documentation

5.15.3.1 `void Test::bitFlipTest ( bool input )`

Initialize for bit flip

Parameters

<i>input</i>	input bit
--------------	-----------

5.15.3.2 `void Test::code5Test ( bool input )`

Initialize for 5Qubit

Parameters

<i>input</i>	input bit
--------------	-----------

5.15.3.3 `std::string Test::getResult ( )` `[inline]`, `[override]`, `[virtual]`

Descriptor string

If all are okay **TEST SUCCESFULL**



If some failed **SOME TESTS FAILED**

Returns

Implements [Runnable](#).

**5.15.3.4** `void Test::initailize ( )` `[override]`, `[virtual]`

Initializes code list. To be defined by child!

Implements [Runnable](#).

**5.15.3.5** `void Test::runAllTests ( )`

Runs all inbuilt tests

**5.15.3.6** `void Test::shorTest ( bool input )`

Initialize for shor

Parameters

<i>input</i>	input bit
--------------	-----------

**5.15.3.7** `void Test::steaneTest ( bool input )`

Initialize for steane

Parameters

<i>input</i>	input bit
--------------	-----------

**5.15.3.8** `void Test::testMixed ( )` `[inline]`

Run tests using density matrix notation

## 5.15.4 Member Data Documentation

**5.15.4.1** `const int Test::bitflpruns = BitFlip::CS` `[static]`

Number of runs for bit flip code

5.15.4.2 `const int Test::code5runs = 4 * Code5::CS` `[static]`

Number of runs for 5Qubit code

5.15.4.3 `const int Test::shorruns = 4 * Shor::CS + 6` `[static]`

Number of runs for [Shor](#) code

5.15.4.4 `const int Test::steanneruns = 5 * Steane::CS` `[static]`

Number of runs for steane code

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

- [/home/attila/src/qecc-netbeans/qecc/Test.h](#)
- [/home/attila/src/qecc-netbeans/qecc/Test.cpp](#)

## Chapter 6

# File Documentation

### 6.1 /home/attila/src/qecc-netbeans/qecc/.dep.inc File Reference

### 6.2 /home/attila/src/qecc-netbeans/qecc/Aad4.cpp File Reference

```
#include "Aad4.h"  
#include <complex>  
#include <math.h>
```

### 6.3 /home/attila/src/qecc-netbeans/qecc/Aad4.h File Reference

```
#include "qpp.h"  
#include "Code.h"
```

#### Classes

- class [Aad4](#)

### 6.4 /home/attila/src/qecc-netbeans/qecc/BitFlip.cpp File Reference

```
#include "BitFlip.h"
```

### 6.5 /home/attila/src/qecc-netbeans/qecc/BitFlip.h File Reference

```
#include "qpp.h"  
#include "Code.h"  
#include <iostream>
```

## Classes

- class [BitFlip](#)

## 6.6 /home/attila/src/qecc-netbeans/qecc/Channel.cpp File Reference

```
#include "Channel.h"
```

## 6.7 /home/attila/src/qecc-netbeans/qecc/Channel.h File Reference

```
#include "Error.h"  
#include "Node.h"  
#include <queue>
```

## Classes

- class [Channel](#)

## 6.8 /home/attila/src/qecc-netbeans/qecc/Code.cpp File Reference

```
#include "Code.h"
```

## 6.9 /home/attila/src/qecc-netbeans/qecc/Code.h File Reference

```
#include "qpp.h"  
#include "Error.h"  
#include <iostream>  
#include <queue>  
#include <atomic>
```

## Classes

- class [Code](#)

## 6.10 /home/attila/src/qecc-netbeans/qecc/Code5.cpp File Reference

```
#include "Code5.h"
```

## 6.11 /home/attila/src/qecc-netbeans/qecc/Code5.h File Reference

```
#include "qpp.h"
#include "Code.h"
#include <iostream>
```

### Classes

- class [Code5](#)

## 6.12 /home/attila/src/qecc-netbeans/qecc/Error.cpp File Reference

```
#include "Error.h"
#include "Code.h"
```

## 6.13 /home/attila/src/qecc-netbeans/qecc/Error.h File Reference

```
#include "qpp.h"
```

### Classes

- class [Error](#)

## 6.14 /home/attila/src/qecc-netbeans/qecc/Network.cpp File Reference

```
#include "Network.h"
#include <fstream>
#include <list>
#include "Steane.h"
#include "Shor.h"
#include "Code5.h"
#include "BitFlip.h"
#include "Code.h"
#include "None.h"
#include "Aad4.h"
```

## 6.15 /home/attila/src/qecc-netbeans/qecc/Network.h File Reference

```
#include "Node.h"
#include "Channel.h"
#include "Runnable.h"
#include <list>
```

## Classes

- class [Network](#)

## 6.16 /home/attila/src/qecc-netbeans/qecc/Node.cpp File Reference

```
#include "Node.h"
```

## 6.17 /home/attila/src/qecc-netbeans/qecc/Node.h File Reference

```
#include <atomic>
#include <string>
```

## Classes

- class [Node](#)

## 6.18 /home/attila/src/qecc-netbeans/qecc/None.cpp File Reference

```
#include "None.h"
```

## 6.19 /home/attila/src/qecc-netbeans/qecc/None.h File Reference

```
#include "qpp.h"
#include "Code.h"
#include <iostream>
```

## Classes

- class [None](#)

## 6.20 /home/attila/src/qecc-netbeans/qecc/qecc.cpp File Reference

```
#include <cstdlib>
#include "Shor.h"
#include "Steane.h"
#include "Code.h"
#include "Code5.h"
#include "None.h"
#include "Runner.h"
#include "Test.h"
#include <iostream>
#include <unistd.h>
#include "Single.h"
#include "Network.h"
```

## Functions

- void `printHelp` ()
- int `main` (int argc, char \*\*argv)

## Variables

- volatile int `numthreads` = 0
- volatile int `errRun` = 0

### 6.20.1 Function Documentation

#### 6.20.1.1 int main ( int *argc*, char \*\* *argv* )

Main function of qecc

##### Parameters

<i>argc</i>	
<i>argv</i>	

##### Returns

#### 6.20.1.2 void printHelp ( )

Print help for qecc

### 6.20.2 Variable Documentation

#### 6.20.2.1 volatile int errRun = 0

#### 6.20.2.2 volatile int numthreads = 0

## 6.21 /home/attila/src/qecc-netbeans/qecc/README.md File Reference

## 6.22 /home/attila/src/qecc-netbeans/qecc/Runnable.cpp File Reference

```
#include "Runnable.h"
```

## 6.23 /home/attila/src/qecc-netbeans/qecc/Runnable.h File Reference

```
#include "Runner.h"
```

### Classes

- class [Runnable](#)

## 6.24 /home/attila/src/qecc-netbeans/qecc/Runner.cpp File Reference

```
#include <queue>
#include "Runner.h"
#include <unistd.h>
```

### Functions

- void \* [runCode](#) (void \*c)

### 6.24.1 Function Documentation

#### 6.24.1.1 void\* runCode ( void \* c )

Function to be run by [Runner](#) using pthread

#### Parameters

<code>c</code>	Pointer to <a href="#">Code</a> instance to be run
----------------	--

#### Returns

Unused

## 6.25 /home/attila/src/qecc-netbeans/qecc/Runner.h File Reference

```
#include "Code.h"
```

### Classes

- class [Runner](#)



## Functions

- void \* [runCode](#) (void \*c)

### 6.25.1 Function Documentation

#### 6.25.1.1 void\* runCode ( void \* c )

Function to be run by [Runner](#) using pthread

##### Parameters

c	Pointer to <a href="#">Code</a> instance to be run
---	--

##### Returns

Unused

## 6.26 /home/attila/src/qecc-netbeans/qecc/Shor.cpp File Reference

```
#include "Shor.h"
```

## 6.27 /home/attila/src/qecc-netbeans/qecc/Shor.h File Reference

```
#include "qpp.h"
#include "Code.h"
#include <iostream>
```

## Classes

- class [Shor](#)

## 6.28 /home/attila/src/qecc-netbeans/qecc/Single.cpp File Reference

```
#include "Single.h"
#include "Runnable.h"
#include "Code5.h"
#include "Steane.h"
#include "Shor.h"
#include "None.h"
#include "BitFlip.h"
#include "Aad4.h"
```

## 6.29 /home/attila/src/qecc-netbeans/qecc/Single.h File Reference

```
#include "Runner.h"  
#include "Runnable.h"
```

### Classes

- class [Single](#)

## 6.30 /home/attila/src/qecc-netbeans/qecc/Steane.cpp File Reference

```
#include "Steane.h"
```

## 6.31 /home/attila/src/qecc-netbeans/qecc/Steane.h File Reference

```
#include "qpp.h"  
#include "Code.h"  
#include <iostream>
```

### Classes

- class [Steane](#)

## 6.32 /home/attila/src/qecc-netbeans/qecc/Test.cpp File Reference

```
#include "Test.h"  
#include "Shor.h"  
#include "Steane.h"  
#include "Code5.h"  
#include "BitFlip.h"
```

## 6.33 /home/attila/src/qecc-netbeans/qecc/Test.h File Reference

```
#include "Runner.h"  
#include "Runnable.h"
```

### Classes

- class [Test](#)

# Index

- /home/attila/src/qecc-netbeans/qecc/.dep.inc, 53
- /home/attila/src/qecc-netbeans/qecc/Aad4.cpp, 53
- /home/attila/src/qecc-netbeans/qecc/Aad4.h, 53
- /home/attila/src/qecc-netbeans/qecc/BitFlip.cpp, 53
- /home/attila/src/qecc-netbeans/qecc/BitFlip.h, 53
- /home/attila/src/qecc-netbeans/qecc/Channel.cpp, 54
- /home/attila/src/qecc-netbeans/qecc/Channel.h, 54
- /home/attila/src/qecc-netbeans/qecc/Code.cpp, 54
- /home/attila/src/qecc-netbeans/qecc/Code.h, 54
- /home/attila/src/qecc-netbeans/qecc/Code5.cpp, 54
- /home/attila/src/qecc-netbeans/qecc/Code5.h, 55
- /home/attila/src/qecc-netbeans/qecc/Error.cpp, 55
- /home/attila/src/qecc-netbeans/qecc/Error.h, 55
- /home/attila/src/qecc-netbeans/qecc/Network.cpp, 55
- /home/attila/src/qecc-netbeans/qecc/Network.h, 55
- /home/attila/src/qecc-netbeans/qecc/Node.cpp, 56
- /home/attila/src/qecc-netbeans/qecc/Node.h, 56
- /home/attila/src/qecc-netbeans/qecc/None.cpp, 56
- /home/attila/src/qecc-netbeans/qecc/None.h, 56
- /home/attila/src/qecc-netbeans/qecc/Runnable.cpp, 57
- /home/attila/src/qecc-netbeans/qecc/Runnable.h, 57
- /home/attila/src/qecc-netbeans/qecc/Runner.cpp, 58
- /home/attila/src/qecc-netbeans/qecc/Runner.h, 58
- /home/attila/src/qecc-netbeans/qecc/Shor.cpp, 59
- /home/attila/src/qecc-netbeans/qecc/Shor.h, 59
- /home/attila/src/qecc-netbeans/qecc/Single.cpp, 59
- /home/attila/src/qecc-netbeans/qecc/Single.h, 59
- /home/attila/src/qecc-netbeans/qecc/Steane.cpp, 60
- /home/attila/src/qecc-netbeans/qecc/Steane.h, 60
- /home/attila/src/qecc-netbeans/qecc/Test.cpp, 60
- /home/attila/src/qecc-netbeans/qecc/Test.h, 60
- /home/attila/src/qecc-netbeans/qecc/qecc.cpp, 56
- ~Channel
  - Channel, 15
- ~Code
  - Code, 18
- ~Network
  - Network, 29
- ~Runnable
  - Runnable, 36
- ~Runner
  - Runner, 38
- AAD4
  - Runnable, 37
- ADC
  - Error, 27
- Aad4, 7
  - Aad4, 9
  - CS, 10
  - encode, 9
  - getCS, 9
  - getDescriptor, 9
  - run, 9
- addCode
  - Runner, 39
- addError
  - Code, 18
- applyCGT
  - Code, 18
- applyGT
  - Code, 18
- BITFLIP
  - Runnable, 37
- BitFlip, 10
  - BitFlip, 12
  - CS, 13
  - encode, 12
  - getCS, 12
  - getDescriptor, 12
  - run, 13
- bitFlipTest
  - Test, 50
- bitflipruns
  - Test, 51
- c
  - Code, 21
- CODE5
  - Runnable, 37
- CONST
  - Error, 27
- Channel, 13
  - ~Channel, 15
  - Channel, 14
  - debugPrint, 15
  - getError, 15
  - getIds, 15
  - matchId, 15
  - matchOneGetOther, 16
- Code, 16
  - ~Code, 18
  - addError, 18
  - applyCGT, 18
  - applyGT, 18
  - c, 21
  - Code, 17
  - convertToMixed, 18
  - d, 21

- deleteError, 21
- disableErrorDeletion, 18
- encode, 19
- Error, 21
- error, 19
- errorCounter, 21
- errorlist, 21
- getCS, 19
- getDescriptor, 19
- getMes, 19
- getOK, 20
- hadamardAllCodeBits, 20
- hadamardCodeBits, 20
- input, 21
- mixed, 21
- ok, 22
- result, 22
- run, 20
- setMixed, 21
- setandmesAnc, 20
- threadCounter, 22
- Code5, 22
  - Code5, 24
  - CS, 25
  - encode, 24
  - getCS, 24
  - getDescriptor, 24
  - run, 25
  - testError, 25
- code5Test
  - Test, 50
- code5runs
  - Test, 51
- convertToMixed
  - Code, 18
- CS
  - Aad4, 10
  - BitFlip, 13
  - Code5, 25
  - None, 34
  - Shor, 42
  - Steane, 48
- d
  - Code, 21
- debugPrint
  - Channel, 15
- deleteError
  - Code, 21
- disableErrorDeletion
  - Code, 18
- encode
  - Aad4, 9
  - BitFlip, 12
  - Code, 19
  - Code5, 24
  - None, 33
  - Shor, 41
  - Steane, 47
- errRun
  - qecc.cpp, 57
- Error, 25
  - ADC, 27
  - CONST, 27
  - Code, 21
  - Error, 26
  - getError, 26
  - RAND, 27
  - runError, 26
  - runErrorOneBit, 27
  - setError, 27
- error
  - Code, 19
- errorCounter
  - Code, 21
- errorlist
  - Code, 21
- getBER
  - Runner, 39
- getCodeType
  - Runnable, 36
- getCS
  - Aad4, 9
  - BitFlip, 12
  - Code, 19
  - Code5, 24
  - None, 33
  - Shor, 41
  - Steane, 47
- getDescriptor
  - Aad4, 9
  - BitFlip, 12
  - Code, 19
  - Code5, 24
  - None, 33
  - Shor, 42
  - Steane, 47
- getError
  - Channel, 15
  - Error, 26
- getId
  - Node, 30
- getIds
  - Channel, 15
- getMes
  - Code, 19
- getName
  - Node, 30
- getOK
  - Code, 20
- getResult
  - Network, 29
  - Runnable, 36
  - Single, 45
  - Test, 50

- hadamardAllCodeBits
  - Code, [20](#)
- hadamardCodeBits
  - Code, [20](#)
- inititalize
  - Network, [29](#)
  - Runnable, [36](#)
  - Single, [45](#)
  - Test, [51](#)
- input
  - Code, [21](#)
- main
  - qecc.cpp, [57](#)
- matchId
  - Channel, [15](#)
- matchOneGetOther
  - Channel, [16](#)
- mixed
  - Code, [21](#)
- NONE
  - Runnable, [37](#)
- Network, [28](#)
  - ~Network, [29](#)
  - getResult, [29](#)
  - inititalize, [29](#)
  - Network, [29](#)
  - runAll, [29](#)
- Node, [30](#)
  - getId, [30](#)
  - getName, [30](#)
  - Node, [30](#)
- None, [31](#)
  - CS, [34](#)
  - encode, [33](#)
  - getCS, [33](#)
  - getDescriptor, [33](#)
  - None, [33](#)
  - run, [34](#)
- numthreads
  - qecc.cpp, [57](#)
- ok
  - Code, [22](#)
- printHelp
  - qecc.cpp, [57](#)
- qecc.cpp
  - errRun, [57](#)
  - main, [57](#)
  - numthreads, [57](#)
  - printHelp, [57](#)
- RAND
  - Error, [27](#)
- reset
  - Runner, [39](#)
- result
  - Code, [22](#)
- run
  - Aad4, [9](#)
  - BitFlip, [13](#)
  - Code, [20](#)
  - Code5, [25](#)
  - None, [34](#)
  - Runnable, [37](#)
  - Runner, [39](#)
  - Shor, [42](#)
  - Steane, [47](#)
- runAll
  - Network, [29](#)
- runAllTests
  - Test, [51](#)
- runCode
  - Runner.cpp, [58](#)
  - Runner.h, [58](#)
- runError
  - Error, [26](#)
- runErrorOneBit
  - Error, [27](#)
- Runnable, [34](#)
  - ~Runnable, [36](#)
  - AAD4, [37](#)
  - BITFLIP, [37](#)
  - CODE5, [37](#)
  - getCodeType, [36](#)
  - getResult, [36](#)
  - inititalize, [36](#)
  - NONE, [37](#)
  - run, [37](#)
  - Runnable, [36](#)
  - runner, [37](#)
  - SHOR, [37](#)
  - STEANE, [38](#)
  - setCodeType, [37](#)
- Runner, [38](#)
  - ~Runner, [38](#)
  - addCode, [39](#)
  - getBER, [39](#)
  - reset, [39](#)
  - run, [39](#)
  - Runner, [38](#)
- runner
  - Runnable, [37](#)
- Runner.cpp
  - runCode, [58](#)
- Runner.h
  - runCode, [58](#)
- SHOR
  - Runnable, [37](#)
- STEANE
  - Runnable, [38](#)
- setCodeType
  - Runnable, [37](#)
- setError

- Error, [27](#)
- setMixed
  - Code, [21](#)
- setandmesAnc
  - Code, [20](#)
- Shor, [39](#)
  - CS, [42](#)
  - encode, [41](#)
  - getCS, [41](#)
  - getDescriptor, [42](#)
  - run, [42](#)
  - Shor, [41](#)
  - super, [41](#)
- shorTest
  - Test, [51](#)
- shorruns
  - Test, [52](#)
- Single, [43](#)
  - getResult, [45](#)
  - initalize, [45](#)
  - Single, [44](#)
- Steane, [45](#)
  - CS, [48](#)
  - encode, [47](#)
  - getCS, [47](#)
  - getDescriptor, [47](#)
  - run, [47](#)
  - Steane, [47](#)
- steaneTest
  - Test, [51](#)
- steaneruns
  - Test, [52](#)
- super
  - Shor, [41](#)
- Test, [48](#)
  - bitFlipTest, [50](#)
  - bitflipruns, [51](#)
  - code5Test, [50](#)
  - code5runs, [51](#)
  - getResult, [50](#)
  - initalize, [51](#)
  - runAllTests, [51](#)
  - shorTest, [51](#)
  - shorruns, [52](#)
  - steaneTest, [51](#)
  - steaneruns, [52](#)
  - Test, [50](#)
  - testMixed, [51](#)
- testError
  - Code5, [25](#)
- testMixed
  - Test, [51](#)
- threadCounter
  - Code, [22](#)