

Ruwavec

231

Generated by Doxygen 1.5.6

Fri Jul 2 00:33:01 2010

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	Annotation Struct Reference	3
2.1.1	Detailed Description	3
2.2	Contacts Class Reference	4
2.2.1	Detailed Description	4
2.3	LoginDialog Class Reference	5
2.3.1	Detailed Description	5
2.3.2	Constructor & Destructor Documentation	5
2.3.2.1	LoginDialog	5
2.3.3	Member Function Documentation	6
2.3.3.1	accept	6
2.3.3.2	setDefaultValues	6
2.3.3.3	attemptLogin	6
2.4	MainWindow Class Reference	7
2.4.1	Detailed Description	7
2.4.2	Constructor & Destructor Documentation	8
2.4.2.1	MainWindow	8
2.4.3	Member Function Documentation	8
2.4.3.1	addOutputMessage	8
2.4.3.2	addOutputMessage	8
2.4.3.3	addLogMessage	8
2.5	Ruwavec Class Reference	9
2.5.1	Detailed Description	10
2.5.2	Member Function Documentation	10
2.5.2.1	parseError	10

2.5.2.2	parseLogMessage	10
2.5.2.3	openWave	10
2.5.2.4	attemptLogin	11
2.5.2.5	sendPacket	11
2.5.2.6	addWave	11
2.5.2.7	amountWaves	11
2.5.2.8	getWave	11
2.5.2.9	error	12
2.5.2.10	logMessageRecieved	12
2.5.2.11	waveOpened	12
2.6	RuwavecUI Class Reference	13
2.6.1	Detailed Description	13
2.6.2	Constructor & Destructor Documentation	13
2.6.2.1	RuwavecUI	13
2.6.3	Member Function Documentation	13
2.6.3.1	startLogin	13
2.7	Wave Class Reference	15
2.7.1	Detailed Description	15
2.7.2	Constructor & Destructor Documentation	15
2.7.2.1	Wave	15
2.7.3	Member Function Documentation	15
2.7.3.1	addWavelet	15
2.7.3.2	amountWavelets	16
2.7.3.3	getWavelet	16
2.8	Wavelet Class Reference	17
2.8.1	Detailed Description	17
2.8.2	Constructor & Destructor Documentation	17
2.8.2.1	Wavelet	17
2.8.3	Member Function Documentation	18
2.8.3.1	setText	18
2.8.3.2	addAnnotation	18
2.8.3.3	addAnnotation	18
2.8.3.4	text	18
2.8.3.5	annotation	18
2.8.3.6	amountAnnotations	19
2.9	WaveList Class Reference	20

2.9.1	Detailed Description	20
2.9.2	Constructor & Destructor Documentation	20
2.9.2.1	WaveList	20
2.10	WaveView Class Reference	21
2.10.1	Detailed Description	21
2.10.2	Constructor & Destructor Documentation	21
2.10.2.1	WaveView	21
2.10.3	Member Function Documentation	21
2.10.3.1	convertWaveletToHtmlString	21
2.10.3.2	setWave	22

Chapter 1

Class Index

1.1 Class List

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

Annotation (A brief Wavelet Annotation representation (just an example))	3
Contacts (Contacts widget)	4
LoginDialog (Login Dialog)	5
MainWindow (Main Window, displays general information)	7
Ruwavec (Ruwavec is the main client class)	9
RuwavecUI (RuwavecUI , the interface for Ruwavec)	13
Wave (A brief Wave representation (just an example))	15
Wavelet (A brief Wavelet representation (just an example))	17
WaveList (WaveList widget, displays users current wavelets)	20
WaveView (WaveView widget, translate a wave in html and displays it)	21

Chapter 2

Class Documentation

2.1 Annotation Struct Reference

A brief [Wavelet Annotation](#) representation (just an example).

```
#include <wavelet.h>
```

Public Member Functions

- [Annotation](#) * `copy` () const
Copy an annotation.

Public Attributes

- `QString` `type`
- `int` `offset`
- `int` `length`

2.1.1 Detailed Description

A brief [Wavelet Annotation](#) representation (just an example).

Author:

Koray Yanik

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

- `wavelet.h`

2.2 Contacts Class Reference

[Contacts](#) widget.

```
#include <contacts.h>
```

Public Member Functions

- [Contacts](#) (QWidget *parent=0)

2.2.1 Detailed Description

[Contacts](#) widget.

Author:

Cathalijne van Wettum
Koray Yanik

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

- contacts.h
- contacts.cpp

2.3 LoginDialog Class Reference

Login Dialog.

```
#include <logindialog.h>
```

Public Member Functions

- [LoginDialog](#) ([Ruwavec](#) *parentRuwavec, [QWidget](#) *parent=0)
Constructor, requires a parent ruwavec and may have a parent qwidget.
- void [accept](#) ()
- void [setDefaultValues](#) (const [QString](#) &user, const [QString](#) &password, const [QString](#) &host, const [QString](#) &domain, int port)
Set default values to display.

Protected Slots

- void [attemptLogin](#) (const [QString](#) &user, const [QString](#) &password, const [QString](#) &host, const [QString](#) &domain, int port)
Disable inputs and attempt a login at the parent ruwavec with the given data.

Protected Member Functions

- void [changeEvent](#) ([QEvent](#) *e)

2.3.1 Detailed Description

Login Dialog.

Author:

Koray Yanik

2.3.2 Constructor & Destructor Documentation

2.3.2.1 LoginDialog::LoginDialog ([Ruwavec](#) *parentRuwavec, [QWidget](#) *parent = 0)

Constructor, requires a parent ruwavec and may have a parent qwidget.

Parameters:

parentRuwavec a pointer to the parent [Ruwavec](#)

parent a pointer to the given [QWidget](#) (optional)

2.3.3 Member Function Documentation

2.3.3.1 void LoginDialog::accept ()

Check to accept, if both inputs are not empty.

2.3.3.2 void LoginDialog::setDefaultValues (const QString & *user*, const QString & *password*, const QString & *host*, const QString & *domain*, int *port*)

Set default values to display.

Parameters:

user the username to display
password the password to display
host the host to display
domain the domain to display
port the port to display

2.3.3.3 void LoginDialog::attemptLogin (const QString & *user*, const QString & *password*, const QString & *host*, const QString & *domain*, int *port*) [protected, slot]

Disable inputs and attempt a login at the parent ruwavec with the given data.

Parameters:

user the given username
password the given password
host the host to use
domain the domain to use
port the port to use

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

- logindialog.h
- logindialog.cpp

2.4 MainWindow Class Reference

Main Window, displays general information.

```
#include <mainwindow.h>
```

Public Slots

- void [addLogMessage](#) (QXmppLogger::MessageType type, const QString &message)
Display a given log message.

Public Member Functions

- [MainWindow](#) (Ruwavec *parentRuwavec, QWidget *parent=0)
Constructor, requires a parent ruwavec and may have a parent QWidget.
- void [setConnected](#) ()
Set the main window to its connected state.
- void [setDisconnected](#) ()
Set the main window to its disconnected state.
- void [addOutputMessage](#) (const QXmppMessage &message)
Display a QXmppMessage.
- void [addOutputMessage](#) (const QString &message)
Display a QString.
- void [openContacts](#) ()
Open the contacts window.

Protected Member Functions

- void [changeEvent](#) (QEvent *e)

2.4.1 Detailed Description

Main Window, displays general information.

Author:

Cathalijne van Wettum
Koray Yanik

2.4.2 Constructor & Destructor Documentation

2.4.2.1 `MainWindow::MainWindow (Ruwavec * parentRuwavec, QWidget * parent = 0)`

Constructor, requires a parent ruwavec and may have a parent qwidget.

Parameters:

parentRuwavec a pointer to the parent [Ruwavec](#)

parent a pointer to the given QWidget (optional)

2.4.3 Member Function Documentation

2.4.3.1 `void MainWindow::addOutputMessage (const QXmppMessage & message)`

Display a QXmppMessage.

Parameters:

message a reference to the given message to display

2.4.3.2 `void MainWindow::addOutputMessage (const QString & message)`

Display a QString.

Parameters:

message a reference to the given string to display

2.4.3.3 `void MainWindow::addLogMessage (QXmppLogger::MessageType type, const QString & message) [slot]`

Display a given log message.

Parameters:

type the type of the message

message a reference of the message

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

- mainwindow.h
- mainwindow.cpp

2.5 Ruwavec Class Reference

[Ruwavec](#) is the main client class.

```
#include <ruwavec.h>
```

Signals

- void [connected](#) ()
Emitted when we are done connecting.
- void [disconnected](#) ()
Emitted when we are done disconnecting.
- void [error](#) (QString error)
Emitted when an error occurred.
- void [logMessageRecieved](#) (QXmppLogger::MessageType type, const QString &message)
Emitted when we recieve a log message.
- void [waveOpened](#) (Wave *wave)
Emitted when a wave is opened.
- void [waveAdded](#) ()
Emitted when a wave is added to the client.

Public Member Functions

- void [attemptLogin](#) (const QString &user, const QString &password, const QString &host, const QString &domain, int port)
Attempt to log in with a given user, password, host, domain and port.
- void [attemptLogout](#) ()
Attempt to logout.
- void [sendPacket](#) (const QXmppPacket &packet)
Send a given QXmppPacket.
- void [addWave](#) (Wave *wave)
For testing purposes we add this function to directly add a wavelet to the client. This should be replaced later, the client should recieve wavelets and add them to itself.
- int [amountWaves](#) ()
Get the amount of waves in the client at the moment.
- Wave * [getWave](#) (int index)
Get a pointer to the wave with the given index, or NULL if it doesnt exist.

Protected Slots

- void `createWaveServerComponent` ()
After logging in, create a waveserver component and emit connected signal.
- void `parseError` (QXmppClient::Error error)
Parse and displays a given QXmppClient error.
- void `parseLogMessage` (QXmppLogger::MessageType type, const QString &message)
Parse and displays a given QXmppLogger message.
- void `openWave` (int index)
Open a wave with a given index in the WaveView.

2.5.1 Detailed Description

`Ruwavec` is the main client class.

Author:

Koray Yanik

2.5.2 Member Function Documentation

2.5.2.1 void Ruwavec::parseError (QXmppClient::Error error) [protected, slot]

Parse and displays a given QXmppClient error.

Parameters:

error the given error

2.5.2.2 void Ruwavec::parseLogMessage (QXmppLogger::MessageType type, const QString &message) [protected, slot]

Parse and displays a given QXmppLogger message.

Parameters:

type the type of the message

message a reference to the message

2.5.2.3 void Ruwavec::openWave (int index) [protected, slot]

Open a wave with a given index in the `WaveView`.

Parameters:

index the index of the wave to open

2.5.2.4 void Ruwavec::attemptLogin (const QString & *user*, const QString & *password*, const QString & *host*, const QString & *domain*, int *port*)

Attempt to log in with a given user, password, host, domain and port.

Parameters:

user the user

password the password

host the host

domain the domain

port the port

2.5.2.5 void Ruwavec::sendPacket (const QXmppPacket & *packet*)

Send a given QXmppPacket.

Parameters:

a reference to the given packet

2.5.2.6 void Ruwavec::addWave (Wave * *wave*)

For testing purposes we add this function to directly add a wavelet to the client. This should be replaced later, the client should receive wavelets and add them to itself.

Parameters:

wavelet a pointer to the wavelet to add

2.5.2.7 int Ruwavec::amountWaves ()

Get the amount of waves in the client at the moment.

Returns:

the amount of waves

2.5.2.8 Wave * Ruwavec::getWave (int *index*)

Get a pointer to the wave with the given index, or NULL if it doesn't exist.

Parameters:

index the index of the wave to retrieve a pointer of

Returns:

a pointer to the asked wave or NULL if failed

2.5.2.9 void Ruwavec::error (QString *error*) [signal]

Emitted when an error occurred.

Parameters:

error the given error

2.5.2.10 void Ruwavec::logMessageRecieved (QXmppLogger::MessageType *type*, const QString & *message*) [signal]

Emitted when we receive a log message.

Parameters:

type message type

message the message itself

2.5.2.11 void Ruwavec::waveOpened (Wave * *wave*) [signal]

Emitted when a wave is opened.

Parameters:

wave a pointer to the opened wave.

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

- ruwavec.h
- moc_ruwavec.cpp
- ruwavec.cpp

2.6 RuwavecUI Class Reference

[RuwavecUI](#), the interface for [Ruwavec](#).

```
#include <ruwavecui.h>
```

Public Member Functions

- [RuwavecUI](#) ([Ruwavec](#) *client)
Create a new [RuwavecUI](#) to communicate with a given client.
- void [startLogin](#) (QString user="", QString password="", QString host="", QString domain="", int port=5222)
Start the login process by displaying the login window. Optional default login values can be passed.

Protected Slots

- void [connected](#) ()
Call when connection is succesful. Will open all windows and close the login window.

2.6.1 Detailed Description

[RuwavecUI](#), the interface for [Ruwavec](#).

Author:

Koray Yanik

2.6.2 Constructor & Destructor Documentation

2.6.2.1 RuwavecUI::RuwavecUI ([Ruwavec](#) * client)

Create a new [RuwavecUI](#) to communicate with a given client.

Parameters:

client a pointer to the given client to communicate with

2.6.3 Member Function Documentation

2.6.3.1 void RuwavecUI::startLogin (QString *user* = "", QString *password* = "", QString *host* = "", QString *domain* = "", int *port* = 5222)

Start the login process by displaying the login window. Optional default login values can be passed.

Parameters:

user (optional) default value for username

password (optional) default value for the password

host (optional) default host to connect to

domain (optional) default domain to use

port (optional) default port to use

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

- ruwavecui.h
- ruwavecui.cpp

2.7 Wave Class Reference

A brief [Wave](#) representation (just an example).

```
#include <wave.h>
```

Public Member Functions

- [Wave](#) ([Wavelet](#) *wavelet)
Create a new wave with a pointer to the first wavelet.
- void [addWavelet](#) ([Wavelet](#) *wavelet)
Add a pointer to a given wavelet to the wave.
- int [amountWavelets](#) ()
Returns amount of wavelets in this wave.
- [Wavelet](#) * [getWavelet](#) (int index)
Returns a pointer to the wavelet at the given index. Returns null if the wavelet at the given index does not exist.

2.7.1 Detailed Description

A brief [Wave](#) representation (just an example).

Author:

Koray Yanik

2.7.2 Constructor & Destructor Documentation

2.7.2.1 [Wave::Wave](#) ([Wavelet](#) * *wavelet*)

Create a new wave with a pointer to the first wavelet.

Parameters:

wavelet a pointer to the first wavelet

2.7.3 Member Function Documentation

2.7.3.1 void [Wave::addWavelet](#) ([Wavelet](#) * *wavelet*)

Add a pointer to a given wavelet to the wave.

Parameters:

wavelet a pointer to the wavelet to add

2.7.3.2 int Wave::amountWavelets ()

Returns amount of wavelets in this wave.

Returns:

amount of wavelets

2.7.3.3 Wavelet * Wave::getWavelet (int *index*)

Returns a pointer to the wavelet at the given index. Returns null if the wavelet at the given index does not exist.

Parameters:

index the given index of the wavelet

Returns:

a pointer to the asked wavelet

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

- wave.h
- wave.cpp

2.8 Wavelet Class Reference

A brief [Wavelet](#) representation (just an example).

```
#include <wavelet.h>
```

Public Member Functions

- [Wavelet](#) (const QString &text="")
Creates a new wavelet with optional text.
- void [setText](#) (const QString &text)
Set the text of the wavelet.
- bool [addAnnotation](#) (const [Annotation](#) *annotation)
Add an annotation giving its type and its offset. Copies the given annotation so you can free it whenever you like.
- bool [addAnnotation](#) (const QString &type, int offset, int length)
Add an annotation giving its type and its offset.
- int [length](#) ()
Returns the length of the text. Note that this is purely the text, without taking annotations into account.
- const QString & [text](#) () const
Returns the text contained by the wavelet.
- const [Annotation](#) * [annotation](#) (int index) const
Return the annotation with a given index.
- int [amountAnnotations](#) () const
Returns the amount of annotations within the wavelet.

2.8.1 Detailed Description

A brief [Wavelet](#) representation (just an example).

Author:

Koray Yanik

2.8.2 Constructor & Destructor Documentation

2.8.2.1 [Wavelet::Wavelet](#) (const QString & text = " ")

Creates a new wavelet with optional text.

Parameters:

text the given text (optional)

2.8.3 Member Function Documentation

2.8.3.1 void Wavelet::setText (const QString & *text*)

Set the text of the wavelet.

Parameters:

text the given text

2.8.3.2 bool Wavelet::addAnnotation (const Annotation * *annotation*)

Add an annotation giving its type and its offset. Copies the given annotation so you can free it whenever you like.

Parameters:

annotation the given annotation

Returns:

true if succesful

2.8.3.3 bool Wavelet::addAnnotation (const QString & *type*, int *offset*, int *length*)

Add an annotation giving its type and its offset.

Parameters:

type the type of the annotation

offset the offset of the annotation

length of the annotation (amount of characters)

Returns:

true if succesful

2.8.3.4 const QString & Wavelet::text () const

Returns the text contained by the wavelet.

Returns:

the text.

2.8.3.5 const Annotation * Wavelet::annotation (int *index*) const

Return the annotation with a given index.

Parameters:

index the given index

Returns:

the annotation with the given index

2.8.3.6 int Wavelet::amountAnnotations () const

Returns the amount of annotations within the wavelet.

Returns:

the amount of annotations

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

- wavelet.h
- wavelet.cpp

2.9 WaveList Class Reference

[WaveList](#) widget, displays users current wavelets.

```
#include <wavelist.h>
```

Public Slots

- void [renderList](#) ()
Render the list after an update in the client's waves.

Public Member Functions

- [WaveList](#) ([Ruwavec](#) *client, [QWidget](#) *parent=0)
Create a new widget given a client and an optional parent widget.

2.9.1 Detailed Description

[WaveList](#) widget, displays users current wavelets.

Author:

Koray Yanik

2.9.2 Constructor & Destructor Documentation

2.9.2.1 [WaveList::WaveList](#) ([Ruwavec](#) * *client*, [QWidget](#) * *parent* = 0)

Create a new widget given a client and an optional parent widget.

Parameters:

- client* a pointer to the client
- a* pointer to the parent widget (optional)

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

- wavelist.h
- wavelist.cpp

2.10 WaveView Class Reference

[WaveView](#) widget, translate a wave in html and displays it.

```
#include <waveview.h>
```

Public Slots

- void [setWave](#) ([Wave](#) *wave)
Set the current [Wave](#).

Public Member Functions

- [WaveView](#) ([Ruwavec](#) *parentRuwavec, [QWidget](#) *parent=0)
Constructor, requires a parent ruwavec and may have a parent qwidget.
- void [renderWave](#) ()
Convert all wavelets in the wave to html and render them. This clears the current waveview.
- [QString](#) [convertWaveletToHtmlString](#) (const [Wavelet](#) *wavelet)
Convert a wavelet to html in a [QString](#).

2.10.1 Detailed Description

[WaveView](#) widget, translate a wave in html and displays it.

Author:

Koray Yanik

2.10.2 Constructor & Destructor Documentation

2.10.2.1 [WaveView::WaveView](#) ([Ruwavec](#) *parentRuwavec, [QWidget](#) *parent = 0)

Constructor, requires a parent ruwavec and may have a parent qwidget.

Parameters:

- parentRuwavec* a pointer to the parent [Ruwavec](#)
- parent* a pointer to the given [QWidget](#) (optional)

2.10.3 Member Function Documentation

2.10.3.1 [QString](#) [WaveView::convertWaveletToHtmlString](#) (const [Wavelet](#) *wavelet)

Convert a wavelet to html in a [QString](#).

Parameters:

wavelet the given wavelet to convert

Returns:

the QString containing a html representation of the wavelet.

2.10.3.2 void WaveView::setWave (Wave * wave) [slot]

Set the current [Wave](#).

Parameters:

wave the wave to set

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

- waveview.h
- waveview.cpp

Index

- accept
 - LoginDialog, 6
- addAnnotation
 - Wavelet, 18
- addLogMessage
 - MainWindow, 8
- addOutputMessage
 - MainWindow, 8
- addWave
 - Ruwavec, 11
- addWavelet
 - Wave, 15
- amountAnnotations
 - Wavelet, 19
- amountWavelets
 - Wave, 15
- amountWaves
 - Ruwavec, 11
- Annotation, 3
- annotation
 - Wavelet, 18
- attemptLogin
 - LoginDialog, 6
 - Ruwavec, 10
- Contacts, 4
- convertWaveletToHtmlString
 - WaveView, 21
- error
 - Ruwavec, 11
- getWave
 - Ruwavec, 11
- getWavelet
 - Wave, 16
- LoginDialog, 5
 - accept, 6
 - attemptLogin, 6
 - LoginDialog, 5
 - setDefaultValues, 6
- logMessageRecieved
 - Ruwavec, 12
- MainWindow, 7
 - addLogMessage, 8
 - addOutputMessage, 8
 - MainWindow, 8
- openWave
 - Ruwavec, 10
- parseError
 - Ruwavec, 10
- parseLogMessage
 - Ruwavec, 10
- Ruwavec, 9
 - addWave, 11
 - amountWaves, 11
 - attemptLogin, 10
 - error, 11
 - getWave, 11
 - logMessageRecieved, 12
 - openWave, 10
 - parseError, 10
 - parseLogMessage, 10
 - sendPacket, 11
 - waveOpened, 12
- RuwavecUI, 13
 - RuwavecUI, 13
 - startLogin, 13
- sendPacket
 - Ruwavec, 11
- setDefaultValues
 - LoginDialog, 6
- setText
 - Wavelet, 18
- setWave
 - WaveView, 22
- startLogin
 - RuwavecUI, 13
- text
 - Wavelet, 18
- Wave, 15
 - addWavelet, 15
 - amountWavelets, 15
 - getWavelet, 16

- Wave, 15
- Wavelet, 17
 - addAnnotation, 18
 - amountAnnotations, 19
 - annotation, 18
 - setText, 18
 - text, 18
 - Wavelet, 17
- WaveList, 20
 - WaveList, 20
- waveOpened
 - Ruwavec, 12
- WaveView, 21
 - convertWaveletToHtmlString, 21
 - setWave, 22
 - WaveView, 21