Nessie, reconocedor óptico de texto en recortes de prensa escrita

1.0

Generated by Doxygen 1.5.6

Wed Sep 24 17:40:25 2008

Contents

1 Directory Hierarchy **1**

1.1 Directories 1

2 Data Structure Index **3**

2.1 Data Structures 3

3 File Index **5**

3.1 File List 5

4 Directory Documentation **7**

4.1 inc/ Directory Reference 7

4.2 src/ Directory Reference 8

5 Data Structure Documentation **9**

5.1 Clip Class Reference 9

5.1.1 Detailed Description 11

5.1.2 Constructor & Destructor Documentation 13

5.1.2.1 Clip 13

5.1.2.2 ~Clip 13

5.1.3 Member Function Documentation 14

5.1.3.1 getImage 14

5.1.3.2 setImage 14

5.1.3.3 getXOrigin 14

5.1.3.4 setXOrigin 15

5.1.3.5 getYOrigin 15

5.1.3.6 setYOrigin 15

5.1.3.7 getHeight 15

5.1.3.8 setHeight 16

5.1.3.9 getWidth 16

5.1.3.10 setWidth 16

5.1.3.11 getColorspace . . . 16
5.1.3.12 setColorspace . . . 17
5.1.3.13 isGrayscale . . . 17
5.1.3.14 isColor . . . 17
5.1.3.15 isMonochromatic . . . 17
5.1.3.16 getPixel . . . 18
5.1.3.17 setPixel . . . 18
5.1.3.18 setPixel . . . 18
5.1.3.19 setPixel . . . 19
5.1.4 Field Documentation . . . 19
5.1.4.1 colorspace_ . . . 19
5.1.4.2 frame_ . . . 19
5.1.4.3 originPixel_ . . . 19
5.2 FontColor Struct Reference . . . 20
5.2.1 Detailed Description . . . 20
5.2.2 Constructor & Destructor Documentation . . . 21
5.2.2.1 FontColor . . . 21
5.2.2.2 FontColor . . . 21
5.2.2.3 ~FontColor . . . 21
5.3 Pixel Class Reference . . . 23
5.3.1 Detailed Description . . . 23
5.3.2 Constructor & Destructor Documentation . . . 26
5.3.2.1 Pixel . . . 26
5.3.2.2 Pixel . . . 26
5.3.2.3 Pixel . . . 26
5.3.2.4 Pixel . . . 26
5.3.2.5 Pixel . . . 27
5.3.2.6 Pixel . . . 27
5.3.2.7 ~Pixel . . . 27
5.3.3 Member Function Documentation . . . 27
5.3.3.1 getX . . . 27
5.3.3.2 getY . . . 28
5.3.3.3 getGrayLevel . . . 28
5.3.3.4 getColor . . . 28
5.3.3.5 getForeground . . . 28
5.3.3.6 setColor . . . 29

5.3.3.7 setColor . . . 29
5.3.3.8 setColor . . . 29
5.3.3.9 setColor . . . 29
5.3.3.10 setColor . . . 30
5.3.3.11 setColor . . . 30
5.4 Statistics Struct Reference . . . 31
5.4.1 Detailed Description . . . 31
5.4.2 Constructor & Destructor Documentation . . . 33
5.4.2.1 Statistics . . . 33
5.4.2.2 ∼Statistics . . . 33
5.5 Style Class Reference . . . 34
5.5.1 Detailed Description . . . 34
5.5.2 Constructor & Destructor Documentation . . . 36
5.5.2.1 Style . . . 36
5.5.2.2 Style . . . 36
5.5.2.3 ∼Style . . . 36
5.5.3 Member Function Documentation . . . 37
5.5.3.1 isBold . . . 37
5.5.3.2 isItalic . . . 37
5.5.3.3 isUnderlined . . . 37
5.5.3.4 isNormal . . . 38
5.5.3.5 getSize . . . 38
5.5.3.6 setSize . . . 38
5.5.3.7 getColor . . . 38
5.5.3.8 setColor . . . 39
5.5.3.9 setColor . . . 39
5.5.3.10 getWeight . . . 39
5.5.3.11 setWeight . . . 40
5.6 Text Class Reference . . . 41
5.6.1 Detailed Description . . . 43
5.6.2 Constructor & Destructor Documentation . . . 45
5.6.2.1 Text . . . 45
5.6.2.2 Text . . . 45
5.6.2.3 Text . . . 45
5.6.2.4 ∼Text . . . 46
5.6.3 Member Function Documentation . . . 46

5.6.3.1 addCharacter . . . 46
5.6.3.2 addCharacter . . . 46
5.6.3.3 addCharacter . . . 47
5.6.3.4 addCharacter . . . 47
5.6.3.5 removeCharacter . . . 47
5.6.3.6 getCharacterStyle . . . 48
5.6.3.7 setCharacterStyle . . . 48
5.6.3.8 getContent . . . 48
5.6.3.9 setContent . . . 49
5.6.3.10 getStyles . . . 49
5.6.3.11 setStyles . . . 49
5.6.3.12 getLength . . . 49
5.6.3.13 getWordRates . . . 50
5.6.3.14 getProportionality . . . 50
5.6.3.15 setProportionality . . . 50
5.6.3.16 computeWordRates . . . 51
5.6.3.17 updateWordRate . . . 51
5.6.3.18 tokenize . . . 51

6 File Documentation **53**

6.1 Clip.h File Reference . . . 53
6.1.1 Detailed Description . . . 53
6.2 Colorspace.h File Reference . . . 55
6.2.1 Detailed Description . . . 55
6.2.2 Enumeration Type Documentation . . . 55
6.2.2.1 Colorspace . . . 55
6.3 FontColor.h File Reference . . . 57
6.3.1 Detailed Description . . . 57
6.4 FontProportionality.h File Reference . . . 58
6.4.1 Detailed Description . . . 58
6.4.2 Enumeration Type Documentation . . . 58
6.4.2.1 FontProportionality . . . 58
6.5 FontWeight.h File Reference . . . 59
6.5.1 Detailed Description . . . 59
6.5.2 Enumeration Type Documentation . . . 59
6.5.2.1 FontWeight . . . 59
6.6 Pixel.h File Reference . . . 61

6.6.1 Detailed Description 61
6.7 Statistics.h File Reference 62
6.7.1 Detailed Description 62
6.8 Style.h File Reference 63
6.8.1 Detailed Description 63
6.9 Text.h File Reference 64
6.9.1 Detailed Description 64
6.10 WordRate.h File Reference 65
6.10.1 Detailed Description 65
6.10.2 Typedef Documentation 65
6.10.2.1 WordRate 65

Chapter 1

Directory Hierarchy

1.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

inc 7
src 8

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Clip (Press clip where the recognizer has to extract the text from) 9
FontColor (Font color of a character) 20
Pixel (Information about an image pixel) 23
Statistics (Statistics about a text and its recognition process) 31
Style (Font style of a character) 34
Text (Text extracted from the clip by the recognizer) 41

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Classifier.cpp . . . . . **??**
Classifier.h . . . . . **??**
Clip.cpp . . . . . **??**
Clip.h 53
Colorspace.h 55
FeatureVector.cpp . . . . . **??**
FeatureVector.h . . . . . **??**
FontColor.cpp . . . . . **??**
FontColor.h 57
FontProportionality.h 58
FontWeight.h 59
loadPDF.cpp . . . . . **??**
main.cpp . . . . . **??**
Partitioner.cpp . . . . . **??**
Partitioner.h . . . . . **??**
Pixel.cpp . . . . . **??**
Pixel.h 61
preprocessingTests.cpp . . . . . **??**
Preprocessor.cpp . . . . . **??**
Preprocessor.h . . . . . **??**
Recognizer.cpp . . . . . **??**
Recognizer.h . . . . . **??**
Shape.cpp . . . . . **??**
Shape.h . . . . . **??**
Statistics.cpp . . . . . **??**
Statistics.h 62
Style.cpp . . . . . **??**
Style.h 63
Text.cpp . . . . . **??**
Text.h 64
WordRate.h 65

Chapter 4

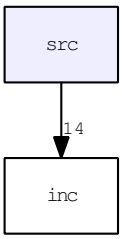
Directory Documentation

4.1 inc/ Directory Reference

Files

- file **Classifier.h**
- file Clip.h
- file Colorspace.h
- file **FeatureVector.h**
- file FontColor.h
- file FontProportionality.h
- file FontWeight.h
- file **Partitioner.h**
- file Pixel.h
- file **Preprocessor.h**
- file **Recognizer.h**
- file **Shape.h**
- file Statistics.h
- file Style.h
- file Text.h
- file WordRate.h

4.2 src/ Directory Reference



Files

- file **Classifier.cpp**
- file **Clip.cpp**
- file **FeatureVector.cpp**
- file **FontColor.cpp**
- file **loadPDF.cpp**
- file **main.cpp**
- file **Partitioner.cpp**
- file **Pixel.cpp**
- file **preprocessingTests.cpp**
- file **Preprocessor.cpp**
- file **Recognizer.cpp**
- file **Shape.cpp**
- file **Statistics.cpp**
- file **Style.cpp**
- file **Text.cpp**

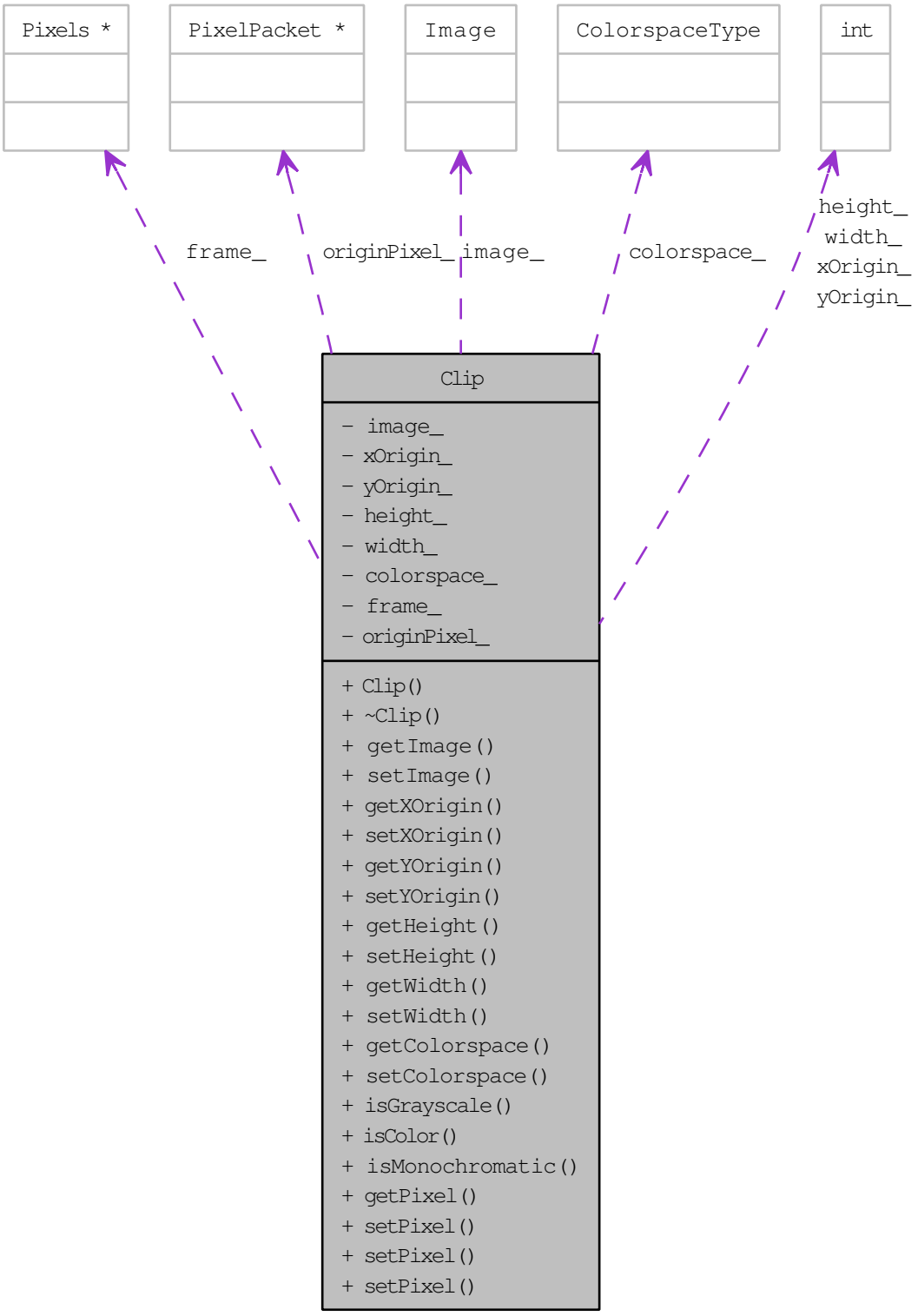
Chapter 5

Data Structure Documentation

5.1 Clip Class Reference

`#include <Clip.h>`

Collaboration diagram for Clip:



5.1.1 Detailed Description

Press clip where the recognizer has to extract the text from.

This class manages the press clip where the recognizer has to extract the text, loading it with the Magick++ utilities. The press clip is an image that may come in several formats, such JPEG, PDF, PNG, etc. The ImageMagick library provides an abstraction layer to keep the code independent from the format.

See also:

Pixel.h

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 32 of file Clip.h.

Public Member Functions

- Clip (Image image, unsigned int xOrigin, unsigned int yOrigin, unsigned int height, unsigned int width)

  Constructor.

- ~Clip ()

  Destructor.

- Image getImage ()

  Returns the current Image object.

- void setImage (Image image)

  Sets the Image object where the clip is present.

- unsigned int getXOrigin () const

  Returns the X coordinate of the clip's upper leftmost pixel.

- void setXOrigin (unsigned int x)

  Sets the X coordinate of the clip's upper leftmost pixel.

- unsigned int getYOrigin () const

  Returns the Y coordinate of the clip's upper leftmost pixel.

- void setYOrigin (unsigned int y)

  Sets the Y coordinate of the clip's upper leftmost pixel.

- unsigned int getHeight () const

  Returns the clip's height.

- void setHeight (unsigned int height)

  Sets the clip's height.

- unsigned int getWidth () const

  Returns the clip's width.

- void setWidth (unsigned int width)

  Sets the clip's width.

- ColorspaceType getColorspace () const

  Returns the clip's current colorspace.

- void setColorspace (ColorspaceType colorspace)

  Sets the clip's current colorspace.

- bool isGrayscale () const

  Returns true if the clip is in grayscale.

- bool isColor () const

  Returns true if the clip is in color.

- bool isMonochromatic () const

  Returns true if the clips is in black and white.

- Pixel getPixel (unsigned int x, unsigned int y) const

  Returns the pixel at coordinates (x,y).

- void setPixel (unsigned int x, unsigned int y, double grayLevel)

  Sets the gray level of a pixel at coordinates (x,y).

- void setPixel (unsigned int x, unsigned int y, double red, double green, double blue)

  Sets the color of a pixel at coordinates (x,y).

- void setPixel (unsigned int x, unsigned int y, bool isForeground)

  Sets whether a pixel at coordinates (x,y) belongs to the foreground or not.

Private Attributes

- Image image_

  Image where the clip belongs to.

- unsigned int xOrigin_

  X coordinate of the clip's upper leftmost pixel.

- unsigned int yOrigin_

  Y coordinate of the clip's upper leftmost pixel.

- unsigned int height_

Height of the clip in pixels.

- unsigned int width_

  Width of the clip in pixels.

- ColorspaceType colorspace_

  Clip's colorspace.

- Pixels * frame_

  Pixels frame where the clip is located.

- PixelPacket * originPixel_

  Frame's upper leftmost pixel.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Clip::Clip (Image *image*, unsigned int *xOrigin*, unsigned int *yOrigin*, unsigned int *height*, unsigned int *width*)

Constructor.

Initializes a Clip object located at coordinates () in the source image, with the height and width passed. If the xOrigin and yOrigin are over the image borders, an exception is thrown. If the width and height are over the image borders the clip is truncated.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 12 of file Clip.cpp.

5.1.2.2 Clip::~Clip ()

Destructor.

Destroys a Clip object, deleting all the associated data from Magick++ API

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 69 of file Clip.cpp.

5.1.3 Member Function Documentation

5.1.3.1 Image Clip::getImage ()

Returns the current Image object.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 79 of file Clip.cpp.

5.1.3.2 void Clip::setImage (Image *image*)

Sets the Image object where the clip is present.

By setting a new image, the clip will automatically try to relocate itself into it. That means the constructor behaviour is replicated to ensure all clip attributes regarding size and location remains valid.

See also:

Clip::Clip()

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 94 of file Clip.cpp.

5.1.3.3 unsigned int Clip::getXOrigin () const

Returns the X coordinate of the clip's upper leftmost pixel.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 146 of file Clip.cpp.

5.1.3.4 void Clip::setXOrigin (unsigned int *x*)

Sets the X coordinate of the clip's upper leftmost pixel.

If the new value x is over the image borders, an exception is thrown.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 158 of file Clip.cpp.

5.1.3.5 unsigned int Clip::getYOrigin () const

Returns the Y coordinate of the clip's upper leftmost pixel.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 171 of file Clip.cpp.

5.1.3.6 void Clip::setYOrigin (unsigned int *y*)

Sets the Y coordinate of the clip's upper leftmost pixel.

If the new value y is over the image borders, an exception is thrown.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 183 of file Clip.cpp.

5.1.3.7 unsigned int Clip::getHeight () const

Returns the clip's height.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 196 of file Clip.cpp.

5.1.3.8 void Clip::setHeight (unsigned int *height*)

Sets the clip's height.

If the height is over the image borders the clip is truncated to its maximum allowed value.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 208 of file Clip.cpp.

5.1.3.9 unsigned int Clip::getWidth () const

Returns the clip's width.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 221 of file Clip.cpp.

5.1.3.10 void Clip::setWidth (unsigned int *width*)

Sets the clip's width.

If the width is over the image borders the clip is truncated to its maximum allowed value.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 233 of file Clip.cpp.

5.1.3.11 ColorspaceType Clip::getColorspace () const

Returns the clip's current colorspace.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 246 of file Clip.cpp.

5.1.3.12 void Clip::setColorspace (ColorspaceType *colorspace*)

Sets the clip's current colorspace.

By setting the colorspace to a certain value the image colorspace changes, affecting all its pixels and the information regarding the image colors.

See also:

ColorspaceType

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 261 of file Clip.cpp.

5.1.3.13 bool Clip::isGrayscale () const

Returns true if the clip is in grayscale.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 273 of file Clip.cpp.

5.1.3.14 bool Clip::isColor () const

Returns true if the clip is in color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 285 of file Clip.cpp.

5.1.3.15 bool Clip::isMonochromatic () const

Returns true if the clips is in black and white.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 297 of file Clip.cpp.

5.1.3.16 Pixel Clip::getPixel (unsigned int *x*, unsigned int *y*) const

Returns the pixel at coordinates (x,y).

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 309 of file Clip.cpp.

5.1.3.17 void Clip::setPixel (unsigned int *x*, unsigned int *y*, double *grayLevel*)

Sets the gray level of a pixel at coordinates (x,y).

Sets the color of a pixel at coordinates (x,y).

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 321 of file Clip.cpp.

5.1.3.18 void Clip::setPixel (unsigned int *x*, unsigned int *y*, double *red*, double *green*, double *blue*)

Sets the color of a pixel at coordinates (x,y).

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 333 of file Clip.cpp.

5.1.3.19 void Clip::setPixel (unsigned int *x*, unsigned int *y*, bool *isForeground*)

Sets whether a pixel at coordinates (x,y) belongs to the foreground or not.

Sets the color of a pixel at coordinates (x,y).

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-24

Definition at line 345 of file Clip.cpp.

5.1.4 Field Documentation

5.1.4.1 ColorspaceType Clip::colorspace_ [private]

Clip's colorspace.

It depends on Magick++ implementation and it is modified automatically when the clip's image source changes

Definition at line 170 of file Clip.h.

5.1.4.2 Pixels* Clip::frame_ [private]

Pixels frame where the clip is located.

It is modified automatically when the clip attributes change

Definition at line 175 of file Clip.h.

5.1.4.3 PixelPacket* Clip::originPixel_ [private]

Frame's upper leftmost pixel.

It is modified automatically when information about the clip attributes change

Definition at line 180 of file Clip.h.

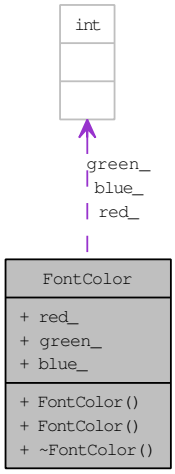
The documentation for this class was generated from the following files:

- Clip.h (107)
- Clip.cpp (107)

5.2 FontColor Struct Reference

`#include <FontColor.h>`

Collaboration diagram for FontColor:



5.2.1 Detailed Description

Font color of a character.

This struct represents the color of font that a character has. The color value is stored using the RGB color space with 256 values per component, and the three components separated.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 18 of file FontColor.h.

Public Member Functions

- FontColor ()

  Constructor.

- FontColor (unsigned int red, unsigned int green, unsigned int blue)

  Constructor.

- ~FontColor ()

  Destructor.

Data Fields

- unsigned int red_

  Red component.

- unsigned int green_

  Green component.

- unsigned int blue_

  Blue component.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 FontColor::FontColor ()

Constructor.

Initializes a FontColor object with all RGB components set to 0

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 10 of file FontColor.cpp.

5.2.2.2 FontColor::FontColor (unsigned int *red*, unsigned int *green*, unsigned int *blue*)

Constructor.

Initializes a FontColor object with the components set to the values passed. Since the color must be expressed using a RGB scale of 256 possible values, any value over 255 will be truncated

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 24 of file FontColor.cpp.

5.2.2.3 FontColor::~FontColor ()

Destructor.

Destroys a FontColor object

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-18

Definition at line 37 of file FontColor.cpp.

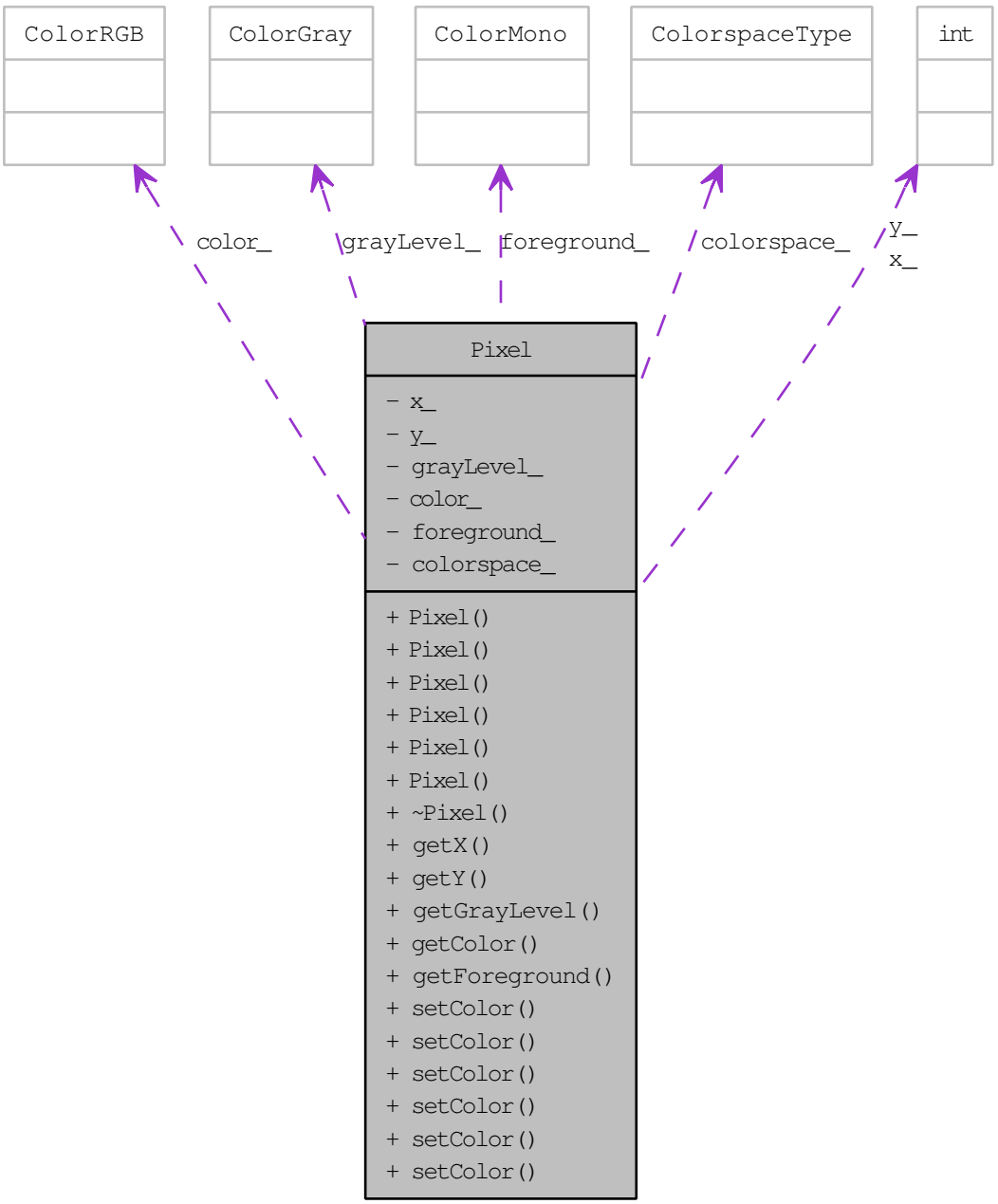
The documentation for this struct was generated from the following files:

- FontColor.h (110)
- FontColor.cpp (111)

5.3 Pixel Class Reference

`#include <Pixel.h>`

Collaboration diagram for Pixel:



5.3.1 Detailed Description

Information about an image pixel.

This class stores all the information relative to a pixel within a press clip, no matter if we are dealing with

the clip itself, or some other data structure which makes use of it. Every pixel has a location, given by a pair of coordinates x and y, and a color information, which may come in three formats: grayscale, RGB or monochromatic.

The Clip::getColorspace() method gives the current colorspace that is associated with the pixel and its underlying image, so that the right methods can be invoked. If a method related to one colorspace is called when the current colorspace is actually another, a null value is returned. See those methods to know what are the null values in each case.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 28 of file Pixel.h.

Public Member Functions

- Pixel (unsigned int x, unsigned int y, double grayLevel)
  Constructor.
- Pixel (unsigned int x, unsigned int y, ColorGray grayLevel)
  Constructor.
- Pixel (unsigned int x, unsigned int y, double red, double green, double blue)
  Constructor.
- Pixel (unsigned int x, unsigned int y, ColorRGB color)
  Constructor.
- Pixel (unsigned int x, unsigned int y, bool foreground)
  Constructor.
- Pixel (unsigned int x, unsigned int y, ColorMono foreground)
  Constructor.
- ~Pixel ()
  Destructor.
- unsigned int getX () const
  Returns the pixel's X coordinate.
- unsigned int getY () const
  Returns the pixel's Y coordinate.
- ColorGray getGrayLevel () const
  Returns the pixel's gray level.
- ColorRGB getColor () const

Returns the pixel's color.

- ColorMono getForeground () const

  Returns the pixel's foreground value.

- void setColor (ColorGray grayLevel)

  Sets the pixel's color.

- void setColor (double grayLevel)

  Sets the pixel's color.

- void setColor (ColorRGB color)

  Sets the pixel's color.

- void setColor (double red, double green, double blue)

  Sets the pixel's color.

- void setColor (ColorMono foreground)

  Sets the pixel's color.

- void setColor (bool foreground)

  Sets the pixel's color.

Private Attributes

- unsigned int x_

  Pixel's X coordinate.

- unsigned int y_

  Pixel's Y coordinate.

- ColorGray grayLevel_

  Color information when the colorspace comes in grayscale mode.

- ColorRGB color_

  Color information when the colorspace comes in RGB mode.

- ColorMono foreground_

  Color information when the colorspace comes in monochromatic mode.

- ColorspaceType colorspace_

  Pixel's colorspace.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Pixel::Pixel (unsigned int *x*, unsigned int *y*, double *grayLevel*)

Constructor.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 10 of file Pixel.cpp.

5.3.2.2 Pixel::Pixel (unsigned int *x*, unsigned int *y*, ColorGray *grayLevel*)

Constructor.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 22 of file Pixel.cpp.

5.3.2.3 Pixel::Pixel (unsigned int *x*, unsigned int *y*, double *red*, double *green*, double *blue*)

Constructor.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 34 of file Pixel.cpp.

5.3.2.4 Pixel::Pixel (unsigned int *x*, unsigned int *y*, ColorRGB *color*)

Constructor.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 46 of file Pixel.cpp.

5.3.2.5 Pixel::Pixel (unsigned int *x*, unsigned int *y*, bool *foreground*)

Constructor.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 58 of file Pixel.cpp.

5.3.2.6 Pixel::Pixel (unsigned int *x*, unsigned int *y*, ColorMono *foreground*)

Constructor.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 70 of file Pixel.cpp.

5.3.2.7 Pixel::~Pixel ()

Destructor.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 82 of file Pixel.cpp.

5.3.3 Member Function Documentation

5.3.3.1 unsigned int Pixel::getX () const

Returns the pixel's X coordinate.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 94 of file Pixel.cpp.

5.3.3.2 unsigned int Pixel::getY () const

Returns the pixel's Y coordinate.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 106 of file Pixel.cpp.

5.3.3.3 ColorGray Pixel::getGrayLevel () const

Returns the pixel's gray level.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 118 of file Pixel.cpp.

5.3.3.4 ColorRGB Pixel::getColor () const

Returns the pixel's color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 130 of file Pixel.cpp.

5.3.3.5 ColorMono Pixel::getForeground () const

Returns the pixel's foreground value.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 142 of file Pixel.cpp.

5.3.3.6 void Pixel::setColor (ColorGray *grayLevel*)

Sets the pixel's color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 154 of file Pixel.cpp.

5.3.3.7 void Pixel::setColor (double *grayLevel*)

Sets the pixel's color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 166 of file Pixel.cpp.

5.3.3.8 void Pixel::setColor (ColorRGB *color*)

Sets the pixel's color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 178 of file Pixel.cpp.

5.3.3.9 void Pixel::setColor (double *red*, double *green*, double *blue*)

Sets the pixel's color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 190 of file Pixel.cpp.

5.3.3.10 void Pixel::setColor (ColorMono *foreground*)

Sets the pixel's color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 202 of file Pixel.cpp.

5.3.3.11 void Pixel::setColor (bool *foreground*)

Sets the pixel's color.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 214 of file Pixel.cpp.

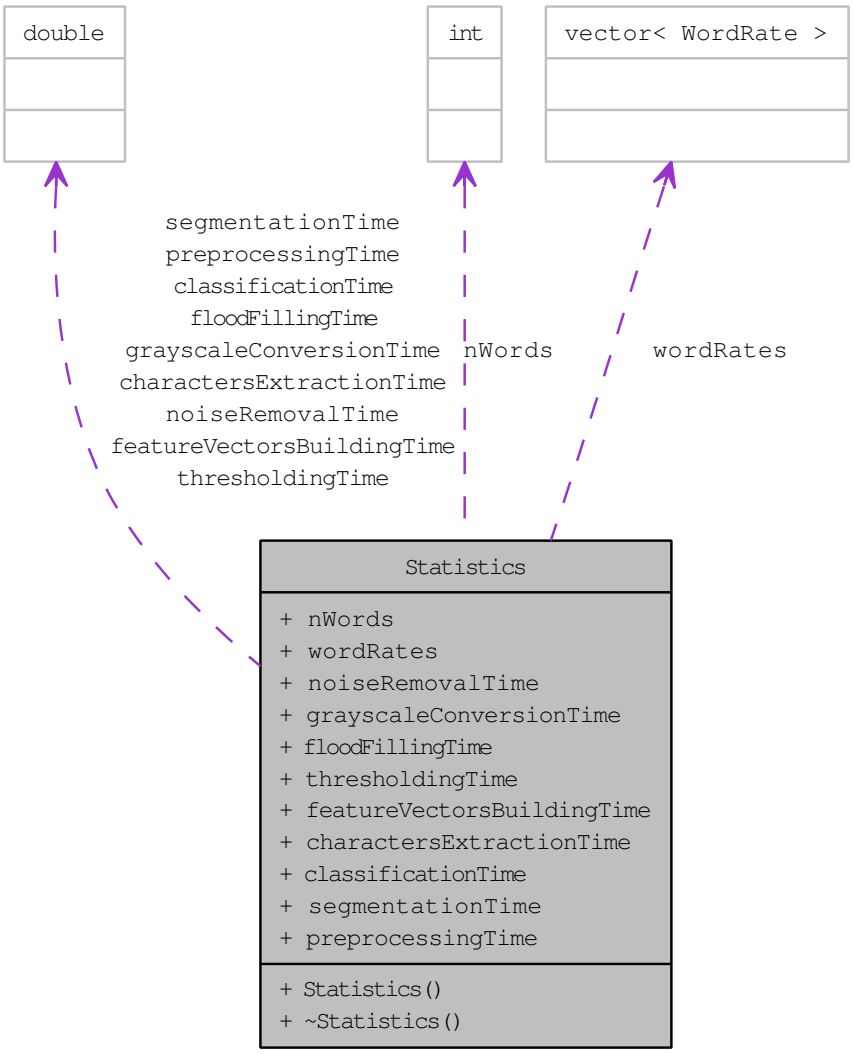
The documentation for this class was generated from the following files:

- Pixel.h (107)
- Pixel.cpp (107)

5.4 Statistics Struct Reference

`#include <Statistics.h>`

Collaboration diagram for Statistics:



5.4.1 Detailed Description

Statistics about a text and its recognition process.

This struct stores a number of statistic data regarding the recognition process and the text produced.

First, it stores the elapsed time on every stage, accumulated internally in the critical processes (mostly, image processing algorithms). Second, it also stores the number of words in the text and the appearance rate of every single word.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-18

Definition at line 28 of file Statistics.h.

Public Member Functions

- Statistics ()

  Constructor.

- ~Statistics ()

  Destructor.

Data Fields

- unsigned int nWords

  Number of words in the recognized text.

- vector< WordRate > wordRates

  List of appearance rates of every word in the recognized text.

- double noiseRemovalTime

  Elapsed time in the noise removal stage.

- double grayscaleConversionTime

  Elapsed time in the grayscale conversion stage.

- double floodFillingTime

  Elapsed time in the flood filling stage.

- double thresholdingTime

  Elapsed time in the thresholding stage.

- double featureVectorsBuildingTime

  Elapsed time in the feature vectors building stage.

- double charactersExtractionTime

  Elapsed time in the characters extraction stage.

- double classificationTime

  Returns the total time within the classification stage.

- double segmentationTime

  Returns the total time within the segmentation stage.

- double preprocessingTime

  Returns the total time within the preprocessing stage.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 Statistics::Statistics ()

Constructor.

Initializes a Statistics object with timers set to 0

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-22

Definition at line 10 of file Statistics.cpp.

5.4.2.2 Statistics::~Statistics ()

Destructor.

Destroys a Statistics object

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-17

Definition at line 25 of file Statistics.cpp.

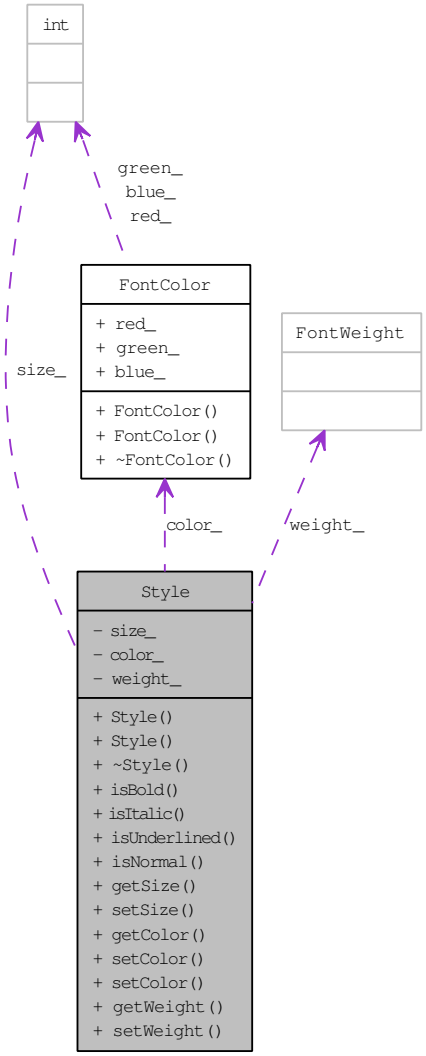
The documentation for this struct was generated from the following files:

- Statistics.h (111)
- Statistics.cpp (111)

5.5 Style Class Reference

`#include <Style.h>`

Collaboration diagram for Style:



5.5.1 Detailed Description

Font style of a character.

This class stores information about the font style of a character in the recognized text. Due to the different design styles in newspapers, a text may contain different kinds of style. That's why we must keep the style of every single character.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 23 of file Style.h.

Public Member Functions

- Style ()
  Constructor.
- Style (unsigned int size, FontColor color, FontWeight weight)
  Constructor.
- ~Style ()
  Destructor.
- bool isBold () const
  Returns true if the font weight is bold.
- bool isItalic () const
  Returns true if the font weight is italic.
- bool isUnderlined () const
  Returns true if the font weight is underlined.
- bool isNormal () const
  Returns true if the font weight is normal.
- unsigned int getSize () const
  Returns the font size.
- void setSize (unsigned int size)
  Sets the font size.
- FontColor getColor () const
  Returns the font color.
- void setColor (FontColor color)
  Sets the font color.
- void setColor (unsigned int red, unsigned int green, unsigned int blue)
  Sets the font color.
- FontWeight getWeight () const
  Returns the font weight.
- void setWeight (FontWeight weight)
  Sets the font weight.

Private Attributes

- unsigned int size_

  Size of the character in points.

- FontColor color_

  Color of the character.

- FontWeight weight_

  Font weight of the character.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 Style::Style ()

Constructor.

Initializes a Style object with size set to 0, color set to black and weight set to normal

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 10 of file Style.cpp.

5.5.2.2 Style::Style (unsigned int *size*, FontColor *color*, FontWeight *weight*)

Constructor.

Initializes a Style object with size, color and weight set to the values passed

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 22 of file Style.cpp.

5.5.2.3 Style::~Style ()

Destructor.

Destroys a Style object

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 34 of file Style.cpp.

5.5.3 Member Function Documentation

5.5.3.1 bool Style::isBold () const

Returns true if the font weight is bold.

A bold font may be only bold, bold and italic or bold and underlined

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 46 of file Style.cpp.

5.5.3.2 bool Style::isItalic () const

Returns true if the font weight is italic.

An italic font may be only italic, italic and bold or italic and underlined

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 70 of file Style.cpp.

5.5.3.3 bool Style::isUnderlined () const

Returns true if the font weight is underlined.

An underlined font may be only underlined, underlined and italic or underlined and bold

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 94 of file Style.cpp.

5.5.3.4 bool Style::isNormal () const

Returns true if the font weight is normal.

A normal font is neither bold, nor italic, nor underlined

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 118 of file Style.cpp.

5.5.3.5 unsigned int Style::getSize () const

Returns the font size.

The size is expressed in points (pt)

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 142 of file Style.cpp.

5.5.3.6 void Style::setSize (unsigned int *size*)

Sets the font size.

The size is expressed in points (pt)

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 154 of file Style.cpp.

5.5.3.7 FontColor Style::getColor () const

Returns the font color.

See also:

FontColor

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 166 of file Style.cpp.

5.5.3.8 void Style::setColor (FontColor *color*)

Sets the font color.

See also:

FontColor

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 178 of file Style.cpp.

5.5.3.9 void Style::setColor (unsigned int *red*, unsigned int *green*, unsigned int *blue*)

Sets the font color.

The font color is created internally as a FontColor object with the values passed

See also:

FontColor

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 192 of file Style.cpp.

5.5.3.10 FontWeight Style::getWeight () const

Returns the font weight.

See also:

FontWeight

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 204 of file Style.cpp.

5.5.3.11 void Style::setWeight (FontWeight *weight*)

Sets the font weight.

The weight must be passed using a literal value of FontWeight enumeration.

See also:

FontWeight

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 218 of file Style.cpp.

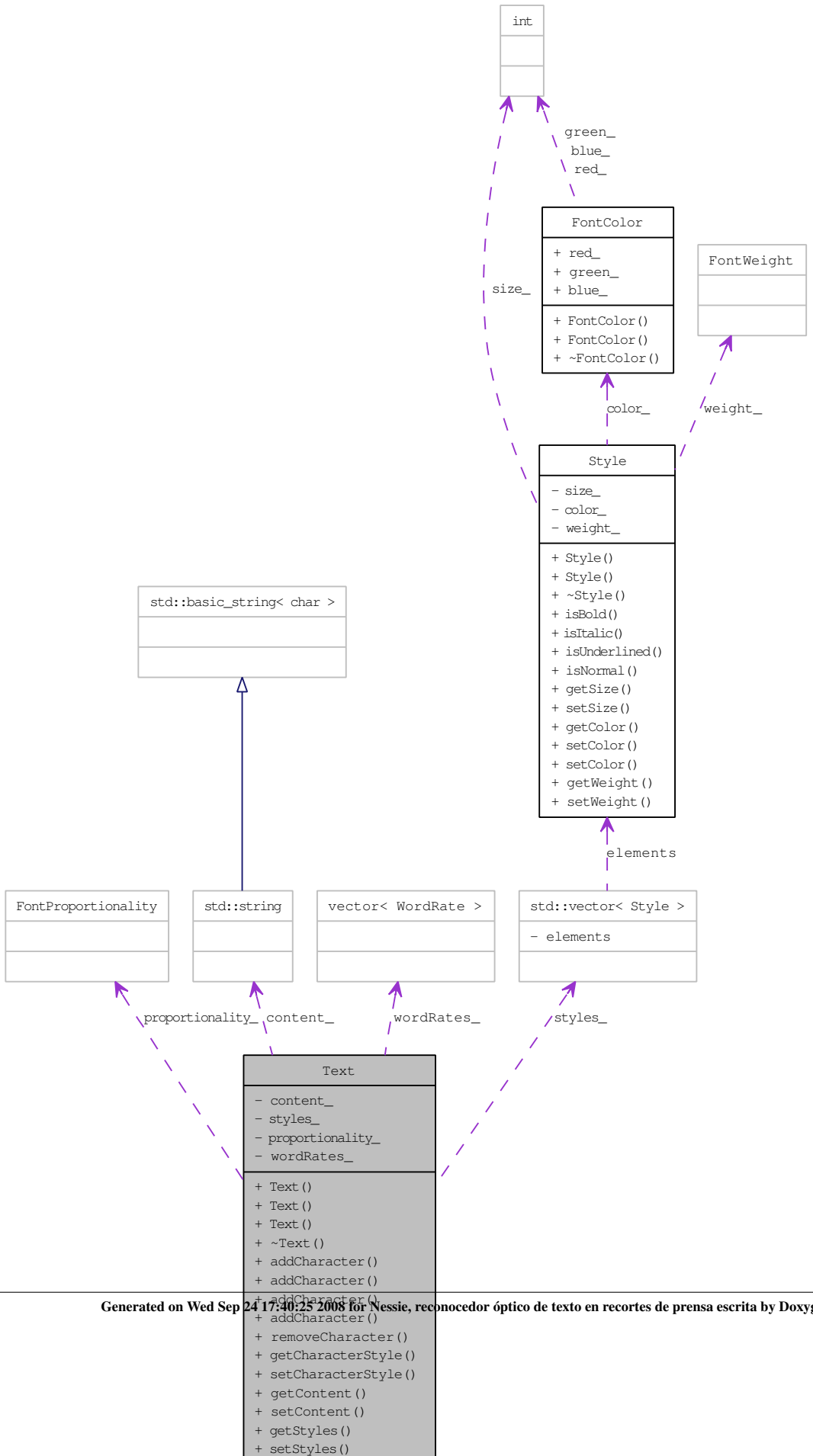
The documentation for this class was generated from the following files:

- Style.h (111)
- Style.cpp (111)

5.6 Text Class Reference

`#include <Text.h>`

Collaboration diagram for Text:



5.6.1 Detailed Description

Text extracted from the clip by the recognizer.

This class stores the text que has been extracted from the press-clip during the recognition process. Besides the text itself, it also keeps some extra information: appearance rate of every word, font style of every character and the font proportionality, among others.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 29 of file Text.h.

Public Member Functions

- Text ()

  Constructor.

- Text (string content)

  Constructor.

- Text (string content, vector< Style > styles)

  Constructor.

- ~Text ()

  Destructor.

- void addCharacter (char character)

  Adds a character to the text.

- void addCharacter (char character, Style style)

  Adds a character to the text.

- void addCharacter (char character, unsigned int position)

  Adds a character to the text.

- void addCharacter (char character, Style style, unsigned int position)

  Adds a character to the text.

- void removeCharacter (unsigned int position)

  Removes a character from the text.

- Style getCharacterStyle (unsigned int position) const

  Returns the style of a single character.

- void setCharacterStyle (unsigned int position, Style style)

Sets the style of a single character.

- string getContent () const
  Returns the text itself.
- void setContent (string content)
  Sets the text.
- vector< Style > getStyles () const
  Returns the list of text styles.
- void setStyles (vector< Style > styles)
  Sets the list of text styles.
- unsigned int getLength () const
  Returns the text length.
- vector< WordRate > getWordRates () const
  Returns the appearance rates of every single word in text.
- FontProportionality getProportionality () const
  Returns the font proportionality in text.
- void setProportionality (FontProportionality proportionality)
  Sets the font proportionality in text.

Private Member Functions

- void computeWordRates ()
  Builds the vector with every word appearance rate.
- void updateWordRate (string word_)
  Increases by one the number of appearances of a word.
- void tokenize (vector< string > &tokens_, const string &delimiters_=" ,.\n\t:;!¡¿?&/()=") const
  Extracts the words surrounded by default delimiters from text.

Private Attributes

- string content_
  The text itself.
- vector< Style > styles_
  A list of every character style.
- FontProportionality proportionality_
  The text proportionality.

- vector< WordRate > wordRates_

  A list of appearance rates of every single word in text.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 Text::Text ()

Constructor.

Initializes a Text object with empty content and no styles

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 10 of file Text.cpp.

5.6.2.2 Text::Text (string *content*)

Constructor.

Initializes a Text object with the content passed in content_ and no styles

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 23 of file Text.cpp.

5.6.2.3 Text::Text (string *content*, vector< Style > *styles*)

Constructor.

Initializes a Text object with the content passed in content_ and styles passed in styles_

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 39 of file Text.cpp.

5.6.2.4 Text::~Text ()

Destructor.

Destroys a Text object

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-18

Definition at line 52 of file Text.cpp.

5.6.3 Member Function Documentation

5.6.3.1 void Text::addCharacter (char *character*)

Adds a character to the text.

The character passed is appended to the end of the text with a default style.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-22

Definition at line 152 of file Text.cpp.

5.6.3.2 void Text::addCharacter (char *character*, Style *style*)

Adds a character to the text.

The character is appended to the end of the text with the style passed.

See also:

Style

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 139 of file Text.cpp.

5.6.3.3 void Text::addCharacter (char *character*, unsigned int *position*)

Adds a character to the text.

The character is appended to the text at position passed, with a default style. If the position passed is over the text total length, the character is appended to the end of the text.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 118 of file Text.cpp.

5.6.3.4 void Text::addCharacter (char *character*, Style *style*, unsigned int *position*)

Adds a character to the text.

The character is appended to the text at position passed in position_, and with style passed in style_. If no position is passed, the character is appended to the end. Similarly, if no style is passed a default style is assigned to the character. If the position passed is over the text total length, the character is appended to the end of the text.

See also:

Style

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 70 of file Text.cpp.

5.6.3.5 void Text::removeCharacter (unsigned int *position*)

Removes a character from the text.

The character is removed from the position in text passed in position_, unless it is over the text total length. In such case, the last character is removed

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 166 of file Text.cpp.

5.6.3.6 Style Text::getCharacterStyle (unsigned int *position*) const

Returns the style of a single character.

The style of a character at position passed is return unless the position will be over the text length. In this case an empty style is returned.

See also:

Style

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 206 of file Text.cpp.

5.6.3.7 void Text::setCharacterStyle (unsigned int *position*, Style *style*)

Sets the style of a single character.

The character at position position_ in text is set with the style passed. If the position passed in position_ is over the text total length no changes are made.

See also:

Style

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 228 of file Text.cpp.

5.6.3.8 string Text::getContent () const

Returns the text itself.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 241 of file Text.cpp.

5.6.3.9 void Text::setContent (string *content*)

Sets the text.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 251 of file Text.cpp.

5.6.3.10 vector< Style > Text::getStyles () const

Returns the list of text styles.

See also:

Style

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 267 of file Text.cpp.

5.6.3.11 void Text::setStyles (vector< Style > *styles*)

Sets the list of text styles.

See also:

Style

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 279 of file Text.cpp.

5.6.3.12 unsigned int Text::getLength () const

Returns the text length.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 289 of file Text.cpp.

5.6.3.13 vector< WordRate > Text::getWordRates () const

Returns the appearance rates of every single word in text.

See also:

WordRate

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Definition at line 301 of file Text.cpp.

5.6.3.14 FontProportionality Text::getProportionality () const

Returns the font proportionality in text.

The proportionality may be monospaced or proportional

See also:

FontProportionality

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-18

Definition at line 315 of file Text.cpp.

5.6.3.15 void Text::setProportionality (FontProportionality *proportionality*)

Sets the font proportionality in text.

See also:

FontProportionality

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-18

Definition at line 327 of file Text.cpp.

5.6.3.16 void Text::computeWordRates () [private]

Builds the vector with every word appearance rate.

This method must be called every time the content changes, since there is no public method for a class user to make it by itself. By the way, the number of words in text is also computed.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-22

Definition at line 364 of file Text.cpp.

5.6.3.17 void Text::updateWordRate (string *word*) [private]

Increases by one the number of appearances of a word.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-22

Definition at line 337 of file Text.cpp.

5.6.3.18 void Text::tokenize (vector< string > & *tokens*, const string & *delimiters* = " ,.\n\t:;!¡¿?&/()=") const [private]

Extracts the words surrounded by default delimiters from text.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-22

Definition at line 387 of file Text.cpp.

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

- Text.h (111)
- Text.cpp (112)

Chapter 6

File Documentation

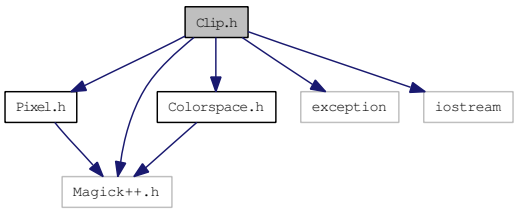
6.1 Clip.h File Reference

6.1.1 Detailed Description

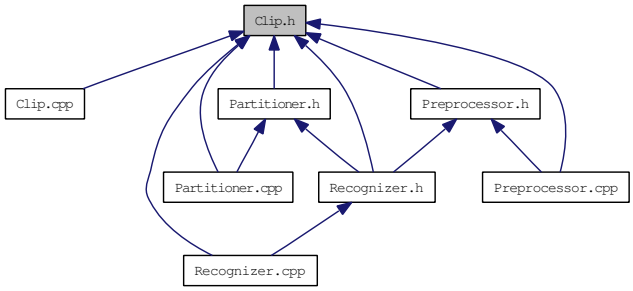
Definition in file Clip.h.

```
#include "Pixel.h"
#include "Colorspace.h"
#include <Magick++.h>
#include <exception>
#include <iostream>
```

Include dependency graph for Clip.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Clip

  Press clip where the recognizer has to extract the text from.

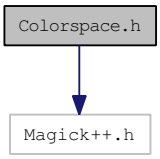
6.2 Colorspace.h File Reference

6.2.1 Detailed Description

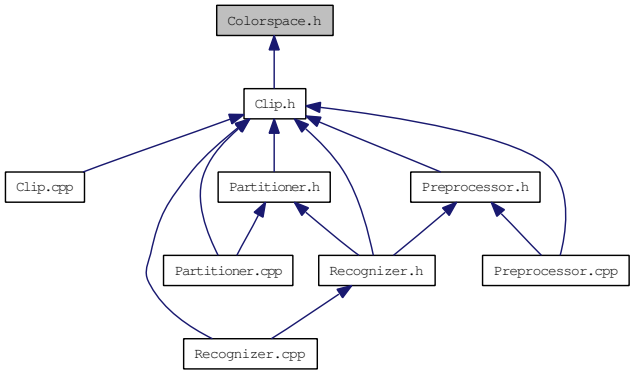
Definition in file Colorspace.h.

`#include <Magick++.h>`

Include dependency graph for Colorspace.h:



This graph shows which files directly or indirectly include this file:



Enumerations

- enum Colorspace { COLORSPACE_GRAYSCALE, COLORSPACE_RGB, COLORSPACE_-MONOCHROMATIC }

  Colorspace of an image.

6.2.2 Enumeration Type Documentation

6.2.2.1 enum Colorspace

Colorspace of an image.

This enumeration represents the different colorspaces that an image may have. This enumeration depends internally on Magick++ colorspaces implementation (see `ColorspaceType on Magick++ documentation`). Only three choices are given, though there are many colorspaces available in Magick++.

Author:

Eliezer Talón (`elitalon@gmail.com`)

Date:

2008-09-24

Enumerator:

COLORSPACE_GRAYSCALE A grayscale colorspace.

COLORSPACE_RGB A RGB colorspace.

COLORSPACE_MONOCHROMATIC A monochromatic colorspace.

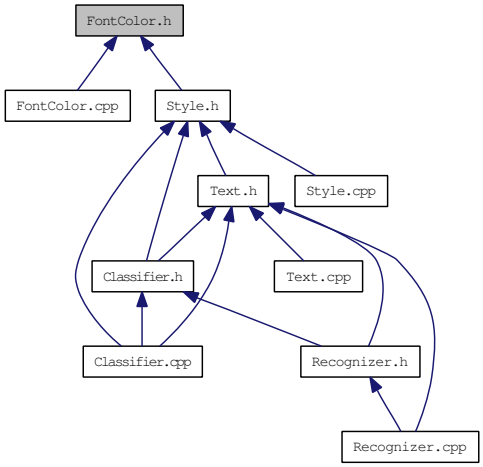
Definition at line 22 of file Colorspace.h.

6.3 FontColor.h File Reference

6.3.1 Detailed Description

Definition in file FontColor.h.

This graph shows which files directly or indirectly include this file:



Data Structures

- struct FontColor

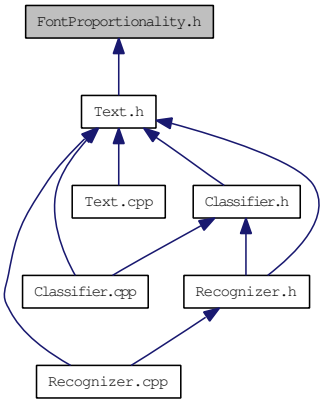
  Font color of a character.

6.4 FontProportionality.h File Reference

6.4.1 Detailed Description

Definition in file FontProportionality.h.

This graph shows which files directly or indirectly include this file:



Enumerations

- enum FontProportionality { FONT_MONOSPACED, FONT_PROPORTIONAL }

  Font proportionality of a character.

6.4.2 Enumeration Type Documentation

6.4.2.1 enum FontProportionality

Font proportionality of a character.

This enumeration represents the different types of font proportionality that a text may have. The monospaced type is usual in fixed width fonts, while proportional type is usual in Roman fonts.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Enumerator:

FONT_MONOSPACED In fixed width fonts.

FONT_PROPORTIONAL In Roman fonts.

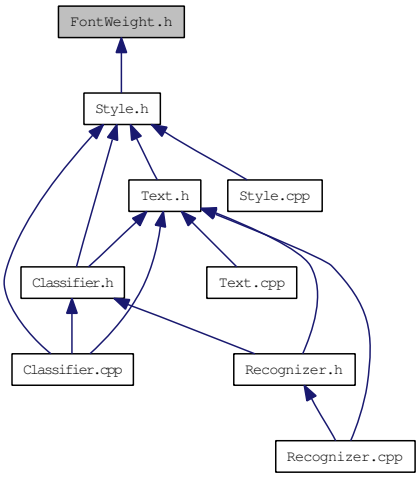
Definition at line 19 of file FontProportionality.h.

6.5 FontWeight.h File Reference

6.5.1 Detailed Description

Definition in file FontWeight.h.

This graph shows which files directly or indirectly include this file:



Enumerations

- enum FontWeight {
  FONT_NORMAL, FONT_BOLD, FONT_ITALIC, FONT_UNDERLINED,
  FONT_BOLD_ITALIC, FONT_BOLD_UNDERLINED, FONT_ITALIC_UNDERLINED }
  Font weight of a character.

6.5.2 Enumeration Type Documentation

6.5.2.1 enum FontWeight

Font weight of a character.

This enumeration represents the different types of font weight that a character may have. All the possible combinations are present, though in a normal text the most common are the three first font weights: bold, italic and normal.

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-23

Enumerator:

FONT_NORMAL Normal font, without decorations.

FONT_BOLD Only bold.

FONT_ITALIC Only italic.

FONT_UNDERLINED Normal but underlined.

FONT_BOLD_ITALIC Both bold and italic.

FONT_BOLD_UNDERLINED Both bold and underlined.

FONT_ITALIC_UNDERLINED Both italic and underlined.

Definition at line 18 of file FontWeight.h.

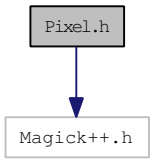
6.6 Pixel.h File Reference

6.6.1 Detailed Description

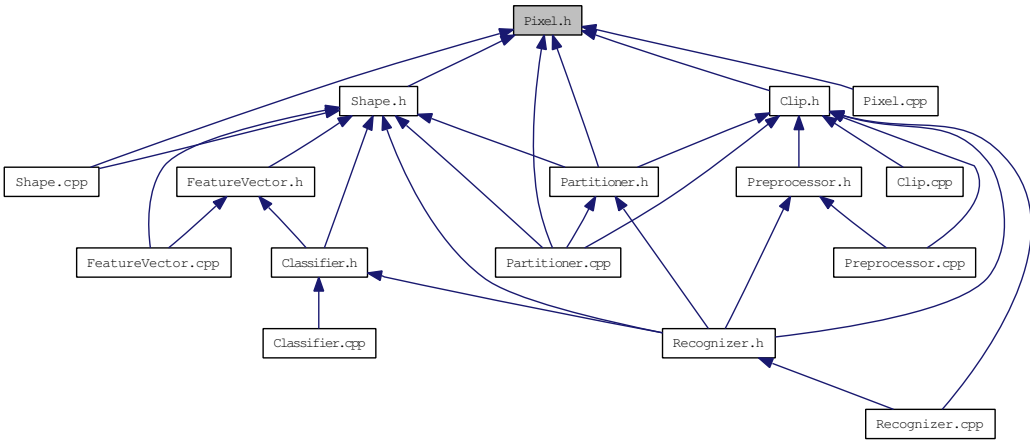
Definition in file Pixel.h.

```
#include <Magick++.h>
```

Include dependency graph for Pixel.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Pixel

  Information about an image pixel.

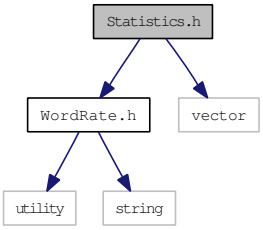
6.7 Statistics.h File Reference

6.7.1 Detailed Description

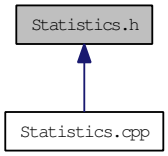
Definition in file Statistics.h.

```
#include "WordRate.h"
#include <vector>
```

Include dependency graph for Statistics.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct Statistics

  Statistics about a text and its recognition process.

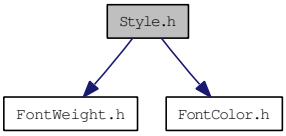
6.8 Style.h File Reference

6.8.1 Detailed Description

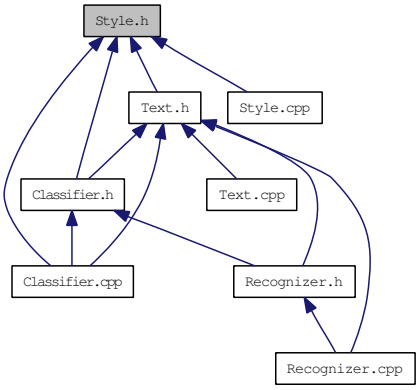
Definition in file Style.h.

```
#include "FontWeight.h"
#include "FontColor.h"
```

Include dependency graph for Style.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Style

  Font style of a character.

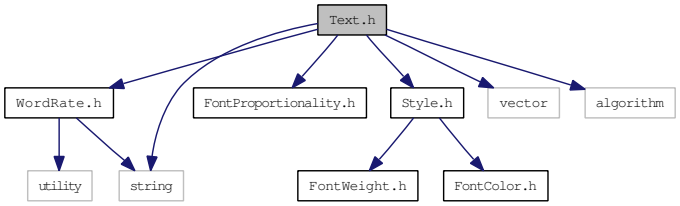
6.9 Text.h File Reference

6.9.1 Detailed Description

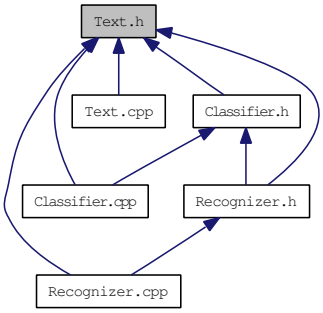
Definition in file Text.h.

```
#include "WordRate.h"
#include "FontProportionality.h"
#include "Style.h"
#include <string>
#include <vector>
#include <algorithm>
```

Include dependency graph for Text.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- class Text

  Text extracted from the clip by the recognizer.

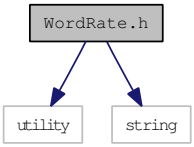
6.10 WordRate.h File Reference

6.10.1 Detailed Description

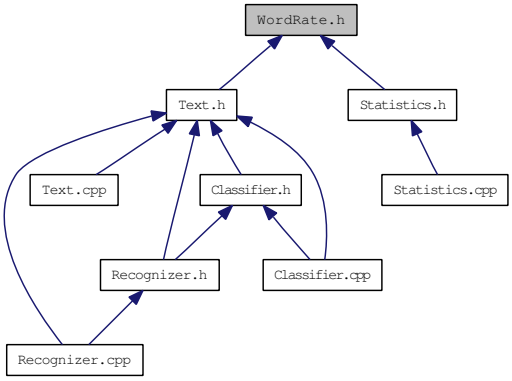
Definition in file WordRate.h.

```
#include <utility>
#include <string>
```

Include dependency graph for WordRate.h:



This graph shows which files directly or indirectly include this file:



Typedefs

- typedef pair< string, unsigned int > WordRate

  Appearance rate of a word.

6.10.2 Typedef Documentation

6.10.2.1 typedef pair<string, unsigned int> WordRate

Appearance rate of a word.

This pair keeps the number of appearances of a word in a text

Author:

Eliezer Talón (elitalon@gmail.com)

Date:

2008-09-19

Definition at line 20 of file WordRate.h.

Index

~Clip
  Clip, 13
~FontColor
  FontColor, 21
~Pixel
  Pixel, 27
~Statistics
  Statistics, 33
~Style
  Style, 36
~Text
  Text, 45

addCharacter
  Text, 46, 47

Clip, 9
  ~Clip, 13
  Clip, 13
  colorspace_, 19
  frame_, 19
  getColorspace, 16
  getHeight, 15
  getImage, 14
  getPixel, 18
  getWidth, 16
  getXOrigin, 14
  getYOrigin, 15
  isColor, 17
  isGrayscale, 17
  isMonochromatic, 17
  originPixel_, 19
  setColorspace, 16
  setHeight, 15
  setImage, 14
  setPixel, 18
  setWidth, 16
  setXOrigin, 14
  setYOrigin, 15
Clip.h(107), 53
Colorspace
  Colorspace.h, 55
Colorspace.h, 55
  Colorspace, 55
  COLORSPACE_GRAYSCALE, 56
  COLORSPACE_MONOCHROMATIC, 56
  COLORSPACE_RGB, 56
COLORSPACE_GRAYSCALE
  Colorspace.h, 56
COLORSPACE_MONOCHROMATIC
  Colorspace.h, 56
COLORSPACE_RGB
  Colorspace.h, 56
colorspace_
  Clip, 19
computeWordRates
  Text, 51

FONT_BOLD
  FontWeight.h, 59
FONT_BOLD_ITALIC
  FontWeight.h, 60
FONT_BOLD_UNDERLINED
  FontWeight.h, 60
FONT_ITALIC
  FontWeight.h, 60
FONT_ITALIC_UNDERLINED
  FontWeight.h, 60
FONT_MONOSPACED
  FontProportionality.h, 58
FONT_NORMAL
  FontWeight.h, 59
FONT_PROPORTIONAL
  FontProportionality.h, 58
FONT_UNDERLINED
  FontWeight.h, 60
FontColor, 20
  ~FontColor, 21
  FontColor, 21
FontColor.h(110), 57
FontProportionality
  FontProportionality.h, 58
FontProportionality.h
  FONT_MONOSPACED, 58
  FONT_PROPORTIONAL, 58
  FontProportionality, 58
FontProportionality.h(111), 58
FontWeight
  FontWeight.h, 59
FontWeight.h

FONT_BOLD, 59
FONT_BOLD_ITALIC, 60
FONT_BOLD_UNDERLINED, 60
FONT_ITALIC, 60
FONT_ITALIC_UNDERLINED, 60
FONT_NORMAL, 59
FONT_UNDERLINED, 60
FontWeight, 59
FontWeight.h(111), 59
frame_
Clip, 19

getCharacterStyle
Text, 47
getColor
Pixel, 28
Style, 38
getColorspace
Clip, 16
getContent
Text, 48
getForeground
Pixel, 28
getGrayLevel
Pixel, 28
getHeight
Clip, 15
getImage
Clip, 14
getLength
Text, 49
getPixel
Clip, 18
getProportionality
Text, 50
getSize
Style, 38
getStyles
Text, 49
getWeight
Style, 39
getWidth
Clip, 16
getWordRates
Text, 50
getX
Pixel, 27
getXOrigin
Clip, 14
getY
Pixel, 27
getYOrigin
Clip, 15

inc/ Directory Reference, 7
isBold
Style, 37
isColor
Clip, 17
isGrayscale
Clip, 17
isItalic
Style, 37
isMonochromatic
Clip, 17
isNormal
Style, 37
isUnderlined
Style, 37

originPixel_
Clip, 19

Pixel, 23
∼Pixel, 27
getColor, 28
getForeground, 28
getGrayLevel, 28
getX, 27
getY, 27
Pixel, 26, 27
setColor, 28–30
Pixel.h(107), 61

removeCharacter
Text, 47

setCharacterStyle
Text, 48
setColor
Pixel, 28–30
Style, 39
setColorspace
Clip, 16
setContent
Text, 48
setHeight
Clip, 15
setImage
Clip, 14
setPixel
Clip, 18
setProportionality
Text, 50
setSize
Style, 38
setStyles
Text, 49

setWeight
Style, 40
setWidth
Clip, 16
setXOrigin
Clip, 14
setYOrigin
Clip, 15
src/ Directory Reference, 8
Statistics, 31
∼Statistics, 33
Statistics, 33
Statistics.h(111), 62
Style, 34
∼Style, 36
getColor, 38
getSize, 38
getWeight, 39
isBold, 37
isItalic, 37
isNormal, 37
isUnderlined, 37
setColor, 39
setSize, 38
setWeight, 40
Style, 36
Style.h(111), 63

Text, 41
∼Text, 45
addCharacter, 46, 47
computeWordRates, 51
getCharacterStyle, 47
getContent, 48
getLength, 49
getProportionality, 50
getStyles, 49
getWordRates, 50
removeCharacter, 47
setCharacterStyle, 48
setContent, 48
setProportionality, 50
setStyles, 49
Text, 45
tokenize, 51
updateWordRate, 51
Text.h(111), 64
tokenize
Text, 51

updateWordRate
Text, 51

WordRate
WordRate.h, 65
WordRate.h
WordRate, 65
WordRate.h(111), 65