GuidoQt

1

Generated by Doxygen 1.7.2

Thu Oct 23 2014 11:29:32

Contents

Chapter 1

Main Page

Here's the documentation of the GUIDO Engine Library's Qt binding classes.

The architecture is divided in 3 levels :

- **Low level:** GSystemQt, GDeviceQt & GFontQt are the Qt implementations of the GUIDO interfaces VGSystem, VGDevice & VGFont.
- **Medium level:** The QGuidoPainter is a wrapper that uses GSystemQt, GDeviceQt & GFontQt.
- **High level:** QGuidoWidget & QGuidoGraphicsItem are ready-to-use QWidget/Q-GraphicsItem displaying a GUIDO Score.

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 GDeviceQt Class Reference

Qt implementation of the VGDevice interface, more precisely : a wrapper between the VGDevice and the QPainter objects.

```
#include <GDeviceQt.h>
```

3.1.1 Detailed Description

Qt implementation of the VGDevice interface, more precisely : a wrapper between the VGDevice and the QPainter objects.

Warning

Only the methods needed by the Guido Engine are implemented.

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

- GDeviceQt.h
- GDeviceQt.cpp

3.2 GFontQt Class Reference

Qt implementation of the VGFont interface.

```
#include <GFontQt.h>
```

Public Member Functions

- QFont $*$ GetNativeFont () const

Returns the font associated with the current object.

- QChar Symbol (unsigned int sym) const

  Returns the symbol corresponding to the input index.

3.2.1 Detailed Description

Qt implementation of the VGFont interface. More precisely : a wrapper between the VGFont interface and the QtFont object.

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

- GFontQt.h
- GFontQt.cpp

3.3 GSystemQt Class Reference

Qt implementation of the VGSystem interface.

```
#include <GSystemQt.h>
```

3.3.1 Detailed Description

Qt implementation of the VGSystem interface. For now, among the VGDevice factory functions, only the CreateDisplayDevice works, but you can use the created VGDevice to draw with any QPainter anyway (QPrinter, QWidget, QImage ...), so you needn't the other factory functions.

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

- GSystemQt.h
- GSystemQt.cpp

3.4 Guido2Image Class Reference

Offers functions to export GMN code (from a string or a file) to various formats of images, or to PDF.

```
#include <Guido2Image.h>
```

Static Public Member Functions

- static const char $*$ getErrorString (Guido2ImageErrorCodes err)

  gives an error code textual description

- static Guido2ImageErrorCodes gmnString2Image (const Params &p)

  converts a gmn string to an image

- static Guido2ImageErrorCodes gmnFile2Image (const Params &p)

  converts a gmn file to an image

- static Guido2ImageErrorCodes guidoPianoRoll2Image (const Params &p, PianoRoll ∗pianoRoll, int width, int height)

  converts a gmn file to an image

3.4.1 Detailed Description

Offers functions to export GMN code (from a string or a file) to various formats of images, or to PDF.

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

- Guido2Image.h
- Guido2Image.cpp

3.5 QGuidoGraphicsItem Class Reference

A QGraphicsItem displaying a Guido Score.

`#include <QGuidoGraphicsItem.h>`

Public Member Functions

- QGuidoGraphicsItem (QGraphicsItem ∗parent=0)

  Default constructor.

- virtual ∼QGuidoGraphicsItem ()

  Destructor.

- virtual bool setGMNFile (const QString &fileName)

  Sets the current Guido Score file to draw.

- QString fileName () const

  Returns the current Guido Score file.

- virtual bool setGMNCode (const QString &gmnCode, const QString &path=0)

  Sets the current Guido code that will be displayed by the guido item.

- QString gmnCode () const

  Returns the current Guido code.

- bool isGMNValid () const

  Returns the validity of the last GMN code loaded with setGMNCode or setGMNFile.

- QString getLastErrorMessage () const

  Returns a description of the last encountered error.

- void getLastParseErrorLine (int &line, int &col) const

  Gets the parse error line/col.

- virtual void setGuidoLayoutSettings (const GuidoLayoutSettings &layoutSettings)

  Sets the Guido layout settings used to draw with this QGuidoPainter.

- GuidoLayoutSettings guidoLayoutSettings () const

  Returns the Guido layout settings of the QGuidoPainter.

- void resetSystemsDistance ()

  sets the minimum systems distance to its default value

- void setSystemsDistance (float distance)

  sets the minimum systems distance

- float getSystemsDistance () const

  returns the minimum systems distance

- void setResizePageToMusic (bool isOn)

  Disable/enable automatic ResizePageToMusic.

- bool isResizePageToMusic () const

  Returns the state of the automatic ResizePageToMusic mode (enabled or disabled)

- void setGuidoPageFormat (const GuidoPageFormat &pageFormat)

  Sets the page format used when no page format is specified by the GMN.

- GuidoPageFormat guidoPageFormat () const

  Gets the page format used when no page format is specified by the GMN.

- int pageCount () const

  Returns the number of pages of the Guido Score.

- QSizeF pageSizeMM (int pageIndex) const

  Returns the size of a page (specified by its index), in millimeters.

- bool setPage (int pageIndex)

  Sets the first displayed page of the Guido Score.

- void setGridHeight (int height)

  Sets the number of the lines of the grid of pages.

- void setGridWidth (int width)

  Sets the number of the columns of the grid of pages.

- int gridHeight () const

  Returns the number of lines in the grid of pages.

- int gridWidth () const

  Returns the number of columns in the grid of pages.

- int firstVisiblePage () const

  Returns the first visible page index.

- int lastVisiblePage () const

  Returns the last visible page index.

- CGRHandler getGRHandler () const

  Gives access to the GRHandler (graphic representation) of the Score in read-only.

- CARHandler getARHandler () const

  Gives access to the ARHandler (abstract representation) of the Score in read-only.

- void setScoreColor (const QColor &color)

  sets the color used to draw the score

- const QColor & getScoreColor () const

  returns the color used to draw the score

3.5.1 Detailed Description

A QGraphicsItem displaying a Guido Score. The Guido Score may be loaded via a QString containing the GMN code (setGMNCode()), or via a QString containing the path to a GMN file (setFile()).

The pages of the Guido Score will be displayed in a "grid of pages":

- you can specify the number of columns and lines of this grid with the setGrid-Height / setGridWidth functions ;
- the pages are placed in the grid in increasing order of indexes ; the first page is at the top-left, the second page is placed at the right of the first page, and so on, until the end of the line, when it goes on on the next line ;

- you can specify the first (top left) displayed page with the setPage function.
- if the grid is too small to display all the Guido Score pages, it doesn't matter : other pages are simply not visible, and you have to use setPage to display them. See QPageManager for more details.

Warning

Don't forget to use QGuidoPainter's static startGuidoEngine method before building any QGuidoGraphicsItem, or else you'll have an assertion failed in the QGuidoGraphicsItem constructor.

3.5.2 Member Function Documentation

3.5.2.1 QString QGuidoGraphicsItem::gmnCode ( ) const

Returns the current Guido code.

Note

This will work only if the code has been set with setGMNCode. If the code has been loaded via setFile, this will return "".

3.5.2.2 QSizeF QGuidoGraphicsItem::pageSizeMM (int *pageIndex*) const

Returns the size of a page (specified by its index), in millimeters.

The page format & size are defined in the GMN code.

3.5.2.3 bool QGuidoGraphicsItem::setGMNCode (const QString & *gmnCode,* const QString & *path* = 0) `[virtual]`

Sets the current Guido code that will be displayed by the guido item.

Parameters

gmnCode	The Guido Music Notation code

Returns

true if the GMN code is valid.

3.5.2.4 bool QGuidoGraphicsItem::setGMNFile (const QString & *fileName*) `[virtual]`

Sets the current Guido Score file to draw.

Parameters

fileName	Full path to the Guido Score Notation file.

Returns

true if the file is a valid Guido Score file.

Note

If any GMN code has been previously set, it will be erased.

3.5.2.5 void QGuidoGraphicsItem::setGuidoLayoutSettings (const GuidoLayoutSettings & *layoutSettings*) [virtual]

Sets the Guido layout settings used to draw with this QGuidoPainter.

Note

You can have more informations on GuidoLayoutSettings in GUIDOlib documentation.

3.5.2.6 bool QGuidoGraphicsItem::setPage (int *pageIndex*)

Sets the first displayed page of the Guido Score.

Returns

True if the pageIndex is valid, false else.

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

- QGuidoGraphicsItem.h
- QGuidoGraphicsItem.cpp

3.6 QGuidoImporter Class Reference

An importer to support the MusicXML format.

`#include <QGuidoImporter.h>`

Static Public Member Functions

- static bool musicxmlSupported ()
- static const char ∗ musicxmlVersion ()
- static const char ∗ musicxml2guidoVersion ()
- static bool musicxmlFile2Guido (const char ∗file, bool generateBars, std::ostream &out)

  converts a musicxml file to guido

- static bool musicxmlString2Guido (const char ∗str, bool generateBars, std::ostream &out)

    converts a musicxml string to guido

3.6.1 Detailed Description

An importer to support the MusicXML format. The QGuidoImporter is a static object. When initialized, it checks for the libmusicml2 library, and when present, it loads the library and resolves musicxml to guido conversion entry points.

3.6.2 Member Function Documentation

3.6.2.1 const char ∗ QGuidoImporter::musicxml2guidoVersion ( ) `[static]`

Returns

the musicxml to guido converter version as a string

3.6.2.2 bool QGuidoImporter::musicxmlFile2Guido (const char ∗ *file,* bool *generateBars,* std::ostream & *out*) `[static]`

converts a musicxml file to guido

Parameters

file	the musicxml file name
generateBars	a boolean to force or inhibit measures bar generation
out	the output stream

Returns

true when the conversion is successful

3.6.2.3 bool QGuidoImporter::musicxmlString2Guido (const char ∗ *str,* bool *generateBars,* std::ostream & *out*) `[static]`

converts a musicxml string to guido

Parameters

str	the musicxml string
generateBars	a boolean to force or inhibit measures bar generation
out	the output stream

Returns

true when the conversion is successful

3.6.2.4 bool QGuidoImporter::musicxmlSupported ( ) `[static]`

Returns

true when the conversion methods are available

3.6.2.5 const char ∗ QGuidoImporter::musicxmlVersion ( ) `[static]`

Returns

the musicxml lib version as a string

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

- QGuidoImporter.h
- QGuidoImporter.cpp

3.7 QGuidoPainter Class Reference

The QGuidoPainter object is a Qt encapsulation of the Guido Engine, basically allowing you to draw a Guido Score with a QPainter.

`#include <QGuidoPainter.h>`

Public Member Functions

- bool setGMNFile (const QString &fileName)

  Sets the current Guido code to draw with the content of the file.
- const QString & fileName () const

  Returns the last file loaded with setFile.
- bool setGMNCode (const QString &gmnCode, const char ∗datapath=0)

  Sets the current Guido code to draw.
- QString gmnCode () const

  Returns the current Guido code.
- bool isGMNValid () const

  Returns the validity of the last GMN code loaded with setGMNCode or setGMNFile.

- int pageCount () const

  Returns the number of page of the current Guido Score.

- void draw (QPainter ∗painter, int page, const QRect &drawRectangle, const QRect &redrawRectangle=QRect())

  Draws the current Guido Score using the specified QPainter.

- void drawPianoRoll (QPainter ∗painter, const QRect &drawRectangle, PianoRoll ∗pianoRoll)

  Draws the current Guido Score using the specified QPainter, under Piano Roll form.

- int heightForWidth (int w, int page) const

  Returns the height corresponding to the specified width for the specified page, according to the page format.

- QSizeF pageSizeMM (int page) const

  Returns the size of the specified page, in millimeters.

- QString getLastErrorMessage () const

  Returns a description of the last encountered error.

- void getLastParseErrorLine (int &line, int &col) const

  Gets the parse error line/col.

- void setGuidoLayoutSettings (const GuidoLayoutSettings &layoutSettings)

  sets the guido layout settings

- GuidoLayoutSettings guidoLayoutSettings () const

  returns the guido layout settings

- void setScoreColor (const QColor &color)

  sets the color used to draw the score

- const QColor & getScoreColor () const

  returns the color used to draw the score

- void resetSystemsDistance ()

  sets the minimum systems distance to its default value

- void setSystemsDistance (float distance)

  sets the minimum systems distance

- float getSystemsDistance () const

  returns the minimum systems distance

- void setResizePageToMusic (bool isOn)

Disable/enable automatic ResizePageToMusic.

- bool isResizePageToMusic () const

  Returns the state of the automatic ResizePageToMusic mode (enabled or disabled)

- void setGuidoPageFormat (const GuidoPageFormat &pageFormat)

  Sets the page format used when no page format is specified by the GMN.

- GuidoPageFormat guidoPageFormat () const

  Gets the page format used when no page format is specified by the GMN.

- CGRHandler getGRHandler () const

  Gives access to the GRHandler (graphic representation) of the Score in read-only.

- CARHandler getARHandler () const

  Gives access to the ARHandler (abstract representation) of the Score in read-only.

- void setARHandler (ARHandler ar)

  Directly set the AR handler.

Static Public Member Functions

- static void startGuidoEngine ()

  Initialize the GUIDO score engine.

- static QGuidoPainter ∗ createGuidoPainter ()

  Creates a new QGuidoPainter object.

- static void destroyGuidoPainter (QGuidoPainter ∗painter)

  Destroys the specified QGuidoPainter.

- static void stopGuidoEngine ()

  Stops the GUIDO score engine.

Static Protected Member Functions

- static bool isGuidoEngineStarted ()

  Returns the GuidoEngine state : started or not.

3.7.1 Detailed Description

The QGuidoPainter object is a Qt encapsulation of the Guido Engine, basically allowing you to draw a Guido Score with a QPainter. You first specifie the Guido Score file with the setGMNFile or setGMNCode methods, and then just call the draw method, specifying a QPainter, draw bounding rect, and a page index.

Note

QGuidoPainter constructor and destructor are protected. You must use the factory function createGuidoPainter to build one, and destroyGuidoPainter to destroy one. You must call startGuidoEngine and destroyGuidoEngine at the beginning and the end of your application.

Warning

You can NOT re-start the GuidoEngine once you've already stopped it.

3.7.2 Member Function Documentation

3.7.2.1 QGuidoPainter ∗ QGuidoPainter::createGuidoPainter ( ) `[static]`

Creates a new QGuidoPainter object.

Returns

a pointer to the new QGuidoPainter object, or NULL if you didn't previously called the startGuidoEngine function.

3.7.2.2 void QGuidoPainter::destroyGuidoPainter (QGuidoPainter ∗ *painter*) `[static]`

Destroys the specified QGuidoPainter.

If the specified QGuidoPainter is NULL, does nothing.

Note

You mustn't call "delete" in you own application ; you have to use the destroyGuidoPainter function to avoid shared-library memory problems.

3.7.2.3 void QGuidoPainter::draw (QPainter ∗ *painter,* int *page,* const QRect & *drawRectangle,* const QRect & *redrawRectangle* = `QRect()`)

Draws the current Guido Score using the specified QPainter.

The Guido Score won't be streched and will keep its width/height ratio.

Parameters

painter	The QPainter to be used for the draw.
page	Index of the score page to draw (starts with 1).
drawRectangle	Specifies the zone of the QPaintDevice in which to draw.
redrawRectangle	(optionnal) Specifies the rectangle to be redrawn. A null redrawRectangle will redraw everything.

Note

drawRectangle and redrawRectangle are in QPainter's QPaintDevice coordinates.

3.7.2.4 void QGuidoPainter::drawPianoRoll (QPainter $*$ *painter,* const QRect & *drawRectangle,* PianoRoll $*$ *pianoRoll*)

Draws the current Guido Score using the specified QPainter, under Piano Roll form.

The Guido Score won't be streched and will keep its width/height ratio.

Parameters

painter	The QPainter to be used for the draw.
drawRectangle	Specifies the zone of the QPaintDevice in which to draw.
pianoRoll	the current PianoRoll

Note

drawRectangle is in QPainter's QPaintDevice coordinates.

3.7.2.5 GuidoLayoutSettings QGuidoPainter::guidoLayoutSettings ( ) const

returns the guido layout settings

See also

GUIDOEngine interface

3.7.2.6 int QGuidoPainter::heightForWidth (int *w,* int *page*) const

Returns the height corresponding to the specified width for the specified page, according to the page format.

The page format & size are defined in the Guido Score file.

3.7.2.7 bool QGuidoPainter::isGuidoEngineStarted ( ) `[static, protected]`

Returns the GuidoEngine state : started or not.

3.7.2.8 QSizeF QGuidoPainter::pageSizeMM (int *page*) const

Returns the size of the specified page, in millimeters.

The page format & size are defined in the Guido Score file.

3.7.2.9 bool QGuidoPainter::setGMNCode (const QString & *gmnCode,* const char ∗ *datapath* = `0`)

Sets the current Guido code to draw.

Parameters

gmnCode	The Guido Music Notation code

Returns

true if the GMN code is valid.

3.7.2.10 bool QGuidoPainter::setGMNFile (const QString & *fileName*)

Sets the current Guido code to draw with the content of the file.

Parameters

fileName	Full path to the Guido Score Notation file.

Returns

true if the file is a valid Guido Score file.

3.7.2.11 void QGuidoPainter::setGuidoLayoutSettings (const GuidoLayoutSettings & *layoutSettings*)

sets the guido layout settings

See also

GUIDOEngine interface

3.7.2.12 void QGuidoPainter::startGuidoEngine ( ) `[static]`

Initialize the GUIDO score engine.

You must call this function to be able to instanciate QGuidoPainter objects, or else the createGuidoPainter function will return NULL.

Note

Calling this function more than once doesn't affect the score engine.

3.7.2.13 void QGuidoPainter::stopGuidoEngine ( ) `[static]`

Stops the GUIDO score engine.

Note

You must call the function at the end of your application to free the internal Guido score engine objects.

Warning

You mustn't call this function before every QGuidoPainter objects have been destroyed.

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

- QGuidoPainter.h
- QGuidoPainter.cpp

3.8 QGuidoWidget Class Reference

A QWidget displaying one/several pages of a Guido Score.

`#include <QGuidoWidget.h>`

Public Member Functions

- QGuidoWidget (QWidget *parent=0)

  Constructor.

- bool setGMNFile (const QString &fileName)

  Sets the current Guido Score file to draw.

- QString fileName () const

  Returns the current Guido Score file.

- bool setGMNCode (const QString &gmnCode, const QString &path)

  Sets the current Guido code to draw.

- QString gmnCode () const

  Returns the current Guido code.

- bool isGMNValid () const

  Returns the validity of the last GMN code loaded with setGMNCode or setGMNFile.

- int pageCount () const

Returns the number of pages of the current Guido Score.

- bool setPage (int pageIndex)

  Sets the first displayed page of the Guido Score.

- void setGridHeight (int height)

  Sets the number of the lines of the grid of pages.

- void setGridWidth (int width)

  Sets the number of the columns of the grid of pages.

- int gridHeight () const

  Returns the number of lines in the grid of pages.

- int gridWidth () const

  Returns the number of columns in the grid of pages.

- int firstVisiblePage () const

  Returns the first visible page index.

- int lastVisiblePage () const

  Returns the last visible page index.

- int heightForWidth (int w) const

  Returns the height corresponding to the specified width.

- QSizeF pageSizeMM (int page) const

  Returns the size of the specified page, in millimeters.

- QString getLastErrorMessage () const

  Returns a description of the last encountered error.

- void getLastParseErrorLine (int &line, int &col) const

  Gets the parse error line/col.

- void setGuidoLayoutSettings (const GuidoLayoutSettings &layoutSettings)

  Sets the Guido layout settings used to draw with this QGuidoPainter.

- GuidoLayoutSettings guidoLayoutSettings () const

  Returns the Guido layout settings of the QGuidoPainter.

- void resetSystemsDistance ()

  sets the minimum systems distance to its default value

- void setSystemsDistance (float distance)

  sets the minimum systems distance

- float getSystemsDistance () const

  returns the minimum systems distance

- void setResizePageToMusic (bool isOn)

  Disable/enable automatic ResizePageToMusic.

- bool isResizePageToMusic () const

  Returns the state of the automatic ResizePageToMusic mode (enabled or disabled)

- void setGuidoPageFormat (const GuidoPageFormat &pageFormat)

  Sets the page format used when no page format is specified by the GMN.

- GuidoPageFormat guidoPageFormat () const

  Gets the page format used when no page format is specified by the GMN.

- QSize sizeHint () const

  QWidget implementation. See Qt doc on QWidget.

- CGRHandler getGRHandler () const

  Gives access to the GRHandler (graphic representation) of the Score in read-only.

- CARHandler getARHandler () const

  Gives access to the ARHandler (abstract representation) of the Score in read-only.

- void setARHandler (ARHandler ar)

  Directly set the AR handler .

- void setScoreColor (const QColor &color)

  sets the color used to draw the score

- const QColor & getScoreColor () const

  returns the color used to draw the score

- void clearCache ()

  Clears the widget's draw-cache, forcing it to redraw.

Protected Member Functions

- void paintEvent (QPaintEvent $*$event)

  QWidget implementation.

- void updateGuidoPagesSizes ()

  Must be called when the GR has changed.

- QPixmap generatePixmap ()

  Generates a pixmap with a score.

3.8.1 Detailed Description

A QWidget displaying one/several pages of a Guido Score. You can navigate through the pages of the score using setPage function. You can have information on the number of pages in the score (pageCount()) and the format of the pages (pageSizeMM(int), heightForWidth()).

The pages of the Guido Score will be displayed in a "grid of pages":

- you can specify the number of columns and lines of this grid with the setGrid-Height / setGridWidth functions ;
- the pages are placed in the grid in increasing order of indexes ; the first page is at the top-left, the second page is placed at the right of the first page, and so on, until the end of the line, when it goes on on the next line ;
- you can specify the first (top left) displayed page with the setPage function.
- if the grid is too small to display all the Guido Score pages, it doesn't matter : other pages are simply not visible, and you have to use setPage to display them. See QPageManager for more details.

Warning

Don't forget to use QGuidoPainter's static startGuidoEngine method, or else you'll have an assertion failed in the QGuidoWidget constructor.

3.8.2 Member Function Documentation

3.8.2.1 QString QGuidoWidget::gmnCode ( ) const

Returns the current Guido code.

Note

This will work only if the code has been set with setGMNCode. If the code has been loaded via setFile, this will return "".

3.8.2.2 QSizeF QGuidoWidget::pageSizeMM (int *page*) const

Returns the size of the specified page, in millimeters.

The page format & size are defined in the Guido Score file.

3.8.2.3 bool QGuidoWidget::setGMNCode (const QString & *gmnCode,* const QString & *path*)

Sets the current Guido code to draw.

Parameters

gmnCode	The Guido Music Notation code
path	The path to external ressources (typically the file path for file based code)

Returns

true if the GMN code is valid.

3.8.2.4 bool QGuidoWidget::setGMNFile (const QString & *fileName*)

Sets the current Guido Score file to draw.

Parameters

fileName	Full path to the Guido Score Notation file.

Returns

true if the file is a valid Guido Score file.

Note

If any GMN code has been previously set, it will be erased.

3.8.2.5 void QGuidoWidget::setGuidoLayoutSettings (const GuidoLayoutSettings & *layoutSettings*)

Sets the Guido layout settings used to draw with this QGuidoPainter.

Note

You can have more informations on GuidoLayoutSettings in GUIDOlib documentation.

3.8.2.6 bool QGuidoWidget::setPage (int *pageIndex*)

Sets the first displayed page of the Guido Score.

Returns

True if the pageIndex is valid, false else.

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

- QGuidoWidget.h
- QGuidoWidget.cpp

3.9 QPageManager Class Reference

Arranges a set of pages in a grid.

`#include <QPageManager.h>`

Public Member Functions

- QPageManager ()

  Default constructor.
- virtual ∼QPageManager ()

  Destructor.
- void setPages (const QList< QSizeF > &pages)

  Sets the set of pages.
- void setGridHeight (int height)

  Sets the height of the grid (<=> number of lines)
- void setGridWidth (int width)

  Sets the width of the grid (<=> number of columns)
- bool setPage (int index)

  Sets the index of the first visible page.
- QSizeF pageSize (int index) const

  Returns the size of the page (as defined by setPages)
- QPointF pagePos (int pageIndex) const

  Returns the position of the page.
- QSizeF totalSize () const

  Returns the current total size of the grid of pages.
- float lineHeight (int lineIndex) const

  Returns the height of a line defined by its index.
- float columnWidth (int columnIndex) const

  Returns the width of a column defined by its index.
- int firstVisiblePage () const

  Returns the first visible page (top-left of the grid) index.
- int lastVisiblePage () const

  Returns the last visible page (bottom-right of the grid) index.

- int gridWidth () const

  Returns the grid's width (<=> number of columns)

- int gridHeight () const

  Returns the grid's height (<=> number of lines)

3.9.1 Detailed Description

Arranges a set of pages in a grid. Basically: 1. give a set of pages to the QPageManager (setPages) (a page is defined by its size), 2. specify the number of lines/columns of the grid of pages (setGridHeight / setGridWidth) ; 3. define the first visible page (setPage) ; 4. then the QPageManager can tell you the position of each page in the grid.

The pages are placed in the grid in increasing order of indexes ; the first page is at the top-left, the second page is placed at the right of the first page, and so on, until the end of the line, when it goes on on the next line.

The total number of pages may be greater than gridWidth() $*$ gridHeight() ; you can get the firstVisiblePage() and the lastVisiblePage(). Other pages are just considered as non-visible at that moment.

Each line has its own height, which is the one of the highest item of the line. Each column has its own width, which is the width of the item with the biggest width of the column.

Notes:

- lineIndex & columnIndex start at 0.
- pageIndex starts at 1 (like in a book).

3.9.2 Member Function Documentation

3.9.2.1 QPointF QPageManager::pagePos (int *pageIndex*) const

Returns the position of the page.

If the page is not visible, returns (-1,-1).

3.9.2.2 QSizeF QPageManager::pageSize (int *index*) const

Returns the size of the page (as defined by setPages)

Warning

The index parameter starts with 1 and no more with 0 (in opposition with the setPages function)

3.9.2.3 void QPageManager::setGridHeight (int *height*)

Sets the height of the grid (<=> number of lines)

If invalid argument (<=0), does nothing.

3.9.2.4 void QPageManager::setGridWidth (int *width*)

Sets the width of the grid (<=> number of columns)

If invalid argument (<=0), does nothing.

3.9.2.5 bool QPageManager::setPage (int *index*)

Sets the index of the first visible page.

Returns

False if invalid index.

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

- QPageManager.h
- QPageManager.cpp

Index