Large annotated image collection management

Generated by Doxygen 1.8.11

Contents

Chapter 1

Large-annotated-image-collection-management

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 Ui Namespace Reference

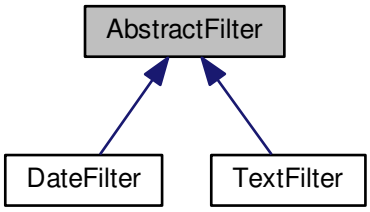
Chapter 7

Class Documentation

7.1 AbstractFilter Class Reference

`#include <AbstractFilter.hpp>`

Inheritance diagram for AbstractFilter:



Public Member Functions

- virtual ~AbstractFilter ()
- virtual AbstractFilter * makeFilter (const DbContext &dbContext)=0
- virtual QWidget * makeControl ()=0
- virtual QPushButton & removeButton ()=0

7.1.1 Detailed Description

Definition at line 10 of file AbstractFilter.hpp.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 virtual AbstractFilter::~AbstractFilter ( ) [inline], [virtual]

Definition at line 12 of file AbstractFilter.hpp.

7.1.3 Member Function Documentation

7.1.3.1 virtual QWidget* AbstractFilter::makeControl ( ) [pure virtual]

Implemented in DateFilter, and TextFilter.

7.1.3.2 virtual AbstractFilter* AbstractFilter::makeFilter (const DbContext & *dbContext*) [pure virtual]

Implemented in DateFilter, and TextFilter.

7.1.3.3 virtual QPushButton& AbstractFilter::removeButton ( ) [pure virtual]

Implemented in DateFilter, and TextFilter.

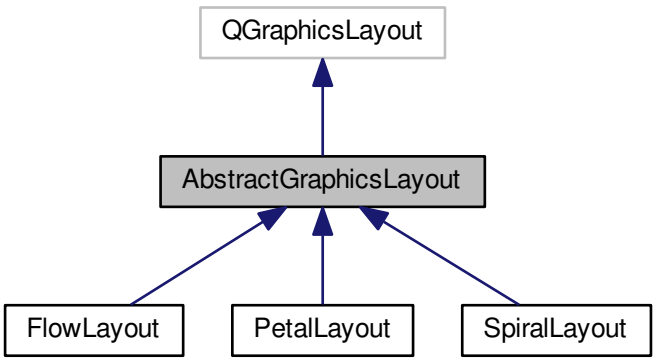
The documentation for this class was generated from the following file:

- filters/AbstractFilter.hpp

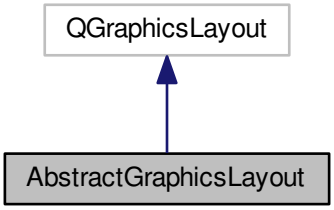
7.2 AbstractGraphicsLayout Class Reference

`#include <AbstractGraphicsLayout.hpp>`

Inheritance diagram for AbstractGraphicsLayout:



Collaboration diagram for AbstractGraphicsLayout:



Public Member Functions

- virtual void setGeometry (const QRectF &geom)=0
- virtual int count () const =0
- virtual QGraphicsLayoutItem * itemAt (int index) const =0
- virtual void removeAt (int index)=0
- virtual void addItem (QGraphicsLayoutItem *item)=0
- virtual void clearAll ()=0
- virtual QList< QGraphicsLayoutItem * > & items ()=0

Protected Member Functions

- AbstractGraphicsLayout ()
- virtual void insertItem (int index, QGraphicsLayoutItem *item)=0
- virtual void setSpacing (Qt::Orientations o, qreal spacing)=0
- virtual qreal spacing (Qt::Orientation o) const =0
- virtual QSizeF sizeHint (Qt::SizeHint which, const QSizeF &constraint=QSizeF()) const =0
- virtual qreal doLayout (const QRectF &geom, bool applyNewGeometry) const =0
- virtual QSizeF minSize (const QSizeF &constraint) const =0
- virtual QSizeF prefSize () const =0
- virtual QSizeF maxSize () const =0

7.2.1 Detailed Description

Definition at line 9 of file AbstractGraphicsLayout.hpp.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 AbstractGraphicsLayout::AbstractGraphicsLayout ( ) [inline],[protected]

Definition at line 25 of file AbstractGraphicsLayout.hpp.

7.2.3 Member Function Documentation

7.2.3.1 virtual void AbstractGraphicsLayout::addItem (QGraphicsLayoutItem * *item*) [pure virtual]

inserts the item to the front of the list

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.2 virtual void AbstractGraphicsLayout::clearAll ( ) [pure virtual]

deletes and removes every element from the layout

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.3 virtual int AbstractGraphicsLayout::count ( ) const [pure virtual]

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.4 virtual qreal AbstractGraphicsLayout::doLayout (const QRectF & *geom,* bool *applyNewGeometry*) const [protected], [pure virtual]

7.2.3.5 virtual void AbstractGraphicsLayout::insertItem (int *index,* QGraphicsLayoutItem * *item*) [protected], [pure virtual]

Implemented in FlowLayout.

7.2.3.6 virtual QGraphicsLayoutItem* AbstractGraphicsLayout::itemAt (int *index*) const [pure virtual]

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.7 virtual QList<QGraphicsLayoutItem*>& AbstractGraphicsLayout::items ( ) [pure virtual]

Implemented in FlowLayout, PetalLayout, and SpiralLayout.

7.2.3.8 virtual QSizeF AbstractGraphicsLayout::maxSize ( ) const [protected], [pure virtual]

7.2.3.9 virtual QSizeF AbstractGraphicsLayout::minSize (const QSizeF & *constraint*) const [protected], [pure virtual]

7.2.3.10 virtual QSizeF AbstractGraphicsLayout::prefSize ( ) const [protected], [pure virtual]

7.2.3.11 virtual void AbstractGraphicsLayout::removeAt (int *index*) [pure virtual]

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.12 virtual void AbstractGraphicsLayout::setGeometry (const QRectF & *geom*) [pure virtual]

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.13 virtual void AbstractGraphicsLayout::setSpacing (Qt::Orientations *o,* qreal *spacing*) [protected], [pure virtual]

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.14 virtual QSizeF AbstractGraphicsLayout::sizeHint (Qt::SizeHint *which,* const QSizeF & *constraint* = QSizeF()) const [protected], [pure virtual]

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

7.2.3.15 virtual qreal AbstractGraphicsLayout::spacing (Qt::Orientation *o*) const [protected], [pure virtual]

Implemented in FlowLayout, SpiralLayout, and PetalLayout.

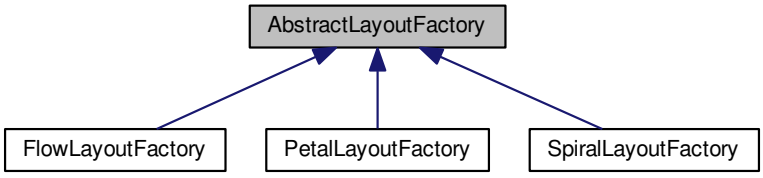
The documentation for this class was generated from the following file:

- layouts/AbstractGraphicsLayout.hpp

7.3 AbstractLayoutFactory Class Reference

`#include <AbstractLayoutFactory.hpp>`

Inheritance diagram for AbstractLayoutFactory:



Public Member Functions

- AbstractLayoutFactory ()
- virtual AbstractGraphicsLayout * makeLayout ()=0

7.3.1 Detailed Description

Definition at line 8 of file AbstractLayoutFactory.hpp.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 AbstractLayoutFactory::AbstractLayoutFactory ( ) [inline]

Definition at line 10 of file AbstractLayoutFactory.hpp.

7.3.3 Member Function Documentation

7.3.3.1 virtual AbstractGraphicsLayout* AbstractLayoutFactory::makeLayout ( ) [pure virtual]

creates a new AbstractGraphicsLayout instance

Implemented in FlowLayoutFactory, PetalLayoutFactory, and SpiralLayoutFactory.

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

- utils/AbstractLayoutFactory.hpp

7.4 CBIR Class Reference

`#include <CBIR.hpp>`

Classes

- struct MatCompare
- struct MatKey

Public Member Functions

- CBIR ()
- std::multimap< cv::Mat, cv::Mat, MatCompare > * computeHashes (const QList< cv::Mat > &images, cv::Ptr< cv::img_hash::ImgHashBase > hasher)
- void setHasher (cv::Ptr< cv::img_hash::ImgHashBase > hasher)
- cv::Mat getHash (const cv::Mat &image, cv::Ptr< cv::img_hash::ImgHashBase > hasher) const
- double getHashValue (const cv::Mat &image) const
- double getDistance (const cv::Mat &hashmatA, const cv::Mat &hashmatB) const

Static Public Attributes

- static cv::Ptr< cv::img_hash::ImgHashBase > static_hasher

7.4.1 Detailed Description

Definition at line 19 of file CBIR.hpp.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 CBIR::CBIR ( ) [explicit]

Definition at line 5 of file CBIR.cpp.

7.4.3 Member Function Documentation

7.4.3.1 ImageMap * CBIR::computeHashes (const QList< cv::Mat > & *images*, cv::Ptr< cv::img_hash::ImgHashBase > *hasher*)

using OpenCV's img_hash library

Definition at line 16 of file CBIR.cpp.

7.4.3.2 double CBIR::getDistance (const cv::Mat & *hashmatA*, const cv::Mat & *hashmatB*) const

Definition at line 41 of file CBIR.cpp.

7.4.3.3 cv::Mat CBIR::getHash (const cv::Mat & *image*, cv::Ptr< cv::img_hash::ImgHashBase > *hasher*) const

returns the image's hash value

Definition at line 29 of file CBIR.cpp.

7.4.3.4 double CBIR::getHashValue (const cv::Mat & *image*) const

Definition at line 35 of file CBIR.cpp.

7.4.3.5 void CBIR::setHasher (cv::Ptr< cv::img_hash::ImgHashBase > *hasher*) [inline]

Definition at line 39 of file CBIR.hpp.

7.4.4 Member Data Documentation

7.4.4.1 cv::Ptr< cv::img_hash::ImgHashBase > CBIR::static_hasher [static]

Definition at line 44 of file CBIR.hpp.

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

- utils/CBIR.hpp
- utils/CBIR.cpp

7.5 ImageCollection::Collection Struct Reference

`#include <ImageCollection.hpp>`

Public Member Functions

- Collection (GraphicsImage *image, cv::Mat *hash, QString *originalUrl)
- const GraphicsImage & getImage () const
- const cv::Mat & getHash () const
- const QString & getOriginalUrl () const

7.5.1 Detailed Description

Definition at line 71 of file ImageCollection.hpp.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 ImageCollection::Collection::Collection (GraphicsImage * *image*, cv::Mat * *hash*, QString * *originalUrl*) [inline]

Definition at line 73 of file ImageCollection.hpp.

7.5.3 Member Function Documentation

7.5.3.1 const cv::Mat& ImageCollection::Collection::getHash ( ) const [inline]

Definition at line 79 of file ImageCollection.hpp.

7.5.3.2 const GraphicsImage& ImageCollection::Collection::getImage ( ) const [inline]

Definition at line 78 of file ImageCollection.hpp.

7.5.3.3 const QString& ImageCollection::Collection::getOriginalUrl ( ) const [inline]

Definition at line 80 of file ImageCollection.hpp.

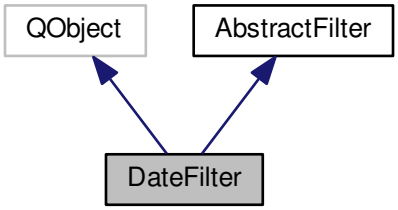
The documentation for this struct was generated from the following file:

- utils/ImageCollection.hpp

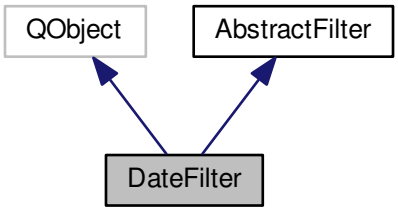
7.6 DateFilter Class Reference

`#include <DateFilter.hpp>`

Inheritance diagram for DateFilter:



Collaboration diagram for DateFilter:



Signals

- void datesChanged (const QJsonArray &results)

Public Member Functions

- DateFilter (const DbContext &dbContext)
- ~DateFilter ()=default
- DateFilter * makeFilter (const DbContext &dbContext)

  create a new DateFilter for filtering the images between 2 dates
- QGroupBox * makeControl ()
  - *create a GroupBox containing 2 DateEdit widgets, which will serve as a range picker between two dates*
- QPushButton & removeButton ()
- QPushButton & applyButton ()
- QStringList getDates ()

  calculate the 2 dates in milliseconds (since the epoch) represented in strings

7.6.1 Detailed Description

Definition at line 16 of file DateFilter.hpp.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 DateFilter::DateFilter (const DbContext & *dbContext*) [inline]

Definition at line 19 of file DateFilter.hpp.

7.6.2.2 DateFilter::~DateFilter ( ) [default]

7.6.3 Member Function Documentation

7.6.3.1 QPushButton& DateFilter::applyButton ( ) [inline]

Definition at line 41 of file DateFilter.hpp.

7.6.3.2 void DateFilter::datesChanged (const QJsonArray & *results*) [signal]

7.6.3.3 QStringList DateFilter::getDates ( )

calculate the 2 dates in milliseconds (since the epoch) represented in strings

Returns

Definition at line 42 of file DateFilter.cpp.

7.6.3.4 QGroupBox * DateFilter::makeControl ( ) [virtual]

- create a GroupBox containing 2 DateEdit widgets, which will serve as a range picker between two dates
- create a remove button for the filter

  Returns

  2 QDateEdits and a QPushButton inside a QGroupBox

Implements AbstractFilter.

Definition at line 4 of file DateFilter.cpp.

7.6.3.5 DateFilter* DateFilter::makeFilter (const DbContext & *dbContext*) [inline], [virtual]

create a new DateFilter for filtering the images between 2 dates

Returns

DateFilter

Implements AbstractFilter.

Definition at line 29 of file DateFilter.hpp.

7.6.3.6 QPushButton& DateFilter::removeButton ( ) [inline], [virtual]

Implements AbstractFilter.

Definition at line 39 of file DateFilter.hpp.

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

- filters/DateFilter.hpp
- filters/DateFilter.cpp

7.7 DbContext Class Reference

`#include <DbContext.hpp>`

Classes

- class MongoAccess

Public Member Functions

- mongocxx::uri loadUri ()
- void init ()
- QJsonArray queryAll ()
- QJsonArray queryText (const QString &text)
- QJsonArray queryImagePath (const QString &image_path)
- QJsonArray queryImagePaths (const QStringList &image_paths)
- QJsonArray queryDateRange (const QStringList &dates)

Static Public Member Functions

- static std::string bdate_to_string (const bsoncxx::document::element &bdate)

Public Attributes

- mongocxx::collection feedsNameCollection
- mongocxx::collection feedsCollection
- mongocxx::collection imageCollection
- std::string uri
- std::string databaseName
- std::string feedsNameCollection_name
- std::string feedsCollection_name
- std::string imageCollection_name
- std::vector< std::string > _keys

7.7.1 Detailed Description

Definition at line 41 of file DbContext.hpp.

7.7.2 Member Function Documentation

7.7.2.1 static std::string DbContext::bdate_to_string (const bsoncxx::document::element & *bdate*) [inline], [static]

Definition at line 52 of file DbContext.hpp.

7.7.2.2 void DbContext::init ( )

Definition at line 35 of file DbContext.cpp.

7.7.2.3 mongocxx::uri DbContext::loadUri ( )

Definition at line 3 of file DbContext.cpp.

7.7.2.4 QJsonArray DbContext::queryAll ( )

Definition at line 68 of file DbContext.cpp.

7.7.2.5 QJsonArray DbContext::queryDateRange (const QStringList & *dates*)

Definition at line 178 of file DbContext.cpp.

7.7.2.6 QJsonArray DbContext::queryImagePath (const QString & *image_path*)

Definition at line 120 of file DbContext.cpp.

7.7.2.7 QJsonArray DbContext::queryImagePaths (const QStringList & *image_paths*)

Definition at line 144 of file DbContext.cpp.

7.7.2.8 QJsonArray DbContext::queryText (const QString & *text*)

Definition at line 94 of file DbContext.cpp.

7.7.3 Member Data Documentation

7.7.3.1 std::vector<std::string> DbContext::_keys

Initial value:

```
= {"link", "image_url", "rss", "author", "title", "summary",
                                "published", "image_path"}
```

Definition at line 71 of file DbContext.hpp.

7.7.3.2 std::string DbContext::databaseName

Definition at line 66 of file DbContext.hpp.

7.7.3.3 mongocxx::collection DbContext::feedsCollection

Definition at line 62 of file DbContext.hpp.

7.7.3.4 std::string DbContext::feedsCollection_name

Definition at line 68 of file DbContext.hpp.

7.7.3.5 mongocxx::collection DbContext::feedsNameCollection

Definition at line 61 of file DbContext.hpp.

7.7.3.6 std::string DbContext::feedsNameCollection_name

Definition at line 67 of file DbContext.hpp.

7.7.3.7 mongocxx::collection DbContext::imageCollection

Definition at line 63 of file DbContext.hpp.

7.7.3.8 std::string DbContext::imageCollection_name

Definition at line 69 of file DbContext.hpp.

7.7.3.9 std::string DbContext::uri

Definition at line 65 of file DbContext.hpp.

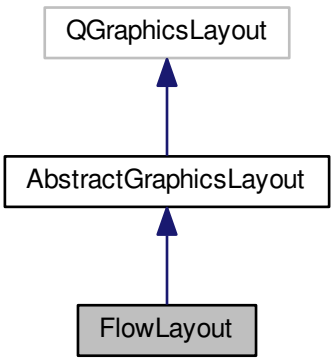
The documentation for this class was generated from the following files:

- db/DbContext.hpp
- db/DbContext.cpp

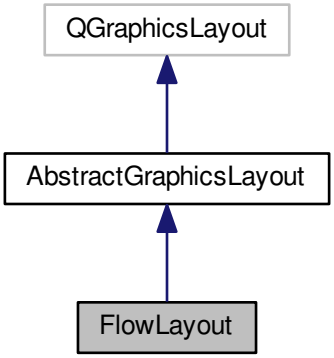
7.8 FlowLayout Class Reference

```
#include <FlowLayout.hpp>
```

Inheritance diagram for FlowLayout:



Collaboration diagram for FlowLayout:



Public Member Functions

- FlowLayout ()
- void setSpacing (Qt::Orientations o, qreal spacing) Q_DECL_OVERRIDE
- qreal spacing (Qt::Orientation o) const Q_DECL_OVERRIDE
- void setGeometry (const QRectF &geom) Q_DECL_OVERRIDE
- int count () const Q_DECL_OVERRIDE
- QGraphicsLayoutItem * itemAt (int index) const Q_DECL_OVERRIDE
- void removeAt (int index) Q_DECL_OVERRIDE
- void addItem (QGraphicsLayoutItem *item) Q_DECL_OVERRIDE
- void clearAll () Q_DECL_OVERRIDE
- QList< QGraphicsLayoutItem * > & items () Q_DECL_OVERRIDE

Protected Member Functions

- void insertItem (int index, QGraphicsLayoutItem *item) Q_DECL_OVERRIDE
- QSizeF sizeHint (Qt::SizeHint which, const QSizeF &constraint=QSizeF()) const Q_DECL_OVERRIDE

7.8.1 Detailed Description

Definition at line 61 of file FlowLayout.hpp.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 FlowLayout::FlowLayout ( ) [explicit]

Definition at line 53 of file FlowLayout.cpp.

7.8.3 Member Function Documentation

7.8.3.1 void FlowLayout::addItem (QGraphicsLayoutItem * *item*) [inline], [virtual]

inserts the item to the front of the list

Implements AbstractGraphicsLayout.

Definition at line 72 of file FlowLayout.hpp.

7.8.3.2 void FlowLayout::clearAll ( ) [virtual]

deletes and removes every element from the layout

Implements AbstractGraphicsLayout.

Definition at line 70 of file FlowLayout.cpp.

7.8.3.3 int FlowLayout::count ( ) const [virtual]

Implements AbstractGraphicsLayout.

Definition at line 78 of file FlowLayout.cpp.

7.8.3.4 void FlowLayout::insertItem (int *index,* QGraphicsLayoutItem * *item*) [protected], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 61 of file FlowLayout.cpp.

7.8.3.5 QGraphicsLayoutItem * FlowLayout::itemAt (int *index*) const [virtual]

Implements AbstractGraphicsLayout.

Definition at line 82 of file FlowLayout.cpp.

7.8.3.6 QList<QGraphicsLayoutItem*>& FlowLayout::items ( ) [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 74 of file FlowLayout.hpp.

7.8.3.7 void FlowLayout::removeAt (int *index*) [virtual]

Implements AbstractGraphicsLayout.

Definition at line 86 of file FlowLayout.cpp.

7.8.3.8 void FlowLayout::setGeometry (const QRectF & *geom*) [virtual]

Implements AbstractGraphicsLayout.

Definition at line 102 of file FlowLayout.cpp.

7.8.3.9 void FlowLayout::setSpacing (Qt::Orientations *o*, qreal *spacing*) [virtual]

Implements AbstractGraphicsLayout.

Definition at line 95 of file FlowLayout.cpp.

7.8.3.10 QSizeF FlowLayout::sizeHint (Qt::SizeHint *which*, const QSizeF & *constraint* = QSizeF()) const [protected], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 201 of file FlowLayout.cpp.

7.8.3.11 qreal FlowLayout::spacing (Qt::Orientation *o*) const [virtual]

Implements AbstractGraphicsLayout.

Definition at line 91 of file FlowLayout.cpp.

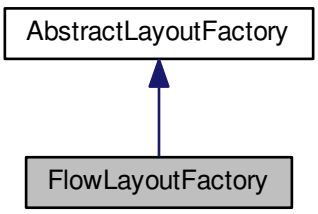
The documentation for this class was generated from the following files:

- layouts/FlowLayout.hpp
- layouts/FlowLayout.cpp

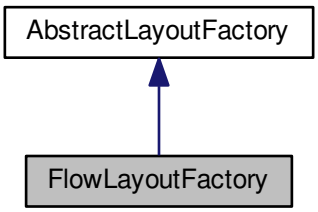
7.9 FlowLayoutFactory Class Reference

```
#include <FlowLayoutFactory.hpp>
```

Inheritance diagram for FlowLayoutFactory:



Collaboration diagram for FlowLayoutFactory:



Public Member Functions

- FlowLayoutFactory ()=default
- FlowLayout * makeLayout ()

7.9.1 Detailed Description

Definition at line 8 of file FlowLayoutFactory.hpp.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 FlowLayoutFactory::FlowLayoutFactory ( ) [default]

7.9.3 Member Function Documentation

7.9.3.1 FlowLayout* FlowLayoutFactory::makeLayout ( ) [inline], [virtual]

creates a new AbstractGraphicsLayout instance

Implements AbstractLayoutFactory.

Definition at line 12 of file FlowLayoutFactory.hpp.

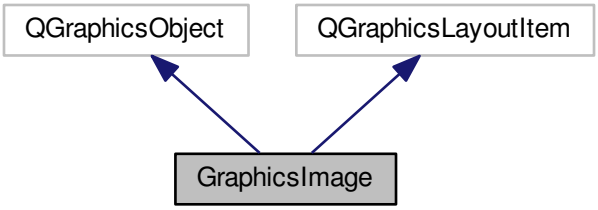
The documentation for this class was generated from the following file:

- utils/FlowLayoutFactory.hpp

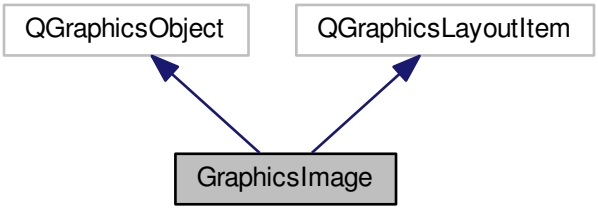
7.10 GraphicsImage Class Reference

`#include <GraphicsImage.hpp>`

Inheritance diagram for GraphicsImage:



Collaboration diagram for GraphicsImage:



Signals

- void clicked (const QString &url)
- void doubleClick (const QString &url)
- void hoverEnter (const QString &url, GraphicsImage *)
- void hoverLeave ()

Public Member Functions

- GraphicsImage ()=default
- GraphicsImage (const QImage &image, const QString &url, const QString &originalUrl)
- GraphicsImage (const QImage &image)
- GraphicsImage (const GraphicsImage &other)
- GraphicsImage & operator= (const GraphicsImage &other)
- ~GraphicsImage ()=default

- void setGeometry (const QRectF &geom)
- QSizeF sizeHint (Qt::SizeHint which, const QSizeF &constraint=QSizeF()) const
- QRectF boundingRect () const
- void paint (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget=0)
- qreal getWidth () const
- qreal getHeight () const
- QPixmap getPixmap () const
- QString getUrl () const
- QString getOriginalUrl () const

Protected Member Functions

- void mousePressEvent (QGraphicsSceneMouseEvent *event)
- void hoverEnterEvent (QGraphicsSceneHoverEvent *event)
- void hoverLeaveEvent (QGraphicsSceneHoverEvent *event)
- void mouseDoubleClickEvent (QGraphicsSceneMouseEvent *event)

7.10.1 Detailed Description

Definition at line 56 of file GraphicsImage.hpp.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 GraphicsImage::GraphicsImage ( ) [default]

7.10.2.2 GraphicsImage::GraphicsImage (const QImage & *image*, const QString & *url*, const QString & *originalUrl*)

Definition at line 43 of file GraphicsImage.cpp.

7.10.2.3 GraphicsImage::GraphicsImage (const QImage & *image*)

Definition at line 55 of file GraphicsImage.cpp.

7.10.2.4 GraphicsImage::GraphicsImage (const GraphicsImage & *other*)

Definition at line 65 of file GraphicsImage.cpp.

7.10.2.5 GraphicsImage::~GraphicsImage ( ) [default]

7.10.3 Member Function Documentation

7.10.3.1 QRectF GraphicsImage::boundingRect ( ) const

Definition at line 100 of file GraphicsImage.cpp.

7.10.3.2 void GraphicsImage::clicked (const QString & *url*) [signal]

7.10.3.3 void GraphicsImage::doubleClick (const QString & *url*) [signal]

7.10.3.4 qreal GraphicsImage::getHeight ( ) const [inline]

Definition at line 81 of file GraphicsImage.hpp.

7.10.3.5 QString GraphicsImage::getOriginalUrl ( ) const [inline]

Definition at line 84 of file GraphicsImage.hpp.

7.10.3.6 QPixmap GraphicsImage::getPixmap ( ) const [inline]

Definition at line 82 of file GraphicsImage.hpp.

7.10.3.7 QString GraphicsImage::getUrl ( ) const [inline]

Definition at line 83 of file GraphicsImage.hpp.

7.10.3.8 qreal GraphicsImage::getWidth ( ) const [inline]

Definition at line 80 of file GraphicsImage.hpp.

7.10.3.9 void GraphicsImage::hoverEnter (const QString & *url,* GraphicsImage *) [signal]

7.10.3.10 void GraphicsImage::hoverEnterEvent (QGraphicsSceneHoverEvent * *event*) [protected]

Definition at line 143 of file GraphicsImage.cpp.

7.10.3.11 void GraphicsImage::hoverLeave ( ) [signal]

7.10.3.12 void GraphicsImage::hoverLeaveEvent (QGraphicsSceneHoverEvent * *event*) [protected]

Definition at line 151 of file GraphicsImage.cpp.

7.10.3.13 void GraphicsImage::mouseDoubleClickEvent (QGraphicsSceneMouseEvent * *event*) [protected]

Definition at line 138 of file GraphicsImage.cpp.

7.10.3.14 void GraphicsImage::mousePressEvent (QGraphicsSceneMouseEvent * *event*) [protected]

Definition at line 124 of file GraphicsImage.cpp.

7.10.3.15 GraphicsImage & GraphicsImage::operator= (const GraphicsImage & *other*)

Definition at line 79 of file GraphicsImage.cpp.

7.10.3.16 void GraphicsImage::paint (QPainter * *painter,* const QStyleOptionGraphicsItem * *option,* QWidget * *widget* = 0)

Definition at line 92 of file GraphicsImage.cpp.

7.10.3.17 void GraphicsImage::setGeometry (const QRectF & *geom*)

Definition at line 104 of file GraphicsImage.cpp.

7.10.3.18 QSizeF GraphicsImage::sizeHint (Qt::SizeHint *which,* const QSizeF & *constraint* = QSizeF()) const

Definition at line 110 of file GraphicsImage.cpp.

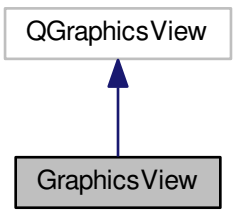
The documentation for this class was generated from the following files:

- utils/GraphicsImage.hpp
- utils/GraphicsImage.cpp

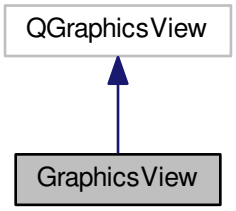
7.11 GraphicsView Class Reference

`#include <GraphicsView.hpp>`

Inheritance diagram for GraphicsView:



Collaboration diagram for GraphicsView:



Public Slots

- void onAddItem (const QGraphicsLayoutItem *item)

Signals

- void imageClick (QGraphicsItem *image)

Public Member Functions

- GraphicsView (QWidget *parent=0)
- void init ()
- void setLayout (const QString &value)
- void setMinSceneSize (const QSizeF value)
- void addItem (const QGraphicsLayoutItem *item)
- int itemCount () const
- void clear ()
- void addPopupImage (QLabel *label, GraphicsImage *item)
- void wheelEvent (QWheelEvent *event)
- QGraphicsScene & scene ()
- void setNrOfPetals (int value)
- void setRadius (double value)
- void setSpiralDistance (int value)
- void setSpiralTurn (int value)
- QList< GraphicsImage > & getSelectedImages ()

Protected Member Functions

- void mouseReleaseEvent (QMouseEvent *event)

7.11.1 Detailed Description

Definition at line 32 of file GraphicsView.hpp.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 GraphicsView::GraphicsView (QWidget * *parent* = 0) [explicit]

Definition at line 3 of file GraphicsView.cpp.

7.11.3 Member Function Documentation

7.11.3.1 void GraphicsView::addItem (const QGraphicsLayoutItem * *item*)

Definition at line 43 of file GraphicsView.cpp.

7.11.3.2 void GraphicsView::addPopupImage (QLabel * *label,* GraphicsImage * *item*)

Definition at line 95 of file GraphicsView.cpp.

7.11.3.3 void GraphicsView::clear ( ) [inline]

Definition at line 41 of file GraphicsView.hpp.

7.11.3.4 QList< GraphicsImage > & GraphicsView::getSelectedImages ( )

Definition at line 144 of file GraphicsView.cpp.

7.11.3.5 void GraphicsView::imageClick (QGraphicsItem * *image*) [signal]

7.11.3.6 void GraphicsView::init ( )

Definition at line 8 of file GraphicsView.cpp.

7.11.3.7 int GraphicsView::itemCount ( ) const [inline]

Definition at line 40 of file GraphicsView.hpp.

7.11.3.8 void GraphicsView::mouseReleaseEvent (QMouseEvent * *event*) [protected]

Definition at line 139 of file GraphicsView.cpp.

7.11.3.9 void GraphicsView::onAddItem (const QGraphicsLayoutItem * *item*) [inline], [slot]

Definition at line 85 of file GraphicsView.hpp.

7.11.3.10 QGraphicsScene& GraphicsView::scene ( ) [inline]

expose a reference to the scene to get it's signals; reference, so the ownership won't move

Definition at line 51 of file GraphicsView.hpp.

7.11.3.11 void GraphicsView::setLayout (const QString & *value*)

makes a new layout from the available factories and sets it onto the scene

Definition at line 50 of file GraphicsView.cpp.

7.11.3.12 void GraphicsView::setMinSceneSize (const QSizeF *value*) [inline]

Definition at line 38 of file GraphicsView.hpp.

7.11.3.13 void GraphicsView::setNrOfPetals (int *value*)

opencv img_hash & pHash display

Definition at line 67 of file GraphicsView.cpp.

7.11.3.14 void GraphicsView::setRadius (double *value*)

Definition at line 74 of file GraphicsView.cpp.

7.11.3.15 void GraphicsView::setSpiralDistance (int *value*)

Definition at line 81 of file GraphicsView.cpp.

7.11.3.16 void GraphicsView::setSpiralTurn (int *value*)

Definition at line 88 of file GraphicsView.cpp.

7.11.3.17 void GraphicsView::wheelEvent (QWheelEvent * *event*)

Definition at line 120 of file GraphicsView.cpp.

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

- view/GraphicsView.hpp
- view/GraphicsView.cpp

7.12 ImageCollection Class Reference

`#include <ImageCollection.hpp>`

Classes

- struct Collection

Public Member Functions

- ImageCollection ()
- ~ImageCollection ()=default
- void init ()

  define the hashing algorithm types -> collection_map key
- QStringList getHashingAlgorithms () const
- void insert (cv::Mat *image, QString *url, QString *originalUrl)
- cv::Mat getHashValue (const QString &hasherName, const QString &url)
- cv::Ptr< cv::img_hash::ImgHashBase > getHasher (const QString &hasherName) const
- GraphicsImage getImage (const QString &hasherName, const QString &url)
- QList< cv::Mat > getHashes (const QString &hasherName) const

  returns all the results from the hasherName hash algorithm
- QList< GraphicsImage > * getHashedImages (const QString &hasherName)

  getHashedImages returns all of the images in order of their hash values
- QList< GraphicsImage > * getSimilarImages (const QString &url, const QString &hasherName)

  getSimilarImages returns the images in order of their similarity to the selected image
- QList< GraphicsImage > * getImagesByUrl (const QStringList &imgUrls) const

7.12.1 Detailed Description

Definition at line 21 of file ImageCollection.hpp.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 ImageCollection::ImageCollection ( )

Definition at line 5 of file ImageCollection.cpp.

7.12.2.2 ImageCollection::~ImageCollection ( ) [default]

7.12.3 Member Function Documentation

7.12.3.1 QList< GraphicsImage > * ImageCollection::getHashedImages (const QString & *hasherName*)

getHashedImages returns all of the images in order of their hash values

Parameters

hasherName	hash algorithm

Returns

the hashed images

Definition at line 49 of file ImageCollection.cpp.

7.12.3.2 cv::Ptr<cv::img_hash::ImgHashBase> ImageCollection::getHasher (const QString & *hasherName*) const [inline]

Definition at line 39 of file ImageCollection.hpp.

7.12.3.3 QList< cv::Mat > ImageCollection::getHashes (const QString & *hasherName*) const

returns all the results from the hasherName hash algorithm

Parameters

hasherName	hash algorithm

Returns

list containing the hash values

Definition at line 40 of file ImageCollection.cpp.

7.12.3.4 QStringList ImageCollection::getHashingAlgorithms ( ) const [inline]

Definition at line 31 of file ImageCollection.hpp.

7.12.3.5 cv::Mat ImageCollection::getHashValue (const QString & *hasherName,* const QString & *url*) [inline]

Definition at line 35 of file ImageCollection.hpp.

7.12.3.6 GraphicsImage ImageCollection::getImage (const QString & *hasherName,* const QString & *url*) [inline]

Definition at line 43 of file ImageCollection.hpp.

7.12.3.7 QList< GraphicsImage > * ImageCollection::getImagesByUrl (const QStringList & *imgUrls*) const

Definition at line 112 of file ImageCollection.cpp.

7.12.3.8 QList< GraphicsImage > * ImageCollection::getSimilarImages (const QString & *url,* const QString & *hasherName*)

getSimilarImages returns the images in order of their similarity to the selected image

Parameters

url	the selected image's url
hasherName	

Returns

a list containing the images in a specific order

Definition at line 64 of file ImageCollection.cpp.

7.12.3.9 void ImageCollection::init ( )

define the hashing algorithm types -> collection_map key

Definition at line 9 of file ImageCollection.cpp.

7.12.3.10 void ImageCollection::insert (cv::Mat * *image*, QString * *url*, QString * *originalUrl*)

compute the image's hash value with every function, then insert the results into the map

calculate the hash

insert the results into the hasher's map

Definition at line 28 of file ImageCollection.cpp.

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

- utils/ImageCollection.hpp
- utils/ImageCollection.cpp

7.13 ImageConverter Class Reference

`#include <ImageConverter.hpp>`

Public Member Functions

- ImageConverter ()=default

Static Public Member Functions

- static QImage Mat2QImage (const cv::Mat &cvImage)
- static cv::Mat QImage2Mat (const QImage &image)

7.13.1 Detailed Description

Definition at line 13 of file ImageConverter.hpp.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 ImageConverter::ImageConverter ( ) [default]

7.13.3 Member Function Documentation

7.13.3.1 QImage ImageConverter::Mat2QImage (const cv::Mat & *cvImage*) [static]

Definition at line 3 of file ImageConverter.cpp.

7.13.3.2 cv::Mat ImageConverter::QImage2Mat (const QImage & *image*) [static]

Definition at line 36 of file ImageConverter.cpp.

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

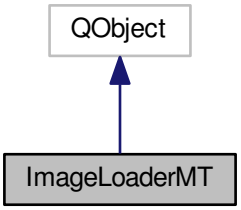
- utils/ImageConverter.hpp
- utils/ImageConverter.cpp

7.14 ImageLoaderMT Class Reference

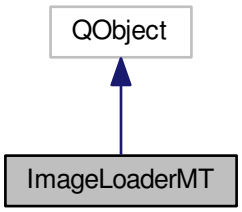
handles image loading in a multi-threaded asynchronous way

`#include <ImageLoaderMT.hpp>`

Inheritance diagram for ImageLoaderMT:



Collaboration diagram for ImageLoaderMT:



Signals

- void imageReady (const GraphicsImage &image)

Public Member Functions

- ImageLoaderMT (const QStringList &imageNames, int width, int height, ImageCollection &imageCollection)
- ~ImageLoaderMT ()=default
- void run ()

Public Attributes

- QFutureWatcher< QList< GraphicsImage > > loaderWatcher

7.14.1 Detailed Description

handles image loading in a multi-threaded asynchronous way

Definition at line 17 of file ImageLoaderMT.hpp.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 ImageLoaderMT::ImageLoaderMT (const QStringList & *imageNames*, int *width*, int *height*, ImageCollection & *imageCollection*) [inline]

Definition at line 20 of file ImageLoaderMT.hpp.

7.14.2.2 ImageLoaderMT::~ImageLoaderMT ( ) [default]

7.14.3 Member Function Documentation

7.14.3.1 void ImageLoaderMT::imageReady (const GraphicsImage & *image*) [signal]

7.14.3.2 void ImageLoaderMT::run ( )

Definition at line 3 of file ImageLoaderMT.cpp.

7.14.4 Member Data Documentation

7.14.4.1 QFutureWatcher<QList<GraphicsImage> > ImageLoaderMT::loaderWatcher

Definition at line 32 of file ImageLoaderMT.hpp.

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

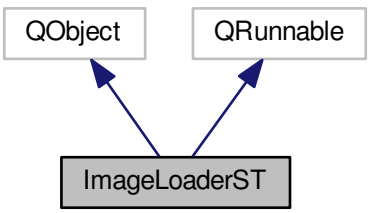
- utils/image_load/ImageLoaderMT.hpp
- utils/image_load/ImageLoaderMT.cpp

7.15 ImageLoaderST Class Reference

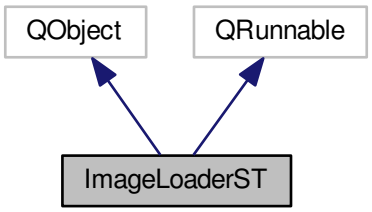
handles image loading in a single-threaded asyncronous way

`#include <ImageLoaderST.hpp>`

Inheritance diagram for ImageLoaderST:



Collaboration diagram for ImageLoaderST:



Public Slots

- void onCancel ()

Signals

- void resultReady (int index)
- void finished ()

Public Member Functions

- ImageLoaderST (QStringList &imageNames, QList< GraphicsImage > &results, const cv::Size &size, ImageCollection &imageCollection, QObject *parent=0)
- ~ImageLoaderST ()=default
- void run ()
- bool isRunning () const
- void cancel ()

7.15.1 Detailed Description

handles image loading in a single-threaded asyncronous way

Definition at line 20 of file ImageLoaderST.hpp.

7.15.2 Constructor & Destructor Documentation

7.15.2.1 ImageLoaderST::ImageLoaderST (QStringList & *imageNames,* QList< GraphicsImage > & *results,* const cv::Size & *size,* ImageCollection & *imageCollection,* QObject * *parent* = 0)

Definition at line 3 of file ImageLoaderST.cpp.

7.15.2.2 ImageLoaderST::~ImageLoaderST ( ) [default]

7.15.3 Member Function Documentation

7.15.3.1 void ImageLoaderST::cancel ( ) [inline]

Definition at line 33 of file ImageLoaderST.hpp.

7.15.3.2 void ImageLoaderST::finished ( ) [signal]

7.15.3.3 bool ImageLoaderST::isRunning ( ) const [inline]

Definition at line 32 of file ImageLoaderST.hpp.

7.15.3.4 void ImageLoaderST::onCancel ( ) [inline], [slot]

Definition at line 51 of file ImageLoaderST.hpp.

7.15.3.5 void ImageLoaderST::resultReady (int *index*) [signal]

7.15.3.6 void ImageLoaderST::run ( )

Definition at line 13 of file ImageLoaderST.cpp.

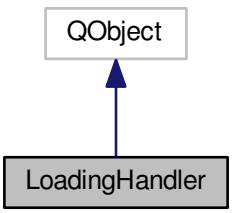
The documentation for this class was generated from the following files:

- utils/image_load/ImageLoaderST.hpp
- utils/image_load/ImageLoaderST.cpp

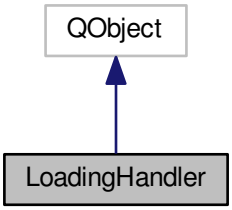
7.16 LoadingHandler Class Reference

`#include <LoadingHandler.hpp>`

Inheritance diagram for LoadingHandler:



Collaboration diagram for LoadingHandler:



Public Slots

- void onFinishedLoading ()
- void onCancel ()

Signals

- void imageReady_st (int index)
- void imageReady_mt (const GraphicsImage &image)
- void finishedLoading ()

Public Member Functions

- LoadingHandler (ImageCollection &imageCollection)
- void loadImages_mt (QStringList *imageNames)

  loadImages_mt loads the images located at the chosen path using multiple threads. The loaded images will be accessible via signals
- QList< GraphicsImage > * loadImages_st (QStringList *imageNames)

  loadImages_st loads the images located at the chosen path using a single thread
- cv::Mat loadImage (const QString &fileName) const
- void setWidth (int width)
- void setHeight (int height)

7.16.1 Detailed Description

Definition at line 13 of file LoadingHandler.hpp.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 LoadingHandler::LoadingHandler (ImageCollection & *imageCollection*) [inline]

Definition at line 16 of file LoadingHandler.hpp.

7.16.3 Member Function Documentation

7.16.3.1 void LoadingHandler::finishedLoading ( ) [signal]

7.16.3.2 void LoadingHandler::imageReady_mt (const GraphicsImage & *image*) [signal]

7.16.3.3 void LoadingHandler::imageReady_st (int *index*) [signal]

7.16.3.4 cv::Mat LoadingHandler::loadImage (const QString & *fileName*) const

7.16.3.5 void LoadingHandler::loadImages_mt (QStringList * *imageNames*)

loadImages_mt loads the images located at the chosen path using multiple threads. The loaded images will be accessible via signals

Parameters

imageNames	the file names

Definition at line 3 of file LoadingHandler.cpp.

7.16.3.6 QList< GraphicsImage > * LoadingHandler::loadImages_st (QStringList * *imageNames*)

loadImages_st loads the images located at the chosen path using a single thread

Parameters

imageNames	the file names

Returns

the loaded images

Definition at line 15 of file LoadingHandler.cpp.

7.16.3.7 void LoadingHandler::onCancel ( ) [slot]

Definition at line 33 of file LoadingHandler.cpp.

7.16.3.8 void LoadingHandler::onFinishedLoading ( ) [slot]

Definition at line 29 of file LoadingHandler.cpp.

7.16.3.9 void LoadingHandler::setHeight (int *height*) [inline]

Definition at line 38 of file LoadingHandler.hpp.

7.16.3.10 void LoadingHandler::setWidth (int *width*) [inline]

Definition at line 37 of file LoadingHandler.hpp.

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

- utils/image_load/LoadingHandler.hpp
- utils/image_load/LoadingHandler.cpp

7.17 Logger Class Reference

`#include <Logger.hpp>`

Static Public Member Functions

- static void log (const std::string &message)
  log a simple message to the stdout and a text file

Static Public Attributes

- static std::string file_name = "log_file.txt"

7.17.1 Detailed Description

Definition at line 10 of file Logger.hpp.

7.17.2 Member Function Documentation

7.17.2.1 static void Logger::log (const std::string & *message*) [inline], [static]

log a simple message to the stdout and a text file

Parameters

message	

Definition at line 16 of file Logger.hpp.

7.17.3 Member Data Documentation

7.17.3.1 std::string Logger::file_name = "log_file.txt" [static]

Definition at line 28 of file Logger.hpp.

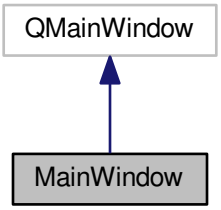
The documentation for this class was generated from the following files:

- utils/Logger.hpp
- ui/MainWindow.cpp

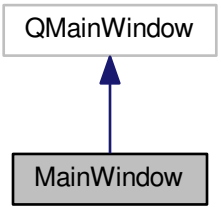
7.18 MainWindow Class Reference

`#include <MainWindow.hpp>`

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Signals

- void addViewItem (const QGraphicsLayoutItem *item)
- void clearLayout ()
- void resizeImages (int newWidth, int newHeight)
- void saveProgress (int value)
- void display (const QList< GraphicsImage > &images)

Public Member Functions

- MainWindow (QWidget *parent=0)
- MainWindow (MainWindow const &otherWindow)=delete
- MainWindow & operator= (MainWindow const &otherWindow)=delete
- ~MainWindow ()

7.18.1 Detailed Description

Definition at line 44 of file MainWindow.hpp.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 MainWindow::MainWindow (QWidget * *parent* = 0) [explicit]

Definition at line 9 of file MainWindow.cpp.

7.18.2.2 MainWindow::MainWindow (MainWindow const & *otherWindow*) [delete]

7.18.2.3 MainWindow::~MainWindow ( )

Definition at line 22 of file MainWindow.cpp.

7.18.3 Member Function Documentation

7.18.3.1 void MainWindow::addViewItem (const QGraphicsLayoutItem * *item*) [signal]

7.18.3.2 void MainWindow::clearLayout ( ) [signal]

7.18.3.3 void MainWindow::display (const QList< GraphicsImage > & *images*) [signal]

7.18.3.4 MainWindow& MainWindow::operator= (MainWindow const & *otherWindow*) [delete]

7.18.3.5 void MainWindow::resizeImages (int *newWidth*, int *newHeight*) [signal]

7.18.3.6 void MainWindow::saveProgress (int *value*) [signal]

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

- ui/MainWindow.hpp
- ui/MainWindow.cpp

7.19 Mapper Class Reference

`#include <Mapper.hpp>`

Public Types

- using result_type = GraphicsImage

Public Member Functions

- Mapper ()=default
- Mapper (const int &width, const int &height, ImageCollection &imageCollection)
- ~Mapper ()=default
- GraphicsImage operator() (const QString &imageName)
- Mapper & setWidth (const int &width)
- Mapper & setHeight (const int &height)

7.19.1 Detailed Description

Definition at line 15 of file Mapper.hpp.

7.19.2 Member Typedef Documentation

7.19.2.1 using Mapper::result_type = GraphicsImage

Definition at line 25 of file Mapper.hpp.

7.19.3 Constructor & Destructor Documentation

7.19.3.1 Mapper::Mapper ( ) [default]

7.19.3.2 Mapper::Mapper (const int & *width*, const int & *height*, ImageCollection & *imageCollection*) [inline]

Definition at line 18 of file Mapper.hpp.

7.19.3.3 Mapper::~Mapper ( ) [default]

7.19.4 Member Function Documentation

7.19.4.1 GraphicsImage Mapper::operator() (const QString & *imageName*) [inline]

Definition at line 27 of file Mapper.hpp.

7.19.4.2 Mapper& Mapper::setHeight (const int & *height*) [inline]

Definition at line 48 of file Mapper.hpp.

7.19.4.3 Mapper& Mapper::setWidth (const int & *width*) [inline]

Definition at line 43 of file Mapper.hpp.

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

- utils/image_load/Mapper.hpp

7.20 CBIR::MatCompare Struct Reference

`#include <CBIR.hpp>`

Public Member Functions

- double operator() (const cv::Mat &hashmatA, const cv::Mat &hashmatB) const

7.20.1 Detailed Description

compares 2 cv::Mat objects

Definition at line 30 of file CBIR.hpp.

7.20.2 Member Function Documentation

7.20.2.1 double CBIR::MatCompare::operator() (const cv::Mat & *hashmatA*, const cv::Mat & *hashmatB*) const

Definition at line 10 of file CBIR.cpp.

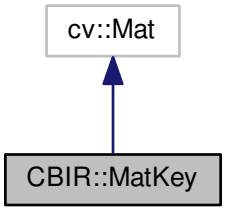
The documentation for this struct was generated from the following files:

- utils/CBIR.hpp
- utils/CBIR.cpp

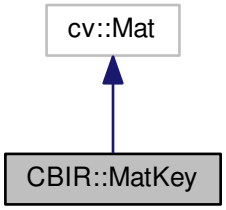
7.21 CBIR::MatKey Struct Reference

`#include <CBIR.hpp>`

Inheritance diagram for CBIR::MatKey:



Collaboration diagram for CBIR::MatKey:



Public Member Functions

- double operator< (const MatKey &other)

7.21.1 Detailed Description

Definition at line 23 of file CBIR.hpp.

7.21.2 Member Function Documentation

7.21.2.1 double CBIR::MatKey::operator< (const MatKey & *other*) [inline]

Definition at line 24 of file CBIR.hpp.

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

- utils/CBIR.hpp

7.22 Metadata Class Reference

`#include <Metadata.hpp>`

Public Member Functions

- Metadata ()=default
- ~Metadata ()=default
- std::string & operator[ ] (const std::string &key)
- const std::string & operator[ ] (const std::string &key) const
- std::vector< std::string > keys () const

Public Attributes

- std::string link
- std::string image_url
- std::string rss
- std::string author
- std::string title
- std::string summary
- std::string published
- std::string image_path

7.22.1 Detailed Description

Definition at line 12 of file Metadata.hpp.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 Metadata::Metadata ( ) [default]

7.22.2.2 Metadata::~Metadata ( ) [default]

7.22.3 Member Function Documentation

7.22.3.1 std::vector<std::string> Metadata::keys ( ) const [inline]

Definition at line 25 of file Metadata.hpp.

7.22.3.2 std::string& Metadata::operator[ ] (const std::string & *key*) [inline]

Definition at line 17 of file Metadata.hpp.

7.22.3.3 const std::string& Metadata::operator[ ] (const std::string & *key*) const [inline]

Definition at line 21 of file Metadata.hpp.

7.22.4 Member Data Documentation

7.22.4.1 std::string Metadata::author

Definition at line 32 of file Metadata.hpp.

7.22.4.2 std::string Metadata::image_path

Definition at line 36 of file Metadata.hpp.

7.22.4.3 std::string Metadata::image_url

Definition at line 30 of file Metadata.hpp.

7.22.4.4 std::string Metadata::link

Definition at line 29 of file Metadata.hpp.

7.22.4.5 std::string Metadata::published

Definition at line 35 of file Metadata.hpp.

7.22.4.6 std::string Metadata::rss

Definition at line 31 of file Metadata.hpp.

7.22.4.7 std::string Metadata::summary

Definition at line 34 of file Metadata.hpp.

7.22.4.8 std::string Metadata::title

Definition at line 33 of file Metadata.hpp.

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

- utils/metadata/Metadata.hpp

7.23 MetadataParser Class Reference

`#include <MetadataParser.hpp>`

Public Member Functions

- MetadataParser ()=default
- ~MetadataParser ()=default

Static Public Member Functions

- static QList< Metadata > & getMetadata (const QJsonArray &metadata)

  returns the images' metadata
- static QList< GraphicsImage > & getImages (const QList< Metadata > &metadata, const ImageCollection &imageCollection)

  selects the images from the metadata

7.23.1 Detailed Description

Definition at line 16 of file MetadataParser.hpp.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 MetadataParser::MetadataParser ( ) [default]

7.23.2.2 MetadataParser::~MetadataParser ( ) [default]

7.23.3 Member Function Documentation

7.23.3.1 QList< GraphicsImage > & MetadataParser::getImages (const QList< Metadata > & *metadata,* const ImageCollection & *imageCollection*) [static]

selects the images from the metadata

Parameters

metadata	
imageCollection	

Returns

Definition at line 23 of file MetadataParser.cpp.

7.23.3.2 QList< Metadata > & MetadataParser::getMetadata (const QJsonArray & *metadata*) [static]

returns the images' metadata

Parameters

metadata	

Returns

Definition at line 3 of file MetadataParser.cpp.

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

- utils/metadata/MetadataParser.hpp
- utils/metadata/MetadataParser.cpp

7.24 DbContext::MongoAccess Class Reference

`#include <DbContext.hpp>`

Public Types

- using connection = mongocxx::pool::entry

Public Member Functions

- void configure (std::unique_ptr< mongocxx::instance > instance, std::unique_ptr< mongocxx::pool > pool)
- connection get_connection ()
- mongocxx::stdx::optional< connection > try_get_connection ()

Static Public Member Functions

- static MongoAccess & instance ()

7.24.1 Detailed Description

Definition at line 74 of file DbContext.hpp.

7.24.2 Member Typedef Documentation

7.24.2.1 using DbContext::MongoAccess::connection = mongocxx::pool::entry

Definition at line 87 of file DbContext.hpp.

7.24.3 Member Function Documentation

7.24.3.1 void DbContext::MongoAccess::configure (std::unique_ptr< mongocxx::instance > *instance*, std::unique_ptr< mongocxx::pool > *pool*) [inline]

Definition at line 81 of file DbContext.hpp.

7.24.3.2 connection DbContext::MongoAccess::get_connection ( ) [inline]

Definition at line 89 of file DbContext.hpp.

7.24.3.3 static MongoAccess& DbContext::MongoAccess::instance ( ) [inline], [static]

Definition at line 76 of file DbContext.hpp.

7.24.3.4 mongocxx::stdx::optional<connection> DbContext::MongoAccess::try_get_connection ( ) [inline]

Definition at line 93 of file DbContext.hpp.

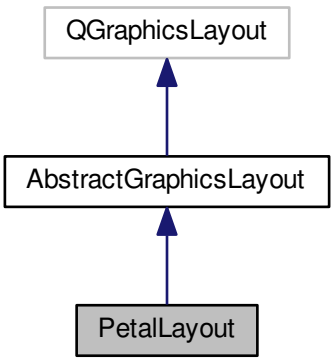
The documentation for this class was generated from the following file:

- db/DbContext.hpp

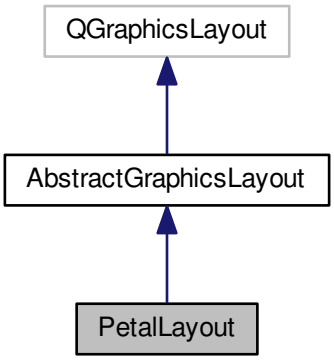
7.25 PetalLayout Class Reference

`#include <PetalLayout.hpp>`

Inheritance diagram for PetalLayout:



Collaboration diagram for PetalLayout:



Public Member Functions

- PetalLayout ()
- void setSpacing (Qt::Orientations orientation, qreal spacing) Q_DECL_OVERRIDE
- qreal spacing (Qt::Orientation orientation) const Q_DECL_OVERRIDE
- void setGeometry (const QRectF &geometry) Q_DECL_OVERRIDE
- int count () const Q_DECL_OVERRIDE
- QGraphicsLayoutItem * itemAt (int index) const Q_DECL_OVERRIDE
- void removeAt (int index) Q_DECL_OVERRIDE
- void addItem (QGraphicsLayoutItem *item) Q_DECL_OVERRIDE
- void clearAll () Q_DECL_OVERRIDE
- void setNrOfPetals (int value)
- void setRadius (qreal value)
- QList< QGraphicsLayoutItem * > & items () Q_DECL_OVERRIDE

Protected Member Functions

- QSizeF sizeHint (Qt::SizeHint which, const QSizeF &constraint=QSizeF()) const Q_DECL_OVERRIDE

7.25.1 Detailed Description

Definition at line 12 of file PetalLayout.hpp.

7.25.2 Constructor & Destructor Documentation

7.25.2.1 PetalLayout::PetalLayout ( ) [explicit]

Definition at line 3 of file PetalLayout.cpp.

7.25.3 Member Function Documentation

7.25.3.1 void PetalLayout::addItem (QGraphicsLayoutItem * *item*) [inline], [virtual]

inserts the item to the front of the list

Implements AbstractGraphicsLayout.

Definition at line 23 of file PetalLayout.hpp.

7.25.3.2 void PetalLayout::clearAll ( ) [virtual]

deletes and removes every element from the layout

Implements AbstractGraphicsLayout.

Definition at line 23 of file PetalLayout.cpp.

7.25.3.3 int PetalLayout::count ( ) const [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 19 of file PetalLayout.hpp.

7.25.3.4 QGraphicsLayoutItem* PetalLayout::itemAt (int *index*) const [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 20 of file PetalLayout.hpp.

7.25.3.5 QList<QGraphicsLayoutItem*>& PetalLayout::items ( ) [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 27 of file PetalLayout.hpp.

7.25.3.6 void PetalLayout::removeAt (int *index*) [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 21 of file PetalLayout.hpp.

7.25.3.7 void PetalLayout::setGeometry (const QRectF & *geometry*) [virtual]

Implements AbstractGraphicsLayout.

Definition at line 40 of file PetalLayout.cpp.

7.25.3.8 void PetalLayout::setNrOfPetals (int *value*) [inline]

Definition at line 25 of file PetalLayout.hpp.

7.25.3.9 void PetalLayout::setRadius (qreal *value*) [inline]

Definition at line 26 of file PetalLayout.hpp.

7.25.3.10 void PetalLayout::setSpacing (Qt::Orientations *orientation,* qreal *spacing*) [virtual]

Implements AbstractGraphicsLayout.

Definition at line 31 of file PetalLayout.cpp.

7.25.3.11 QSizeF PetalLayout::sizeHint (Qt::SizeHint *which,* const QSizeF & *constraint* = QSizeF()) const [protected], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 107 of file PetalLayout.cpp.

7.25.3.12 qreal PetalLayout::spacing (Qt::Orientation *orientation*) const [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 17 of file PetalLayout.hpp.

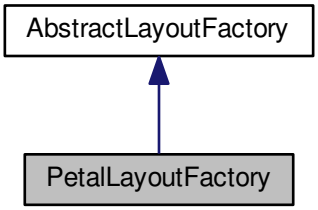
The documentation for this class was generated from the following files:

- layouts/PetalLayout.hpp
- layouts/PetalLayout.cpp

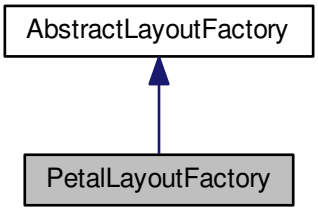
7.26 PetalLayoutFactory Class Reference

`#include <PetalLayoutFactory.hpp>`

Inheritance diagram for PetalLayoutFactory:



Collaboration diagram for PetalLayoutFactory:



Public Member Functions

- PetalLayoutFactory ()=default
- PetalLayout * makeLayout ()

7.26.1 Detailed Description

Definition at line 8 of file PetalLayoutFactory.hpp.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 PetalLayoutFactory::PetalLayoutFactory ( ) [default]

7.26.3 Member Function Documentation

7.26.3.1 PetalLayout* PetalLayoutFactory::makeLayout ( ) [inline], [virtual]

creates a new AbstractGraphicsLayout instance

Implements AbstractLayoutFactory.

Definition at line 12 of file PetalLayoutFactory.hpp.

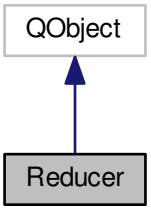
The documentation for this class was generated from the following file:

- utils/PetalLayoutFactory.hpp

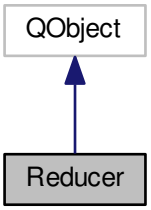
7.27 Reducer Class Reference

`#include <Reducer.hpp>`

Inheritance diagram for Reducer:



Collaboration diagram for Reducer:



Signals

- void imageReady (const GraphicsImage &image)

Public Member Functions

- Reducer ()=default
- ~Reducer ()=default
- void operator() (QList< GraphicsImage > &images, const GraphicsImage &image)

7.27.1 Detailed Description

Definition at line 12 of file Reducer.hpp.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 Reducer::Reducer ( ) [default]

7.27.2.2 Reducer::~Reducer ( ) [default]

7.27.3 Member Function Documentation

7.27.3.1 void Reducer::imageReady (const GraphicsImage & *image*) [signal]

7.27.3.2 void Reducer::operator() (QList< GraphicsImage > & *images*, const GraphicsImage & *image*) [inline]

Definition at line 17 of file Reducer.hpp.

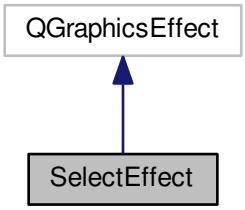
The documentation for this class was generated from the following file:

- utils/image_load/Reducer.hpp

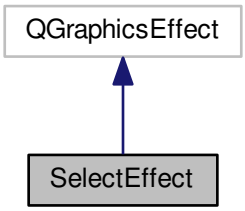
7.28 SelectEffect Class Reference

`#include <SelectEffect.hpp>`

Inheritance diagram for SelectEffect:



Collaboration diagram for SelectEffect:



Public Member Functions

- SelectEffect (qreal offset=1.2)
- QRectF boundingRectFor (const QRectF &sourceRect) const
- void setColor (const QColor &color)
- void setOffset (const QPointF &offset)

Protected Member Functions

- void draw (QPainter *painter)

7.28.1 Detailed Description

Definition at line 8 of file SelectEffect.hpp.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 SelectEffect::SelectEffect (qreal *offset* = 1.2) [inline]

Definition at line 12 of file SelectEffect.hpp.

7.28.3 Member Function Documentation

7.28.3.1 QRectF SelectEffect::boundingRectFor (const QRectF & *sourceRect*) const [inline]

Definition at line 17 of file SelectEffect.hpp.

7.28.3.2 void SelectEffect::draw (QPainter * *painter*) [inline], [protected]

Definition at line 25 of file SelectEffect.hpp.

7.28.3.3 void SelectEffect::setColor (const QColor & *color*) [inline]

Definition at line 21 of file SelectEffect.hpp.

7.28.3.4 void SelectEffect::setOffset (const QPointF & *offset*) [inline]

Definition at line 22 of file SelectEffect.hpp.

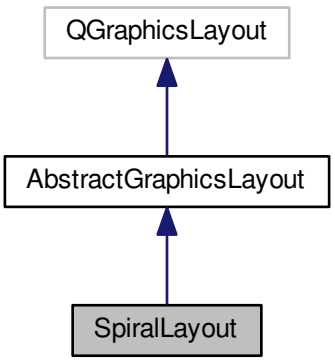
The documentation for this class was generated from the following file:

- utils/graphics/SelectEffect.hpp

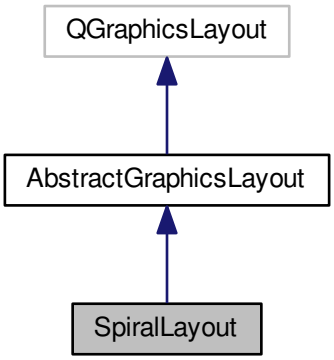
7.29 SpiralLayout Class Reference

`#include <SpiralLayout.hpp>`

Inheritance diagram for SpiralLayout:



Collaboration diagram for SpiralLayout:



Public Member Functions

- SpiralLayout ()
- void setSpacing (Qt::Orientations orientation, qreal spacing) Q_DECL_OVERRIDE
- qreal spacing (Qt::Orientation orientation) const Q_DECL_OVERRIDE
- void setGeometry (const QRectF &geometry) Q_DECL_OVERRIDE
- int count () const Q_DECL_OVERRIDE
- QGraphicsLayoutItem * itemAt (int index) const Q_DECL_OVERRIDE
- void removeAt (int index) Q_DECL_OVERRIDE
- void addItem (QGraphicsLayoutItem *item) Q_DECL_OVERRIDE
- void clearAll () Q_DECL_OVERRIDE
- QList< QGraphicsLayoutItem * > & items () Q_DECL_OVERRIDE
- void setTurn (int value)
- void setDistance (int value)

Protected Member Functions

- QSizeF sizeHint (Qt::SizeHint which, const QSizeF &constraint=QSizeF()) const Q_DECL_OVERRIDE

7.29.1 Detailed Description

Definition at line 13 of file SpiralLayout.hpp.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 SpiralLayout::SpiralLayout ( ) [explicit]

Definition at line 3 of file SpiralLayout.cpp.

7.29.3 Member Function Documentation

7.29.3.1 void SpiralLayout::addItem (QGraphicsLayoutItem * *item*) [inline], [virtual]

inserts the item to the front of the list

Implements AbstractGraphicsLayout.

Definition at line 24 of file SpiralLayout.hpp.

7.29.3.2 void SpiralLayout::clearAll ( ) [virtual]

deletes and removes every element from the layout

Implements AbstractGraphicsLayout.

Definition at line 23 of file SpiralLayout.cpp.

7.29.3.3 int SpiralLayout::count ( ) const [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 20 of file SpiralLayout.hpp.

7.29.3.4 QGraphicsLayoutItem* SpiralLayout::itemAt (int *index*) const [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 21 of file SpiralLayout.hpp.

7.29.3.5 QList<QGraphicsLayoutItem*>& SpiralLayout::items ( ) [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 26 of file SpiralLayout.hpp.

7.29.3.6 void SpiralLayout::removeAt (int *index*) [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 22 of file SpiralLayout.hpp.

7.29.3.7 void SpiralLayout::setDistance (int *value*) [inline]

Definition at line 29 of file SpiralLayout.hpp.

7.29.3.8 void SpiralLayout::setGeometry (const QRectF & *geometry*) [virtual]

Implements AbstractGraphicsLayout.

Definition at line 40 of file SpiralLayout.cpp.

7.29.3.9 void SpiralLayout::setSpacing (Qt::Orientations *orientation,* qreal *spacing*) [virtual]

Implements AbstractGraphicsLayout.

Definition at line 31 of file SpiralLayout.cpp.

7.29.3.10 void SpiralLayout::setTurn (int *value*) [inline]

Definition at line 28 of file SpiralLayout.hpp.

7.29.3.11 QSizeF SpiralLayout::sizeHint (Qt::SizeHint *which,* const QSizeF & *constraint* = QSizeF()) const [protected], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 84 of file SpiralLayout.cpp.

7.29.3.12 qreal SpiralLayout::spacing (Qt::Orientation *orientation*) const [inline], [virtual]

Implements AbstractGraphicsLayout.

Definition at line 18 of file SpiralLayout.hpp.

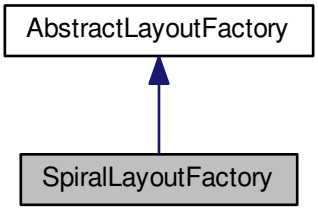
The documentation for this class was generated from the following files:

- layouts/SpiralLayout.hpp
- layouts/SpiralLayout.cpp

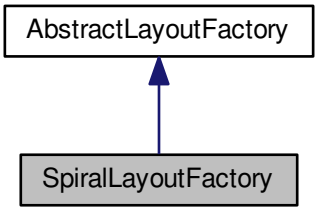
7.30 SpiralLayoutFactory Class Reference

`#include <SpiralLayoutFactory.hpp>`

Inheritance diagram for SpiralLayoutFactory:



Collaboration diagram for SpiralLayoutFactory:



Public Member Functions

- SpiralLayoutFactory ()=default
- SpiralLayout * makeLayout ()

7.30.1 Detailed Description

Definition at line 8 of file SpiralLayoutFactory.hpp.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 SpiralLayoutFactory::SpiralLayoutFactory ( ) [default]

7.30.3 Member Function Documentation

7.30.3.1 SpiralLayout* SpiralLayoutFactory::makeLayout ( ) [inline], [virtual]

creates a new AbstractGraphicsLayout instance

Implements AbstractLayoutFactory.

Definition at line 12 of file SpiralLayoutFactory.hpp.

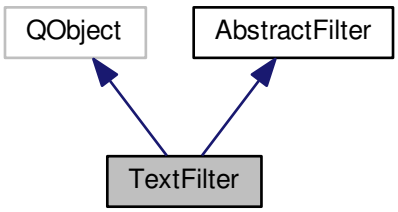
The documentation for this class was generated from the following file:

- utils/SpiralLayoutFactory.hpp

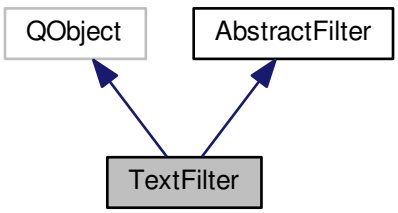
7.31 TextFilter Class Reference

`#include <TextFilter.hpp>`

Inheritance diagram for TextFilter:



Collaboration diagram for TextFilter:



Signals

- void changed (const QJsonArray &results)

Public Member Functions

- TextFilter (const DbContext &dbContext)
- ~TextFilter ()=default
- TextFilter * makeFilter (const DbContext &dbContext)
- QGroupBox * makeControl ()
  - *creates 2 QLineEdits to filter by title and summary*
- QPushButton & removeButton ()
- QString getText ()

7.31.1 Detailed Description

Definition at line 12 of file TextFilter.hpp.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 TextFilter::TextFilter (const DbContext & *dbContext*) [inline]

Definition at line 15 of file TextFilter.hpp.

7.31.2.2 TextFilter::~TextFilter ( ) [default]

7.31.3 Member Function Documentation

7.31.3.1 void TextFilter::changed (const QJsonArray & *results*) [signal]

7.31.3.2 QString TextFilter::getText ( ) [inline]

Definition at line 32 of file TextFilter.hpp.

7.31.3.3 QGroupBox * TextFilter::makeControl ( ) [virtual]

- creates 2 QLineEdits to filter by title and summary
- create a remove button for the filter

  Returns

  2 QLineEdits and a QPushButton

Implements AbstractFilter.

Definition at line 4 of file TextFilter.cpp.

7.31.3.4 TextFilter* TextFilter::makeFilter (const DbContext & *dbContext*) [inline], [virtual]

Implements AbstractFilter.

Definition at line 21 of file TextFilter.hpp.

7.31.3.5 QPushButton& TextFilter::removeButton ( ) [inline], [virtual]

Implements AbstractFilter.

Definition at line 30 of file TextFilter.hpp.

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

- filters/TextFilter.hpp
- filters/TextFilter.cpp

Chapter 8

File Documentation

8.1 db/DbContext.cpp File Reference

```
#include "DbContext.hpp"
```

Include dependency graph for DbContext.cpp:

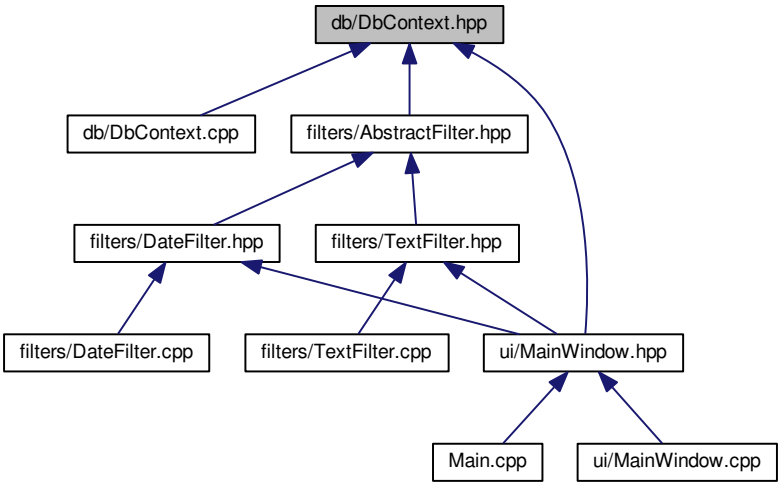
8.2 db/DbContext.hpp File Reference

```
#include <QString>
```

```
#include <QStringList>
#include <QFile>
#include <QJsonDocument>
#include <QJsonObject>
#include <QJsonValue>
#include <QJsonArray>
#include <QDebug>
#include <string>
#include <iostream>
#include <vector>
#include <chrono>
#include <ctime>
#include <bsoncxx/array/view.hpp>
#include <bsoncxx/builder/basic/document.hpp>
#include <bsoncxx/builder/stream/document.hpp>
#include <bsoncxx/builder/stream/helpers.hpp>
#include <bsoncxx/builder/stream/array.hpp>
#include <bsoncxx/document/value.hpp>
#include <bsoncxx/document/view.hpp>
#include <bsoncxx/json.hpp>
#include <bsoncxx/stdx/make_unique.hpp>
#include <bsoncxx/stdx/optional.hpp>
#include <bsoncxx/types.hpp>
#include <bsoncxx/types/value.hpp>
#include <bsoncxx/stdx/string_view.hpp>
#include <mongocxx/uri.hpp>
#include <mongocxx/client.hpp>
#include <mongocxx/pool.hpp>
#include <mongocxx/instance.hpp>
#include <mongocxx/logger.hpp>
#include <mongocxx/exception/operation_exception.hpp>
#include <mongocxx/v_noabi/mongocxx/exception/query_exception.hpp>
```

Include dependency graph for DbContext.hpp:

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



Classes

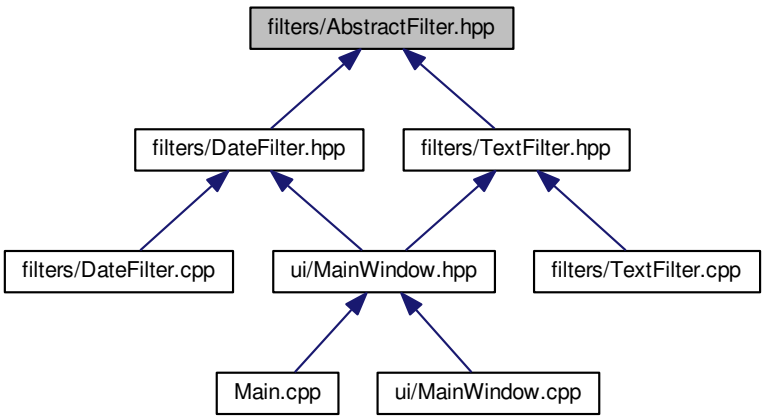
- class DbContext
- class DbContext::MongoAccess

8.3 filters/AbstractFilter.hpp File Reference

```
#include <QWidget>
#include <QPushButton>
#include "../db/DbContext.hpp"
```

Include dependency graph for AbstractFilter.hpp:

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



Classes

- class AbstractFilter

8.4 filters/DateFilter.cpp File Reference

```
#include "DateFilter.hpp"
```

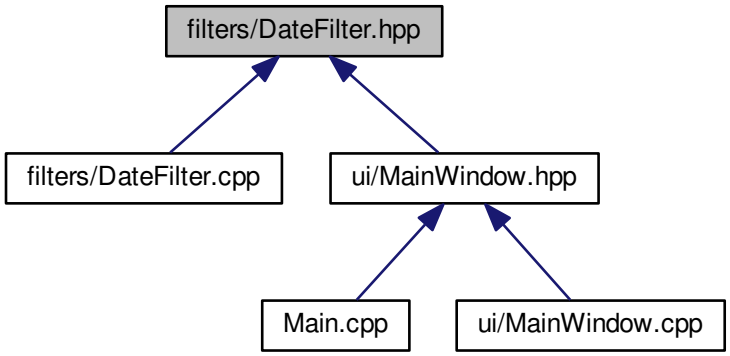
Include dependency graph for DateFilter.cpp:

8.5 filters/DateFilter.hpp File Reference

```
#include <chrono>
#include <string>
#include <QObject>
#include <QDateEdit>
#include <QGroupBox>
#include <QGridLayout>
#include <QList>
#include "AbstractFilter.hpp"
```

Include dependency graph for DateFilter.hpp:

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



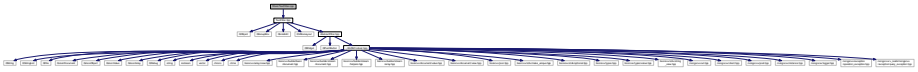
Classes

- class DateFilter

8.6 filters/TextFilter.cpp File Reference

```
#include "TextFilter.hpp"
```

Include dependency graph for TextFilter.cpp:

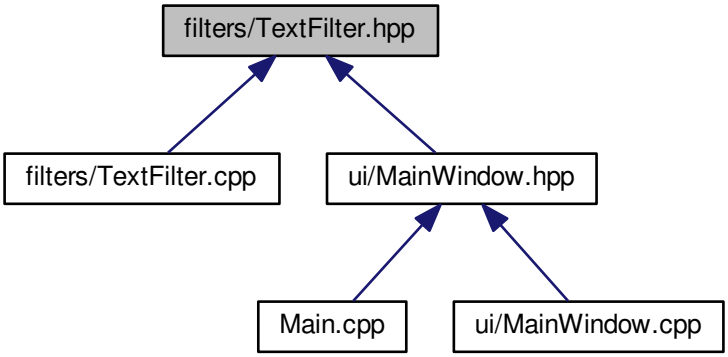


8.7 filters/TextFilter.hpp File Reference

```
#include <QObject>
#include <QGroupBox>
#include <QLineEdit>
#include <QVBoxLayout>
#include "AbstractFilter.hpp"
```

Include dependency graph for TextFilter.hpp:

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



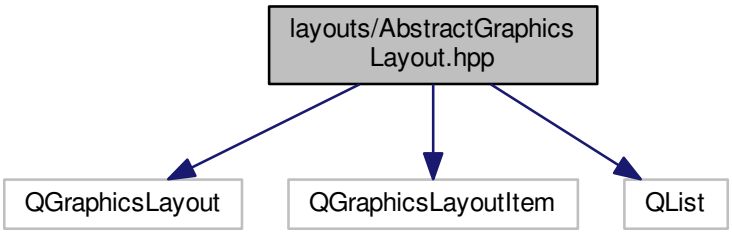
Classes

- class TextFilter

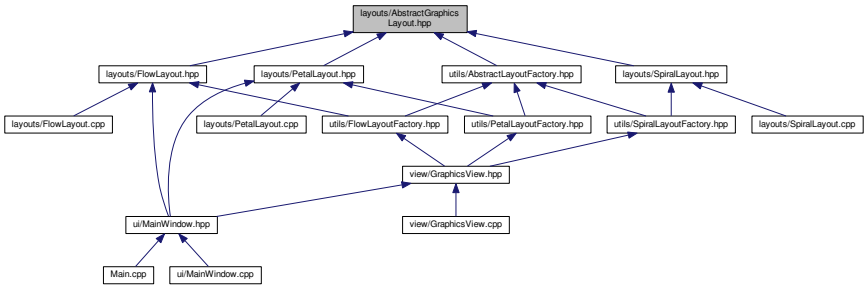
8.8 layouts/AbstractGraphicsLayout.hpp File Reference

```
#include <QGraphicsLayout>
#include <QGraphicsLayoutItem>
#include <QList>
```

Include dependency graph for AbstractGraphicsLayout.hpp:



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



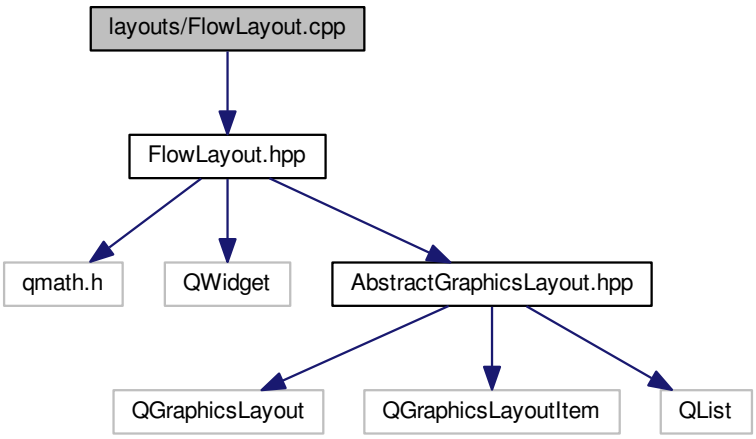
Classes

- class AbstractGraphicsLayout

8.9 layouts/FlowLayout.cpp File Reference

```
#include "FlowLayout.hpp"
```

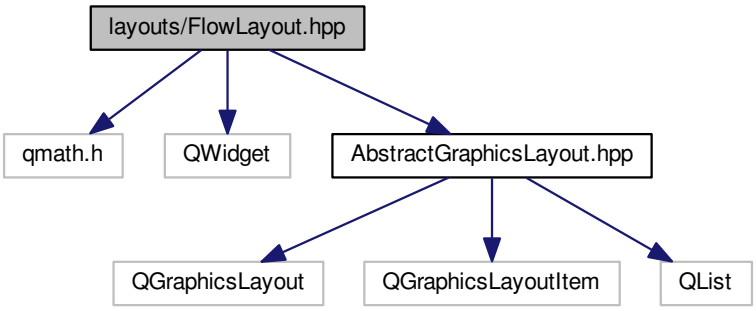
Include dependency graph for FlowLayout.cpp:



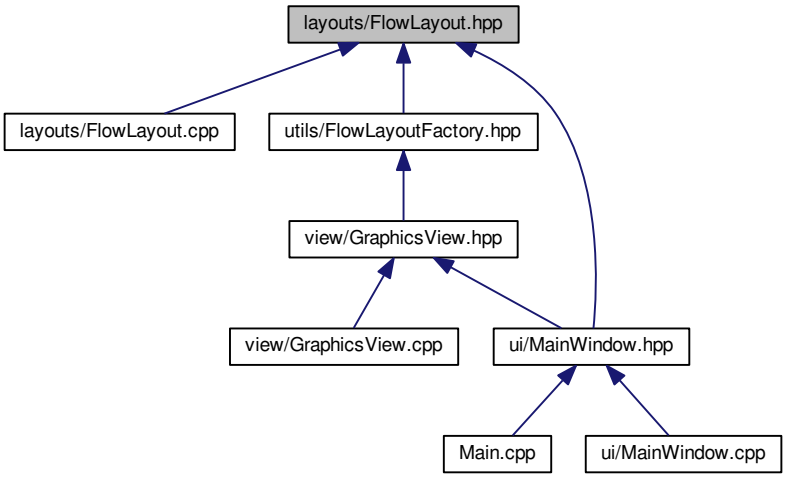
8.10 layouts/FlowLayout.hpp File Reference

```
#include <qmath.h>
#include <QWidget>
#include "AbstractGraphicsLayout.hpp"
```

Include dependency graph for FlowLayout.hpp:



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



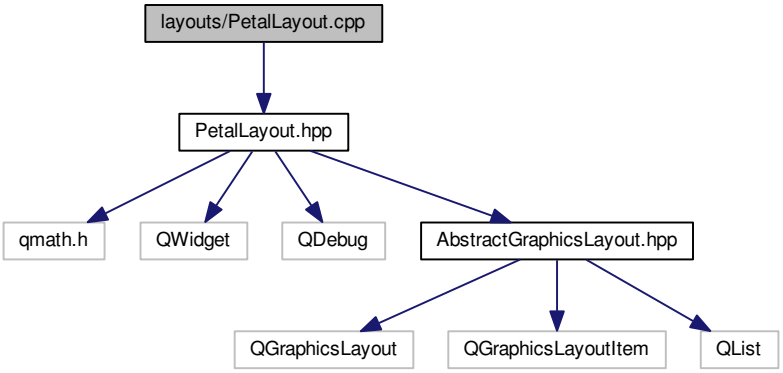
Classes

- class FlowLayout

8.11 layouts/PetalLayout.cpp File Reference

```
#include "PetalLayout.hpp"
```

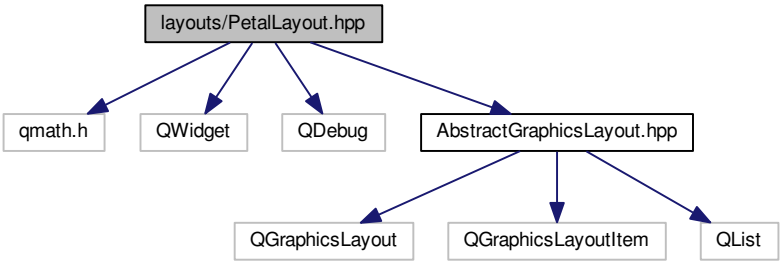
Include dependency graph for PetalLayout.cpp:



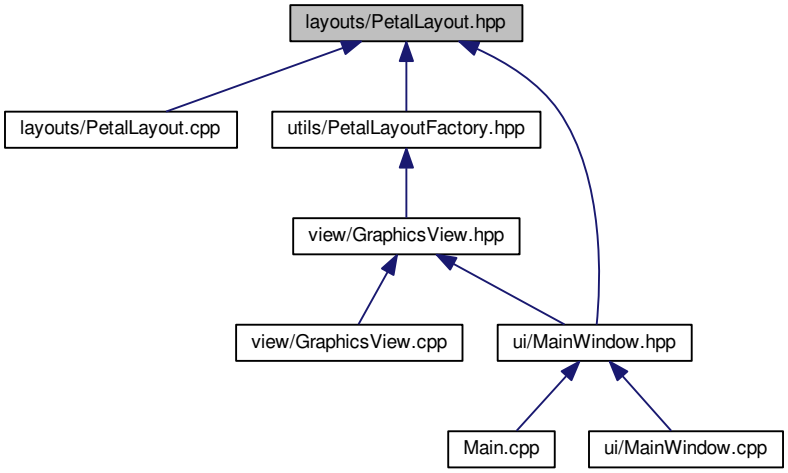
8.12 layouts/PetalLayout.hpp File Reference

```
#include <qmath.h>
#include <QWidget>
#include <QDebug>
#include "AbstractGraphicsLayout.hpp"
```

Include dependency graph for PetalLayout.hpp:



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



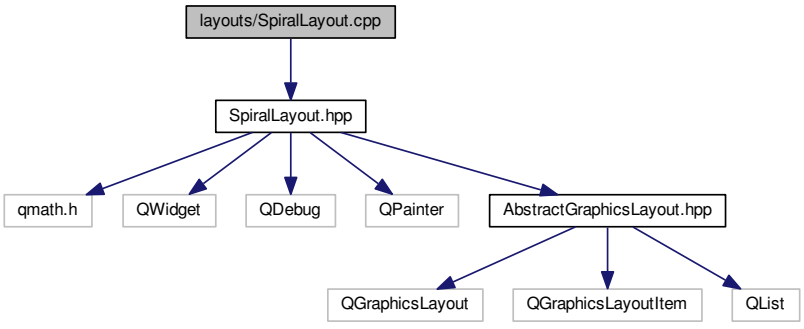
Classes

- class PetalLayout

8.13 layouts/SpiralLayout.cpp File Reference

```
#include "SpiralLayout.hpp"
```

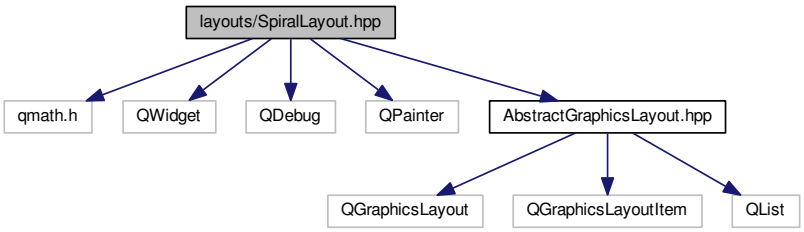
Include dependency graph for SpiralLayout.cpp:



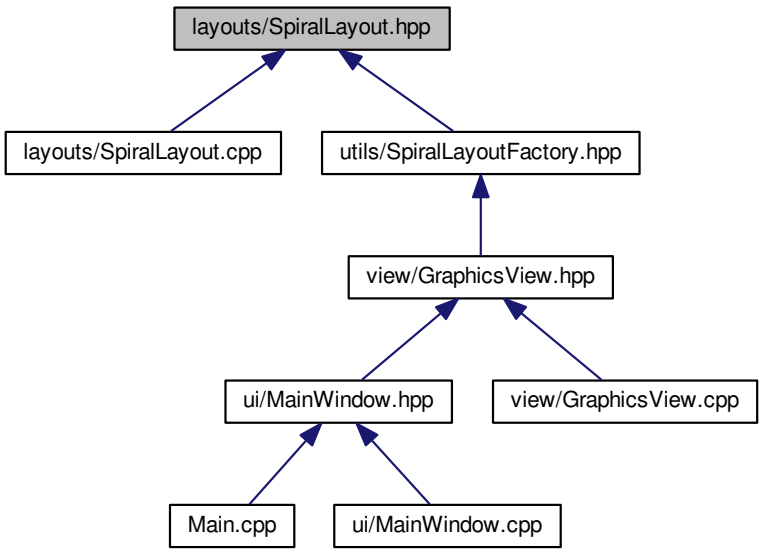
8.14 layouts/SpiralLayout.hpp File Reference

```
#include <qmath.h>
#include <QWidget>
#include <QDebug>
#include <QPainter>
#include "AbstractGraphicsLayout.hpp"
```

Include dependency graph for SpiralLayout.hpp:



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



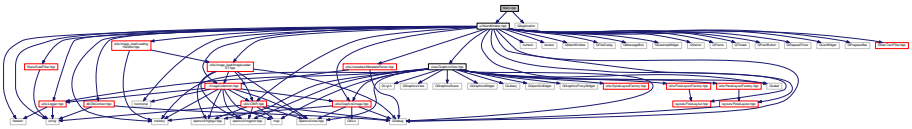
Classes

- class SpiralLayout

8.15 Main.cpp File Reference

```
#include "ui/MainWindow.hpp"
#include <QApplication>
```

Include dependency graph for Main.cpp:



Functions

- int main (int argc, char ∗argv[ ])

8.15.1 Function Documentation

8.15.1.1 int main (int *argc,* char ∗ *argv[ ]*)

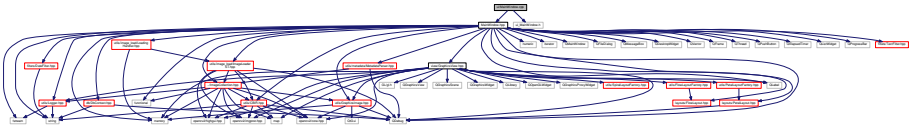
Definition at line 4 of file Main.cpp.

8.16 README.md File Reference

8.17 ui/MainWindow.cpp File Reference

```
#include "MainWindow.hpp"
#include "ui_MainWindow.h"
```

Include dependency graph for MainWindow.cpp:



Typedefs

- using CollectionMap = std::map< GraphicsImage, cv::Mat, CBIR::MatCompare >

8.17.1 Typedef Documentation

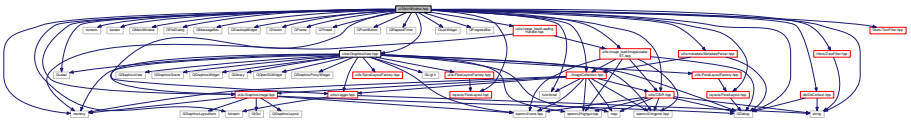
8.17.1.1 using CollectionMap = std::map<GraphicsImage, cv::Mat, CBIR::MatCompare>

Definition at line 5 of file MainWindow.cpp.

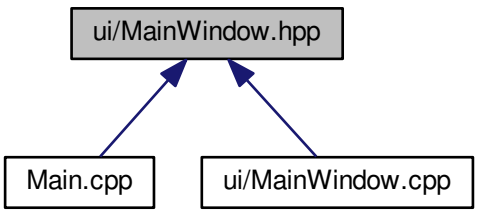
8.18 ui/MainWindow.hpp File Reference

```
#include <string>
#include <fstream>
#include <memory>
#include <functional>
#include <numeric>
#include <iterator>
#include <QMainWindow>
#include <QFileDialog>
#include <QMessageBox>
#include <QDesktopWidget>
#include <QVector>
#include <QFrame>
#include <QThread>
#include <QLabel>
#include <QPushButton>
#include <QElapsedTimer>
#include <QDebug>
#include <QListWidget>
#include <QProgressBar>
#include "view/GraphicsView.hpp"
#include "utils/GraphicsImage.hpp"
#include "utils/CBIR.hpp"
#include "utils/image_load/LoadingHandler.hpp"
#include "utils/image_load/ImageLoaderST.hpp"
#include "utils/ImageCollection.hpp"
#include "utils/metadata/MetadataParser.hpp"
#include "utils/Logger.hpp"
#include "layouts/FlowLayout.hpp"
#include "layouts/PetalLayout.hpp"
#include "db/DbContext.hpp"
#include "filters/DateFilter.hpp"
#include "filters/TextFilter.hpp"
```

Include dependency graph for MainWindow.hpp:



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



Classes

- class MainWindow

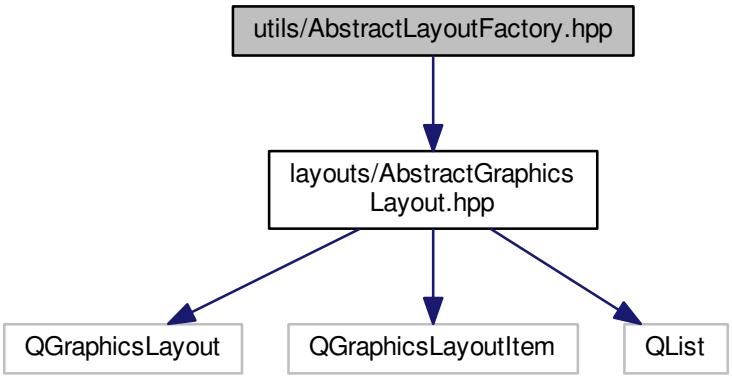
Namespaces

- Ui

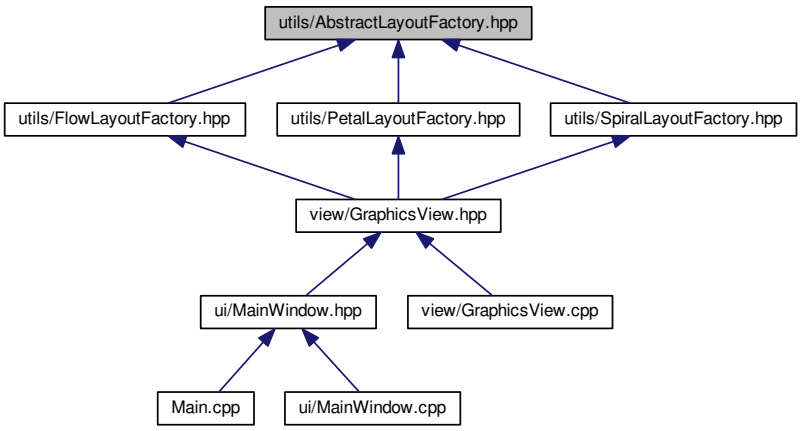
8.19 utils/AbstractLayoutFactory.hpp File Reference

```
#include "layouts/AbstractGraphicsLayout.hpp"
```

Include dependency graph for AbstractLayoutFactory.hpp:



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

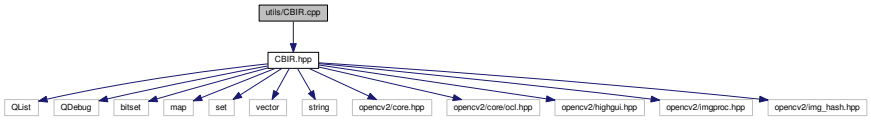


Classes

- class AbstractLayoutFactory

8.20 utils/CBIR.cpp File Reference

```
#include "CBIR.hpp"
```
Include dependency graph for CBIR.cpp:



Typedefs

- using ImageMap = std::multimap< cv::Mat, cv::Mat, CBIR::MatCompare >

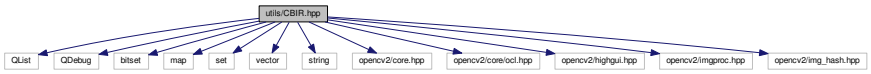
8.20.1 Typedef Documentation

8.20.1.1 using ImageMap = std::multimap<cv::Mat, cv::Mat, CBIR::MatCompare>

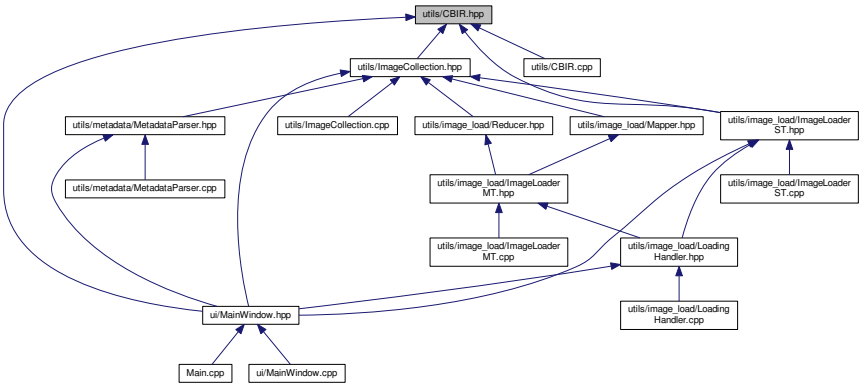
Definition at line 14 of file CBIR.cpp.

8.21 utils/CBIR.hpp File Reference

```
#include <QList>
#include <QDebug>
#include <bitset>
#include <map>
#include <set>
#include <vector>
#include <string>
#include <opencv2/core.hpp>
#include <opencv2/core/ocl.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/img_hash.hpp>
```
Include dependency graph for CBIR.hpp:



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



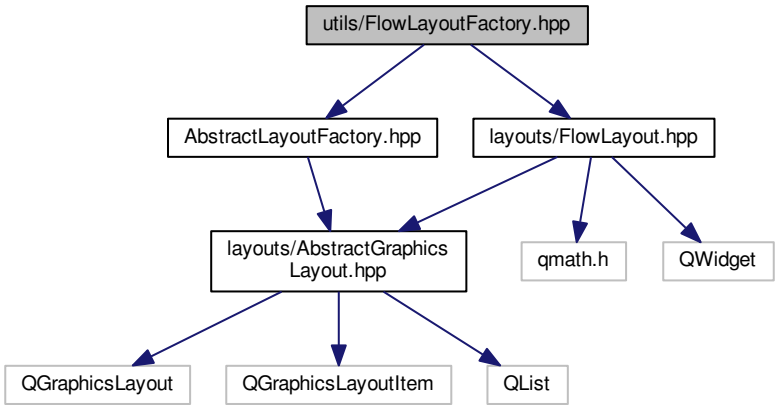
Classes

- class CBIR
- struct CBIR::MatKey
- struct CBIR::MatCompare

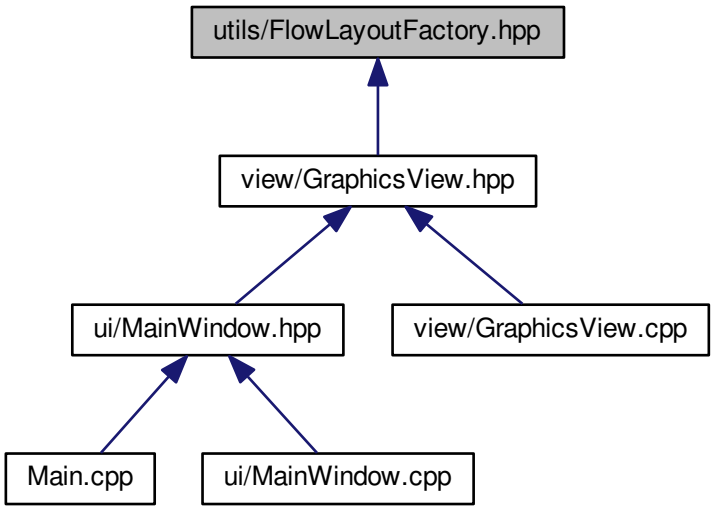
8.22 utils/FlowLayoutFactory.hpp File Reference

```
#include "AbstractLayoutFactory.hpp"
#include "layouts/FlowLayout.hpp"
```

Include dependency graph for FlowLayoutFactory.hpp:



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



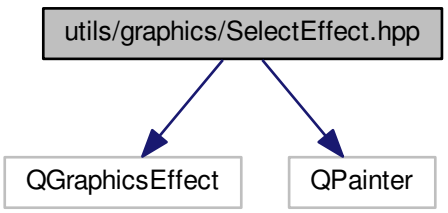
Classes

- class FlowLayoutFactory

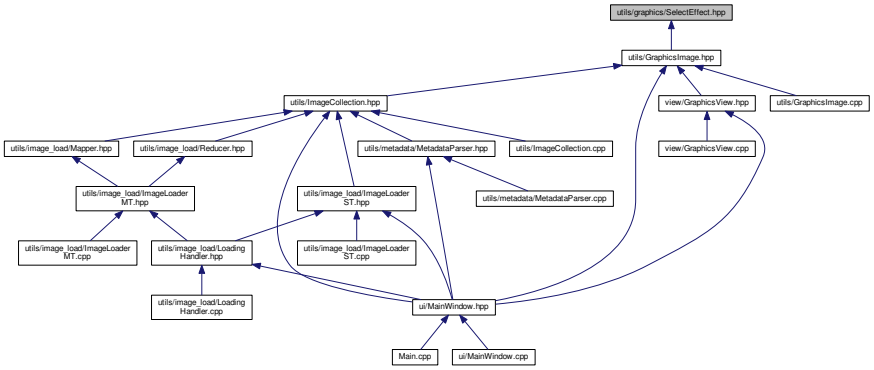
8.23 utils/graphics/SelectEffect.hpp File Reference

```
#include <QGraphicsEffect>
#include <QPainter>
```

Include dependency graph for SelectEffect.hpp:



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



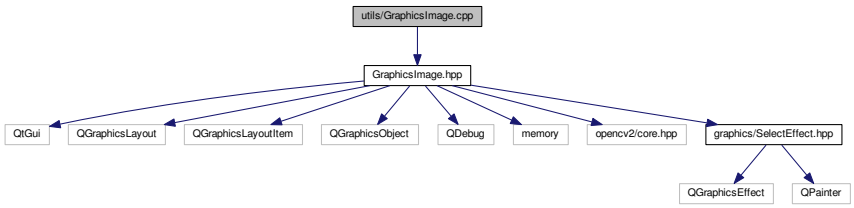
Classes

- class SelectEffect

8.24 utils/GraphicsImage.cpp File Reference

```
#include "GraphicsImage.hpp"
```

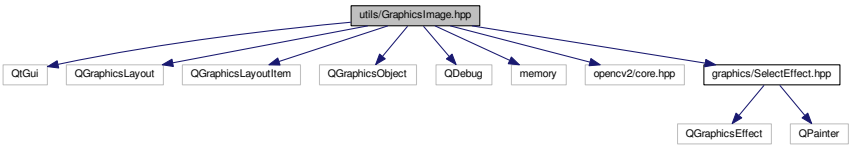
Include dependency graph for GraphicsImage.cpp:



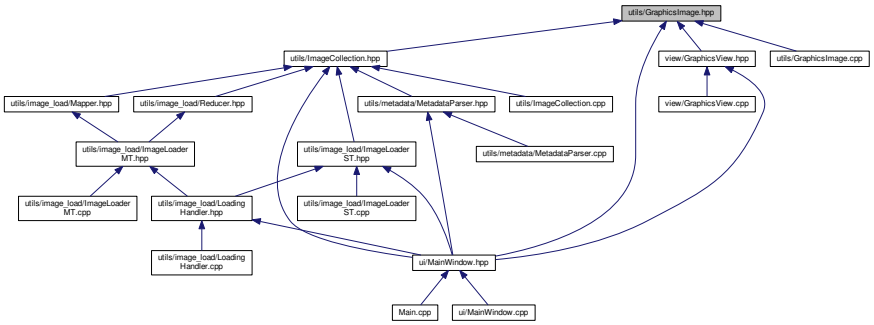
8.25 utils/GraphicsImage.hpp File Reference

```
#include <QtGui>
#include <QGraphicsLayout>
#include <QGraphicsLayoutItem>
#include <QGraphicsObject>
#include <QDebug>
#include <memory>
#include <opencv2/core.hpp>
#include "graphics/SelectEffect.hpp"
```

Include dependency graph for GraphicsImage.hpp:



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



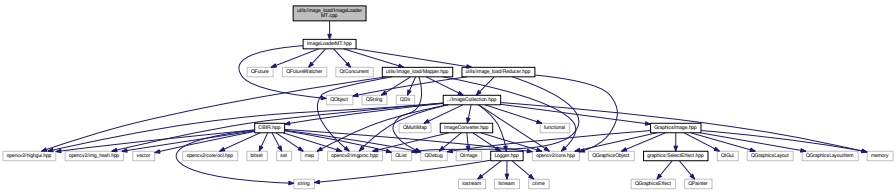
Classes

- class GraphicsImage

8.26 utils/image_load/ImageLoaderMT.cpp File Reference

```
#include "ImageLoaderMT.hpp"
```

Include dependency graph for ImageLoaderMT.cpp:

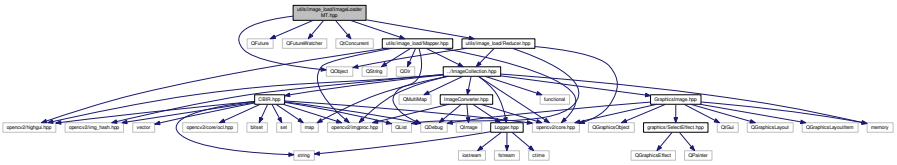


8.27 utils/image_load/ImageLoaderMT.hpp File Reference

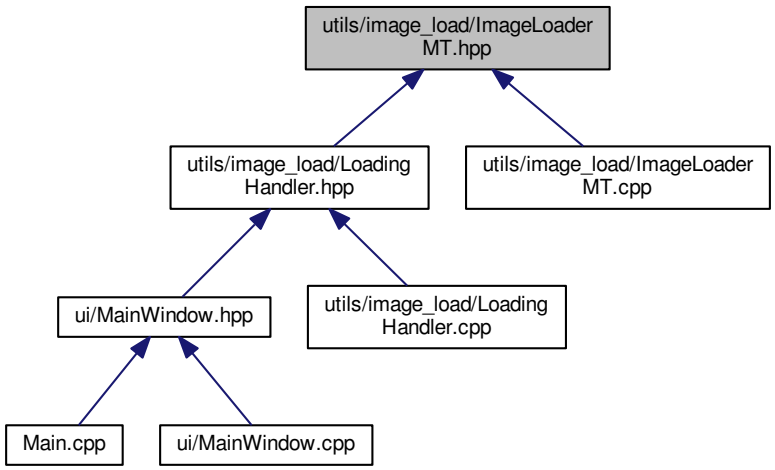
```
#include <QObject>
```

```
#include <QFuture>
#include <QFutureWatcher>
#include <QtConcurrent>
#include "utils/image_load/Mapper.hpp"
#include "utils/image_load/Reducer.hpp"
```

Include dependency graph for ImageLoaderMT.hpp:



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



Classes

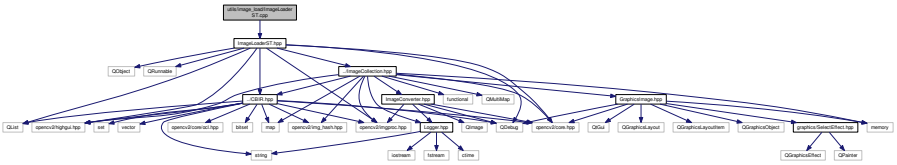
- class ImageLoaderMT

  handles image loading in a multi-threaded asynchronous way

8.28 utils/image_load/ImageLoaderST.cpp File Reference

```
#include "ImageLoaderST.hpp"
```

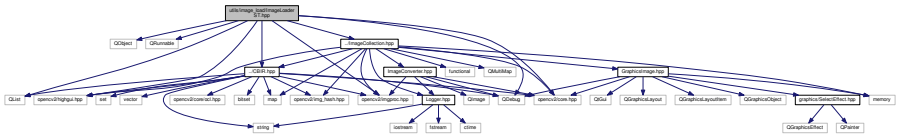
Include dependency graph for ImageLoaderST.cpp:



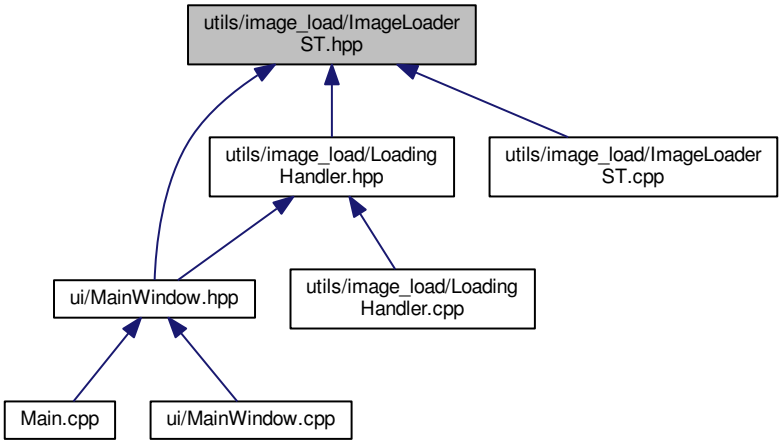
8.29 utils/image_load/ImageLoaderST.hpp File Reference

```
#include <QObject>
#include <QRunnable>
#include <QList>
#include <QDebug>
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
#include "../CBIR.hpp"
#include "../ImageCollection.hpp"
```

Include dependency graph for ImageLoaderST.hpp:



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



Classes

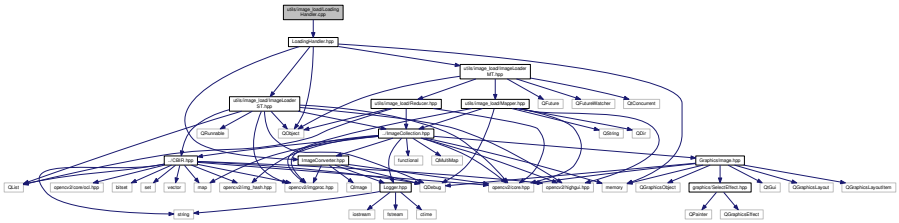
- class ImageLoaderST

  handles image loading in a single-threaded asyncronous way

8.30 utils/image_load/LoadingHandler.cpp File Reference

```
#include "LoadingHandler.hpp"
```

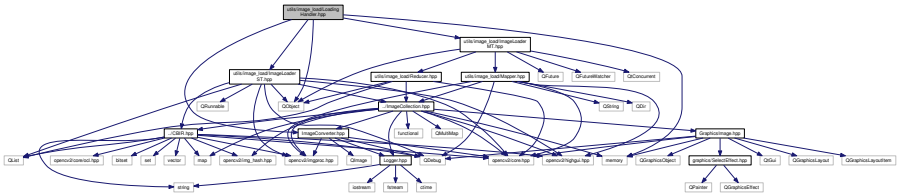
Include dependency graph for LoadingHandler.cpp:



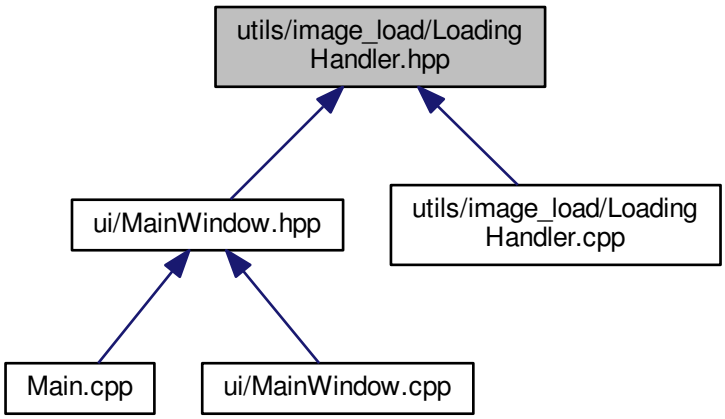
8.31 utils/image_load/LoadingHandler.hpp File Reference

```
#include <memory>
#include <QObject>
#include "utils/image_load/ImageLoaderST.hpp"
#include "utils/image_load/ImageLoaderMT.hpp"
#include "utils/ImageConverter.hpp"
```

Include dependency graph for LoadingHandler.hpp:



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



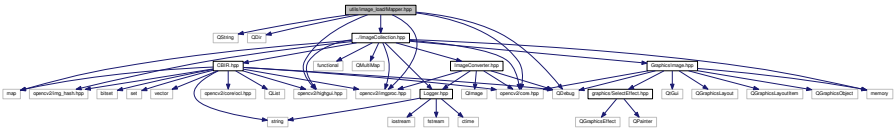
Classes

- class LoadingHandler

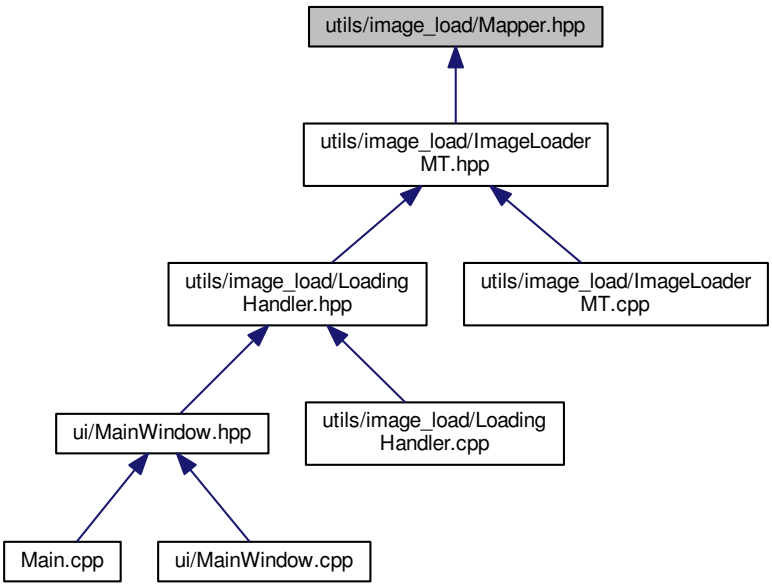
8.32 utils/image_load/Mapper.hpp File Reference

```
#include <QString>
#include <QDir>
#include <QDebug>
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
#include "../ImageCollection.hpp"
```

Include dependency graph for Mapper.hpp:



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



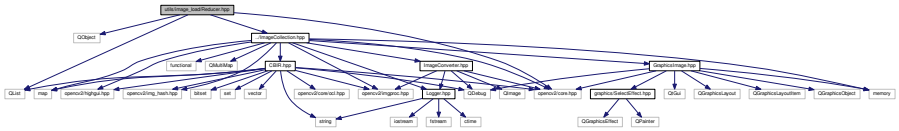
Classes

- class Mapper

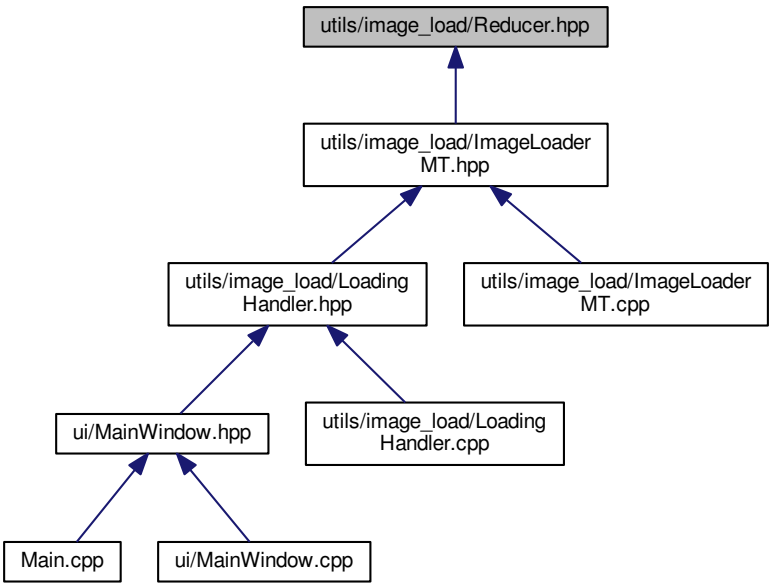
8.33 utils/image_load/Reducer.hpp File Reference

```
#include <QObject>
#include <QList>
#include <opencv2/core.hpp>
#include "../ImageCollection.hpp"
```

Include dependency graph for Reducer.hpp:



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



Classes

- class Reducer

8.34 utils/ImageCollection.cpp File Reference

```
#include "ImageCollection.hpp"
```

Include dependency graph for ImageCollection.cpp:

Typedefs

- using ImageMap = std::map< QString, ImageCollection::Collection >

8.34.1 Typedef Documentation

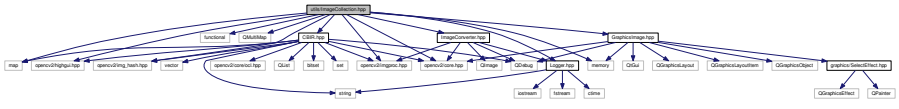
8.34.1.1 using ImageMap = std::map<QString, ImageCollection::Collection>

Definition at line 3 of file ImageCollection.cpp.

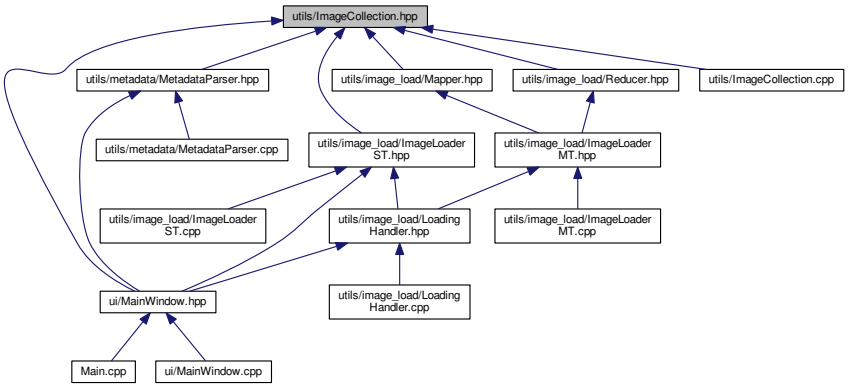
8.35 utils/ImageCollection.hpp File Reference

```
#include <memory>
#include <map>
#include <functional>
#include <QMultiMap>
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/img_hash.hpp>
#include "CBIR.hpp"
#include "Logger.hpp"
#include "GraphicsImage.hpp"
#include "ImageConverter.hpp"
```

Include dependency graph for ImageCollection.hpp:



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



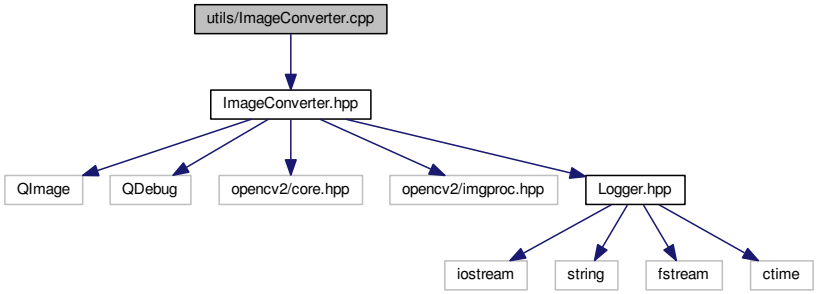
Classes

- class ImageCollection
- struct ImageCollection::Collection

8.36 utils/ImageConverter.cpp File Reference

```
#include "ImageConverter.hpp"
```

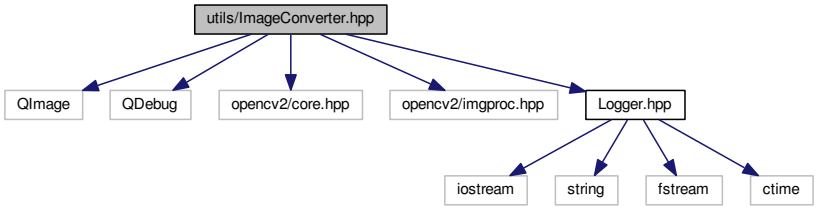
Include dependency graph for ImageConverter.cpp:



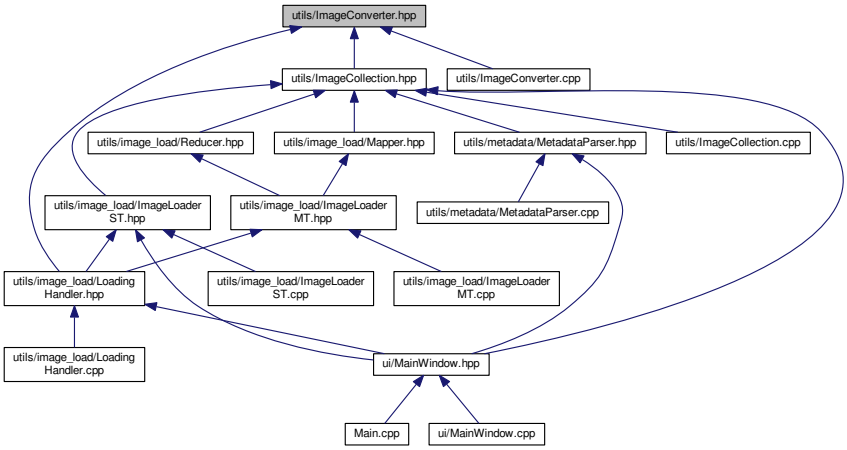
8.37 utils/ImageConverter.hpp File Reference

```
#include <QImage>
#include <QDebug>
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include "Logger.hpp"
```

Include dependency graph for ImageConverter.hpp:



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



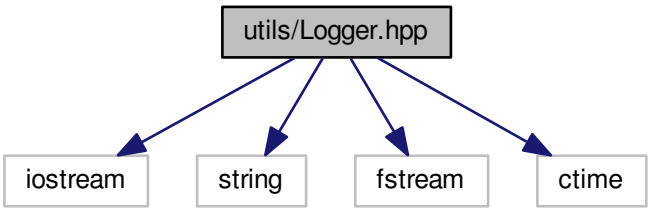
Classes

- class ImageConverter

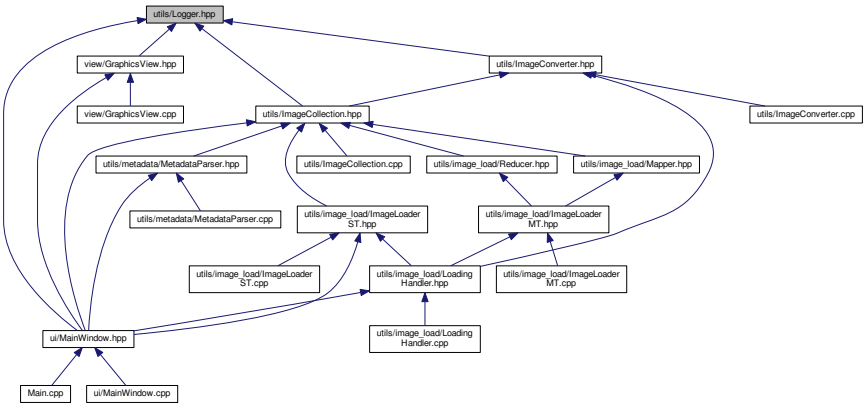
8.38 utils/Logger.hpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <ctime>
```

Include dependency graph for Logger.hpp:



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



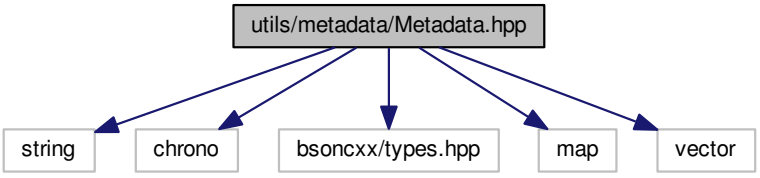
Classes

- class Logger

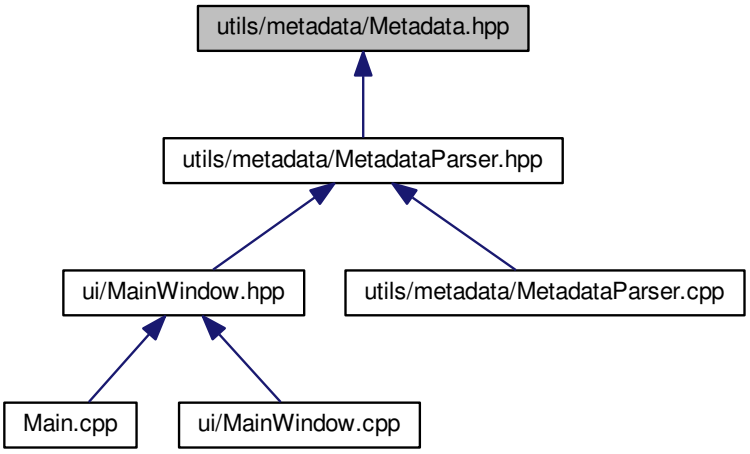
8.39 utils/metadata/Metadata.hpp File Reference

```
#include <string>
#include <chrono>
#include <bsoncxx/types.hpp>
#include <map>
#include <vector>
```

Include dependency graph for Metadata.hpp:



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



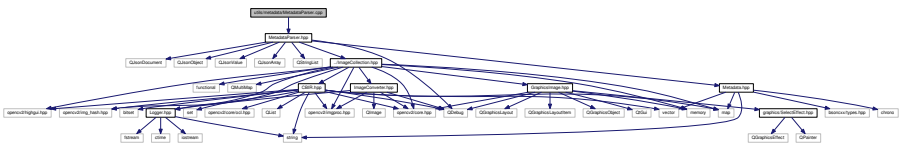
Classes

- class Metadata

8.40 utils/metadata/MetadataParser.cpp File Reference

```
#include "MetadataParser.hpp"
```

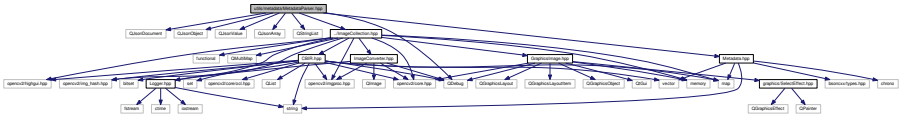
Include dependency graph for MetadataParser.cpp:



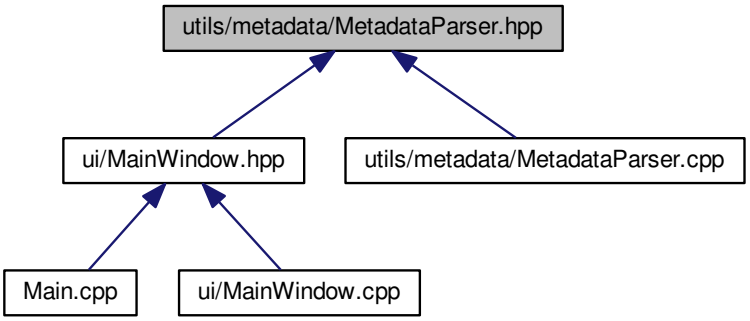
8.41 utils/metadata/MetadataParser.hpp File Reference

```
#include <QJsonDocument>
#include <QJsonObject>
#include <QJsonValue>
#include <QJsonArray>
#include <QStringList>
#include <QDebug>
#include "../ImageCollection.hpp"
#include "Metadata.hpp"
```

Include dependency graph for MetadataParser.hpp:



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



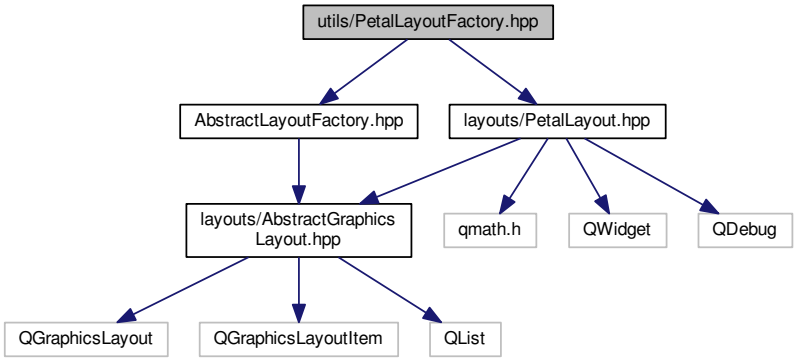
Classes

- class MetadataParser

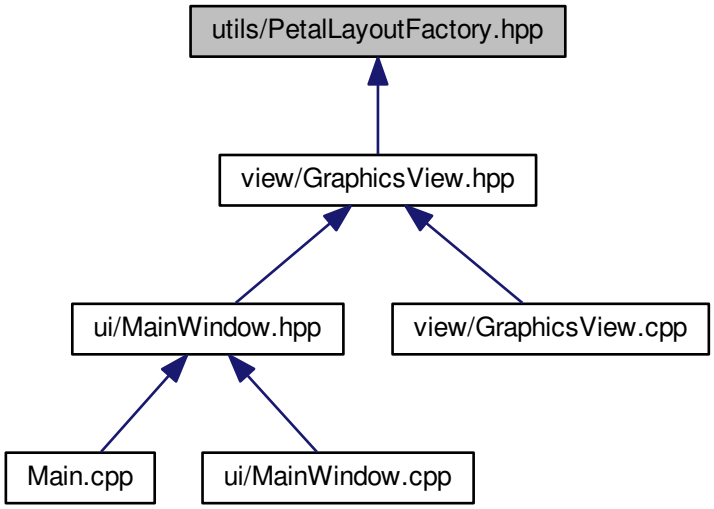
8.42 utils/PetalLayoutFactory.hpp File Reference

```
#include "AbstractLayoutFactory.hpp"
#include "layouts/PetalLayout.hpp"
```

Include dependency graph for PetalLayoutFactory.hpp:



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



Classes

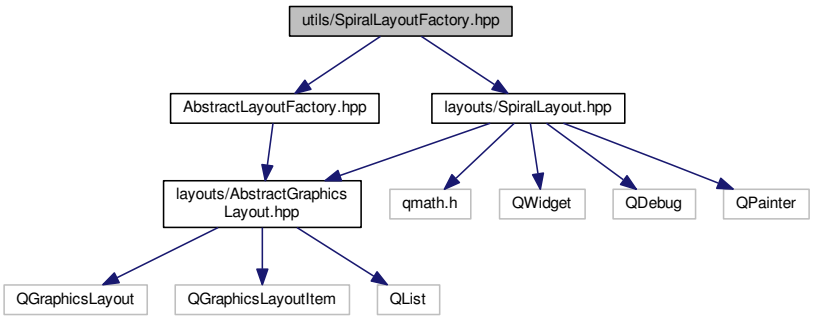
- class PetalLayoutFactory

8.43 utils/SpiralLayoutFactory.hpp File Reference

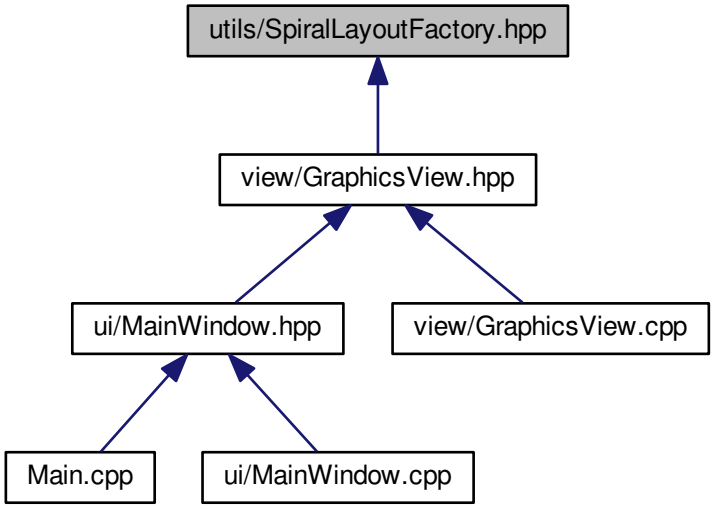
```
#include "AbstractLayoutFactory.hpp"
```

```
#include "layouts/SpiralLayout.hpp"
```

Include dependency graph for SpiralLayoutFactory.hpp:



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



Classes

- class SpiralLayoutFactory

8.44 view/GraphicsView.cpp File Reference

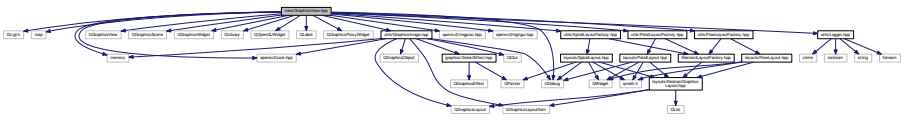
```
#include "GraphicsView.hpp"
```

Include dependency graph for GraphicsView.cpp:

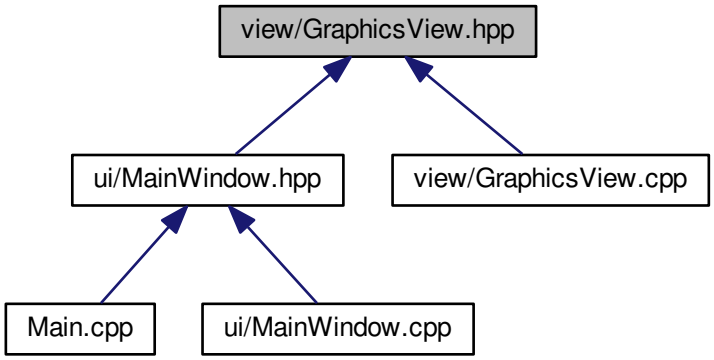
8.45 view/GraphicsView.hpp File Reference

```
#include <GL/gl.h>
#include <map>
#include <memory>
#include <QDebug>
#include <QGraphicsView>
#include <QGraphicsScene>
#include <QGraphicsWidget>
#include <QLibrary>
#include <QOpenGLWidget>
#include <QLabel>
#include <QGraphicsProxyWidget>
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
#include "utils/GraphicsImage.hpp"
#include "utils/FlowLayoutFactory.hpp"
#include "utils/PetalLayoutFactory.hpp"
#include "utils/SpiralLayoutFactory.hpp"
#include "utils/Logger.hpp"
```

Include dependency graph for GraphicsView.hpp:



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



Classes

- class GraphicsView

Index