[LE]**1** [CE] [RE] [LO] [CO] [RO]**1** [LE] [CE] [RE]**Generated by Doxygen**
[LO]**Generated by Doxygen** [CO] [RO]

labelsep=space,justification=centering,font=bf,singlelinecheck=off,skip=4pt,position=top

b

PluriNotes

2

Generated by Doxygen 1.8.13

Contents

Chapter 1

OL13

OL13

OL13

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

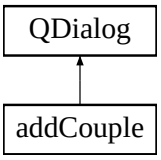
Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 addCouple Class Reference

Inheritance diagram for addCouple:



Public Slots

- void **updateModel_to** (QModelIndex index)
- void **on_toView_clicked** (QModelIndex i)
- void **on_save_clicked** ()

Signals

- void **addNewCouple** (QString, QString, QString, bool)

Public Member Functions

- **addCouple** (QWidget ∗parent=0)

5.1.1 Detailed Description

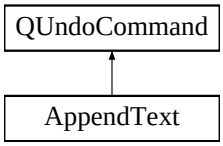
Definition at line 10 of file addcouple.h.

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

- OL13/addcouple.h
- OL13/addcouple.cpp

5.2 AppendText Class Reference

Inheritance diagram for AppendText:



Public Member Functions

- **AppendText** (QString ∗doc, const QString &text)
- virtual void **undo** ()
- virtual void **redo** ()

5.2.1 Detailed Description

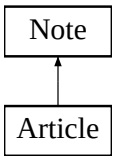
Definition at line 19 of file undoredo.h.

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

- OL13/undoredo.h

5.3 Article Class Reference

Inheritance diagram for Article:



Public Member Functions

- const QTextDocument & **getText** () const
- void **setText** (const QString &t)
- **Article** (const QString &i, const QString &ti, const QString &te)
- **Article** (const QString &i, const QString &ti, const QDateTime &cd, const QDateTime &lmd, bool iA, bool iD, const QString &te)
- **Article** (const Article &a)
- Article & operator= (const Article &a)

  Surcharge de l'opérateur = dans le cas nouvel article Article B=A.
- std::string toString () const

  Transforme un article en un flux ostream a afficher.
- void **saveNote** (QFile ∗file)

Additional Inherited Members

5.3.1 Detailed Description

Definition at line 130 of file notes.h.

5.3.2 Member Function Documentation

5.3.2.1 toString()

```
std::string Article::toString ( ) const  [virtual]
```

Transforme un article en un flux ostream a afficher.

Returns

Le flux ostream

Reimplemented from Note.

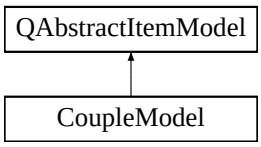
Definition at line 194 of file notes.cpp.

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

- OL13/notes.h
- OL13/notes.cpp

5.4 CoupleModel Class Reference

Inheritance diagram for CoupleModel:



Public Member Functions

- **CoupleModel** (QObject *parent=nullptr)
- QVariant **headerData** (int section, Qt::Orientation orientation, int role=Qt::DisplayRole) const override
- bool **setHeaderData** (int section, Qt::Orientation orientation, const QVariant &value, int role=Qt::EditRole) override
- QModelIndex **index** (int row, int column, const QModelIndex &parent=QModelIndex()) const override
- QModelIndex **parent** (const QModelIndex &index) const override
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const override
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const override
- bool **hasChildren** (const QModelIndex &parent=QModelIndex()) const override
- bool **canFetchMore** (const QModelIndex &parent) const override
- void **fetchMore** (const QModelIndex &parent) override
- QVariant **data** (const QModelIndex &index, int role=Qt::DisplayRole) const override
- bool **setData** (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole) override
- Qt::ItemFlags **flags** (const QModelIndex &index) const override

5.4.1 Detailed Description

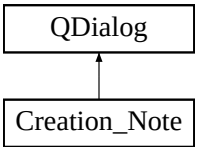
Definition at line 6 of file couplemodel.h.

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

- OL13/couplemodel.h
- OL13/couplemodel.cpp

5.5 Creation_Note Class Reference

Inheritance diagram for Creation_Note:



Public Slots

- void **Creer_Note** ()
- void **select_type** (int type)
- void **fenclose** ()
- void **activer_E_title_not_null** ()
- void **activer_E_note_not_null** (bool status)
- void **activer_E_id_not_null** ()
- void **activer_Creer** ()

Signals

- void **change_Creer** ()
- void **newNote** (Note &n)

Public Member Functions

- **Creation_Note** (QWidget ∗parent)

5.5.1 Detailed Description

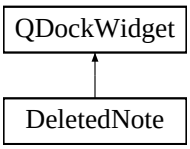
Definition at line 19 of file Creation_Note.h.

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

- OL13/Creation_Note.h
- OL13/Creation_Note.cpp

5.6 DeletedNote Class Reference

Inheritance diagram for DeletedNote:



Public Member Functions

- **DeletedNote** (QWidget ∗parent=0)

5.6.1 Detailed Description

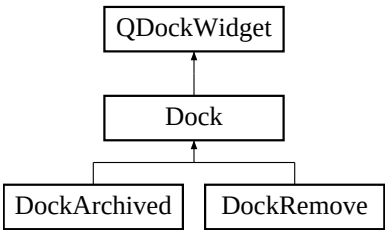
Definition at line 10 of file deletednote.h.

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

- OL13/deletednote.h
- OL13/deletednote.cpp

5.7 Dock Class Reference

Inheritance diagram for Dock:



Public Slots

- virtual void **on_remove_clicked** ()=0
- virtual void **update_archNoteModel** ()=0
- void **on_aff_clicked** ()
- void **on_ArchView_clicked** (QModelIndex i)

Signals

- void **update_removeDock** ()
- void **selection** (QString, int)

Public Member Functions

- **Dock** (QWidget ∗parent=0)

Protected Attributes

- Ui::DockArchived ∗ **ui**
- QString **currentNote**
- QStandardItemModel ∗ **ArchNote**

5.7.1 Detailed Description

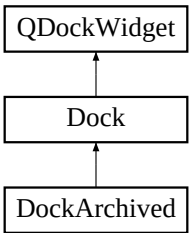
Definition at line 11 of file dockarchived.h.

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

- OL13/dockarchived.h
- OL13/dockarchived.cpp

5.8 DockArchived Class Reference

Inheritance diagram for DockArchived:



Public Slots

- void **on_remove_clicked** ()

Public Member Functions

- **DockArchived** (QWidget ∗parent)
- void **update_archNoteModel** ()

Additional Inherited Members

5.8.1 Detailed Description

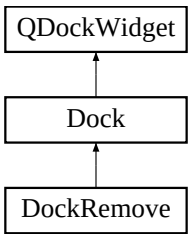
Definition at line 34 of file dockarchived.h.

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

- OL13/dockarchived.h
- OL13/dockarchived.cpp

5.9 DockRemove Class Reference

Inheritance diagram for DockRemove:



Public Slots

- void **on_remove_clicked** ()

Public Member Functions

- void **update_archNoteModel** ()
- **DockRemove** (QWidget ∗parent)

Additional Inherited Members

5.9.1 Detailed Description

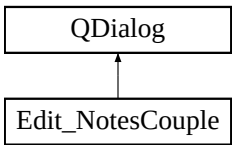
Definition at line 53 of file dockarchived.h.

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

- OL13/dockarchived.h
- OL13/dockarchived.cpp

5.10 Edit_NotesCouple Class Reference

Inheritance diagram for Edit_NotesCouple:



Public Slots

- void **fermer** ()
- void **eneableE_label** ()

Signals

- void **newCouple** (Note ∗, Note ∗, QString, bool)
- void **setCouple** (QString)

Public Member Functions

- Edit_NotesCouple (Note ∗n1, Note ∗n2, QWidget ∗parent=nullptr, bool s=false)

5.10.1 Detailed Description

Definition at line 60 of file qrelations.h.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 Edit_NotesCouple()

```
Edit_NotesCouple::Edit_NotesCouple (
            Note * n1,
            Note * n2,
            QWidget * parent = nullptr,
            bool s = false )
```

Connect:

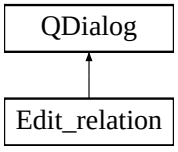
Definition at line 188 of file qrelations.cpp.

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

- OL13/qrelations.h
- OL13/qrelations.cpp

5.11 Edit_relation Class Reference

Inheritance diagram for Edit_relation:



Public Slots

- void **clicSelection** ()
- void **enabledAppend** ()
- void **addCouple** (Note ∗n1, Note ∗n2, QString label, bool s)

Signals

- void **newRelation** ()

Public Member Functions

- **Edit_relation** (QStandardItemModel ∗m, QString id, QWidget ∗parent)

5.11.1 Detailed Description

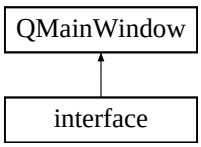
Definition at line 116 of file qrelations.h.

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

- OL13/qrelations.h
- OL13/qrelations.cpp

5.12 interface Class Reference

Inheritance diagram for interface:



Public Slots

- void **update_model** ()
- void **E_relation** ()
- void **Aff_relation** ()
- void **ViderCorbeille** ()
- void addAction_new_rel ()

  fonction outils, permettant de connecter, afficher l'action d'achiver une note a la page courante.
- void OuvrirFichier ()

  Fenêtre permettant de selectionner/chargé un fichier de sauvegarde Met à jour le manager de note et les docks.
- void CreerNote ()

  Ouvre une fenetre de dialogue de type Creation_Note, pour permettre la création d'une note. Connecte cette fennetre au docks.
- void afficher_note (QString id, int i)

  interface::afficher_note
- void **supp_dock_editer** ()
- void **supp_dock_aff_rel** ()
- void **supprimer_note** ()
- void close_page_note ()

  Ferme la note en cours, si elle est bien encore ouverte.
- void save ()

  Permet de choisir un fichier de sauvegarde.

Signals

- void **S_update_model** ()
- void **L_update_model** ()
- void **A_update_model** ()

Public Member Functions

- interface ()

  Constructeur de la classe interface Fenêtre principale de l'application, gére tous les docks, effectue la liason entre les docks, les widget, et les boites de dialogues.

5.12.1 Detailed Description

Definition at line 53 of file interface.h.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 interface()

```
interface::interface ( )
```

Constructeur de la classe interface Fenêtre principale de l'application, gére tous les docks, effectue la liason entre les docks, les widget, et les boites de dialogues.

Parameters

Definition at line 29 of file interface.cpp.

5.12.3 Member Function Documentation

5.12.3.1 afficher_note

```
void interface::afficher_note (
            QString id,
            int i )  [slot]
```

interface::afficher_note

Parameters

id	
i	

Definition at line 348 of file interface.cpp.

5.12.3.2 CreerNote

```
void interface::CreerNote ( )  [slot]
```

Ouvre une fenetre de dialogue de type Creation_Note, pour permettre la création d'une note. Connecte cette fennetre au docks.

Parameters

Definition at line 257 of file interface.cpp.

5.12.3.3 OuvrirFichier

```
void interface::OuvrirFichier ( )  [slot]
```

Fenêtre permettant de selectionner/chargé un fichier de sauvegarde Met à jour le manager de note et les docks.

Parameters

Definition at line 197 of file interface.cpp.

5.12.3.4 save

```
void interface::save ( )  [slot]
```

Permet de choisir un fichier de sauvegarde.

Parameters

Definition at line 217 of file interface.cpp.

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

- OL13/interface.h
- OL13/interface.cpp

5.13 RelationManager::Iterator Class Reference

Public Member Functions

- void **next** ()
- bool **isDone** () const
- Relation & **current** () const
- Relation ∗ **listeRel** ()

Friends

- class **RelationManager**

5.13.1 Detailed Description

Definition at line 191 of file manager.h.

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

- OL13/manager.h

5.14 Relation::Iterator Class Reference

Public Member Functions

- void **next** ()
- bool **isDone** () const
- NotesCouple & **current** () const

Friends

- class **Relation**

5.14.1 Detailed Description

Definition at line 87 of file relations.h.

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

- OL13/relations.h

5.15 NotesManager::Iterator Class Reference

Public Member Functions

- void **next** ()
- bool **isDone** () const
- Note & **current** () const
- QList< Note * > * **liste** ()
- QList< Note * >::iterator **getIteratorVersions** ()

Friends

- class **NotesManager**

5.15.1 Detailed Description

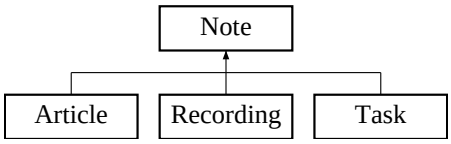
Definition at line 124 of file manager.h.

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

- OL13/manager.h

5.16 Note Class Reference

Inheritance diagram for Note:



Public Member Functions

- const QString **getId** () const
- const QString **getTitle** () const
- const QDateTime **getCreation_date** () const
- const QDateTime **getLastmodif_date** () const
- bool **getIsArchive** () const
- bool **getIsDeleted** () const
- void **setIsArchive** (bool a)
- void **setIsDeleted** (bool d)
- QString **getType** () const
- Note (const QString &i, const QString &ti)

  Constructeur des classes notes, articles task et recording.
- **Note** (const QString &i, const QString &ti, const QDateTime &cd, const QDateTime &lmd, bool iA, bool iD)
- void **setTitle** (const QString &t)

- void **setCreation_date** (const QDateTime &d)
- void **setLastmodif_date** (const QDateTime &d)
- Note (const Note &n)

  Constructeur de recopie.
- Note & operator= (const Note &n)

  Surcharge de l'opérateur = dans le cas nouvelle note B=A.
- virtual ∼Note ()

  Destructeur de la classe Note.
- virtual std::string toString () const

  Transforme une note en un flux ostream a afficher.
- void **display** (std::ostream &f=std::cout) const
- virtual void **saveNote** (QFile ∗file)
- void **setNbIsRef** (unsigned int n)
- unsigned int **getNbIsRef** () const
- void deleteReference (const QString &id)

  Supprime la référence sur une note spécifiée par son ID.
- void deleteAllReference ()

  Supprime l'ensemble des références d'une note.
- void setNewRef (const QString &id)

  Définie une note comme référence d'une autre.
- Note & getReference (const QString &id) const

  Retourne une note référencée par une autre.
- void generateRef (const QString &champTexte)

  Détecte la présence d'une expression régulière dans un champ de text et ajoute une référence en conséquence.

Public Attributes

- QList< QString > **references**

5.16.1 Detailed Description

Definition at line 74 of file notes.h.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Note() [1/2]

```
Note::Note (
            const QString & i,
            const QString & ti )
```

Constructeur des classes notes, articles task et recording.

Les classes dérivées Article, Task, Recording utilise en premier lieu le constructeur de Note. Dans le constructeur de Note, la date de création et de dernière modification sont mises à jours avec la date courrante.

Definition at line 106 of file notes.cpp.

5.16.2.2 Note() [2/2]

```
Note::Note (
            const Note & n )
```

Constructeur de recopie.

Recopie l'ensemble des informations d'une note Mets à jour la date de dernière modification avec currentDateTime.

Parameters

const	Note& n La note a recopier.

Definition at line 35 of file notes.cpp.

5.16.3 Member Function Documentation

5.16.3.1 deleteAllReference()

```
void Note::deleteAllReference ( )
```

Supprime l'ensemble des références d'une note.

À chaque suppression, les notes anciennement référencées par cette note diminuent le nombre de notes qui les référence.

Definition at line 399 of file notes.cpp.

5.16.3.2 deleteReference()

```
void Note::deleteReference (
            const QString & id )
```

Supprime la référence sur une note spécifiée par son ID.

Parameters

const	QString& id L'ID de la note référencée a supprimer.

Definition at line 374 of file notes.cpp.

5.16.3.3 generateRef()

```
void Note::generateRef (
            const QString & champTexte )
```

Détecte la présence d'une expression régulière dans un champ de text et ajoute une référence en conséquence.

Cette méthode est appelée par le constructeur de notes et par les setters de champ de texte de notes. Si l'expression régulière \:ref{(.+)} est repérée, et si l'ID retourné correspond à celui d'une note active, la note correspondant à l'ID est ajoutée en référence de la note this.

Parameters

const	QString& champTexte champ de texte dans laquelle la regex doit être repérée

Definition at line 418 of file notes.cpp.

5.16.3.4 getReference()

```
Note & Note::getReference (
            const QString & id ) const
```

Retourne une note référencée par une autre.

Parameters

const	QString& id L'ID de la note référencée.

Definition at line 358 of file notes.cpp.

5.16.3.5 setNewRef()

```
void Note::setNewRef (
            const QString & id )
```

Définie une note comme référence d'une autre.

Lorsqu'une note prend comme référence une autre, cette autre note augmente son attribut nbIsRef lui permettant de connaître le nombre de notes qui la référencent.

Parameters

const	QString& L'ID de la note référencée.

Definition at line 344 of file notes.cpp.

5.16.3.6 toString()

```
std::string Note::toString ( ) const  [virtual]
```

Transforme une note en un flux ostream a afficher.

Returns

Le flux ostream

Reimplemented in Recording, Task, and Article.

Definition at line 182 of file notes.cpp.

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

- OL13/notes.h
- OL13/notes.cpp

5.17 NotesCouple Class Reference

Public Member Functions

- **NotesCouple** (Note *nx, Note *ny, QString l=0, bool s=false)
- Note * **getCoupleNoteX** () const
- Note * **getCoupleNoteY** () const
- QString **getLabel** () const
- void **setLabel** (QString l)
- bool **getSymetric** () const

5.17.1 Detailed Description

Definition at line 46 of file relations.h.

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

- OL13/relations.h

5.18 NotesException Class Reference

Public Member Functions

- **NotesException** (const QString &message)
- QString **getinfo** () const

5.18.1 Detailed Description

Definition at line 63 of file notes.h.

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

- OL13/notes.h

5.19 NotesManager Class Reference

Classes

- class Iterator

Public Member Functions

- Article & getNewArticle (const QString &id, const QString &ti, const QString &te)

  Créer une nouvelle note de type article La première version de cette note est ajoutée à la liste des versions d'une note.
- Task & **getNewTask** (const QString &id, const QString &ti, const QString &a, ENUM::StatusType s, unsigned int p, const QDateTime d)
- Task & **getNewTask** (const QString &id, const QString &ti, const QString &a, ENUM::StatusType s, unsigned int p)
- Task & **getNewTask** (const QString &id, const QString &ti, const QString &a, ENUM::StatusType s, const QDateTime d)
- Task & getNewTask (const QString &id, const QString &ti, const QString &a, ENUM::StatusType s)

  Créer une nouvelle note de type task La première version de cette note est ajoutée à la liste des versions d'une note.
- Recording & getNewRecording (const QString &id, const QString &ti, const QString &d, ENUM::Recording↩ Type r, QString l)

  Créé une nouvelle note de type recording La première version de cette note est ajoutée à la liste des versions d'une note.
- Article & editArticle (Article &A)

  Créé une nouvelle instance d'un article passé en paramètre.
- Task & editTask (Task &T)

  Créé une nouvelle instance d'une task passée en paramètre.
- Recording & editRecording (Recording &R)

  Créé une nouvelle instance d'un recording passé en paramètre.
- Note & getNote (const QString &id)

  Retourne une réfèrence sur l'ID d'une note spécifiée en paramètre.
- Note & getNoteVersion (const QString &id, int indice)

  Retourne la i-ième versions d'une note spécifiée en paramètre.
- QList< Note * > * getListeVersions (const QString &id)

  Retourne la liste des versions d'une note spécifiée en paramètre.
- void deleteNote (const QString &id)

  Action de suppression d'une note spécifiée en paramètre.
- void setFilename (const QString f)

  Affectation de l'attribut filename du manager à f.
- QString getFilename () const

  Accesseur de l'attribut filename du manager.

- void load ()
  Charge les notes d'un fichier XML dans l'application.
- void save () const
  Sauvegarde des notes dans un fichier XML.
- void **load_fichier** () const
- void **save_fichier** ()
- QString **updateId** (QString Id2) const
- QList< TupleNote_Relation * > getListTupleAscendants (const QString &id)
  Renvoie la liste de tous les couples ascendants de la relation portant l'ID id.
- QList< TupleNote_Relation * > getListTupleDescendants (const QString &id)
  Renvoie la liste de tous les couples descendants de la relation portant l'ID id.
- QList< Note * > getListAscendants (const QString &id)
  Retourne la liste des notes en relation ascendant avec une note specifiée en paramètre.
- QList< Note * > getListDescendants (const QString &id)
  Retourne la liste des notes en relation descendante avec une note specifiée en paramètre.
- QList< Note * > getListArchive ()
  Retourne la liste des notes archivée.
- QList< Note * > getListDeleted ()
  Retourne la liste des notes supprimée.
- void emptyTrash ()
  Supprime définitivement l'ensemble des notes déclarées comme supprimées.
- void restoreNoteTrash (const QString &id)
  Restaure une note supprimée en passant à false son attribut isDeleted.
- int getnbNote ()
  Accesseur du nombre de notes.
- Iterator getIterator ()
  Permet d'accéder à l'itérateur de notes.

Static Public Member Functions

- static NotesManager * getInstance ()
  Permet d'obtenir un pointeur sur le manager de notes.
- static void libererInstance ()
  Permet de libérer le manager de note.

5.19.1 Detailed Description

Definition at line 50 of file manager.h.

5.19.2 Member Function Documentation

5.19.2.1 deleteNote()

```
void NotesManager::deleteNote (
            const QString & id )
```

Action de suppression d'une note spécifiée en paramètre.

Lorsque la suppression d'une note est demandée, celle ci est archivée si elle est référencée par d'autres notes (l'attribut isArchive est mis à true). Sinon, supprime sa présence dans l'ensemble des relations, puis supprime l'ensemble des références qu'à cette note, puis déplace la note dans la corbeille (l'attribut siDeleted est mis à true).

Parameters

const	QString& id ID de la note a supprimer

Definition at line 865 of file manager.cpp.

5.19.2.2 editArticle()

```
Article & NotesManager::editArticle (
            Article & A )
```

Créé une nouvelle instance d'un article passé en paramètre.

Recopie une note passée en paramètre en modifiant sa date de dernière mise à jour et l'insère en première position dans la liste des versions d'une note

Parameters

Article&	A Référence sur l'article a éditer

Definition at line 237 of file manager.cpp.

5.19.2.3 editRecording()

```
Recording & NotesManager::editRecording (
            Recording & R )
```

Créé une nouvelle instance d'un recording passé en paramètre.

Recopie une note passée en paramètre en modifiant sa date de dernière mise à jour et l'insère en première position dans la liste des versions d'une note

Parameters

Recording&	R Référence sur le recording a éditer

Definition at line 269 of file manager.cpp.

5.19.2.4 editTask()

```
Task & NotesManager::editTask (
            Task & T )
```

Créé une nouvelle instance d'une task passée en paramètre.

Recopie une note passée en paramètre en modifiant sa date de dernière mise à jour et l'insère en première position dans la liste des versions d'une note

Parameters

Task&	T Référence sur la tache a éditer

Definition at line 253 of file manager.cpp.

5.19.2.5 emptyTrash()

```
void NotesManager::emptyTrash ( )
```

Supprime définitivement l'ensemble des notes déclarées comme supprimées.

Parcourt l'ensemble des notes, lorsqu'une note est reconnue comme supprimée chaque version de cette note est supprimée puis la liste des versions elle-même est vidée et supprimée

Definition at line 893 of file manager.cpp.

5.19.2.6 getFilename()

```
QString NotesManager::getFilename ( ) const  [inline]
```

Accesseur de l'attribut filename du manager.

Returns

Qstring

Definition at line 95 of file manager.h.

5.19.2.7 getListAscendants()

```
QList< Note * > NotesManager::getListAscendants (
            const QString & id )
```

Retourne la liste des notes en relation ascendant avec une note specifiée en paramètre.

Parameters

const	QString& id ID de la note dont il faut trouver les notes en relation ascendante.

Definition at line 1011 of file manager.cpp.

5.19.2.8 getListDescendants()

```
QList< Note * > NotesManager::getListDescendants (
            const QString & id )
```

Retourne la liste des notes en relation descendante avec une note specifiée en paramètre.

Parameters

const	QString& id ID de la note dont il faut trouver les notes en relation descendante.

Definition at line 1050 of file manager.cpp.

5.19.2.9 getListeVersions()

```
QList< Note * > * NotesManager::getListeVersions (
            const QString & id )
```

Retourne la liste des versions d'une note spécifiée en paramètre.

Parameters

const	QString& id ID de la note dont les versions doivent être retournée

Definition at line 314 of file manager.cpp.

5.19.2.10 getListTupleAscendants()

```
QList< TupleNote_Relation * > NotesManager::getListTupleAscendants (
            const QString & id )
```

Renvoie la liste de tous les couples ascendants de la relation portant l'ID id.

Parameters

const	QString& id ID de la relation concernée

Definition at line 1031 of file manager.cpp.

5.19.2.11 getListTupleDescendants()

```
QList< TupleNote_Relation * > NotesManager::getListTupleDescendants (
            const QString & id )
```

Renvoie la liste de tous les couples descendants de la relation portant l'ID id.

Parameters

const	QString& id ID de la relation concernée

Definition at line 1069 of file manager.cpp.

5.19.2.12 getNewArticle()

```
Article & NotesManager::getNewArticle (
            const QString & id,
            const QString & ti,
            const QString & te )
```

Créer une nouvelle note de type article La première version de cette note est ajoutée à la liste des versions d'une note.

Parameters

const	QString& id ID de la note à créer
const	QString& ti Titre de l'article à créer
const	QString& de Text de l'article a créer

Returns

Article&

Definition at line 32 of file manager.cpp.

5.19.2.13 getNewRecording()

```
Recording & NotesManager::getNewRecording (
            const QString & id,
            const QString & ti,
            const QString & d,
            ENUM::RecordingType r,
            QString l )
```

Créé une nouvelle note de type recording La première version de cette note est ajoutée à la liste des versions d'une note.

Parameters

const	QString& id ID de la note a créer
const	QString& ti Titre du recording a créer
const	QString& d Description du recording a créer
ENUM::RecordingType	r Type du recording a créer
QString	l Link du recording a créer

Definition at line 205 of file manager.cpp.

5.19.2.14 getNewTask()

```
Task & NotesManager::getNewTask (
            const QString & id,
            const QString & ti,
            const QString & a,
            ENUM::StatusType s )
```

Créer une nouvelle note de type task La première version de cette note est ajoutée à la liste des versions d'une note.

Parameters

const	QString& id ID de la note à créer.
const	QString& ti Titre de la tache à créer.
const	QString& a Action de la tache à créer.
ENUM::StatusType	s Status de la tache à créer.
unsigned	int p Priority de la tache à créer.
const	QDateTime d Date limite de la tache à créer.
const	QString& id ID de la note a créer
const	QString& ti Titre de la tache a créer
const	QString& a Action de la tache a créer
ENUM::StatusType	s Status de la tache a créer
unsigned	int p Priority de la tache à créer
const	QString& id ID de la note a créer
const	QString& ti Titre de la tache a créer
const	QString& a Action de la tache a créer
ENUM::StatusType	s Status de la tache a créer
const	QDateTime d Date limite de la tache à créer
const	QString& id ID de la note a créer
const	QString& ti Titre de la tache a créer
const	QString& a Action de la tache a créer
ENUM::StatusType	s Status de la tache a créer

Definition at line 170 of file manager.cpp.

5.19.2.15 getNote()

```
Note & NotesManager::getNote (
            const QString & id )
```

Retourne une réfèrence sur l'ID d'une note spécifiée en paramètre.

Parameters

const	QString& id ID de la note a retourner

Definition at line 283 of file manager.cpp.

5.19.2.16 getNoteVersion()

```
Note & NotesManager::getNoteVersion (
            const QString & id,
            int indice )
```

Retourne la i-ième versions d'une note spécifiée en paramètre.

Parameters

const	QString& id ID de la note concernée int indice Numéro de la version à retourner

Definition at line 299 of file manager.cpp.

5.19.2.17 load()

```
void NotesManager::load ( )
```

Charge les notes d'un fichier XML dans l'application.

Cette fonction ne charge que les notes une fonction dédiée aux relations est définie dans la classe RelationManager. Le nom du fichier XML est stocké dans le NotesManager.

Definition at line 433 of file manager.cpp.

5.19.2.18 restoreNoteTrash()

```
void NotesManager::restoreNoteTrash (
            const QString & id )
```

Restaure une note supprimée en passant à false son attribut isDeleted.

Parameters

const	QString& id ID de la note supprimée à restaurer

Definition at line 916 of file manager.cpp.

5.19.2.19 save()

```
void NotesManager::save ( ) const
```

Sauvegarde des notes dans un fichier XML.

Cette fonction sauvegarde les notes ainsi que les relations. Le nom du fichier XML est stocké dans le Notes↩ Manager.

Definition at line 375 of file manager.cpp.

5.19.2.20 setFilename()

```
void NotesManager::setFilename (
            const QString f )  [inline]
```

Affectation de l'attribut filename du manager à f.

Parameters

const	QString f

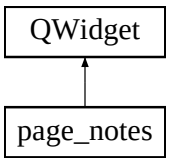
Definition at line 89 of file manager.h.

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

- OL13/manager.h
- OL13/manager.cpp

5.20 page_notes Class Reference

Inheritance diagram for page_notes:



Public Slots

- void **on_savebutton_clicked** ()
- void **editer_note** (bool status)
- void **Archiver_page_note** ()
- void **aff_Relation** (QString titre)

Signals

- void **supp_dock_editer** ()
- void **update_model** ()
- void **supp_dock_aff_rel** ()
- void **close_page** ()

Public Member Functions

- page_notes (Note &N)
- QWidget & **getdock_editer** ()
- QDockRelation * **getdock_aff_rel** ()

Protected Attributes

- QNote * **note**
- Note & **n**
- Note * **newNote**
- QWidget * **dock_editer**
- QDockRelation * **dock_aff_Rel**
- QPushButton * **savebutton**
- Qreference * **widget_ref**
- QLabel * **info**
- QHBoxLayout * **layout_info**
- QVBoxLayout * **layout_p**
- QVBoxLayout * **L_titre**

5.20.1 Detailed Description

Definition at line 26 of file aff_notes.h.

5.20.2 Constructor & Destructor Documentation

5.20.2.1 page_notes()

```
page_notes::page_notes (
            Note & N )
```

Window fenetre principale:

Création des docks Dock editer

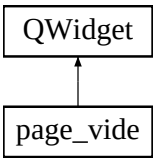
Definition at line 15 of file aff_notes.cpp.

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

- OL13/aff_notes.h
- OL13/aff_notes.cpp

5.21 page_vide Class Reference

Inheritance diagram for page_vide:



5.21.1 Detailed Description

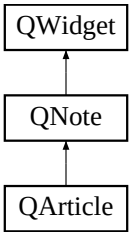
Definition at line 105 of file aff_notes.h.

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

- OL13/aff_notes.h
- OL13/aff_notes.cpp

5.22 QArticle Class Reference

Inheritance diagram for QArticle:



Public Slots

- void **check_creer** ()

Signals

- void **checked_creer** (bool)

Public Member Functions

- void **load_note** (Note &N)
- void **readOnly** (bool status)
- void **saveNote** (Note &N)

Additional Inherited Members

5.22.1 Detailed Description

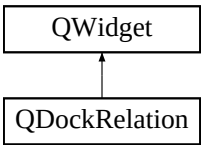
Definition at line 41 of file qnote.h.

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

- OL13/qnote.h
- OL13/qnote.cpp

5.23 QDockRelation Class Reference

Inheritance diagram for QDockRelation:



Public Slots

- void **emit_From_selection** (QModelIndex i)
- void **emit_to_selection** (QModelIndex i)
- void **updateModels** ()

Signals

- void **selectionNote** (QString, int)
- void **selectionRelation** (QString titre)

Public Member Functions

- **QDockRelation** (const QString &id)

5.23.1 Detailed Description

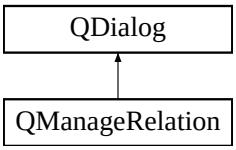
Definition at line 24 of file qrelations.h.

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

- OL13/qrelations.h
- OL13/qrelations.cpp

5.24 QManageRelation Class Reference

Inheritance diagram for QManageRelation:



Public Slots

- void **on_show_clicked** ()
- void **on_RelationView_doubleClicked** (QModelIndex index)
- void **on_remove_clicked** ()

Public Member Functions

- **QManageRelation** (QWidget ∗parent=0)
- QUiRelation ∗ **getSelectedR** ()

5.24.1 Detailed Description

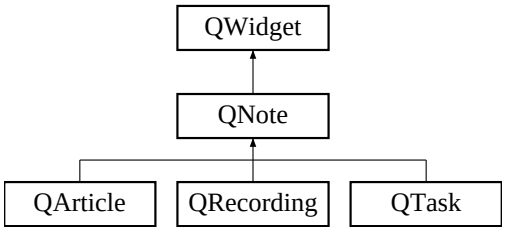
Definition at line 12 of file qmanagerelation.h.

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

- OL13/qmanagerelation.h
- OL13/qmanagerelation.cpp

5.25 QNote Class Reference

Inheritance diagram for QNote:



Public Slots

- virtual void **check_creer** ()=0
- virtual void **saveNote** (Note &n)

Signals

- void **checked_creer** (bool)

Public Member Functions

- virtual Note & **get_note** (QString id, QString title)=0
- virtual void **readOnly** (bool status)
- QNote ()

  Constructueur de la clase QNote.
- virtual void **load_note** (Note &n)
- QVBoxLayout ∗ **getLayout_titre** ()

Protected Attributes

- QGridLayout ∗ **grid**

5.25.1 Detailed Description

Definition at line 21 of file qnote.h.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 QNote()

```
QNote::QNote ( )
```

Constructueur de la clase QNote.

Constructueur de la clase QRecording.

Constructueur de la clase QTask.

Constructueur de la clase QArticle.

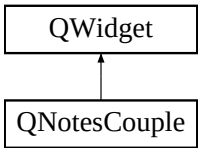
Definition at line 17 of file qnote.cpp.

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

- OL13/qnote.h
- OL13/qnote.cpp

5.26 QNotesCouple Class Reference

Inheritance diagram for QNotesCouple:



Public Member Functions

- **QNotesCouple** (Note *x, Note *y, bool s)

5.26.1 Detailed Description

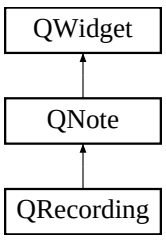
Definition at line 9 of file qrelations.h.

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

- OL13/qrelations.h

5.27 QRecording Class Reference

Inheritance diagram for QRecording:



Public Slots

- QString **OuvrirFichier** ()
- void **check_creer** ()
- void **read_record** ()
- void **stop_record** ()
- void **saveNote** (Note &n)

Signals

- void **checked_creer** (bool)
- void **recording_ready** ()

Public Member Functions

- virtual void **load_note** (Note &N)
- void **readOnly** (bool status)

Additional Inherited Members

5.27.1 Detailed Description

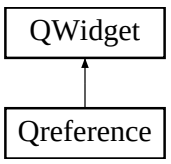
Definition at line 85 of file qnote.h.

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

- OL13/qnote.h
- OL13/qnote.cpp

5.28 Qreference Class Reference

Inheritance diagram for Qreference:



Public Slots

- void **update_model** ()

Public Member Functions

- **Qreference** (Note &n, QWidget *parent=0)

Public Attributes

- QStandardItemModel ∗ **model_ref**

5.28.1 Detailed Description

Definition at line 11 of file qreference.h.

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

- OL13/qreference.h
- OL13/qreference.cpp

5.29 Qrelations Class Reference

Public Member Functions

- **Qrelations** (QString t, QString d)
- void **getNewCoupleRelation** (Note ∗n1, Note ∗n2, QString l=0, bool s=false)
- string **displayRelation** ()

5.29.1 Detailed Description

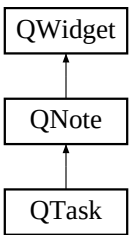
Definition at line 46 of file qrelations.h.

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

- OL13/qrelations.h
- OL13/qrelations.cpp

5.30 QTask Class Reference

Inheritance diagram for QTask:



Public Slots

- void **check_creer** ()

Signals

- void **checked_creer** (bool)

Public Member Functions

- void **load_note** (Note &N)
- void **readOnly** (bool status)
- void **saveNote** (Note &N)

Additional Inherited Members

5.30.1 Detailed Description

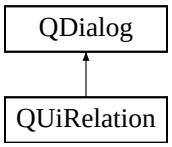
Definition at line 58 of file qnote.h.

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

- OL13/qnote.h
- OL13/qnote.cpp

5.31 QUiRelation Class Reference

Inheritance diagram for QUiRelation:



Public Slots

- void **setCouple** (QString l)
- void **on_show_clicked** ()
- void **on_ETitre_textChanged** (QString t)
- void **on_EDesciption_textChanged** ()
- void **on_RelationView_doubleClicked** (QModelIndex index)
- void **on_remove_clicked** ()
- void **on_addCouple_clicked** ()
- void **addNewCouple** (QString id1, QString id2, QString l, bool s)

Signals

- void **newCouple** ()

Public Member Functions

- **QUiRelation** (Relation &r, QWidget ∗parent=0)

5.31.1 Detailed Description

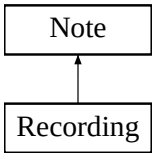
Definition at line 14 of file quirelation.h.

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

- OL13/quirelation.h
- OL13/quirelation.cpp

5.32 Recording Class Reference

Inheritance diagram for Recording:



Public Member Functions

- const QTextDocument & **getDescription** () const
- ENUM::RecordingType **getType** () const
- const QString **getLink** () const
- void **setDescription** (const QString &d)
- void **setType** (const ENUM::RecordingType &r)
- void **setLink** (const QString &l)
- **Recording** (const QString i, const QString &ti, const QString d, ENUM::RecordingType r, QString l)
- **Recording** (const QString &i, const QString &ti, const QDateTime &cd, const QDateTime &lmd, bool iA, bool iD, const QString &d, ENUM::RecordingType ty, const QString &li)
- **Recording** (const Recording &r)
- Recording & operator= (const Recording &r)

  Surcharge de l'opérateur = dans le cas nouveau recording B=A.
- std::string toString () const

  Transforme un recording en un flux ostream a afficher.
- void **saveNote** (QFile ∗file)

Additional Inherited Members

5.32.1 Detailed Description

Definition at line 185 of file notes.h.

5.32.2 Member Function Documentation

5.32.2.1 toString()

```
std::string Recording::toString ( ) const  [virtual]
```

Transforme un recording en un flux ostream a afficher.

Returns

Le flux ostream

Reimplemented from Note.

Definition at line 264 of file notes.cpp.

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

- OL13/notes.h
- OL13/notes.cpp

5.33 Relation Class Reference

Classes

- class Iterator

Public Member Functions

- **Relation** (QString t, QString d)
- const QString **getTitle** () const
- void **setTitle** (QString t)
- void **setDescription** (QString d)
- NotesCouple & getNewCoupleRelation (Note *n1, Note *n2, QString label=0, bool s=false)

  Créer un couple de note dans une relation.
- const QString **getDescription** () const
- NotesCouple * getCoupleRelation (Note *n1, Note *n2) const

  Retourne un pointeur sur un couple d'une référence dont les deux notes sont passées en paramètre.
- void removeCoupleRelation (Note *n1, Note *n2)

  Supprime un couple de note d'une relation.
- void removeNoteRelation (Note *n1)

  Supprime une note d'une relation.
- std::string displayRelation ()

  Affiche une relation et l'ensemble des couples de notes qui en font parti.
- Iterator **getIterator** ()
- void displayCoupleRelation (Note *n1, Note *n2) const

  Affiche un couple dans une relation.

5.33.1 Detailed Description

Definition at line 63 of file relations.h.

5.33.2 Member Function Documentation

5.33.2.1 displayCoupleRelation()

```
void Relation::displayCoupleRelation (
            Note * n1,
            Note * n2 ) const
```

Affiche un couple dans une relation.

Parameters

Note∗	n1 Note X du couple de la relation a afficher
Note∗	n2 Note Y du couple de la relation a afficher

Definition at line 127 of file relations.cpp.

5.33.2.2 getCoupleRelation()

```
NotesCouple * Relation::getCoupleRelation (
            Note * n1,
            Note * n2 ) const
```

Retourne un pointeur sur un couple d'une référence dont les deux notes sont passées en paramètre.

Parameters

Note∗	n1 Note X du couple de la relation a retourner
Note∗	n2 Note Y du couple de la relation a retourner

Definition at line 110 of file relations.cpp.

5.33.2.3 getNewCoupleRelation()

```
NotesCouple & Relation::getNewCoupleRelation (
            Note * n1,
```

```
            Note * n2,
            QString label = 0,
            bool s = false )
```

Créer un couple de note dans une relation.

Parameters

Note∗	n1 Note X du couple dans la relation Note∗ n2 Note Y du couple dans la relation QString label Label du couple dans la relation bool s Symétrie du couple dans la relation

Definition at line 66 of file relations.cpp.

5.33.2.4 removeCoupleRelation()

```
void Relation::removeCoupleRelation (
            Note * n1,
            Note * n2 )
```

Supprime un couple de note d'une relation.

Si le couple demandé est symétrique, le deux couples créés par ces notes sont supprimés.

Parameters

Note∗	n1 Note X du couple a supprimer de la relation
Note∗	n2 Note Y du couple a supprimer de la relation

Definition at line 154 of file relations.cpp.

5.33.2.5 removeNoteRelation()

```
void Relation::removeNoteRelation (
            Note * n1 )
```

Supprime une note d'une relation.

Si une note passée en paramètre appartient à un couple présent au sein de la relation this, le couple est supprimée de la relation.

Parameters

Note∗	n1 Note a supprimer de la relation

Definition at line 182 of file relations.cpp.

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

- OL13/relations.h
- OL13/relations.cpp

5.34 RelationManager Class Reference

Classes

- class Iterator

Public Member Functions

- Relation & getNewRelation (const QString &title, const QString &desc)

  Créé une nouvelle relation à partir d'un titre et d'une description si une autre relation du même titre n'existe pas déjà
- Relation & getRelation (const QString &title)

  Retourne une référence sur une relation dont le titre est donné en paramètre.
- void deleteRelation (const QString &title)

  Supprime une relation dont le titre est passé en paramètre.
- unsigned int getNbRelations () const

  Accesseur du nombre de relations.
- void load (const QString &file)

  Charge les relations d'un fichier XML dans l'application.
- Iterator **getIterator** ()

Static Public Member Functions

- static RelationManager & getInstance ()

  Permet d'obtenir un pointeur sur le manager de relations.
- static void libererInstance ()

  Permet de libérer le manager de relation.

5.34.1 Detailed Description

Definition at line 157 of file manager.h.

5.34.2 Member Function Documentation

5.34.2.1 deleteRelation()

```
void RelationManager::deleteRelation (
            const QString & title )
```

Supprime une relation dont le titre est passé en paramètre.

Parameters

const	QString& title Titre de la relation a supprimer

Definition at line 995 of file manager.cpp.

5.34.2.2 getNewRelation()

```
Relation & RelationManager::getNewRelation (
            const QString & title,
            const QString & desc )
```

Créé une nouvelle relation à partir d'un titre et d'une description si une autre relation du même titre n'existe pas déjà

Parameters

const	QString& title Titre de la nouvelle relation à créer const QString& desc Description de la nouvelle relation à créer

Definition at line 960 of file manager.cpp.

5.34.2.3 getRelation()

```
Relation & RelationManager::getRelation (
            const QString & title )
```

Retourne une référence sur une relation dont le titre est donné en paramètre.

Parameters

const	QString& title Titre de la relation a retourner

Definition at line 983 of file manager.cpp.

5.34.2.4 load()

```
void RelationManager::load (
            const QString & file )
```

Charge les relations d'un fichier XML dans l'application.

Cette fonction ne charge que les relations une fonction dédiée aux notes est définie dans la classe NotesManager.

Parameters

const	QString& file Nom du fichier XML à parser

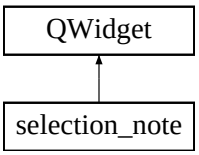
Definition at line 1116 of file manager.cpp.

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

- OL13/manager.h
- OL13/manager.cpp

5.35 selection_note Class Reference

Inheritance diagram for selection_note:



Public Slots

- void emit_selection (QModelIndex i)

  Identifie la note/versions que l'utilisateur souhaité affiché
- void update_model ()

  Etablie/Met à jours le model contennant tous les Notes actifs, en appellant Le Note Manager.

Signals

- void **selection** (QString, int)

Public Member Functions

- selection_note ()

  création de la widget de selection de note
- QStandardItemModel ∗ **getModel** () const
- QTreeView ∗ **getVue** ()

5.35.1 Detailed Description

Definition at line 34 of file interface.h.

5.35.2 Member Function Documentation

5.35.2.1 emit_selection

```
void selection_note::emit_selection (
            QModelIndex i )  [slot]
```

Identifie la note/versions que l'utilisateur souhaité affiché

Parameters

i	

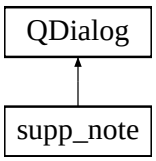
Definition at line 325 of file interface.cpp.

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

- OL13/interface.h
- OL13/interface.cpp

5.36 supp_note Class Reference

Inheritance diagram for supp_note:



Public Slots

- void **supp_selection_note** ()

Signals

- void **close_note** ()

Public Member Functions

- **supp_note** (QStandardItemModel ∗m, QWidget ∗parent)

5.36.1 Detailed Description

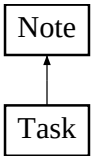
Definition at line 18 of file supp_note.h.

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

- OL13/supp_note.h
- OL13/supp_note.cpp

5.37 Task Class Reference

Inheritance diagram for Task:



Public Member Functions

- const QString **getAction** () const
- void **setAction** (const QString &a)
- ENUM::StatusType **getStatus** () const
- void **setStatus** (const ENUM::StatusType &s)
- int **getPriority** () const
- void **setPriority** (unsigned int p)
- const QDateTime & **getDueDate** () const
- void **setDueDate** (const QDateTime d)
- **Task** (const QString &i, const QString &ti, const QString &a, ENUM::StatusType s)
- **Task** (const QString &i, const QString &ti, const QString &a, ENUM::StatusType s, unsigned int p)
- **Task** (const QString &i, const QString &ti, const QString &a, ENUM::StatusType s, const QDateTime d)
- Task (const QString &i, const QString &ti, const QString &a, ENUM::StatusType s, unsigned int p, const QDateTime d)

  Constructeur de la classe task.
- **Task** (const QString &i, const QString &ti, const QDateTime &cd, const QDateTime &lmd, bool iA, bool iD, const QString a, ENUM::StatusType s, unsigned int p, const QDateTime dD)
- Task (const Task &t)

  Surcharge la méthode constructeur dans le cas nouvelle task B(A)
- Task & operator= (const Task &a)

  Surcharge de l'opérateur = dans le cas nouvelle task B=A.
- std::string toString () const

  Transforme une task en un flux ostream a afficher.
- void **saveNote** (QFile ∗file)

Additional Inherited Members

5.37.1 Detailed Description

Definition at line 150 of file notes.h.

5.37.2 Constructor & Destructor Documentation

5.37.2.1 Task()

```
Task::Task (
            const QString & i,
            const QString & ti,
            const QString & a,
            ENUM::StatusType s,
            unsigned int p,
            const QDateTime d )
```

Constructeur de la classe task.

La classe dérivé Task utilise en premier lieu le constructeur de Note. Task possède 4 constructeurs différents car deux de ses attributs sont optionnels.

Definition at line 149 of file notes.cpp.

5.37.3 Member Function Documentation

5.37.3.1 toString()

```
std::string Task::toString ( ) const  [virtual]
```

Transforme une task en un flux ostream a afficher.

Returns

Le flux ostream

Reimplemented from Note.

Definition at line 218 of file notes.cpp.

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

- OL13/notes.h
- OL13/notes.cpp

5.38 TupleNote_Relation Class Reference

Public Member Functions

- Note & **getNote** ()
- Relation & **getRelation** ()
- **TupleNote_Relation** (Note &N, Relation &R)

5.38.1 Detailed Description

Definition at line 39 of file manager.h.

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

- OL13/manager.h

Chapter 6

File Documentation

6.1 OL13/aff_notes.cpp File Reference

//Bref

```
#include "aff_notes.h"
```

6.1.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.2 OL13/aff_notes.h File Reference

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

```
#include "QInclude.h"
#include "notes.h"
#include "qnote.h"
#include "manager.h"
#include "qrelations.h"
#include "quirelation.h"
#include "qreference.h"
```

Classes

- class page_notes
- class page_vide

6.2.1 Detailed Description

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Expliquer en détail. //EX :Cette classe surcharge les accesseurs standards du module_voiture pour convenir aux spécificités des différents modèles possibles.

6.3 OL13/Creation_Note.cpp File Reference

//Bref

```
#include "Creation_Note.h"
```

6.3.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.4 OL13/Creation_Note.h File Reference

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

```
#include "qnote.h"
```

Classes

- class Creation_Note

6.4.1 Detailed Description

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Expliquer en détail. //EX :Cette classe surcharge les accesseurs standards du module_voiture pour convenir aux spécificités des différents modèles possibles.

6.5 OL13/include.h File Reference

Regroupe une partie des includes nécessaires au projet.

```
#include <iostream>
#include <iomanip>
#include <stdio.h>
#include <string>
#include <string.h>
#include <ctime>
#include <ratio>
#include <chrono>
#include <QDateTime>
#include <sstream>
#include "notes.h"
#include "Creation_Note.h"
#include "interface.h"
```

6.5.1 Detailed Description

Regroupe une partie des includes nécessaires au projet.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

6.6 OL13/interface.cpp File Reference

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

```
#include "interface.h"
```

6.6.1 Detailed Description

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Expliquer en détail. //EX :Cette classe surcharge les accesseurs standards du module_voiture pour convenir aux spécificités des différents modèles possibles.

6.7 OL13/interface.h File Reference

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

```
#include "Creation_Note.h"
#include <QDockWidget>
#include "manager.h"
#include "aff_notes.h"
#include "qrelations.h"
#include "sstream"
#include <QList>
#include <QDate>
#include <QStandardItemModel>
#include <typeinfo>
#include "supp_note.h"
#include "quirelation.h"
#include "qmanagerelation.h"
#include "dockarchived.h"
```

Classes

- class selection_note
- class interface

6.7.1 Detailed Description

//Expliquer brievement à quoi sert ce fichier. //EX : Définit les modèles de voiture et leur particularités.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Expliquer en détail. //EX :Cette classe surcharge les accesseurs standards du module_voiture pour convenir aux spécificités des différents modèles possibles.

6.8 OL13/main.cpp File Reference

//Bref

```
#include "QInclude.h"
#include "notes.h"
#include "relations.h"
#include "interface.h"
#include <list>
```

Functions

- void **displayAllNote** ()
- void **displayAllVersion** ()
- void **displayAllRelation** ()
- int **PROGRAMME** (int argc, char ∗argv[ ])
- int **main** (int argc, char ∗argv[ ])

6.8.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.9 OL13/manager.cpp File Reference

Définitions des fonctions déclarées dans le manager.h.

```
#include "manager.h"
#include <fstream>
#include <QMessageBox>
#include "QInclude.h"
```

Functions

- ostream & operator<< (ostream &f, const Note &n)

  Surcharge de l'opérateur << d'affichage d'une note.

6.9.1 Detailed Description

Définitions des fonctions déclarées dans le manager.h.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

Domaines des méthodes comprises dans ce fichier :

- Méthodes relatives au NotesManager
- Méthodes relatives au RelationManager Le détail est donné dans la suite du fichier.

6.10 OL13/manager.h File Reference

Regroupe les manager nécessaires pour la gestion des notes et des relations.

```
#include <string>
#include <iostream>
#include "notes.h"
#include "relations.h"
#include "QInclude.h"
```

Classes

- class TupleNote_Relation
- class NotesManager
- class NotesManager::Iterator
- class RelationManager
- class RelationManager::Iterator

Functions

- ostream & operator<< (ostream &f, const Note &n)

  Surcharge de l'opérateur << d'affichage d'une note.

6.10.1 Detailed Description

Regroupe les manager nécessaires pour la gestion des notes et des relations.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

Classes présentes :

- NotesManager Un manager sur l'ensemble des notes, qu'elles soient articles, taches, enregistrement. Qu'elles soient archivées ou supprimées. Manager multi-fonction qui permet de traiter l'ensemble des notes depuis d'un seul élément.

  Possède un attribut QList<Note*>** notes qui regroupe l'ensemble des notes et de leurs versions.
- RelationManager Un manager sur l'ensemble des relations. Possède un Handler pour assurer l'unité de l'accès aux données. Possède un Iterator pour faciliter la manipulation des relations.

  Possède un attribut Relation** tabrelations permettant d'accéder à chaque relation et à l'ensemble de ses couples de notes associé.

L'ensemble des méthodes définies dans ce fichier sont explicitées dans le fichier .cpp associé.

6.11 OL13/notes.cpp File Reference

Définitions des fonctions déclarées dans le notes.h.

```
#include "notes.h"
#include "sstream"
#include "manager.h"
#include "QInclude.h"
#include "QString"
```

Functions

- std::ostream & operator<< (std::ostream &f, const Article &a)

  Surcharge opérateur <<.
- std::ostream & **operator**<< (std::ostream &f, const Task &t)
- std::ostream & **operator**<< (std::ostream &f, const Recording &r)
- QString getStatustoStr (ENUM::StatusType status)

  Retourner directement le statut de int à string.
- QString getRecordingtoStr (ENUM::StatusType recording)

  Retourner directement le recording de int à string.

6.11.1 Detailed Description

Définitions des fonctions déclarées dans le notes.h.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

Domaines des méthodes comprises dans ce fichier :

- Opérateur d'affectation et constructeur de recopie
- Constructeur
- Destructeur
- Affichage
- Référencement de notes Le détail est donné dans la suite du fichier.

6.11.2 Function Documentation

6.11.2.1 getStatustoStr()

```
QString getStatustoStr (
            ENUM::StatusType status )
```

Retourner directement le statut de int à string.

Retourne directement le statut de int à stringQ.

Definition at line 319 of file notes.cpp.

6.11.2.2 operator<<()

```
std::ostream & operator<< (
            std::ostream & f,
            const Article & a )
```

Surcharge opérateur <<.

Surcharge affichage avec polymorphisme.

Version personnalisée pour le type article, task et recording.

Definition at line 300 of file notes.cpp.

6.12 OL13/QInclude.h File Reference

//Bref

```
#include <QApplication>
#include <QTextCodec>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QGroupBox>
#include <QPushButton>
#include <QLabel>
#include <QLineEdit>
#include <QTextEdit>
#include <QCheckBox>
#include <QWidget>
#include <QGridLayout>
#include <QMessageBox>
#include <QComboBox>
#include <QFileDialog>
#include <QSpinBox>
#include <QDateEdit>
#include <QMainWindow>
#include <QAction>
#include <QMenuBar>
#include <QToolBar>
#include <QDirModel>
#include <QTreeView>
#include <QString>
#include <QFormLayout>
#include "QTextDocument"
#include <QtXml>
#include <QFile>
#include <typeinfo>
#include <QList>
#include <QRadioButton>
#include <QStandardItemModel>
#include <QHeaderView>
#include <QTableView>
#include <QListView>
#include <QColumnView>
```

6.12.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.13 OL13/qnote.cpp File Reference

//Bref

```
#include "qnote.h"
```

6.13.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.14 OL13/qnote.h File Reference

//Bref

```
#include "QInclude.h"
#include "notes.h"
#include "manager.h"
#include "relations.h"
#include <QMediaPlayer>
#include <QMediaPlaylist>
#include <QVideoWidget>
```

Classes

- class QNote
- class QArticle
- class QTask
- class QRecording

6.14.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.15 OL13/relations.cpp File Reference

Définitions des fonctions déclarées dans le relations.h.

```
#include "relations.h"
#include "sstream"
```

6.15.1 Detailed Description

Définitions des fonctions déclarées dans le relations.h.

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

Classes présentes :

- NotesCouple La classe NotesCouple est en agrégation avec la classe Relation : une relation contient un ensemble de couple de notes. Dans le cas où l'ensemble des couples d'une relation est supprimé, la relation existe toujours mais est vide. Elle pourra être repeuplée par de nouveaux couples.

  La classe NotesCouple possède deux attributs noteX et noteY pointant chacun sur une note du couple, un attribut pour le label du couple et un attribut booléen spécifiant si le couple est symétrique. C'est à dire que la relation va de noteX vers noteY et réciproquement.

  La classe NotesCouple est composée par la classe Notes : si les notes sont supprimées, les couplent n'existent plus non plus.

- Relation La classe Relation possède un titre, une description et un ensemble de couple de notes. Dans l'implémentation de cette classe, un Design Pattern Iterator est utilisé pour faciliter la manipulation des données.

  Une classe manager dans manager.h/.cpp est utilisée pour gérer l'ensemble des relations.

L'ensemble des méthodes de ces classes sont explicitées dans la suite du fichier.

6.16 OL13/supp_note.cpp File Reference

//Bref

```
#include "supp_note.h"
```

6.16.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.17 OL13/supp_note.h File Reference

//Bref

```
#include "QInclude.h"
#include "manager.h"
```

Classes

- class supp_note

6.17.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.18 OL13/undoredo.cpp File Reference

//Bref

```
#include "undoredo.h"
```

6.18.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

6.19 OL13/undoredo.h File Reference

//Bref

```
#include <QUndoCommand>
```

Classes

- class AppendText

6.19.1 Detailed Description

//Bref

Author

Garnier Maxime, Naudin Louise, Pépin Hugues

Version

1.0

Date

14 Juin 2017

//Détail

Index