

ScolaSync

1.0

Généré par Doxygen 1.7.6.1

Mercredi Décembre 28 2011 08 :56 :29



# Table des matières

<b>1</b>	<b>ScolaSync</b>	<b>1</b>
1.1	But de l'application . . . . .	1
1.2	CAHIER DE CHARGES DE SCOLASYNC . . . . .	1
1.3	Licence . . . . .	2
1.4	Support . . . . .	2
1.5	Architecture de ScolaSync . . . . .	2
<b>2</b>	<b>Hiérarchie de répertoires</b>	<b>3</b>
2.1	Répertoires . . . . .	3
<b>3</b>	<b>Index des espaces de nommage</b>	<b>5</b>
3.1	Paquetages . . . . .	5
<b>4</b>	<b>Index des classes</b>	<b>7</b>
4.1	Hiérarchie des classes . . . . .	7
<b>5</b>	<b>Index des classes</b>	<b>9</b>
5.1	Liste des classes . . . . .	9
<b>6</b>	<b>Index des fichiers</b>	<b>11</b>
6.1	Liste des fichiers . . . . .	11
<b>7</b>	<b>Documentation des répertoires</b>	<b>13</b>
7.1	Répertoire de référence de src/ . . . . .	13
<b>8</b>	<b>Documentation des espaces de nommage</b>	<b>15</b>
8.1	Référence de l'espace de nommage scolasync . . . . .	15
8.1.1	Description détaillée . . . . .	15

8.2	Référence de l'espace de nommage src	15
8.3	Référence de l'espace de nommage src : :chooseInSticks	16
8.3.1	Documentation des variables	16
8.3.1.1	licenceEn	16
8.4	Référence de l'espace de nommage src : :copyToDialog1	16
8.4.1	Documentation des variables	17
8.4.1.1	app	17
8.4.1.2	licenceEn	17
8.4.1.3	windows	17
8.5	Référence de l'espace de nommage src : :db	17
8.5.1	Documentation des fonctions	18
8.5.1.1	checkVersion	18
8.5.1.2	knowsId	18
8.5.1.3	openDb	18
8.5.1.4	readPrefs	19
8.5.1.5	readStudent	19
8.5.1.6	setWd	19
8.5.1.7	tattooList	19
8.5.1.8	writePrefs	19
8.5.1.9	writeStudent	20
8.5.2	Documentation des variables	20
8.5.2.1	cursor	20
8.5.2.2	database	20
8.5.2.3	licence	20
8.6	Référence de l'espace de nommage src : :diskFull	20
8.6.1	Documentation des fonctions	21
8.6.1.1	sceneWithUsage	21
8.6.2	Documentation des variables	21
8.6.2.1	licence	21
8.7	Référence de l'espace de nommage src : :globaldef	21
8.7.1	Documentation des variables	21
8.7.1.1	licenceEn	21
8.7.1.2	logFileName	22
8.7.1.3	markFileName	22

8.7.1.4	userShareDir	22
8.8	Référence de l'espace de nommage src : :help	22
8.8.1	Documentation des variables	22
8.8.1.1	licence	22
8.9	Référence de l'espace de nommage src : :mainWindow	22
8.9.1	Documentation des fonctions	23
8.9.1.1	CheckBoxRect	23
8.9.1.2	firstdir	23
8.9.2	Documentation des variables	23
8.9.2.1	licence	23
8.10	Référence de l'espace de nommage src : :marques	23
8.11	Référence de l'espace de nommage src : :mytextbrowser	23
8.11.1	Documentation des variables	24
8.11.1.1	licence	24
8.12	Référence de l'espace de nommage src : :notification	24
8.12.1	Documentation des variables	24
8.12.1.1	licence	24
8.12.1.2	notif	24
8.13	Référence de l'espace de nommage src : :ownedUsbDisk	25
8.13.1	Documentation des fonctions	25
8.13.1.1	editRecord	25
8.13.2	Documentation des variables	25
8.13.2.1	licence	25
8.14	Référence de l'espace de nommage src : :preferences	25
8.14.1	Documentation des variables	26
8.14.1.1	licence	26
8.15	Référence de l'espace de nommage src : :scolasync	26
8.15.1	Documentation des fonctions	26
8.15.1.1	run	26
8.15.1.2	usage	26
8.15.2	Documentation des variables	26
8.15.2.1	licence	27
8.15.2.2	licenceEn	27
8.15.2.3	licenceFr	27

8.16	Référence de l'espace de nommage <code>src :usbDisk</code>	28
8.16.1	Documentation des variables	28
8.16.1.1	licence	28
8.16.1.2	licence_en	28
8.16.1.3	machin	28
8.17	Référence de l'espace de nommage <code>src :usbThread</code>	29
8.17.1	Documentation des variables	29
8.17.1.1	_threadNumber	29
8.17.1.2	licenceEn	29
8.18	Référence de l'espace de nommage <code>src :version</code>	30
8.18.1	Documentation des fonctions	30
8.18.1.1	major	30
8.18.1.2	minor	30
8.18.1.3	version	30
8.18.2	Documentation des variables	30
8.18.2.1	licence	30
<b>9</b>	<b>Documentation des classes</b>	<b>31</b>
9.1	Référence de la classe <code>src.usbThread.abstractThreadUSB</code>	31
9.1.1	Description détaillée	31
9.1.2	Documentation des constructeurs et destructeur	32
9.1.2.1	__init__	32
9.1.3	Documentation des fonctions membres	32
9.1.3.1	__str__	32
9.1.3.2	threadType	32
9.1.3.3	todo	32
9.1.4	Documentation des données membres	33
9.1.4.1	cmd	33
9.1.4.2	dest	33
9.1.4.3	fileList	33
9.1.4.4	logfile	33
9.1.4.5	subdir	33
9.1.4.6	ud	33
9.2	Référence de la classe <code>src.ownedUsbDisk.Available</code>	34

9.2.1	Description détaillée	34
9.2.2	Documentation des constructeurs et destructeur	34
9.2.2.1	__init__	34
9.2.3	Documentation des données membres	35
9.2.3.1	access	35
9.2.3.2	bus	35
9.2.3.3	checkable	35
9.2.3.4	disks	35
9.2.3.5	enumDev	35
9.2.3.6	firstFats	35
9.3	Référence de la classe src.usbDisk.Available	36
9.3.1	Description détaillée	36
9.3.2	Documentation des constructeurs et destructeur	37
9.3.2.1	__init__	37
9.3.3	Documentation des fonctions membres	37
9.3.3.1	__getitem__	37
9.3.3.2	__len__	37
9.3.3.3	__str__	37
9.3.3.4	__trunc__	38
9.3.3.5	compare	38
9.3.3.6	contains	38
9.3.3.7	getFirstFats	38
9.3.3.8	summary	39
9.3.4	Documentation des données membres	39
9.3.4.1	access	39
9.3.4.2	bus	39
9.3.4.3	checkable	39
9.3.4.4	disks	39
9.3.4.5	enumDev	40
9.3.4.6	firstFats	40
9.4	Référence de la classe src.mainWindow.CheckBoxDelegate	40
9.4.1	Description détaillée	40
9.4.2	Documentation des constructeurs et destructeur	40
9.4.2.1	__init__	40

9.4.3	Documentation des fonctions membres	40
9.4.3.1	editorEvent	41
9.4.3.2	paint	41
9.5	Référence de la classe src.chooseInSticks.chooseDialog	41
9.5.1	Description détaillée	42
9.5.2	Documentation des constructeurs et destructeur	42
9.5.2.1	__init__	42
9.5.3	Documentation des fonctions membres	42
9.5.3.1	activate	42
9.5.3.2	append	42
9.5.3.3	baseDir	43
9.5.3.4	changeWd	43
9.5.3.5	checkWorkDirs	43
9.5.3.6	choose	43
9.5.3.7	choose_dir	44
9.5.3.8	listStorages	44
9.5.3.9	minus	44
9.5.3.10	pathList	44
9.5.3.11	plus	44
9.5.3.12	selectedDiskMountPoint	44
9.5.3.13	selectedDiskOwner	45
9.5.4	Documentation des données membres	45
9.5.4.1	mainWindow	45
9.5.4.2	ownedUsbDictionary	45
9.6	Référence de la classe src.copyToDialog1.copyToDialog1	45
9.6.1	Description détaillée	46
9.6.2	Documentation des fonctions membres	46
9.6.2.1	cancel	46
9.6.2.2	cd	46
9.6.2.3	changeWd	47
9.6.2.4	cont	47
9.6.2.5	displaySize	47
9.6.2.6	remove	47
9.6.2.7	select	47



9.6.2.8	<a href="#">selectedList</a>	48
9.6.2.9	<a href="#">setFromListeDir</a>	48
9.6.2.10	<a href="#">setupFromListe</a>	48
9.6.2.11	<a href="#">setupToListe</a>	48
9.6.3	<a href="#">Documentation des données membres</a>	48
9.6.3.1	<a href="#">mainWindow</a>	48
9.6.3.2	<a href="#">ok</a>	48
9.7	<a href="#">Référence de la classe src.mainWindow.DiskSizeDelegate</a>	49
9.7.1	<a href="#">Description détaillée</a>	49
9.7.2	<a href="#">Documentation des constructeurs et destructeur</a>	49
9.7.2.1	<a href="#">__init__</a>	49
9.7.3	<a href="#">Documentation des fonctions membres</a>	49
9.7.3.1	<a href="#">paint</a>	49
9.7.3.2	<a href="#">val2txt</a>	49
9.8	<a href="#">Référence de la classe src.help.helpWindow</a>	50
9.8.1	<a href="#">Description détaillée</a>	50
9.8.2	<a href="#">Documentation des constructeurs et destructeur</a>	50
9.8.2.1	<a href="#">__init__</a>	50
9.8.3	<a href="#">Documentation des fonctions membres</a>	50
9.8.3.1	<a href="#">loadBrowsers</a>	50
9.8.4	<a href="#">Documentation des données membres</a>	50
9.8.4.1	<a href="#">ui</a>	51
9.9	<a href="#">Référence de la classe src.mainWindow.mainWindow</a>	51
9.9.1	<a href="#">Description détaillée</a>	52
9.9.2	<a href="#">Documentation des constructeurs et destructeur</a>	52
9.9.2.1	<a href="#">__init__</a>	52
9.9.3	<a href="#">Documentation des fonctions membres</a>	52
9.9.3.1	<a href="#">applyPreferences</a>	52
9.9.3.2	<a href="#">changeWd</a>	52
9.9.3.3	<a href="#">checkDisks</a>	53
9.9.3.4	<a href="#">connectTableModel</a>	53
9.9.3.5	<a href="#">copyFrom</a>	53
9.9.3.6	<a href="#">copyTo</a>	53
9.9.3.7	<a href="#">delFiles</a>	53

9.9.3.8	<a href="#">diskFromTableRow</a>	54
9.9.3.9	<a href="#">diskSizeData</a>	54
9.9.3.10	<a href="#">editOwner</a>	54
9.9.3.11	<a href="#">flashLCD</a>	54
9.9.3.12	<a href="#">help</a>	55
9.9.3.13	<a href="#">normalLCD</a>	55
9.9.3.14	<a href="#">preference</a>	55
9.9.3.15	<a href="#">sameDiskData</a>	55
9.9.3.16	<a href="#">showEvent</a>	55
9.9.3.17	<a href="#">tableClicked</a>	55
9.9.3.18	<a href="#">umount</a>	56
9.9.3.19	<a href="#">updateButtons</a>	56
9.9.4	<a href="#">Documentation des données membres</a>	56
9.9.4.1	<a href="#">checkable</a>	56
9.9.4.2	<a href="#">flashTimer</a>	56
9.9.4.3	<a href="#">header</a>	56
9.9.4.4	<a href="#">locale</a>	56
9.9.4.5	<a href="#">oldThreads</a>	57
9.9.4.6	<a href="#">opts</a>	57
9.9.4.7	<a href="#">t</a>	57
9.9.4.8	<a href="#">timer</a>	57
9.9.4.9	<a href="#">tm</a>	57
9.9.4.10	<a href="#">ui</a>	57
9.9.4.11	<a href="#">visibleheader</a>	57
9.9.4.12	<a href="#">workdir</a>	57
9.10	<a href="#">Référence de la classe src.diskFull.mainWindow</a>	57
9.10.1	<a href="#">Description détaillée</a>	58
9.10.2	<a href="#">Documentation des constructeurs et destructeur</a>	58
9.10.2.1	<a href="#">__init__</a>	58
9.10.3	<a href="#">Documentation des données membres</a>	58
9.10.3.1	<a href="#">total</a>	58
9.10.3.2	<a href="#">ui</a>	58
9.10.3.3	<a href="#">used</a>	58
9.10.3.4	<a href="#">v</a>	59

9.11	Référence de la classe <code>src.mytextbrowser.myTextBrowser</code>	59
9.11.1	Description détaillée	59
9.11.2	Documentation des fonctions membres	59
9.11.2.1	<code>setHtml</code>	59
9.11.2.2	<code>setSource</code>	59
9.12	Référence de la classe <code>src.notification.Notification</code>	60
9.12.1	Description détaillée	60
9.12.2	Documentation des constructeurs et destructeur	60
9.12.2.1	<code>__init__</code>	60
9.12.3	Documentation des fonctions membres	61
9.12.3.1	<code>notify</code>	61
9.12.4	Documentation des données membres	61
9.12.4.1	<code>actions</code>	61
9.12.4.2	<code>app_icon</code>	61
9.12.4.3	<code>app_name</code>	61
9.12.4.4	<code>body</code>	61
9.12.4.5	<code>expire_timeout</code>	61
9.12.4.6	<code>hints</code>	61
9.12.4.7	<code>interface</code>	62
9.12.4.8	<code>replaces_id</code>	62
9.12.4.9	<code>summary</code>	62
9.13	Référence de la classe <code>src.preferences.preferenceWindow</code>	62
9.13.1	Description détaillée	62
9.13.2	Documentation des constructeurs et destructeur	62
9.13.2.1	<code>__init__</code>	62
9.13.3	Documentation des fonctions membres	63
9.13.3.1	<code>setValues</code>	63
9.13.3.2	<code>values</code>	63
9.13.4	Documentation des données membres	63
9.13.4.1	<code>ui</code>	63
9.14	Référence de la classe <code>QAbstractTableModel</code>	63
9.15	Référence de la classe <code>QDialog</code>	63
9.16	Référence de la classe <code>QMainWindow</code>	63
9.17	Référence de la classe <code>QObject</code>	64

9.18	Référence de la classe <code>QStyledItemDelegate</code>	64
9.19	Référence de la classe <code>QTextBrowser</code>	64
9.20	Référence de la classe <code>Thread</code>	64
9.21	Référence de la classe <code>src.usbThread.threadCopyFromUSB</code>	64
9.21.1	Description détaillée	65
9.21.2	Documentation des constructeurs et destructeur	65
9.21.2.1	<code>__init__</code>	65
9.21.3	Documentation des fonctions membres	65
9.21.3.1	<code>toDo</code>	65
9.21.4	Documentation des données membres	66
9.21.4.1	<code>cmd</code>	66
9.21.4.2	<code>rootPath</code>	66
9.22	Référence de la classe <code>src.usbThread.threadCopyToUSB</code>	66
9.22.1	Description détaillée	66
9.22.2	Documentation des constructeurs et destructeur	66
9.22.2.1	<code>__init__</code>	67
9.22.3	Documentation des fonctions membres	67
9.22.3.1	<code>threadType</code>	67
9.22.3.2	<code>toDo</code>	67
9.22.4	Documentation des données membres	67
9.22.4.1	<code>cmd</code>	67
9.23	Référence de la classe <code>src.usbThread.threadDeleteInUSB</code>	68
9.23.1	Description détaillée	68
9.23.2	Documentation des constructeurs et destructeur	68
9.23.2.1	<code>__init__</code>	68
9.23.3	Documentation des fonctions membres	68
9.23.3.1	<code>toDo</code>	69
9.23.4	Documentation des données membres	69
9.23.4.1	<code>cmd</code>	69
9.24	Référence de la classe <code>src.usbThread.ThreadRegister</code>	69
9.24.1	Description détaillée	70
9.24.2	Documentation des constructeurs et destructeur	70
9.24.2.1	<code>__init__</code>	70
9.24.3	Documentation des fonctions membres	70

9.24.3.1	<code>__str__</code>	70
9.24.3.2	<code>busy</code>	70
9.24.3.3	<code>pop</code>	70
9.24.3.4	<code>push</code>	70
9.24.3.5	<code>threadSet</code>	71
9.24.4	Documentation des données membres	71
9.24.4.1	<code>dico</code>	71
9.25	Référence de la classe <code>src.usbDisk.uDisk</code>	71
9.25.1	Description détaillée	72
9.25.2	Documentation des constructeurs et destructeur	72
9.25.2.1	<code>__init__</code>	73
9.25.3	Documentation des fonctions membres	73
9.25.3.1	<code>__getitem__</code>	73
9.25.3.2	<code>__str__</code>	73
9.25.3.3	<code>devicePropProxy</code>	73
9.25.3.4	<code>ensureMounted</code>	74
9.25.3.5	<code>file</code>	74
9.25.3.6	<code>getFatUuid</code>	74
9.25.3.7	<code>getFirstFat</code>	74
9.25.3.8	<code>getProp</code>	75
9.25.3.9	<code>headers</code>	75
9.25.3.10	<code>isDosFat</code>	75
9.25.3.11	<code>isTrue</code>	76
9.25.3.12	<code>isUsbDisk</code>	76
9.25.3.13	<code>master</code>	76
9.25.3.14	<code>mountPoint</code>	77
9.25.3.15	<code>showableProp</code>	77
9.25.3.16	<code>title</code>	77
9.25.3.17	<code>uniqueId</code>	77
9.25.3.18	<code>unNumberProp</code>	78
9.25.3.19	<code>valuableProperties</code>	78
9.25.4	Documentation des données membres	78
9.25.4.1	<code>checkable</code>	78
9.25.4.2	<code>device</code>	78

9.25.4.3	device_prop	79
9.25.4.4	fatuuid	79
9.25.4.5	firstFat	79
9.25.4.6	headers	79
9.25.4.7	path	79
9.25.4.8	selected	79
9.25.4.9	stickid	79
9.25.4.10	uuid	79
9.26	Référence de la classe src.ownedUsbDisk.uDisk	79
9.26.1	Description détaillée	80
9.26.2	Documentation des constructeurs et destructeur	80
9.26.2.1	__init__	80
9.26.3	Documentation des fonctions membres	81
9.26.3.1	__getitem__	81
9.26.3.2	ensureOwner	81
9.26.3.3	headers	81
9.26.3.4	ownerByDb	82
9.26.3.5	readQuirks	82
9.26.3.6	tattoo	82
9.26.3.7	uniqueId	82
9.26.3.8	visibleDir	83
9.26.4	Documentation des données membres	83
9.26.4.1	headers	83
9.26.4.2	model	83
9.26.4.3	owner	83
9.26.4.4	vendor	83
9.26.4.5	visibleDirs	83
9.27	Référence de la classe src.mainWindow.UsbDiskDelegate	83
9.27.1	Description détaillée	84
9.27.2	Documentation des constructeurs et destructeur	84
9.27.2.1	__init__	84
9.27.3	Documentation des fonctions membres	84
9.27.3.1	paint	84
9.27.4	Documentation des données membres	84

9.27.4.1	busyPixmap	84
9.27.4.2	okPixmap	84
9.28	Référence de la classe src.mainWindow.usbTableModel	84
9.28.1	Description détaillée	85
9.28.2	Documentation des constructeurs et destructeur	85
9.28.2.1	__init__	85
9.28.3	Documentation des fonctions membres	85
9.28.3.1	columnCount	85
9.28.3.2	data	86
9.28.3.3	headerData	86
9.28.3.4	partition	86
9.28.3.5	rowCount	86
9.28.3.6	setData	86
9.28.3.7	sort	86
9.28.4	Documentation des données membres	86
9.28.4.1	checkable	87
9.28.4.2	donnees	87
9.28.4.3	header	87
9.28.4.4	pere	87
<b>10</b>	<b>Documentation des fichiers</b>	<b>89</b>
10.1	Référence du fichier src/__init__.py	89
10.2	Référence du fichier src/chooseInSticks.py	89
10.3	Référence du fichier src/copyToDialog1.py	89
10.4	Référence du fichier src/db.py	90
10.5	Référence du fichier src/diskFull.py	90
10.6	Référence du fichier src/globaldef.py	91
10.7	Référence du fichier src/help.py	91
10.8	Référence du fichier src/mainWindow.py	91
10.9	Référence du fichier src/marques.py	92
10.10	Référence du fichier src/mytextbrowser.py	92
10.11	Référence du fichier src/notification.py	93
10.12	Référence du fichier src/ownedUsbDisk.py	93
10.13	Référence du fichier src/preferences.py	93

10.14Référence du fichier <code>src/scolasync.py</code> . . . . .	94
10.15Référence du fichier <code>src/usbDisk.py</code> . . . . .	94
10.16Référence du fichier <code>src/usbThread.py</code> . . . . .	95
10.17Référence du fichier <code>src/version.py</code> . . . . .	95



# Chapitre 1

## ScolaSync

### 1.1 But de l'application

Scolasync est un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB.

### 1.2 CAHIER DE CHARGES DE SCOLASYNC

1. l'application doit pouvoir être utilisable par n'importe quel enseignant, par exemple un prof de langues quelques minutes après la prise en main.
2. une personne-ressource, ou le prof lui-même, doit pouvoir très simplement créer une association permanente entre les identifiants des clés USB et les noms d'élèves. Cette association doit pouvoir évoluer en fonction des classes à la demande de l'enseignant, d'une année sur l'autre, ou d'un cycle de travail à un autre.
3. un prof doit pouvoir envoyer un ensemble de fichiers vers les clés USB de ses élèves identiquement pour tous. L'individualisation peut se faire en branchant/débranchant les clés. Le prof doit avoir la possibilité de choisir, voire de créer le dossier de réception.
4. chaque élève doit pouvoir retrouver facilement ces fichiers et surtout la consigne expliquant ce qu'il doit faire, et comment il sera noté. Comme les lecteurs mp3 stockent souvent des fichiers dans des répertoires de noms variés, il faut pouvoir gérer ça.
5. le prof doit pouvoir récolter les clés USB des élèves et récupérer leur travail en quelques minutes seulement, par exemple en sélectionnant le dossier dans lequel se trouve le fichier à récupérer.
6. l'application doit renommer les fichiers en tenant compte du nom du baladeur, donc du nom de l'élève.
7. il faut pouvoir effacer des fichiers sur les clés, voire les remettre à zéro.

## 1.3 Licence

### ScolaSync version 1.0 :

un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB.

**Copyright © 2010 Georges Khaznadar** <[georgesk@offset.org](mailto:georgesk@offset.org)>

Ce projet est un logiciel libre : vous pouvez le redistribuer, le modifier selon les termes de la GPL (GNU Public License) dans les termes de la Free Software Foundation concernant la version 3 ou plus de la dite licence.

Ce programme est fait avec l'espoir qu'il sera utile mais **SANS AUCUNE GARANTIE**. Lisez la [licence](#) pour plus de détails.

## 1.4 Support

Si vous avez besoin d'un support pour ce programme, tel que : **garantie contractuelle, formation, adaptation plus précise** aux besoins de votre entreprise, etc. contactez l'association [OFFSET](#) et/ou [l'auteur](#) du logiciel.

## 1.5 Architecture de ScolaSync

Scolasync est bâti sur des composants logiciels libres, les plus notables sont les suivants :

- la bibliothèque Qt4 pour l'interface graphique
- la bibliothèque python-dbus pour l'interaction avec le noyau Linux 2.6
- la bibliothèque udisks pour interroger facilement le noyau sur le statut des disques, et pour réaliser certaines actions sur les disques et clés USB
- l'utilisation de threads pour mener en parallèle les actions qui concernent simultanément plusieurs clés USB

## Chapitre 2

# Hiérarchie de répertoires

### 2.1 Répertoires

Cette hiérarchie de répertoire est triée approximativement, mais pas complètement, par ordre alphabétique :

src . . . . . [13](#)



## Chapitre 3

# Index des espaces de nommage

### 3.1 Paquetages

Liste des paquetages avec une brève description (si disponible) :

<a href="#">scolasync</a>	Scolasync est un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB . . . . .	15
<a href="#">src</a>	. . . . .	15
<a href="#">src : :chooseInSticks</a>	. . . . .	16
<a href="#">src : :copyToDialog1</a>	. . . . .	16
<a href="#">src : :db</a>	. . . . .	17
<a href="#">src : :diskFull</a>	. . . . .	20
<a href="#">src : :globaldef</a>	. . . . .	21
<a href="#">src : :help</a>	. . . . .	22
<a href="#">src : :mainWindow</a>	. . . . .	22
<a href="#">src : :marques</a>	. . . . .	23
<a href="#">src : :mytextbrowser</a>	. . . . .	23
<a href="#">src : :notification</a>	. . . . .	24
<a href="#">src : :ownedUsbDisk</a>	. . . . .	25
<a href="#">src : :preferences</a>	. . . . .	25
<a href="#">src : :scolasync</a>	. . . . .	26
<a href="#">src : :usbDisk</a>	. . . . .	28
<a href="#">src : :usbThread</a>	. . . . .	29
<a href="#">src : :version</a>	. . . . .	30



## Chapitre 4

# Index des classes

### 4.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

src.usbDisk.Available . . . . .	36
src.ownedUsbDisk.Available . . . . .	34
src.notification.Notification . . . . .	60
QAbstractTableModel . . . . .	63
src.mainWindow.usbTableModel . . . . .	84
QDialog . . . . .	63
src.chooseInSticks.chooseDialog . . . . .	41
src.copyToDialog1.copyToDialog1 . . . . .	45
src.help.helpWindow . . . . .	50
src.preferences.preferenceWindow . . . . .	62
QMainWindow . . . . .	63
src.diskFull.mainWindow . . . . .	57
src.mainWindow.mainWindow . . . . .	51
QObject . . . . .	64
src.ownedUsbDisk.uDisk . . . . .	79
QStyledItemDelegate . . . . .	64
src.mainWindow.CheckBoxDelegate . . . . .	40
src.mainWindow.DiskSizeDelegate . . . . .	49
src.mainWindow.UsbDiskDelegate . . . . .	83
QTextBrowser . . . . .	64
src.mytextbrowser.myTextBrowser . . . . .	59
Thread . . . . .	64
src.usbThread.abstractThreadUSB . . . . .	31
src.usbThread.threadCopyFromUSB . . . . .	64
src.usbThread.threadCopyToUSB . . . . .	66
src.usbThread.threadDeleteInUSB . . . . .	68
src.usbThread.ThreadRegister . . . . .	69

src.usbDisk.uDisk . . . . .	71
src.ownedUsbDisk.uDisk . . . . .	79



## Chapitre 5

# Index des classes

### 5.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

<a href="#">src.usbThread.abstractThreadUSB</a>	
Une classe abstraite Cette classe sert de creuset pour les classe servant aux copies et aux effacement . . . . .	31
<a href="#">src.ownedUsbDisk.Available</a>	
Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires . . . . .	34
<a href="#">src.usbDisk.Available</a>	
Une classe pour représenter la collection des disques USB connectés . . . . .	36
<a href="#">src.mainWindow.CheckBoxDelegate</a>	40
<a href="#">src.chooseInSticks.chooseDialog</a>	
Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB . . . . .	41
<a href="#">src.copyToDialog1.copyToDialog1</a>	
Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB . . . . .	45
<a href="#">src.mainWindow.DiskSizeDelegate</a>	49
<a href="#">src.help.helpWindow</a>	50
<a href="#">src.mainWindow.mainWindow</a>	51
<a href="#">src.diskFull.mainWindow</a>	57
<a href="#">src.mytextbrowser.myTextBrowser</a>	
Une classe qui ouvre Firefox quand on clique sur un lien externe . . .	59
<a href="#">src.notification.Notification</a>	
Une classe pour afficher des notifications à l'écran . . . . .	60
<a href="#">src.preferences.preferenceWindow</a>	62
<a href="#">QAbstractTableModel</a>	63
<a href="#">QDialog</a>	63
<a href="#">QMainWindow</a>	63
<a href="#">QObject</a>	64

<a href="#">QStyledItemDelegate</a>	64
<a href="#">QTextBrowser</a>	64
<a href="#">Thread</a>	64
<a href="#">src.usbThread.threadCopyFromUSB</a>	
Classe pour les threads copiant depuis les clés USB	64
<a href="#">src.usbThread.threadCopyToUSB</a>	
Classe pour les threads copiant vers les clés USB	66
<a href="#">src.usbThread.threadDeleteInUSB</a>	
Classe pour les threads effaçant des sous-arbres dans les clés USB	68
<a href="#">src.usbThread.ThreadRegister</a>	
Une classe pour tenir un registre des threads concernant les baladeurs	69
<a href="#">src.usbDisk.uDisk</a>	
Une classe pour représenter un disque ou une partition	71
<a href="#">src.ownedUsbDisk.uDisk</a>	
Une classe qui ajoute un nom de propriétaire aux disque USB, et qui en même temps ajoute des particularités selon le nom du vendeur et le modèle	79
<a href="#">src.mainWindow.UsbDiskDelegate</a>	83
<a href="#">src.mainWindow.usbTableModel</a>	
Un modèle de table pour des séries de clés USB	84

## Chapitre 6

# Index des fichiers

### 6.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/___init___py	89
src/chooseInSticks.py	89
src/copyToDialog1.py	89
src/db.py	90
src/diskFull.py	90
src/globaldef.py	91
src/help.py	91
src/mainWindow.py	91
src/marques.py	92
src/mytextbrowser.py	92
src/notification.py	93
src/ownedUsbDisk.py	93
src/preferences.py	93
src/scolasync.py	94
src/usbDisk.py	94
src/usbThread.py	95
src/version.py	95



## Chapitre 7

# Documentation des répertoires

### 7.1 Répertoire de référence de src/

Grphe des dépendances de répertoires pour src/ :

#### Fichiers

- fichier [\\_\\_init\\_\\_.py](#)
- fichier [chooseInSticks.py](#)
- fichier [copyToDialog1.py](#)
- fichier [db.py](#)
- fichier [diskFull.py](#)
- fichier [globaldef.py](#)
- fichier [help.py](#)
- fichier [mainWindow.py](#)
- fichier [marques.py](#)
- fichier [mytextbrowser.py](#)
- fichier [notification.py](#)
- fichier [ownedUsbDisk.py](#)
- fichier [preferences.py](#)
- fichier [scolasync.py](#)
- fichier [usbDisk.py](#)
- fichier [usbThread.py](#)
- fichier [version.py](#)



## Chapitre 8

# Documentation des espaces de nommage

### 8.1 Référence de l'espace de nommage scolasync

Scolasync est un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB.

#### 8.1.1 Description détaillée

Scolasync est un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB.

### 8.2 Référence de l'espace de nommage src

#### Paquetages

- namespace [chooseInSticks](#)
- namespace [copyToDialog1](#)
- namespace [db](#)
- namespace [diskFull](#)
- namespace [globaldef](#)
- namespace [help](#)
- namespace [mainWindow](#)
- namespace [marques](#)
- namespace [mytextbrowser](#)
- namespace [notification](#)
- namespace [ownedUsbDisk](#)
- namespace [preferences](#)
- namespace [scolasync](#)
- namespace [usbDisk](#)
- namespace [usbThread](#)
- namespace [version](#)

### 8.3 Référence de l'espace de nommage src : :chooseInSticks

#### Classes

- class `chooseDialog`  
*Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB.*

#### Variables

- string `licenceEn`

#### 8.3.1 Documentation des variables

##### 8.3.1.1 string src : :chooseInSticks : :licenceEn

#### Valeur initiale :

```
1 """
2     file chooseInSticks.py
3     this file is part of the project scolasync
4
5     Copyright (C) 2010 Georges Khaznadar <georgesk@offset.org>
6
7     This program is free software: you can redistribute it and/or modify
8     it under the terms of the GNU General Public License as published by
9     the Free Software Foundation, either version 3 of the License, or
10    (at your option) any later version.
11
12    This program is distributed in the hope that it will be useful,
13    but WITHOUT ANY WARRANTY; without even the implied warranty of
14    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15    GNU General Public License for more details.
16
17    You should have received a copy of the GNU General Public License
18    along with this program. If not, see <http://www.gnu.org/licenses/>.
19 """
```

Définition à la ligne 4 du fichier `chooseInSticks.py`.

### 8.4 Référence de l'espace de nommage src : :copyToDialog1

#### Classes

- class `copyToDialog1`  
*Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB.*

#### Variables

- string `licenceEn`



- tuple `app` = `QApplication(sys.argv)`
- tuple `windows` = `copyToDialog1()`

#### 8.4.1 Documentation des variables

##### 8.4.1.1 tuple src : :copyToDialog1 : :app = QApplication(sys.argv)

Définition à la ligne 209 du fichier `copyToDialog1.py`.

##### 8.4.1.2 string src : :copyToDialog1 : :licenceEn

Valeur initiale :

```

1  """
2      file copyToDialog1.py
3      this file is part of the project scolasync
4
5      Copyright (C) 2010 Georges Khaznadar <georgesk@offset.org>
6
7      This program is free software: you can redistribute it and/or modify
8      it under the terms of the GNU General Public License as published by
9      the Free Software Foundation, either version 3 of the License, or
10     (at your option) any later version.
11
12     This program is distributed in the hope that it will be useful,
13     but WITHOUT ANY WARRANTY; without even the implied warranty of
14     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15     GNU General Public License for more details.
16
17     You should have received a copy of the GNU General Public License
18     along with this program. If not, see <http://www.gnu.org/licenses/>.
19 """

```

Définition à la ligne 4 du fichier `copyToDialog1.py`.

##### 8.4.1.3 tuple src : :copyToDialog1 : :windows = copyToDialog1()

Définition à la ligne 210 du fichier `copyToDialog1.py`.

## 8.5 Référence de l'espace de nommage src : :db

### Fonctions

- def `openDb`  
*Ouverture de la base de données de l'application, et création si nécessaire.*
- def `checkVersion`  
*Vérifie si la base de données reste compatible.*
- def `knowsId`  
*dît si une clé USB est déjà connue*
- def `tattooList`  
*Renvoie la liste des tatouages connus de la base de données.*

- def `readStudent`  
*renvoie l'étudiant qui possède une clé USB*
- def `readPrefs`  
*renvoie les préférences de ScolaSync*
- def `setWd`  
*définit le nouveau nom du répertoire de travail préféré.*
- def `writeStudent`  
*inscrit un étudiant comme propriétaire d'une clé USB*
- def `writePrefs`  
*inscrit les préférences*

## Variables

- dictionary `licence` = {}
- `database` = None
- `cursor` = None

## 8.5.1 Documentation des fonctions

### 8.5.1.1 def `src.db.checkVersion` ( *major, minor* )

Vérifie si la base de données reste compatible.

Un changement de version majeur implique une mise à jour en cas de base de donnée ancienne. Un changement de version mineur n'implique pas de changement de structure de la base de données.

Définition à la ligne 57 du fichier `db.py`.

Voici le graphe des appelants de cette fonction :

### 8.5.1.2 def `src.db.knowsId` ( *stickid, uuid, tattoo* )

dit si une clé USB est déjà connue

#### Paramètres

<i>stickid</i>	un identifiant de baladeur
<i>uuid</i>	un identifiant de partition
<i>tattoo</i>	un tatouage de partition

#### Renvoie

un booléen vrai si la clé USB est connue, faux sinon

Définition à la ligne 81 du fichier `db.py`.

Voici le graphe des appelants de cette fonction :

### 8.5.1.3 def `src.db.openDb` ( )

Ouverture de la base de données de l'application, et création si nécessaire.

**Renvoie**

une instance de base de données sqlite3

Définition à la ligne 37 du fichier db.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**8.5.1.4 def src.db.readPrefs ( )**

renvoie les préférences de ScolaSync

**Renvoie**

un dictionnaire de préférences

Définition à la ligne 114 du fichier db.py.

**8.5.1.5 def src.db.readStudent ( *stickid*, *uuid*, *tattoo* )**

renvoie l'étudiant qui possède une clé USB

**Renvoie**

un nom d'étudiant ou None si la clé est inconnue

Définition à la ligne 100 du fichier db.py.

**8.5.1.6 def src.db.setWd ( *newDir* )**

définit le nouveau nom du répertoire de travail préféré.

Définition à la ligne 130 du fichier db.py.

**8.5.1.7 def src.db.tattooList ( )**

Renvoie la liste des tatouages connus de la base de données.

Définition à la ligne 90 du fichier db.py.

**8.5.1.8 def src.db.writePrefs ( *prefs* )**

inscrit les préférences

**Paramètres**

<i>prefs</i>	un dictionnaire {"checkable" : booléen vrai si on doit afficher des cases à cocher, "workdir" : le répertoire préféré pour les fichiers de travail}
--------------	---

Définition à la ligne 156 du fichier db.py.

Voici le graphe d'appel pour cette fonction :

**8.5.1.9** `def src.db.writeStudent ( stickid, uuid, tattoo, student )`

inscrit un étudiant comme propriétaire d'une clé USB

#### Paramètres

<i>student</i>	un nom d'étudiant
----------------	-------------------

Définition à la ligne 140 du fichier db.py.

Voici le graphe d'appel pour cette fonction :

## 8.5.2 Documentation des variables

**8.5.2.1** `src : :db : :cursor = None`

Définition à la ligne 30 du fichier db.py.

**8.5.2.2** `src : :db : :database = None`

Définition à la ligne 29 du fichier db.py.

**8.5.2.3** `dictionary src : :db : :licence = {}`

Définition à la ligne 4 du fichier db.py.

## 8.6 Référence de l'espace de nommage src : :diskFull

### Classes

– class `mainWindow`

### Fonctions

– def `sceneWithUsage`

### Variables

– dictionary `licence = {}`

### 8.6.1 Documentation des fonctions

#### 8.6.1.1 `def src.diskFull.sceneWithUsage ( parent, rect, percent )`

##### Paramètres

<i>parent</i>	le widget père
<i>rect</i>	le QRect contenant la scène
<i>percent</i>	pourcentage utilisé

##### Renvoie

une QGraphicsScene avec un symbole d'occupation du disque

Définition à la ligne 60 du fichier diskFull.py.

### 8.6.2 Documentation des variables

#### 8.6.2.1 `dictionary src : :diskFull : :licence = {}`

Définition à la ligne 5 du fichier diskFull.py.

## 8.7 Référence de l'espace de nommage src : :globaldef

### Variables

- string `licenceEn`  
*globaldef.py is part of the package scolasync.*
- string `userShareDir` = "~/scolasync"
- string `logFileName` = "~/scolasync/scolasync.log"
- string `markFileName` = "~/scolasync/marques.py"

### 8.7.1 Documentation des variables

#### 8.7.1.1 `string src : :globaldef : :licenceEn`

##### Valeur initiale :

```

1 """
2     scolasync version %s:
3
4     a program to manage file transfers between a computer and a collection
5     of USB sticks.
6
7     Copyright (C) 2010 Georges Khaznadar <georgesk@offset.org>
8
9     This program is free software: you can redistribute it and/or modify
10    it under the terms of the GNU General Public License as published by
11    the Free Software Foundation, either version 3 of the License, or
12    (at your option) any later version.
13
```

```
14     This program is distributed in the hope that it will be useful,  
15     but WITHOUT ANY WARRANTY; without even the implied warranty of  
16     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the  
17     GNU General Public License for more details.  
18  
19     You should have received a copy of the GNU General Public License  
20     along with this program.  If not, see <http://www.gnu.org/licenses/>.  
21 """
```

[globaldef.py](#) is part of the package scolasync.

This module contains some definitions which can be reused globally in the application

Définition à la ligne 11 du fichier globaldef.py.

**8.7.1.2** `string src : :globaldef : :logFileName = "~/scolasync/scolasync.log"`

Définition à la ligne 37 du fichier globaldef.py.

**8.7.1.3** `string src : :globaldef : :markFileName = "~/scolasync/marques.py"`

Définition à la ligne 38 du fichier globaldef.py.

**8.7.1.4** `string src : :globaldef : :userShareDir = "~/scolasync"`

Définition à la ligne 36 du fichier globaldef.py.

## 8.8 Référence de l'espace de nommage src : :help

### Classes

– class [helpWindow](#)

### Variables

– dictionary [licence](#) = {}

#### 8.8.1 Documentation des variables

**8.8.1.1** `dictionary src : :help : :licence = {}`

Définition à la ligne 5 du fichier help.py.

## 8.9 Référence de l'espace de nommage src : :mainWindow

### Classes

- class [mainWindow](#)
- class [usbTableModel](#)  
*Un modèle de table pour des séries de clés USB.*
- class [CheckBoxDelegate](#)
- class [UsbDiskDelegate](#)
- class [DiskSizeDelegate](#)

### Fonctions

- def [firstdir](#)  
*Renvoie le premier répertoire existant d'une liste de propositions.*
- def [CheckBoxRect](#)

### Variables

- dictionary [licence](#) = {}

## 8.9.1 Documentation des fonctions

### 8.9.1.1 def src.mainWindow.CheckBoxRect ( *view\_item\_style\_options* )

Définition à la ligne 545 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

### 8.9.1.2 def src.mainWindow.firstdir ( *l* )

Renvoie le premier répertoire existant d'une liste de propositions.

#### Paramètres

<i>l</i>	la liste de propositions
----------	--------------------------

Définition à la ligne 43 du fichier mainWindow.py.

## 8.9.2 Documentation des variables

### 8.9.2.1 dictionary src : :mainWindow : :licence = {}

Définition à la ligne 5 du fichier mainWindow.py.

## 8.10 Référence de l'espace de nommage src : :marques

## 8.11 Référence de l'espace de nommage src : :mytextbrowser

### Classes

- class `myTextBrowser`  
*Une classe qui ouvre Firefox quand on clique sur un lien externe.*

### Variables

- dictionary `licence` = {}

#### 8.11.1 Documentation des variables

8.11.1.1 dictionary `src : :mytextbrowser : :licence` = {}

Définition à la ligne 5 du fichier `mytextbrowser.py`.

## 8.12 Référence de l'espace de nommage `src : :notification`

### Classes

- class `Notification`  
*Une classe pour afficher des notifications à l'écran.*

### Variables

- dictionary `licence` = {}
- tuple `notif`

#### 8.12.1 Documentation des variables

8.12.1.1 dictionary `src : :notification : :licence` = {}

Définition à la ligne 5 du fichier `notification.py`.

8.12.1.2 tuple `src : :notification : :notif`

#### Valeur initiale :

```
1 Notification(app_name="AppliTest",
2               summary="Notification de test",
3               body="Voici le corps de la notification",
4               app_icon="/usr/share/pixmaps/vlc.png",
5               expire_timeout=7000)
```

Définition à la ligne 75 du fichier `notification.py`.



### 8.13 Référence de l'espace de nommage src : :ownedUsbDisk

#### Classes

- class `uDisk`  
*une classe qui ajoute un nom de propriétaire aux disque USB, et qui en même temps ajoute des particularités selon le nom du vendeur et le modèle.*
- class `Available`  
*Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires.*

#### Fonctions

- def `editRecord`  
*édition de la base de données.*

#### Variables

- dictionary `licence` = {}

#### 8.13.1 Documentation des fonctions

8.13.1.1 `def src.ownedUsbDisk.editRecord ( owd, student = " " )`

édition de la base de données.

##### Paramètres

<code>owd</code>	une instance de <code>ownedUsbDisk</code>
<code>student</code>	nom de propriétaire pour la clé. Chaîne vide par défaut.

Définition à la ligne 43 du fichier `ownedUsbDisk.py`.

#### 8.13.2 Documentation des variables

8.13.2.1 `dictionary src : :ownedUsbDisk : :licence = {}`

Définition à la ligne 4 du fichier `ownedUsbDisk.py`.

### 8.14 Référence de l'espace de nommage src : :preferences

#### Classes

- class `preferenceWindow`

## Variables

- dictionary `licence` = {}

### 8.14.1 Documentation des variables

#### 8.14.1.1 dictionary `src : :preferences : :licence` = {}

Définition à la ligne 5 du fichier `preferences.py`.

## 8.15 Référence de l'espace de nommage `src : :scolasync`

## Fonctions

- def `usage`  
*affiche le mode d'emploi à la console*
- def `run`  
*C'est la fonction principale.*

## Variables

- dictionary `licence` = {}
- string `licenceEn`
- string `licenceFr`

### 8.15.1 Documentation des fonctions

#### 8.15.1.1 def `src.scolasync.run` ( )

C'est la fonction principale.

Elle crée un fichier de journalisation vide, puis lance l'application interactive elle-même.  
Le fichier de journalisation est `$(HOME)/.scolasync/scolasync.log` sous Linux

Définition à la ligne 156 du fichier `scolasync.py`.

Voici le graphe d'appel pour cette fonction :

#### 8.15.1.2 def `src.scolasync.usage` ( )

affiche le mode d'emploi à la console

Définition à la ligne 140 du fichier `scolasync.py`.

Voici le graphe des appelants de cette fonction :

### 8.15.2 Documentation des variables

**8.15.2.1 dictionary src : :scolasync : :licence = {}**

Définition à la ligne 85 du fichier scolasync.py.

**8.15.2.2 string src : :scolasync : :licenceEn**

**Valeur initiale :**

```
1 """
2     scolasync version %s:
3
4     a program to manage file transfers between a computer and a collection
5     of USB sticks.
6
7     Copyright (C) 2010 Georges Khaznadar <georgesk@offset.org>
8
9     This program is free software: you can redistribute it and/or modify
10    it under the terms of the GNU General Public License as published by
11    the Free Software Foundation, either version 3 of the License, or
12    (at your option) any later version.
13
14    This program is distributed in the hope that it will be useful,
15    but WITHOUT ANY WARRANTY; without even the implied warranty of
16    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17    GNU General Public License for more details.
18
19    You should have received a copy of the GNU General Public License
20    along with this program. If not, see <http://www.gnu.org/licenses/>.
21 """
```

Définition à la ligne 86 du fichier scolasync.py.

**8.15.2.3 string src : :scolasync : :licenceFr**

**Valeur initiale :**

```
1 """
2     scolasync version %s :
3
4     un programme pour gérer des transferts de fichiers entre un
5     ordinateur et une collection de clés USB.
6
7     Copyright (C) 2010 Georges Khaznadar <georgesk@offset.org>
8
9     Ce projet est un logiciel libre : vous pouvez le redistribuer, le
10    modifier selon les terme de la GPL (GNU Public License) dans les
11    termes de la Free Software Foundation concernant la version 3 ou
12    plus de la dite licence.
13
14    Ce programme est fait avec l'espoir qu'il sera utile mais SANS
15    AUCUNE GARANTIE. Lisez la licence pour plus de détails.
16
17    <http://www.gnu.org/licenses/>.
18 """
```

Définition à la ligne 109 du fichier scolasync.py.

## 8.16 Référence de l'espace de nommage src : :usbDisk

### Classes

- class `uDisk`  
*une classe pour représenter un disque ou une partition.*
- class `Available`  
*une classe pour représenter la collection des disques USB connectés*

### Variables

- dictionary `licence` = {}
- string `licence_en`
- tuple `machin` = `Available()`

#### 8.16.1 Documentation des variables

##### 8.16.1.1 dictionary src : :usbDisk : :licence = {}

Définition à la ligne 4 du fichier `usbDisk.py`.

##### 8.16.1.2 string src : :usbDisk : :licence\_en

Valeur initiale :

```

1 """
2     file usbDisk.py
3     this file is part of the project scolasync
4
5     Copyright (C) 2010 Georges Khaznadar <georgesk@ofset.org>
6
7     This program is free software: you can redistribute it and/or modify
8     it under the terms of the GNU General Public License as published by
9     the Free Software Foundation, either version 3 of the License, or
10    (at your option) any later version.
11
12    This program is distributed in the hope that it will be useful,
13    but WITHOUT ANY WARRANTY; without even the implied warranty of
14    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15    GNU General Public License for more details.
16
17    You should have received a copy of the GNU General Public License
18    along with this program. If not, see <http://www.gnu.org/licenses/>.
19 """

```

Définition à la ligne 5 du fichier `usbDisk.py`.

##### 8.16.1.3 tuple src : :usbDisk : :machin = Available()

Définition à la ligne 515 du fichier `usbDisk.py`.

## 8.17 Référence de l'espace de nommage src : :usbThread

### Classes

- class [ThreadRegister](#)  
*Une classe pour tenir un registre des threads concernant les baladeurs.*
- class [abstractThreadUSB](#)  
*Une classe abstraite Cette classe sert de creuset pour les classe servant aux copies et aux effacement.*
- class [threadCopyToUSB](#)  
*Classe pour les threads copiant vers les clés USB.*
- class [threadCopyFromUSB](#)  
*Classe pour les threads copiant depuis les clés USB.*
- class [threadDeleteInUSB](#)  
*Classe pour les threads effaçant des sous-arbres dans les clés USB.*

### Variables

- string [licenceEn](#)
- int [\\_threadNumber](#) = 0

#### 8.17.1 Documentation des variables

##### 8.17.1.1 int src : :usbThread : :\_threadNumber = 0

Définition à la ligne 26 du fichier usbThread.py.

##### 8.17.1.2 string src : :usbThread : :licenceEn

#### Valeur initiale :

```

1  """
2      file usbThread.py
3      this file is part of the project scolasync
4
5      Copyright (C) 2010 Georges Khaznadar <georgesk@ofset.org>
6
7      This program is free software: you can redistribute it and/or modify
8      it under the terms of the GNU General Public License as published by
9      the Free Software Foundation, either version 3 of the License, or
10     (at your option) any later version.
11
12     This program is distributed in the hope that it will be useful,
13     but WITHOUT ANY WARRANTY; without even the implied warranty of
14     MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15     GNU General Public License for more details.
16
17     You should have received a copy of the GNU General Public License
18     along with this program. If not, see <http://www.gnu.org/licenses/>.
19 """

```

Définition à la ligne 4 du fichier usbThread.py.

## 8.18 Référence de l'espace de nommage src : :version

### Fonctions

- def `major`
- def `minor`
- def `version`

### Variables

- dictionary `licence` = {}

### 8.18.1 Documentation des fonctions

#### 8.18.1.1 def `src.version.major` ( )

##### Renvoie

le numéro majeur de version

Définition à la ligne 30 du fichier `version.py`.

Voici le graphe des appelants de cette fonction :

#### 8.18.1.2 def `src.version.minor` ( )

##### Renvoie

le numéro mineur de version

Définition à la ligne 37 du fichier `version.py`.

Voici le graphe des appelants de cette fonction :

#### 8.18.1.3 def `src.version.version` ( )

##### Renvoie

l'identifiant de la version

Définition à la ligne 44 du fichier `version.py`.

Voici le graphe d'appel pour cette fonction :

### 8.18.2 Documentation des variables

#### 8.18.2.1 dictionary `src : :version : :licence` = {}

Définition à la ligne 4 du fichier `version.py`.

## Chapitre 9

# Documentation des classes

### 9.1 Référence de la classe `src.usbThread.abstractThreadUSB`

Une classe abstraite Cette classe sert de creuset pour les classe servant aux copies et aux effacement.

Graphe d'héritage de `src.usbThread.abstractThreadUSB` :

Graphe de collaboration de `src.usbThread.abstractThreadUSB` :

#### Fonctions membres publiques

- `def __init__`  
*Constructeur Crée un thread pour copier une liste de fichiers vers une clé USB.*
- `def __str__`  
*Renvoie une chaîne informative sur le thread.*
- `def threadType`
- `def toDo`  
*La fonction abstraite pour les choses à faire.*

#### Attributs publics

- `cmd`
- `ud`
- `fileList`
- `subdir`
- `dest`
- `logfile`

#### 9.1.1 Description détaillée

Une classe abstraite Cette classe sert de creuset pour les classe servant aux copies et aux effacement.

Définition à la ligne 144 du fichier `usbThread.py`.

## 9.1.2 Documentation des constructeurs et destructeur

9.1.2.1 **def src.usbThread.abstractThreadUSB.\_\_init\_\_( self, ud, fileList, subdir, dest=None, logfile="/dev/null" )**

Constructeur Crée un thread pour copier une liste de fichiers vers une clé USB.

### Paramètres

<i>ud</i>	l'instance uDisk correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à traiter
<i>subdir</i>	un sous-répertoire de la clé USB
<i>dest</i>	un répertoire de destination si nécessaire, None par défaut
<i>logfile</i>	un fichier de journalisation, /dev/null par défaut

Définition à la ligne 155 du fichier usbThread.py.

## 9.1.3 Documentation des fonctions membres

9.1.3.1 **def src.usbThread.abstractThreadUSB.\_\_str\_\_( self )**

Renvoie une chaîne informative sur le thread.

### Renvoie

une chaîne donnant des informations sur ce qui va se passer dans le thread qui a été créé.

Définition à la ligne 173 du fichier usbThread.py.

Voici le graphe d'appel pour cette fonction :

9.1.3.2 **def src.usbThread.abstractThreadUSB.threadType ( self )**

### Renvoie

une chaîne courte qui informe sur le type de thread

Réimplémentée dans [src.usbThread.threadCopyToUSB](#).

Définition à la ligne 188 du fichier usbThread.py.

Voici le graphe des appelants de cette fonction :

9.1.3.3 **def src.usbThread.abstractThreadUSB.todo ( self, ud, fileList, subdir, dest, logfile )**

La fonction abstraite pour les choses à faire.



## Paramètres

<i>ud</i>	l'instance <code>uDisk</code> correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à traiter
<i>subdir</i>	un sous-répertoire de la clé USB
<i>dest</i>	un répertoire de destination
<i>logfile</i>	un fichier de journalisation

Réimplémentée dans [src.usbThread.threadDeleteInUSB](#), [src.usbThread.threadCopyFromUSB](#), et [src.usbThread.threadCopyToUSB](#).

Définition à la ligne 200 du fichier `usbThread.py`.

#### 9.1.4 Documentation des données membres

##### 9.1.4.1 `src :usbThread.abstractThreadUSB :cmd`

Réimplémentée dans [src.usbThread.threadDeleteInUSB](#), [src.usbThread.threadCopyFromUSB](#), et [src.usbThread.threadCopyToUSB](#).

Définition à la ligne 155 du fichier `usbThread.py`.

##### 9.1.4.2 `src :usbThread.abstractThreadUSB :dest`

Définition à la ligne 155 du fichier `usbThread.py`.

##### 9.1.4.3 `src :usbThread.abstractThreadUSB :fileList`

Définition à la ligne 155 du fichier `usbThread.py`.

##### 9.1.4.4 `src :usbThread.abstractThreadUSB :logfile`

Définition à la ligne 155 du fichier `usbThread.py`.

##### 9.1.4.5 `src :usbThread.abstractThreadUSB :subdir`

Définition à la ligne 155 du fichier `usbThread.py`.

##### 9.1.4.6 `src :usbThread.abstractThreadUSB :ud`

Définition à la ligne 155 du fichier `usbThread.py`.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/usbThread.py](#)

## 9.2 Référence de la classe `src.ownedUsbDisk.Available`

Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires.

Graphes d'héritage de `src.ownedUsbDisk.Available` :

Graphes de collaboration de `src.ownedUsbDisk.Available` :

### Fonctions membres publiques

– `def __init__`

### Attributs publics

– `checkable`

– `access`

*on monte les partitions si nécessaire*

– `bus`

– `disks`

– `enumDev`

– `firstFats`

*récupération des disques usb dans le dictionnaire `self.disks`*

### 9.2.1 Description détaillée

Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires.

On a répliqué le code de la classe parente, ne sachant pas si le constructeur prendrait les `uDisks` dans le module courant ou dans le module parent. Pour éviter les confusion il faudrait peut-être faire une classe abstraite [Available](#)

Définition à la ligne 229 du fichier `ownedUsbDisk.py`.

### 9.2.2 Documentation des constructeurs et destructeur

9.2.2.1 `def src.ownedUsbDisk.Available.__init__( self, checkable=False, access="disk" )`

#### Paramètres

<code>checkable</code>	: vrai si on veut pouvoir cocher les disques de la collection. Faux par défaut.
<code>access</code>	définit le type d'accès souhaité. Par défaut, c'est "disk" c'est à dire qu'on veut la liste des disques USB. Autres valeurs possibles : "firstFat" pour les premières partitions vfat.

Réimplémentée à partir de [src.usbDisk.Available](#).

Définition à la ligne 238 du fichier `ownedUsbDisk.py`.

### 9.2.3 Documentation des données membres

#### 9.2.3.1 `src : :ownedUsbDisk.Available : :access`

on monte les partitions si nécessaire

Réimplémentée à partir de [src.usbDisk.Available](#).

Définition à la ligne 238 du fichier `ownedUsbDisk.py`.

#### 9.2.3.2 `src : :ownedUsbDisk.Available : :bus`

Réimplémentée à partir de [src.usbDisk.Available](#).

Définition à la ligne 238 du fichier `ownedUsbDisk.py`.

#### 9.2.3.3 `src : :ownedUsbDisk.Available : :checkable`

Réimplémentée à partir de [src.usbDisk.Available](#).

Définition à la ligne 238 du fichier `ownedUsbDisk.py`.

#### 9.2.3.4 `src : :ownedUsbDisk.Available : :disks`

Réimplémentée à partir de [src.usbDisk.Available](#).

Définition à la ligne 238 du fichier `ownedUsbDisk.py`.

#### 9.2.3.5 `src : :ownedUsbDisk.Available : :enumDev`

Réimplémentée à partir de [src.usbDisk.Available](#).

Définition à la ligne 238 du fichier `ownedUsbDisk.py`.

#### 9.2.3.6 `src : :ownedUsbDisk.Available : :firstFats`

récupération des disques usb dans le dictionnaire `self.disks`

une deuxième passe pour récupérer et associer les partitions on s'assure que chaque disque a bien un propriétaire sinon on le demande on fabrique la liste des premières partitions FAT, en positionnant les attributs de propriétaire

Réimplémentée à partir de [src.usbDisk.Available](#).

Définition à la ligne 244 du fichier `ownedUsbDisk.py`.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/ownedUsbDisk.py](#)

### 9.3 Référence de la classe src.usbDisk.Available

une classe pour représenter la collection des disques USB connectés

Graphes d'héritage de src.usbDisk.Available :

#### Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)  
*Le constructeur.*
- def [\\_\\_trunc\\_\\_](#)
- def [compare](#)  
*Sert à comparer deux collections de disques, par exemple une collection passée et une collection présente.*
- def [contains](#)  
*Permet de déterminer si un disque est dans la collection.*
- def [summary](#)  
*Fournit une représentation imprimable d'un résumé*
- def [\\_\\_str\\_\\_](#)  
*Fournit une représentation imprimable.*
- def [\\_\\_getitem\\_\\_](#)  
*Renvoie le nième disque.*
- def [\\_\\_len\\_\\_](#)  
*Renseigne sur la longueur de la collection.*
- def [getFirstFats](#)  
*Facilite l'accès aux partitions de type DOS-FAT, et a un effet de bord : marque le disque avec l'uuid de la première partition FAT.*

#### Attributs publics

- [checkable](#)
- [access](#)  
*on monte les partitions si nécessaire*
- [bus](#)
- [disks](#)
- [enumDev](#)
- [firstFats](#)  
*récupération des disques usb dans le dictionnaire self.disks*

#### 9.3.1 Description détaillée

une classe pour représenter la collection des disques USB connectés

les attributs publics sont :

- **checkable** booléen vrai si on veut gérer des sélections de disques
- **access** le type d'accès qu'on veut pour les items
- **bus** une instance de dbus.SystemBus
- **disks** la collection de disques USB, organisée en un dictionnaire de disques : les clés sont les disques, qui renvoient à un ensemble de partitions du disque
- **enumdev** une liste de chemins dbus vers les disques trouvés
- **firstFats** une liste composée de la première partition DOS-FAT de chaque disque USB.

Définition à la ligne 356 du fichier usbDisk.py.

### 9.3.2 Documentation des constructeurs et destructeur

9.3.2.1 `def src.usbDisk.Available.__init__( self, checkable=False, access="disk" )`

Le constructeur.

#### Paramètres

<i>checkable</i>	: vrai si on veut pouvoir cocher les disques de la collection. Faux par défaut.
<i>access</i>	définit le type d'accès souhaité. Par défaut, c'est "disk" c'est à dire qu'on veut la liste des disques USB. Autres valeurs possibles : "firstFat" pour les premières partitions vfat.

Réimplémentée dans [src.ownedUsbDisk.Available](#).

Définition à la ligne 367 du fichier `usbDisk.py`.

### 9.3.3 Documentation des fonctions membres

9.3.3.1 `def src.usbDisk.Available.__getitem__( self, n )`

Renvoie le nième disque.

Le fonctionnement dépend du paramètre `self.access`

#### Paramètres

<i>n</i>	un numéro
----------	-----------

#### Renvoie

le nième disque USB connecté

Définition à la ligne 468 du fichier `usbDisk.py`.

9.3.3.2 `def src.usbDisk.Available.__len__( self )`

Renseigne sur la longueur de la collection.

Le fonctionnement dépend du paramètre `self.access`

#### Renvoie

la longueur de la collection de disques renvoyée

Définition à la ligne 480 du fichier `usbDisk.py`.

9.3.3.3 `def src.usbDisk.Available.__str__( self )`

Fournit une représentation imprimable.

**Renvoie**

une représentation imprimable de la collection

Définition à la ligne 449 du fichier usbDisk.py.

**9.3.3.4 def src.usbDisk.Available.\_\_trunc\_\_( self )****Renvoie**

le nombre de medias connectés

Définition à la ligne 403 du fichier usbDisk.py.

**9.3.3.5 def src.usbDisk.Available.compare( self, other )**

Sert à comparer deux collections de disques, par exemple une collection passée et une collection présente.

**Paramètres**

<i>other</i>	une instance de <a href="#">Available</a>
--------------	---

**Renvoie**

vrai si other semble être la même collection de disques USB

Définition à la ligne 413 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

**9.3.3.6 def src.usbDisk.Available.contains( self, ud )**

Permet de déterminer si un disque est dans la collection.

**Paramètres**

<i>ud</i>	une instance de <a href="#">uDisk</a>
-----------	---------------------------------------

**Renvoie**

vrai si le [uDisk](#) ud est dans la collection

Définition à la ligne 423 du fichier usbDisk.py.

**9.3.3.7 def src.usbDisk.Available.getFirstFats( self, setOwners = False )**

Facilite l'accès aux partitions de type DOS-FAT, et a un effet de bord : marque le disque avec l'uuid de la première partition FAT.

## Paramètres

<code>setOwners</code>	si égale à True, signale que la liste devra comporter des attributs de propriétaire de medias.
------------------------	--

## Renvoie

une liste de partitions, constituée de la première partition de type FAT de chaque disque USB connecté

Définition à la ligne 496 du fichier `usbDisk.py`.

9.3.3.8 `def src.usbDisk.Available.summary ( self )`

Fournit une représentation imprimable d'un résumé

## Renvoie

une représentation imprimable d'un résumé de la collection

Définition à la ligne 433 du fichier `usbDisk.py`.

Voici le graphe des appelants de cette fonction :

## 9.3.4 Documentation des données membres

9.3.4.1 `src : :usbDisk.Available : :access`

on monte les partitions si nécessaire

Réimplémentée dans [src.ownedUsbDisk.Available](#).

Définition à la ligne 367 du fichier `usbDisk.py`.

9.3.4.2 `src : :usbDisk.Available : :bus`

Réimplémentée dans [src.ownedUsbDisk.Available](#).

Définition à la ligne 367 du fichier `usbDisk.py`.

9.3.4.3 `src : :usbDisk.Available : :checkable`

Réimplémentée dans [src.ownedUsbDisk.Available](#).

Définition à la ligne 367 du fichier `usbDisk.py`.

9.3.4.4 `src : :usbDisk.Available : :disks`

Réimplémentée dans [src.ownedUsbDisk.Available](#).

Définition à la ligne 367 du fichier `usbDisk.py`.

#### 9.3.4.5 `src : :usbDisk.Available : :enumDev`

Réimplémentée dans [src.ownedUsbDisk.Available](#).

Définition à la ligne 367 du fichier `usbDisk.py`.

#### 9.3.4.6 `src : :usbDisk.Available : :firstFats`

récupération des disques usb dans le dictionnaire `self.disks`

une deuxième passe pour récupérer et associer les partitions on fabrique la liste des premières partitions FAT

Réimplémentée dans [src.ownedUsbDisk.Available](#).

Définition à la ligne 370 du fichier `usbDisk.py`.

La documentation de cette classe a été générée à partir du fichier suivant :

– `src/usbDisk.py`

## 9.4 Référence de la classe `src.mainWindow.CheckBoxDelegate`

Graphe d'héritage de `src.mainWindow.CheckBoxDelegate` :

Graphe de collaboration de `src.mainWindow.CheckBoxDelegate` :

### Fonctions membres publiques

- `def __init__`
- `def paint`
- `def editorEvent`

#### 9.4.1 Description détaillée

Définition à la ligne 551 du fichier `mainWindow.py`.

#### 9.4.2 Documentation des constructeurs et destructeur

9.4.2.1 `def src.mainWindow.CheckBoxDelegate.__init__( self, parent )`

Définition à la ligne 552 du fichier `mainWindow.py`.

#### 9.4.3 Documentation des fonctions membres



9.4.3.1 `def src.mainWindow.CheckBoxDelegate.editorEvent ( self, event, model, option, index )`

Définition à la ligne 566 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

9.4.3.2 `def src.mainWindow.CheckBoxDelegate.paint ( self, painter, option, index )`

Définition à la ligne 555 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/mainWindow.py](#)

## 9.5 Référence de la classe src.chooseInSticks.chooseDialog

Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB.

Graphe d'héritage de src.chooseInSticks.chooseDialog :

Graphe de collaboration de src.chooseInSticks.chooseDialog :

### Fonctions membres publiques

- `def __init__`  
*Le constructeur.*
- `def listStorages`  
*Met en place la liste des noms de baladeurs connectés en tenant compte du nom de répertoire de travail et d'un baladeur éventuellement sélectionné dans la fenêtre principale.*
- `def checkWorkDirs`  
*met à jour la possibilité de sélectionner les baladeurs dans la liste selon qu'ils ont ou pas un répertoire de travail, puis sélectionne si possible un baladeur, si aucun ne l'était avant.*
- `def baseDir`
- `def selectedDiskMountPoint`
- `def selectedDiskOwner`
- `def changeWd`  
*changement du répertoire de travail*
- `def choose`  
*Facilite le choix de motifs de fichiers en recherchant dans les clés USB, modifie l'éditeur de ligne de texte et place le fichier choisi dans la liste.*
- `def choose_dir`  
*Facilite le choix de motifs de répertoires en recherchant dans les clés USB, modifie l'éditeur de ligne de texte et place le répertoire choisi dans la liste.*
- `def activate`  
*Fonction de rappel quand un item de la liste est activé*
- `def plus`  
*Permet de choisir et d'ajouter un nouveau fichier ou répertoire à supprimer.*
- `def minus`  
*Permet de retirer de la liste des fichiers à supprimer ceux qu'on a sélectionnés.*
- `def append`  
*Ajoute un chemin avec ou sans jokers à la liste des chemins à supprimer.*
- `def pathList`

*renvoie la liste des chemins sélectionnés*

### Attributs publics

- [mainWindow](#)
- [ownedUsbDictionary](#)

*peuplement de la zone des noms de baladeurs*

### 9.5.1 Description détaillée

Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB.

Définition à la ligne 34 du fichier chooseInSticks.py.

### 9.5.2 Documentation des constructeurs et destructeur

9.5.2.1 `def src.chooseInSticks.chooseDialog.__init__( self, parent=None, title1=" ", title2=" ", ok="OK" )`

Le constructeur.

#### Paramètres

<i>parent</i>	un <a href="#">mainWindow</a> , qui est censé contenir des données telles que parent.-workdir, ...
<i>title1</i>	le titre du dialogue
<i>title2</i>	le titre pour la série de fichiers/modèles
<i>ok</i>	le texte du bouton OK

Définition à la ligne 44 du fichier chooseInSticks.py.

### 9.5.3 Documentation des fonctions membres

9.5.3.1 `def src.chooseInSticks.chooseDialog.activate( self, item )`

Fonction de rappel quand un item de la liste est activé

#### Paramètres

<i>item</i>	désignation de l'item activé
-------------	------------------------------

Définition à la ligne 241 du fichier chooseInSticks.py.

9.5.3.2 `def src.chooseInSticks.chooseDialog.append( self, path )`

Ajoute un chemin avec ou sans jokers à la liste des chemins à supprimer.

## Paramètres

<i>path</i>	le chemin
-------------	-----------

Définition à la ligne 275 du fichier `chooseInSticks.py`.

Voici le graphe des appelants de cette fonction :

9.5.3.3 `def src.chooseInSticks.chooseDialog.baseDir ( self )`

## Renvoie

le répertoire à partir duquel on peut commencer à faire un choix de fichier ou de sous-répertoire. Il dépend du baladeur sélectionné s'il y en a un et du nom du répertoire de travail. Si on n'arrive pas à déterminer ce répertoire, renvoie `None`

Définition à la ligne 151 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

9.5.3.4 `def src.chooseInSticks.chooseDialog.changeWd ( self )`

changement du répertoire de travail

Définition à la ligne 185 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

9.5.3.5 `def src.chooseInSticks.chooseDialog.checkWorkDirs ( self )`

met à jour la possibilité de sélectionner les baladeurs dans la liste selon qu'ils ont ou pas un répertoire de travail, puis sélectionne si possible un baladeur, si aucun ne l'était avant.

Définition à la ligne 110 du fichier `chooseInSticks.py`.

Voici le graphe des appelants de cette fonction :

9.5.3.6 `def src.chooseInSticks.chooseDialog.choose ( self, kind = "file" )`

Facilite le choix de motifs de fichiers en recherchant dans les clés USB, modifie l'éditeur de ligne de texte et place le fichier choisi dans la liste.

## Paramètres

<i>kind</i>	type d'élément à choisir : "file" pour un fichier, "dir" pour un répertoire
-------------	---

Définition à la ligne 198 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

#### 9.5.3.7 `def src.chooseInSticks.chooseDialog.choose_dir ( self )`

Facilite le choix de motifs de répertoires en recherchant dans les clés USB, modifie l'éditeur de ligne de texte et place le répertoire choisi dans la liste.

Définition à la ligne 233 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.5.3.8 `def src.chooseInSticks.chooseDialog.listStorages ( self )`

Met en place la liste des noms de baladeurs connectés en tenant compte du nom de répertoire de travail et d'un baladeur éventuellement sélectionné dans la fenêtre principale.

Définition à la ligne 89 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.5.3.9 `def src.chooseInSticks.chooseDialog.minus ( self )`

Permet de retirer de la liste des fichiers à supprimer ceux qu'on a sélectionnés.

Définition à la ligne 259 du fichier `chooseInSticks.py`.

#### 9.5.3.10 `def src.chooseInSticks.chooseDialog.pathList ( self )`

renvoie la liste des chemins sélectionnés

##### Renvoie

une liste de chemins, sous forme de `QStrings`

Définition à la ligne 289 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.5.3.11 `def src.chooseInSticks.chooseDialog.plus ( self )`

Permet de choisir et d'ajouter un nouveau fichier ou répertoire à supprimer.

Définition à la ligne 249 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

#### 9.5.3.12 `def src.chooseInSticks.chooseDialog.selectedDiskMountPoint ( self )`

**Renvoie**

le point de montage du support sélectionné s'il y en a un

Définition à la ligne 162 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.5.3.13 def src.chooseInSticks.chooseDialog.selectedDiskOwner ( self )****Renvoie**

le nom du propriétaire du disque sélectionné s'il y en a un, sinon None.

Définition à la ligne 174 du fichier chooseInSticks.py.

Voici le graphe des appelants de cette fonction :

**9.5.4 Documentation des données membres****9.5.4.1 src : :chooseInSticks.chooseDialog : :mainWindow**

Définition à la ligne 44 du fichier chooseInSticks.py.

**9.5.4.2 src : :chooseInSticks.chooseDialog : :ownedUsbDictionary**

peuplement de la zone des noms de baladeurs

Définition à la ligne 49 du fichier chooseInSticks.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– src/[chooseInSticks.py](#)

**9.6 Référence de la classe src.copyToDialog1.copyToDialog1**

Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB.

Graphe d'héritage de src.copyToDialog1.copyToDialog1 :

Graphe de collaboration de src.copyToDialog1.copyToDialog1 :

**Fonctions membres publiques**

- def [changeWd](#)  
*changement du répertoire de travail*
- def [cancel](#)  
*L'action provoquée par le bouton d'échappement : fermeture du dialogue.*
- def [cont](#)

- L'action provoquée par le bouton de continuation : fermeture du dialogue et self.ok devient vrai.*
- def `setupFromListe`  
*Met en place un visionneur de fichiers dans la liste source.*
  - def `setFromListeDir`  
*Choisit un répertoire pour la liste source.*
  - def `cd`  
*Change le répertoire courant si possible.*
  - def `setupToListe`  
*Met en place un visionneur de fichiers pour les fichiers reçus.*
  - def `select`  
*Ajoute le répertoire ou le fichier sélectionné dans le navigateur de fichiers à la liste de sélections.*
  - def `displaySize`  
*Affiche la taille de la sélection courante.*
  - def `remove`  
*Supprime le répertoire ou le fichier sélectionné dans la liste de sélections.*
  - def `selectedList`  
*Renvoie une liste de répertoires et de fichiers qui ont été sélectionnés pour la copie sur clé USB.*

### Attributs publics

- `mainWindow`
- `ok`

## 9.6.1 Description détaillée

Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB.

### Paramètres

<i>parent</i>	un widget
<i>workdir</i>	un répertoire cible sur les baladeurs

Définition à la ligne 37 du fichier `copyToDialog1.py`.

## 9.6.2 Documentation des fonctions membres

### 9.6.2.1 `def src.copyToDialog1.copyToDialog1.cancel ( self )`

L'action provoquée par le bouton d'échappement : fermeture du dialogue.

Définition à la ligne 74 du fichier `copyToDialog1.py`.

### 9.6.2.2 `def src.copyToDialog1.copyToDialog1.cd ( self, index )`

Change le répertoire courant si possible.

## Paramètres

<code>ev</code>	un évènement
-----------------	--------------

Définition à la ligne 112 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.6.2.3 `def src.copyToDialog1.copyToDialog1.changeWd ( self )`**

changement du répertoire de travail

Définition à la ligne 66 du fichier `copyToDialog1.py`.

**9.6.2.4 `def src.copyToDialog1.copyToDialog1.cont ( self )`**

L'action provoquée par le bouton de continuation : fermeture du dialogue et `self.ok` devient vrai.

Définition à la ligne 82 du fichier `copyToDialog1.py`.

**9.6.2.5 `def src.copyToDialog1.copyToDialog1.displaySize ( self )`**

Affiche la taille de la sélection courante.

Définition à la ligne 163 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.6.2.6 `def src.copyToDialog1.copyToDialog1.remove ( self )`**

Supprime le répertoire ou le fichier sélectionné dans la liste de sélections.

Définition à la ligne 187 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :

**9.6.2.7 `def src.copyToDialog1.copyToDialog1.select ( self )`**

Ajoute le répertoire ou le fichier sélectionné dans le navigateur de fichiers à la liste de sélections.

Définition à la ligne 143 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.6.2.8 `def src.copyToDialog1.copyToDialog1.selectedList ( self )`

Renvoie une liste de répertoires et de fichiers qui ont été sélectionnés pour la copie sur clé USB.

##### Renvoie

une liste de QStrings

Définition à la ligne 203 du fichier `copyToDialog1.py`.

Voici le graphe des appelants de cette fonction :

#### 9.6.2.9 `def src.copyToDialog1.copyToDialog1.setFromListeDir ( self, directory )`

Choisit un répertoire pour la liste source.

##### Paramètres

<i>directory</i>	une instance de QDir
------------------	----------------------

Définition à la ligne 101 du fichier `copyToDialog1.py`.

Voici le graphe des appelants de cette fonction :

#### 9.6.2.10 `def src.copyToDialog1.copyToDialog1.setupFromListe ( self )`

Met en place un visionneur de fichiers dans la liste source.

Définition à la ligne 90 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.6.2.11 `def src.copyToDialog1.copyToDialog1.setupToListe ( self )`

Met en place un visionneur de fichiers pour les fichiers reçus.

Définition à la ligne 124 du fichier `copyToDialog1.py`.

### 9.6.3 Documentation des données membres

#### 9.6.3.1 `src : :copyToDialog1.copyToDialog1 : :mainWindow`

Définition à la ligne 54 du fichier `copyToDialog1.py`.

#### 9.6.3.2 `src : :copyToDialog1.copyToDialog1 : :ok`

Définition à la ligne 1 du fichier `copyToDialog1.py`.

La documentation de cette classe a été générée à partir des fichiers suivants :



– [src/copyToDialog1.py](#)

## 9.7 Référence de la classe src.mainWindow.DiskSizeDelegate

Grappe d'héritage de src.mainWindow.DiskSizeDelegate :

Grappe de collaboration de src.mainWindow.DiskSizeDelegate :

### Fonctions membres publiques

– def [\\_\\_init\\_\\_](#)  
– def [paint](#)  
– def [val2txt](#)

#### 9.7.1 Description détaillée

Définition à la ligne 604 du fichier mainWindow.py.

#### 9.7.2 Documentation des constructeurs et destructeur

9.7.2.1 def src.mainWindow.DiskSizeDelegate.\_\_init\_\_( self, parent )

Définition à la ligne 605 du fichier mainWindow.py.

#### 9.7.3 Documentation des fonctions membres

9.7.3.1 def src.mainWindow.DiskSizeDelegate.paint( self, painter, option, index )

Définition à la ligne 609 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

9.7.3.2 def src.mainWindow.DiskSizeDelegate.val2txt( self, val )

##### Renvoie

a string with a value with unit K, M, or G

Définition à la ligne 630 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/mainWindow.py](#)

## 9.8 Référence de la classe src.help.helpWindow

Graphe d'héritage de src.help.helpWindow :

Graphe de collaboration de src.help.helpWindow :

### Fonctions membres publiques

- def `__init__`  
*Le constructeur.*
- def `loadBrowsers`  
*met en place les textes dans les afficheurs, en fonction de la locale.*

### Attributs publics

- `ui`

#### 9.8.1 Description détaillée

Définition à la ligne 30 du fichier help.py.

#### 9.8.2 Documentation des constructeurs et destructeur

9.8.2.1 `def src.help.helpWindow.__init__( self, parent=None )`

Le constructeur.

Définition à la ligne 35 du fichier help.py.

#### 9.8.3 Documentation des fonctions membres

9.8.3.1 `def src.help.helpWindow.loadBrowsers( self, dir, locale )`

met en place les textes dans les afficheurs, en fonction de la locale.

le répertoire où sont les textes au format HTML est **dir**.

##### Paramètres

<i>dir</i>	le répertoire où sont les fichiers HTML
<i>locale</i>	la langue choisie

Définition à la ligne 50 du fichier help.py.

#### 9.8.4 Documentation des données membres

## 9.8.4.1 src : :help.helpWindow : :ui

Définition à la ligne 35 du fichier help.py.

La documentation de cette classe a été générée à partir du fichier suivant :

- src/[help.py](#)

## 9.9 Référence de la classe src.mainWindow.mainWindow

Graphe d'héritage de src.mainWindow.mainWindow :

Graphe de collaboration de src.mainWindow.mainWindow :

## Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)  
*Le constructeur.*
- def [showEvent](#)  
*modification du comportement du widget original, pour démarrer le timer et les vérifications de baladeurs après construction de la fenêtre seulement*
- def [applyPreferences](#)  
*Applique les préférences et les options de ligne de commande.*
- def [changeWd](#)  
*change le répertoire par défaut contenant les fichiers de travail*
- def [tableClicked](#)  
*fonction de rappel pour un double clic sur un élément de la table*
- def [diskSizeData](#)
- def [diskFromTableRow](#)  
*trouve le disque qui correspond à une ligne du tableau*
- def [editOwner](#)  
*Édition du propriétaire d'une clé.*
- def [updateButtons](#)  
*Désactive ou active les flèches selon que l'option correspondante est possible ou non.*
- def [preference](#)  
*lance le dialogue des préférences*
- def [delFiles](#)  
*Lance l'action de supprimer des fichiers ou des répertoires dans les clés USB.*
- def [copyTo](#)  
*Lance l'action de copier vers les clés USB.*
- def [copyFrom](#)  
*Lance l'action de copier depuis les clés USB.*
- def [help](#)  
*Affiche le widget d'aide.*
- def [umount](#)  
*Démonte et détache les clés USB affichées.*
- def [connectTableModel](#)  
*Connecte le modèle de table à la table.*
- def [checkDisks](#)  
*fonction relancée périodiquement pour vérifier s'il y a un changement dans le baladeurs, et signaler dans le tableau les threads en cours.*
- def [sameDiskData](#)
- def [flashLCD](#)  
*change le style de l'afficheur LCD pendant une fraction de seconde*
- def [normalLCD](#)  
*remet le style par défaut pour l'afficheur LCD*

### Attributs publics

- locale
- ui
- t
- opts
- timer
- oldThreads
- flashTimer
- workdir
- checkable
- header
- visibleheader
- tm

#### 9.9.1 Description détaillée

Définition à la ligne 63 du fichier mainWindow.py.

#### 9.9.2 Documentation des constructeurs et destructeur

9.9.2.1 `def src.mainWindow.mainWindow.__init__( self, parent, opts, locale = "fr_FR" )`

Le constructeur.

##### Paramètres

<i>parent</i>	un QWidget
<i>opts</i>	une liste d'options extraite à l'aide de getopt
<i>locale</i>	la langue de l'application

Définition à la ligne 71 du fichier mainWindow.py.

#### 9.9.3 Documentation des fonctions membres

9.9.3.1 `def src.mainWindow.mainWindow.applyPreferences ( self )`

Applique les préférences et les options de ligne de commande.

Définition à la ligne 113 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

9.9.3.2 `def src.mainWindow.mainWindow.changeWd ( self, newDir )`

change le répertoire par défaut contenant les fichiers de travail

##### Paramètres

<i>newDir</i>	le nouveau nom de répertoire
---------------	------------------------------

Définition à la ligne 132 du fichier `mainWindow.py`.

#### 9.9.3.3 `def src.mainWindow.mainWindow.checkDisks ( self, force = False )`

fonction relancée périodiquement pour vérifier s'il y a un changement dans le baladeurs, et signaler dans le tableau les threads en cours.

Le tableau est complètement régénéré à chaque fois, ce qui n'est pas toujours souhaitable. À la fin de chaque vérification, un court flash est déclenché sur l'afficheur de nombre de baladeurs connectés et sa valeur est mise à jour.

##### Paramètres

<i>force</i>	pour forcer une mise à jour du tableau
--------------	--

Définition à la ligne 413 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

#### 9.9.3.4 `def src.mainWindow.mainWindow.connectTableModel ( self, data )`

Connecte le modèle de table à la table.

##### Paramètres

<i>data</i>	les données de la table
-------------	-------------------------

Définition à la ligne 385 du fichier `mainWindow.py`.

Voici le graphe des appelants de cette fonction :

#### 9.9.3.5 `def src.mainWindow.mainWindow.copyFrom ( self )`

Lance l'action de copier depuis les clés USB.

Définition à la ligne 293 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.9.3.6 `def src.mainWindow.mainWindow.copyTo ( self )`

Lance l'action de copier vers les clés USB.

Définition à la ligne 274 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.9.3.7 `def src.mainWindow.mainWindow.delFiles ( self )`

Lance l'action de supprimer des fichiers ou des répertoires dans les clés USB.

Définition à la ligne 249 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

**9.9.3.8** `def src.mainWindow.mainWindow.diskFromTableRow ( self, ligne )`

trouve le disque qui correspond à une ligne du tableau

#### Paramètres

<i>ligne</i>	la ligne du tableau qui concerne une clé
--------------	--

#### Renvoie

le disque correspondant à la ligne de tableau (type uDisk)

Définition à la ligne 189 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

**9.9.3.9** `def src.mainWindow.mainWindow.diskSizeData ( self, rowOrDev )`

#### Paramètres

<i>rowOrDev</i>	a row number in the tableView, or a device string
-----------------	---

#### Renvoie

a tuple dev,total,used,remain,pcent,path for the disk in the given row of the tableView (the tuple comes from the command df)

Définition à la ligne 172 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

**9.9.3.10** `def src.mainWindow.mainWindow.editOwner ( self, ligne )`

Édition du propriétaire d'une clé.

#### Paramètres

<i>ligne</i>	la ligne du tableau qui concerne une clé
--------------	--

Définition à la ligne 206 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.9.3.11** `def src.mainWindow.mainWindow.flashLCD ( self )`

change le style de l'afficheur LCD pendant une fraction de seconde

Définition à la ligne 437 du fichier `mainWindow.py`.

Voici le graphe des appelants de cette fonction :

**9.9.3.12** `def src.mainWindow.mainWindow.help ( self )`

Affiche le widget d'aide.

Définition à la ligne 342 du fichier `mainWindow.py`.

**9.9.3.13** `def src.mainWindow.mainWindow.normalLCD ( self )`

remet le style par défaut pour l'afficheur LCD

Définition à la ligne 445 du fichier `mainWindow.py`.

**9.9.3.14** `def src.mainWindow.mainWindow.preference ( self )`

lance le dialogue des préférences

Définition à la ligne 235 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :

**9.9.3.15** `def src.mainWindow.mainWindow.sameDiskData ( self, one, two )`

Renvoie

True si les ensembles de `uniqueId` de `one` et `two` sont identiques, et si les informations sur les threads n'ont pas varié

Définition à la ligne 430 du fichier `mainWindow.py`.

Voici le graphe des appelants de cette fonction :

**9.9.3.16** `def src.mainWindow.mainWindow.showEvent ( self, ev )`

modification du comportement du widget original, pour démarrer le timer et les vérifications de baladeurs après construction de la fenêtre seulement

Définition à la ligne 103 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :

**9.9.3.17** `def src.mainWindow.mainWindow.tableClicked ( self, idx )`

fonction de rappel pour un double clic sur un élément de la table

## Paramètres

<i>idx</i>	un QModelIndex
------------	----------------

Définition à la ligne 141 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

#### 9.9.3.18 `def src.mainWindow.mainWindow.umount ( self )`

Démonte et détache les clés USB affichées.

Définition à la ligne 352 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

#### 9.9.3.19 `def src.mainWindow.mainWindow.updateButtons ( self )`

Désactive ou active les flèches selon que l'option correspondante est possible ou non.

Pour les flèches : ça aurait du sens de préparer une opération de copie avant même de brancher des clés, donc on les active. Par contre démonter les clés quand elles sont absentes ça n'a pas d'utilité.

Définition à la ligne 221 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

### 9.9.4 Documentation des données membres

#### 9.9.4.1 `src : :mainWindow.mainWindow : :checkable`

Définition à la ligne 113 du fichier mainWindow.py.

#### 9.9.4.2 `src : :mainWindow.mainWindow : :flashTimer`

Définition à la ligne 71 du fichier mainWindow.py.

#### 9.9.4.3 `src : :mainWindow.mainWindow : :header`

Définition à la ligne 113 du fichier mainWindow.py.

#### 9.9.4.4 `src : :mainWindow.mainWindow : :locale`

Définition à la ligne 71 du fichier mainWindow.py.



**9.9.4.5 src : :mainWindow.mainWindow : :oldThreads**

Définition à la ligne 71 du fichier mainWindow.py.

**9.9.4.6 src : :mainWindow.mainWindow : :opts**

Définition à la ligne 71 du fichier mainWindow.py.

**9.9.4.7 src : :mainWindow.mainWindow : :t**

Définition à la ligne 71 du fichier mainWindow.py.

**9.9.4.8 src : :mainWindow.mainWindow : :timer**

Définition à la ligne 71 du fichier mainWindow.py.

**9.9.4.9 src : :mainWindow.mainWindow : :tm**

Définition à la ligne 385 du fichier mainWindow.py.

**9.9.4.10 src : :mainWindow.mainWindow : :ui**

Définition à la ligne 71 du fichier mainWindow.py.

**9.9.4.11 src : :mainWindow.mainWindow : :visibleheader**

Définition à la ligne 385 du fichier mainWindow.py.

**9.9.4.12 src : :mainWindow.mainWindow : :workdir**

Définition à la ligne 113 du fichier mainWindow.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/mainWindow.py](#)

**9.10 Référence de la classe src.diskFull.mainWindow**

Graphe d'héritage de src.diskFull.mainWindow :

Graphe de collaboration de src.diskFull.mainWindow :

## Fonctions membres publiques

- def `__init__`  
*Le constructeur.*

## Attributs publics

- `ui`
- `v`
- `total`
- `used`

### 9.10.1 Description détaillée

Définition à la ligne 29 du fichier `diskFull.py`.

### 9.10.2 Documentation des constructeurs et destructeur

9.10.2.1 `def src.diskFull.mainWindow.__init__( self, parent, percent, total=0, used=0, title="Disk" )`

Le constructeur.

#### Paramètres

<i>parent</i>	un QWidget
<i>percent</i>	un pourcentage de remplissage de disque
<i>total</i>	place totale en kilo-octets
<i>used</i>	place utilisée en kilo-octets
<i>title</i>	le titre pour la fenêtre

Définition à la ligne 39 du fichier `diskFull.py`.

### 9.10.3 Documentation des données membres

9.10.3.1 `src : :diskFull.mainWindow : :total`

Définition à la ligne 39 du fichier `diskFull.py`.

9.10.3.2 `src : :diskFull.mainWindow : :ui`

Définition à la ligne 39 du fichier `diskFull.py`.

9.10.3.3 `src : :diskFull.mainWindow : :used`

Définition à la ligne 39 du fichier `diskFull.py`.

## 9.10.3.4 src : :diskFull.mainWindow : :v

Définition à la ligne 39 du fichier diskFull.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– src/diskFull.py

## 9.11 Référence de la classe src.mytextbrowser.myTextBrowser

Une classe qui ouvre Firefox quand on clique sur un lien externe.

Graphes d'héritage de src.mytextbrowser.myTextBrowser :

Graphes de collaboration de src.mytextbrowser.myTextBrowser :

### Fonctions membres publiques

- def [setSource](#)  
*lance Firefox en tâche de fond.*
- def [setHtml](#)  
*lien vers la méthode setSource originale*

#### 9.11.1 Description détaillée

Une classe qui ouvre Firefox quand on clique sur un lien externe.

Définition à la ligne 34 du fichier mytextbrowser.py.

#### 9.11.2 Documentation des fonctions membres

##### 9.11.2.1 def src.mytextbrowser.myTextBrowser.setHtml ( self, url )

lien vers la méthode setSource originale

##### Paramètres

<i>url</i>	l'adresse à ouvrir.
------------	---------------------

Définition à la ligne 48 du fichier mytextbrowser.py.

##### 9.11.2.2 def src.mytextbrowser.myTextBrowser.setSource ( self, url )

lance Firefox en tâche de fond.

##### Paramètres

<i>url</i>	l'adresse à ouvrir.
------------	---------------------

Définition à la ligne 40 du fichier mytextbrowser.py.

La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mytextbrowser.py](#)

## 9.12 Référence de la classe `src.notification.Notification`

Une classe pour afficher des notifications à l'écran.

### Fonctions membres publiques

- `def __init__`  
*Le constructeur.*
- `def notify`

### Attributs publics

- `app_name`
- `replaces_id`
- `app_icon`
- `summary`
- `body`
- `actions`
- `hints`
- `expire_timeout`
- `interface`

#### 9.12.1 Description détaillée

Une classe pour afficher des notifications à l'écran.

Doit fonctionner avec tous les gestionnaires de bureau qui adhèrent aux standards de freedesktop.org. Cette classe est basée sur la documentation disponible à <http://www.galago-project.org/specs/notification/0.9/x408.-html>

Définition à la ligne 37 du fichier `notification.py`.

#### 9.12.2 Documentation des constructeurs et destructeur

9.12.2.1 `def src.notification.Notification.__init__( self, app_name = "", replaces_id = 0, app_icon = "", summary = "", body = "", actions = [], hints = {}, expire_timeout = 1000 )`

Le constructeur.

##### Paramètres

<code>app_name</code>	nom d'une application, valeur par défaut = ""
<code>replaces_id</code>	identifiant d'une notification à remplacer valeur par défaut=0
<code>app_icon</code>	nom d'un fichier servant pour l'icône valeur par défaut=""

<i>summary</i>	description brève de la notification valeur par défaut = ""
<i>body</i>	le texte de la notification, valeur par défaut = ""
<i>actions</i>	une liste de paires représentant des actions, valeur par défaut = []
<i>hints</i>	un dictionnaire de suggestions, valeur par défaut = {},
<i>expire_ - timeout</i>	durée maximale d'affichage en millisecondes, valeur par défaut = 1000

Définition à la ligne 51 du fichier `notification.py`.

### 9.12.3 Documentation des fonctions membres

#### 9.12.3.1 `def src.notification.Notification.notify ( self )`

Définition à la ligne 70 du fichier `notification.py`.

### 9.12.4 Documentation des données membres

#### 9.12.4.1 `src : :notification.Notification : :actions`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.2 `src : :notification.Notification : :app_icon`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.3 `src : :notification.Notification : :app_name`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.4 `src : :notification.Notification : :body`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.5 `src : :notification.Notification : :expire_timeout`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.6 `src : :notification.Notification : :hints`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.7 `src : :notification.Notification : :interface`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.8 `src : :notification.Notification : :replaces_id`

Définition à la ligne 51 du fichier `notification.py`.

#### 9.12.4.9 `src : :notification.Notification : :summary`

Définition à la ligne 51 du fichier `notification.py`.

La documentation de cette classe a été générée à partir du fichier suivant :

– `src/notification.py`

### 9.13 Référence de la classe `src.preferences.preferenceWindow`

Graphe d'héritage de `src.preferences.preferenceWindow` :

Graphe de collaboration de `src.preferences.preferenceWindow` :

#### Fonctions membres publiques

- `def __init__`  
*Le constructeur.*
- `def values`
- `def setValues`  
*Met en place les préférences dans le dialogue.*

#### Attributs publics

- `ui`

#### 9.13.1 Description détaillée

Définition à la ligne 29 du fichier `preferences.py`.

#### 9.13.2 Documentation des constructeurs et destructeur

##### 9.13.2.1 `def src.preferences.preferenceWindow.__init__( self, parent = None )`

Le constructeur.

Définition à la ligne 34 du fichier `preferences.py`.

### 9.13.3 Documentation des fonctions membres

9.13.3.1 `def src.preferences.preferenceWindow.setValues ( self, prefs )`

Met en place les préférences dans le dialogue.

Paramètres

<code>prefs</code>	un dictionnaire de préférences
--------------------	--------------------------------

Définition à la ligne 55 du fichier preferences.py.

9.13.3.2 `def src.preferences.preferenceWindow.values ( self )`

Renvoie

un dictionnaire de préférences

Définition à la ligne 44 du fichier preferences.py.

### 9.13.4 Documentation des données membres

9.13.4.1 `src : :preferences.preferenceWindow : :ui`

Définition à la ligne 34 du fichier preferences.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/preferences.py](#)

## 9.14 Référence de la classe QAbstractTableModel

Graphe d'héritage de QAbstractTableModel :

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/mainWindow.py](#)

## 9.15 Référence de la classe QDialog

Graphe d'héritage de QDialog :

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/copyToDialog1.py](#)

## 9.16 Référence de la classe QMainWindow

Graphe d'héritage de QMainWindow :

La documentation de cette classe a été générée à partir du fichier suivant :

– src/[mainWindow.py](#)

## 9.17 Référence de la classe QObject

Graphe d'héritage de QObject :

La documentation de cette classe a été générée à partir du fichier suivant :

– src/[ownedUsbDisk.py](#)

## 9.18 Référence de la classe QStyledItemDelegate

Graphe d'héritage de QStyledItemDelegate :

La documentation de cette classe a été générée à partir du fichier suivant :

– src/[mainWindow.py](#)

## 9.19 Référence de la classe QTextBrowser

Graphe d'héritage de QTextBrowser :

La documentation de cette classe a été générée à partir du fichier suivant :

– src/[mytextbrowser.py](#)

## 9.20 Référence de la classe Thread

Graphe d'héritage de Thread :

La documentation de cette classe a été générée à partir du fichier suivant :

– src/[usbThread.py](#)

## 9.21 Référence de la classe src.usbThread.threadCopyFromUSB

Classe pour les threads copiant depuis les clés USB.

Graphe d'héritage de src.usbThread.threadCopyFromUSB :

Graphe de collaboration de src.usbThread.threadCopyFromUSB :

### Fonctions membres publiques

– def [\\_\\_init\\_\\_](#)

*Constructeur Crée un thread pour copier une liste de fichiers depuis une clé USB vers un répertoire de disque.*



– `def toDo`

*Copie une liste de fichiers d'une clé USB sous un répertoire donné.*

### Attributs publics

– `rootPath`  
– `cmd`

#### 9.21.1 Description détaillée

Classe pour les threads copiant depuis les clés USB.

Définition à la ligne 259 du fichier `usbThread.py`.

#### 9.21.2 Documentation des constructeurs et destructeur

9.21.2.1 `def src.usbThread.threadCopyFromUSB.__init__( self, ud, fileList, subdir  
= " . ", dest = "/tmp", rootPath = "/", logfile = "/dev/null" )`

Constructeur Crée un thread pour copier une liste de fichiers depuis une clé USB vers un répertoire de disque.

##### Paramètres

<code>ud</code>	l'instance <code>uDisk</code> correspondant à une partition de clé USB
<code>fileList</code>	la liste des fichiers à copier
<code>subdir</code>	le sous-répertoire de la clé USB d'où faire la copie
<code>dest</code>	un répertoire de destination
<code>logfile</code>	un fichier de journalisation, <code>/dev/null</code> par défaut

Définition à la ligne 271 du fichier `usbThread.py`.

#### 9.21.3 Documentation des fonctions membres

9.21.3.1 `def src.usbThread.threadCopyFromUSB.toDo( self, ud, fileList, subdir,  
dest, logfile )`

Copie une liste de fichiers d'une clé USB sous un répertoire donné.

À chaque fichier ou répertoire copié, une ligne est journalisée dans le fichier de journal de l'application.

##### Paramètres

<code>ud</code>	l'instance <code>uDisk</code> correspondant à une partition de clé USB
<code>fileList</code>	la liste des fichiers à copier
<code>dest</code>	un répertoire de destination
<code>logfile</code>	un fichier de journalisation
<code>subdir</code>	le sous-répertoire de la clé USB où faire la copie

Réimplémentée à partir de [src.usbThread.abstractThreadUSB](#).

Définition à la ligne 289 du fichier usbThread.py.

#### 9.21.4 Documentation des données membres

##### 9.21.4.1 `src :usbThread.threadCopyFromUSB : :cmd`

Réimplémentée à partir de [src.usbThread.abstractThreadUSB](#).

Définition à la ligne 271 du fichier usbThread.py.

##### 9.21.4.2 `src :usbThread.threadCopyFromUSB : :rootPath`

Définition à la ligne 271 du fichier usbThread.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– `src/usbThread.py`

## 9.22 Référence de la classe `src.usbThread.threadCopyToUSB`

Classe pour les threads copiant vers les clés USB.

Graphe d'héritage de `src.usbThread.threadCopyToUSB` :

Graphe de collaboration de `src.usbThread.threadCopyToUSB` :

### Fonctions membres publiques

- `def __init__`  
*Constructeur Crée un thread pour copier une liste de fichiers vers une clé USB.*
- `def threadType`
- `def toDo`  
*Copie une liste de fichiers vers une clé USB sous un répertoire donné.*

### Attributs publics

- `cmd`

#### 9.22.1 Description détaillée

Classe pour les threads copiant vers les clés USB.

Définition à la ligne 208 du fichier usbThread.py.

#### 9.22.2 Documentation des constructeurs et destructeur

9.22.2.1 `def src.usbThread.threadCopyToUSB.__init__( self, ud, fileList, subdir, logfile = "/dev/null" )`

Constructeur Crée un thread pour copier une liste de fichiers vers une clé USB.

#### Paramètres

<i>ud</i>	l'instance uDisk correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à copier
<i>subdir</i>	le sous-répertoire de la clé USB où faire la copie
<i>logfile</i>	un fichier de journalisation, /dev/null par défaut

Définition à la ligne 218 du fichier `usbThread.py`.

### 9.22.3 Documentation des fonctions membres

9.22.3.1 `def src.usbThread.threadCopyToUSB.threadType( self )`

#### Renvoie

une chaîne courte qui informe sur le type de thread

Réimplémentée à partir de [src.usbThread.abstractThreadUSB](#).

Définition à la ligne 226 du fichier `usbThread.py`.

9.22.3.2 `def src.usbThread.threadCopyToUSB.todo( self, ud, fileList, subdir, dest, logfile )`

Copie une liste de fichiers vers une clé USB sous un répertoire donné.

Ce répertoire est composé de `ud.visibleDir()` joint au sous-répertoire `subdir`. À chaque fichier ou répertoire copié, une ligne est journalisée dans le fichier de journal de l'application.

#### Paramètres

<i>ud</i>	l'instance uDisk correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à copier
<i>logfile</i>	un fichier de journalisation
<i>subdir</i>	le sous-répertoire de la clé USB où faire la copie

Réimplémentée à partir de [src.usbThread.abstractThreadUSB](#).

Définition à la ligne 241 du fichier `usbThread.py`.

### 9.22.4 Documentation des données membres

9.22.4.1 `src : :usbThread.threadCopyToUSB : :cmd`

Réimplémentée à partir de [src.usbThread.abstractThreadUSB](#).

Définition à la ligne 218 du fichier usbThread.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– src/[usbThread.py](#)

## 9.23 Référence de la classe src.usbThread.threadDeleteInUSB

Classe pour les threads effaçant des sous-arbres dans les clés USB.

Graphes d'héritage de src.usbThread.threadDeleteInUSB :

Graphes de collaboration de src.usbThread.threadDeleteInUSB :

### Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)  
*Constructeur Crée un thread pour supprimer une liste de fichiers dans une clé USB.*
- def [todo](#)  
*Supprime une liste de fichiers dans une clé USB.*

### Attributs publics

- [cmd](#)

#### 9.23.1 Description détaillée

Classe pour les threads effaçant des sous-arbres dans les clés USB.

Définition à la ligne 309 du fichier usbThread.py.

#### 9.23.2 Documentation des constructeurs et destructeur

**9.23.2.1** `def src.usbThread.threadDeleteInUSB.__init__( self, ud, fileList, subdir, logfile = "/dev/null" )`

Constructeur Crée un thread pour supprimer une liste de fichiers dans une clé USB.

##### Paramètres

<i>ud</i>	l'instance uDisk correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à supprimer
<i>subdir</i>	le sous-répertoire de la clé USB où faire les suppressions
<i>logfile</i>	un fichier de journalisation, /dev/null par défaut

Définition à la ligne 319 du fichier usbThread.py.

#### 9.23.3 Documentation des fonctions membres

9.23.3.1 `def src.usbThread.threadDeleteInUSB.toDo ( self, ud, fileList, subdir, dest, logfile )`

Supprime une liste de fichiers dans une clé USB.

La liste est prise sous un répertoire donné. Le répertoire visible qui dépend du constructeur d ela clé est pris en compte. À chaque fichier ou répertoire supprimé, une ligne est journalisée dans le fichier de journal de l'application.

#### Paramètres

<i>l'instance</i>	uDisk correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à copier
<i>dest</i>	un répertoire de destination
<i>logfile</i>	un fichier de journalisation
<i>subdir</i>	le sous-répertoire de la clé USB où faire la copie

Réimplémentée à partir de [src.usbThread.abstractThreadUSB](#).

Définition à la ligne 336 du fichier `usbThread.py`.

#### 9.23.4 Documentation des données membres

9.23.4.1 `src : :usbThread.threadDeleteInUSB : :cmd`

Réimplémentée à partir de [src.usbThread.abstractThreadUSB](#).

Définition à la ligne 319 du fichier `usbThread.py`.

La documentation de cette classe a été générée à partir du fichier suivant :

– `src/usbThread.py`

## 9.24 Référence de la classe `src.usbThread.ThreadRegister`

Une classe pour tenir un registre des threads concernant les baladeurs.

#### Fonctions membres publiques

- `def __init__`  
*Le constructeur met en place un dictionnaire.*
- `def __str__`
- `def push`
- `def pop`
- `def busy`  
*Indique si le disque est occupé par des threads.*
- `def threadSet`  
*renvoie l'ensemble des threads actifs*

#### Attributs publics

- `dico`

### 9.24.1 Description détaillée

Une classe pour tenir un registre des threads concernant les baladeurs.

Définition à la ligne 32 du fichier usbThread.py.

### 9.24.2 Documentation des constructeurs et destructeur

#### 9.24.2.1 `def src.usbThread.ThreadRegister.__init__( self )`

Le constructeur met en place un dictionnaire.

Définition à la ligne 38 du fichier usbThread.py.

### 9.24.3 Documentation des fonctions membres

#### 9.24.3.1 `def src.usbThread.ThreadRegister.__str__( self )`

Définition à la ligne 41 du fichier usbThread.py.

#### 9.24.3.2 `def src.usbThread.ThreadRegister.busy( self, owner )`

Indique si le disque est occupé par des threads.

##### Paramètres

<i>owner</i>	le propriétaire du disque
--------------	---------------------------

##### Renvoie

les données associées par le dictionnaire

Définition à la ligne 71 du fichier usbThread.py.

#### 9.24.3.3 `def src.usbThread.ThreadRegister.pop( self, ud, thread )`

##### Paramètres

<i>ud</i>	un disque
<i>thread</i>	un thread Dépile un thread pour le baladeur ud

Définition à la ligne 62 du fichier usbThread.py.

#### 9.24.3.4 `def src.usbThread.ThreadRegister.push( self, ud, thread )`

##### Paramètres

<i>ud</i>	un disque
<i>thread</i>	un thread Empile un thread pour le baladeur ud

Définition à la ligne 50 du fichier usbThread.py.

9.24.3.5 `def src.usbThread.ThreadRegister.threadSet ( self )`

renvoie l'ensemble des threads actifs

Définition à la ligne 80 du fichier usbThread.py.

## 9.24.4 Documentation des données membres

9.24.4.1 `src : :usbThread.ThreadRegister : :dico`

Définition à la ligne 38 du fichier usbThread.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/usbThread.py](#)

## 9.25 Référence de la classe src.usbDisk.uDisk

une classe pour représenter un disque ou une partition.

Graphe d'héritage de src.usbDisk.uDisk :

Graphe de collaboration de src.usbDisk.uDisk :

### Fonctions membres publiques

- `def __init__`  
*Le constructeur.*
- `def getFatUuid`  
*renvoie l'uuid de la première partition FAT après que celle-ci aura été identifiée (utile pour les disques partitionnés)*
- `def uniqueId`  
*renvoie un identifiant unique.*
- `def headers`  
*Méthode statique, pour avoir des titres de colonne.*
- `def devicePropProxy`  
*renvoie un proxy vers un navigateur de propriétés*
- `def isTrue`  
*Renvoie la valeur de vérité d'une propriété*
- `def isUsbDisk`  
*Facilite le réprage des disques USB USB.*
- `def __str__`  
*Fournit une représentation imprimable.*
- `def title`  
*Permet d'obtenir un identifiant unique de disque.*
- `def file`  
*Permet d'accéder à l'instance par un nom de fichier.*
- `def mountPoint`  
*Permet d'accéder à l'instance par un point de montage.*
- `def getProp`

- *Facilite l'accès aux propriétés à l'aide des mots clés du module udisks.*
- def `isDosFat`  
*Permet de reconnaître les partitions DOS-FAT.*
- def `valuableProperties`  
*Facilite l'accès aux propriétés intéressantes d'une instance.*
- def `master`  
*renvoie le chemin du disque, dans le cas où self est une partition*
- def `unNumberProp`  
*retire le numéro des en-têtes pour en faire un nom de propriété valide pour interroger dbus*
- def `__getitem__`  
*Renvoie un élément de listage de données internes au disque.*
- def `showableProp`  
*Renvoie une propriété dans un type "montrable" par QT.*
- def `getFirstFat`  
*Renvoie la première partition VFAT.*
- def `ensureMounted`  
*Permet de s'assurer qu'une partition ou un disque sera bien monté*

### Attributs publics

- `path`
- `device`
- `device_prop`
- `selected`
- `checkable`
- `stickid`
- `uuid`
- `fatuuid`
- `firstFat`

### Attributs publics statiques

- tuple `headers` = `staticmethod(headers)`

#### 9.25.1 Description détaillée

une classe pour représenter un disque ou une partition.

les attributs publics sont :

- **path** le chemin dans le système dbus
- **device** l'objet dbus qui correspond à l'instance
- **device\_prop** un proxy pour questionner cet objet dbus
- **selected** booléen vrai si on doit considérer cette instance comme sélectionnée. Vrai à l'initialisation
- **checkable** booléen vrai si on veut que la sélection puisse être modifiée par l'utilisateur dans l'interface graphique

Définition à la ligne 42 du fichier `usbDisk.py`.

#### 9.25.2 Documentation des constructeurs et destructeur



9.25.2.1 `def src.usbDisk.uDisk.__init__( self, path, bus, checkable = False )`

Le constructeur.

Paramètres

<i>path</i>	un chemin dans le système dbus
<i>bus</i>	un objet <code>dbus.BusSystem</code>
<i>checkable</i>	vrai si on fera usage de <code>self.selected</code>

Réimplémentée dans [src.ownedUsbDisk.uDisk](#).

Définition à la ligne 51 du fichier `usbDisk.py`.

### 9.25.3 Documentation des fonctions membres

9.25.3.1 `def src.usbDisk.uDisk.__getitem__( self, n )`

Renvoie un élément de listage de données internes au disque.

Paramètres

<i>n</i>	un nombre
<i>checkable</i>	vrai si on doit renvoyer une propriété supplémentaire pour <code>n==0</code>

Renvoie

si `checkable` est vrai, un élément si `n>0`, et le drapeau `self.selected` si `n==0` ; sinon un élément de façon ordinaire. Les noms des éléments sont dans la liste `item-Names` utilisée dans la fonction statique `headers`

Réimplémentée dans [src.ownedUsbDisk.uDisk](#).

Définition à la ligne 283 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

9.25.3.2 `def src.usbDisk.uDisk.__str__( self )`

Fournit une représentation imprimable.

Renvoie

une représentation imprimable de l'instance

Définition à la ligne 148 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

9.25.3.3 `def src.usbDisk.uDisk.devicePropProxy( self, bus )`

renvoie un proxy vers un navigateur de propriétés

**Paramètres**

<i>bus</i>	une instace de dbus.SystemBus
------------	-------------------------------

**Renvoie**

l'objet proxy

Définition à la ligne 119 du fichier usbDisk.py.

**9.25.3.4 def src.usbDisk.uDisk.ensureMounted ( self )**

Permet de s'assurer qu'une partition ou un disque sera bien monté

**Renvoie**

le chemin du point de montage

Définition à la ligne 329 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

**9.25.3.5 def src.usbDisk.uDisk.file ( self )**

Permet d'accéder à l'instance par un nom de fichier.

**Renvoie**

un nom valide dans le système de fichiers, pour accéder à l'instance.

Définition à la ligne 165 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

**9.25.3.6 def src.usbDisk.uDisk.getFatUuid ( self )**

renvoie l'uuid de la première partition FAT après que celle-ci aura été identifiée (utile pour les disques partitionnés)

**Renvoie**

un uuid

Définition à la ligne 82 du fichier usbDisk.py.

Voici le graphe des appelants de cette fonction :

**9.25.3.7 def src.usbDisk.uDisk.getFirstFat ( self )**

Renvoie la première partition VFAT.

**Renvoie**

la première partition VFAT ou None s'il n'y en a pas

Définition à la ligne 320 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.25.3.8 `def src.usbDisk.uDisk.getProp ( self, name )`**

Facilite l'accès aux propriétés à l'aide des mots clés du module `udisks`.

**Paramètres**

<i>name</i>	le nom d'une propriété
-------------	------------------------

**Renvoie**

une propriété dbus du disque ou de la partition, sinon None si le nom `name` est illégal

Définition à la ligne 188 du fichier `usbDisk.py`.

Voici le graphe des appelants de cette fonction :

**9.25.3.9 `def src.usbDisk.uDisk.headers ( checkable=False, locale="C" )`**

Méthode statique, pour avoir des titres de colonne.

renvoie des titres pour les items obtenus par `__getitem__`. Le résultat dépend du paramètre `checkable`.

**Paramètres**

<i>checkable</i>	vrai si le premier en-tête correspond à une colonne de cases à cocher
<i>locale</i>	la locale, pour traduire les titres éventuellement. Valeur par défaut : "C"

**Renvoie**

une liste de titres de colonnes

Réimplémentée dans [src.ownedUsbDisk.uDisk](#).

Définition à la ligne 104 du fichier `usbDisk.py`.

**9.25.3.10 `def src.usbDisk.uDisk.isDosFat ( self )`**

Permet de reconnaître les partitions DOS-FAT.

**Renvoie**

vrai dans le cas d'une partition FAT16 ou FAT32

Définition à la ligne 199 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.25.3.11** `def src.usbDisk.uDisk.isTrue ( self, prop, value = None )`

Renvoie la valeur de vérité d'une propriété

**Paramètres**

<i>prop</i>	une propriété
<i>value</i>	

**Renvoie**

vrai si la propriété est vraie (cas où value==None) ou vrai si la propriété a exactement la valeur value.

Définition à la ligne 129 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.25.3.12** `def src.usbDisk.uDisk.isUsbDisk ( self )`

Facilite le réprage des disques USB USB.

**Renvoie**

vrai dans le cas d'un disque USB

Définition à la ligne 140 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.25.3.13** `def src.usbDisk.uDisk.master ( self )`

renvoie le chemin du disque, dans le cas où self est une partition

**Renvoie**

le chemin dbus du disque maître, sinon "/"

Définition à la ligne 257 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

#### 9.25.3.14 `def src.usbDisk.uDisk.mountPoint ( self )`

Permet d'accéder à l'instance par un point de montage.

##### Renvoie

un point de montage, s'il en existe, sinon None

Définition à la ligne 175 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.25.3.15 `def src.usbDisk.uDisk.showableProp ( self, name )`

Renvoie une propriété dans un type "montrable" par QT.

les propriétés que renvoie dbus ont des types inconnus de Qt4, cette fonction les transtype pour que QVariant arrive à les prendre en compte.

##### Paramètres

<i>name</i>	le nom de la propriété
-------------	------------------------

##### Renvoie

une nombre ou une chaîne selon le type de propriété

Définition à la ligne 303 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

#### 9.25.3.16 `def src.usbDisk.uDisk.title ( self )`

Permet d'obtenir un identifiant unique de disque.

##### Renvoie

le chemin dbus de l'instance

Définition à la ligne 156 du fichier `usbDisk.py`.

Voici le graphe des appelants de cette fonction :

#### 9.25.3.17 `def src.usbDisk.uDisk.uniqueId ( self )`

renvoie un identifiant unique.

Dans cette classe, cette fonction est synonyme de `getFatUuid`

**Renvoie**

un identifiant supposément unique

Réimplémentée dans [src.ownedUsbDisk.uDisk](#).

Définition à la ligne 91 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

**9.25.3.18 def src.usbDisk.uDisk.unNumberProp ( self, n )**

retire le numéro des en-têtes pour en faire un nom de propriété valide pour interroger dbus

**Paramètres**

<i>n</i>	un numéro de propriété qui se réfère aux headers
----------	--

**Renvoie**

une propriété renvoyée par dbus, dans un format imprimable

Définition à la ligne 267 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.25.3.19 def src.usbDisk.uDisk.valuableProperties ( self, indent = 4 )**

Facilite l'accès aux propriétés intéressantes d'une instance.

**Renvoie**

une chaîne indentée avec les propriétés intéressantes, une par ligne

Définition à la ligne 212 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.25.4 Documentation des données membres****9.25.4.1 src : :usbDisk.uDisk : :checkable**

Définition à la ligne 51 du fichier usbDisk.py.

**9.25.4.2 src : :usbDisk.uDisk : :device**

Définition à la ligne 51 du fichier usbDisk.py.

**9.25.4.3 `src : :usbDisk.uDisk : :device_prop`**

Définition à la ligne 51 du fichier `usbDisk.py`.

**9.25.4.4 `src : :usbDisk.uDisk : :fatuuid`**

Définition à la ligne 51 du fichier `usbDisk.py`.

**9.25.4.5 `src : :usbDisk.uDisk : :firstFat`**

Définition à la ligne 51 du fichier `usbDisk.py`.

**9.25.4.6 `tuple src : :usbDisk.uDisk : :headers = staticmethod(headers)` `[static]`**

Réimplémentée dans [src.ownedUsbDisk.uDisk](#).

Définition à la ligne 111 du fichier `usbDisk.py`.

**9.25.4.7 `src : :usbDisk.uDisk : :path`**

Définition à la ligne 51 du fichier `usbDisk.py`.

**9.25.4.8 `src : :usbDisk.uDisk : :selected`**

Définition à la ligne 51 du fichier `usbDisk.py`.

**9.25.4.9 `src : :usbDisk.uDisk : :stickid`**

Définition à la ligne 51 du fichier `usbDisk.py`.

**9.25.4.10 `src : :usbDisk.uDisk : :uuid`**

Définition à la ligne 51 du fichier `usbDisk.py`.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/usbDisk.py](#)

## 9.26 Référence de la classe `src.ownedUsbDisk.uDisk`

une classe qui ajoute un nom de propriétaire aux disque USB, et qui en même temps ajoute des particularités selon le nom du vendeur et le modèle.

Graphes d'héritage de `src.ownedUsbDisk.uDisk` :

Graphes de collaboration de `src.ownedUsbDisk.uDisk` :

## Fonctions membres publiques

- def `__init__`
- def `uniqueId`
- def `tattoo`  
*Renvoie un tatouage présent sur la clé, quitte à le créer.*
- def `readQuirks`  
*Lit un dictionnaire indexé par le noms de vendeurs et les noms de modèle pour associer à ces modèles particuliers un répertoire visible.*
- def `visibleDir`  
*Renvoie le répertoire particulier de la partition qui sera visible quand le baladeur est utilisé par son interface utilisateur.*
- def `headers`  
*Méthode statique renvoie des titres pour les items obtenus par `__getitem__` la deuxième colonne sera toujours le propriétaire.*
- def `ownerByDb`  
*renvoie un nom de propriétaire dans tous les cas.*
- def `__getitem__`  
*renvoie un élément de listage de données internes au disque Fait en sorte que la deuxième colonne soit toujours le propriétaire*
- def `ensureOwner`  
*Demande un nom de propriétaire si celui-ci n'est pas encore défini pour cette clé USB.*

## Attributs publics

- `owner`
- `vendor`
- `model`
- `visibleDirs`

## Attributs publics statiques

- tuple `headers` = `staticmethod(headers)`

### 9.26.1 Description détaillée

une classe qui ajoute un nom de propriétaire aux disque USB, et qui en même temps ajoute des particularités selon le nom du vendeur et le modèle.

Définition à la ligne 59 du fichier `ownedUsbDisk.py`.

### 9.26.2 Documentation des constructeurs et destructeur

9.26.2.1 `def src.ownedUsbDisk.uDisk.__init__( self, path, bus, checkable=False )`

#### Paramètres

<code>path</code>	un chemin dans le système dbus
<code>bus</code>	un objet <code>dbus.BusSystem</code>
<code>checkable</code>	vrai si on fera usage de <code>self.selected</code>



Réimplémentée à partir de [src.usbDisk.uDisk](#).

Définition à la ligne 66 du fichier `ownedUsbDisk.py`.

### 9.26.3 Documentation des fonctions membres

#### 9.26.3.1 `def src.ownedUsbDisk.uDisk.__getitem__( self, n )`

renvoie un élément de listage de données internes au disque Fait en sorte que la deuxième colonne soit toujours le propriétaire

##### Paramètres

<i>n</i>	un nombre
<i>checkable</i>	vrai si on doit renvoyer une propriété supplémentaire pour <code>n==0</code>

##### Renvoie

si `n==-1`, renvoie `self` ; si `checkable` est vrai, renvoie un élément si `n>0`, et le drapeau `self.selected` si `n==0` ; sinon un élément de façon ordinaire. Les noms des éléments sont dans la liste `self.itemNames`

Réimplémentée à partir de [src.usbDisk.uDisk](#).

Définition à la ligne 180 du fichier `ownedUsbDisk.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.26.3.2 `def src.ownedUsbDisk.uDisk.ensureOwner ( self )`

Demande un nom de propriétaire si celui-ci n'est pas encore défini pour cette clé USB.

##### Renvoie

un nom de propriétaire si c'est un disque, sinon `None`

Définition à la ligne 210 du fichier `ownedUsbDisk.py`.

Voici le graphe d'appel pour cette fonction :

#### 9.26.3.3 `def src.ownedUsbDisk.uDisk.headers ( checkable=False, locale="C" )`

Méthode statique renvoie des titres pour les items obtenus par `__getitem__` la deuxième colonne sera toujours le propriétaire.

##### Paramètres

<i>checkable</i>	vrai si le premier en-tête correspond à une colonne de cases à cocher
<i>locale</i>	la locale, pour traduire les titres

**Renvoie**

une liste de titres de colonnes

Réimplémentée à partir de [src.usbDisk.uDisk](#).

Définition à la ligne 151 du fichier ownedUsbDisk.py.

**9.26.3.4 def src.ownedUsbDisk.uDisk.ownerByDb ( self )**

renvoie un nom de propriétaire dans tous les cas.

Définition à la ligne 161 du fichier ownedUsbDisk.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.26.3.5 def src.ownedUsbDisk.uDisk.readQuirks ( self )**

Lit un dictionnaire indexé par le noms de vendeurs et les noms de modèle pour associer à ces modèles particuliers un répertoire visible.

voir la fonction visibleDir. Ce dictionnaire est dans le fichier /usr/share/scolasync/marques.py ou dans \${HOME}/.scolasync/marques.py, (sous Linux) cette dernière place étant prépondérante.

Définition à la ligne 119 du fichier ownedUsbDisk.py.

**9.26.3.6 def src.ownedUsbDisk.uDisk.tattoo ( self )**

Renvoie un tatouage présent sur la clé, quitte à le créer.

**Renvoie**

un tatouage, supposément unique.

Définition à la ligne 87 du fichier ownedUsbDisk.py.

Voici le graphe d'appel pour cette fonction :

Voici le graphe des appelants de cette fonction :

**9.26.3.7 def src.ownedUsbDisk.uDisk.uniqueId ( self )****Renvoie**

un identifiant unique, composé du nom du propriétaire suivi du tatouage

Réimplémentée à partir de [src.usbDisk.uDisk](#).

Définition à la ligne 79 du fichier ownedUsbDisk.py.

Voici le graphe d'appel pour cette fonction :

#### 9.26.3.8 def src.ownedUsbDisk.uDisk.visibleDir ( self )

Renvoie le répertoire particulier de la partition qui sera visible quand le baladeur est utilisé par son interface utilisateur.

Ce répertoire peut varier selon les vendeurs et les modèles.

Définition à la ligne 135 du fichier ownedUsbDisk.py.

### 9.26.4 Documentation des données membres

#### 9.26.4.1 tuple src : :ownedUsbDisk.uDisk : :headers = staticmethod(headers) [static]

Réimplémentée à partir de [src.usbDisk.uDisk](#).

Définition à la ligne 202 du fichier ownedUsbDisk.py.

#### 9.26.4.2 src : :ownedUsbDisk.uDisk : :model

Définition à la ligne 66 du fichier ownedUsbDisk.py.

#### 9.26.4.3 src : :ownedUsbDisk.uDisk : :owner

Définition à la ligne 66 du fichier ownedUsbDisk.py.

#### 9.26.4.4 src : :ownedUsbDisk.uDisk : :vendor

Définition à la ligne 66 du fichier ownedUsbDisk.py.

#### 9.26.4.5 src : :ownedUsbDisk.uDisk : :visibleDirs

Définition à la ligne 66 du fichier ownedUsbDisk.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/ownedUsbDisk.py](#)

## 9.27 Référence de la classe src.mainWindow.UsbDiskDelegate

Graphe d'héritage de src.mainWindow.UsbDiskDelegate :

Graphe de collaboration de src.mainWindow.UsbDiskDelegate :

### Fonctions membres publiques

– def [\\_\\_init\\_\\_](#)

– def [paint](#)

### Attributs publics

– [okPixmap](#)  
– [busyPixmap](#)

#### 9.27.1 Description détaillée

Définition à la ligne 582 du fichier mainWindow.py.

#### 9.27.2 Documentation des constructeurs et destructeur

9.27.2.1 `def src.mainWindow.UsbDiskDelegate.__init__( self, parent )`

Définition à la ligne 583 du fichier mainWindow.py.

#### 9.27.3 Documentation des fonctions membres

9.27.3.1 `def src.mainWindow.UsbDiskDelegate.paint( self, painter, option, index )`

Définition à la ligne 588 du fichier mainWindow.py.

#### 9.27.4 Documentation des données membres

9.27.4.1 `src : :mainWindow.UsbDiskDelegate : :busyPixmap`

Définition à la ligne 583 du fichier mainWindow.py.

9.27.4.2 `src : :mainWindow.UsbDiskDelegate : :okPixmap`

Définition à la ligne 583 du fichier mainWindow.py.

La documentation de cette classe a été générée à partir du fichier suivant :

– [src/mainWindow.py](#)

### 9.28 Référence de la classe src.mainWindow.usbTableModel

Un modèle de table pour des séries de clés USB.

Graphe d'héritage de src.mainWindow.usbTableModel :

Graphe de collaboration de src.mainWindow.usbTableModel :

### Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)
- def [rowCount](#)  
un *QModelIndex*
- def [columnCount](#)  
un *QModelIndex*
- def [setData](#)
- def [partition](#)
- def [data](#)
- def [headerData](#)
- def [sort](#)  
Sort table by given column number.

### Attributs publics

- [header](#)
- [donnees](#)
- [checkable](#)
- [pere](#)

#### 9.28.1 Description détaillée

Un modèle de table pour des séries de clés USB.

Définition à la ligne 452 du fichier mainWindow.py.

#### 9.28.2 Documentation des constructeurs et destructeur

9.28.2.1 `def src.mainWindow.usbTableModel.__init__( self, parent=None, header = [], donnees=None, checkable=False )`

##### Paramètres

<i>parent</i>	un <a href="#">QObject</a>
<i>header</i>	les en-têtes de colonnes
<i>donnees</i>	les données
<i>checkable</i>	vrai si la première colonne est composée de boîtes à cocher. Faux par défaut

Définition à la ligne 461 du fichier mainWindow.py.

#### 9.28.3 Documentation des fonctions membres

9.28.3.1 `def src.mainWindow.usbTableModel.columnCount( self, parent )`

un *QModelIndex*

Définition à la ligne 479 du fichier mainWindow.py.

9.28.3.2 `def src.mainWindow.usbTableModel.data ( self, index, role )`

Définition à la ligne 497 du fichier mainWindow.py.

9.28.3.3 `def src.mainWindow.usbTableModel.headerData ( self, section, orientation, role )`

Définition à la ligne 526 du fichier mainWindow.py.

9.28.3.4 `def src.mainWindow.usbTableModel.partition ( self, index )`

#### Paramètres

<i>index</i>	in QModelIndex
--------------	----------------

#### Renvoie

la partition pointée par index

Définition à la ligne 494 du fichier mainWindow.py.

9.28.3.5 `def src.mainWindow.usbTableModel.rowCount ( self, parent )`

un QModelIndex

Définition à la ligne 472 du fichier mainWindow.py.

9.28.3.6 `def src.mainWindow.usbTableModel.setData ( self, index, value, role )`

Définition à la ligne 482 du fichier mainWindow.py.

9.28.3.7 `def src.mainWindow.usbTableModel.sort ( self, Ncol, order = Qt.DescendingOrder )`

Sort table by given column number.

#### Paramètres

<i>Ncol</i>	numéro de la colonne de tri
<i>order</i>	l'ordre de tri, Qt.DescendingOrder par défaut

Définition à la ligne 538 du fichier mainWindow.py.

## 9.28.4 Documentation des données membres

**9.28.4.1 `src : :mainWindow.usbTableModel : :checkable`**

Définition à la ligne 461 du fichier `mainWindow.py`.

**9.28.4.2 `src : :mainWindow.usbTableModel : :donnees`**

Définition à la ligne 461 du fichier `mainWindow.py`.

**9.28.4.3 `src : :mainWindow.usbTableModel : :header`**

Définition à la ligne 461 du fichier `mainWindow.py`.

**9.28.4.4 `src : :mainWindow.usbTableModel : :pere`**

Définition à la ligne 461 du fichier `mainWindow.py`.

La documentation de cette classe a été générée à partir du fichier suivant :

– `src/mainWindow.py`





## Chapitre 10

# Documentation des fichiers

### 10.1 Référence du fichier `src/__init__.py`

#### Paquetages

- namespace `src`

### 10.2 Référence du fichier `src/chooseInSticks.py`

#### Classes

- class `src.chooseInSticks.chooseDialog`  
*Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB.*

#### Paquetages

- namespace `src : :chooseInSticks`

#### Variables

- string `src : :chooseInSticks : :licenceEn`

### 10.3 Référence du fichier `src/copyToDialog1.py`

#### Classes

- class `src.copyToDialog1.copyToDialog1`  
*Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB.*

### Paquetages

- namespace `src : :copyToDialog1`

### Variables

- string `src : :copyToDialog1 : :licenceEn`
- tuple `src : :copyToDialog1.app` = `QApplication(sys.argv)`
- tuple `src : :copyToDialog1.windows` = `copyToDialog1()`

## 10.4 Référence du fichier `src/db.py`

### Paquetages

- namespace `src : :db`

### Fonctions

- def `src : :db.openDb`  
*Ouverture de la base de données de l'application, et création si nécessaire.*
- def `src : :db.checkVersion`  
*Vérifie si la base de données reste compatible.*
- def `src : :db.knowsId`  
*dit si une clé USB est déjà connue*
- def `src : :db.tattooList`  
*Renvoie la liste des tatouages connus de la base de données.*
- def `src : :db.readStudent`  
*renvoie l'étudiant qui possède une clé USB*
- def `src : :db.readPrefs`  
*renvoie les préférences de ScolaSync*
- def `src : :db.setWd`  
*définit le nouveau nom du répertoire de travail préféré.*
- def `src : :db.writeStudent`  
*inscrit un étudiant comme propriétaire d'une clé USB*
- def `src : :db.writePrefs`  
*inscrit les préférences*

### Variables

- dictionary `src : :db : :licence` = {}
- `src : :db.database` = `None`
- `src : :db.cursor` = `None`

## 10.5 Référence du fichier `src/diskFull.py`

### Classes

- class `src.diskFull.mainWindow`

### Paquetages

- namespace `src : :diskFull`

### Fonctions

- def `src : :diskFull.sceneWithUsage`

### Variables

- dictionary `src : :diskFull : :licence = {}`

## 10.6 Référence du fichier src/globaldef.py

### Paquetages

- namespace `src : :globaldef`

### Variables

- string `src : :globaldef : :licenceEn`  
*globaldef.py is part of the package scolasync.*
- string `src : :globaldef.userShareDir = "~/scolasync"`
- string `src : :globaldef.logFileName = "~/scolasync/scolasync.log"`
- string `src : :globaldef.markFileName = "~/scolasync/marques.py"`

## 10.7 Référence du fichier src/help.py

### Classes

- class `src.help.helpWindow`

### Paquetages

- namespace `src : :help`

### Variables

- dictionary `src : :help : :licence = {}`

## 10.8 Référence du fichier src/mainWindow.py

### Classes

- class [src.mainWindow.mainWindow](#)
- class [src.mainWindow.usbTableModel](#)  
*Un modèle de table pour des séries de clés USB.*
- class [src.mainWindow.CheckBoxDelegate](#)
- class [src.mainWindow.UsbDiskDelegate](#)
- class [src.mainWindow.DiskSizeDelegate](#)

### Paquetages

- namespace [src : :mainWindow](#)

### Fonctions

- def [src : :mainWindow.firstDir](#)  
*Renvoie le premier répertoire existant d'une liste de propositions.*
- def [src : :mainWindow.CheckBoxRect](#)

### Variables

- dictionary [src : :mainWindow : :licence](#) = {}

## 10.9 Référence du fichier src/marques.py

### Paquetages

- namespace [src : :marques](#)

## 10.10 Référence du fichier src/mytextbrowser.py

### Classes

- class [src.mytextbrowser.myTextBrowser](#)  
*Une classe qui ouvre Firefox quand on clique sur un lien externe.*

### Paquetages

- namespace [src : :mytextbrowser](#)

### Variables

- dictionary [src : :mytextbrowser : :licence](#) = {}

## 10.11 Référence du fichier src/notification.py

### Classes

- class [src.notification.Notification](#)  
*Une classe pour afficher des notifications à l'écran.*

### Paquetages

- namespace [src](#) : [:notification](#)

### Variables

- dictionary [src](#) : [:notification](#) : [:licence](#) = {}
- tuple [src](#) : [:notification](#).[notif](#)

## 10.12 Référence du fichier src/ownedUsbDisk.py

### Classes

- class [src.ownedUsbDisk.uDisk](#)  
*une classe qui ajoute un nom de propriétaire aux disque USB, et qui en même temps ajoute des particularités selon le nom du vendeur et le modèle.*
- class [src.ownedUsbDisk.Available](#)  
*Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires.*

### Paquetages

- namespace [src](#) : [:ownedUsbDisk](#)

### Fonctions

- def [src](#) : [:ownedUsbDisk](#).[editRecord](#)  
*édition de la base de données.*

### Variables

- dictionary [src](#) : [:ownedUsbDisk](#) : [:licence](#) = {}

## 10.13 Référence du fichier src/preferences.py

### Classes

- class [src.preferences.preferenceWindow](#)

### Paquetages

- namespace `src : :preferences`

### Variables

- dictionary `src : :preferences : :licence = {}`

## 10.14 Référence du fichier `src/scolasync.py`

### Paquetages

- namespace `src : :scolasync`
- namespace `scolasync`
  - Scolasync est un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB.*

### Fonctions

- def `src : :scolasync.usage`
  - affiche le mode d'emploi à la console*
- def `src : :scolasync.run`
  - C'est la fonction principale.*

### Variables

- dictionary `src : :scolasync.licence = {}`
- string `src : :scolasync.licenceEn`
- string `src : :scolasync.licenceFr`

## 10.15 Référence du fichier `src/usbDisk.py`

### Classes

- class `src.usbDisk.uDisk`
  - une classe pour représenter un disque ou une partition.*
- class `src.usbDisk.Available`
  - une classe pour représenter la collection des disques USB connectés*

### Paquetages

- namespace `src : :usbDisk`

### Variables

- dictionary `src : :usbDisk : :licence = {}`
- string `src : :usbDisk.licence_en`

- tuple `src : :usbDisk.machin` = Available()

## 10.16 Référence du fichier src/usbThread.py

### Classes

- class `src.usbThread.ThreadRegister`  
*Une classe pour tenir un registre des threads concernant les baladeurs.*
- class `src.usbThread.abstractThreadUSB`  
*Une classe abstraite Cette classe sert de creuset pour les classe servant aux copies et aux effacement.*
- class `src.usbThread.threadCopyToUSB`  
*Classe pour les threads copiant vers les clés USB.*
- class `src.usbThread.threadCopyFromUSB`  
*Classe pour les threads copiant depuis les clés USB.*
- class `src.usbThread.threadDeleteInUSB`  
*Classe pour les threads effaçant des sous-arbres dans les clés USB.*

### Paquetages

- namespace `src : :usbThread`

### Variables

- string `src : :usbThread : :licenceEn`
- int `src : :usbThread._threadNumber` = 0

## 10.17 Référence du fichier src/version.py

### Paquetages

- namespace `src : :version`

### Fonctions

- def `src : :version.major`
- def `src : :version.minor`
- def `src : :version.version`

### Variables

- dictionary `src : :version : :licence` = {}