

ScolaSync

4.0

Généré par Doxygen 1.8.1.2

Samedi Juin 22 2013 15 :36 :36



# 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	1
1.4	Support	2
1.5	Architecture de ScolaSync	2
<b>2</b>	<b>Index des espaces de nommage</b>	<b>3</b>
2.1	Paquetages	3
<b>3</b>	<b>Index des classes</b>	<b>5</b>
3.1	Hiérarchie des classes	5
<b>4</b>	<b>Index des classes</b>	<b>7</b>
4.1	Liste des classes	7
<b>5</b>	<b>Index des fichiers</b>	<b>9</b>
5.1	Liste des fichiers	9
<b>6</b>	<b>Documentation des espaces de nommage</b>	<b>11</b>
6.1	Référence de l'espace de nommage scolasync	11
6.1.1	Description détaillée	11
6.2	Référence de l'espace de nommage src	11
6.2.1	Documentation des variables	12
6.2.1.1	python3safe	12
6.3	Référence de l'espace de nommage src.checkBoxDialog	12
6.3.1	Documentation des variables	12
6.3.1.1	licenceEn	12
6.3.1.2	python3safe	12
6.4	Référence de l'espace de nommage src.choixEleves	12
6.4.1	Documentation des variables	13
6.4.1.1	app	13
6.4.1.2	d	13

6.4.1.3	i	13
6.4.1.4	licence	13
6.4.1.5	python3safe	13
6.5	Référence de l'espace de nommage src.chooseInSticks	13
6.5.1	Documentation des variables	13
6.5.1.1	licenceEn	13
6.5.1.2	python3safe	14
6.6	Référence de l'espace de nommage src.copyToDialog1	14
6.6.1	Documentation des variables	14
6.6.1.1	app	14
6.6.1.2	licenceEn	14
6.6.1.3	python3safe	15
6.6.1.4	windows	15
6.7	Référence de l'espace de nommage src.db	15
6.7.1	Documentation des fonctions	15
6.7.1.1	checkVersion	15
6.7.1.2	hasStudent	16
6.7.1.3	knowsId	16
6.7.1.4	openDb	16
6.7.1.5	readPrefs	17
6.7.1.6	readStudent	17
6.7.1.7	setWd	17
6.7.1.8	tattooList	18
6.7.1.9	writePrefs	18
6.7.1.10	writeStudent	18
6.7.2	Documentation des variables	18
6.7.2.1	cursor	18
6.7.2.2	database	19
6.7.2.3	licence	19
6.7.2.4	python3safe	19
6.8	Référence de l'espace de nommage src.debug	19
6.8.1	Documentation des fonctions	19
6.8.1.1	button	19
6.8.1.2	listePartitionsCochees	19
6.8.2	Documentation des variables	19
6.8.2.1	licence	20
6.8.2.2	licenceEn	20
6.8.2.3	licenceFr	20
6.9	Référence de l'espace de nommage src.deviceListener	20
6.9.1	Documentation des variables	21

6.9.1.1	licence	21
6.9.1.2	python3safe	21
6.10	Référence de l'espace de nommage src.diskFull	21
6.10.1	Documentation des fonctions	21
6.10.1.1	sceneWithUsage	21
6.10.2	Documentation des variables	21
6.10.2.1	licence	21
6.10.2.2	python3safe	21
6.11	Référence de l'espace de nommage src.essai	21
6.12	Référence de l'espace de nommage src.gestClasse	22
6.12.1	Documentation des variables	22
6.12.1.1	licence	22
6.12.1.2	python3safe	22
6.13	Référence de l'espace de nommage src.gestclassetreeview	22
6.13.1	Documentation des variables	22
6.13.1.1	licence	22
6.13.1.2	python3safe	22
6.14	Référence de l'espace de nommage src.globaldef	23
6.14.1	Documentation des fonctions	23
6.14.1.1	firstdir	23
6.14.2	Documentation des variables	23
6.14.2.1	licenceEn	23
6.14.2.2	logFileName	23
6.14.2.3	markFileName	24
6.14.2.4	python3safe	24
6.14.2.5	userShareDir	24
6.15	Référence de l'espace de nommage src.help	24
6.15.1	Documentation des variables	24
6.15.1.1	licence	24
6.15.1.2	python3safe	24
6.16	Référence de l'espace de nommage src.mainWindow	24
6.16.1	Documentation des fonctions	25
6.16.1.1	CheckBoxRect	25
6.16.1.2	registerCmd	25
6.16.2	Documentation des variables	26
6.16.2.1	activeThreads	26
6.16.2.2	lastCommand	26
6.16.2.3	licence	26
6.16.2.4	pastCommands	26
6.17	Référence de l'espace de nommage src.marques	26

6.18	Référence de l'espace de nommage <code>src.mytextbrowser</code>	26
6.18.1	Documentation des variables	26
6.18.1.1	licence	27
6.18.1.2	python3safe	27
6.19	Référence de l'espace de nommage <code>src.nameAdrive</code>	27
6.19.1	Documentation des variables	27
6.19.1.1	licence	27
6.19.1.2	python3safe	27
6.20	Référence de l'espace de nommage <code>src.notification</code>	27
6.20.1	Documentation des variables	27
6.20.1.1	licence	27
6.20.1.2	notif	28
6.20.1.3	python3safe	28
6.21	Référence de l'espace de nommage <code>src.ownedUsbDisk</code>	28
6.21.1	Documentation des fonctions	28
6.21.1.1	editRecord	28
6.21.1.2	tattooInDir	28
6.21.2	Documentation des variables	29
6.21.2.1	licence	29
6.21.2.2	machin	29
6.21.2.3	python3safe	29
6.22	Référence de l'espace de nommage <code>src.preferences</code>	29
6.22.1	Documentation des variables	29
6.22.1.1	licence	29
6.22.1.2	python3safe	29
6.23	Référence de l'espace de nommage <code>src.scolasync</code>	30
6.23.1	Documentation des fonctions	30
6.23.1.1	run	30
6.23.2	Documentation des variables	30
6.23.2.1	licence	30
6.23.2.2	licenceEn	30
6.23.2.3	licenceFr	30
6.24	Référence de l'espace de nommage <code>src.sconet</code>	31
6.24.1	Documentation des variables	31
6.24.1.1	licence	31
6.24.1.2	python3safe	31
6.24.1.3	s	31
6.25	Référence de l'espace de nommage <code>src.test3</code>	31
6.25.1	Documentation des variables	32
6.25.1.1	files	32

6.25.1.2	module	32
6.25.1.3	moduleName	32
6.25.1.4	notsafe	32
6.25.1.5	pattern	32
6.25.1.6	python3safe	32
6.25.1.7	safe	32
6.26	Référence de l'espace de nommage src.usbDisk	32
6.26.1	Documentation des variables	32
6.26.1.1	dependences	32
6.26.1.2	licence	33
6.26.1.3	licence_en	33
6.26.1.4	machin	33
6.26.1.5	python3safe	33
6.27	Référence de l'espace de nommage src.usbThread	33
6.27.1	Documentation des fonctions	34
6.27.1.1	ensureDirExists	34
6.27.1.2	test_copy2	34
6.27.1.3	test_copytree	34
6.27.2	Documentation des variables	34
6.27.2.1	_threadNumber	34
6.27.2.2	licenceEn	35
6.28	Référence de l'espace de nommage src.version	35
6.28.1	Documentation des fonctions	35
6.28.1.1	major	35
6.28.1.2	minor	36
6.28.1.3	version	36
6.28.2	Documentation des variables	36
6.28.2.1	licence	36
<b>7</b>	<b>Documentation des classes</b>	<b>37</b>
7.1	Référence de la classe src.gestClasse.AbstractGestClasse	37
7.1.1	Description détaillée	37
7.1.2	Documentation des constructeurs et destructeur	37
7.1.2.1	__init__	37
7.1.3	Documentation des fonctions membres	38
7.1.3.1	collectClasses	38
7.1.3.2	elevesDeClasse	38
7.1.3.3	showable_name	38
7.1.3.4	unique_name	38
7.2	Référence de la classe src.usbThread.abstractThreadUSB	39

7.2.1	Description détaillée	40
7.2.2	Documentation des constructeurs et destructeur	40
7.2.2.1	__init__	40
7.2.3	Documentation des fonctions membres	40
7.2.3.1	__str__	40
7.2.3.2	copytree	41
7.2.3.3	run	41
7.2.3.4	threadType	42
7.2.3.5	todo	42
7.2.3.6	writeToLog	43
7.2.4	Documentation des données membres	43
7.2.4.1	dest	43
7.2.4.2	fileList	43
7.2.4.3	logfile	44
7.2.4.4	parent	44
7.2.4.5	subdir	44
7.2.4.6	ud	44
7.3	Référence de la classe src.ownedUsbDisk.Available	44
7.3.1	Description détaillée	45
7.3.2	Documentation des constructeurs et destructeur	45
7.3.2.1	__init__	45
7.3.3	Documentation des fonctions membres	45
7.3.3.1	finishInit	46
7.3.4	Documentation des données membres	46
7.3.4.1	noLoop	46
7.4	Référence de la classe src.usbDisk.Available	46
7.4.1	Description détaillée	47
7.4.2	Documentation des constructeurs et destructeur	47
7.4.2.1	__init__	47
7.4.3	Documentation des fonctions membres	47
7.4.3.1	__getitem__	48
7.4.3.2	__len__	48
7.4.3.3	__str__	48
7.4.3.4	__trunc__	48
7.4.3.5	compare	48
7.4.3.6	contains	49
7.4.3.7	finishInit	49
7.4.3.8	getFirstFats	49
7.4.3.9	hasDev	50
7.4.3.10	mountFirstFats	50

7.4.3.11	summary	50
7.4.4	Documentation des données membres	51
7.4.4.1	access	51
7.4.4.2	bus	51
7.4.4.3	disks	51
7.4.4.4	enumDev	51
7.4.4.5	fatPaths	51
7.4.4.6	firstFats	51
7.5	Référence de la classe src.mainWindow.CheckBoxDelegate	52
7.5.1	Description détaillée	52
7.5.2	Documentation des constructeurs et destructeur	52
7.5.2.1	__init__	52
7.5.3	Documentation des fonctions membres	52
7.5.3.1	editorEvent	53
7.5.3.2	paint	53
7.6	Référence de la classe src.checkBoxDialog.CheckBoxDialog	53
7.6.1	Description détaillée	54
7.6.2	Documentation des constructeurs et destructeur	54
7.6.2.1	__init__	54
7.6.3	Documentation des fonctions membres	54
7.6.3.1	all	55
7.6.3.2	esc	55
7.6.3.3	none	55
7.6.3.4	toggle	55
7.6.4	Documentation des données membres	55
7.6.4.1	mainWindow	55
7.6.4.2	ui	55
7.7	Référence de la classe src.choixEleves.choixElevesDialog	55
7.7.1	Description détaillée	57
7.7.2	Documentation des constructeurs et destructeur	57
7.7.2.1	__init__	57
7.7.3	Documentation des fonctions membres	57
7.7.3.1	addToList	57
7.7.3.2	checkNum	58
7.7.3.3	coche	58
7.7.3.4	connecteGestionnaire	58
7.7.3.5	decoche	58
7.7.3.6	dellnList	58
7.7.3.7	escape	59
7.7.3.8	fichierEleves	59

7.7.3.9	itemStrings	59
7.7.3.10	listeChoix	60
7.7.3.11	listeUnique_Names	60
7.7.3.12	pop	60
7.7.3.13	replie	61
7.7.3.14	takeItem	61
7.7.3.15	updateParentIcon	61
7.7.3.16	valid	62
7.7.4	Documentation des données membres	62
7.7.4.1	gestionnaire	62
7.7.4.2	ok	62
7.7.4.3	prefs	62
7.7.4.4	ui	62
7.8	Référence de la classe src.chooseInSticks.chooseDialog	63
7.8.1	Description détaillée	64
7.8.2	Documentation des constructeurs et destructeur	64
7.8.2.1	__init__	64
7.8.3	Documentation des fonctions membres	64
7.8.3.1	activate	64
7.8.3.2	append	65
7.8.3.3	baseDir	65
7.8.3.4	changeWd	65
7.8.3.5	checkValues	66
7.8.3.6	checkWorkDirs	66
7.8.3.7	choose	66
7.8.3.8	choose_dir	67
7.8.3.9	listStorages	67
7.8.3.10	minus	68
7.8.3.11	pathList	68
7.8.3.12	plus	68
7.8.3.13	selectedDiskMountPoint	69
7.8.3.14	selectedDiskOwner	69
7.8.4	Documentation des données membres	69
7.8.4.1	mainWindow	69
7.8.4.2	ok	69
7.8.4.3	okButton	70
7.8.4.4	ownedUsbDictionary	70
7.9	Référence de la classe src.copyToDialog1.copyToDialog1	70
7.9.1	Description détaillée	71
7.9.2	Documentation des fonctions membres	71

7.9.2.1	cancel	71
7.9.2.2	cd	71
7.9.2.3	changeWd	72
7.9.2.4	cont	72
7.9.2.5	displaySize	72
7.9.2.6	remove	73
7.9.2.7	select	73
7.9.2.8	selectedList	73
7.9.2.9	setFromListeDir	74
7.9.2.10	setupFromListe	74
7.9.2.11	setupToListe	74
7.9.3	Documentation des données membres	75
7.9.3.1	mainWindow	75
7.9.3.2	ok	75
7.10	Référence de la classe src.deviceListener.DeviceListener	75
7.10.1	Description détaillée	75
7.10.2	Documentation des constructeurs et destructeur	75
7.10.2.1	__init__	75
7.10.3	Documentation des fonctions membres	76
7.10.3.1	cbAdd	76
7.10.3.2	cbChange	76
7.10.3.3	cbDel	76
7.10.3.4	identify	76
7.10.3.5	isVfatUsb	77
7.10.3.6	pollDevices	77
7.10.3.7	vfatUsbPath	77
7.10.4	Documentation des données membres	78
7.10.4.1	bus	78
7.10.4.2	connectedVolumes	78
7.10.4.3	interface	78
7.10.4.4	manager	79
7.10.4.5	widget	79
7.11	Référence de la classe src.mainWindow.DiskSizeDelegate	79
7.11.1	Description détaillée	80
7.11.2	Documentation des constructeurs et destructeur	80
7.11.2.1	__init__	80
7.11.3	Documentation des fonctions membres	80
7.11.3.1	paint	80
7.11.3.2	val2txt	80
7.12	Référence de la classe src.gestclasstreeview.gestClasseTreeView	81

7.12.1	Description détaillée	81
7.12.2	Documentation des constructeurs et destructeur	82
7.12.2.1	__init__	82
7.12.3	Documentation des fonctions membres	82
7.12.3.1	allItems	82
7.12.3.2	checkedItems	82
7.12.3.3	connecteGestionnaire	82
7.12.3.4	expandedItems	82
7.12.4	Documentation des données membres	82
7.12.4.1	gest	82
7.12.4.2	root	83
7.13	Référence de la classe src.help.helpWindow	83
7.13.1	Description détaillée	84
7.13.2	Documentation des constructeurs et destructeur	84
7.13.2.1	__init__	84
7.13.3	Documentation des fonctions membres	84
7.13.3.1	loadBrowsers	84
7.13.4	Documentation des données membres	84
7.13.4.1	ui	84
7.14	Référence de la classe src.essai.machin	84
7.14.1	Description détaillée	84
7.14.2	Documentation des constructeurs et destructeur	84
7.14.2.1	__init__	85
7.14.3	Documentation des données membres	85
7.14.3.1	tm	85
7.15	Référence de la classe src.diskFull.mainWindow	85
7.15.1	Description détaillée	86
7.15.2	Documentation des constructeurs et destructeur	86
7.15.2.1	__init__	86
7.15.3	Documentation des données membres	86
7.15.3.1	total	86
7.15.3.2	ui	86
7.15.3.3	used	86
7.15.3.4	v	86
7.16	Référence de la classe src.mainWindow.mainWindow	87
7.16.1	Description détaillée	89
7.16.2	Documentation des constructeurs et destructeur	89
7.16.2.1	__init__	89
7.16.3	Documentation des fonctions membres	89
7.16.3.1	applyPreferences	89

7.16.3.2	changeWd	89
7.16.3.3	checkAll	90
7.16.3.4	checkDisks	90
7.16.3.5	checkModify	91
7.16.3.6	checkNone	91
7.16.3.7	checkToggle	92
7.16.3.8	connectTableModel	92
7.16.3.9	copyFrom	92
7.16.3.10	copyTo	93
7.16.3.11	delFiles	93
7.16.3.12	deviceAdded	93
7.16.3.13	deviceRemoved	94
7.16.3.14	diskFromOwner	94
7.16.3.15	diskSizeData	94
7.16.3.16	editOwner	95
7.16.3.17	flashLCD	95
7.16.3.18	help	96
7.16.3.19	initRedoStuff	96
7.16.3.20	manageCheckBoxes	96
7.16.3.21	namesCmd	96
7.16.3.22	namingADrive	96
7.16.3.23	normalLCD	97
7.16.3.24	preference	97
7.16.3.25	redoCmd	97
7.16.3.26	sameDiskData	97
7.16.3.27	setAvailableNames	98
7.16.3.28	setTimer	98
7.16.3.29	showEvent	98
7.16.3.30	tableClicked	99
7.16.3.31	umount	99
7.16.3.32	updateButtons	100
7.16.4	Documentation des données membres	100
7.16.4.1	availableNames	100
7.16.4.2	checkDisksLock	100
7.16.4.3	copyfromIcon	100
7.16.4.4	flashTimer	100
7.16.4.5	header	100
7.16.4.6	iconRedo	100
7.16.4.7	iconStop	100
7.16.4.8	listener	101

7.16.4.9	locale	101
7.16.4.10	manFileLocation	101
7.16.4.11	movefromIcon	101
7.16.4.12	mv	101
7.16.4.13	namesDialog	101
7.16.4.14	namesEmptyIcon	101
7.16.4.15	namesEmptyTip	101
7.16.4.16	namesFullIcon	101
7.16.4.17	namesFullTip	101
7.16.4.18	oldThreads	101
7.16.4.19	operations	101
7.16.4.20	proxy	102
7.16.4.21	recentConnect	102
7.16.4.22	redoStatusTip	102
7.16.4.23	redoToolTip	102
7.16.4.24	refreshDelay	102
7.16.4.25	refreshEnabled	102
7.16.4.26	schoolFile	102
7.16.4.27	stopStatusTip	102
7.16.4.28	stopToolTip	102
7.16.4.29	t	102
7.16.4.30	timer	102
7.16.4.31	tm	102
7.16.4.32	ui	103
7.16.4.33	visibleheader	103
7.16.4.34	workdir	103
7.17	Référence de la classe src.mytextbrowser.myTextBrowser	103
7.17.1	Description détaillée	104
7.17.2	Documentation des fonctions membres	104
7.17.2.1	setHtml	104
7.17.2.2	setSource	104
7.18	Référence de la classe src.nameAdrive.nameAdriveDialog	104
7.18.1	Description détaillée	106
7.18.2	Documentation des constructeurs et destructeur	106
7.18.2.1	__init__	106
7.18.3	Documentation des fonctions membres	106
7.18.3.1	esc	106
7.18.3.2	makeSelection	106
7.18.3.3	ok	106
7.18.3.4	selectionChanged	106

7.18.4	Documentation des données membres	106
7.18.4.1	nameList	106
7.18.4.2	numPattern	107
7.18.4.3	oldName	107
7.18.4.4	tattoo	107
7.18.4.5	ui	107
7.19	Référence de la classe src.notification.Notification	107
7.19.1	Description détaillée	107
7.19.2	Documentation des constructeurs et destructeur	107
7.19.2.1	__init__	108
7.19.3	Documentation des fonctions membres	108
7.19.3.1	notify	108
7.19.4	Documentation des données membres	108
7.19.4.1	actions	108
7.19.4.2	app_icon	108
7.19.4.3	app_name	108
7.19.4.4	body	108
7.19.4.5	expire_timeout	108
7.19.4.6	hints	108
7.19.4.7	interface	108
7.19.4.8	replaces_id	109
7.19.4.9	summary	109
7.20	Référence de la classe src.preferences.preferenceWindow	109
7.20.1	Description détaillée	110
7.20.2	Documentation des constructeurs et destructeur	110
7.20.2.1	__init__	110
7.20.3	Documentation des fonctions membres	110
7.20.3.1	enableDelay	110
7.20.3.2	setValues	110
7.20.3.3	updateRefreshLabel	110
7.20.3.4	values	110
7.20.4	Documentation des données membres	111
7.20.4.1	ui	111
7.21	Référence de la classe QAbstractTableModel	111
7.22	Référence de la classe QDialog	112
7.23	Référence de la classe QMainWindow	113
7.24	Référence de la classe QObject	113
7.25	Référence de la classe QStyledItemDelegate	114
7.26	Référence de la classe QTextBrowser	114
7.27	Référence de la classe QTreeView	115

7.28	Référence de la classe <code>src.sconet.Sconet</code>	115
7.28.1	Description détaillée	115
7.28.2	Documentation des constructeurs et destructeur	115
7.28.2.1	<code>__init__</code>	116
7.28.3	Documentation des fonctions membres	116
7.28.3.1	<code>__str__</code>	116
7.28.3.2	<code>collectClasses</code>	116
7.28.3.3	<code>collectNullTexts</code>	116
7.28.3.4	<code>collectOneClass</code>	116
7.28.3.5	<code>elementsWalk</code>	116
7.28.3.6	<code>makeCompact</code>	117
7.28.4	Documentation des données membres	117
7.28.4.1	<code>classes</code>	117
7.28.4.2	<code>donnees</code>	117
7.28.4.3	<code>nullTexts</code>	117
7.29	Référence de la classe <code>src.gestClasse.Sconet</code>	117
7.29.1	Description détaillée	119
7.29.2	Documentation des constructeurs et destructeur	119
7.29.2.1	<code>__init__</code>	119
7.29.3	Documentation des fonctions membres	119
7.29.3.1	<code>__str__</code>	119
7.29.3.2	<code>collectClasses</code>	119
7.29.3.3	<code>collectNullTexts</code>	119
7.29.3.4	<code>collectOneClass</code>	119
7.29.3.5	<code>elementsWalk</code>	120
7.29.3.6	<code>eleveParID</code>	120
7.29.3.7	<code>elevesDeClasse</code>	120
7.29.3.8	<code>makeCompact</code>	121
7.29.3.9	<code>showable_name</code>	121
7.29.3.10	<code>unIDEleveDeClasse</code>	121
7.29.3.11	<code>unique_name</code>	121
7.29.4	Documentation des données membres	121
7.29.4.1	<code>classes</code>	121
7.29.4.2	<code>currentClassName</code>	121
7.29.4.3	<code>currentID</code>	121
7.29.4.4	<code>currentResult</code>	122
7.29.4.5	<code>donnees</code>	122
7.29.4.6	<code>nullTexts</code>	122
7.30	Référence de la classe <code>Thread</code>	122
7.31	Référence de la classe <code>src.usbThread.threadCopyFromUSB</code>	122

7.31.1	Description détaillée	124
7.31.2	Documentation des constructeurs et destructeur	124
7.31.2.1	__init__	124
7.31.3	Documentation des fonctions membres	124
7.31.3.1	todo	124
7.31.4	Documentation des données membres	125
7.31.4.1	rootPath	125
7.32	Référence de la classe src.usbThread.threadCopyToUSB	125
7.32.1	Description détaillée	126
7.32.2	Documentation des constructeurs et destructeur	126
7.32.2.1	__init__	126
7.32.3	Documentation des fonctions membres	127
7.32.3.1	threadType	127
7.32.3.2	todo	127
7.33	Référence de la classe src.usbThread.threadDeleteInUSB	127
7.33.1	Description détaillée	129
7.33.2	Documentation des constructeurs et destructeur	129
7.33.2.1	__init__	129
7.33.3	Documentation des fonctions membres	129
7.33.3.1	todo	129
7.34	Référence de la classe src.usbThread.threadMoveFromUSB	130
7.34.1	Description détaillée	131
7.34.2	Documentation des constructeurs et destructeur	131
7.34.2.1	__init__	131
7.34.3	Documentation des fonctions membres	131
7.34.3.1	todo	131
7.34.4	Documentation des données membres	132
7.34.4.1	rootPath	132
7.35	Référence de la classe src.usbThread.ThreadRegister	132
7.35.1	Description détaillée	132
7.35.2	Documentation des constructeurs et destructeur	133
7.35.2.1	__init__	133
7.35.3	Documentation des fonctions membres	133
7.35.3.1	__str__	133
7.35.3.2	busy	133
7.35.3.3	pop	133
7.35.3.4	push	133
7.35.3.5	threadSet	133
7.35.4	Documentation des données membres	133
7.35.4.1	dico	133

7.36	Référence de la classe <code>src.ownedUsbDisk.uDisk</code> . . . . .	134
7.36.1	Description détaillée . . . . .	135
7.36.2	Documentation des constructeurs et destructeur . . . . .	135
7.36.2.1	<code>__init__</code> . . . . .	135
7.36.3	Documentation des fonctions membres . . . . .	135
7.36.3.1	<code>__getitem__</code> . . . . .	135
7.36.3.2	<code>ensureOwner</code> . . . . .	136
7.36.3.3	<code>headers</code> . . . . .	136
7.36.3.4	<code>ownerByDb</code> . . . . .	136
7.36.3.5	<code>readQuirks</code> . . . . .	137
7.36.3.6	<code>tattoo</code> . . . . .	137
7.36.3.7	<code>uniqueId</code> . . . . .	138
7.36.3.8	<code>visibleDir</code> . . . . .	138
7.36.4	Documentation des données membres . . . . .	138
7.36.4.1	<code>headers</code> . . . . .	139
7.36.4.2	<code>model</code> . . . . .	139
7.36.4.3	<code>owner</code> . . . . .	139
7.36.4.4	<code>vendor</code> . . . . .	139
7.36.4.5	<code>visibleDirs</code> . . . . .	139
7.37	Référence de la classe <code>src.usbDisk.uDisk</code> . . . . .	139
7.37.1	Description détaillée . . . . .	141
7.37.2	Documentation des constructeurs et destructeur . . . . .	141
7.37.2.1	<code>__init__</code> . . . . .	141
7.37.3	Documentation des fonctions membres . . . . .	141
7.37.3.1	<code>__getitem__</code> . . . . .	141
7.37.3.2	<code>__str__</code> . . . . .	142
7.37.3.3	<code>devicePropProxy</code> . . . . .	142
7.37.3.4	<code>ensureMounted</code> . . . . .	142
7.37.3.5	<code>file</code> . . . . .	143
7.37.3.6	<code>getFatUuid</code> . . . . .	143
7.37.3.7	<code>getFirstFat</code> . . . . .	144
7.37.3.8	<code>getProp</code> . . . . .	144
7.37.3.9	<code>headers</code> . . . . .	145
7.37.3.10	<code>isDosFat</code> . . . . .	145
7.37.3.11	<code>isMounted</code> . . . . .	146
7.37.3.12	<code>isTrue</code> . . . . .	146
7.37.3.13	<code>isUsbDisk</code> . . . . .	147
7.37.3.14	<code>master</code> . . . . .	147
7.37.3.15	<code>mountPoint</code> . . . . .	148
7.37.3.16	<code>showableProp</code> . . . . .	148

7.37.3.17 title	149
7.37.3.18 uniqueId	149
7.37.3.19 unNumberProp	150
7.37.3.20 valuableProperties	150
7.37.4 Documentation des données membres	151
7.37.4.1 device	151
7.37.4.2 device_prop	151
7.37.4.3 devStuff	151
7.37.4.4 fatuuid	151
7.37.4.5 firstFat	151
7.37.4.6 headers	152
7.37.4.7 mp	152
7.37.4.8 path	152
7.37.4.9 rlock	152
7.37.4.10 selected	152
7.37.4.11 stickid	152
7.37.4.12 uuid	152
7.38 Référence de la classe src.mainWindow.UsbDiskDelegate	152
7.38.1 Description détaillée	153
7.38.2 Documentation des constructeurs et destructeur	153
7.38.2.1 __init__	153
7.38.3 Documentation des fonctions membres	153
7.38.3.1 paint	153
7.38.4 Documentation des données membres	153
7.38.4.1 busyPixmap	153
7.38.4.2 okPixmap	154
7.39 Référence de la classe src.essai.usbT	154
7.39.1 Description détaillée	154
7.39.2 Documentation des constructeurs et destructeur	154
7.39.2.1 __init__	154
7.39.3 Documentation des fonctions membres	154
7.39.3.1 threadCopyToUSB	154
7.40 Référence de la classe src.mainWindow.usbTableModel	154
7.40.1 Description détaillée	155
7.40.2 Documentation des constructeurs et destructeur	155
7.40.2.1 __init__	155
7.40.3 Documentation des fonctions membres	156
7.40.3.1 columnCount	156
7.40.3.2 data	156
7.40.3.3 headerData	156

7.40.3.4	partition	156
7.40.3.5	popCmd	156
7.40.3.6	pushCmd	156
7.40.3.7	rowCount	157
7.40.3.8	setData	157
7.40.3.9	sort	157
7.40.3.10	updateOwnerColumn	157
7.40.4	Documentation des données membres	157
7.40.4.1	donnees	157
7.40.4.2	header	158
7.40.4.3	pere	158
<b>8</b>	<b>Documentation des fichiers</b>	<b>159</b>
8.1	Référence du fichier src/__init__.py	159
8.2	Référence du fichier src/checkboxDialog.py	159
8.3	Référence du fichier src/choixEleves.py	159
8.4	Référence du fichier src/chooseInSticks.py	160
8.5	Référence du fichier src/copyToDialog1.py	160
8.6	Référence du fichier src/db.py	160
8.7	Référence du fichier src/debug.py	161
8.8	Référence du fichier src/deviceListener.py	161
8.9	Référence du fichier src/diskFull.py	162
8.10	Référence du fichier src/essai.py	162
8.11	Référence du fichier src/gestClasse.py	162
8.12	Référence du fichier src/gestclassetreeview.py	163
8.13	Référence du fichier src/globaldef.py	163
8.14	Référence du fichier src/help.py	163
8.15	Référence du fichier src/mainWindow.py	164
8.16	Référence du fichier src/marques.py	164
8.17	Référence du fichier src/mytextbrowser.py	164
8.18	Référence du fichier src/nameAdrive.py	165
8.19	Référence du fichier src/notification.py	165
8.20	Référence du fichier src/ownedUsbDisk.py	165
8.21	Référence du fichier src/preferences.py	166
8.22	Référence du fichier src/scolasync.py	166
8.23	Référence du fichier src/sconet.py	166
8.24	Référence du fichier src/test3.py	167
8.25	Référence du fichier src/usbDisk.py	167
8.26	Référence du fichier src/usbThread.py	167
8.27	Référence du fichier src/version.py	168

# 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 4.0 :**

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

**Copyright © 2010-2012 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 [OFSET](#) 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 ou plus
- 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

# Index des espaces de nommage

### 2.1 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 baladeurs, de dictaphones ou de clés USB	11
<a href="#">src</a>		11
<a href="#">src.checkBoxDialog</a>		12
<a href="#">src.choixEleves</a>		12
<a href="#">src.chooseInSticks</a>		13
<a href="#">src.copyToDialog1</a>		14
<a href="#">src.db</a>		15
<a href="#">src.debug</a>		19
<a href="#">src.deviceListener</a>		20
<a href="#">src.diskFull</a>		21
<a href="#">src.essai</a>		21
<a href="#">src.gestClasse</a>		22
<a href="#">src.gestclassetreeview</a>		22
<a href="#">src.globaldef</a>		23
<a href="#">src.help</a>		24
<a href="#">src.mainWindow</a>		24
<a href="#">src.marques</a>		26
<a href="#">src.mytextbrowser</a>		26
<a href="#">src.nameAdrive</a>		27
<a href="#">src.notification</a>		27
<a href="#">src.ownedUsbDisk</a>		28
<a href="#">src.preferences</a>		29
<a href="#">src.scolasync</a>		30
<a href="#">src.sconet</a>		31
<a href="#">src.test3</a>		31
<a href="#">src.usbDisk</a>		32
<a href="#">src.usbThread</a>		33
<a href="#">src.version</a>		35



# Chapitre 3

## Index des classes

### 3.1 Hiérarchie des classes

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

src.gestClasse.AbstractGestClasse . . . . .	37
src.gestClasse.Sconet . . . . .	117
src.usbDisk.Available . . . . .	46
src.ownedUsbDisk.Available . . . . .	44
src.deviceListener.DeviceListener . . . . .	75
src.essai.machin . . . . .	84
src.notification.Notification . . . . .	107
QAbstractTableModel . . . . .	111
src.mainWindow.usbTableModel . . . . .	154
QDialog . . . . .	112
src.checkBoxDialog.CheckBoxDialog . . . . .	53
src.choixEleves.choixElevesDialog . . . . .	55
src.chooseInSticks.chooseDialog . . . . .	63
src.copyToDialog1.copyToDialog1 . . . . .	70
src.help.helpWindow . . . . .	83
src.nameAdrive.nameAdriveDialog . . . . .	104
src.preferences.preferenceWindow . . . . .	109
QMainWindow . . . . .	113
src.diskFull.mainWindow . . . . .	85
src.mainWindow.mainWindow . . . . .	87
QObject . . . . .	113
src.ownedUsbDisk.uDisk . . . . .	134
QStyledItemDelegate . . . . .	114
src.mainWindow.CheckBoxDelegate . . . . .	52
src.mainWindow.DiskSizeDelegate . . . . .	79
src.mainWindow.UsbDiskDelegate . . . . .	152
QTextBrowser . . . . .	114
src.mytextbrowser.myTextBrowser . . . . .	103
QTreeView . . . . .	115
src.gestclassetreeview.gestClasseTreeView . . . . .	81
src.sconet.Sconet . . . . .	115
Thread . . . . .	122
src.usbThread.abstractThreadUSB . . . . .	39
src.usbThread.threadCopyFromUSB . . . . .	122
src.usbThread.threadCopyToUSB . . . . .	125

---

src.usbThread.threadDeleteInUSB . . . . .	127
src.usbThread.threadMoveFromUSB . . . . .	130
src.usbThread.ThreadRegister . . . . .	132
src.usbDisk.uDisk . . . . .	139
src.ownedUsbDisk.uDisk . . . . .	134
src.essai.usbT . . . . .	154

# Chapitre 4

## Index des classes

### 4.1 Liste des classes

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

<a href="#">src.gestClasse.AbstractGestClasse</a> . . . . .	37
<a href="#">src.usbThread.abstractThreadUSB</a>	
Une classe abstraite, qui sert de creuset pour les classe servant aux copies et aux effacements	39
<a href="#">src.ownedUsbDisk.Available</a>	
Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires . . .	44
<a href="#">src.usbDisk.Available</a>	
Une classe pour représenter la collection des disques USB connectés . . . . .	46
<a href="#">src.mainWindow.CheckBoxDelegate</a> . . . . .	52
<a href="#">src.checkBoxDialog.CheckBoxDialog</a>	
Un dialogue pour gérer les cases à cocher de l'application . . . . .	53
<a href="#">src.choixEleves.choixElevesDialog</a>	
Implémente un dialogue permettant de choisir des élèves les propriétés importantes sont self.ok, vrai si on doit prendre en compte la liste sélectionnée, et le contenu de la liste des sélectionnés, dont on peut récupérer les élèves un par un à l'aide de self.pop() . . . . .	55
<a href="#">src.chooseInSticks.chooseDialog</a>	
Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB . . . . .	63
<a href="#">src.copyToDialog1.copyToDialog1</a>	
Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB .	70
<a href="#">src.deviceListener.DeviceListener</a> . . . . .	75
<a href="#">src.mainWindow.DiskSizeDelegate</a>	
Classe pour figurer la taille de la mémoire du baladeur . . . . .	79
<a href="#">src.gestclassetreeview.gestClasseTreeView</a> . . . . .	81
<a href="#">src.help.helpWindow</a> . . . . .	83
<a href="#">src.essai.machin</a> . . . . .	84
<a href="#">src.diskFull.mainWindow</a> . . . . .	85
<a href="#">src.mainWindow.mainWindow</a> . . . . .	87
<a href="#">src.mytextbrowser.myTextBrowser</a>	
Une classe qui ouvre Firefox quand on clique sur un lien externe . . . . .	103
<a href="#">src.nameAdrive.nameAdriveDialog</a>	
Un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles . . . . .	104
<a href="#">src.notification.Notification</a>	
Une classe pour afficher des notifications à l'écran . . . . .	107
<a href="#">src.preferences.preferenceWindow</a> . . . . .	109
<a href="#">QAbstractTableModel</a> . . . . .	111
<a href="#">QDialog</a> . . . . .	112
<a href="#">QMainWindow</a> . . . . .	113
<a href="#">QObject</a> . . . . .	113
<a href="#">QStyledItemDelegate</a> . . . . .	114

<a href="#">QTextBrowser</a> . . . . .	114
<a href="#">QTreeView</a> . . . . .	115
<a href="#">src.sconet.Sconet</a>	
Une classe pour travailler avec des données <a href="#">Sconet</a> . . . . .	115
<a href="#">src.gestClasse.Sconet</a>	
Une classe pour travailler avec des données <a href="#">Sconet</a> . . . . .	117
<a href="#">Thread</a> . . . . .	122
<a href="#">src.usbThread.threadCopyFromUSB</a>	
Classe pour les threads copiant depuis les clés USB . . . . .	122
<a href="#">src.usbThread.threadCopyToUSB</a>	
Classe pour les threads copiant vers les clés USB . . . . .	125
<a href="#">src.usbThread.threadDeletelnUSB</a>	
Classe pour les threads effaçant des sous-arbres dans les clés USB . . . . .	127
<a href="#">src.usbThread.threadMoveFromUSB</a>	
Classe pour les threads déplaçant des fichiers depuis les clés USB . . . . .	130
<a href="#">src.usbThread.ThreadRegister</a>	
Une classe pour tenir un registre des threads concernant les baladeurs . . . . .	132
<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 . . . . .	134
<a href="#">src.usbDisk.uDisk</a>	
Une classe pour représenter un disque ou une partition . . . . .	139
<a href="#">src.mainWindow.UsbDiskDelegate</a>	
Classe pour identifier le baladeur dans le tableau . . . . .	152
<a href="#">src.essai.usbT</a> . . . . .	154
<a href="#">src.mainWindow.usbTableModel</a>	
Un modèle de table pour des séries de clés USB . . . . .	154

# Chapitre 5

## Index des fichiers

### 5.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/__init__.py	159
src/checkBoxDialog.py	159
src/choixEleves.py	159
src/chooseInSticks.py	160
src/copyToDialog1.py	160
src/db.py	160
src/debug.py	161
src/deviceListener.py	161
src/diskFull.py	162
src/essai.py	162
src/gestClasse.py	162
src/gestclassetreeview.py	163
src/globaldef.py	163
src/help.py	163
src/mainWindow.py	164
src/marques.py	164
src/mytextbrowser.py	164
src/nameAdrive.py	165
src/notification.py	165
src/ownedUsbDisk.py	165
src/preferences.py	166
src/scolasync.py	166
src/sconet.py	166
src/test3.py	167
src/usbDisk.py	167
src/usbThread.py	167
src/version.py	168



## Chapitre 6

# Documentation des espaces de nommage

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

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

#### 6.1.1 Description détaillée

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

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

#### Paquetages

- namespace [checkBoxDialog](#)
- namespace [choixEleves](#)
- namespace [chooseInSticks](#)
- namespace [copyToDialog1](#)
- namespace [db](#)
- namespace [debug](#)
- namespace [deviceListener](#)
- namespace [diskFull](#)
- namespace [essai](#)
- namespace [gestClasse](#)
- namespace [gestclassetreeview](#)
- namespace [globaldef](#)
- namespace [help](#)
- namespace [mainWindow](#)
- namespace [marques](#)
- namespace [mytextbrowser](#)
- namespace [nameADrive](#)
- namespace [notification](#)
- namespace [ownedUsbDisk](#)
- namespace [preferences](#)
- namespace [scolasync](#)
- namespace [sconet](#)
- namespace [test3](#)
- namespace [usbDisk](#)
- namespace [usbThread](#)
- namespace [version](#)

#### Variables

- [python3safe](#) = True

## 6.2.1 Documentation des variables

### 6.2.1.1 `src.python3safe = True`

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

## 6.3 Référence de l'espace de nommage `src.checkBoxDialog`

### Classes

- class `CheckBoxDialog`  
*Un dialogue pour gérer les cases à cocher de l'application.*

### Variables

- string `licenceEn`
- `python3safe = True`

## 6.3.1 Documentation des variables

### 6.3.1.1 `string src.checkBoxDialog.licenceEn`

#### Valeur initiale :

```

1 """
2     file checkBoxDialog.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 3 du fichier `checkBoxDialog.py`.

### 6.3.1.2 `src.checkBoxDialog.python3safe = True`

Définition à la ligne 23 du fichier `checkBoxDialog.py`.

## 6.4 Référence de l'espace de nommage `src.choixEleves`

### Classes

- class `choixElevesDialog`  
*implémente un dialogue permettant de choisir des élèves les propriétés importantes sont `self.ok`, vrai si on doit prendre en compte la liste sélectionnée, et le contenu de la liste des sélectionnés, dont on peut récupérer les élèves un par un à l'aide de `self.pop()`*

## Variables

- dictionary `licence` = {}
- `python3safe` = True
- tuple `app` = `QApplication(sys.argv)`
- tuple `d` = `choixElevesDialog(gestionnaire=gestClasse.Sconet)`
- tuple `i` = `d.pop()`

### 6.4.1 Documentation des variables

#### 6.4.1.1 tuple `src.choixEleves.app` = `QApplication(sys.argv)`

Définition à la ligne 266 du fichier `choixEleves.py`.

#### 6.4.1.2 tuple `src.choixEleves.d` = `choixElevesDialog(gestionnaire=gestClasse.Sconet)`

Définition à la ligne 267 du fichier `choixEleves.py`.

#### 6.4.1.3 tuple `src.choixEleves.i` = `d.pop()`

Définition à la ligne 270 du fichier `choixEleves.py`.

#### 6.4.1.4 dictionary `src.choixEleves.licence` = {}

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

#### 6.4.1.5 `src.choixEleves.python3safe` = True

Définition à la ligne 25 du fichier `choixEleves.py`.

## 6.5 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`
- `python3safe` = True

### 6.5.1 Documentation des variables

#### 6.5.1.1 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@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 version3 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.

#### 6.5.1.2 src.chooseInSticks.python3safe = True

Définition à la ligne 24 du fichier chooseInSticks.py.

## 6.6 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`
- `python3safe = True`
- tuple `app = QApplication(sys.argv)`
- tuple `windows = copyToDialog1()`

### 6.6.1 Documentation des variables

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

Définition à la ligne 211 du fichier copyToDialog1.py.

#### 6.6.1.2 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@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 version3 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.

### 6.6.1.3 src.copyToDialog1.python3safe = True

Définition à la ligne 24 du fichier copyToDialog1.py.

### 6.6.1.4 tuple src.copyToDialog1.windows = copyToDialog1()

Définition à la ligne 212 du fichier copyToDialog1.py.

## 6.7 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 `hasStudent`  
*vérifie qu'un étudiant est déjà connu*
- def `knowsId`  
*dit 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` = {}
- `python3safe` = True
- `database` = None
- `cursor` = None

### 6.7.1 Documentation des fonctions

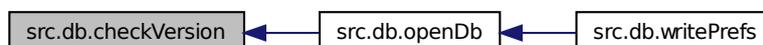
#### 6.7.1.1 def src.db.checkVersion ( major, minor )

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 58 du fichier db.py.

Voici le graphe des appelants de cette fonction :



6.7.1.2 `def src.db.hasStudent ( student )`

vérifie qu'un étudiant est déjà connu

## Paramètres

<i>student</i>	propriétaire du baladeur
----------------	--------------------------

## Renvoi

True si le propriétaire existe déjà

Définition à la ligne 80 du fichier db.py.

6.7.1.3 `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

## Renvoi

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

Définition à la ligne 93 du fichier db.py.

Voici le graphe des appelants de cette fonction :

6.7.1.4 `def src.db.openDb ( )`

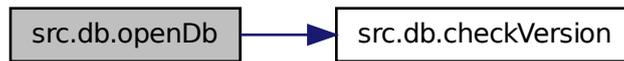
Ouverture de la base de données de l'application, et création si nécessaire.

**Renvoie**

une instance de base de données sqlite3

Définition à la ligne 38 du fichier db.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

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

renvoie les préférences de ScolaSync

**Renvoie**

un dictionnaire de préférences

Définition à la ligne 126 du fichier db.py.

**6.7.1.6 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 112 du fichier db.py.

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

définit le nouveau nom du répertoire de travail préféré.

Définition à la ligne 161 du fichier db.py.

### 6.7.1.8 def src.db.tattooList ( )

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

Définition à la ligne 102 du fichier db.py.

### 6.7.1.9 def src.db.writePrefs ( *prefs* )

inscrit les préférences

#### Paramètres

<i>prefs</i>	un dictionnaire {"checkable" : booléen toujours vrai, "workdir" : le répertoire préféré pour les fichiers de travail}
--------------	---

Définition à la ligne 187 du fichier db.py.

Voici le graphe d'appel pour cette fonction :



### 6.7.1.10 def src.db.writeStudent ( *stickid*, *uuid*, *tattoo*, *student* )

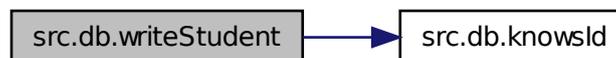
inscrit un étudiant comme propriétaire d'une clé USB

#### Paramètres

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

Définition à la ligne 171 du fichier db.py.

Voici le graphe d'appel pour cette fonction :



## 6.7.2 Documentation des variables

### 6.7.2.1 src.db.cursor = None

Définition à la ligne 31 du fichier db.py.

## 6.7.2.2 src.db.database = None

Définition à la ligne 30 du fichier db.py.

## 6.7.2.3 dictionary src.db.licence = {}

Définition à la ligne 4 du fichier db.py.

## 6.7.2.4 src.db.python3safe = True

Définition à la ligne 25 du fichier db.py.

## 6.8 Référence de l'espace de nommage src.debug

### Fonctions

- def [button](#)  
*ajoute un bouton de débogage dans une fenêtre*
- def [listePartitionsCochees](#)  
*renseigne sur la liste des partions cochées de la fenêtre principale*

### Variables

- dictionary [licence](#) = {}  
*Ce module facilite le debogage.*
- string [licenceEn](#)
- string [licenceFr](#)

### 6.8.1 Documentation des fonctions

## 6.8.1.1 def src.debug.button ( w, cb )

ajoute un bouton de débogage dans une fenêtre

#### Paramètres

<i>w</i>	la fenêtre
<i>cb</i>	une fonction de rappel à effectuer ; celle ci accepte w comme premier paramètre fonction pour passer la paramètre mw à la fonction de rappel cb

Définition à la ligne 62 du fichier debug.py.

## 6.8.1.2 def src.debug.listePartitionsCochees ( w )

renseigne sur la liste des partions cochées de la fenêtre principale

#### Paramètres

<i>w</i>	la fenêtre principale
----------	-----------------------

Définition à la ligne 81 du fichier debug.py.

### 6.8.2 Documentation des variables

### 6.8.2.1 dictionary `src.debug.licence = {}`

Ce module facilite le debogage.

Définition à la ligne 8 du fichier `debug.py`.

### 6.8.2.2 string `src.debug.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) 2010-2013 Georges Khaznadar <georgesk@debian.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 9 du fichier `debug.py`.

### 6.8.2.3 string `src.debug.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) 2010-2013 Georges Khaznadar <georgesk@debian.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 32 du fichier `debug.py`.

## 6.9 Référence de l'espace de nommage `src.deviceListener`

### Classes

— class [DeviceListener](#)

### Variables

— dictionary `licence = {}`  
 — `python3safe = True`

### 6.9.1 Documentation des variables

6.9.1.1 dictionary `src.deviceListener.licence = {}`

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

6.9.1.2 `src.deviceListener.python3safe = True`

Définition à la ligne 25 du fichier `deviceListener.py`.

## 6.10 Référence de l'espace de nommage src.diskFull

### Classes

— class `mainWindow`

### Fonctions

— def `sceneWithUsage`

### Variables

— dictionary `licence = {}`  
 — `python3safe = True`

### 6.10.1 Documentation des fonctions

6.10.1.1 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 62 du fichier `diskFull.py`.

### 6.10.2 Documentation des variables

6.10.2.1 dictionary `src.diskFull.licence = {}`

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

6.10.2.2 `src.diskFull.python3safe = True`

Définition à la ligne 26 du fichier `diskFull.py`.

## 6.11 Référence de l'espace de nommage src.essai

## Classes

- class [usbT](#)
- class [machin](#)

## 6.12 Référence de l'espace de nommage src.gestClasse

### Classes

- class [AbstractGestClasse](#)
- class [Sconet](#)  
*Une classe pour travailler avec des données [Sconet](#).*

### Variables

- dictionary [licence](#) = {}  
*Ce module permet de gérer des classes d'élèves.*
- [python3safe](#) = True

#### 6.12.1 Documentation des variables

##### 6.12.1.1 dictionary `src.gestClasse.licence` = {}

Ce module permet de gérer des classes d'élèves.

La classe [AbstractGestClasse](#) définit les fonctions minimales à implémenter pour chaque gestionnaire de classes.

Définition à la ligne 11 du fichier `gestClasse.py`.

##### 6.12.1.2 `src.gestClasse.python3safe` = True

Définition à la ligne 32 du fichier `gestClasse.py`.

## 6.13 Référence de l'espace de nommage src.gestclassetreeview

### Classes

- class [gestClasseTreeView](#)

### Variables

- dictionary [licence](#) = {}
- [python3safe](#) = True

#### 6.13.1 Documentation des variables

##### 6.13.1.1 dictionary `src.gestclassetreeview.licence` = {}

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

##### 6.13.1.2 `src.gestclassetreeview.python3safe` = True

Définition à la ligne 25 du fichier `gestclassetreeview.py`.

## 6.14 Référence de l'espace de nommage src.globaldef

### Fonctions

- def `firstdir`  
Renvoie le premier répertoire existant d'une liste de propositions.

### Variables

- string `licenceEn`  
*globaldef.py* is part of the package *scolasync*.
- `python3safe` = True
- string `userShareDir` = "~/scolasync"
- string `logFileName` = "~/scolasync/scolasync.log"
- string `markFileName` = "~/scolasync/marques.py"

#### 6.14.1 Documentation des fonctions

##### 6.14.1.1 def src.globaldef.firstdir ( / )

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

##### Paramètres

	/ la liste de propositions
--	----------------------------

Définition à la ligne 52 du fichier `globaldef.py`.

#### 6.14.2 Documentation des variables

##### 6.14.2.1 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`.

##### 6.14.2.2 string src.globaldef.logFileName = "~/scolasync/scolasync.log"

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

6.14.2.3 `string src.globaldef.markFileName = "~/scolasync/marques.py"`

Définition à la ligne 40 du fichier `globaldef.py`.

6.14.2.4 `src.globaldef.python3safe = True`

Définition à la ligne 33 du fichier `globaldef.py`.

6.14.2.5 `string src.globaldef.userShareDir = "~/scolasync"`

Définition à la ligne 38 du fichier `globaldef.py`.

## 6.15 Référence de l'espace de nommage `src.help`

### Classes

— class [helpWindow](#)

### Variables

— dictionary [licence](#) = {}  
— [python3safe](#) = True

### 6.15.1 Documentation des variables

6.15.1.1 dictionary `src.help.licence = {}`

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

6.15.1.2 `src.help.python3safe = True`

Définition à la ligne 26 du fichier `help.py`.

## 6.16 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](#)  
*Classe pour identifier le baladeur dans le tableau.*  
— class [DiskSizeDelegate](#)  
*Classe pour figurer la taille de la mémoire du baladeur.*

### Fonctions

— def [registerCmd](#)  
*enregistre la commande `cmd` pour la partition donnée*  
— def [CheckBoxRect](#)

## Variables

- dictionary `licence` = {}
- dictionary `activeThreads` = {}
- dictionary `pastCommands` = {}
- `lastCommand` = None

### 6.16.1 Documentation des fonctions

#### 6.16.1.1 `def src.mainWindow.CheckBoxRect ( view_item_style_options )`

##### Paramètres

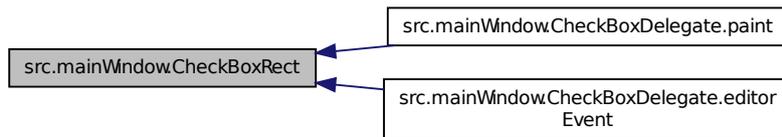
<code>view_item_style_options</code>	des options permettant de décider de la taille d'un rectangle
--------------------------------------	---

##### Renvoie

un QRect dimensionné selon les bonnes options

Définition à la ligne 862 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



#### 6.16.1.2 `def src.mainWindow.registerCmd ( cmd, partition )`

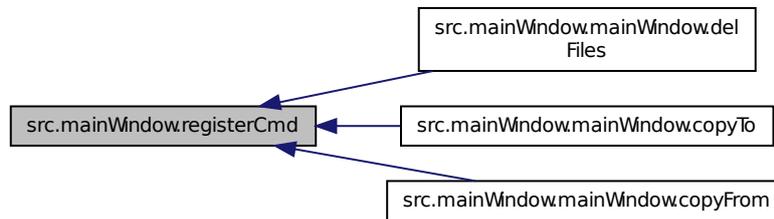
enregistre la commande cmd pour la partition donnée

##### Paramètres

<code>cmd</code>	une commande pour créer un thread t
<code>partition</code>	une partition

Définition à la ligne 54 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



## 6.16.2 Documentation des variables

### 6.16.2.1 dictionnaire `src.mainWindow.activeThreads = {}`

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

### 6.16.2.2 `src.mainWindow.lastCommand = None`

Définition à la ligne 46 du fichier `mainWindow.py`.

### 6.16.2.3 dictionnaire `src.mainWindow.licence = {}`

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

### 6.16.2.4 dictionnaire `src.mainWindow.pastCommands = {}`

Définition à la ligne 45 du fichier `mainWindow.py`.

## 6.17 Référence de l'espace de nommage `src.marques`

## 6.18 Référence de l'espace de nommage `src.mytextbrowser`

### Classes

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

### Variables

- dictionnaire [licence](#) = {}
- [python3safe](#) = True

### 6.18.1 Documentation des variables

6.18.1.1 dictionary `src.mytextbrowser.licence = {}`

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

6.18.1.2 `src.mytextbrowser.python3safe = True`

Définition à la ligne 26 du fichier `mytextbrowser.py`.

## 6.19 Référence de l'espace de nommage src.nameAdrive

### Classes

- class `nameAdriveDialog`  
*un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles*

### Variables

- dictionary `licence = {}`
- `python3safe = True`

#### 6.19.1 Documentation des variables

6.19.1.1 dictionary `src.nameAdrive.licence = {}`

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

6.19.1.2 `src.nameAdrive.python3safe = True`

Définition à la ligne 25 du fichier `nameAdrive.py`.

## 6.20 Référence de l'espace de nommage src.notification

### Classes

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

### Variables

- dictionary `licence = {}`
- `python3safe = True`
- tuple `notif`

#### 6.20.1 Documentation des variables

6.20.1.1 dictionary `src.notification.licence = {}`

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

### 6.20.1.2 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 77 du fichier `notification.py`.

### 6.20.1.3 `src.notification.python3safe = True`

Définition à la ligne 26 du fichier `notification.py`.

## 6.21 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 `tattooInDir`  
*Renvoie le tatouage pour un point de montage donné, quitte à le créer si nécessaire.*
- def `editRecord`  
*édition de la base de données.*

### Variables

- dictionary `licence = {}`
- `python3safe = True`
- tuple `machin = Available()`

### 6.21.1 Documentation des fonctions

#### 6.21.1.1 `def src.ownedUsbDisk.editRecord ( owd, hint = " " )`

édition de la base de données.

#### Paramètres

<code>owd</code>	une instance de <code>ownedUsbDisk</code>
<code>hint</code>	chaîne vide par défaut. Peut être le nom de l'ancien propriétaire

Définition à la ligne 71 du fichier `ownedUsbDisk.py`.

#### 6.21.1.2 `def src.ownedUsbDisk.tattooInDir ( mountPoint )`

Renvoie le tatouage pour un point de montage donné, quitte à le créer si nécessaire.

## Paramètres

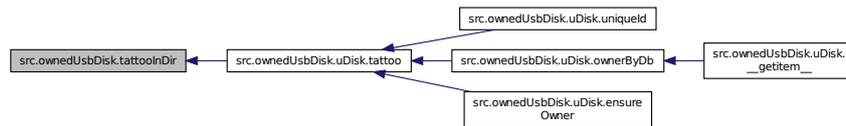
<i>mountPoint</i>	un point de montage de partition
-------------------	----------------------------------

## Renvoi

le tatouage

Définition à la ligne 44 du fichier ownedUsbDisk.py.

Voici le graphe des appelants de cette fonction :



## 6.21.2 Documentation des variables

## 6.21.2.1 dictionary src.ownedUsbDisk.licence = {}

Définition à la ligne 4 du fichier ownedUsbDisk.py.

## 6.21.2.2 tuple src.ownedUsbDisk.machin = Available()

Définition à la ligne 252 du fichier ownedUsbDisk.py.

## 6.21.2.3 src.ownedUsbDisk.python3safe = True

Définition à la ligne 25 du fichier ownedUsbDisk.py.

## 6.22 Référence de l'espace de nommage src.preferences

## Classes

— class [preferenceWindow](#)

## Variables

— dictionary [licence](#) = {}  
 — [python3safe](#) = True

## 6.22.1 Documentation des variables

## 6.22.1.1 dictionary src.preferences.licence = {}

Définition à la ligne 5 du fichier preferences.py.

## 6.22.1.2 src.preferences.python3safe = True

Définition à la ligne 26 du fichier preferences.py.

## 6.23 Référence de l'espace de nommage src.scolasync

### Fonctions

- def `run`  
Le lancement de l'application.

### Variables

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

### 6.23.1 Documentation des fonctions

6.23.1.1 `def src.scolasync.run ( debugger=False, callback=lambda x: print(x) )`

Le lancement de l'application.

#### Paramètres

<code>debugger</code>	s'il est vrai, un bouton de débogage est ajouté
<code>callback</code>	une fonction de rappel à un paramètre (qui sera la fenêtre principale, le cas échéant)

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

### 6.23.2 Documentation des variables

6.23.2.1 `dictionary src.scolasync.licence = {}`

Définition à la ligne 85 du fichier `scolasync.py`.

6.23.2.2 `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) 2010-2012 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`.

6.23.2.3 `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) 2010-2012 Georges Khaznadar <georgesk@ofset.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.

## 6.24 Référence de l'espace de nommage src.sconet

### Classes

- class `Sconet`  
*Une classe pour travailler avec des données `Sconet`.*

### Variables

- dictionary `licence` = {}
- `python3safe` = True
- tuple `s` = `Sconet("../exemples/SCONET_test.xml")`

#### 6.24.1 Documentation des variables

##### 6.24.1.1 dictionary `src.sconet.licence` = {}

Définition à la ligne 4 du fichier sconet.py.

##### 6.24.1.2 `src.sconet.python3safe` = True

Définition à la ligne 25 du fichier sconet.py.

##### 6.24.1.3 tuple `src.sconet.s` = `Sconet("../exemples/SCONET_test.xml")`

Définition à la ligne 105 du fichier sconet.py.

## 6.25 Référence de l'espace de nommage src.test3

### Variables

- `python3safe` = True
- tuple `files` = `os.listdir(".")`
- tuple `pattern` = `re.compile(".*\.py$")`
- list `safe` = []
- list `notsafe` = []
- tuple `moduleName` = `f.replace(".py", "")`
- tuple `module` = `__import__(moduleName)`

### 6.25.1 Documentation des variables

#### 6.25.1.1 `list src.test3.files = os.listdir(".")`

Définition à la ligne 8 du fichier test3.py.

#### 6.25.1.2 `tuple src.test3.module = __import__(moduleName)`

Définition à la ligne 16 du fichier test3.py.

#### 6.25.1.3 `tuple src.test3.moduleName = f.replace(".py", "")`

Définition à la ligne 14 du fichier test3.py.

#### 6.25.1.4 `list src.test3.notsafe = []`

Définition à la ligne 12 du fichier test3.py.

#### 6.25.1.5 `tuple src.test3.pattern = re.compile(".*\.py$")`

Définition à la ligne 9 du fichier test3.py.

#### 6.25.1.6 `src.test3.python3safe = True`

Définition à la ligne 5 du fichier test3.py.

#### 6.25.1.7 `list src.test3.safe = []`

Définition à la ligne 11 du fichier test3.py.

## 6.26 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`
- string `dependences` = "python3-dbus python3-dbus.mainloop.qt"
- string `python3safe` = "True"
- tuple `machin` = `Available()`

### 6.26.1 Documentation des variables

#### 6.26.1.1 `string src.usbDisk.dependences = "python3-dbus python3-dbus.mainloop.qt"`

Définition à la ligne 26 du fichier usbDisk.py.

## 6.26.1.2 dictionary src.usbDisk.licence = {}

Définition à la ligne 4 du fichier usbDisk.py.

## 6.26.1.3 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@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 version3 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.

## 6.26.1.4 tuple src.usbDisk.machin = Available()

Définition à la ligne 568 du fichier usbDisk.py.

## 6.26.1.5 string src.usbDisk.python3safe = "True"

Définition à la ligne 27 du fichier usbDisk.py.

## 6.27 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, qui sert de creuset pour les classe servant aux copies et aux effacements.*
- class [threadCopyToUSB](#)  
*Classe pour les threads copiant vers les clés USB.*
- class [threadCopyFromUSB](#)  
*Classe pour les threads copiant depuis les clés USB.*
- class [threadMoveFromUSB](#)  
*Classe pour les threads déplaçant des fichiers depuis les clés USB.*
- class [threadDeletelnUSB](#)  
*Classe pour les threads effaçant des sous-arbres dans les clés USB.*

### Fonctions

- def [ensureDirExists](#)  
*force l'existence d'un répertoire, récursivement si nécessaire*
- def [test\\_copytree](#)  
*Teste la fonction copytree.*
- def [test\\_copy2](#)

Teste la copie d'un fichier vers une destination telle qu'elle est pratiquée dans la méthode copytree de [abstract-ThreadUSB](#).

## Variables

- string `licenceEn`
- int `_threadNumber` = 0

## 6.27.1 Documentation des fonctions

### 6.27.1.1 def `src.usbThread.ensureDirExists` ( `destpath` )

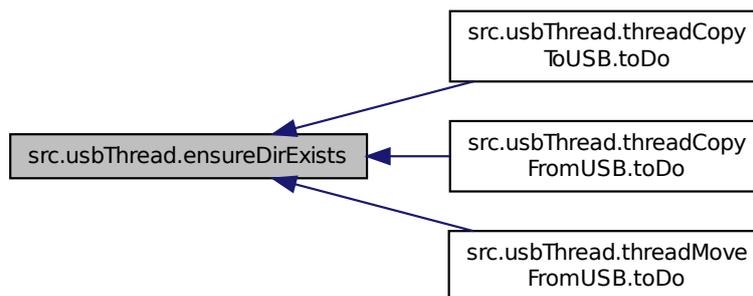
force l'existence d'un répertoire, récursivement si nécessaire

#### Paramètres

<code>destpath</code>	le chemin de ce répertoire
-----------------------	----------------------------

Définition à la ligne 35 du fichier `usbThread.py`.

Voici le graphe des appelants de cette fonction :



### 6.27.1.2 def `src.usbThread.test_copy2` ( )

Teste la copie d'un fichier vers une destination telle qu'elle est pratiquée dans la méthode copytree de [abstract-ThreadUSB](#).

Définition à la ligne 594 du fichier `usbThread.py`.

### 6.27.1.3 def `src.usbThread.test_copytree` ( )

Teste la fonction copytree.

Définition à la ligne 577 du fichier `usbThread.py`.

## 6.27.2 Documentation des variables

### 6.27.2.1 int `src.usbThread._threadNumber` = 0

Définition à la ligne 28 du fichier `usbThread.py`.

### 6.27.2.2 string src.usbThread.licenceEn

#### Valeur initiale :

```
1 """
2     file usbThread.py
3     this file is part of the project scolasync
4
5     Copyright (C) 2010-2012 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 version3 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.

## 6.28 Référence de l'espace de nommage src.version

### Fonctions

- def [major](#)
- def [minor](#)
- def [version](#)

### Variables

- dictionary [licence](#) = {}

### 6.28.1 Documentation des fonctions

#### 6.28.1.1 def src.version.major ( )

#### Renvoie

le numéro majeur de version

Définition à la ligne 29 du fichier version.py.

Voici le graphe des appelants de cette fonction :



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

#### Renvoie

le numéro mineur de version

Définition à la ligne 36 du fichier `version.py`.

Voici le graphe des appelants de cette fonction :



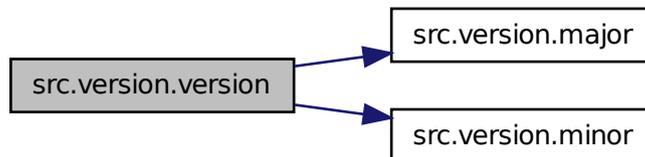
### 6.28.1.3 `def src.version.version ( )`

#### Renvoie

l'identifiant de la version

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

Voici le graphe d'appel pour cette fonction :



## 6.28.2 Documentation des variables

### 6.28.2.1 `dictionary src.version.licence = {}`

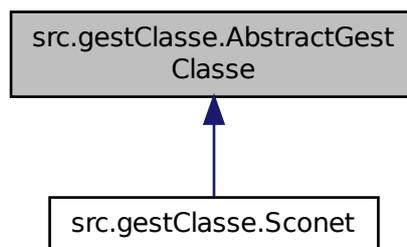
Définition à la ligne 4 du fichier `version.py`.

# Chapitre 7

## Documentation des classes

### 7.1 Référence de la classe `src.gestClasse.AbstractGestClasse`

Graphe d'héritage de `src.gestClasse.AbstractGestClasse` :



#### Fonctions membres publiques

- def `__init__`  
*le constructeur*
- def `collectClasses`
- def `elevésDeClasse`
- def `unique_name`
- def `showable_name`

#### 7.1.1 Description détaillée

Définition à la ligne 36 du fichier `gestClasse.py`.

#### 7.1.2 Documentation des constructeurs et destructeur

7.1.2.1 `def src.gestClasse.AbstractGestClasse.__init__( self, f )`

le constructeur

## Paramètres

<i>f</i>	le nom d'un fichier, ou un fichier ouvert en lecture qui contient les données permettant la gestion des classes d'un établissement scolaire
----------	---

Réimplémentée dans [src.gestClasse.Sconet](#).

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

### 7.1.3 Documentation des fonctions membres

#### 7.1.3.1 `def src.gestClasse.AbstractGestClasse.collectClasses ( self )`

## Renvoie

une liste de noms de classes d'un établissement scolaire

Réimplémentée dans [src.gestClasse.Sconet](#).

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

#### 7.1.3.2 `def src.gestClasse.AbstractGestClasse.elevesDeClasse ( self, cl )`

## Paramètres

<i>cl</i>	une classe dans un établissement scolaire
-----------	---

## Renvoie

une liste d'élèves (sous forme d'objets)

Réimplémentée dans [src.gestClasse.Sconet](#).

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

#### 7.1.3.3 `def src.gestClasse.AbstractGestClasse.showable_name ( self, el )`

## Paramètres

<i>el</i>	un objet élève
-----------	----------------

## Renvoie

une chaîne unicode, pour nommer l'élève

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

#### 7.1.3.4 `def src.gestClasse.AbstractGestClasse.unique_name ( self, el )`

## Paramètres

<i>el</i>	un objet élève
-----------	----------------

## Renvoie

une chaîne unicode, unique dans l'établissement

Définition à la ligne 67 du fichier `gestClasse.py`.

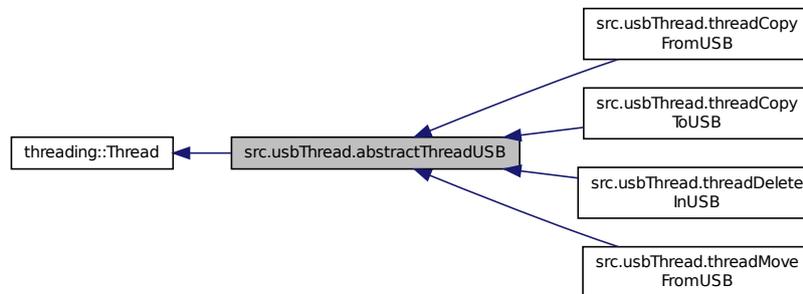
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/gestClasse.py](#)

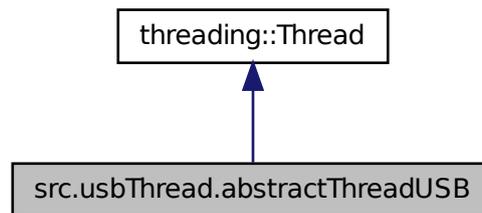
## 7.2 Référence de la classe src.usbThread.abstractThreadUSB

Une classe abstraite, qui sert de creuset pour les classe servant aux copies et aux effacements.

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 `run`
- def `writeToLog`  
*Écrit un message dans le fichier de journalisation.*
- def `copytree`  
*Une version modifiée de `shutil.copytree` qui accepte que les répertoires destination soient déjà existants.*
- def `__str__`  
*Renvoie une chaîne informative sur le thread.*
- def `threadType`  
*information sur le thread.*
- def `todo`  
*La fonction abstraite pour les choses à faire.*

### Attributs publics

- `ud`
- `fileList`
- `subdir`

- [dest](#)
- [logfile](#)
- [parent](#)

### 7.2.1 Description détaillée

Une classe abstraite, qui sert de creuset pour les classe servant aux copies et aux effacements.

Les classes filles doivent redéfinir la méthode **toDo** : c'est celle qui est démarrée quand le thread est lancé. Cette méthode est appelée dans le contexte « **with** ud.rlock », qui évite que deux threads en même temps ne cherchent à accéder au même média.

Une méthode **copytree** est définie pour remplacer shutil.copytree qui ne fait pas tout à fait l'affaire.

Définition à la ligne 150 du fichier usbThread.py.

### 7.2.2 Documentation des constructeurs et destructeur

7.2.2.1 `def src.usbThread.abstractThreadUSB.__init__( self, ud, fileList, subdir, dest=None, logfile = "/dev/null", parent=None )`

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
<i>parent</i>	un widget qui recevra de signaux en début et en fin d'exécution

Définition à la ligne 164 du fichier usbThread.py.

### 7.2.3 Documentation des fonctions membres

7.2.3.1 `def src.usbThread.abstractThreadUSB.__str__( self )`

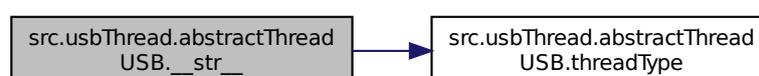
Renvoie une chaîne informative sur le thread.

#### Renvoie

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

Définition à la ligne 252 du fichier usbThread.py.

Voici le graphe d'appel pour cette fonction :



```
7.2.3.2 def src.usbThread.abstractThreadUSB.copytree ( self, src, dst, symlinks = False, ignore = None, erase =
False, errors = [] )
```

Une version modifiée de `shutil.copytree` qui accepte que les répertoires destination soient déjà existants.

Cette source dérive de la documentation fournie avec Python 2.7

#### Paramètres

<code>src</code>	un nom de fichier ou de répertoire
<code>dst</code>	un nom de de répertoire (déjà existant ou à créer)
<code>symlinks</code>	vrai si on veut recopier les liens tels quels
<code>ignore</code>	une fonction qui construit une liste de fichiers à ignorer (profil : répertoire, liste de noms de fichiers -> liste de noms de fichiers à ignorer)
<code>erase</code>	s'il est vrai la source est effacée après copie réussie
<code>errors</code>	la liste d'erreurs déjà relevées jusque là

#### Renvoie

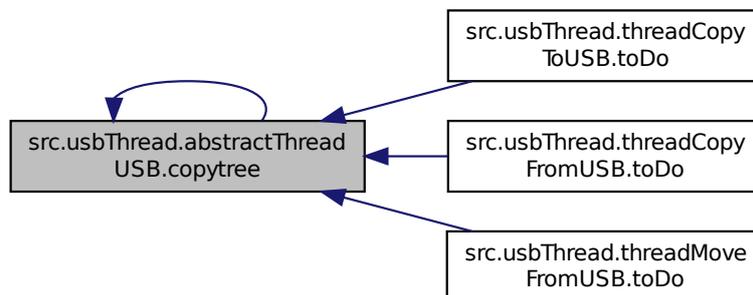
une liste d'erreurs éventuellement relevées, sinon une liste vide

Définition à la ligne 201 du fichier `usbThread.py`.

Voici le graphe d'appel pour cette fonction :



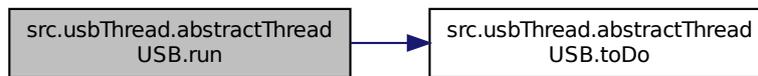
Voici le graphe des appelants de cette fonction :



```
7.2.3.3 def src.usbThread.abstractThreadUSB.run ( self )
```

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

Voici le graphe d'appel pour cette fonction :



#### 7.2.3.4 def src.usbThread.abstractThreadUSB.threadType ( self )

information sur le thread.

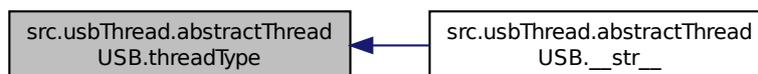
##### Renvoie

une chaîne courte qui informe sur le type de thread

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

Définition à la ligne 267 du fichier usbThread.py.

Voici le graphe des appelants de cette fonction :



#### 7.2.3.5 def src.usbThread.abstractThreadUSB.todo ( self, ud, fileList, subdir, dest, logfile )

La fonction abstraite pour les choses à faire.

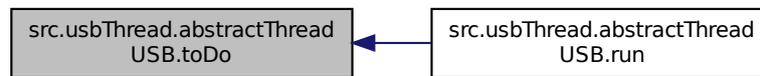
##### Paramètres

<i>ud</i>	l'instance 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
<i>logfile</i>	un fichier de journalisation

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

Définition à la ligne 279 du fichier usbThread.py.

Voici le graphe des appelants de cette fonction :



#### 7.2.3.6 `def src.usbThread.abstractThreadUSB.writeToLog ( self, msg )`

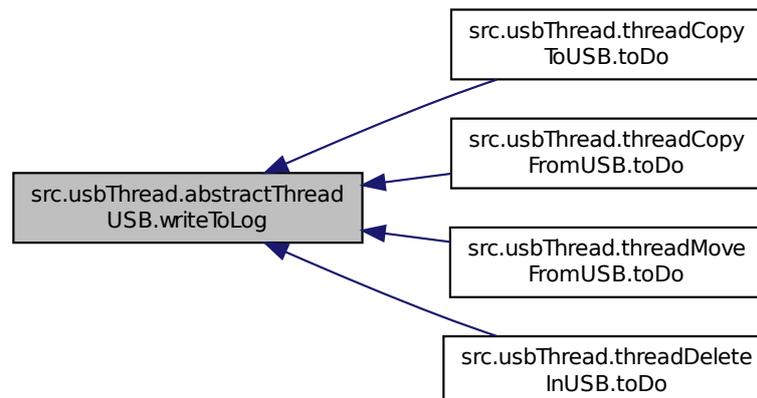
Écrit un message dans le fichier de journalisation.

##### Paramètres

<i>msg</i>	le message
------------	------------

Définition à la ligne 184 du fichier `usbThread.py`.

Voici le graphe des appelants de cette fonction :



## 7.2.4 Documentation des données membres

### 7.2.4.1 `src.usbThread.abstractThreadUSB.dest`

Définition à la ligne 171 du fichier `usbThread.py`.

### 7.2.4.2 `src.usbThread.abstractThreadUSB.fileList`

Définition à la ligne 169 du fichier `usbThread.py`.

#### 7.2.4.3 `src.usbThread.abstractThreadUSB.logfile`

Définition à la ligne 172 du fichier `usbThread.py`.

#### 7.2.4.4 `src.usbThread.abstractThreadUSB.parent`

Définition à la ligne 173 du fichier `usbThread.py`.

#### 7.2.4.5 `src.usbThread.abstractThreadUSB.subdir`

Définition à la ligne 170 du fichier `usbThread.py`.

#### 7.2.4.6 `src.usbThread.abstractThreadUSB.ud`

Définition à la ligne 167 du fichier `usbThread.py`.

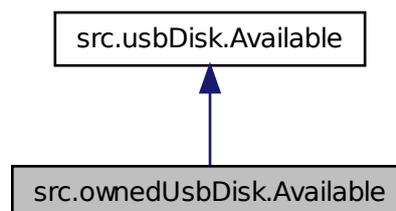
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

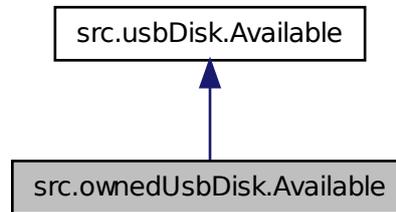
### 7.3 Référence de la classe `src.ownedUsbDisk.Available`

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

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



Graphes de collaboration de src.ownedUsbDisk.Available :



### Fonctions membres publiques

- def `__init__`  
*Le constructeur est un proxy pour `usbDisk.Available.__init__` qui force la classe de disques à utiliser : en effet ici `uDisk` désigne `ownedUsbDisk.uDisk`.*
- def `finishInit`  
*Fin de l'initialisation : trouve les propriétaires des disques puis identifie les partitions FAT et les monte.*

### Attributs publics

- `noLoop`

#### 7.3.1 Description détaillée

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

Les propriétaires sont recensés juste avant le montage des partitions FAT.

Définition à la ligne 224 du fichier `ownedUsbDisk.py`.

#### 7.3.2 Documentation des constructeurs et destructeur

7.3.2.1 `def src.ownedUsbDisk.Available.__init__( self, access = "disk", diskClass = uDisk, diskDict = None, noLoop = True )`

Le constructeur est un proxy pour `usbDisk.Available.__init__` qui force la classe de disques à utiliser : en effet ici `uDisk` désigne `ownedUsbDisk.uDisk`.

##### Paramètres

<code>access</code>	le mode d'accès : 'disk' ou 'firstFat'
<code>diskClass</code>	la classe d'objets à créer pour chaque disque
<code>diskDict</code>	un dictionnaire des disques maintenu par <code>deviceListener</code>
<code>noLoop</code>	doit être True pour éviter de lancer un dialogue

Définition à la ligne 236 du fichier `ownedUsbDisk.py`.

#### 7.3.3 Documentation des fonctions membres

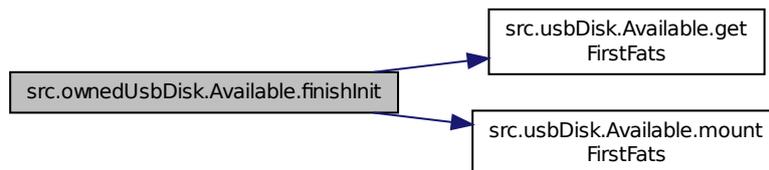
### 7.3.3.1 def src.ownedUsbDisk.Available.finishInit ( self )

Fin de l'initialisation : trouve les propriétaires des disques puis identifie les partitions FAT et les monte.

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

Définition à la ligne 245 du fichier ownedUsbDisk.py.

Voici le graphe d'appel pour cette fonction :



## 7.3.4 Documentation des données membres

### 7.3.4.1 src.ownedUsbDisk.Available.noLoop

Définition à la ligne 237 du fichier ownedUsbDisk.py.

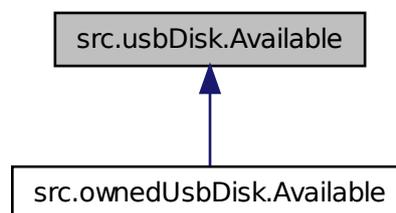
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/ownedUsbDisk.py](#)

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

une classe pour représenter la collection des disques USB connectés

Graphe d'héritage de `src.usbDisk.Available` :



### Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)  
*Le constructeur.*
- def [finishInit](#)

- *Fin de l'initialisation.*
- `def mountFirstFats`  
*fabrique la liste des partitions FAT, monte les partitions FAT si elles ne le sont pas*
- `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 des effets de bord :*
- `def hasDev`

### Attributs publics

- `access`
- `bus`
- `disks`
- `enumDev`
- `firstFats`
- `fatPaths`

#### 7.4.1 Description détaillée

une classe pour représenter la collection des disques USB connectés

les attributs publics sont :

- **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 379 du fichier `usbDisk.py`.

#### 7.4.2 Documentation des constructeurs et destructeur

7.4.2.1 `def src.usbDisk.Available.__init__( self, access = "disk", diskClass = uDisk, diskDict = None )`

Le constructeur.

Paramètres

<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.
<code>diskClass</code>	la classe de disques à créer
<code>diskDict</code>	un dictionnaire des disques maintenu par <code>deviceListener</code>

Définition à la ligne 390 du fichier `usbDisk.py`.

#### 7.4.3 Documentation des fonctions membres

7.4.3.1 `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 503 du fichier `usbDisk.py`.

7.4.3.2 `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 515 du fichier `usbDisk.py`.

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

Fournit une représentation imprimable.

#### Renvoie

une représentation imprimable de la collection

Définition à la ligne 484 du fichier `usbDisk.py`.

7.4.3.4 `def src.usbDisk.Available.__trunc__( self )`

#### Renvoie

le nombre de medias connectés

Définition à la ligne 438 du fichier `usbDisk.py`.

7.4.3.5 `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 448 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

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

Permet de déterminer si un disque est dans la collection.

**Paramètres**

<code>ud</code>	une instance de <code>uDisk</code>
-----------------	------------------------------------

**Renvoie**

vrai si le `uDisk` `ud` est dans la collection

Définition à la ligne 458 du fichier `usbDisk.py`.

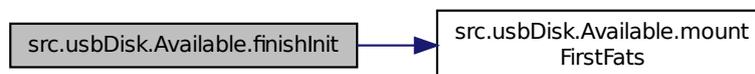
**7.4.3.7 def src.usbDisk.Available.finishInit ( self )**

Fin de l'initialisation.

Réimplémentée dans `src.ownedUsbDisk.Available`.

Définition à la ligne 420 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

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

Facilite l'accès aux partitions de type DOS-FAT, et a des effets de bord :

- marque le disque avec l'uuid de la première partition FAT.
- construit une liste des chemins `uDisk` des FATs

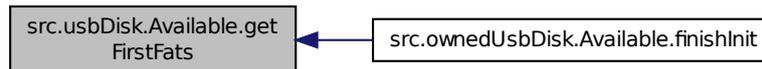
## Paramètres

<i>setOwners</i>	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 533 du fichier usbDisk.py.

Voici le graphe des appelants de cette fonction :



## 7.4.3.9 def src.usbDisk.Available.hasDev ( self, dev )

## Paramètres

<i>dev</i>	un chemin comme /org/freedesktop/UDisks/devices/sdb3
------------	--

## Renvoie

True si la partition est dans la liste des partions disponibles

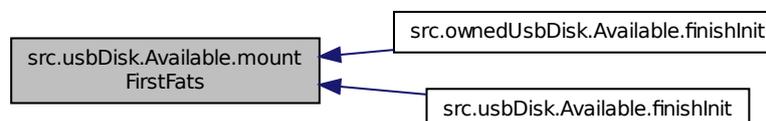
Définition à la ligne 558 du fichier usbDisk.py.

## 7.4.3.10 def src.usbDisk.Available.mountFirstFats ( self )

fabrique la liste des partitions FAT, monte les partitions FAT si elles ne le sont pas

Définition à la ligne 428 du fichier usbDisk.py.

Voici le graphe des appelants de cette fonction :



## 7.4.3.11 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 468 du fichier `usbDisk.py`.

Voici le graphe des appelants de cette fonction :

**7.4.4 Documentation des données membres****7.4.4.1 `src.usbDisk.Available.access`**

Définition à la ligne 391 du fichier `usbDisk.py`.

**7.4.4.2 `src.usbDisk.Available.bus`**

Définition à la ligne 392 du fichier `usbDisk.py`.

**7.4.4.3 `src.usbDisk.Available.disks`**

Définition à la ligne 396 du fichier `usbDisk.py`.

**7.4.4.4 `src.usbDisk.Available.enumDev`**

Définition à la ligne 397 du fichier `usbDisk.py`.

**7.4.4.5 `src.usbDisk.Available.fatPaths`**

Définition à la ligne 535 du fichier `usbDisk.py`.

**7.4.4.6 `src.usbDisk.Available.firstFats`**

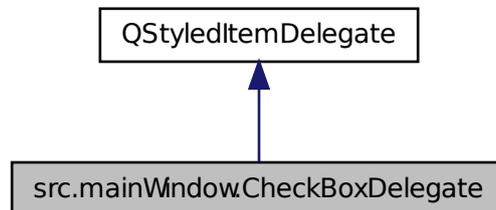
Définition à la ligne 429 du fichier `usbDisk.py`.

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

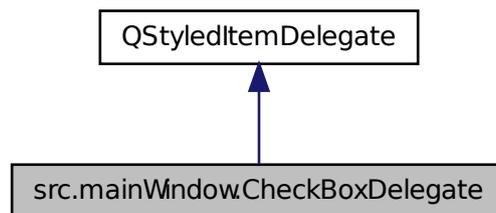
— [src/usbDisk.py](#)

## 7.5 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](#)

#### 7.5.1 Description détaillée

Définition à la ligne 868 du fichier `mainWindow.py`.

#### 7.5.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 869 du fichier `mainWindow.py`.

#### 7.5.3 Documentation des fonctions membres

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

Définition à la ligne 883 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



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

Définition à la ligne 872 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



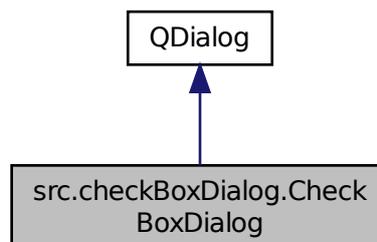
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

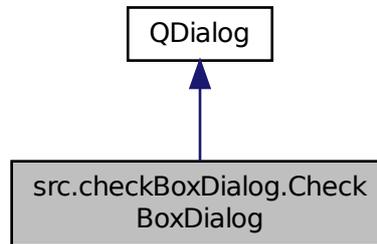
## 7.6 Référence de la classe src.checkBoxDialog.CheckBoxDialog

Un dialogue pour gérer les cases à cocher de l'application.

Graphe d'héritage de src.checkBoxDialog.CheckBoxDialog :



Graphe de collaboration de `src.checkBoxDialog.CheckBoxDialog` :



### Fonctions membres publiques

- def `__init__`  
*Le constructeur.*
- def `all`  
*Fait cocher tous les baladeurs.*
- def `toggle`  
*Fait inverser tous les boutons.*
- def `none`  
*Fait décocher tous les boutons.*
- def `esc`  
*termine le dialogue sans rien faire*

### Attributs publics

- `mainWindow`
- `ui`

#### 7.6.1 Description détaillée

Un dialogue pour gérer les cases à cocher de l'application.

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

#### 7.6.2 Documentation des constructeurs et destructeur

7.6.2.1 `def src.checkBoxDialog.CheckBoxDialog.__init__( self, parent = None )`

Le constructeur.

##### Paramètres

<i>parent</i>	un <code>mainWindow</code> , qui est censé contenir des données
---------------	---

Définition à la ligne 40 du fichier `checkBoxDialog.py`.

#### 7.6.3 Documentation des fonctions membres

#### 7.6.3.1 `def src.checkBoxDialog.CheckBoxDialog.all ( self )`

Fait cocher tous les baladeurs.

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

#### 7.6.3.2 `def src.checkBoxDialog.CheckBoxDialog.esc ( self )`

termine le dialogue sans rien faire

Définition à la ligne 78 du fichier `checkBoxDialog.py`.

#### 7.6.3.3 `def src.checkBoxDialog.CheckBoxDialog.none ( self )`

Fait décocher tous les boutons.

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

#### 7.6.3.4 `def src.checkBoxDialog.CheckBoxDialog.toggle ( self )`

Fait inverser tous les boutons.

Définition à la ligne 62 du fichier `checkBoxDialog.py`.

### 7.6.4 Documentation des données membres

#### 7.6.4.1 `src.checkBoxDialog.CheckBoxDialog.mainWindow`

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

#### 7.6.4.2 `src.checkBoxDialog.CheckBoxDialog.ui`

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

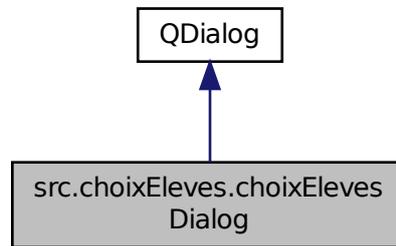
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/checkBoxDialog.py](#)

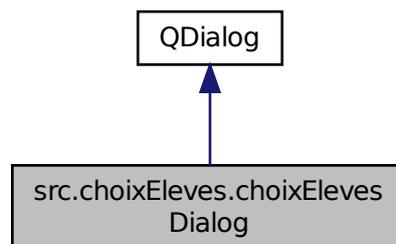
## 7.7 Référence de la classe `src.choixElevés.choixElevésDialog`

implémente un dialogue permettant de choisir des élèves les propriétés importantes sont `self.ok`, vrai si on doit prendre en compte la liste sélectionnée, et le contenu de la liste des sélectionnés, dont on peut récupérer les élèves un par un à l'aide de `self.pop()`

Graphe d'héritage de src.choixElevés.choixElevésDialog :



Graphe de collaboration de src.choixElevés.choixElevésDialog :



## Fonctions membres publiques

- def `__init__`  
*le constructeur récupérer des données SCONET*
- def `fichierElevés`  
*choisit et ouvre un nouveau fichiers d'élèves*
- def `connecteGestionnaire`  
*met en place l'arbre des noms d'élèves*
- def `checkNum`  
*fonction de rappel utilisée quand on coche/décoche la case pour prendre en compte le numéro*
- def `replie`  
*replie toutes les classes du dialogue*
- def `coche`  
*coche toutes les cases d'élèves visibles*
- def `decoche`  
*décoche toutes les cases d'élèves, visibles ou cachées*
- def `updateParentIcon`  
*Met à jour l'icône du bouton d'activation dans l'application parente pour refléter la présence d'éléments dans la liste.*
- def `addToList`  
*ajoute les élèves cochés dans la liste (s'ils n'y sont pas déjà)*
- def `dellnList`  
*retire les élèves de la liste quand ils y sont sélectionnés*
- def `pop`  
*renvoie et supprime le premier élément de la liste de noms ; si cette liste est vide, renvoie None*

- def `itemStrings`
- def `takeltem`  
*retire un item de la liste et le renvoie (pourvu qu'il y existe)*
- def `valid`  
*Prend acte de la validation.*
- def `escape`  
*Prend acte de l'abandon ; supprime les éléments de la liste et ferme le dialogue.*
- def `listeChoix`
- def `listeUnique_Names`

### Attributs publics

- `ok`
- `ui`
- `prefs`
- `gestionnaire`

#### 7.7.1 Description détaillée

implémente un dialogue permettant de choisir des élèves les propriétés importantes sont `self.ok`, vrai si on doit prendre en compte la liste sélectionnée, et le contenu de la liste des sélectionnés, dont on peut récupérer les élèves un par un à l'aide de `self.pop()`

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

#### 7.7.2 Documentation des constructeurs et destructeur

7.7.2.1 `def src.choixEleves.choixElevesDialog.__init__( self, parent = None, gestionnaire = gestClasse.Sconet )`

le constructeur récupérer des données SCONET

##### Paramètres

<code>parent</code>	le widget parent
<code>gestionnaire</code>	le système censé gérer les données du fichier f

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

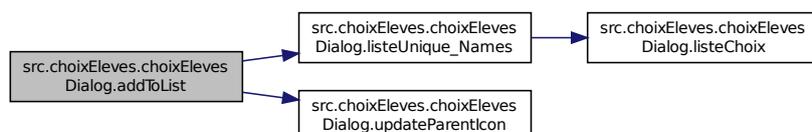
#### 7.7.3 Documentation des fonctions membres

7.7.3.1 `def src.choixEleves.choixElevesDialog.addToList ( self )`

ajoute les élèves cochés dans la liste (s'ils n'y sont pas déjà)

Définition à la ligne 157 du fichier `choixEleves.py`.

Voici le graphe d'appel pour cette fonction :



### 7.7.3.2 `def src.choixEleves.choixElevesDialog.checkNum ( self, state )`

fonction de rappel utilisée quand on coche/décoche la case pour prendre en compte le numéro

#### Paramètres

<code>state</code>	: l'état coché ou décoché
--------------------	---------------------------

Définition à la ligne 110 du fichier choixEleves.py.

### 7.7.3.3 `def src.choixEleves.choixElevesDialog.coche ( self )`

coche toutes les cases d'élèves visibles

Définition à la ligne 129 du fichier choixEleves.py.

### 7.7.3.4 `def src.choixEleves.choixElevesDialog.connecteGestionnaire ( self, renew = False )`

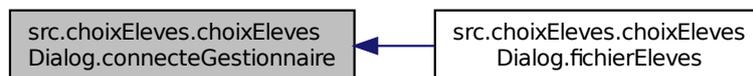
met en place l'arbre des noms d'élèves

#### Paramètres

<code>renew</code>	vrai si on veut vider tout l'arbre et recommencer
--------------------	---

Définition à la ligne 93 du fichier choixEleves.py.

Voici le graphe des appelants de cette fonction :



### 7.7.3.5 `def src.choixEleves.choixElevesDialog.decoche ( self )`

décoche toutes les cases d'élèves, visibles ou cachées

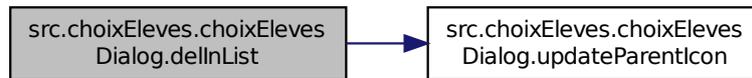
Définition à la ligne 138 du fichier choixEleves.py.

### 7.7.3.6 `def src.choixEleves.choixElevesDialog.dellnList ( self )`

retire les élèves de la liste quand ils y sont sélectionnés

Définition à la ligne 168 du fichier choixEleves.py.

Voici le graphe d'appel pour cette fonction :

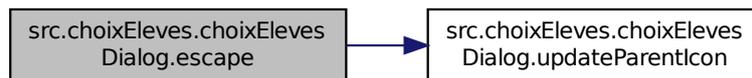


#### 7.7.3.7 `def src.choixElevés.choixElevésDialog.escape ( self )`

Prend acte de l'abandon ; supprime les éléments de la liste et ferme le dialogue.

Définition à la ligne 239 du fichier `choixElevés.py`.

Voici le graphe d'appel pour cette fonction :

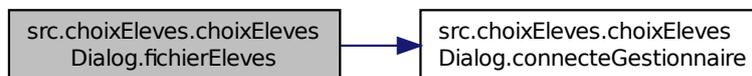


#### 7.7.3.8 `def src.choixElevés.choixElevésDialog.fichierElevés ( self )`

choisit et ouvre un nouveau fichiers d'élèves

Définition à la ligne 78 du fichier `choixElevés.py`.

Voici le graphe d'appel pour cette fonction :



#### 7.7.3.9 `def src.choixElevés.choixElevésDialog.itemStrings ( self )`

Renvoie

une liste des chaînes contenues dans les items

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

### 7.7.3.10 `def src.choixElevés.choixElevésDialog.listeChoix ( self )`

#### Renvoie

la liste de `QStandardItem`s sélectionnés

Définition à la ligne 251 du fichier `choixElevés.py`.

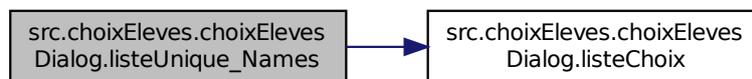
Voici le graphe des appelants de cette fonction :



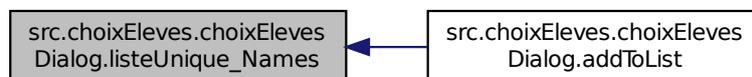
### 7.7.3.11 `def src.choixElevés.choixElevésDialog.listeUnique_Names ( self )`

Définition à la ligne 254 du fichier `choixElevés.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 7.7.3.12 `def src.choixElevés.choixElevésDialog.pop ( self )`

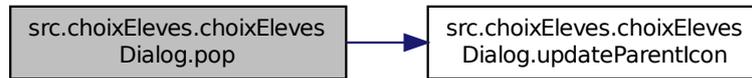
renvoie et supprime le premier élément de la liste de noms ; si cette liste est vide, renvoie `None`

**Renvoie**

un nom pour un baladeur, sinon None

Définition à la ligne 184 du fichier choixEleves.py.

Voici le graphe d'appel pour cette fonction :

**7.7.3.13 def src.choixEleves.choixElevesDialog.replie ( self )**

replie toutes les classes du dialogue

Définition à la ligne 121 du fichier choixEleves.py.

**7.7.3.14 def src.choixEleves.choixElevesDialog.takeltem ( self, item )**

retire un item de la liste et le renvoie (pourvu qu'il y existe)

**Paramètres**

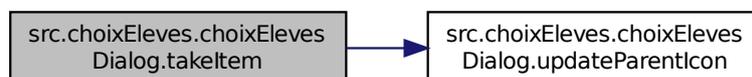
<i>une</i>	chaîne donnant le texte d'un item à trouver
------------	---

**Renvoie**

un nom pour un baladeur, sinon None

Définition à la ligne 213 du fichier choixEleves.py.

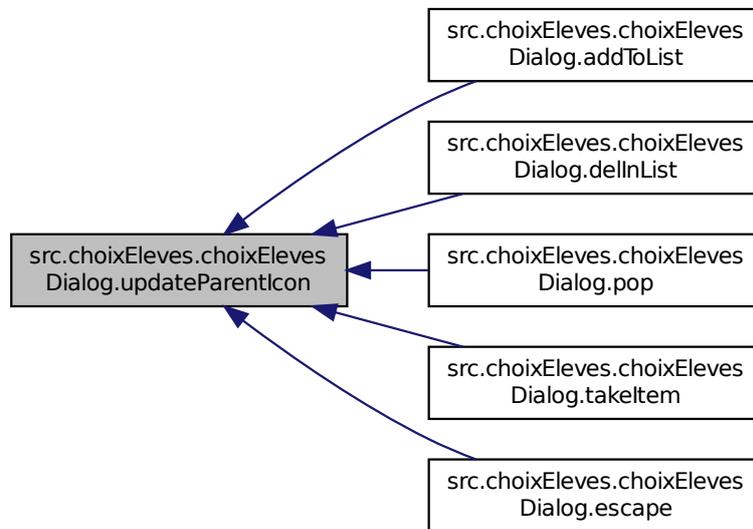
Voici le graphe d'appel pour cette fonction :

**7.7.3.15 def src.choixEleves.choixElevesDialog.updateParentIcon ( self )**

Met à jour l'icône du bouton d'activation dans l'application parente pour refléter la présence d'éléments dans la liste.

Définition à la ligne 148 du fichier choixEleves.py.

Voici le graphe des appelants de cette fonction :



7.7.3.16 `def src.choixElevés.choixElevésDialog.valid ( self )`

Prend acte de la validation.

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

## 7.7.4 Documentation des données membres

7.7.4.1 `src.choixElevés.choixElevésDialog.gestionnaire`

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

7.7.4.2 `src.choixElevés.choixElevésDialog.ok`

Définition à la ligne 53 du fichier `choixElevés.py`.

7.7.4.3 `src.choixElevés.choixElevésDialog.prefs`

Définition à la ligne 56 du fichier `choixElevés.py`.

7.7.4.4 `src.choixElevés.choixElevésDialog.ui`

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

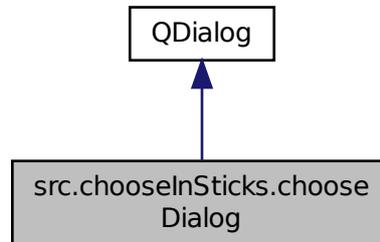
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/choixElevés.py](#)

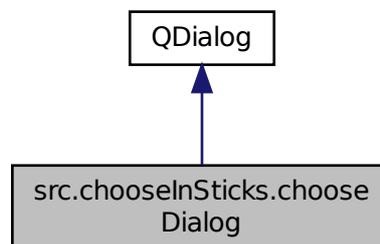
## 7.8 Référence de la classe src.chooseInSticks.chooseDialog

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 `checkValues`  
*fonction de rappel liée au bouton de validation, vérifie s'il y a bien au moins un fichier ou un répertoire sélectionné*
- 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 ; dans le cas de chemins sans jokers (caractères \* ou ?), les chemins sont protégés par des guillemets, afin qu'ils soient adaptés à un shell POSIX.*

### Attributs publics

- `mainWindow`
- `okButton`  
*mise en place des titres personnalisés*
- `ownedUsbDictionary`  
*peuplement de la zone des noms de baladeurs*
- `ok`

### 7.8.1 Description détaillée

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

Définition à la ligne 36 du fichier `chooseInSticks.py`.

### 7.8.2 Documentation des constructeurs et destructeur

7.8.2.1 `def src.chooseInSticks.chooseDialog...init...( self, parent=None, title1="", title2="", okPrompt="OK" )`

Le constructeur.

#### Paramètres

<code>parent</code>	un <code>mainWindow</code> , qui est censé contenir des données telles que <code>parent.workdir</code> , ...
<code>title1</code>	le titre du dialogue
<code>title2</code>	le titre pour la série de fichiers/modèles
<code>okPrompt</code>	le texte du bouton OK

Définition à la ligne 46 du fichier `chooseInSticks.py`.

### 7.8.3 Documentation des fonctions membres

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

Fonction de rappel quand un item de la liste est activé

#### Paramètres

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

Définition à la ligne 256 du fichier `chooseInSticks.py`.

## 7.8.3.2 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 290 du fichier chooseInSticks.py.

Voici le graphe des appelants de cette fonction :



## 7.8.3.3 def src.chooseInSticks.chooseDialog.baseDir ( self )

## Renvoi

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 163 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

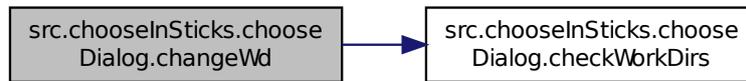


## 7.8.3.4 def src.chooseInSticks.chooseDialog.changeWd ( self )

changement du répertoire de travail

Définition à la ligne 197 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



#### 7.8.3.5 def src.chooseInSticks.chooseDialog.checkValues ( self )

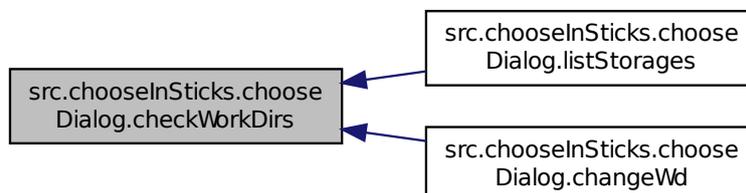
fonction de rappel liée au bouton de validation, vérifie s'il y a bien au moins un fichier ou un répertoire sélectionné  
Définition à la ligne 92 du fichier chooseInSticks.py.

#### 7.8.3.6 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 122 du fichier chooseInSticks.py.

Voici le graphe des appelants de cette fonction :



#### 7.8.3.7 def src.chooseInSticks.chooseDialog.choose ( self, kind = "file" )

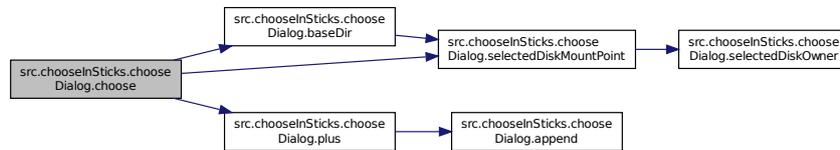
Facilite le choix de motifs de fichiers en recherchant dans les clés USB, modifie l'éditeur de ligne de texte et place le fichier choisi dans la liste.

##### Paramètres

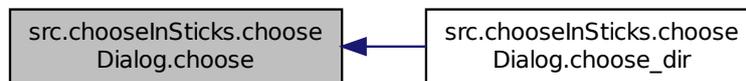
<i>kind</i>	type d'élément à choisir : "file" pour un fichier, "dir" pour un répertoire
-------------	---

Définition à la ligne 210 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

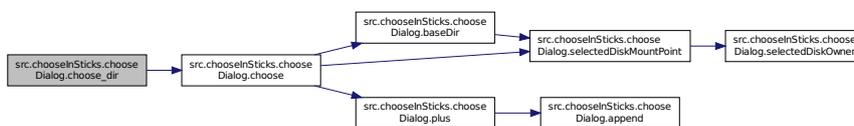


#### 7.8.3.8 `def src.chooseInSticks.chooseDialog.choose_dir ( self )`

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 248 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :

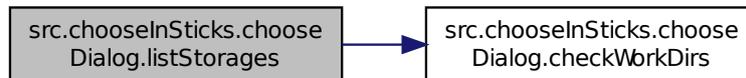


#### 7.8.3.9 `def src.chooseInSticks.chooseDialog.listStorages ( self )`

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 101 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :



#### 7.8.3.10 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 274 du fichier chooseInSticks.py.

#### 7.8.3.11 def src.chooseInSticks.chooseDialog.pathList ( self )

renvoie la liste des chemins sélectionnés ; dans le cas de chemins sans jokers (caractères \* ou ?), les chemins sont protégés par des guillemets, afin qu'ils soient adaptés à un shell POSIX.

#### Renvoie

une liste de chemins, sous forme de QStrings

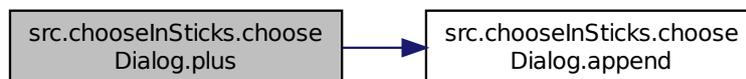
Définition à la ligne 306 du fichier chooseInSticks.py.

#### 7.8.3.12 def src.chooseInSticks.chooseDialog.plus ( self )

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

Définition à la ligne 264 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.8.3.13 `def src.chooseInSticks.chooseDialog.selectedDiskMountPoint ( self )`

## Renvoie

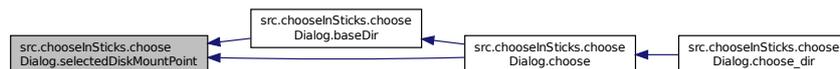
le point de montage du support sélectionné s'il y en a un

Définition à la ligne 174 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

7.8.3.14 `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 186 du fichier chooseInSticks.py.

Voici le graphe des appelants de cette fonction :



## 7.8.4 Documentation des données membres

7.8.4.1 `src.chooseInSticks.chooseDialog.mainWindow`

Définition à la ligne 48 du fichier chooseInSticks.py.

7.8.4.2 `src.chooseInSticks.chooseDialog.ok`

Définition à la ligne 84 du fichier chooseInSticks.py.

#### 7.8.4.3 src.chooseInSticks.chooseDialog.okButton

mise en place des titres personnalisés

mise en place du bouton personnalisé

Définition à la ligne 57 du fichier chooseInSticks.py.

#### 7.8.4.4 src.chooseInSticks.chooseDialog.ownedUsbDictionary

peuplement de la zone des noms de baladeurs

Définition à la ligne 73 du fichier chooseInSticks.py.

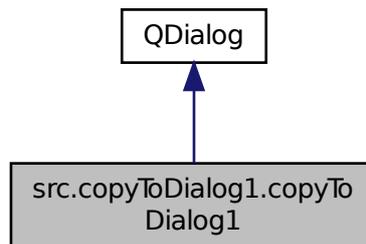
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/chooseInSticks.py](#)

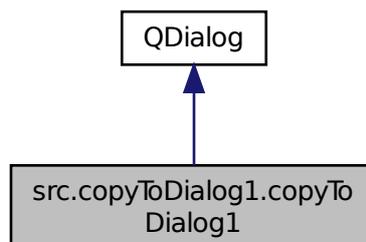
## 7.9 Référence de la classe src.copyToDialog1.copyToDialog1

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

Grappe d'héritage de src.copyToDialog1.copyToDialog1 :



Grappe 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`

### 7.9.1 Description détaillée

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

#### Paramètres

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

Définition à la ligne 39 du fichier `copyToDialog1.py`.

### 7.9.2 Documentation des fonctions membres

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

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

Définition à la ligne 76 du fichier `copyToDialog1.py`.

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

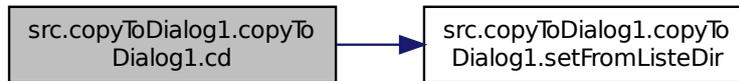
Change le répertoire courant si possible.

#### Paramètres

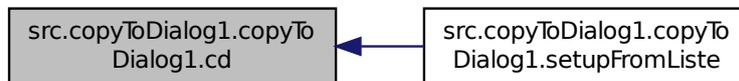
<code>ev</code>	un évènement
-----------------	--------------

Définition à la ligne 114 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 7.9.2.3 `def src.copyToDialog1.copyToDialog1.changeWd ( self )`

changement du répertoire de travail

Définition à la ligne 68 du fichier `copyToDialog1.py`.

#### 7.9.2.4 `def src.copyToDialog1.copyToDialog1.cont ( self )`

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

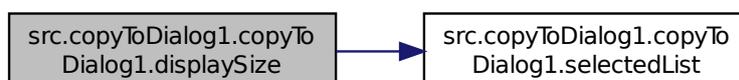
Définition à la ligne 84 du fichier `copyToDialog1.py`.

#### 7.9.2.5 `def src.copyToDialog1.copyToDialog1.displaySize ( self )`

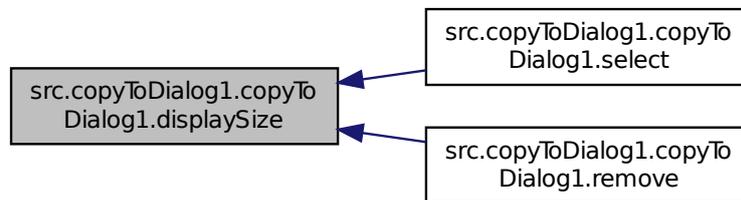
Affiche la taille de la sélection courante.

Définition à la ligne 165 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 7.9.2.6 `def src.copyToDialog1.copyToDialog1.remove ( self )`

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

Définition à la ligne 189 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :



#### 7.9.2.7 `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 145 du fichier `copyToDialog1.py`.

Voici le graphe d'appel pour cette fonction :



#### 7.9.2.8 `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 205 du fichier copyToDialog1.py.

Voici le graphe des appelants de cette fonction :

**7.9.2.9 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 103 du fichier copyToDialog1.py.

Voici le graphe des appelants de cette fonction :

**7.9.2.10 def src.copyToDialog1.copyToDialog1.setupFromListe ( self )**

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

Définition à la ligne 92 du fichier copyToDialog1.py.

Voici le graphe d'appel pour cette fonction :

**7.9.2.11 def src.copyToDialog1.copyToDialog1.setupToListe ( self )**

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

Définition à la ligne 126 du fichier `copyToDialog1.py`.

### 7.9.3 Documentation des données membres

#### 7.9.3.1 `src.copyToDialog1.copyToDialog1.mainWindow`

Définition à la ligne 47 du fichier `copyToDialog1.py`.

#### 7.9.3.2 `src.copyToDialog1.copyToDialog1.ok`

Définition à la ligne 85 du fichier `copyToDialog1.py`.

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

- [src/copyToDialog1.py](#)

## 7.10 Référence de la classe `src.deviceListener.DeviceListener`

### Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)  
*le constructeur*
- def [pollDevices](#)  
*peuple le dictionnaire `self.connectedVolumes` avec les volumes actuellement gérés par dbus.*
- def [cbAdd](#)  
*fonction de rappel pour une clé qu'on vient de brancher*
- def [cbChange](#)  
*fonction de rappel pour une clé qui vient de changer de type*
- def [cbDel](#)  
*fonction de rappel pour une clé retirée*
- def [vfatUsbPath](#)  
*Détermine le chemin UDisks pour une partition vfat connectée par USB.*
- def [identify](#)  
*Renvoie une identification de baladeur pour scolasync.*
- def [isVfatUsb](#)  
*Décide si une partition est de type vfat, et connectée par USB.*

### Attributs publics

- [bus](#)
- [manager](#)
- [interface](#)
- [connectedVolumes](#)
- [widget](#)

#### 7.10.1 Description détaillée

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

#### 7.10.2 Documentation des constructeurs et destructeur

##### 7.10.2.1 `def src.deviceListener.DeviceListener.__init__( self, widget = None )`

le constructeur

##### Paramètres

<code>widget</code>	un QWidget qui s'intéresse aux disques amovibles
---------------------	--

Définition à la ligne 36 du fichier deviceListener.py.

### 7.10.3 Documentation des fonctions membres

#### 7.10.3.1 `def src.deviceListener.DeviceListener.cbAdd ( self, path )`

fonction de rappel pour une clé qu'on vient de brancher

##### Paramètres

<i>path</i>	un chemin de type UDisks vers une partition
-------------	---

Définition à la ligne 69 du fichier deviceListener.py.

Voici le graphe d'appel pour cette fonction :



#### 7.10.3.2 `def src.deviceListener.DeviceListener.cbChange ( self, path )`

fonction de rappel pour une clé qui vient de changer de type

##### Paramètres

<i>path</i>	un chemin de type UDisks vers une partition
-------------	---

Définition à la ligne 82 du fichier deviceListener.py.

Voici le graphe d'appel pour cette fonction :



#### 7.10.3.3 `def src.deviceListener.DeviceListener.cbDel ( self, path )`

fonction de rappel pour une clé retirée

##### Paramètres

<i>path</i>	un chemin de type UDisks vers une partition
-------------	---

Définition à la ligne 96 du fichier deviceListener.py.

#### 7.10.3.4 `def src.deviceListener.DeviceListener.identify ( self, dev )`

Renvoie une identification de baladeur pour scolasync.

## Paramètres

<code>dev</code>	est un objet dbus renvoyé par <code>EnumerateDevices()</code> , ou une simple chaîne de caractères, clé dans le tableau <code>connectedVolumes</code>
------------------	---

## Renvoie

un triplet (`stickId`, `tatouage`, `uuid`)

Définition à la ligne 126 du fichier `deviceListener.py`.

7.10.3.5 `def src.deviceListener.DeviceListener.isVfatUsb ( self, o )`

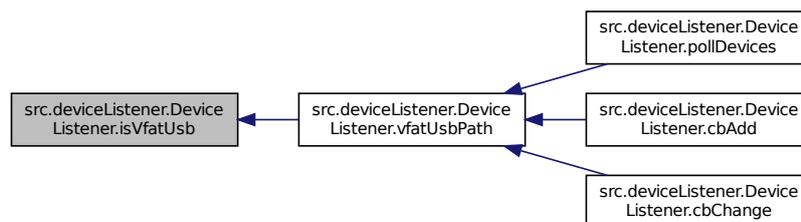
Décide si une partition est de type `vfat`, et connectée par USB.

## Paramètres

<code>o</code>	un objet dbus correspondant à une partition
----------------	---

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

Voici le graphe des appelants de cette fonction :

7.10.3.6 `def src.deviceListener.DeviceListener.pollDevices ( self )`

peuple le dictionnaire `self.connectedVolumes` avec les volumes actuellement gérés par dbus.

Définition à la ligne 56 du fichier `deviceListener.py`.

Voici le graphe d'appel pour cette fonction :

7.10.3.7 `def src.deviceListener.DeviceListener.vfatUsbPath ( self, dev )`

Détermine le chemin `UDisks` pour une partition `vfat` connectée par USB.

## Paramètres

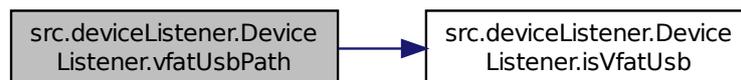
<i>dev</i>	est un objet dbus renvoyé par EnumerateDevices(), ou une simple chaîne de caractères, clé dans le tableau connectedVolumes
------------	--

## Renvoie

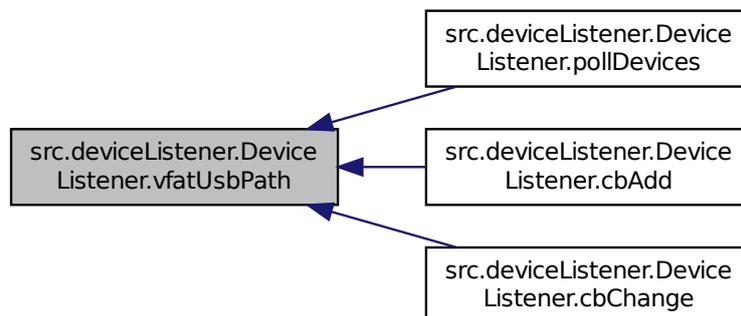
un chemin vers le disque (selon UDisks), sinon ""

Définition à la ligne 111 du fichier deviceListener.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



## 7.10.4 Documentation des données membres

### 7.10.4.1 src.deviceListener.DeviceListener.bus

Définition à la ligne 37 du fichier deviceListener.py.

### 7.10.4.2 src.deviceListener.DeviceListener.connectedVolumes

Définition à la ligne 47 du fichier deviceListener.py.

### 7.10.4.3 src.deviceListener.DeviceListener.interface

Définition à la ligne 41 du fichier deviceListener.py.

## 7.10.4.4 src.deviceListener.DeviceListener.manager

Définition à la ligne 38 du fichier deviceListener.py.

## 7.10.4.5 src.deviceListener.DeviceListener.widget

Définition à la ligne 48 du fichier deviceListener.py.

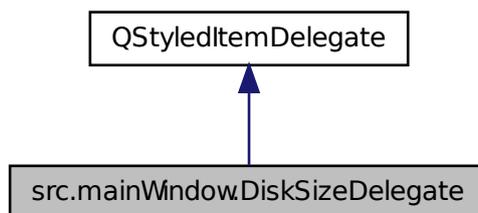
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/deviceListener.py](#)

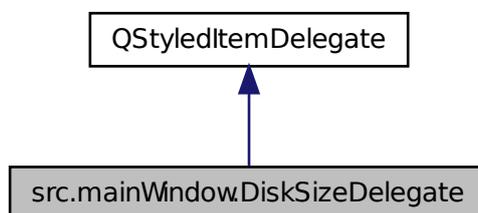
## 7.11 Référence de la classe src.mainWindow.DiskSizeDelegate

Classe pour figurer la taille de la mémoire du baladeur.

Grappe d'héritage de src.mainWindow.DiskSizeDelegate :



Grappe de collaboration de src.mainWindow.DiskSizeDelegate :



### Fonctions membres publiques

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

### 7.11.1 Description détaillée

Classe pour figurer la taille de la mémoire du baladeur.

Trace un petit secteur représentant la place occupée, puis affiche la place avec l'unité la plus parpropiée.

Définition à la ligne 935 du fichier mainWindow.py.

### 7.11.2 Documentation des constructeurs et destructeur

7.11.2.1 `def src.mainWindow.DiskSizeDelegate.__init__( self, parent )`

Définition à la ligne 936 du fichier mainWindow.py.

### 7.11.3 Documentation des fonctions membres

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

Définition à la ligne 940 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



7.11.3.2 `def src.mainWindow.DiskSizeDelegate.val2txt ( self, val )`

**Renvoie**

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

Définition à la ligne 961 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

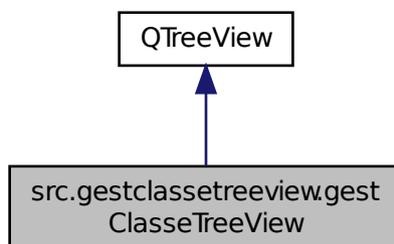


La documentation de cette classe a été générée à partir du fichier suivant :

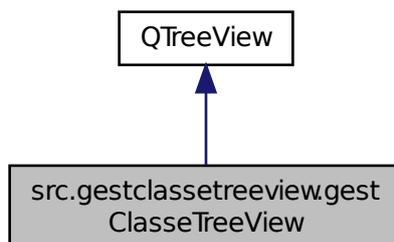
— [src/mainWindow.py](#)

## 7.12 Référence de la classe src.gestclassetreeview.gestClasseTreeView

Graphe d'héritage de src.gestclassetreeview.gestClasseTreeView :



Graphe de collaboration de src.gestclassetreeview.gestClasseTreeView :



### Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)  
*Le constructeur.*
- def [connecteGestionnaire](#)
- def [expandedItems](#)
- def [allItems](#)
- def [checkedItems](#)

### Attributs publics

- [gest](#)
- [root](#)

#### 7.12.1 Description détaillée

Définition à la ligne 32 du fichier gestclassetreeview.py.

## 7.12.2 Documentation des constructeurs et destructeur

7.12.2.1 `def src.gestclassetreeview.gestClasseTreeView.__init__( self, parent = None )`

Le constructeur.

### Paramètres

<i>parent</i>	un parent pour le widget
---------------	--------------------------

Définition à la ligne 38 du fichier `gestclassetreeview.py`.

## 7.12.3 Documentation des fonctions membres

7.12.3.1 `def src.gestclassetreeview.gestClasseTreeView.allItems ( self )`

### Revoie

la liste de tous les élèves

Définition à la ligne 91 du fichier `gestclassetreeview.py`.

7.12.3.2 `def src.gestclassetreeview.gestClasseTreeView.checkedItems ( self )`

### Revoie

la liste de tous les élèves sélectionnés

Définition à la ligne 105 du fichier `gestclassetreeview.py`.

7.12.3.3 `def src.gestclassetreeview.gestClasseTreeView.connecteGestionnaire ( self, fichier, gestionnaire = gestClasse.Sconet, renew = False )`

### Paramètres

<i>fichier</i>	le nom d'un fichier, ou un fichier ouvert en lecture, pour récupérer des données SCONET
<i>gestionnaire</i>	un gestionnaire pour exploiter les données du fichier
<i>renew</i>	vrai si on doit tout effacer avant de recommencer

Définition à la ligne 53 du fichier `gestclassetreeview.py`.

7.12.3.4 `def src.gestclassetreeview.gestClasseTreeView.expandedItems ( self )`

### Revoie

la liste des items non repliés (donc visibles)

Définition à la ligne 76 du fichier `gestclassetreeview.py`.

## 7.12.4 Documentation des données membres

7.12.4.1 `src.gestclassetreeview.gestClasseTreeView.gest`

Définition à la ligne 40 du fichier `gestclassetreeview.py`.

## 7.12.4.2 src.gestclassetreeview.gestClasseTreeView.root

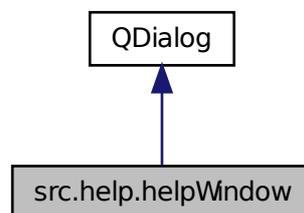
Définition à la ligne 43 du fichier gestclassetreeview.py.

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

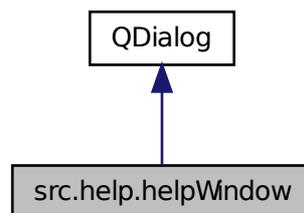
— [src/gestclassetreeview.py](#)

## 7.13 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](#)

### 7.13.1 Description détaillée

Définition à la ligne 34 du fichier help.py.

### 7.13.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Définition à la ligne 39 du fichier help.py.

### 7.13.3 Documentation des fonctions membres

7.13.3.1 `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 55 du fichier help.py.

### 7.13.4 Documentation des données membres

7.13.4.1 `src.help.helpWindow.ui`

Définition à la ligne 42 du fichier help.py.

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

— [src/help.py](#)

## 7.14 Référence de la classe src.essai.machin

### Fonctions membres publiques

— `def __init__`

### Attributs publics

— `tm`

### 7.14.1 Description détaillée

Définition à la ligne 8 du fichier essai.py.

### 7.14.2 Documentation des constructeurs et destructeur

7.14.2.1 `def src.essai.machin.__init__( self )`

Définition à la ligne 9 du fichier `essai.py`.

### 7.14.3 Documentation des données membres

7.14.3.1 `src.essai.machin.tm`

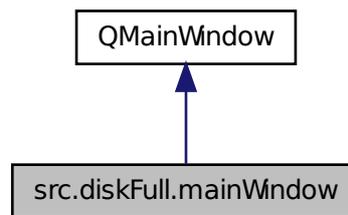
Définition à la ligne 10 du fichier `essai.py`.

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

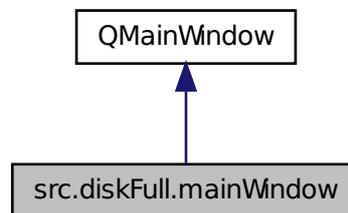
— [src/essai.py](#)

## 7.15 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](#)

### 7.15.1 Description détaillée

Définition à la ligne 31 du fichier `diskFull.py`.

### 7.15.2 Documentation des constructeurs et destructeur

7.15.2.1 `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 41 du fichier `diskFull.py`.

### 7.15.3 Documentation des données membres

7.15.3.1 `src.diskFull.mainWindow.total`

Définition à la ligne 49 du fichier `diskFull.py`.

7.15.3.2 `src.diskFull.mainWindow.ui`

Définition à la ligne 45 du fichier `diskFull.py`.

7.15.3.3 `src.diskFull.mainWindow.used`

Définition à la ligne 50 du fichier `diskFull.py`.

7.15.3.4 `src.diskFull.mainWindow.v`

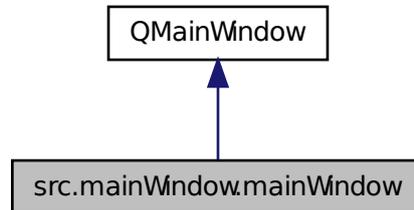
Définition à la ligne 48 du fichier `diskFull.py`.

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

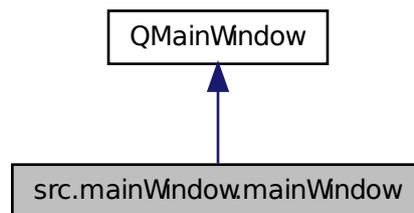
- [src/diskFull.py](#)

## 7.16 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 `checkModify`
- def `checkAll`  
*Coche tous les baladeurs.*
- def `checkToggle`  
*Inverse la coche des baladeurs.*
- def `checkNone`  
*Décoche tous les baladeurs.*
- def `namingADrive`  
*Gère un dialogue pour renommer un baladeur désigné par self.recentConnect.*
- def `deviceAdded`  
*fonction de rappel pour un medium ajouté*
- def `deviceRemoved`  
*fonction de rappel pour un medium retiré*
- def `initRedoStuff`  
*Initialise des données pour le bouton central (refaire/stopper)*
- def `showEvent`  
*modification du comportement du widget original, pour démarrer le timer et les vérifications de baladeurs après construction de la fenêtre seulement*
- def `setTimer`  
*sets the main timer*

- 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 [manageCheckBoxes](#)  
*ouvre un dialogue pour permettre de gérer les cases à cocher globalement*
- def [diskSizeData](#)
- def [diskFromOwner](#)  
*trouve le disque qui correspond à un propriétaire*
- def [editOwner](#)  
*Édition du propriétaire d'une clé.*
- def [setAvailableNames](#)  
*Met à jour l'icône qui reflète la disponibilité de noms pour renommer automatiquement des baladeurs.*
- 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 [redoCmd](#)  
*Relance la dernière commande, mais en l'appliquant seulement aux baladeurs nouvellement branchés.*
- def [namesCmd](#)  
*montre le dialogue de choix de nouveaux noms à partir d'un fichier administratif.*
- 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](#)
- [copyfromIcon](#)
- [movefromIcon](#)
- [namesFullIcon](#)
- [namesEmptyIcon](#)
- [namesFullTip](#)
- [namesEmptyTip](#)
- [namesDialog](#)
- [recentConnect](#)
- [t](#)
- [proxy](#)
- [timer](#)
- [listener](#)
- [operations](#)
- [oldThreads](#)
- [flashTimer](#)
- [checkDisksLock](#)
- [iconRedo](#)
- [iconStop](#)
- [redoToolTip](#)
- [redoStatusTip](#)
- [stopToolTip](#)
- [stopStatusTip](#)
- [schoolFile](#)

- [workdir](#)
- [refreshEnabled](#)
- [refreshDelay](#)
- [manFileLocation](#)
- [mv](#)
- [header](#)
- [availableNames](#)
- [visibleheader](#)
- [tm](#)

### 7.16.1 Description détaillée

Définition à la ligne 62 du fichier mainWindow.py.

### 7.16.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

<i>parent</i>	un QWidget
<i>locale</i>	la langue de l'application

Définition à la ligne 69 du fichier mainWindow.py.

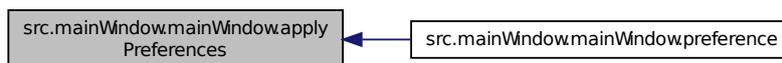
### 7.16.3 Documentation des fonctions membres

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

Applique les préférences et les options de ligne de commande.

Définition à la ligne 241 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



7.16.3.2 `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
---------------	------------------------------

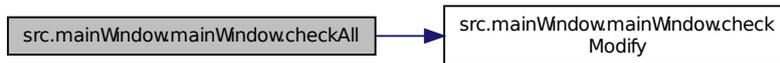
Définition à la ligne 261 du fichier mainWindow.py.

### 7.16.3.3 `def src.mainWindow.mainWindow.checkAll ( self )`

Coche tous les baladeurs.

Définition à la ligne 139 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



### 7.16.3.4 `def src.mainWindow.mainWindow.checkDisks ( self, force=False, noLoop=True )`

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

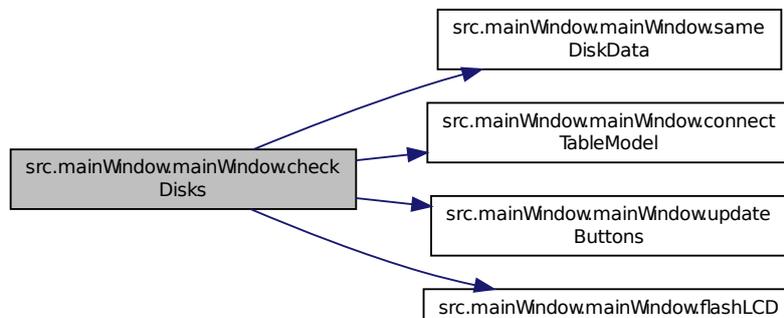
Le tableau est complètement régénéré à chaque fois, ce qui n'est pas toujours souhaitable. À la fin de chaque vérification, un court flash est déclenché sur l'afficheur de nombre de baladeurs connectés et sa valeur est mise à jour.

#### Paramètres

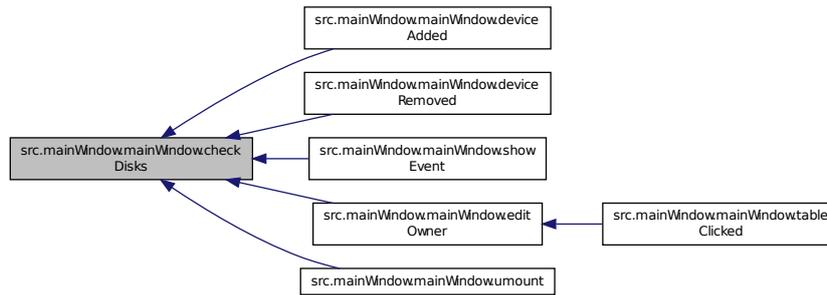
<i>force</i>	pour forcer une mise à jour du tableau
<i>noLoop</i>	si <code>False</code> , on ne rentrera pas dans une boucle de Qt

Définition à la ligne 660 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



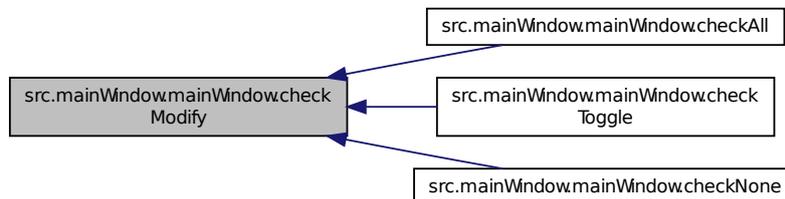
### 7.16.3.5 def src.mainWindow.mainWindow.checkModify ( self, boolFunc )

#### Paramètres

<i>boolfunc</i>	une fonction pour décider du futur état de la coche étant donné l'état antérieur Modifie les coches des baladeurs
-----------------	---

Définition à la ligne 126 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



### 7.16.3.6 def src.mainWindow.mainWindow.checkNone ( self )

Décoche tous les baladeurs.

Définition à la ligne 153 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

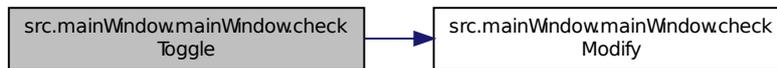


### 7.16.3.7 `def src.mainWindow.mainWindow.checkToggle ( self )`

Inverse la coche des baladeurs.

Définition à la ligne 146 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



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

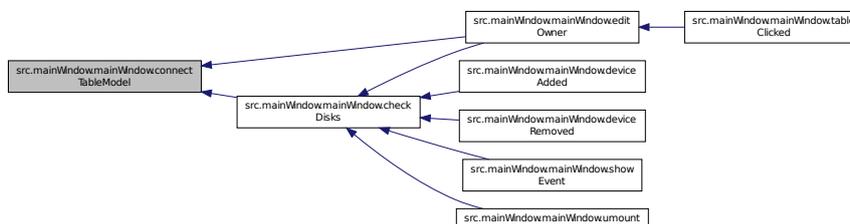
Connecte le modèle de table à la table.

Paramètres

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

Définition à la ligne 633 du fichier `mainWindow.py`.

Voici le graphe des appelants de cette fonction :



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

Lance l'action de copier depuis les clés USB.

Définition à la ligne 497 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



## 7.16.3.10 def src.mainWindow.mainWindow.copyTo ( self )

Lance l'action de copier vers les clés USB.

Définition à la ligne 473 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



## 7.16.3.11 def src.mainWindow.mainWindow.delFiles ( self )

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

Définition à la ligne 438 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



## 7.16.3.12 def src.mainWindow.mainWindow.deviceAdded ( self, s )

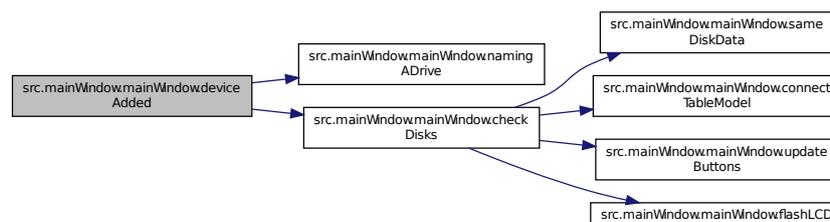
fonction de rappel pour un medium ajouté

## Paramètres

s	chemin UDisks, exemple : /org/freedesktop/UDisks/devices/sdb3
---	---

Définition à la ligne 181 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



### 7.16.3.13 `def src.mainWindow.mainWindow.deviceRemoved ( self, s )`

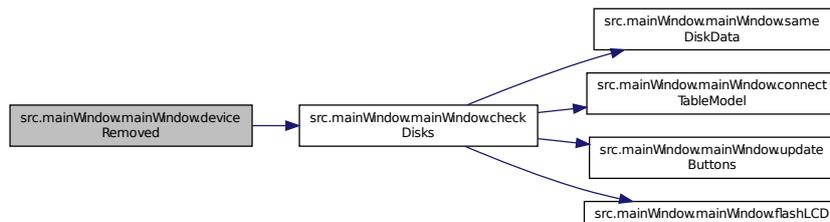
fonction de rappel pour un medium retiré

#### Paramètres

<code>s</code>	une chaine de caractères du type /dev/sdxy
----------------	--

Définition à la ligne 195 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



### 7.16.3.14 `def src.mainWindow.mainWindow.diskFromOwner ( self, student )`

trouve le disque qui correspond à un propriétaire

#### Paramètres

<code>student</code>	le propriétaire du disque
----------------------	---------------------------

#### Renvoie

le disque correspondant à l'étudiant

Définition à la ligne 328 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



### 7.16.3.15 `def src.mainWindow.mainWindow.diskSizeData ( self, rowOrDev )`

#### Paramètres

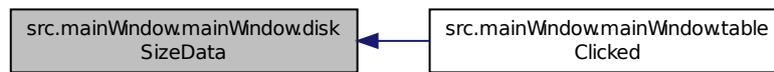
<code>rowOrDev</code>	a row number in the tableView, or a device string
-----------------------	---

#### 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 310 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



#### 7.16.3.16 def src.mainWindow.mainWindow.editOwner ( self, idx )

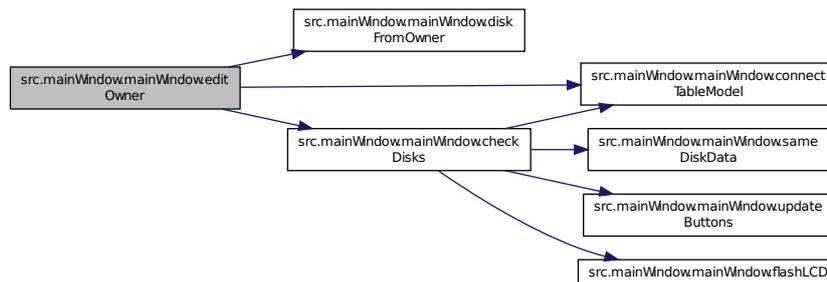
Édition du propriétaire d'une clé.

##### Paramètres

<i>idx</i>	un QModelIndex qui pointe sur le propriétaire d'une clé
------------	---

Définition à la ligne 349 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

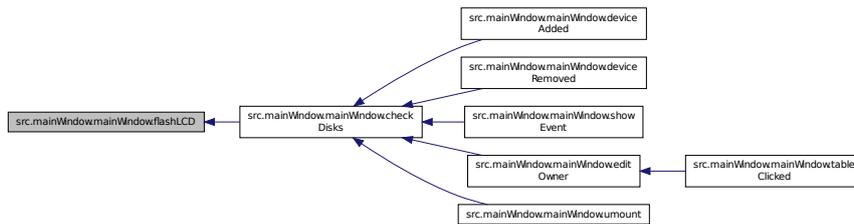


#### 7.16.3.17 def src.mainWindow.mainWindow.flashLCD ( self )

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

Définition à la ligne 695 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



#### 7.16.3.18 def src.mainWindow.mainWindow.help ( self )

Affiche le widget d'aide.

Définition à la ligne 600 du fichier mainWindow.py.

#### 7.16.3.19 def src.mainWindow.mainWindow.initRedoStuff ( self )

Initialise des données pour le bouton central (refaire/stopper)

Définition à la ligne 203 du fichier mainWindow.py.

#### 7.16.3.20 def src.mainWindow.mainWindow.manageCheckBoxes ( self )

ouvre un dialogue pour permettre de gérer les cases à cocher globalement

Définition à la ligne 299 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



#### 7.16.3.21 def src.mainWindow.mainWindow.namesCmd ( self )

montre le dialogue de choix de nouveaux noms à partir d'un fichier administratif.

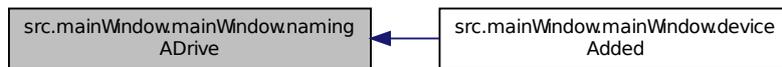
Définition à la ligne 593 du fichier mainWindow.py.

#### 7.16.3.22 def src.mainWindow.mainWindow.namingADrive ( self )

Gère un dialogue pour renommer un baladeur désigné par self.recentConnect.

Définition à la ligne 161 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



#### 7.16.3.23 def src.mainWindow.mainWindow.normalLCD ( self )

remet le style par défaut pour l'afficheur LCD

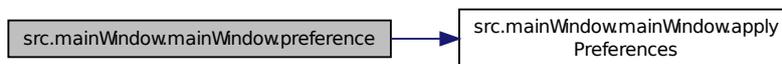
Définition à la ligne 703 du fichier mainWindow.py.

#### 7.16.3.24 def src.mainWindow.mainWindow.preference ( self )

lance le dialogue des préférences

Définition à la ligne 424 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



#### 7.16.3.25 def src.mainWindow.mainWindow.redoCmd ( self )

Relance la dernière commande, mais en l'appliquant seulement aux baladeurs nouvellement branchés.

Définition à la ligne 562 du fichier mainWindow.py.

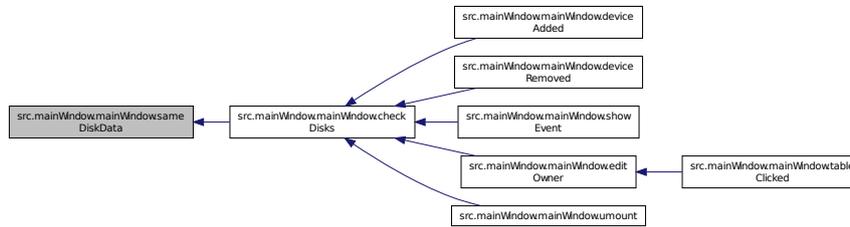
#### 7.16.3.26 def src.mainWindow.mainWindow.sameDiskData ( self, one, two )

**Renvoie**

True si les ensembles de uniqueId de one et two sont identiques

Définition à la ligne 688 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

**7.16.3.27 def src.mainWindow.mainWindow.setAvailableNames ( self, available )**

Met à jour l'icône qui reflète la disponibilité de noms pour renommer automatiquement des baladeurs.

**Paramètres**

<i>available</i>	vrai s'il y a des noms disponibles pour renommer des baladeurs.
------------------	---

Définition à la ligne 364 du fichier mainWindow.py.

**7.16.3.28 def src.mainWindow.mainWindow.setTimer ( self, enabled = True )**

sets the main timer

Définition à la ligne 231 du fichier mainWindow.py.

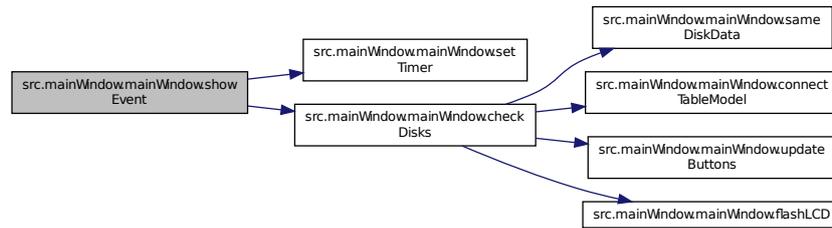
Voici le graphe des appelants de cette fonction :

**7.16.3.29 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 221 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



### 7.16.3.30 def src.mainWindow.mainWindow.tableClicked ( self, idx )

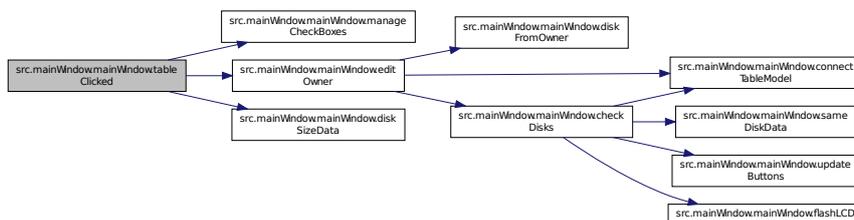
fonction de rappel pour un double clic sur un élément de la table

#### Paramètres

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

Définition à la ligne 270 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

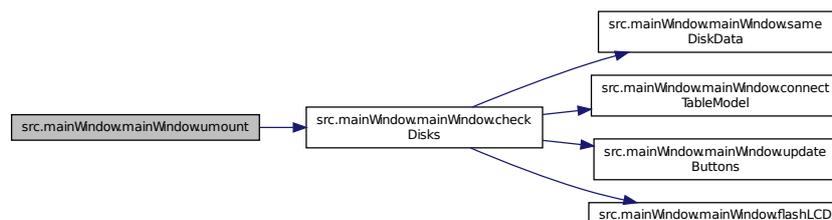


### 7.16.3.31 def src.mainWindow.mainWindow.umount ( self )

Démonte et détache les clés USB affichées.

Définition à la ligne 609 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



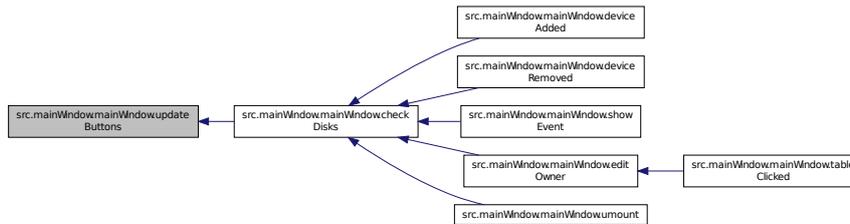
### 7.16.3.32 `def src.mainWindow.mainWindow.updateButtons ( self )`

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. Change l'icône du dialogue des noms selon qu'il reste ou non des noms disponibles dans le dialogue des noms.

Définition à la ligne 386 du fichier `mainWindow.py`.

Voici le graphe des appelants de cette fonction :



## 7.16.4 Documentation des données membres

### 7.16.4.1 `src.mainWindow.mainWindow.availableNames`

Définition à la ligne 365 du fichier `mainWindow.py`.

### 7.16.4.2 `src.mainWindow.mainWindow.checkDisksLock`

Définition à la ligne 100 du fichier `mainWindow.py`.

### 7.16.4.3 `src.mainWindow.mainWindow.copyfromIcon`

Définition à la ligne 76 du fichier `mainWindow.py`.

### 7.16.4.4 `src.mainWindow.mainWindow.flashTimer`

Définition à la ligne 98 du fichier `mainWindow.py`.

### 7.16.4.5 `src.mainWindow.mainWindow.header`

Définition à la ligne 252 du fichier `mainWindow.py`.

### 7.16.4.6 `src.mainWindow.mainWindow.iconRedo`

Définition à la ligne 205 du fichier `mainWindow.py`.

### 7.16.4.7 `src.mainWindow.mainWindow.iconStop`

Définition à la ligne 207 du fichier `mainWindow.py`.

**7.16.4.8 `src.mainWindow.mainWindow.listener`**

Définition à la ligne 92 du fichier `mainWindow.py`.

**7.16.4.9 `src.mainWindow.mainWindow.locale`**

Définition à la ligne 72 du fichier `mainWindow.py`.

**7.16.4.10 `src.mainWindow.mainWindow.manFileLocation`**

Définition à la ligne 248 du fichier `mainWindow.py`.

**7.16.4.11 `src.mainWindow.mainWindow.movefromIcon`**

Définition à la ligne 77 du fichier `mainWindow.py`.

**7.16.4.12 `src.mainWindow.mainWindow.mv`**

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

**7.16.4.13 `src.mainWindow.mainWindow.namesDialog`**

Définition à la ligne 83 du fichier `mainWindow.py`.

**7.16.4.14 `src.mainWindow.mainWindow.namesEmptyIcon`**

Définition à la ligne 80 du fichier `mainWindow.py`.

**7.16.4.15 `src.mainWindow.mainWindow.namesEmptyTip`**

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

**7.16.4.16 `src.mainWindow.mainWindow.namesFullIcon`**

Définition à la ligne 79 du fichier `mainWindow.py`.

**7.16.4.17 `src.mainWindow.mainWindow.namesFullTip`**

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

**7.16.4.18 `src.mainWindow.mainWindow.oldThreads`**

Définition à la ligne 97 du fichier `mainWindow.py`.

**7.16.4.19 `src.mainWindow.mainWindow.operations`**

Définition à la ligne 96 du fichier `mainWindow.py`.

**7.16.4.20** `src.mainWindow.mainWindow.proxy`

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

**7.16.4.21** `src.mainWindow.mainWindow.recentConnect`

Définition à la ligne 84 du fichier `mainWindow.py`.

**7.16.4.22** `src.mainWindow.mainWindow.redoStatusTip`

Définition à la ligne 211 du fichier `mainWindow.py`.

**7.16.4.23** `src.mainWindow.mainWindow.redoToolTip`

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

**7.16.4.24** `src.mainWindow.mainWindow.refreshDelay`

Définition à la ligne 246 du fichier `mainWindow.py`.

**7.16.4.25** `src.mainWindow.mainWindow.refreshEnabled`

Définition à la ligne 245 du fichier `mainWindow.py`.

**7.16.4.26** `src.mainWindow.mainWindow.schoolFile`

Définition à la ligne 243 du fichier `mainWindow.py`.

**7.16.4.27** `src.mainWindow.mainWindow.stopStatusTip`

Définition à la ligne 213 du fichier `mainWindow.py`.

**7.16.4.28** `src.mainWindow.mainWindow.stopToolTip`

Définition à la ligne 212 du fichier `mainWindow.py`.

**7.16.4.29** `src.mainWindow.mainWindow.t`

Définition à la ligne 88 du fichier `mainWindow.py`.

**7.16.4.30** `src.mainWindow.mainWindow.timer`

Définition à la ligne 91 du fichier `mainWindow.py`.

**7.16.4.31** `src.mainWindow.mainWindow.tm`

Définition à la ligne 640 du fichier `mainWindow.py`.

## 7.16.4.32 src.mainWindow.mainWindow.ui

Définition à la ligne 74 du fichier mainWindow.py.

## 7.16.4.33 src.mainWindow.mainWindow.visibleheader

Définition à la ligne 634 du fichier mainWindow.py.

## 7.16.4.34 src.mainWindow.mainWindow.workdir

Définition à la ligne 244 du fichier mainWindow.py.

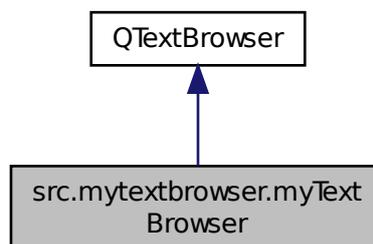
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

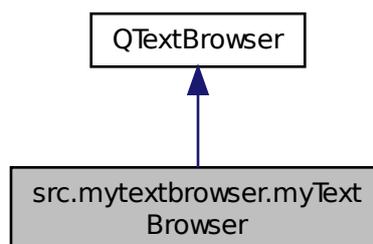
## 7.17 Référence de la classe src.mytextbrowser.myTextBrowser

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

Graphe d'héritage de src.mytextbrowser.myTextBrowser :



Graphe 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*

### 7.17.1 Description détaillée

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

Définition à la ligne 36 du fichier mytextbrowser.py.

### 7.17.2 Documentation des fonctions membres

#### 7.17.2.1 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 50 du fichier mytextbrowser.py.

#### 7.17.2.2 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 42 du fichier mytextbrowser.py.

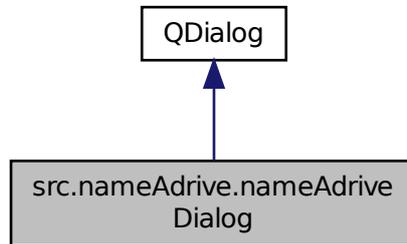
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mytextbrowser.py](#)

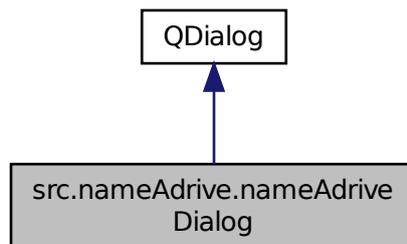
## 7.18 Référence de la classe src.nameAdrive.nameAdriveDialog

un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles

Graphe d'héritage de src.nameAdrive.nameAdriveDialog :



Graphe de collaboration de src.nameAdrive.nameAdriveDialog :



### Fonctions membres publiques

- def `__init__`  
*Le constructeur.*
- def `makeSelection`  
*Si l'ancien nom commence par un numéro, sélectionne le premier élément de la liste commençant par le même, sinon sélectionne le tout premier élément de la liste.*
- def `selectionChanged`  
*fonction de rappel quand la sélection change dans la liste ; recopie l'élément sélectionné comme nouveau nom de baladeur*
- def `ok`  
*fonction de rappel quand l'utilisateur valide le choix*
- def `esc`  
*fonction de rappel quand l'utilisateur cherche à échapper au choix*

### Attributs publics

- `oldName`
- `nameList`
- `tattoo`
- `ui`
- `numPattern`

### 7.18.1 Description détaillée

un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles

Définition à la ligne 38 du fichier nameAdrive.py.

### 7.18.2 Documentation des constructeurs et destructeur

7.18.2.1 `def src.nameAdrive.nameAdriveDialog.__init__( self, parent=None, oldName="", nameList=[], driveldent=None )`

Le constructeur.

#### Paramètres

<i>parent</i>	le widget parent
<i>oldName</i>	le nom précédent du baladeur
<i>nameList</i>	une liste de noms disponibles
<i>driveldent</i>	identité d'un baladeur sous forme d'un triplet (stickId, Uuid, Tattoo)

Définition à la ligne 48 du fichier nameAdrive.py.

### 7.18.3 Documentation des fonctions membres

7.18.3.1 `def src.nameAdrive.nameAdriveDialog.esc ( self )`

fonction de rappel quand l'utilisateur cherche à échapper au choix

Définition à la ligne 116 du fichier nameAdrive.py.

7.18.3.2 `def src.nameAdrive.nameAdriveDialog.makeSelection ( self )`

Si l'ancien nom commence par un numéro, sélectionne le premier élément de la liste commençant par le même, sinon sélectionne le tout premier élément de la liste.

Définition à la ligne 72 du fichier nameAdrive.py.

7.18.3.3 `def src.nameAdrive.nameAdriveDialog.ok ( self )`

fonction de rappel quand l'utilisateur valide le choix

Définition à la ligne 103 du fichier nameAdrive.py.

7.18.3.4 `def src.nameAdrive.nameAdriveDialog.selectionChanged ( self )`

fonction de rappel quand la sélection change dans la liste ; recopie l'élément sélectionné comme nouveau nom de baladeur

Définition à la ligne 92 du fichier nameAdrive.py.

### 7.18.4 Documentation des données membres

7.18.4.1 `src.nameAdrive.nameAdriveDialog.nameList`

Définition à la ligne 51 du fichier nameAdrive.py.

#### 7.18.4.2 src.nameAdrive.nameAdriveDialog.numPattern

Définition à la ligne 59 du fichier nameAdrive.py.

#### 7.18.4.3 src.nameAdrive.nameAdriveDialog.oldName

Définition à la ligne 50 du fichier nameAdrive.py.

#### 7.18.4.4 src.nameAdrive.nameAdriveDialog.tattoo

Définition à la ligne 53 du fichier nameAdrive.py.

#### 7.18.4.5 src.nameAdrive.nameAdriveDialog.ui

Définition à la ligne 54 du fichier nameAdrive.py.

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

— [src/nameAdrive.py](#)

## 7.19 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`

#### 7.19.1 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 39 du fichier notification.py.

#### 7.19.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

<i>app_name</i>	nom d'une application, valeur par défaut = ""
<i>replaces_id</i>	identifiant d'une notification à remplacer valeur par défaut=0
<i>app_icon</i>	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 pa défaut=""
<i>actions</i>	une liste de paires représeantant 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 55 du fichier notification.py.

### 7.19.3 Documentation des fonctions membres

```
7.19.3.1 def src.notification.Notification.notify ( self )
```

Définition à la ligne 72 du fichier notification.py.

### 7.19.4 Documentation des données membres

```
7.19.4.1 src.notification.Notification.actions
```

Définition à la ligne 61 du fichier notification.py.

```
7.19.4.2 src.notification.Notification.app_icon
```

Définition à la ligne 58 du fichier notification.py.

```
7.19.4.3 src.notification.Notification.app_name
```

Définition à la ligne 56 du fichier notification.py.

```
7.19.4.4 src.notification.Notification.body
```

Définition à la ligne 60 du fichier notification.py.

```
7.19.4.5 src.notification.Notification.expire_timeout
```

Définition à la ligne 63 du fichier notification.py.

```
7.19.4.6 src.notification.Notification.hints
```

Définition à la ligne 62 du fichier notification.py.

```
7.19.4.7 src.notification.Notification.interface
```

Définition à la ligne 68 du fichier notification.py.

## 7.19.4.8 src.notification.Notification.replaces\_id

Définition à la ligne 57 du fichier notification.py.

## 7.19.4.9 src.notification.Notification.summary

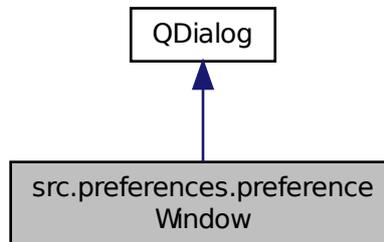
Définition à la ligne 59 du fichier notification.py.

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

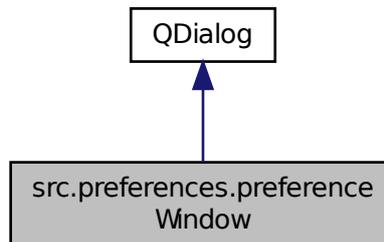
— [src/notification.py](#)

## 7.20 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 [enableDelay](#)  
*active ou désactive le glisseur pour modifier le délai de rafraichissement*

- def `updateRefreshLabel`  
*Met à jour l'affichage de la valeur du délai de rafraichissement.*
- def `values`
- def `setValues`  
*Met en place les préférences dans le dialogue.*

## Attributs publics

- `ui`

### 7.20.1 Description détaillée

Définition à la ligne 31 du fichier preferences.py.

### 7.20.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Définition à la ligne 36 du fichier preferences.py.

### 7.20.3 Documentation des fonctions membres

7.20.3.1 `def src.preferences.preferenceWindow.enableDelay ( self, state )`

active ou désactive le glisseur pour modifier le délai de rafraichissement

#### Paramètres

<code>state</code>	l'état coché ou décoché de la boîte qui contrôle le rafraichissement
--------------------	--

Définition à la ligne 49 du fichier preferences.py.

7.20.3.2 `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 83 du fichier preferences.py.

7.20.3.3 `def src.preferences.preferenceWindow.updateRefreshLabel ( self, val )`

Met à jour l'affichage de la valeur du délai de rafraichissement.

#### Paramètres

<code>val</code>	un nombre entier qui exprime le délai en secondes
------------------	---

Définition à la ligne 57 du fichier preferences.py.

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

**Renvoie**

un dictionnaire de préférences

Définition à la ligne 67 du fichier preferences.py.

**7.20.4 Documentation des données membres****7.20.4.1 src.preferences.preferenceWindow.ui**

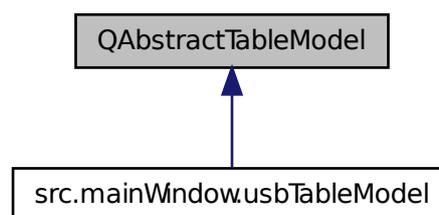
Définition à la ligne 39 du fichier preferences.py.

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

— [src/preferences.py](#)

**7.21 Référence de la classe QAbstractTableModel**

Graphe d'héritage de QAbstractTableModel :

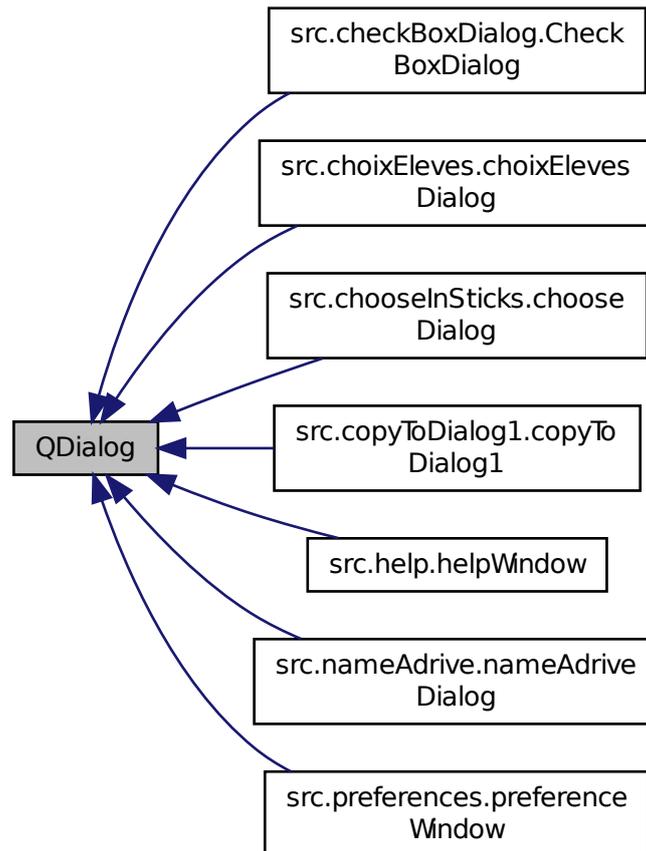


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

## 7.22 Référence de la classe QDialog

Graphe d'héritage de QDialog :

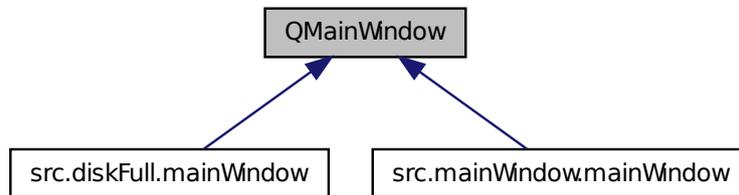


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/copyToDialog1.py](#)

## 7.23 Référence de la classe QMainWindow

Graphe d'héritage de QMainWindow :

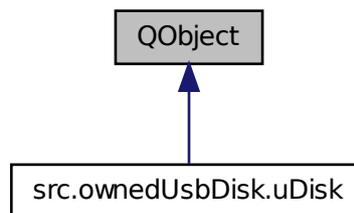


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

## 7.24 Référence de la classe QObject

Graphe d'héritage de QObject :

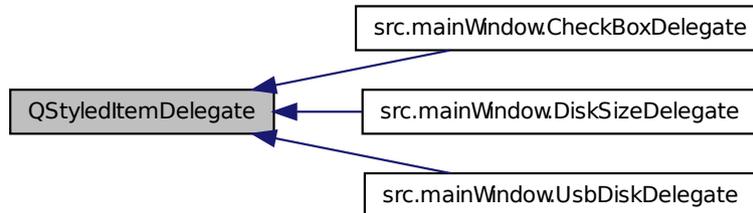


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/ownedUsbDisk.py](#)

## 7.25 Référence de la classe QStyledItemDelegate

Grappe d'héritage de QStyledItemDelegate :

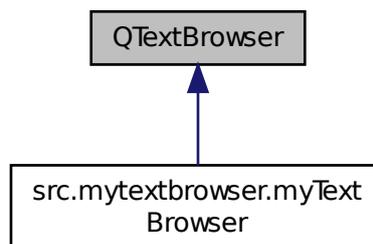


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

## 7.26 Référence de la classe QTextBrowser

Grappe d'héritage de QTextBrowser :

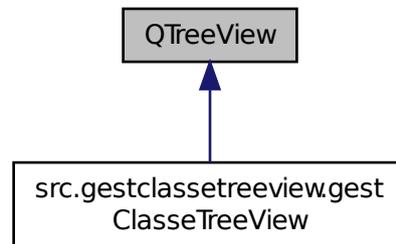


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mytextbrowser.py](#)

## 7.27 Référence de la classe QTreeView

Graphe d'héritage de QTreeView :



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

- [src/gestclassetreeview.py](#)

## 7.28 Référence de la classe src.sconet.Sconet

Une classe pour travailler avec des données [Sconet](#).

### Fonctions membres publiques

- def [\\_\\_init\\_\\_](#)  
*Le constructeur.*
- def [makeCompact](#)  
*removes useless text nodes containing only spaces.*
- def [collectNullTexts](#)
- def [collectClasses](#)
- def [collectOneClass](#)
- def [elementsWalk](#)  
*implemente un parcours des éléments d'un arbre, pour y appliquer une procédure*
- def [\\_\\_str\\_\\_](#)

### Attributs publics

- [donnees](#)
- [nullTexts](#)
- [classes](#)

### 7.28.1 Description détaillée

Une classe pour travailler avec des données [Sconet](#).

Définition à la ligne 33 du fichier sconet.py.

### 7.28.2 Documentation des constructeurs et destructeur

7.28.2.1 `def src.sconet.Sconet.__init__( self, file )`

Le constructeur.

Paramètres

<i>file</i>	le nom d'un fichier, ou un fichier ouvert en lecture
-------------	--

Définition à la ligne 40 du fichier sconet.py.

### 7.28.3 Documentation des fonctions membres

7.28.3.1 `def src.sconet.Sconet.__str__( self )`

Définition à la ligne 100 du fichier sconet.py.

7.28.3.2 `def src.sconet.Sconet.collectClasses ( self )`

Revoie

the list of classes containg students

Définition à la ligne 72 du fichier sconet.py.

7.28.3.3 `def src.sconet.Sconet.collectNullTexts ( self, el )`

Définition à la ligne 62 du fichier sconet.py.

7.28.3.4 `def src.sconet.Sconet.collectOneClass ( self, el )`

Revoie

the name of a class if it is a class with students

Définition à la ligne 81 du fichier sconet.py.

7.28.3.5 `def src.sconet.Sconet.elementsWalk ( self, el, proc )`

implemente un parcour des éléments d'un arbre, pour y appliquer une procédure

Paramètres

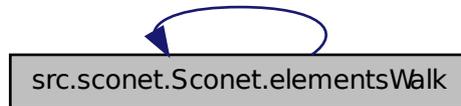
<i>el</i>	un élément
<i>proc</i>	la procédure à appliquer (paramètres : l'élément)

Définition à la ligne 95 du fichier sconet.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.28.3.6 `def src.sconet.Sconet.makeCompact ( self )`

removes useless thext nodes containing only spaces.

Définition à la ligne 55 du fichier `sconet.py`.

## 7.28.4 Documentation des données membres

7.28.4.1 `src.sconet.Sconet.classes`

Définition à la ligne 73 du fichier `sconet.py`.

7.28.4.2 `src.sconet.Sconet.donnees`

Définition à la ligne 48 du fichier `sconet.py`.

7.28.4.3 `src.sconet.Sconet.nullTexts`

Définition à la ligne 56 du fichier `sconet.py`.

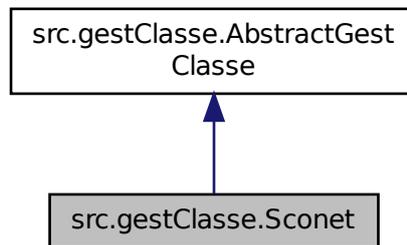
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/sconet.py](#)

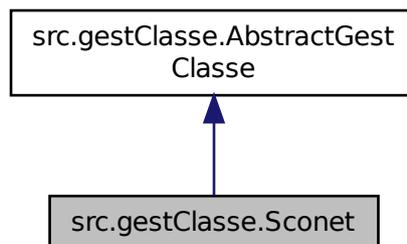
## 7.29 Référence de la classe `src.gestClasse.Sconet`

Une classe pour travailler avec des données [Sconet](#).

Graphe d'héritage de src.gestClasse.Sconet :



Graphe de collaboration de src.gestClasse.Sconet :



## Fonctions membres publiques

- def `__init__`  
*Le constructeur.*
- def `makeCompact`  
*removes useless text nodes containing only spaces.*
- def `collectNullTexts`
- def `collectClasses`
- def `elevesDeClasse`
- def `eleveParID`  
*appends the "eleve" element to the list self.currentResult if self.currentID is matched*
- def `unIDEleveDeClasse`  
*appends the ID of an "eleve" to self.currentResult if he belongs to the class self.currentClassName*
- def `collectOneClass`  
*adds one class name to the set self.classes*
- def `unique_name`  
*a unique name for an "eleve", based on a few fields and on the ID*
- def `showable_name`
- def `elementsWalk`  
*implemente un parcours des éléments d'un arbre, pour y appliquer une procédure*
- def `__str__`

## Attributs publics

- `donnees`
- `nullTexts`
- `classes`
- `currentResult`
- `currentClassName`
- `currentID`

### 7.29.1 Description détaillée

Une classe pour travailler avec des données [Sconet](#).

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

### 7.29.2 Documentation des constructeurs et destructeur

7.29.2.1 `def src.gestClasse.Sconet.__init__( self, f )`

Le constructeur.

#### Paramètres

<code>f</code>	le nom d'un fichier, ou un fichier ouvert en lecture
----------------	--

Réimplémentée à partir de [src.gestClasse.AbstractGestClasse](#).

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

### 7.29.3 Documentation des fonctions membres

7.29.3.1 `def src.gestClasse.Sconet.__str__( self )`

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

7.29.3.2 `def src.gestClasse.Sconet.collectClasses ( self )`

#### Renvoie

the list of classes containg students

Réimplémentée à partir de [src.gestClasse.AbstractGestClasse](#).

Définition à la ligne 122 du fichier `gestClasse.py`.

7.29.3.3 `def src.gestClasse.Sconet.collectNullTexts ( self, el )`

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

7.29.3.4 `def src.gestClasse.Sconet.collectOneClass ( self, el )`

adds one class name to the set `self.classes`

#### Paramètres

<code>el</code>	an element
-----------------	------------

Définition à la ligne 173 du fichier `gestClasse.py`.

### 7.29.3.5 def src.gestClasse.Sconet.elementsWalk ( self, el, proc )

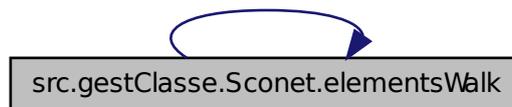
implemente un parcour des éléments d'un arbre, pour y appliquer une procédure

#### Paramètres

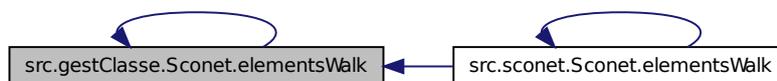
<i>el</i>	un élément
<i>proc</i>	la procédure à appliquer (paramètres : l'élément)

Définition à la ligne 213 du fichier gestClasse.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 7.29.3.6 def src.gestClasse.Sconet.eleveParID ( self, el )

appends the "eleve" element to the list `self.currentResult` if `self.currentID` is matched

Définition à la ligne 149 du fichier gestClasse.py.

### 7.29.3.7 def src.gestClasse.Sconet.elevsDeClasse ( self, className )

#### Paramètres

<i>className</i>	name of a school class
------------------	------------------------

#### Renvoie

list of "eleve" elements

Réimplémentée à partir de [src.gestClasse.AbstractGestClasse](#).

Définition à la ligne 132 du fichier gestClasse.py.

7.29.3.8 `def src.gestClasse.Sconet.makeCompact ( self )`

removes useless thext nodes containing only spaces.

Définition à la ligne 105 du fichier `gestClasse.py`.

7.29.3.9 `def src.gestClasse.Sconet.showable_name ( self, el, fields = [ "NOM", PRENOM ]`

## Paramètres

<i>el</i>	un objet élève
<i>fields</i>	les champs de donnée à exploiter

## Renvoie

une chaîne unicode, pour nommer l'élève

Définition à la ligne 199 du fichier `gestClasse.py`.

7.29.3.10 `def src.gestClasse.Sconet.unIDeEleveDeClasse ( self, el )`

appends the ID of an "eleve" to `self.currentResult` if he belongs to the class `self.currentClassName`

## Paramètres

<i>el</i>	an element
-----------	------------

Définition à la ligne 159 du fichier `gestClasse.py`.

7.29.3.11 `def src.gestClasse.Sconet.unique_name ( self, el, fields = [ "NOM", PRENOM ]`

a unique name for an "eleve", based on a few fields and on the ID

## Paramètres

<i>el</i>	en "eleve" element
<i>fields</i>	the fields used to build the result a printable unique id

Définition à la ligne 186 du fichier `gestClasse.py`.

## 7.29.4 Documentation des données membres

7.29.4.1 `src.gestClasse.Sconet.classes`

Définition à la ligne 123 du fichier `gestClasse.py`.

7.29.4.2 `src.gestClasse.Sconet.currentClassName`

Définition à la ligne 134 du fichier `gestClasse.py`.

7.29.4.3 `src.gestClasse.Sconet.currentID`

Définition à la ligne 139 du fichier `gestClasse.py`.

#### 7.29.4.4 `src.gestClasse.Sconet.currentResult`

Définition à la ligne 133 du fichier `gestClasse.py`.

#### 7.29.4.5 `src.gestClasse.Sconet.donnees`

Définition à la ligne 98 du fichier `gestClasse.py`.

#### 7.29.4.6 `src.gestClasse.Sconet.nullTexts`

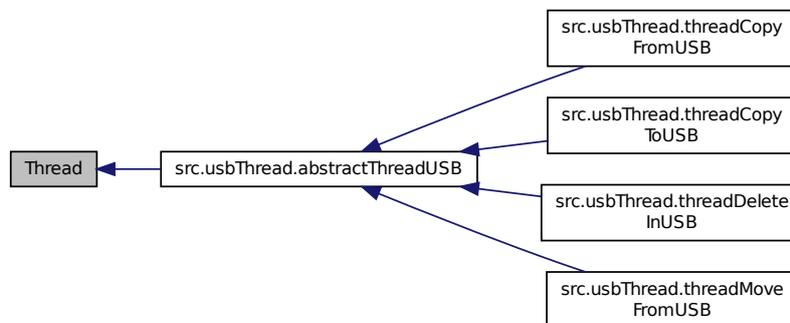
Définition à la ligne 106 du fichier `gestClasse.py`.

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

— [src/gestClasse.py](#)

## 7.30 Référence de la classe Thread

Graphe d'héritage de Thread :



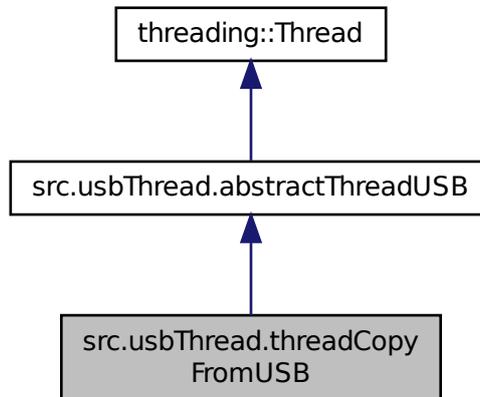
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

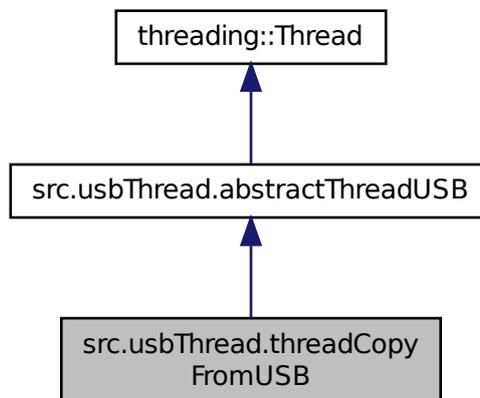
## 7.31 Référence de la classe `src.usbThread.threadCopyFromUSB`

Classe pour les threads copiant depuis les clés USB.

Grappe d'héritage de src.usbThread.threadCopyFromUSB :



Grappe 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`

### 7.31.1 Description détaillée

Classe pour les threads copiant depuis les clés USB.

Définition à la ligne 360 du fichier usbThread.py.

### 7.31.2 Documentation des constructeurs et destructeur

7.31.2.1 `def src.usbThread.threadCopyFromUSB.__init__( self, ud, fileList, subdir = ".", dest = "/tmp", rootPath = "/", logfile = "/dev/null", parent = None )`

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

#### 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 d'où faire la copie
<i>dest</i>	un répertoire de destination
<i>logfile</i>	un fichier de journalisation, /dev/null par défaut
<i>parent</i>	un widget qui recevra de signaux en début et en fin d'exécution

Définition à la ligne 375 du fichier usbThread.py.

### 7.31.3 Documentation des fonctions membres

7.31.3.1 `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.

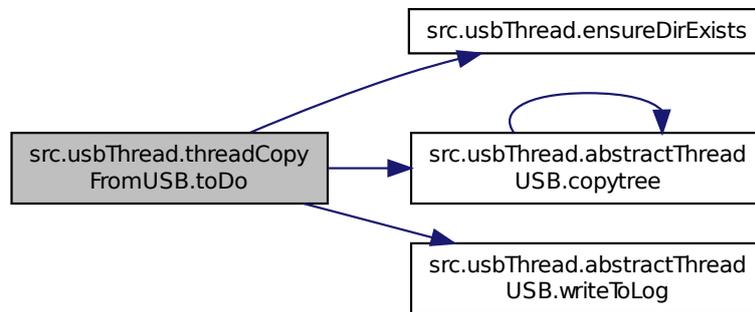
#### Paramètres

<i>ud</i>	l'instance uDisk correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à copier, qui peut contenir des jokers
<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 391 du fichier usbThread.py.

Voici le graphe d'appel pour cette fonction :



### 7.31.4 Documentation des données membres

#### 7.31.4.1 `src.usbThread.threadCopyFromUSB.rootPath`

Définition à la ligne 378 du fichier `usbThread.py`.

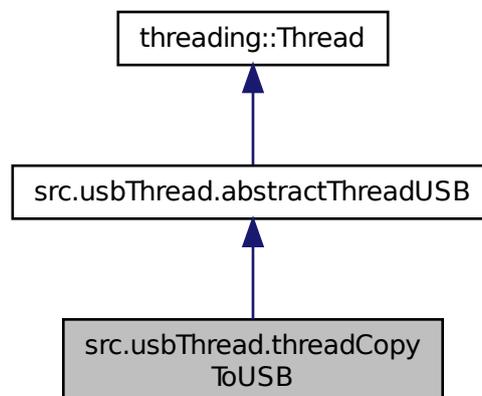
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

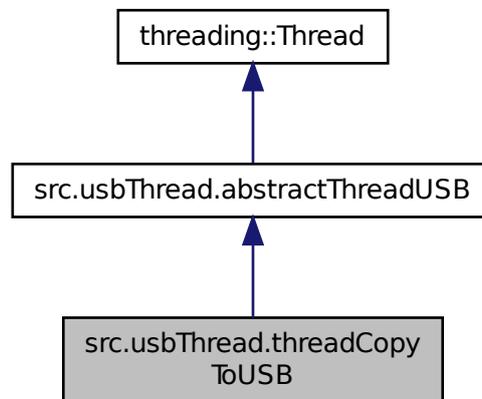
## 7.32 Référence de la classe `src.usbThread.threadCopyToUSB`

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.*

### Additional Inherited Members

#### 7.32.1 Description détaillée

Classe pour les threads copiant vers les clés USB.

Définition à la ligne 287 du fichier `usbThread.py`.

#### 7.32.2 Documentation des constructeurs et destructeur

7.32.2.1 `def src.usbThread.threadCopyToUSB.__init__( self, ud, fileList, subdir, logfile = "/dev/null", parent = None )`

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

##### 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 où faire la copie
<code>logfile</code>	un fichier de journalisation, <code>/dev/null</code> par défaut
<code>parent</code>	un widget qui recevra de signaux en début et en fin d'exécution

Définition à la ligne 300 du fichier `usbThread.py`.

### 7.32.3 Documentation des fonctions membres

#### 7.32.3.1 `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 307 du fichier `usbThread.py`.

#### 7.32.3.2 `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.

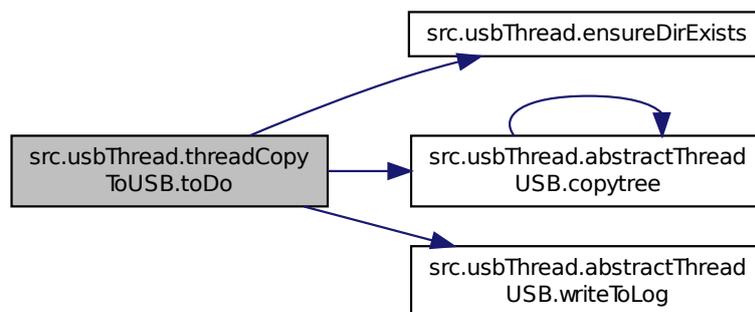
Paramètres

<i>ud</i>	l'instance <code>uDisk</code> 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 322 du fichier `usbThread.py`.

Voici le graphe d'appel pour cette fonction :



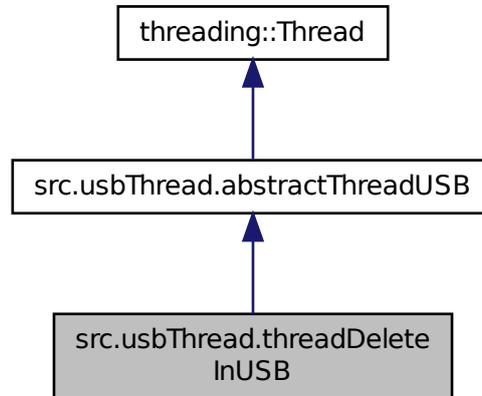
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

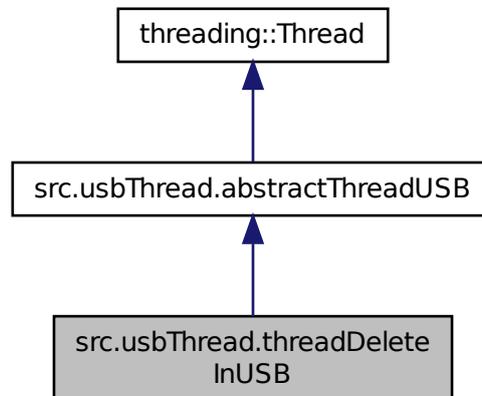
## 7.33 Référence de la classe `src.usbThread.threadDeleteInUSB`

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

Grappe d'héritage de `src.usbThread.threadDeleteInUSB` :



Grappe 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.*

### Additional Inherited Members

### 7.33.1 Description détaillée

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

Définition à la ligne 508 du fichier `usbThread.py`.

### 7.33.2 Documentation des constructeurs et destructeur

7.33.2.1 `def src.usbThread.threadDeleteInUSB.__init__( self, ud, fileList, subdir, logfile = "/dev/null", parent = None )`

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

#### Paramètres

<i>ud</i>	l'instance <code>uDisk</code> 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, <code>/dev/null</code> par défaut
<i>parent</i>	un widget qui recevra de signaux en début et en fin d'exécution

Définition à la ligne 521 du fichier `usbThread.py`.

### 7.33.3 Documentation des fonctions membres

7.33.3.1 `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 de la 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>	<code>uDisk</code> 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 538 du fichier `usbThread.py`.

Voici le graphe d'appel pour cette fonction :



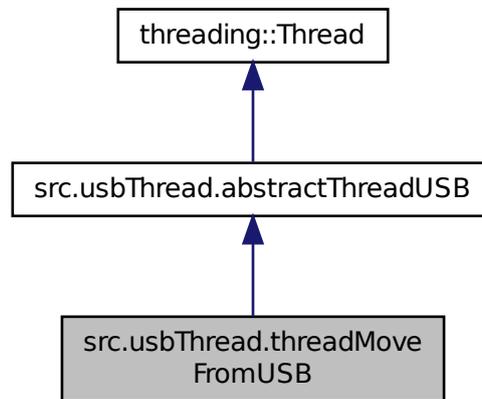
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

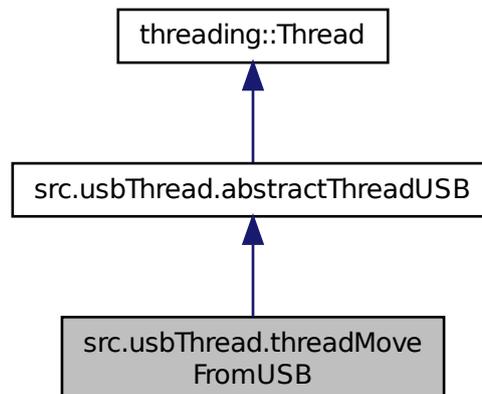
## 7.34 Référence de la classe src.usbThread.threadMoveFromUSB

Classe pour les threads déplaçant des fichiers depuis les clés USB.

Graphe d'héritage de src.usbThread.threadMoveFromUSB :



Graphe de collaboration de src.usbThread.threadMoveFromUSB :



### Fonctions membres publiques

- def `__init__`  
*Constructeur Crée un thread pour déplacer 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](#)

### 7.34.1 Description détaillée

Classe pour les threads déplaçant des fichiers depuis les clés USB.

Définition à la ligne 431 du fichier `usbThread.py`.

### 7.34.2 Documentation des constructeurs et destructeur

7.34.2.1 `def src.usbThread.threadMoveFromUSB.__init__( self, ud, fileList, subdir = ".", dest = "/tmp", rootPath = "/", logfile = "/dev/null", parent = None )`

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

#### Paramètres

<i>ud</i>	l'instance <code>uDisk</code> 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 d'où faire la copie
<i>dest</i>	un répertoire de destination
<i>logfile</i>	un fichier de journalisation, <code>/dev/null</code> par défaut
<i>parent</i>	un widget qui recevra de signaux en début et en fin d'exécution

Définition à la ligne 446 du fichier `usbThread.py`.

### 7.34.3 Documentation des fonctions membres

7.34.3.1 `def src.usbThread.threadMoveFromUSB.todo( self, ud, fileList, subdir, dest, logfile )`

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

Après chaque copie réussie la source est effacée. À chaque fichier ou répertoire copié, une ligne est journalisée dans le fichier de journal de l'application.

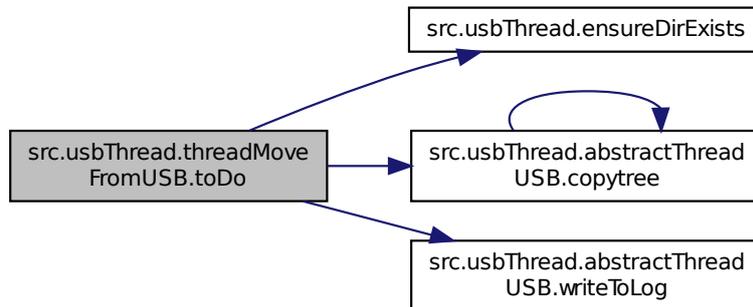
#### Paramètres

<i>ud</i>	l'instance <code>uDisk</code> 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 463 du fichier `usbThread.py`.

Voici le graphe d'appel pour cette fonction :



## 7.34.4 Documentation des données membres

### 7.34.4.1 src.usbThread.threadMoveFromUSB.rootPath

Définition à la ligne 449 du fichier usbThread.py.

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

- [src/usbThread.py](#)

## 7.35 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](#)

### 7.35.1 Description détaillée

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

Définition à la ligne 43 du fichier usbThread.py.

### 7.35.2 Documentation des constructeurs et destructeur

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

Le constructeur met en place un dictionnaire.  
Définition à la ligne 49 du fichier `usbThread.py`.

### 7.35.3 Documentation des fonctions membres

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

Définition à la ligne 52 du fichier `usbThread.py`.

#### 7.35.3.2 `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 82 du fichier `usbThread.py`.

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

##### Paramètres

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

Définition à la ligne 73 du fichier `usbThread.py`.

#### 7.35.3.4 `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 <i>ud</i>

Définition à la ligne 61 du fichier `usbThread.py`.

#### 7.35.3.5 `def src.usbThread.ThreadRegister.threadSet( self )`

renvoie l'ensemble des threads actifs

Définition à la ligne 91 du fichier `usbThread.py`.

### 7.35.4 Documentation des données membres

#### 7.35.4.1 `src.usbThread.ThreadRegister.dico`

Définition à la ligne 50 du fichier `usbThread.py`.

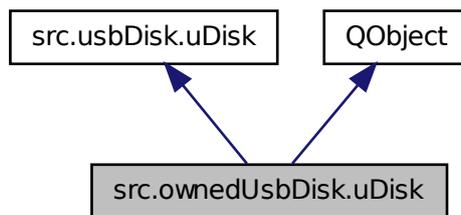
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

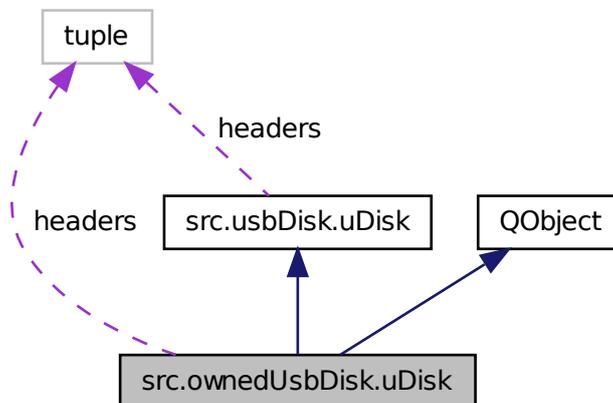
### 7.36 Référence de la classe `src.ownedUsbDisk.uDisk`

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.

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



Graphe 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)`

#### 7.36.1 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 85 du fichier `ownedUsbDisk.py`.

#### 7.36.2 Documentation des constructeurs et destructeur

7.36.2.1 `def src.ownedUsbDisk.uDisk.__init__( self, path, bus )`

##### Paramètres

<code>path</code>	un chemin dans le système dbus
<code>bus</code>	un objet <code>dbus.BusSystem</code>

Réimplémentée à partir de `src.usbDisk.uDisk`.

Définition à la ligne 91 du fichier `ownedUsbDisk.py`.

#### 7.36.3 Documentation des fonctions membres

7.36.3.1 `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

<code>n</code>	un nombre
----------------	-----------

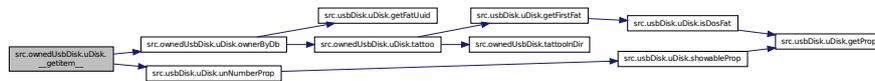
##### Renvoie

si `n==1`, renvoie `self` ; renvoie un élément si `n>0`, et le drapeau `self.selected` si `n==0`. Les noms des éléments sont dans la liste `self.itemNames`

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

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

Voici le graphe d'appel pour cette fonction :



### 7.36.3.2 def src.ownedUsbDisk.uDisk.ensureOwner ( self, noLoop )

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

#### Paramètres

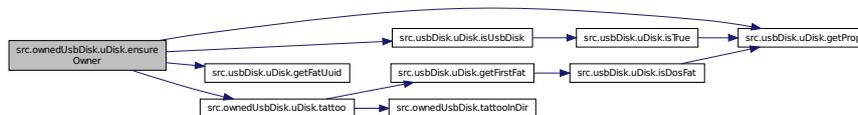
<i>noLoop</i>	si True : ne fait pas de dialogue interactif
---------------	--

#### Renvoie

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

Définition à la ligne 208 du fichier `ownedUsbDisk.py`.

Voici le graphe d'appel pour cette fonction :



### 7.36.3.3 def src.ownedUsbDisk.uDisk.headers ( locale = "C" )

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

#### Paramètres

<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 159 du fichier `ownedUsbDisk.py`.

### 7.36.3.4 def src.ownedUsbDisk.uDisk.ownerByDb ( self )

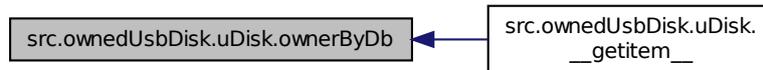
renvoie un nom de propriétaire dans tous les cas.

Définition à la ligne 169 du fichier `ownedUsbDisk.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 7.36.3.5 `def src.ownedUsbDisk.uDisk.readQuirks ( self )`

Lit un dictionnaire indexé par les 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 128 du fichier `ownedUsbDisk.py`.

#### 7.36.3.6 `def src.ownedUsbDisk.uDisk.tattoo ( self )`

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

**Renvoie**

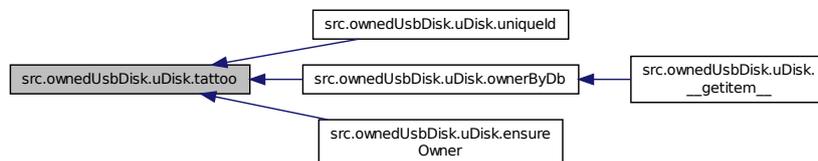
un tatouage, supposément unique.

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

**7.36.3.7 def src.ownedUsbDisk.uDisk.uniqueld ( self )****Renvoie**

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

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

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

Voici le graphe d'appel pour cette fonction :

**7.36.3.8 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 144 du fichier `ownedUsbDisk.py`.

**7.36.4 Documentation des données membres**

7.36.4.1 `tuple src.ownedUsbDisk.uDisk.headers = staticmethod(headers) [static]`

Définition à la ligne 199 du fichier `ownedUsbDisk.py`.

7.36.4.2 `src.ownedUsbDisk.uDisk.model`

Définition à la ligne 96 du fichier `ownedUsbDisk.py`.

7.36.4.3 `src.ownedUsbDisk.uDisk.owner`

Définition à la ligne 94 du fichier `ownedUsbDisk.py`.

7.36.4.4 `src.ownedUsbDisk.uDisk.vendor`

Définition à la ligne 95 du fichier `ownedUsbDisk.py`.

7.36.4.5 `src.ownedUsbDisk.uDisk.visibleDirs`

Définition à la ligne 97 du fichier `ownedUsbDisk.py`.

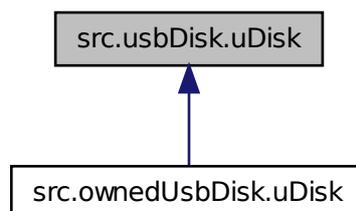
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/ownedUsbDisk.py](#)

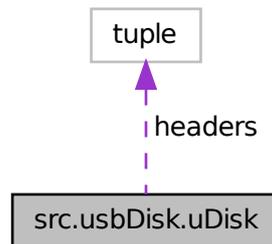
## 7.37 Référence de la classe `src.usbDisk.uDisk`

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 `isMounted`
- 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`
- `mp`
- `device`
- `device_prop`
- `selected`
- `rlock`
- `stickid`
- `uuid`
- `fatuuid`
- `firstFat`
- `devStuff`

### Attributs publics statiques

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

#### 7.37.1 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
- **rlock** un verrou récursif permettant de réserver l'usage du media pour un seul thread

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

#### 7.37.2 Documentation des constructeurs et destructeur

##### 7.37.2.1 `def src.usbDisk.uDisk.__init__( self, path, bus )`

Le constructeur.

##### Paramètres

<i>path</i>	un chemin dans le système dbus
<i>bus</i>	un objet <code>dbus.BusSystem</code>

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

Définition à la ligne 52 du fichier `usbDisk.py`.

#### 7.37.3 Documentation des fonctions membres

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

Renvoie un élément de listage de données internes au disque.

##### Paramètres

<i>n</i>	un nombre
----------	-----------

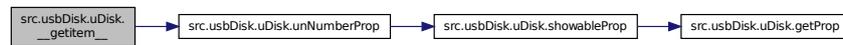
##### Renvoie

un élément si  $n > 0$ , et le drapeau `self.selected` si  $n == 0$ . Les noms des éléments sont dans la liste `itemNames` utilisée dans la fonction statique `headers`

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

Définition à la ligne 299 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :



### 7.37.3.2 def src.usbDisk.uDisk.\_\_str\_\_( self )

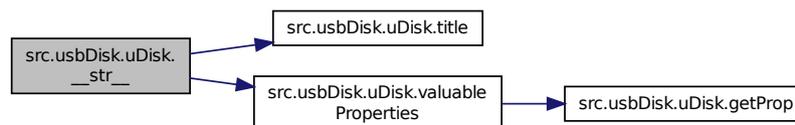
Fournit une représentation imprimable.

#### Renvoie

une représentation imprimable de l'instance

Définition à la ligne 151 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :



### 7.37.3.3 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 122 du fichier usbDisk.py.

### 7.37.3.4 def src.usbDisk.uDisk.ensureMounted ( self )

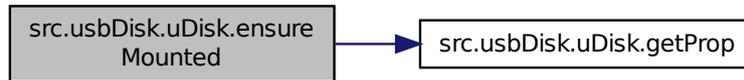
Permet de s'assurer qu'une partition ou un disque sera bien monté

**Renvoie**

le chemin du point de montage

Définition à la ligne 341 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :

**7.37.3.5 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 168 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

**7.37.3.6 def `src.usbDisk.uDisk.getFatUuid ( self )`**

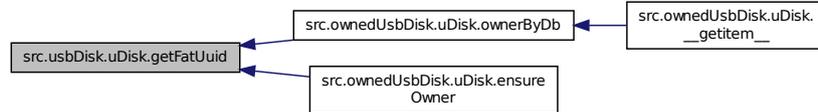
renvoie l'uuid de la première partition FAT après que celle-ci aura été identifiée (utile pour les disques partitionnés)

**Renvoie**

un uuid

Définition à la ligne 90 du fichier usbDisk.py.

Voici le graphe des appelants de cette fonction :

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

Renvoie la première partition VFAT.

**Renvoie**

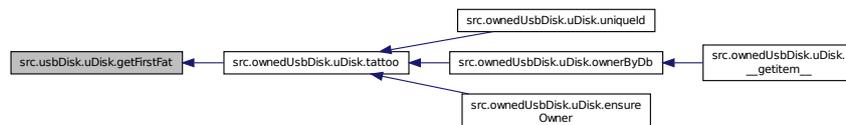
la première partition VFAT ou None s'il n'y en a pas

Définition à la ligne 332 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

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

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

**Paramètres**

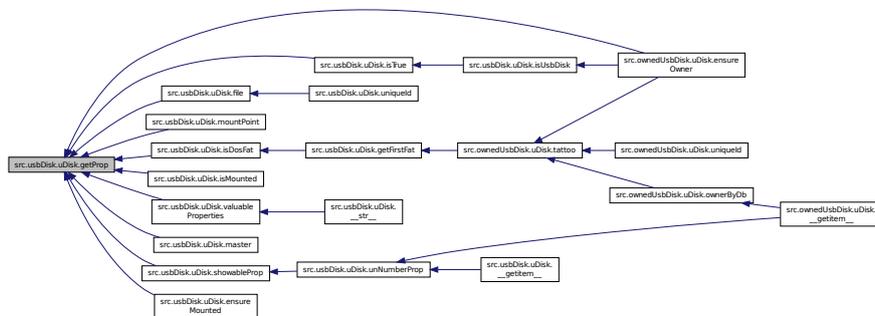
<i>name</i>	le nom d'une propriété
-------------	------------------------

**Renvoie**

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

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

Voici le graphe des appelants de cette fonction :

**7.37.3.9** `def src.usbDisk.uDisk.headers ( locale = "C" )`

Méthode statique, pour avoir des titres de colonne.

renvoie des titres pour les items obtenus par `getitem`.

**Paramètres**

<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 110 du fichier `usbDisk.py`.

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

Permet de reconnaître les partitions DOS-FAT.

**Renvoie**

True dans le cas d'une partition FAT16 ou FAT32

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



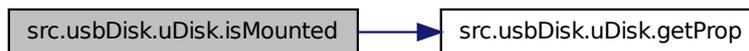
### 7.37.3.11 def src.usbDisk.uDisk.isMounted ( self )

#### Renvoie

True si le disque ou la partition est montée

Définition à la ligne 216 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :



### 7.37.3.12 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 132 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 7.37.3.13 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 143 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 7.37.3.14 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 272 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

**7.37.3.15 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 181 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

**7.37.3.16 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 315 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

**7.37.3.17 def src.usbDisk.uDisk.title ( self )**

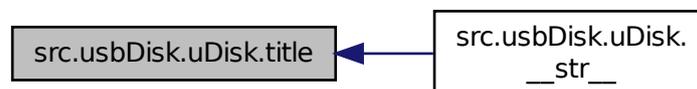
Permet d'obtenir un identifiant unique de disque.

**Renvoie**

le chemin dbus de l'instance

Définition à la ligne 159 du fichier usbDisk.py.

Voici le graphe des appelants de cette fonction :

**7.37.3.18 def src.usbDisk.uDisk.uniqueId ( self )**

renvoie un identifiant unique.

Dans cette classe, cette fonction est synonyme de [file\(\)](#)

**Renvoie**

un identifiant unique, garanti par le système de fichiers

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

Définition à la ligne 99 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :

**7.37.3.19 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**

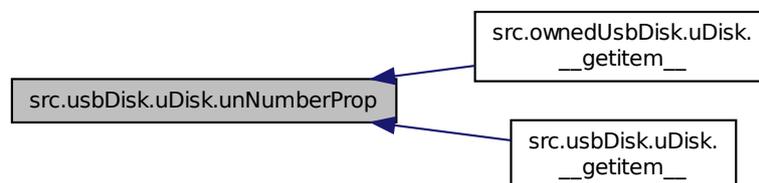
une propriété renvoyée par dbus, dans un format imprimable

Définition à la ligne 282 du fichier usbDisk.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

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

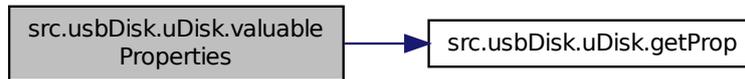
Facilite l'accès aux propriétés intéressantes d'une instance.

**Renvoie**

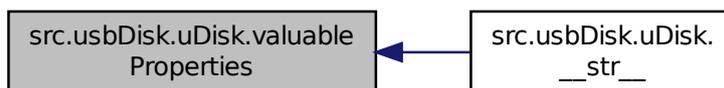
une chaîne indentée avec les propriétés intéressantes, une par ligne

Définition à la ligne 224 du fichier `usbDisk.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 7.37.4 Documentation des données membres

#### 7.37.4.1 `src.usbDisk.uDisk.device`

Définition à la ligne 55 du fichier `usbDisk.py`.

#### 7.37.4.2 `src.usbDisk.uDisk.device_prop`

Définition à la ligne 56 du fichier `usbDisk.py`.

#### 7.37.4.3 `src.usbDisk.uDisk.devStuff`

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

#### 7.37.4.4 `src.usbDisk.uDisk.fatuuid`

Définition à la ligne 61 du fichier `usbDisk.py`.

#### 7.37.4.5 `src.usbDisk.uDisk.firstFat`

Définition à la ligne 62 du fichier `usbDisk.py`.

7.37.4.6 `tuple src.usbDisk.uDisk.headers = staticmethod(headers)` [static]

Définition à la ligne 114 du fichier `usbDisk.py`.

7.37.4.7 `src.usbDisk.uDisk.mp`

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

7.37.4.8 `src.usbDisk.uDisk.path`

Définition à la ligne 53 du fichier `usbDisk.py`.

7.37.4.9 `src.usbDisk.uDisk.rlock`

Définition à la ligne 58 du fichier `usbDisk.py`.

7.37.4.10 `src.usbDisk.uDisk.selected`

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

7.37.4.11 `src.usbDisk.uDisk.stickid`

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

7.37.4.12 `src.usbDisk.uDisk.uuid`

Définition à la ligne 60 du fichier `usbDisk.py`.

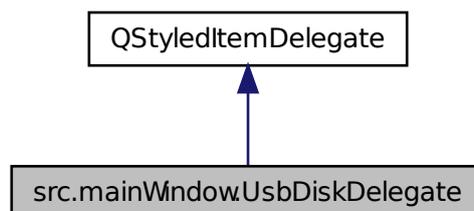
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbDisk.py](#)

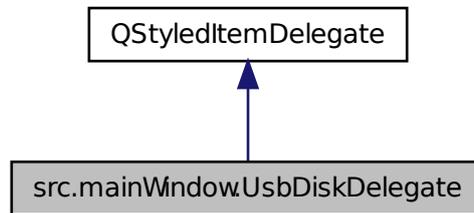
## 7.38 Référence de la classe `src.mainWindow.UsbDiskDelegate`

Classe pour identifier le baladeur dans le tableau.

Graphes d'héritage de `src.mainWindow.UsbDiskDelegate` :



Graphe de collaboration de src.mainWindow.UsbDiskDelegate :



### Fonctions membres publiques

- def `__init__`
- def `paint`

### Attributs publics

- `okPixmap`
- `busyPixmap`

#### 7.38.1 Description détaillée

Classe pour identifier le baladeur dans le tableau.

La routine de rendu à l'écran trace une petite icône et le nom du propriétaire à côté.

Définition à la ligne 905 du fichier mainWindow.py.

#### 7.38.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 906 du fichier mainWindow.py.

#### 7.38.3 Documentation des fonctions membres

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

Définition à la ligne 911 du fichier mainWindow.py.

#### 7.38.4 Documentation des données membres

7.38.4.1 `src.mainWindow.UsbDiskDelegate.busyPixmap`

Définition à la ligne 909 du fichier mainWindow.py.

#### 7.38.4.2 src.mainWindow.UsbDiskDelegate.okPixmap

Définition à la ligne 908 du fichier mainWindow.py.

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

— [src/mainWindow.py](#)

## 7.39 Référence de la classe src.essai.usbT

### Fonctions membres publiques

— def [\\_\\_init\\_\\_](#)  
— def [threadCopyToUSB](#)

#### 7.39.1 Description détaillée

Définition à la ligne 1 du fichier essai.py.

#### 7.39.2 Documentation des constructeurs et destructeur

7.39.2.1 def src.essai.usbT.\_\_init\_\_( self )

Définition à la ligne 2 du fichier essai.py.

#### 7.39.3 Documentation des fonctions membres

7.39.3.1 def src.essai.usbT.threadCopyToUSB ( self, p, l, subdir = ' ', logfile = ' ', parent = None )

Définition à la ligne 5 du fichier essai.py.

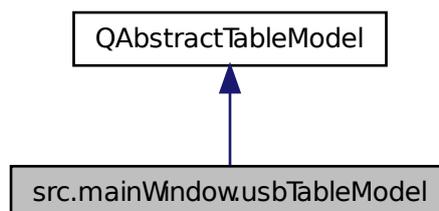
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/essai.py](#)

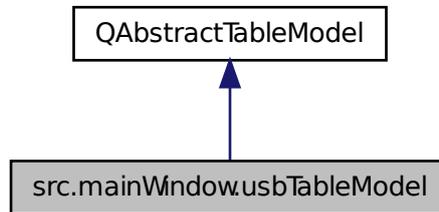
## 7.40 Référence de la classe src.mainWindow.usbTableModel

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 `pushCmd`  
*fonction de rappel déclenchée par les threads (au commencement)*
- def `popCmd`  
*fonction de rappel déclenchée par les threads (à la fin)*
- def `updateOwnerColumn`  
*force la mise à jour de la colonne des propriétaires*
- 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`
- `pere`

#### 7.40.1 Description détaillée

Un modèle de table pour des séries de clés USB.

Définition à la ligne 710 du fichier mainWindow.py.

#### 7.40.2 Documentation des constructeurs et destructeur

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

##### Paramètres

<i>parent</i>	un <a href="#">QObject</a>
<i>header</i>	les en-têtes de colonnes
<i>donnees</i>	les données

Définition à la ligne 718 du fichier mainWindow.py.

### 7.40.3 Documentation des fonctions membres

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

un QModelIndex

Définition à la ligne 790 du fichier mainWindow.py.

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

Définition à la ligne 808 du fichier mainWindow.py.

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

Définition à la ligne 837 du fichier mainWindow.py.

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

##### Paramètres

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

##### Renvoie

la partition pointée par index

Définition à la ligne 805 du fichier mainWindow.py.

#### 7.40.3.5 `def src.mainWindow.usbTableModel.popCmd ( self, owner, cmd )`

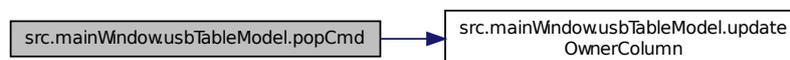
fonction de rappel déclenchée par les threads (à la fin)

##### Paramètres

<i>owner</i>	le propriétaire du baladeur associé au thread
<i>cmd</i>	la commande shell effectuée sur ce baladeur

Définition à la ligne 749 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



#### 7.40.3.6 `def src.mainWindow.usbTableModel.pushCmd ( self, owner, cmd )`

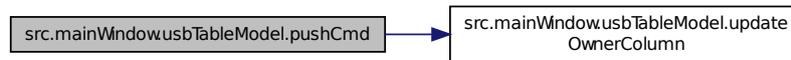
fonction de rappel déclenchée par les threads (au commencement)

##### Paramètres

<i>owner</i>	le propriétaire du baladeur associé au thread
<i>cmd</i>	la commande shell effectuée sur ce baladeur

Définition à la ligne 732 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



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

un `QModelIndex`

Définition à la ligne 783 du fichier `mainWindow.py`.

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

Définition à la ligne 793 du fichier `mainWindow.py`.

7.40.3.9 `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, <code>Qt.DescendingOrder</code> par défaut

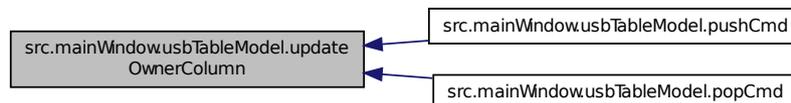
Définition à la ligne 849 du fichier `mainWindow.py`.

7.40.3.10 `def src.mainWindow.usbTableModel.updateOwnerColumn ( self )`

force la mise à jour de la colonne des propriétaires

Définition à la ligne 774 du fichier `mainWindow.py`.

Voici le graphe des appelants de cette fonction :



## 7.40.4 Documentation des données membres

7.40.4.1 `src.mainWindow.usbTableModel.donnees`

Définition à la ligne 721 du fichier `mainWindow.py`.

#### 7.40.4.2 `src.mainWindow.usbTableModel.header`

Définition à la ligne 720 du fichier `mainWindow.py`.

#### 7.40.4.3 `src.mainWindow.usbTableModel.pere`

Définition à la ligne 722 du fichier `mainWindow.py`.

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

— [src/mainWindow.py](#)

# Chapitre 8

## Documentation des fichiers

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

#### Paquetages

— namespace `src`

#### Variables

— `src.python3safe` = True

### 8.2 Référence du fichier `src/checkBoxDialog.py`

#### Classes

— class `src.checkBoxDialog.CheckBoxDialog`  
*Un dialogue pour gérer les cases à cocher de l'application.*

#### Paquetages

— namespace `src.checkBoxDialog`

#### Variables

— string `src.checkBoxDialog.licenceEn`  
— `src.checkBoxDialog.python3safe` = True

### 8.3 Référence du fichier `src/choixEleves.py`

#### Classes

— class `src.choixEleves.choixElevesDialog`  
*implémente un dialogue permettant de choisir des élèves les propriétés importantes sont `self.ok`, vrai si on doit prendre en compte la liste sélectionnée, et le contenu de la liste des sélectionnés, dont on peut récupérer les élèves un par un à l'aide de `self.pop()`*

## Paquetages

- namespace [src.choixElevés](#)

## Variables

- dictionary [src.choixElevés.licence](#) = {}
- [src.choixElevés.python3safe](#) = True
- tuple [src.choixElevés.app](#) = QApplication(sys.argv)
- tuple [src.choixElevés.d](#) = choixElevésDialog(gestionnaire=gestClasse.Sconet)
- tuple [src.choixElevés.i](#) = d.pop()

## 8.4 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](#)
- [src.chooseInSticks.python3safe](#) = True

## 8.5 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](#)
- [src.copyToDialog1.python3safe](#) = True
- tuple [src.copyToDialog1.app](#) = QApplication(sys.argv)
- tuple [src.copyToDialog1.windows](#) = copyToDialog1()

## 8.6 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.hasStudent](#)  
*vérifie qu'un étudiant est déjà connu*
- 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.python3safe](#) = True
- [src.db.database](#) = None
- [src.db.cursor](#) = None

## 8.7 Référence du fichier src/debug.py

### Paquetages

- namespace [src.debug](#)

### Fonctions

- def [src.debug.button](#)  
*ajoute un bouton de débogage dans une fenêtre*
- def [src.debug.listePartitionsCochees](#)  
*renseigne sur la liste des partions cochées de la fenêtre principale*

### Variables

- dictionary [src.debug.licence](#) = {}  
*Ce module facilite le debogage.*
- string [src.debug.licenceEn](#)
- string [src.debug.licenceFr](#)

## 8.8 Référence du fichier src/deviceListener.py

### Classes

- class [src.deviceListener.DeviceListener](#)

### Paquetages

- namespace [src.deviceListener](#)

## Variables

- dictionary `src.deviceListener.licence = {}`
- `src.deviceListener.python3safe = True`

## 8.9 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 = {}`
- `src.diskFull.python3safe = True`

## 8.10 Référence du fichier `src/essai.py`

### Classes

- class `src.essai.usbT`
- class `src.essai.machin`

### Paquetages

- namespace `src.essai`

## 8.11 Référence du fichier `src/gestClasse.py`

### Classes

- class `src.gestClasse.AbstractGestClasse`
- class `src.gestClasse.Sconet`  
*Une classe pour travailler avec des données `Sconet`.*

### Paquetages

- namespace `src.gestClasse`

### Variables

- dictionary `src.gestClasse.licence = {}`  
*Ce module permet de gérer des classes d'élèves.*
- `src.gestClasse.python3safe = True`

## 8.12 Référence du fichier src/gestclassetreeview.py

### Classes

- class [src.gestclassetreeview.gestClasseTreeView](#)

### Paquetages

- namespace [src.gestclassetreeview](#)

### Variables

- dictionary [src.gestclassetreeview.licence](#) = {}
- [src.gestclassetreeview.python3safe](#) = True

## 8.13 Référence du fichier src/globaldef.py

### Paquetages

- namespace [src.globaldef](#)

### Fonctions

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

### Variables

- string [src.globaldef.licenceEn](#)  
*globaldef.py is part of the package scolasync.*
- [src.globaldef.python3safe](#) = True
- string [src.globaldef.userShareDir](#) = "~/scolasync"
- string [src.globaldef.logFileName](#) = "~/scolasync/scolasync.log"
- string [src.globaldef.markFileName](#) = "~/scolasync/marques.py"

## 8.14 Référence du fichier src/help.py

### Classes

- class [src.help.helpWindow](#)

### Paquetages

- namespace [src.help](#)

### Variables

- dictionary [src.help.licence](#) = {}
- [src.help.python3safe](#) = True

## 8.15 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](#)  
*Classe pour identifier le baladeur dans le tableau.*
- class [src.mainWindow.DiskSizeDelegate](#)  
*Classe pour figurer la taille de la mémoire du baladeur.*

### Paquetages

- namespace [src.mainWindow](#)

### Fonctions

- def [src.mainWindow.registerCmd](#)  
*enregistre la commande cmd pour la partition donnée*
- def [src.mainWindow.CheckBoxRect](#)

### Variables

- dictionary [src.mainWindow.licence](#) = {}
- dictionary [src.mainWindow.activeThreads](#) = {}
- dictionary [src.mainWindow.pastCommands](#) = {}
- [src.mainWindow.lastCommand](#) = None

## 8.16 Référence du fichier src/marques.py

### Paquetages

- namespace [src.marques](#)

## 8.17 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](#) = {}
- [src.mytextbrowser.python3safe](#) = True

## 8.18 Référence du fichier src/nameAdrive.py

### Classes

- class [src.nameAdrive.nameAdriveDialog](#)  
*un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles*

### Paquetages

- namespace [src.nameAdrive](#)

### Variables

- dictionary [src.nameAdrive.licence](#) = {}
- [src.nameAdrive.python3safe](#) = True

## 8.19 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](#) = {}
- [src.notification.python3safe](#) = True
- tuple [src.notification.notif](#)

## 8.20 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.tattooInDir](#)  
*Renvoie le tatouage pour un point de montage donné, quitte à le créer si nécessaire.*
- def [src.ownedUsbDisk.editRecord](#)  
*édition de la base de données.*

## Variables

- dictionary `src.ownedUsbDisk.licence` = {}
- `src.ownedUsbDisk.python3safe` = True
- tuple `src.ownedUsbDisk.machin` = Available()

## 8.21 Référence du fichier src/preferences.py

### Classes

- class `src.preferences.preferenceWindow`

### Paquetages

- namespace `src.preferences`

### Variables

- dictionary `src.preferences.licence` = {}
- `src.preferences.python3safe` = True

## 8.22 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 baladeurs, de dictaphones ou de clés USB.*

### Fonctions

- def `src.scolasync.run`  
*Le lancement de l'application.*

### Variables

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

## 8.23 Référence du fichier src/sconet.py

### Classes

- class `src.sconet.Sconet`  
*Une classe pour travailler avec des données `Sconet`.*

### Paquetages

- namespace `src.sconet`

**Variables**

- dictionary `src.sconet.licence` = {}
- `src.sconet.python3safe` = True
- tuple `src.sconet.s` = `Sconet("../exemples/SCONET_test.xml")`

**8.24 Référence du fichier src/test3.py****Paquetages**

- namespace `src.test3`

**Variables**

- `src.test3.python3safe` = True
- tuple `src.test3.files` = `os.listdir(".")`
- tuple `src.test3.pattern` = `re.compile(".*\.py$")`
- list `src.test3.safe` = []
- list `src.test3.notsafe` = []
- tuple `src.test3.moduleName` = `f.replace(".py", "")`
- tuple `src.test3.module` = `__import__(moduleName)`

**8.25 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`
- string `src.usbDisk.dependences` = "python3-dbus python3-dbus.mainloop.qt"
- string `src.usbDisk.python3safe` = "True"
- tuple `src.usbDisk.machin` = `Available()`

**8.26 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, qui sert de creuset pour les classe servant aux copies et aux effacements.*
- 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.threadMoveFromUSB`  
*Classe pour les threads déplaçant des fichiers depuis les clés USB.*
- class `src.usbThread.threadDeleteInUSB`

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

## Paquetages

- namespace [src.usbThread](#)

## Fonctions

- def [src.usbThread.ensureDirExists](#)  
*force l'existence d'un répertoire, récursivement si nécessaire*
- def [src.usbThread.test\\_copytree](#)  
*Teste la fonction copytree.*
- def [src.usbThread.test\\_copy2](#)  
*Teste la copie d'un fichier vers une destination telle qu'elle est pratiquée dans la méthode copytree de [abstract-ThreadUSB](#).*

## Variables

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

## 8.27 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](#) = {}