

ScolaSync

1.0

Généré par Doxygen 1.7.4

Thu Jan 19 2012 00:18:46



# Table des matières

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

8.2	Paquetage src	15
8.3	Paquetage src.chooseInSticks	16
8.3.1	Documentation des variables	16
8.3.1.1	licenceEn	16
8.4	Paquetage src.copyToDialog1	16
8.4.1	Documentation des variables	17
8.4.1.1	app	17
8.4.1.2	licenceEn	17
8.4.1.3	windows	17
8.5	Paquetage src.db	17
8.5.1	Documentation des fonctions	18
8.5.1.1	checkVersion	18
8.5.1.2	hasStudent	18
8.5.1.3	knowsId	18
8.5.1.4	openDb	19
8.5.1.5	readPrefs	19
8.5.1.6	readStudent	19
8.5.1.7	setWd	19
8.5.1.8	tattooList	19
8.5.1.9	writePrefs	20
8.5.1.10	writeStudent	20
8.5.2	Documentation des variables	20
8.5.2.1	cursor	20
8.5.2.2	database	20
8.5.2.3	licence	20
8.6	Paquetage src.deviceListener	20
8.6.1	Documentation des variables	21
8.6.1.1	licence	21
8.7	Paquetage src.diskFull	21
8.7.1	Documentation des fonctions	21
8.7.1.1	sceneWithUsage	21
8.7.2	Documentation des variables	21
8.7.2.1	licence	21
8.8	Paquetage src.globaldef	21

8.8.1	Documentation des fonctions . . . . .	22
8.8.1.1	firstdir . . . . .	22
8.8.2	Documentation des variables . . . . .	22
8.8.2.1	licenceEn . . . . .	22
8.8.2.2	logFileName . . . . .	23
8.8.2.3	markFileName . . . . .	23
8.8.2.4	userShareDir . . . . .	23
8.9	Paquetage src.help . . . . .	23
8.9.1	Documentation des variables . . . . .	23
8.9.1.1	licence . . . . .	23
8.10	Paquetage src.mainWindow . . . . .	23
8.10.1	Documentation des fonctions . . . . .	24
8.10.1.1	CheckBoxRect . . . . .	24
8.10.1.2	registerCmd . . . . .	24
8.10.2	Documentation des variables . . . . .	24
8.10.2.1	activeThreads . . . . .	24
8.10.2.2	lastCommand . . . . .	24
8.10.2.3	licence . . . . .	24
8.10.2.4	pastCommands . . . . .	24
8.11	Paquetage src.marques . . . . .	25
8.12	Paquetage src.mytextbrowser . . . . .	25
8.12.1	Documentation des variables . . . . .	25
8.12.1.1	licence . . . . .	25
8.13	Paquetage src.notification . . . . .	25
8.13.1	Documentation des variables . . . . .	25
8.13.1.1	licence . . . . .	25
8.13.1.2	notif . . . . .	25
8.14	Paquetage src.ownedUsbDisk . . . . .	26
8.14.1	Documentation des fonctions . . . . .	26
8.14.1.1	editRecord . . . . .	26
8.14.2	Documentation des variables . . . . .	26
8.14.2.1	licence . . . . .	26
8.15	Paquetage src.preferences . . . . .	26
8.15.1	Documentation des variables . . . . .	27

8.15.1.1	licence	27
8.16	Paquetage src.scolasync	27
8.16.1	Documentation des fonctions	27
8.16.1.1	run	27
8.16.1.2	usage	27
8.16.2	Documentation des variables	27
8.16.2.1	licence	28
8.16.2.2	licenceEn	28
8.16.2.3	licenceFr	28
8.17	Paquetage src.usbDisk	29
8.17.1	Documentation des variables	29
8.17.1.1	licence	29
8.17.1.2	licence_en	29
8.17.1.3	machin	29
8.18	Paquetage src.usbThread	30
8.18.1	Documentation des variables	30
8.18.1.1	_threadNumber	30
8.18.1.2	licenceEn	30
8.19	Paquetage src.version	31
8.19.1	Documentation des fonctions	31
8.19.1.1	major	31
8.19.1.2	minor	31
8.19.1.3	version	31
8.19.2	Documentation des variables	31
8.19.2.1	licence	31
<b>9</b>	<b>Documentation des classes</b>	<b>33</b>
9.1	Référence de la classe src.usbThread.abstractThreadUSB	33
9.1.1	Description détaillée	34
9.1.2	Documentation des constructeurs et destructeur	34
9.1.2.1	__init__	35
9.1.3	Documentation des fonctions membres	35
9.1.3.1	__str__	35
9.1.3.2	protectPath	35

9.1.3.3	threadType	35
9.1.3.4	todo	36
9.1.4	Documentation des données membres	36
9.1.4.1	cmd	36
9.1.4.2	dest	36
9.1.4.3	fileList	36
9.1.4.4	logfile	36
9.1.4.5	parent	36
9.1.4.6	subdir	37
9.1.4.7	ud	37
9.2	Référence de la classe src.ownedUsbDisk.Available	37
9.2.1	Description détaillée	38
9.2.2	Documentation des constructeurs et destructeur	38
9.2.2.1	__init__	38
9.2.3	Documentation des fonctions membres	39
9.2.3.1	finishInit	39
9.2.4	Documentation des données membres	39
9.2.4.1	noLoop	39
9.3	Référence de la classe src.usbDisk.Available	39
9.3.1	Description détaillée	41
9.3.2	Documentation des constructeurs et destructeur	41
9.3.2.1	__init__	41
9.3.3	Documentation des fonctions membres	41
9.3.3.1	__getitem__	41
9.3.3.2	__len__	42
9.3.3.3	__str__	42
9.3.3.4	__trunc__	42
9.3.3.5	compare	42
9.3.3.6	contains	42
9.3.3.7	finishInit	43
9.3.3.8	getFirstFats	43
9.3.3.9	hasDev	43
9.3.3.10	mountFirstFats	43
9.3.3.11	summary	44

9.3.4	Documentation des données membres . . . . .	44
9.3.4.1	access . . . . .	44
9.3.4.2	bus . . . . .	44
9.3.4.3	checkable . . . . .	44
9.3.4.4	disks . . . . .	44
9.3.4.5	enumDev . . . . .	44
9.3.4.6	fatPaths . . . . .	44
9.3.4.7	firstFats . . . . .	44
9.4	Référence de la classe <code>src.mainWindow.CheckBoxDelegate</code> . . . . .	45
9.4.1	Description détaillée . . . . .	45
9.4.2	Documentation des constructeurs et destructeur . . . . .	46
9.4.2.1	__init__ . . . . .	46
9.4.3	Documentation des fonctions membres . . . . .	46
9.4.3.1	editorEvent . . . . .	46
9.4.3.2	paint . . . . .	46
9.5	Référence de la classe <code>src.chooseInSticks.chooseDialog</code> . . . . .	46
9.5.1	Description détaillée . . . . .	48
9.5.2	Documentation des constructeurs et destructeur . . . . .	48
9.5.2.1	__init__ . . . . .	48
9.5.3	Documentation des fonctions membres . . . . .	48
9.5.3.1	activate . . . . .	48
9.5.3.2	append . . . . .	48
9.5.3.3	baseDir . . . . .	49
9.5.3.4	changeWd . . . . .	49
9.5.3.5	checkWorkDirs . . . . .	49
9.5.3.6	choose . . . . .	49
9.5.3.7	choose_dir . . . . .	49
9.5.3.8	listStorages . . . . .	49
9.5.3.9	minus . . . . .	50
9.5.3.10	pathList . . . . .	50
9.5.3.11	plus . . . . .	50
9.5.3.12	selectedDiskMountPoint . . . . .	50
9.5.3.13	selectedDiskOwner . . . . .	50
9.5.4	Documentation des données membres . . . . .	50

9.5.4.1	mainWindow	50
9.5.4.2	ownedUsbDictionary	51
9.6	Référence de la classe src.copyToDialog1.copyToDialog1	51
9.6.1	Description détaillée	52
9.6.2	Documentation des fonctions membres	52
9.6.2.1	cancel	52
9.6.2.2	cd	53
9.6.2.3	changeWd	53
9.6.2.4	cont	53
9.6.2.5	displaySize	53
9.6.2.6	remove	53
9.6.2.7	select	53
9.6.2.8	selectedList	53
9.6.2.9	setFromListeDir	54
9.6.2.10	setupFromListe	54
9.6.2.11	setupToListe	54
9.6.3	Documentation des données membres	54
9.6.3.1	mainWindow	54
9.6.3.2	ok	54
9.7	Référence de la classe src.deviceListener.DeviceListener	54
9.7.1	Description détaillée	55
9.7.2	Documentation des constructeurs et destructeur	55
9.7.2.1	__init__	55
9.7.3	Documentation des fonctions membres	55
9.7.3.1	cbAdd	55
9.7.3.2	cbChange	55
9.7.3.3	cbDel	56
9.7.3.4	isVfatUsb	56
9.7.3.5	pollDevices	56
9.7.3.6	vfatUsbPath	56
9.7.4	Documentation des données membres	56
9.7.4.1	bus	57
9.7.4.2	connectedVolumes	57
9.7.4.3	interface	57

9.7.4.4	manager	57
9.7.4.5	widget	57
9.8	Référence de la classe src.mainWindow.DiskSizeDelegate	57
9.8.1	Description détaillée	58
9.8.2	Documentation des constructeurs et destructeur	58
9.8.2.1	__init__	58
9.8.3	Documentation des fonctions membres	58
9.8.3.1	paint	58
9.8.3.2	val2txt	59
9.9	Référence de la classe src.help.helpWindow	59
9.9.1	Description détaillée	60
9.9.2	Documentation des constructeurs et destructeur	60
9.9.2.1	__init__	60
9.9.3	Documentation des fonctions membres	60
9.9.3.1	loadBrowsers	60
9.9.4	Documentation des données membres	60
9.9.4.1	ui	60
9.10	Référence de la classe src.diskFull.mainWindow	61
9.10.1	Description détaillée	62
9.10.2	Documentation des constructeurs et destructeur	62
9.10.2.1	__init__	62
9.10.3	Documentation des données membres	62
9.10.3.1	total	62
9.10.3.2	ui	62
9.10.3.3	used	62
9.10.3.4	v	62
9.11	Référence de la classe src.mainWindow.mainWindow	63
9.11.1	Description détaillée	65
9.11.2	Documentation des constructeurs et destructeur	65
9.11.2.1	__init__	65
9.11.3	Documentation des fonctions membres	65
9.11.3.1	applyPreferences	65
9.11.3.2	changeWd	65
9.11.3.3	checkDisks	66

9.11.3.4	connectTableModel	66
9.11.3.5	copyFrom	66
9.11.3.6	copyTo	66
9.11.3.7	delFiles	66
9.11.3.8	deviceAdded	66
9.11.3.9	deviceRemoved	67
9.11.3.10	diskFromOwner	67
9.11.3.11	diskSizeData	67
9.11.3.12	editOwner	67
9.11.3.13	flashLCD	68
9.11.3.14	help	68
9.11.3.15	initRedoStuff	68
9.11.3.16	normalLCD	68
9.11.3.17	preference	68
9.11.3.18	redoCmd	68
9.11.3.19	sameDiskData	68
9.11.3.20	setTimer	68
9.11.3.21	showEvent	69
9.11.3.22	tableClicked	69
9.11.3.23	umount	69
9.11.3.24	updateButtons	69
9.11.4	Documentation des données membres	69
9.11.4.1	checkable	69
9.11.4.2	checkDisksLock	69
9.11.4.3	flashTimer	70
9.11.4.4	header	70
9.11.4.5	iconRedo	70
9.11.4.6	iconStop	70
9.11.4.7	listener	70
9.11.4.8	locale	70
9.11.4.9	manFileLocation	70
9.11.4.10	mv	70
9.11.4.11	oldThreads	70
9.11.4.12	operations	70

9.11.4.13	opts	71
9.11.4.14	proxy	71
9.11.4.15	redoStatusTip	71
9.11.4.16	redoToolTip	71
9.11.4.17	refreshDelay	71
9.11.4.18	refreshEnabled	71
9.11.4.19	stopStatusTip	71
9.11.4.20	stopToolTip	71
9.11.4.21	t	71
9.11.4.22	timer	71
9.11.4.23	tm	72
9.11.4.24	ui	72
9.11.4.25	visibleheader	72
9.11.4.26	workdir	72
9.12	Référence de la classe <code>src.mytextbrowser.myTextBrowser</code>	72
9.12.1	Description détaillée	73
9.12.2	Documentation des fonctions membres	73
9.12.2.1	setHtml	73
9.12.2.2	setSource	73
9.13	Référence de la classe <code>src.notification.Notification</code>	74
9.13.1	Description détaillée	74
9.13.2	Documentation des constructeurs et destructeur	74
9.13.2.1	__init__	74
9.13.3	Documentation des fonctions membres	75
9.13.3.1	notify	75
9.13.4	Documentation des données membres	75
9.13.4.1	actions	75
9.13.4.2	app_icon	75
9.13.4.3	app_name	75
9.13.4.4	body	75
9.13.4.5	expire_timeout	75
9.13.4.6	hints	75
9.13.4.7	interface	76
9.13.4.8	replaces_id	76

9.13.4.9	summary	76
9.14	Référence de la classe <code>src.preferences.preferenceWindow</code>	76
9.14.1	Description détaillée	77
9.14.2	Documentation des constructeurs et destructeur	77
9.14.2.1	<code>__init__</code>	77
9.14.3	Documentation des fonctions membres	77
9.14.3.1	<code>enableDelay</code>	78
9.14.3.2	<code>setValues</code>	78
9.14.3.3	<code>updateRefreshLabel</code>	78
9.14.3.4	<code>values</code>	78
9.14.4	Documentation des données membres	78
9.14.4.1	<code>ui</code>	78
9.15	Référence de la classe <code>QAbstractTableModel</code>	79
9.16	Référence de la classe <code>QDialog</code>	79
9.17	Référence de la classe <code>QMainWindow</code>	80
9.18	Référence de la classe <code>QObject</code>	80
9.19	Référence de la classe <code>QStyledItemDelegate</code>	81
9.20	Référence de la classe <code>QTextBrowser</code>	82
9.21	Référence de la classe <code>Thread</code>	82
9.22	Référence de la classe <code>src.usbThread.threadCopyFromUSB</code>	83
9.22.1	Description détaillée	84
9.22.2	Documentation des constructeurs et destructeur	84
9.22.2.1	<code>__init__</code>	84
9.22.3	Documentation des fonctions membres	84
9.22.3.1	<code>todo</code>	84
9.22.4	Documentation des données membres	85
9.22.4.1	<code>cmd</code>	85
9.22.4.2	<code>rootPath</code>	85
9.23	Référence de la classe <code>src.usbThread.threadCopyToUSB</code>	85
9.23.1	Description détaillée	87
9.23.2	Documentation des constructeurs et destructeur	87
9.23.2.1	<code>__init__</code>	87
9.23.3	Documentation des fonctions membres	87
9.23.3.1	<code>threadType</code>	87

9.23.3.2	ToDo	87
9.23.4	Documentation des données membres	88
9.23.4.1	cmd	88
9.24	Référence de la classe <code>src.usbThread.threadDeleteInUSB</code>	88
9.24.1	Description détaillée	90
9.24.2	Documentation des constructeurs et destructeur	90
9.24.2.1	__init__	90
9.24.3	Documentation des fonctions membres	90
9.24.3.1	ToDo	90
9.24.4	Documentation des données membres	91
9.24.4.1	cmd	91
9.25	Référence de la classe <code>src.usbThread.threadMoveFromUSB</code>	91
9.25.1	Description détaillée	92
9.25.2	Documentation des constructeurs et destructeur	92
9.25.2.1	__init__	93
9.25.3	Documentation des fonctions membres	93
9.25.3.1	ToDo	93
9.25.4	Documentation des données membres	93
9.25.4.1	cmd	93
9.25.4.2	rootPath	93
9.26	Référence de la classe <code>src.usbThread.ThreadRegister</code>	94
9.26.1	Description détaillée	94
9.26.2	Documentation des constructeurs et destructeur	94
9.26.2.1	__init__	94
9.26.3	Documentation des fonctions membres	94
9.26.3.1	__str__	94
9.26.3.2	busy	95
9.26.3.3	pop	95
9.26.3.4	push	95
9.26.3.5	threadSet	95
9.26.4	Documentation des données membres	95
9.26.4.1	dico	95
9.27	Référence de la classe <code>src.usbDisk.uDisk</code>	96
9.27.1	Description détaillée	97

9.27.2	Documentation des constructeurs et destructeur . . . . .	98
9.27.2.1	__init__ . . . . .	98
9.27.3	Documentation des fonctions membres . . . . .	98
9.27.3.1	__getitem__ . . . . .	98
9.27.3.2	__str__ . . . . .	98
9.27.3.3	devicePropProxy . . . . .	99
9.27.3.4	ensureMounted . . . . .	99
9.27.3.5	file . . . . .	99
9.27.3.6	getFatUuid . . . . .	99
9.27.3.7	getFirstFat . . . . .	100
9.27.3.8	getProp . . . . .	100
9.27.3.9	headers . . . . .	100
9.27.3.10	isDosFat . . . . .	100
9.27.3.11	isMounted . . . . .	101
9.27.3.12	isTrue . . . . .	101
9.27.3.13	isUsbDisk . . . . .	101
9.27.3.14	master . . . . .	101
9.27.3.15	mountPoint . . . . .	101
9.27.3.16	showableProp . . . . .	102
9.27.3.17	title . . . . .	102
9.27.3.18	uniqueId . . . . .	102
9.27.3.19	unNumberProp . . . . .	102
9.27.3.20	valuableProperties . . . . .	103
9.27.4	Documentation des données membres . . . . .	103
9.27.4.1	checkable . . . . .	103
9.27.4.2	device . . . . .	103
9.27.4.3	device_prop . . . . .	103
9.27.4.4	fatuuid . . . . .	103
9.27.4.5	firstFat . . . . .	103
9.27.4.6	headers . . . . .	103
9.27.4.7	path . . . . .	103
9.27.4.8	selected . . . . .	103
9.27.4.9	stickid . . . . .	104
9.27.4.10	uuid . . . . .	104

9.28	Référence de la classe <code>src.ownedUsbDisk.uDisk</code>	104
9.28.1	Description détaillée	106
9.28.2	Documentation des constructeurs et destructeur	106
9.28.2.1	<code>__init__</code>	106
9.28.3	Documentation des fonctions membres	106
9.28.3.1	<code>__getitem__</code>	106
9.28.3.2	<code>ensureOwner</code>	107
9.28.3.3	<code>headers</code>	107
9.28.3.4	<code>ownerByDb</code>	107
9.28.3.5	<code>readQuirks</code>	107
9.28.3.6	<code>tattoo</code>	107
9.28.3.7	<code>uniqueId</code>	108
9.28.3.8	<code>visibleDir</code>	108
9.28.4	Documentation des données membres	108
9.28.4.1	<code>headers</code>	108
9.28.4.2	<code>model</code>	108
9.28.4.3	<code>owner</code>	108
9.28.4.4	<code>vendor</code>	108
9.28.4.5	<code>visibleDirs</code>	109
9.29	Référence de la classe <code>src.mainWindow.UsbDiskDelegate</code>	109
9.29.1	Description détaillée	110
9.29.2	Documentation des constructeurs et destructeur	110
9.29.2.1	<code>__init__</code>	110
9.29.3	Documentation des fonctions membres	110
9.29.3.1	<code>paint</code>	110
9.29.4	Documentation des données membres	110
9.29.4.1	<code>busyPixmap</code>	110
9.29.4.2	<code>okPixmap</code>	110
9.30	Référence de la classe <code>src.mainWindow.usbTableModel</code>	111
9.30.1	Description détaillée	112
9.30.2	Documentation des constructeurs et destructeur	112
9.30.2.1	<code>__init__</code>	112
9.30.3	Documentation des fonctions membres	112
9.30.3.1	<code>columnCount</code>	112

9.30.3.2	<a href="#">data</a>	112
9.30.3.3	<a href="#">headerData</a>	113
9.30.3.4	<a href="#">partition</a>	113
9.30.3.5	<a href="#">popCmd</a>	113
9.30.3.6	<a href="#">pushCmd</a>	113
9.30.3.7	<a href="#">rowCount</a>	113
9.30.3.8	<a href="#">setData</a>	113
9.30.3.9	<a href="#">sort</a>	114
9.30.3.10	<a href="#">updateOwnerColumn</a>	114
9.30.4	<a href="#">Documentation des données membres</a>	114
9.30.4.1	<a href="#">checkable</a>	114
9.30.4.2	<a href="#">donnees</a>	114
9.30.4.3	<a href="#">header</a>	114
9.30.4.4	<a href="#">pere</a>	114
<b>10</b>	<b>Documentation des fichiers</b>	<b>115</b>
10.1	<a href="#">Référence du fichier src/__init__.py</a>	115
10.2	<a href="#">Référence du fichier src/chooseInSticks.py</a>	115
10.3	<a href="#">Référence du fichier src/copyToDialog1.py</a>	115
10.4	<a href="#">Référence du fichier src/db.py</a>	116
10.5	<a href="#">Référence du fichier src/deviceListener.py</a>	116
10.6	<a href="#">Référence du fichier src/diskFull.py</a>	117
10.7	<a href="#">Référence du fichier src/globaldef.py</a>	117
10.8	<a href="#">Référence du fichier src/help.py</a>	118
10.9	<a href="#">Référence du fichier src/mainWindow.py</a>	118
10.10	<a href="#">Référence du fichier src/marques.py</a>	118
10.11	<a href="#">Référence du fichier src/mytextbrowser.py</a>	119
10.12	<a href="#">Référence du fichier src/notification.py</a>	119
10.13	<a href="#">Référence du fichier src/ownedUsbDisk.py</a>	119
10.14	<a href="#">Référence du fichier src/preferences.py</a>	120
10.15	<a href="#">Référence du fichier src/scolasync.py</a>	120
10.16	<a href="#">Référence du fichier src/usbDisk.py</a>	121
10.17	<a href="#">Référence du fichier src/usbThread.py</a>	121
10.18	<a href="#">Référence du fichier src/version.py</a>	122



# Chapitre 1

## ScolaSync

### 1.1 But de l'application

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

### 1.2 CAHIER DE CHARGES DE SCOLASYNC

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

## 1.3 Licence

### ScolaSync version 1.0 :

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

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

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

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

## 1.4 Support

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

## 1.5 Architecture de ScolaSync

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

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

## Chapitre 2

# Hiérarchie de répertoires

### 2.1 Répertoires

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

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



## Chapitre 3

# Index des espaces de nommage

### 3.1 Paquetages

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

<a href="#">scolasync</a> (Scolasync est un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB ) . . . . .	15
<a href="#">src</a> . . . . .	15
<a href="#">src.chooseInSticks</a> . . . . .	16
<a href="#">src.copyToDialog1</a> . . . . .	16
<a href="#">src.db</a> . . . . .	17
<a href="#">src.deviceListener</a> . . . . .	20
<a href="#">src.diskFull</a> . . . . .	21
<a href="#">src.globaldef</a> . . . . .	21
<a href="#">src.help</a> . . . . .	23
<a href="#">src.mainWindow</a> . . . . .	23
<a href="#">src.marques</a> . . . . .	25
<a href="#">src.mytextbrowser</a> . . . . .	25
<a href="#">src.notification</a> . . . . .	25
<a href="#">src.ownedUsbDisk</a> . . . . .	26
<a href="#">src.preferences</a> . . . . .	26
<a href="#">src.scolasync</a> . . . . .	27
<a href="#">src.usbDisk</a> . . . . .	29
<a href="#">src.usbThread</a> . . . . .	30
<a href="#">src.version</a> . . . . .	31



## Chapitre 4

# Index des classes

### 4.1 Hiérarchie des classes

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

src.usbDisk.Available . . . . .	39
src.ownedUsbDisk.Available . . . . .	37
src.deviceListener.DeviceListener . . . . .	54
src.notification.Notification . . . . .	74
QAbstractTableModel . . . . .	79
src.mainWindow.usbTableModel . . . . .	111
QDialog . . . . .	79
src.chooseInSticks.chooseDialog . . . . .	46
src.copyToDialog1.copyToDialog1 . . . . .	51
src.help.helpWindow . . . . .	59
src.preferences.preferenceWindow . . . . .	76
QMainWindow . . . . .	80
src.diskFull.mainWindow . . . . .	61
src.mainWindow.mainWindow . . . . .	63
QObject . . . . .	80
src.ownedUsbDisk.uDisk . . . . .	104
QStyledItemDelegate . . . . .	81
src.mainWindow.CheckBoxDelegate . . . . .	45
src.mainWindow.DiskSizeDelegate . . . . .	57
src.mainWindow.UsbDiskDelegate . . . . .	109
QTextBrowser . . . . .	82
src.mytextbrowser.myTextBrowser . . . . .	72
Thread . . . . .	82
src.usbThread.abstractThreadUSB . . . . .	33
src.usbThread.threadCopyFromUSB . . . . .	83
src.usbThread.threadCopyToUSB . . . . .	85
src.usbThread.threadDeleteInUSB . . . . .	88

src.usbThread.threadMoveFromUSB . . . . .	91
src.usbThread.ThreadRegister . . . . .	94
src.usbDisk.uDisk . . . . .	96
src.ownedUsbDisk.uDisk . . . . .	104

## Chapitre 5

# Index des classes

### 5.1 Liste des classes

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

<a href="#">src.usbThread.abstractThreadUSB</a> (Une classe abstraite Cette classe sert de creuset pour les classe servant aux copies et aux effacement ) . . .	33
<a href="#">src.ownedUsbDisk.Available</a> (Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires ) . . . . .	37
<a href="#">src.usbDisk.Available</a> (Une classe pour représenter la collection des disques USB connectés ) . . . . .	39
<a href="#">src.mainWindow.CheckBoxDelegate</a> . . . . .	45
<a href="#">src.chooseInSticks.chooseDialog</a> (Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB ) . . . . .	46
<a href="#">src.copyToDialog1.copyToDialog1</a> (Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB ) . . . . .	51
<a href="#">src.deviceListener.DeviceListener</a> . . . . .	54
<a href="#">src.mainWindow.DiskSizeDelegate</a> (Classe pour figurer la taille de la mémoire du baladeur ) . . . . .	57
<a href="#">src.help.helpWindow</a> . . . . .	59
<a href="#">src.diskFull.mainWindow</a> . . . . .	61
<a href="#">src.mainWindow.mainWindow</a> . . . . .	63
<a href="#">src.mytextbrowser.myTextBrowser</a> (Une classe qui ouvre Firefox quand on clique sur un lien externe ) . . . . .	72
<a href="#">src.notification.Notification</a> (Une classe pour afficher des notifications à l'écran )	74
<a href="#">src.preferences.preferenceWindow</a> . . . . .	76
<a href="#">QAbstractTableModel</a> . . . . .	79
<a href="#">QDialog</a> . . . . .	79
<a href="#">QMainWindow</a> . . . . .	80
<a href="#">QObject</a> . . . . .	80
<a href="#">QStyledItemDelegate</a> . . . . .	81
<a href="#">QTextBrowser</a> . . . . .	82
<a href="#">Thread</a> . . . . .	82

<a href="#">src.usbThread.threadCopyFromUSB</a> (Classe pour les threads copiant depuis les clés USB ) . . . . .	83
<a href="#">src.usbThread.threadCopyToUSB</a> (Classe pour les threads copiant vers les clés USB ) . . . . .	85
<a href="#">src.usbThread.threadDeleteInUSB</a> (Classe pour les threads effaçant des sous-arbres dans les clés USB ) . . . . .	88
<a href="#">src.usbThread.threadMoveFromUSB</a> (Classe pour les threads déplaçant des fichiers depuis les clés USB ) . . . . .	91
<a href="#">src.usbThread.ThreadRegister</a> (Une classe pour tenir un registre des threads concernant les baladeurs ) . . . . .	94
<a href="#">src.usbDisk.uDisk</a> (Une classe pour représenter un disque ou une partition ) .	96
<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 ) . . . . .	104
<a href="#">src.mainWindow.UsbDiskDelegate</a> (Classe pour identifier le baladeur dans le tableau ) . . . . .	109
<a href="#">src.mainWindow.usbTableModel</a> (Un modèle de table pour des séries de clés USB ) . . . . .	111

## Chapitre 6

# Index des fichiers

### 6.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

<a href="#">src/__init__.py</a>	115
<a href="#">src/chooseInSticks.py</a>	115
<a href="#">src/copyToDialog1.py</a>	115
<a href="#">src/db.py</a>	116
<a href="#">src/deviceListener.py</a>	116
<a href="#">src/diskFull.py</a>	117
<a href="#">src/globaldef.py</a>	117
<a href="#">src/help.py</a>	118
<a href="#">src/mainWindow.py</a>	118
<a href="#">src/marques.py</a>	118
<a href="#">src/mytextbrowser.py</a>	119
<a href="#">src/notification.py</a>	119
<a href="#">src/ownedUsbDisk.py</a>	119
<a href="#">src/preferences.py</a>	120
<a href="#">src/scolasync.py</a>	120
<a href="#">src/usbDisk.py</a>	121
<a href="#">src/usbThread.py</a>	121
<a href="#">src/version.py</a>	122

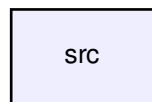


## Chapitre 7

# Documentation des répertoires

### 7.1 Répertoire de référence de src/

Directory dependency graph for src/ :



#### Fichiers

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



## Chapitre 8

# Documentation des espaces de nommage

### 8.1 Paquetage scolasync

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

#### 8.1.1 Description détaillée

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

### 8.2 Paquetage src

#### Paquetages

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

## 8.3 Paquetage src.chooseInSticks

### Classes

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

### Variables

- string `licenceEn`

#### 8.3.1 Documentation des variables

##### 8.3.1.1 string `src : :chooseInSticks.licenceEn`

##### Valeur initiale :

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

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

## 8.4 Paquetage src.copyToDialog1

### Classes

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

### Variables

- string `licenceEn`

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

### 8.4.1 Documentation des variables

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

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

#### 8.4.1.2 string `src` : `:copyToDialog1.licenceEn`

Valeur initiale :

```

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

```

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

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

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

## 8.5 Paquetage 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](#) = {}
- [database](#) = None
- [cursor](#) = None

## 8.5.1 Documentation des fonctions

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

Vérifie si la base de données reste compatible.

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

Définition à la ligne 57 du fichier db.py.

### 8.5.1.2 def src.db.hasStudent ( *student* )

vérifie qu'un étudiant est déjà connu

#### Paramètres

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

#### Renvoie

True si le propriétaire existe déjà

Définition à la ligne 79 du fichier db.py.

### 8.5.1.3 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
---------------	--------------------------

**Renvoie**

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

Définition à la ligne 92 du fichier db.py.

**8.5.1.4 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 37 du fichier db.py.

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

renvoie les préférences de ScolaSync

**Renvoie**

un dictionnaire de préférences

Définition à la ligne 125 du fichier db.py.

**8.5.1.6 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 111 du fichier db.py.

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

définit le nouveau nom du répertoire de travail préféré.

Définition à la ligne 157 du fichier db.py.

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

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

Définition à la ligne 101 du fichier db.py.

#### 8.5.1.9 `def src.db.writePrefs ( prefs )`

inscrit les préférences

##### Paramètres

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

Définition à la ligne 183 du fichier db.py.

#### 8.5.1.10 `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 167 du fichier db.py.

### 8.5.2 Documentation des variables

#### 8.5.2.1 `src : :db.cursor = None`

Définition à la ligne 30 du fichier db.py.

#### 8.5.2.2 `src : :db.database = None`

Définition à la ligne 29 du fichier db.py.

#### 8.5.2.3 `dictionary src : :db.licence = {}`

Définition à la ligne 4 du fichier db.py.

## 8.6 Paquetage `src.deviceListener`

### Classes

– class `DeviceListener`

### Variables

– dictionary `licence = {}`

### 8.6.1 Documentation des variables

8.6.1.1 dictionary src : :deviceListener.licence = {}

Définition à la ligne 4 du fichier deviceListener.py.

## 8.7 Paquetage src.diskFull

### Classes

– class `mainWindow`

### Fonctions

– def `sceneWithUsage`

### Variables

– dictionary `licence` = {}

### 8.7.1 Documentation des fonctions

8.7.1.1 def src.diskFull.sceneWithUsage ( *parent*, *rect*, *percent* )

#### Paramètres

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

#### Renvoie

une QGraphicsScene avec un symbole d'occupation du disque

Définition à la ligne 60 du fichier diskFull.py.

### 8.7.2 Documentation des variables

8.7.2.1 dictionary src : :diskFull.licence = {}

Définition à la ligne 5 du fichier diskFull.py.

## 8.8 Paquetage 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.*
- string `userShareDir` = "~/scolasync"
- string `logFileName` = "~/scolasync/scolasync.log"
- string `markFileName` = "~/scolasync/marques.py"

### 8.8.1 Documentation des fonctions

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

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

#### Paramètres

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

Définition à la ligne 50 du fichier globaldef.py.

### 8.8.2 Documentation des variables

#### 8.8.2.1 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.

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

Définition à la ligne 37 du fichier globaldef.py.

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

Définition à la ligne 38 du fichier globaldef.py.

8.8.2.4 `string src : :globaldef : :userShareDir = "~/scolasync"`

Définition à la ligne 36 du fichier globaldef.py.

## 8.9 Paquetage src.help

### Classes

– class [helpWindow](#)

### Variables

– dictionary [licence](#) = {}

### 8.9.1 Documentation des variables

8.9.1.1 `dictionary src : :help.licence = {}`

Définition à la ligne 5 du fichier help.py.

## 8.10 Paquetage 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

### 8.10.1 Documentation des fonctions

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

Définition à la ligne 766 du fichier mainWindow.py.

#### 8.10.1.2 def src.mainWindow.registerCmd ( *cmd*, *partition* )

enregistre la commande cmd pour la partition donnée

#### Paramètres

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

Définition à la ligne 51 du fichier mainWindow.py.

### 8.10.2 Documentation des variables

#### 8.10.2.1 dictionary src : :mainWindow.activeThreads = {}

Définition à la ligne 39 du fichier mainWindow.py.

#### 8.10.2.2 src : :mainWindow.lastCommand = None

Définition à la ligne 43 du fichier mainWindow.py.

#### 8.10.2.3 dictionary src : :mainWindow.licence = {}

Définition à la ligne 5 du fichier mainWindow.py.

#### 8.10.2.4 dictionary src : :mainWindow.pastCommands = {}

Définition à la ligne 42 du fichier mainWindow.py.

## 8.11 Paquetage src.marques

## 8.12 Paquetage src.mytextbrowser

### Classes

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

### Variables

- dictionary `licence` = {}

#### 8.12.1 Documentation des variables

8.12.1.1 dictionary `src : :mytextbrowser.licence` = {}

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

## 8.13 Paquetage src.notification

### Classes

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

### Variables

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

#### 8.13.1 Documentation des variables

8.13.1.1 dictionary `src : :notification.licence` = {}

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

8.13.1.2 tuple `src : :notification.notif`

#### Valeur initiale :

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

```
5 expire_timeout=7000)
```

Définition à la ligne 75 du fichier notification.py.

## 8.14 Paquetage src.ownedUsbDisk

### Classes

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

### Fonctions

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

### Variables

- dictionary `licence` = {}

### 8.14.1 Documentation des fonctions

8.14.1.1 `def src.ownedUsbDisk.editRecord ( owd, prompt = " " )`

édition de la base de données.

#### Paramètres

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

Définition à la ligne 43 du fichier ownedUsbDisk.py.

### 8.14.2 Documentation des variables

8.14.2.1 dictionary `src : :ownedUsbDisk.licence` = {}

Définition à la ligne 4 du fichier ownedUsbDisk.py.

## 8.15 Paquetage src.preferences

### Classes

- class `preferenceWindow`

### Variables

- dictionary `licence` = {}

#### 8.15.1 Documentation des variables

8.15.1.1 dictionary `src : preferences.licence` = {}

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

## 8.16 Paquetage src.scolasync

### Fonctions

- def `usage`  
*affiche le mode d'emploi à la console*
- def `run`  
*Le lancement de l'application.*

### Variables

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

#### 8.16.1 Documentation des fonctions

8.16.1.1 def `src.scolasync.run` ( )

Le lancement de l'application.

Définition à la ligne 150 du fichier `scolasync.py`.

8.16.1.2 def `src.scolasync.usage` ( )

affiche le mode d'emploi à la console

Définition à la ligne 138 du fichier `scolasync.py`.

#### 8.16.2 Documentation des variables

### 8.16.2.1 dictionary src : :scolasync.licence = {}

Définition à la ligne 85 du fichier scolasync.py.

### 8.16.2.2 string src : :scolasync.licenceEn

Valeur initiale :

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

Définition à la ligne 86 du fichier scolasync.py.

### 8.16.2.3 string src : :scolasync.licenceFr

Valeur initiale :

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

Définition à la ligne 109 du fichier scolasync.py.

## 8.17 Paquetage src.usbDisk

### Classes

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

### Variables

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

#### 8.17.1 Documentation des variables

##### 8.17.1.1 dictionary `src : :usbDisk.licence` = {}

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

##### 8.17.1.2 string `src : :usbDisk.licence_en`

#### Valeur initiale :

```
1 """
2     file usbDisk.py
3     this file is part of the project scolasync
4
5     Copyright (C) 2010 Georges Khaznadar <georgesk@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 5 du fichier `usbDisk.py`.

##### 8.17.1.3 tuple `src : :usbDisk.machin` = `Available()`

Définition à la ligne 564 du fichier `usbDisk.py`.

## 8.18 Paquetage src.usbThread

### Classes

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

### Variables

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

#### 8.18.1 Documentation des variables

##### 8.18.1.1 int src : :usbThread.\_threadNumber = 0

Définition à la ligne 27 du fichier usbThread.py.

##### 8.18.1.2 string src : :usbThread.licenceEn

#### Valeur initiale :

```

1 """
2     file usbThread.py
3     this file is part of the project scolasync
4
5     Copyright (C) 2010 Georges Khaznadar <georgesk@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 4 du fichier usbThread.py.

## 8.19 Paquetage src.version

### Fonctions

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

### Variables

- dictionary `licence` = {}

#### 8.19.1 Documentation des fonctions

8.19.1.1 def `src.version.major` ( )

##### Renvoie

le numéro majeur de version

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

8.19.1.2 def `src.version.minor` ( )

##### Renvoie

le numéro mineur de version

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

8.19.1.3 def `src.version.version` ( )

##### Renvoie

l'identifiant de la version

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

#### 8.19.2 Documentation des variables

8.19.2.1 dictionary `src : :version.licence` = {}

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



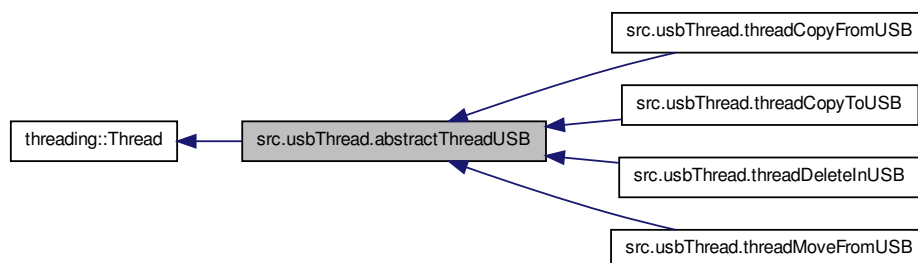
## Chapitre 9

# Documentation des classes

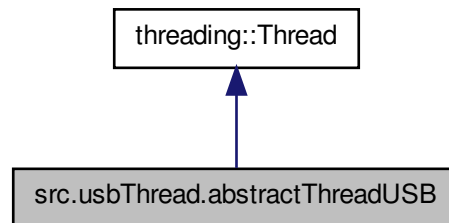
### 9.1 Référence de la classe `src.usbThread.abstractThreadUSB`

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

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



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



### Fonctions membres publiques

- def `__init__`  
*Constructeur Crée un thread pour copier une liste de fichiers vers une clé USB.*
- def `__str__`  
*Renvoie une chaîne informative sur le thread.*
- def `threadType`
- def `todo`  
*La fonction abstraite pour les choses à faire.*
- def `protectPath`  
*protège une chaîne unicode par des guillemets (pour un shell POSIX) à moins que celle-ci contienne des jokers.*

### Attributs publics

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

#### 9.1.1 Description détaillée

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

Définition à la ligne 153 du fichier `usbThread.py`.

#### 9.1.2 Documentation des constructeurs et destructeur

9.1.2.1 `def src.usbThread.abstractThreadUSB.__init__( self, ud, fileList, subdir, dest = None, logfile = "/dev/null", 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 166 du fichier `usbThread.py`.

### 9.1.3 Documentation des fonctions membres

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

Renvoie une chaîne informative sur le thread.

#### Renvoie

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

Définition à la ligne 186 du fichier `usbThread.py`.

9.1.3.2 `def src.usbThread.abstractThreadUSB.protectPath ( self, s )`

protège une chaîne unicode par des guillemets (pour un shell POSIX) à moins que celle-ci contienne des jokers.

#### Paramètres

<i>s</i>	la chaîne d'entrée
----------	--------------------

#### Renvoie

la même, protégée si nécessaire

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

9.1.3.3 `def src.usbThread.abstractThreadUSB.threadType ( self )`

#### Renvoie

une chaîne courte qui informe sur le type de thread

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

Définition à la ligne 201 du fichier usbThread.py.

9.1.3.4 `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.threadCopyToUSB](#), [src.usbThread.threadCopyFromUSB](#), [src.usbThread.threadMoveFromUSB](#), et [src.usbThread.threadDeleteInUSB](#).

Définition à la ligne 213 du fichier usbThread.py.

### 9.1.4 Documentation des données membres

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

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

Définition à la ligne 166 du fichier usbThread.py.

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

Définition à la ligne 166 du fichier usbThread.py.

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

Définition à la ligne 166 du fichier usbThread.py.

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

Définition à la ligne 166 du fichier usbThread.py.

#### 9.1.4.5 `src.usbThread.abstractThreadUSB.parent`

Définition à la ligne 166 du fichier usbThread.py.

9.1.4.6 `src.usbThread.abstractThreadUSB.subdir`

Définition à la ligne 166 du fichier `usbThread.py`.

9.1.4.7 `src.usbThread.abstractThreadUSB.ud`

Définition à la ligne 166 du fichier `usbThread.py`.

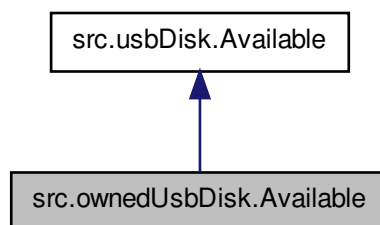
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/usbThread.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

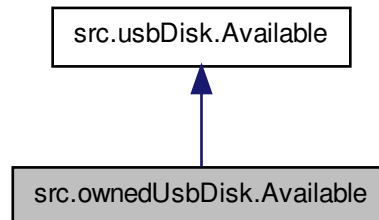
## 9.2 Référence de la classe `src.ownedUsbDisk.Available`

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

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



Graphe 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`

## 9.2.1 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 228 du fichier `ownedUsbDisk.py`.

## 9.2.2 Documentation des constructeurs et destructeur

**9.2.2.1** `def src.ownedUsbDisk.Available.__init__( self, checkable = False, access = "disk", diskClass = uDisk, diskDict = None, noLoop = False )`

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**

<i>checkable</i>	True si on veut pouvoir sélectionner des disques en cochant
<i>access</i>	le mode d'accès : 'disk' ou 'firstFat'
<i>diskClass</i>	la classe d'objets à créer pour chaque disque
<i>diskDict</i>	un dictionnaire des disque maintenu par <a href="#">deviceListener</a>
<i>noLoop</i>	doit être True pour éviter de lancer un dialogue

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

**9.2.3 Documentation des fonctions membres****9.2.3.1 `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 250 du fichier `ownedUsbDisk.py`.

**9.2.4 Documentation des données membres****9.2.4.1 `src.ownedUsbDisk.Available.noLoop`**

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

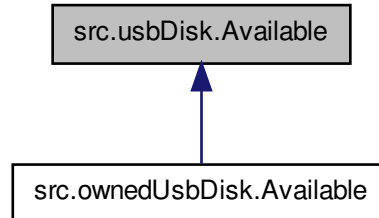
La documentation de cette classe a été générée à partir du fichier suivant :

- `src/ownedUsbDisk.py` ([Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)](#))

**9.3 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 : \* marque le disque avec l'uuid de la première partition FAT.*
- def `hasDev`

### Attributs publics

- `checkable`  
*print "GRRRR should use diskDict=", diskDict*
- `access`
- `bus`
- `disks`
- `enumDev`
- `firstFats`

– [fatPaths](#)

### 9.3.1 Description détaillée

une classe pour représenter la collection des disques USB connectés

les attributs publics sont :

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

Définition à la ligne 371 du fichier `usbDisk.py`.

### 9.3.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

<i>checkable</i>	: vrai si on veut pouvoir cocher les disques de la collection. Faux par défaut.
<i>access</i>	définit le type d'accès souhaité. Par défaut, c'est "disk" c'est à dire qu'on veut la liste des disques USB. Autres valeurs possibles : "firstFat" pour les premières partitions vfat.
<i>diskClass</i>	la classe de disques à créer
<i>diskDict</i>	un dictionnaire des disque maintenu par <a href="#">deviceListener</a>

Définition à la ligne 384 du fichier `usbDisk.py`.

### 9.3.3 Documentation des fonctions membres

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

Renvoie le nième disque.

Le fonctionnement dépend du paramètre `self.access`

#### Paramètres

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

#### Renvoie

le nième disque USB connecté

Définition à la ligne 499 du fichier usbDisk.py.

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

Renseigne sur la longueur de la collection.

Le fonctionnement dépend du paramètre `self.access`

#### **Renvoie**

la longueur de la collection de disques renvoyée

Définition à la ligne 511 du fichier usbDisk.py.

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

Fournit une représentation imprimable.

#### **Renvoie**

une représentation imprimable de la collection

Définition à la ligne 480 du fichier usbDisk.py.

**9.3.3.4** `def src.usbDisk.Available.__trunc__( self )`

#### **Renvoie**

le nombre de medias connectés

Définition à la ligne 434 du fichier usbDisk.py.

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

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

#### **Paramètres**

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

#### **Renvoie**

vrai si `other` semble être la même collection de disques USB

Définition à la ligne 444 du fichier usbDisk.py.

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

Permet de déterminer si un disque est dans la collection.

**Paramètres**

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

**Renvoie**

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

Définition à la ligne 454 du fichier `usbDisk.py`.

**9.3.3.7 `def src.usbDisk.Available.finishInit ( self )`**

Fin de l'initialisation.

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

Définition à la ligne 416 du fichier `usbDisk.py`.

**9.3.3.8 `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**

<code>setOwners</code>	si égale à <code>True</code> , 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 529 du fichier `usbDisk.py`.

**9.3.3.9 `def src.usbDisk.Available.hasDev ( self, dev )`****Paramètres**

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

**Renvoie**

`True` si la partition est dans la liste des partions disponibles

Définition à la ligne 554 du fichier `usbDisk.py`.

**9.3.3.10 `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 424 du fichier `usbDisk.py`.

#### 9.3.3.11 `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 464 du fichier usbDisk.py.

### 9.3.4 Documentation des données membres

#### 9.3.4.1 `src.usbDisk.Available.access`

Définition à la ligne 385 du fichier usbDisk.py.

#### 9.3.4.2 `src.usbDisk.Available.bus`

Définition à la ligne 385 du fichier usbDisk.py.

#### 9.3.4.3 `src.usbDisk.Available.checkable`

`print "GRRRRR should use diskDict=", diskDict`

Définition à la ligne 385 du fichier usbDisk.py.

#### 9.3.4.4 `src.usbDisk.Available.disks`

Définition à la ligne 385 du fichier usbDisk.py.

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

Définition à la ligne 385 du fichier usbDisk.py.

#### 9.3.4.6 `src.usbDisk.Available.fatPaths`

Définition à la ligne 529 du fichier usbDisk.py.

#### 9.3.4.7 `src.usbDisk.Available.firstFats`

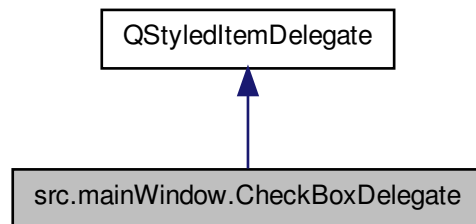
Définition à la ligne 424 du fichier usbDisk.py.

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

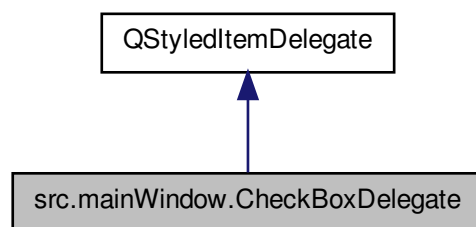
– [src/usbDisk.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/server\)\)](#)

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

Graphe d'héritage de src.mainWindow.CheckBoxDelegate :



Graphe de collaboration de src.mainWindow.CheckBoxDelegate :



### Fonctions membres publiques

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

#### 9.4.1 Description détaillée

Définition à la ligne 772 du fichier `mainWindow.py`.

## 9.4.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 773 du fichier mainWindow.py.

## 9.4.3 Documentation des fonctions membres

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

Définition à la ligne 787 du fichier mainWindow.py.

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

Définition à la ligne 776 du fichier mainWindow.py.

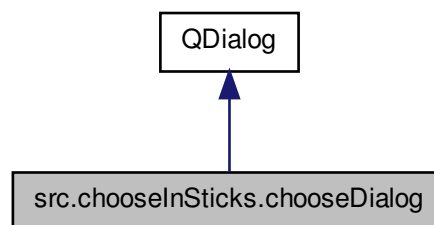
La documentation de cette classe a été générée à partir du fichier suivant :

– [src/mainWindow.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

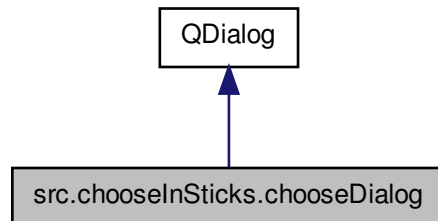
## 9.5 Référence de la classe src.chooseInSticks.chooseDialog

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

Graphe d'héritage de src.chooseInSticks.chooseDialog :



Graphe de collaboration de `src.chooseInSticks.chooseDialog` :



### Fonctions membres publiques

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

### 9.5.1 Description détaillée

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

Définition à la ligne 34 du fichier chooseInSticks.py.

### 9.5.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

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

Définition à la ligne 44 du fichier chooseInSticks.py.

### 9.5.3 Documentation des fonctions membres

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

Fonction de rappel quand un item de la liste est activé

#### Paramètres

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

Définition à la ligne 244 du fichier chooseInSticks.py.

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

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

#### Paramètres

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

Définition à la ligne 278 du fichier chooseInSticks.py.

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

#### Renvoie

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

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

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

changement du répertoire de travail

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

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

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

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

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

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

#### Paramètres

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

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

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

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

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

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

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

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

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

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

Définition à la ligne 262 du fichier `chooseInSticks.py`.

#### 9.5.3.10 `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 294 du fichier `chooseInSticks.py`.

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

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

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

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

##### **Renvoie**

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

Définition à la ligne 162 du fichier `chooseInSticks.py`.

#### 9.5.3.13 `def src.chooseInSticks.chooseDialog.selectedDiskOwner ( self )`

##### **Renvoie**

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

Définition à la ligne 174 du fichier `chooseInSticks.py`.

### 9.5.4 Documentation des données membres

#### 9.5.4.1 `src.chooseInSticks.chooseDialog.mainWindow`

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

## 9.5.4.2 src.chooseInSticks.chooseDialog.ownedUsbDictionary

peuplement de la zone des noms de baladeurs

Définition à la ligne 49 du fichier chooseInSticks.py.

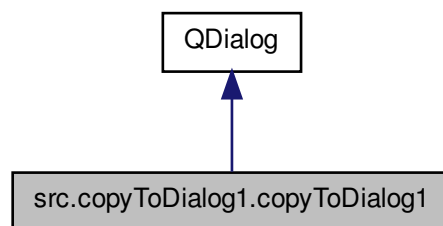
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/chooseInSticks.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

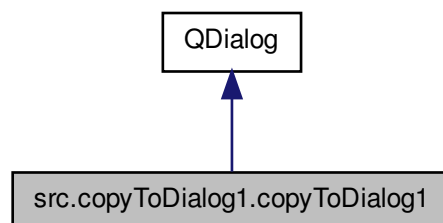
## 9.6 Référence de la classe src.copyToDialog1.copyToDialog1

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

Graphe d'héritage de src.copyToDialog1.copyToDialog1 :



Graphe de collaboration de src.copyToDialog1.copyToDialog1 :



### Fonctions membres publiques

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

### Attributs publics

- [mainWindow](#)
- [ok](#)

#### 9.6.1 Description détaillée

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

##### Paramètres

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

Définition à la ligne 37 du fichier copyToDialog1.py.

#### 9.6.2 Documentation des fonctions membres

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

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

Définition à la ligne 74 du fichier copyToDialog1.py.

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

Change le répertoire courant si possible.

#### Paramètres

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

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

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

changement du répertoire de travail

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

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

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

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

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

Affiche la taille de la sélection courante.

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

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

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

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

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

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

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

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

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

#### Renvoie

une liste de `QStrings`

Définition à la ligne 203 du fichier copyToDialog1.py.

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

Choisit un répertoire pour la liste source.

#### Paramètres

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

Définition à la ligne 101 du fichier copyToDialog1.py.

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

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

Définition à la ligne 90 du fichier copyToDialog1.py.

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

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

Définition à la ligne 124 du fichier copyToDialog1.py.

### 9.6.3 Documentation des données membres

9.6.3.1 `src.copyToDialog1.copyToDialog1.mainWindow`

Définition à la ligne 54 du fichier copyToDialog1.py.

9.6.3.2 `src.copyToDialog1.copyToDialog1.ok`

Définition à la ligne 1 du fichier copyToDialog1.py.

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

- [src/copyToDialog1.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

## 9.7 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 `isVfatUsb`  
*Décide si une partition est de type vfat, et connectée par USB.*

### Attributs publics

- `bus`
- `manager`
- `interface`
- `connectedVolumes`
- `widget`

#### 9.7.1 Description détaillée

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

#### 9.7.2 Documentation des constructeurs et destructeur

9.7.2.1 `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 34 du fichier `deviceListener.py`.

#### 9.7.3 Documentation des fonctions membres

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

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

##### Paramètres

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

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

9.7.3.2 `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.

#### 9.7.3.3 `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 97 du fichier deviceListener.py.

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

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

**Paramètres**

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

Définition à la ligne 126 du fichier deviceListener.py.

#### 9.7.3.5 `def src.deviceListener.DeviceListener.pollDevices ( self )`

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

Définition à la ligne 54 du fichier deviceListener.py.

#### 9.7.3.6 `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 <code>EnumerateDevices()</code> , ou une simple chaîne de caractères, clé dans le tableau <code>connectedVolumes</code>
------------	---

**Renvoie**

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

Définition à la ligne 113 du fichier deviceListener.py.

## 9.7.4 Documentation des données membres

#### 9.7.4.1 src.deviceListener.DeviceListener.bus

Définition à la ligne 34 du fichier deviceListener.py.

#### 9.7.4.2 src.deviceListener.DeviceListener.connectedVolumes

Définition à la ligne 34 du fichier deviceListener.py.

#### 9.7.4.3 src.deviceListener.DeviceListener.interface

Définition à la ligne 34 du fichier deviceListener.py.

#### 9.7.4.4 src.deviceListener.DeviceListener.manager

Définition à la ligne 34 du fichier deviceListener.py.

#### 9.7.4.5 src.deviceListener.DeviceListener.widget

Définition à la ligne 34 du fichier deviceListener.py.

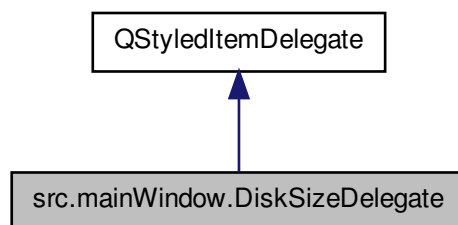
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/deviceListener.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

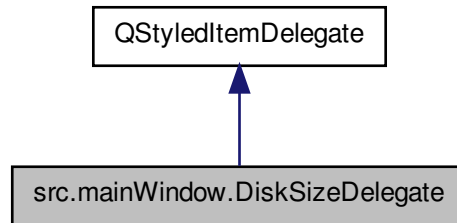
## 9.8 Référence de la classe src.mainWindow.DiskSizeDelegate

Classe pour figurer la taille de la mémoire du baladeur.

Grphe d'héritage de src.mainWindow.DiskSizeDelegate :



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



### Fonctions membres publiques

- def `__init__`
- def `paint`
- def `val2txt`

#### 9.8.1 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 le plus parproprée.

Définition à la ligne 839 du fichier `mainWindow.py`.

#### 9.8.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 840 du fichier `mainWindow.py`.

#### 9.8.3 Documentation des fonctions membres

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

Définition à la ligne 844 du fichier `mainWindow.py`.

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

#### Renvoie

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

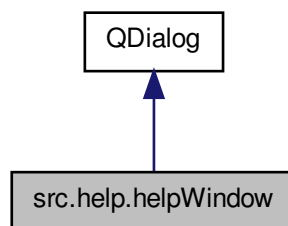
Définition à la ligne 865 du fichier mainWindow.py.

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

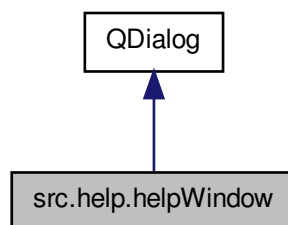
- [src/mainWindow.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

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

Graphe d'héritage de src.help.helpWindow :



Graphe de collaboration de src.help.helpWindow :



### Fonctions membres publiques

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

### Attributs publics

- `ui`

#### 9.9.1 Description détaillée

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

#### 9.9.2 Documentation des constructeurs et destructeur

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

Le constructeur.

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

#### 9.9.3 Documentation des fonctions membres

9.9.3.1 `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 53 du fichier `help.py`.

#### 9.9.4 Documentation des données membres

9.9.4.1 `src.help.helpWindow.ui`

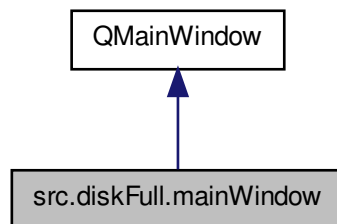
Définition à la ligne 37 du fichier `help.py`.

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

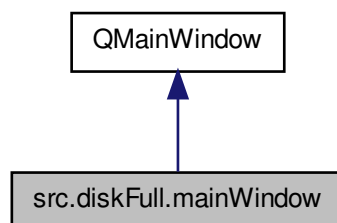
- `src/help.py` ([Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/server\)](#))

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

Graphe d'héritage de src.diskFull.mainWindow :



Graphe de collaboration de src.diskFull.mainWindow :



### Fonctions membres publiques

- def `__init__`  
*Le constructeur.*

### Attributs publics

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

### 9.10.1 Description détaillée

Définition à la ligne 29 du fichier diskFull.py.

### 9.10.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

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

Définition à la ligne 39 du fichier diskFull.py.

### 9.10.3 Documentation des données membres

#### 9.10.3.1 `src.diskFull.mainWindow.total`

Définition à la ligne 39 du fichier diskFull.py.

#### 9.10.3.2 `src.diskFull.mainWindow.ui`

Définition à la ligne 39 du fichier diskFull.py.

#### 9.10.3.3 `src.diskFull.mainWindow.used`

Définition à la ligne 39 du fichier diskFull.py.

#### 9.10.3.4 `src.diskFull.mainWindow.v`

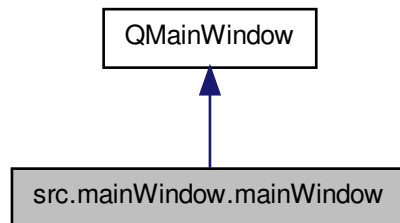
Définition à la ligne 39 du fichier diskFull.py.

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

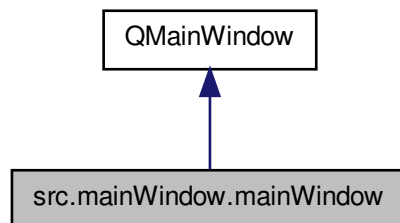
– [src/diskFull.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/server\)\)](#)

## 9.11 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 `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 [diskSizeData](#)
- def [diskFromOwner](#)  
*trouve le disque qui correspond à un propriétaire*
- def [editOwner](#)  
*Édition du propriétaire d'une clé.*
- def [updateButtons](#)  
*Désactive ou active les flèches selon que l'option correspondante est possible ou non.*
- def [preference](#)  
*lance le dialogue des préférences*
- def [delFiles](#)  
*Lance l'action de supprimer des fichiers ou des répertoires dans les clés USB.*
- def [copyTo](#)  
*Lance l'action de copier vers les clés USB.*
- def [copyFrom](#)  
*Lance l'action de copier depuis les clés USB.*
- def [redoCmd](#)  
*Relance la dernière commande, mais en l'appliquant seulement aux baladeurs nouvellement branchés.*
- def [help](#)  
*Affiche le widget d'aide.*
- def [umount](#)  
*Démonte et détache les clés USB affichées.*
- def [connectTableModel](#)  
*Connecte le modèle de table à la table.*
- def [checkDisks](#)  
*fonction relancée périodiquement pour vérifier s'il y a un changement dans le baladeurs, et signaler dans le tableau les threads en cours.*
- def [sameDiskData](#)
- def [flashLCD](#)  
*change le style de l'afficheur LCD pendant une fraction de seconde*
- def [normalLCD](#)  
*remet le style par défaut pour l'afficheur LCD*

### Attributs publics

- [locale](#)
- [ui](#)
- [t](#)
- [proxy](#)
- [opts](#)
- [timer](#)
- [listener](#)
- [operations](#)
- [oldThreads](#)
- [flashTimer](#)
- [checkDisksLock](#)
- [iconRedo](#)
- [iconStop](#)
- [redoToolTip](#)
- [redoStatusTip](#)
- [stopToolTip](#)

- stopStatusTip
- workdir
- refreshEnabled
- refreshDelay
- manFileLocation
- checkable
- mv
- header
- visibleheader
- tm

### 9.11.1 Description détaillée

Définition à la ligne 59 du fichier mainWindow.py.

### 9.11.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

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

Définition à la ligne 67 du fichier mainWindow.py.

### 9.11.3 Documentation des fonctions membres

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

Applique les préférences et les options de ligne de commande.

Définition à la ligne 167 du fichier mainWindow.py.

9.11.3.2 `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 188 du fichier mainWindow.py.

9.11.3.3 `def src.mainWindow.mainWindow.checkDisks ( self, force = False, noLoop = False )`

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

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

#### Paramètres

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

Définition à la ligne 559 du fichier mainWindow.py.

9.11.3.4 `def src.mainWindow.mainWindow.connectTableModel ( self, data )`

Connecte le modèle de table à la table.

#### Paramètres

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

Définition à la ligne 528 du fichier mainWindow.py.

9.11.3.5 `def src.mainWindow.mainWindow.copyFrom ( self )`

Lance l'action de copier depuis les clés USB.

Définition à la ligne 385 du fichier mainWindow.py.

9.11.3.6 `def src.mainWindow.mainWindow.copyTo ( self )`

Lance l'action de copier vers les clés USB.

Définition à la ligne 361 du fichier mainWindow.py.

9.11.3.7 `def src.mainWindow.mainWindow.delFiles ( self )`

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

Définition à la ligne 326 du fichier mainWindow.py.

9.11.3.8 `def src.mainWindow.mainWindow.deviceAdded ( self, s )`

fonction de rappel pour un medium ajouté

#### Paramètres

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

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

9.11.3.9 `def src.mainWindow.mainWindow.deviceRemoved ( self, s )`

fonction de rappel pour un medium retiré

#### Paramètres

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

Définition à la ligne 120 du fichier `mainWindow.py`.

9.11.3.10 `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 247 du fichier `mainWindow.py`.

9.11.3.11 `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 230 du fichier `mainWindow.py`.

9.11.3.12 `def src.mainWindow.mainWindow.editOwner ( self, idx )`

Édition du propriétaire d'une clé.

#### Paramètres

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

Définition à la ligne 268 du fichier `mainWindow.py`.

9.11.3.13 `def src.mainWindow.mainWindow.flashLCD ( self )`

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

Définition à la ligne 599 du fichier mainWindow.py.

9.11.3.14 `def src.mainWindow.mainWindow.help ( self )`

Affiche le widget d'aide.

Définition à la ligne 482 du fichier mainWindow.py.

9.11.3.15 `def src.mainWindow.mainWindow.initRedoStuff ( self )`

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

Définition à la ligne 129 du fichier mainWindow.py.

9.11.3.16 `def src.mainWindow.mainWindow.normalLCD ( self )`

remet le style par défaut pour l'afficheur LCD

Définition à la ligne 607 du fichier mainWindow.py.

9.11.3.17 `def src.mainWindow.mainWindow.preference ( self )`

lance le dialogue des préférences

Définition à la ligne 312 du fichier mainWindow.py.

9.11.3.18 `def src.mainWindow.mainWindow.redoCmd ( self )`

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

Définition à la ligne 450 du fichier mainWindow.py.

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

#### **Renvoie**

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

Définition à la ligne 592 du fichier mainWindow.py.

9.11.3.20 `def src.mainWindow.mainWindow.setTimer ( self, enabled = True )`

sets the main timer

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

**9.11.3.21** `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 147 du fichier `mainWindow.py`.

**9.11.3.22** `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 <code>QModelIndex</code>
------------	-----------------------------

Définition à la ligne 197 du fichier `mainWindow.py`.

**9.11.3.23** `def src.mainWindow.mainWindow.umount ( self )`

Démonte et détache les clés USB affichées.

Définition à la ligne 491 du fichier `mainWindow.py`.

**9.11.3.24** `def src.mainWindow.mainWindow.updateButtons ( self )`

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

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

Définition à la ligne 284 du fichier `mainWindow.py`.

### 9.11.4 Documentation des données membres

**9.11.4.1** `src.mainWindow.mainWindow.checkable`

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

**9.11.4.2** `src.mainWindow.mainWindow.checkDisksLock`

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

#### 9.11.4.3 `src.mainWindow.mainWindow.flashTimer`

Définition à la ligne 67 du fichier mainWindow.py.

#### 9.11.4.4 `src.mainWindow.mainWindow.header`

Définition à la ligne 167 du fichier mainWindow.py.

#### 9.11.4.5 `src.mainWindow.mainWindow.iconRedo`

Définition à la ligne 129 du fichier mainWindow.py.

#### 9.11.4.6 `src.mainWindow.mainWindow.iconStop`

Définition à la ligne 129 du fichier mainWindow.py.

#### 9.11.4.7 `src.mainWindow.mainWindow.listener`

Définition à la ligne 67 du fichier mainWindow.py.

#### 9.11.4.8 `src.mainWindow.mainWindow.locale`

Définition à la ligne 67 du fichier mainWindow.py.

#### 9.11.4.9 `src.mainWindow.mainWindow.manFileLocation`

Définition à la ligne 167 du fichier mainWindow.py.

#### 9.11.4.10 `src.mainWindow.mainWindow.mv`

Définition à la ligne 167 du fichier mainWindow.py.

#### 9.11.4.11 `src.mainWindow.mainWindow.oldThreads`

Définition à la ligne 67 du fichier mainWindow.py.

#### 9.11.4.12 `src.mainWindow.mainWindow.operations`

Définition à la ligne 67 du fichier mainWindow.py.

**9.11.4.13 `src.mainWindow.mainWindow.opts`**

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

**9.11.4.14 `src.mainWindow.mainWindow.proxy`**

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

**9.11.4.15 `src.mainWindow.mainWindow.redoStatusTip`**

Définition à la ligne 129 du fichier `mainWindow.py`.

**9.11.4.16 `src.mainWindow.mainWindow.redoToolTip`**

Définition à la ligne 129 du fichier `mainWindow.py`.

**9.11.4.17 `src.mainWindow.mainWindow.refreshDelay`**

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

**9.11.4.18 `src.mainWindow.mainWindow.refreshEnabled`**

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

**9.11.4.19 `src.mainWindow.mainWindow.stopStatusTip`**

Définition à la ligne 129 du fichier `mainWindow.py`.

**9.11.4.20 `src.mainWindow.mainWindow.stopToolTip`**

Définition à la ligne 129 du fichier `mainWindow.py`.

**9.11.4.21 `src.mainWindow.mainWindow.t`**

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

**9.11.4.22 `src.mainWindow.mainWindow.timer`**

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

#### 9.11.4.23 `src.mainWindow.mainWindow.tm`

Définition à la ligne 528 du fichier `mainWindow.py`.

#### 9.11.4.24 `src.mainWindow.mainWindow.ui`

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

#### 9.11.4.25 `src.mainWindow.mainWindow.visibleheader`

Définition à la ligne 528 du fichier `mainWindow.py`.

#### 9.11.4.26 `src.mainWindow.mainWindow.workdir`

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

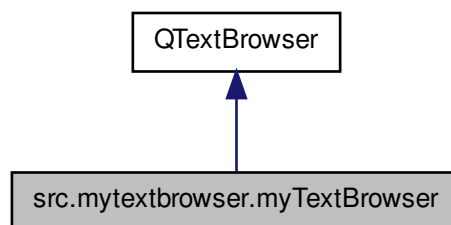
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mainWindow.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

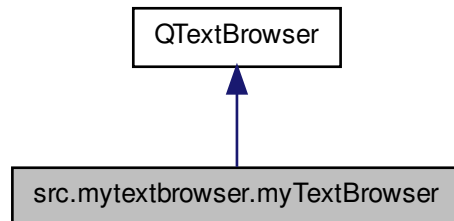
## 9.12 Référence de la classe `src.mytextbrowser.myTextBrowser`

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

Graphes d'héritage de `src.mytextbrowser.myTextBrowser` :



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*

#### 9.12.1 Description détaillée

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

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

#### 9.12.2 Documentation des fonctions membres

9.12.2.1 def `src.mytextbrowser.myTextBrowser.setHtml ( self, url )`

lien vers la méthode `setSource` originale

##### Paramètres

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

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

9.12.2.2 def `src.mytextbrowser.myTextBrowser.setSource ( self, url )`

lance Firefox en tâche de fond.

##### Paramètres

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

Définition à la ligne 40 du fichier mytextbrowser.py.

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

- [src/mytextbrowser.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

## 9.13 Référence de la classe src.notification.Notification

Une classe pour afficher des notifications à l'écran.

### Fonctions membres publiques

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

### Attributs publics

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

#### 9.13.1 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>.

Définition à la ligne 37 du fichier notification.py.

#### 9.13.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

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

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

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

### 9.13.3 Documentation des fonctions membres

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

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

### 9.13.4 Documentation des données membres

#### 9.13.4.1 `src.notification.Notification.actions`

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

#### 9.13.4.2 `src.notification.Notification.app_icon`

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

#### 9.13.4.3 `src.notification.Notification.app_name`

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

#### 9.13.4.4 `src.notification.Notification.body`

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

#### 9.13.4.5 `src.notification.Notification.expire_timeout`

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

#### 9.13.4.6 `src.notification.Notification.hints`

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

#### 9.13.4.7 `src.notification.Notification.interface`

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

#### 9.13.4.8 `src.notification.Notification.replaces_id`

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

#### 9.13.4.9 `src.notification.Notification.summary`

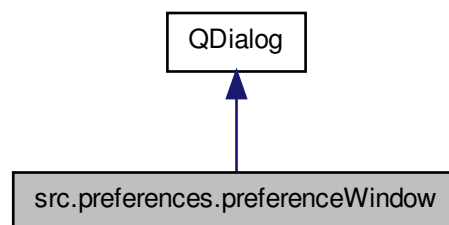
Définition à la ligne 51 du fichier `notification.py`.

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

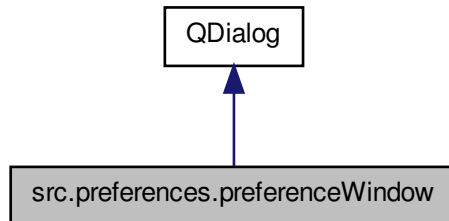
- `src/notification.py` ([Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)](#))

### 9.14 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`

#### 9.14.1 Description détaillée

Définition à la ligne 29 du fichier preferences.py.

#### 9.14.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Définition à la ligne 34 du fichier preferences.py.

#### 9.14.3 Documentation des fonctions membres

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

active ou désactive le glisseur pour modifier le délai de rafraichissement

##### Paramètres

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

Définition à la ligne 47 du fichier preferences.py.

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

Met en place les préférences dans le dialogue.

##### Paramètres

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

Définition à la ligne 80 du fichier preferences.py.

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

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

##### Paramètres

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

Définition à la ligne 55 du fichier preferences.py.

#### 9.14.3.4 `def src.preferences.preferenceWindow.values ( self )`

##### Renvoie

un dictionnaire de préférences

Définition à la ligne 65 du fichier preferences.py.

### 9.14.4 Documentation des données membres

#### 9.14.4.1 `src.preferences.preferenceWindow.ui`

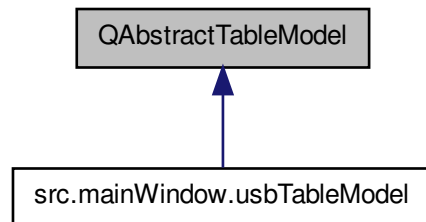
Définition à la ligne 34 du fichier preferences.py.

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

– [src/preferences.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

## 9.15 Référence de la classe QAbstractTableModel

Graphe d'héritage de QAbstractTableModel :

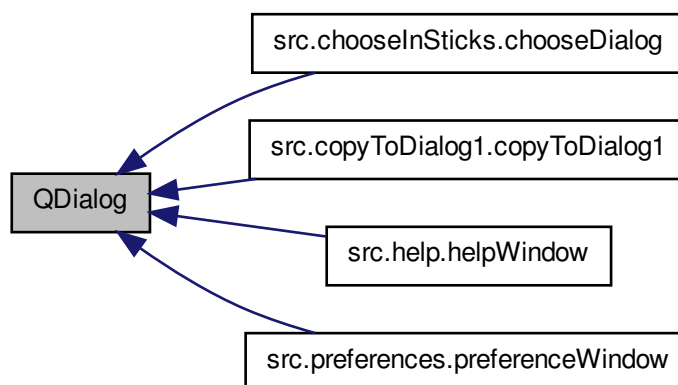


La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mainWindow.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

## 9.16 Référence de la classe QDialog

Graphe d'héritage de QDialog :

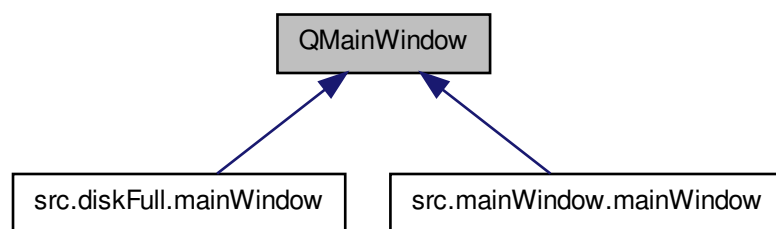


La documentation de cette classe a été générée à partir du fichier suivant :

- [src/copyToDialog1.py](#) (Concurrent Versions System (CVS) 1.12.13-MirDebian-5 (client/-server))

## 9.17 Référence de la classe QMainWindow

Graphe d'héritage de QMainWindow :

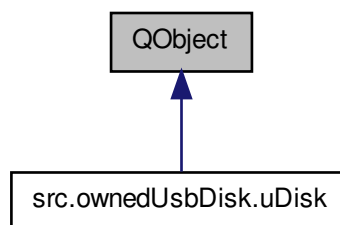


La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mainWindow.py](#) (Concurrent Versions System (CVS) 1.12.13-MirDebian-5 (client/-server))

## 9.18 Référence de la classe QObject

Graphe d'héritage de QObject :

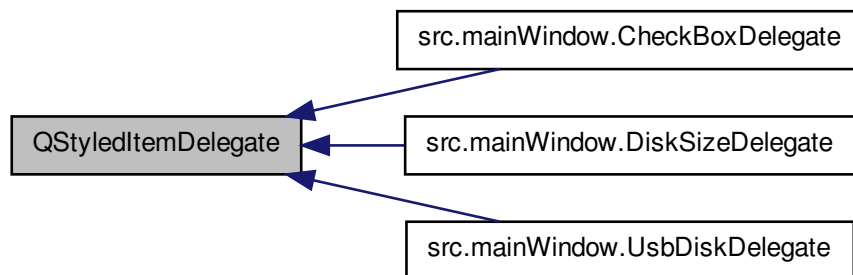


La documentation de cette classe a été générée à partir du fichier suivant :

- [src/ownedUsbDisk.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

## 9.19 Référence de la classe QStyledItemDelegate

Graphes d'héritage de QStyledItemDelegate :

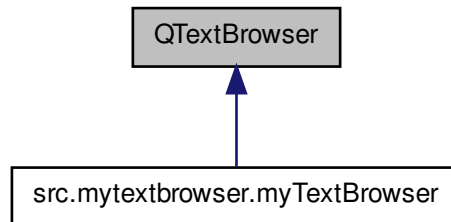


La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mainWindow.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

## 9.20 Référence de la classe QTextBrowser

Graphique d'héritage de QTextBrowser :

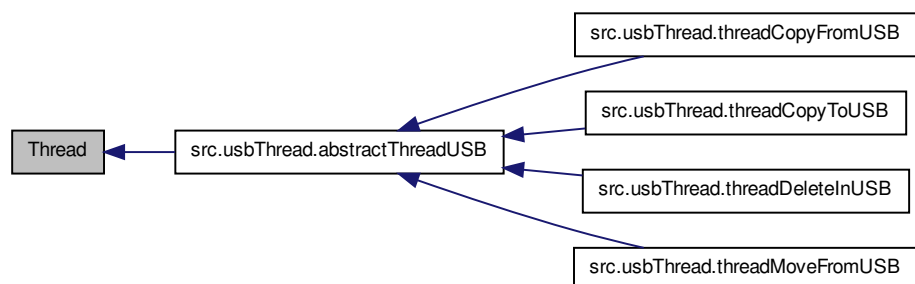


La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mytextbrowser.py](#) (Concurrent Versions System (CVS) 1.12.13-MirDebian-5 (client/-server))

## 9.21 Référence de la classe Thread

Graphique d'héritage de Thread :



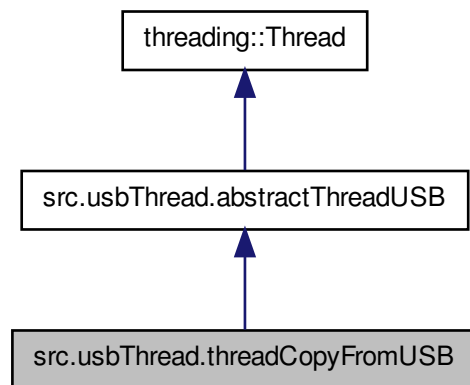
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/usbThread.py](#) (Concurrent Versions System (CVS) 1.12.13-MirDebian-5 (client/-server))

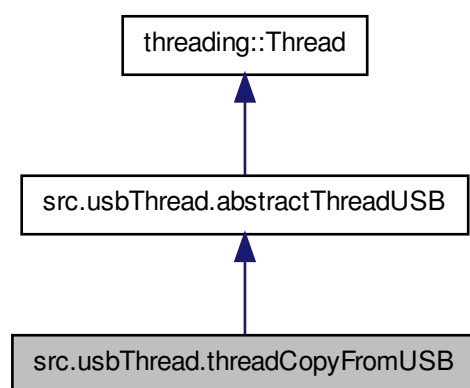
## 9.22 Référence de la classe src.usbThread.threadCopyFromUSB

Classe pour les threads copiant depuis les clés USB.

Graphe d'héritage de src.usbThread.threadCopyFromUSB :



Graphe de collaboration de src.usbThread.threadCopyFromUSB :



## Fonctions membres publiques

- def `__init__`  
*Constructeur Crée un thread pour copier une liste de fichiers depuis une clé USB vers un répertoire de disque.*
- def `todo`  
*Copie une liste de fichiers d'une clé USB sous un répertoire donné.*

## Attributs publics

- `rootPath`
- `cmd`

### 9.22.1 Description détaillée

Classe pour les threads copiant depuis les clés USB.

Définition à la ligne 284 du fichier `usbThread.py`.

### 9.22.2 Documentation des constructeurs et destructeur

9.22.2.1 `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

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

Définition à la ligne 298 du fichier `usbThread.py`.

### 9.22.3 Documentation des fonctions membres

9.22.3.1 `def src.usbThread.threadCopyFromUSB.todo( self, ud, fileList, subdir, dest, logfile )`

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

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

#### Paramètres

<code>ud</code>	l'instance <code>uDisk</code> correspondant à une partition de clé USB
-----------------	--

<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 316 du fichier `usbThread.py`.

#### 9.22.4 Documentation des données membres

##### 9.22.4.1 `src.usbThread.threadCopyFromUSB.cmd`

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

Définition à la ligne 298 du fichier `usbThread.py`.

##### 9.22.4.2 `src.usbThread.threadCopyFromUSB.rootPath`

Définition à la ligne 298 du fichier `usbThread.py`.

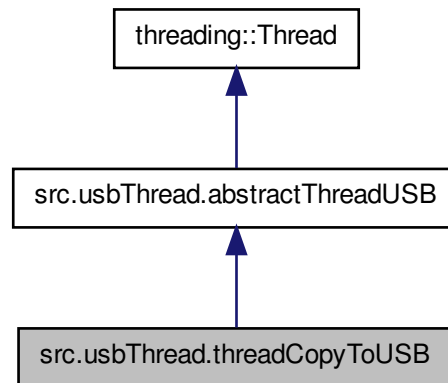
La documentation de cette classe a été générée à partir du fichier suivant :

- `src/usbThread.py` ([Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)](#))

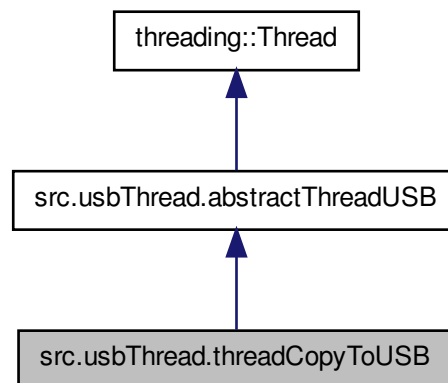
## 9.23 Référence de la classe `src.usbThread.threadCopyToUSB`

Classe pour les threads copiant vers les clés USB.

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



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



### Fonctions membres publiques

– `def __init__`

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

### Attributs publics

- `cmd`

#### 9.23.1 Description détaillée

Classe pour les threads copiant vers les clés USB.

Définition à la ligne 234 du fichier `usbThread.py`.

#### 9.23.2 Documentation des constructeurs et destructeur

9.23.2.1 `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 246 du fichier `usbThread.py`.

#### 9.23.3 Documentation des fonctions membres

9.23.3.1 `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 255 du fichier `usbThread.py`.

9.23.3.2 `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 270 du fichier `usbThread.py`.

### 9.23.4 Documentation des données membres

#### 9.23.4.1 `src.usbThread.threadCopyToUSB.cmd`

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

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

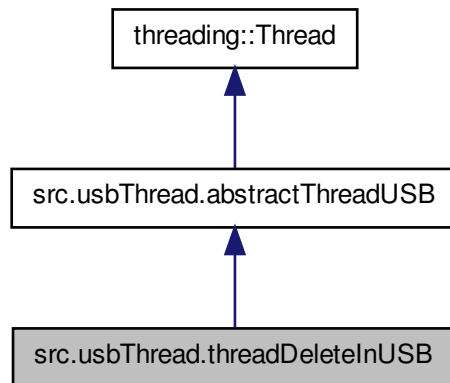
La documentation de cette classe a été générée à partir du fichier suivant :

- `src/usbThread.py` (Concurrent Versions System (CVS) 1.12.13-MirDebian-5 (client/-server))

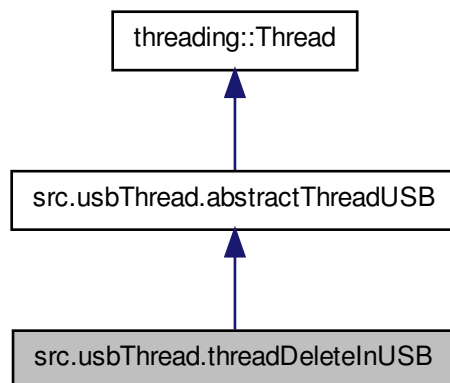
## 9.24 Référence de la classe `src.usbThread.threadDeleteInUSB`

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

Graphe d'héritage de src.usbThread.threadDeleteInUSB :



Graphe de collaboration de src.usbThread.threadDeleteInUSB :



### Fonctions membres publiques

– def [\\_\\_init\\_\\_](#)

- *Constructeur Crée un thread pour supprimer une liste de fichiers dans une clé USB.*
- `def todo`  
*Supprime une liste de fichiers dans une clé USB.*

### Attributs publics

- `cmd`

### 9.24.1 Description détaillée

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

Définition à la ligne 382 du fichier `usbThread.py`.

### 9.24.2 Documentation des constructeurs et destructeur

9.24.2.1 `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

<code>ud</code>	l'instance <code>uDisk</code> correspondant à une partition de clé USB
<code>fileList</code>	la liste des fichiers à supprimer
<code>subdir</code>	le sous-répertoire de la clé USB où faire les suppressions
<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 394 du fichier `usbThread.py`.

### 9.24.3 Documentation des fonctions membres

9.24.3.1 `def src.usbThread.threadDeleteInUSB.todo( self, ud, fileList, subdir, dest, logfile )`

Supprime une liste de fichiers dans une clé USB.

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

#### Paramètres

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

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

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

#### 9.24.4 Documentation des données membres

##### 9.24.4.1 `src.usbThread.threadDeleteInUSB.cmd`

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

Définition à la ligne 394 du fichier `usbThread.py`.

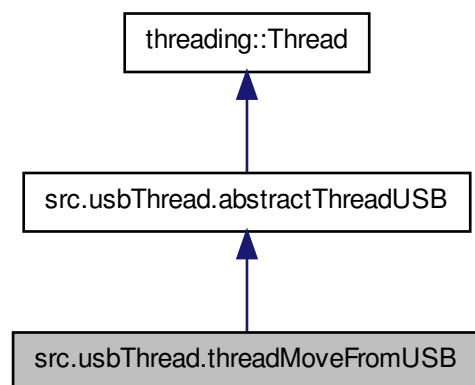
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/usbThread.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

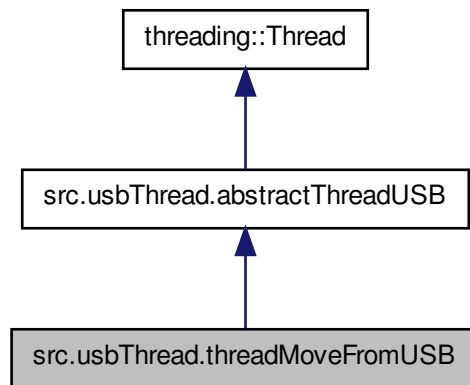
## 9.25 Référence de la classe `src.usbThread.threadMoveFromUSB`

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`
- `cmd`

#### 9.25.1 Description détaillée

Classe pour les threads déplaçant des fichiers depuis les clés USB.

Définition à la ligne 333 du fichier `usbThread.py`.

#### 9.25.2 Documentation des constructeurs et destructeur

9.25.2.1 `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 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 347 du fichier `usbThread.py`.

### 9.25.3 Documentation des fonctions membres

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

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

Définition à la ligne 366 du fichier `usbThread.py`.

### 9.25.4 Documentation des données membres

9.25.4.1 `src.usbThread.threadMoveFromUSB.cmd`

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

Définition à la ligne 347 du fichier `usbThread.py`.

9.25.4.2 `src.usbThread.threadMoveFromUSB.rootPath`

Définition à la ligne 347 du fichier `usbThread.py`.

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

- [src/usbThread.py](#) ([Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5](#) ([client/](#)[server](#)))

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

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

### Fonctions membres publiques

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

### Attributs publics

- `dico`

#### 9.26.1 Description détaillée

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

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

#### 9.26.2 Documentation des constructeurs et destructeur

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

Le constructeur met en place un dictionnaire.

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

#### 9.26.3 Documentation des fonctions membres

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

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

### 9.26.3.2 `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 72 du fichier `usbThread.py`.

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

#### Paramètres

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

Définition à la ligne 63 du fichier `usbThread.py`.

### 9.26.3.4 `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 51 du fichier `usbThread.py`.

### 9.26.3.5 `def src.usbThread.ThreadRegister.threadSet ( self )`

renvoie l'ensemble des threads actifs

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

## 9.26.4 Documentation des données membres

### 9.26.4.1 `src.usbThread.ThreadRegister.dico`

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

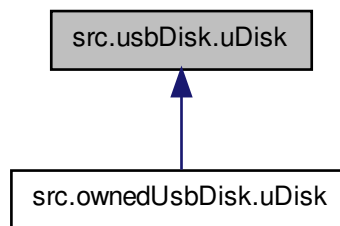
La documentation de cette classe a été générée à partir du fichier suivant :

- `src/usbThread.py` ([Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)](#))

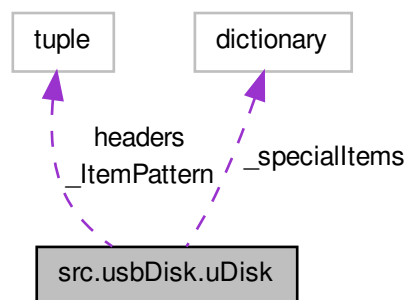
## 9.27 Référence de la classe src.usbDisk.uDisk

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

### Attributs publics statiques

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

#### 9.27.1 Description détaillée

une classe pour représenter un disque ou une partition.

les attributs publics sont :

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

Définition à la ligne 42 du fichier usbDisk.py.

### 9.27.2 Documentation des constructeurs et destructeur

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

Le constructeur.

#### Paramètres

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

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

Définition à la ligne 51 du fichier usbDisk.py.

### 9.27.3 Documentation des fonctions membres

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

Renvoie un élément de listage de données internes au disque.

#### Paramètres

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

#### Renvoie

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

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

Définition à la ligne 286 du fichier usbDisk.py.

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

Fournit une représentation imprimable.

**Renvoie**

une représentation imprimable de l'instance

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

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

renvoie un proxy vers un navigateur de propriétés

**Paramètres**

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

**Renvoie**

l'objet proxy

Définition à la ligne 119 du fichier `usbDisk.py`.

**9.27.3.4** `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 332 du fichier `usbDisk.py`.

**9.27.3.5** `def src.usbDisk.uDisk.file ( self )`

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

**Renvoie**

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

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

**9.27.3.6** `def src.usbDisk.uDisk.getFatUuid ( self )`

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

**Renvoie**

un uuid

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

9.27.3.7 `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 323 du fichier usbDisk.py.

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

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

#### Paramètres

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

#### Renvoie

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

Définition à la ligne 188 du fichier usbDisk.py.

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

Méthode statique, pour avoir des titres de colonne.

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

#### Paramètres

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

#### Renvoie

une liste de titres de colonnes

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

Définition à la ligne 104 du fichier usbDisk.py.

9.27.3.10 `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 199 du fichier usbDisk.py.

9.27.3.11 `def src.usbDisk.uDisk.isMounted ( self )`

#### Renvoie

True si le disque ou la partion est montée

Définition à la ligne 206 du fichier usbDisk.py.

9.27.3.12 `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 129 du fichier usbDisk.py.

9.27.3.13 `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 140 du fichier usbDisk.py.

9.27.3.14 `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 260 du fichier usbDisk.py.

9.27.3.15 `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 175 du fichier usbDisk.py.

9.27.3.16 `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 trans-type 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 306 du fichier usbDisk.py.

9.27.3.17 `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 156 du fichier usbDisk.py.

9.27.3.18 `def src.usbDisk.uDisk.uniqueId ( self )`

renvoie un identifiant unique.

Dans cette classe, cette fonction est synonyme de getFatUuid

#### Renvoie

un identifiant supposé unique

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

Définition à la ligne 91 du fichier usbDisk.py.

9.27.3.19 `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 270 du fichier usbDisk.py.

9.27.3.20 `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 214 du fichier `usbDisk.py`.

### 9.27.4 Documentation des données membres

9.27.4.1 `src.usbDisk.uDisk.checkable`

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

9.27.4.2 `src.usbDisk.uDisk.device`

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

9.27.4.3 `src.usbDisk.uDisk.device_prop`

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

9.27.4.4 `src.usbDisk.uDisk.fatuuid`

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

9.27.4.5 `src.usbDisk.uDisk.firstFat`

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

9.27.4.6 `tuple src.usbDisk.uDisk.headers = staticmethod(headers) [static]`

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

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

9.27.4.7 `src.usbDisk.uDisk.path`

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

9.27.4.8 `src.usbDisk.uDisk.selected`

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

#### 9.27.4.9 `src.usbDisk.uDisk.stickid`

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

#### 9.27.4.10 `src.usbDisk.uDisk.uuid`

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

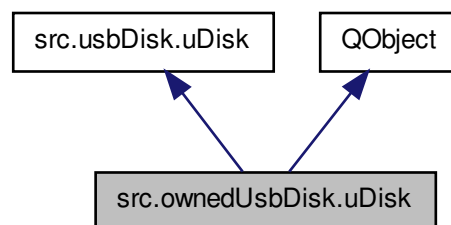
La documentation de cette classe a été générée à partir du fichier suivant :

- `src/usbDisk.py` ([Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/server\)](#))

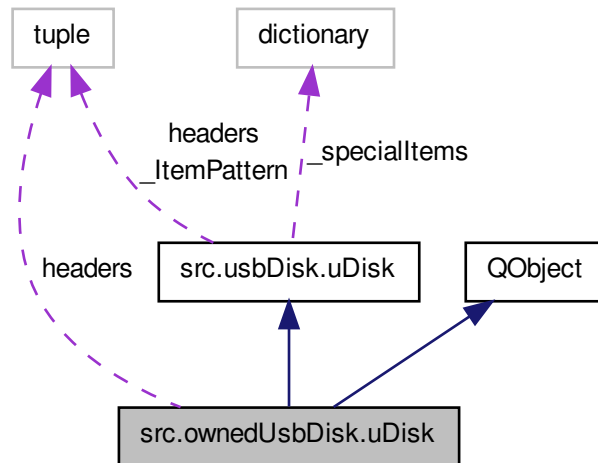
## 9.28 Référence de la classe `src.ownedUsbDisk.uDisk`

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](#))

### 9.28.1 Description détaillée

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

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

### 9.28.2 Documentation des constructeurs et destructeur

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

#### Paramètres

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

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

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

### 9.28.3 Documentation des fonctions membres

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

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

#### Paramètres

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

#### Renvoie

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

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

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

**9.28.3.2** `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 211 du fichier `ownedUsbDisk.py`.

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

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

**Paramètres**

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

**Renvoie**

une liste de titres de colonnes

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

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

**9.28.3.4** `def src.ownedUsbDisk.uDisk.ownerByDb ( self )`

renvoie un nom de propriétaire dans tous les cas.

Définition à la ligne 161 du fichier `ownedUsbDisk.py`.

**9.28.3.5** `def src.ownedUsbDisk.uDisk.readQuirks ( self )`

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

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

Définition à la ligne 119 du fichier `ownedUsbDisk.py`.

**9.28.3.6** `def src.ownedUsbDisk.uDisk.tattoo ( self )`

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

**Renvoie**

un tatouage, supposément unique.

Définition à la ligne 87 du fichier `ownedUsbDisk.py`.

**9.28.3.7** `def src.ownedUsbDisk.uDisk.uniqueId ( self )`

**Renvoie**

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

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

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

**9.28.3.8** `def src.ownedUsbDisk.uDisk.visibleDir ( self )`

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

Ce répertoire peut varier selon les vendeurs et les modèles.

Définition à la ligne 135 du fichier `ownedUsbDisk.py`.

**9.28.4 Documentation des données membres**

**9.28.4.1** `tuple src.ownedUsbDisk.uDisk.headers = staticmethod(headers)`  
`[static]`

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

Définition à la ligne 202 du fichier `ownedUsbDisk.py`.

**9.28.4.2** `src.ownedUsbDisk.uDisk.model`

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

**9.28.4.3** `src.ownedUsbDisk.uDisk.owner`

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

**9.28.4.4** `src.ownedUsbDisk.uDisk.vendor`

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

## 9.28.4.5 src.ownedUsbDisk.uDisk.visibleDirs

Définition à la ligne 66 du fichier ownedUsbDisk.py.

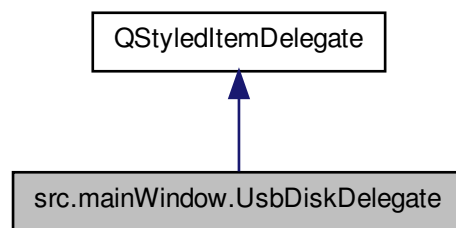
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/ownedUsbDisk.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

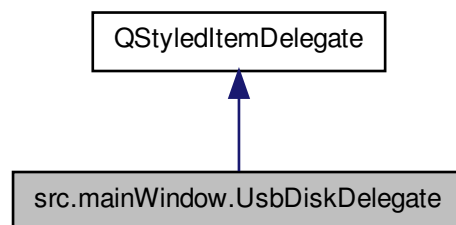
## 9.29 Référence de la classe src.mainWindow.UsbDiskDelegate

Classe pour identifier le baladeur dans le tableau.

Graphe d'héritage de src.mainWindow.UsbDiskDelegate :



Graphe de collaboration de src.mainWindow.UsbDiskDelegate :



### Fonctions membres publiques

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

### Attributs publics

- [okPixmap](#)
- [busyPixmap](#)

#### 9.29.1 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 809 du fichier mainWindow.py.

#### 9.29.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 810 du fichier mainWindow.py.

#### 9.29.3 Documentation des fonctions membres

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

Définition à la ligne 815 du fichier mainWindow.py.

#### 9.29.4 Documentation des données membres

9.29.4.1 `src.mainWindow.UsbDiskDelegate.busyPixmap`

Définition à la ligne 810 du fichier mainWindow.py.

9.29.4.2 `src.mainWindow.UsbDiskDelegate.okPixmap`

Définition à la ligne 810 du fichier mainWindow.py.

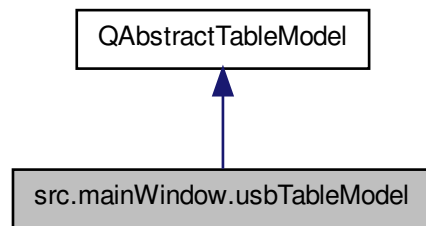
La documentation de cette classe a été générée à partir du fichier suivant :

- [src/mainWindow.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

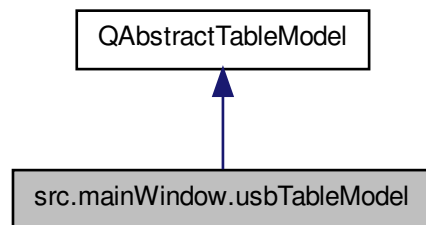
## 9.30 Référence de la classe src.mainWindow.usbTableModel

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`
- `checkable`
- `pere`

### 9.30.1 Description détaillée

Un modèle de table pour des séries de clés USB.

Définition à la ligne 614 du fichier `mainWindow.py`.

### 9.30.2 Documentation des constructeurs et destructeur

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

#### Paramètres

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

Définition à la ligne 623 du fichier `mainWindow.py`.

### 9.30.3 Documentation des fonctions membres

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

un `QModelIndex`

Définition à la ligne 700 du fichier `mainWindow.py`.

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

Définition à la ligne 718 du fichier `mainWindow.py`.

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

Définition à la ligne 747 du fichier mainWindow.py.

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

#### Paramètres

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

#### Renvoie

la partition pointée par index

Définition à la ligne 715 du fichier mainWindow.py.

9.30.3.5 `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 655 du fichier mainWindow.py.

9.30.3.6 `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 638 du fichier mainWindow.py.

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

un QModelIndex

Définition à la ligne 693 du fichier mainWindow.py.

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

Définition à la ligne 703 du fichier mainWindow.py.

**9.30.3.9** `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, Qt.DescendingOrder par défaut

Définition à la ligne 759 du fichier mainWindow.py.

**9.30.3.10** `def src.mainWindow.usbTableModel.updateOwnerColumn ( self )`

force la mise à jour de la colonne des propriétaires

Définition à la ligne 681 du fichier mainWindow.py.

## **9.30.4 Documentation des données membres**

**9.30.4.1** `src.mainWindow.usbTableModel.checkable`

Définition à la ligne 623 du fichier mainWindow.py.

**9.30.4.2** `src.mainWindow.usbTableModel.donnees`

Définition à la ligne 623 du fichier mainWindow.py.

**9.30.4.3** `src.mainWindow.usbTableModel.header`

Définition à la ligne 623 du fichier mainWindow.py.

**9.30.4.4** `src.mainWindow.usbTableModel.pere`

Définition à la ligne 623 du fichier mainWindow.py.

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

– [src/mainWindow.py \(Concurrent Versions System \(CVS\) 1.12.13-MirDebian-5 \(client/-server\)\)](#)

## Chapitre 10

# Documentation des fichiers

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

#### Paquetages

- package `src`

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

#### Classes

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

#### Paquetages

- package `src.chooseInSticks`

#### Variables

- string `src : :chooseInSticks.licenceEn`

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

#### Classes

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

### Paquetages

- package `src.copyToDialog1`

### Variables

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

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

### Paquetages

- package `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.database` = None
- `src : :db.cursor` = None

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

### Classes

- class `src.deviceListener.DeviceListener`

### Paquetages

- package `src.deviceListener`

### Variables

- dictionary `src : :deviceListener.licence = {}`

## 10.6 Référence du fichier src/diskFull.py

### Classes

- class `src.diskFull.mainWindow`

### Paquetages

- package `src.diskFull`

### Fonctions

- def `src : :diskFull.sceneWithUsage`

### Variables

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

## 10.7 Référence du fichier src/globaldef.py

### Paquetages

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

## 10.8 Référence du fichier src/help.py

### Classes

- class [src.help.helpWindow](#)

### Paquetages

- package [src.help](#)

### Variables

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

## 10.9 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

- package [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

## 10.10 Référence du fichier src/marques.py

### Paquetages

- package [src.marques](#)

## 10.11 Référence du fichier src/mytextbrowser.py

### Classes

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

### Paquetages

- package [src.mytextbrowser](#)

### Variables

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

## 10.12 Référence du fichier src/notification.py

### Classes

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

### Paquetages

- package [src.notification](#)

### Variables

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

## 10.13 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

- package [src.ownedUsbDisk](#)

### Fonctions

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

### Variables

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

## 10.14 Référence du fichier src/preferences.py

### Classes

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

### Paquetages

- package [src.preferences](#)

### Variables

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

## 10.15 Référence du fichier src/scolasync.py

### Paquetages

- package [src.scolasync](#)
- package [scolasync](#)  
*Scolasync est un programme pour gérer des transferts de fichiers entre un ordinateur et une collection de clés USB.*

### Fonctions

- def [src : :scolasync.usage](#)  
*affiche le mode d'emploi à la console*
- def [src : :scolasync.run](#)  
*Le lancement de l'application.*

### Variables

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

## 10.16 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

- package `src.usbDisk`

### Variables

- dictionary `src : :usbDisk.licence` = {}
- string `src : :usbDisk.licence_en`
- tuple `src : :usbDisk.machin` = Available()

## 10.17 Référence du fichier src/usbThread.py

### Classes

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

- package `src.usbThread`

### Variables

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

## 10.18 Référence du fichier `src/version.py`

### Paquetages

- package `src.version`

### Fonctions

- def `src : :version.major`
- def `src : :version.minor`
- def `src : :version.version`

### Variables

- dictionary `src : :version.licence` = {}