

ScolaSync

5.1

Généré par Doxygen 1.8.9.1

Jeudi 14 Mai 2015 17 :33 :52

Table des matières

1	ScolaSync	1
1.1	But de l'application	1
1.2	CAHIER DE CHARGES DE SCOLASYNC	1
1.3	Licence	1
1.4	Support	2
1.5	Architecture de ScolaSync	2
2	Index des espaces de nommage	3
2.1	Paquetages	3
3	Index hiérarchique	5
3.1	Hiérarchie des classes	5
4	Index des classes	7
4.1	Liste des classes	7
5	Index des fichiers	9
5.1	Liste des fichiers	9
6	Documentation des espaces de nommage	11
6.1	Référence de l'espace de nommage scolasync	11
6.1.1	Description détaillée	11
6.2	Référence de l'espace de nommage src	11
6.3	Référence de l'espace de nommage src.checkBoxDialog	11
6.3.1	Documentation des variables	12
6.3.1.1	licenceEn	12
6.4	Référence de l'espace de nommage src.choixEleves	12
6.4.1	Documentation des variables	12
6.4.1.1	app	12
6.4.1.2	d	13
6.4.1.3	i	13
6.4.1.4	licence	13
6.5	Référence de l'espace de nommage src.chooseInSticks	13

6.5.1	Documentation des variables	13
6.5.1.1	licenceEn	13
6.6	Référence de l'espace de nommage src.copyToDialog1	13
6.6.1	Documentation des variables	14
6.6.1.1	app	14
6.6.1.2	licenceEn	14
6.6.1.3	windows	14
6.7	Référence de l'espace de nommage src.db	14
6.7.1	Documentation des fonctions	15
6.7.1.1	checkVersion	15
6.7.1.2	hasStudent	15
6.7.1.3	knowsId	15
6.7.1.4	openDb	16
6.7.1.5	readPrefs	16
6.7.1.6	readStudent	17
6.7.1.7	setWd	17
6.7.1.8	tattooList	17
6.7.1.9	writePrefs	17
6.7.1.10	writeStudent	17
6.7.2	Documentation des variables	18
6.7.2.1	cursor	18
6.7.2.2	database	18
6.7.2.3	licence	18
6.8	Référence de l'espace de nommage src.debug	18
6.8.1	Documentation des fonctions	18
6.8.1.1	button	18
6.8.1.2	listePartitionsCochees	19
6.8.2	Documentation des variables	19
6.8.2.1	licence	19
6.8.2.2	licenceEn	19
6.8.2.3	licenceFr	19
6.9	Référence de l'espace de nommage src.diskFull	20
6.9.1	Documentation des fonctions	20
6.9.1.1	sceneWithUsage	20
6.9.2	Documentation des variables	20
6.9.2.1	licence	20
6.10	Référence de l'espace de nommage src.gestClasse	20
6.10.1	Documentation des variables	21
6.10.1.1	licence	21
6.11	Référence de l'espace de nommage src.gestclassetreeview	21

6.11.1	Documentation des variables	21
6.11.1.1	licence	21
6.12	Référence de l'espace de nommage src.globaldef	21
6.12.1	Documentation des fonctions	21
6.12.1.1	firstdir	21
6.12.2	Documentation des variables	22
6.12.2.1	licenceEn	22
6.12.2.2	logFileName	22
6.12.2.3	markFileName	22
6.12.2.4	userShareDir	22
6.13	Référence de l'espace de nommage src.help	22
6.13.1	Documentation des variables	23
6.13.1.1	licence	23
6.14	Référence de l'espace de nommage src.mainWindow	23
6.14.1	Documentation des fonctions	23
6.14.1.1	CheckBoxRect	23
6.14.1.2	registerCmd	24
6.14.2	Documentation des variables	24
6.14.2.1	activeThreads	24
6.14.2.2	lastCommand	24
6.14.2.3	licence	25
6.14.2.4	pastCommands	25
6.15	Référence de l'espace de nommage src.marques	25
6.16	Référence de l'espace de nommage src.mytextbrowser	25
6.16.1	Documentation des variables	25
6.16.1.1	licence	25
6.17	Référence de l'espace de nommage src.nameAdrive	25
6.17.1	Documentation des variables	25
6.17.1.1	licence	25
6.18	Référence de l'espace de nommage src.notification	25
6.18.1	Documentation des variables	26
6.18.1.1	licence	26
6.18.1.2	notif	26
6.19	Référence de l'espace de nommage src.ownedUsbDisk	26
6.19.1	Documentation des fonctions	26
6.19.1.1	editRecord	26
6.19.1.2	print_targets_if_modif	27
6.19.1.3	tattooInDir	27
6.19.2	Documentation des variables	27
6.19.2.1	app	27

6.19.2.2	licence	27
6.19.2.3	main	27
6.20	Référence de l'espace de nommage src.preferences	27
6.20.1	Documentation des variables	28
6.20.1.1	licence	28
6.21	Référence de l'espace de nommage src.scolasync	28
6.21.1	Documentation des fonctions	28
6.21.1.1	run	28
6.21.2	Documentation des variables	28
6.21.2.1	licence	28
6.21.2.2	licenceEn	28
6.21.2.3	licenceFr	29
6.22	Référence de l'espace de nommage src.sconet	29
6.22.1	Documentation des variables	29
6.22.1.1	licence	29
6.22.1.2	s	29
6.23	Référence de l'espace de nommage src.test3	29
6.23.1	Documentation des variables	30
6.23.1.1	files	30
6.23.1.2	module	30
6.23.1.3	moduleName	30
6.23.1.4	notsafe	30
6.23.1.5	pattern	30
6.23.1.6	python3safe	30
6.23.1.7	safe	30
6.24	Référence de l'espace de nommage src.usbDisk2	30
6.24.1	Documentation des fonctions	31
6.24.1.1	fs_size	31
6.24.1.2	inspectData	31
6.24.1.3	print_targets_if_modif	32
6.24.1.4	safePath	32
6.24.2	Documentation des variables	32
6.24.2.1	app	32
6.24.2.2	debug	33
6.24.2.3	dependences	33
6.24.2.4	licence	33
6.24.2.5	licence_en	33
6.24.2.6	main	33
6.24.2.7	no_options	33
6.24.2.8	not_interesting	33

6.25	Référence de l'espace de nommage <code>src.usbThread</code>	34
6.25.1	Documentation des fonctions	34
6.25.1.1	<code>ensureDirExists</code>	34
6.25.1.2	<code>test_copy2</code>	35
6.25.1.3	<code>test_copytree</code>	35
6.25.2	Documentation des variables	35
6.25.2.1	<code>_threadNumber</code>	35
6.25.2.2	<code>licenceEn</code>	35
6.26	Référence de l'espace de nommage <code>src.version</code>	36
6.26.1	Documentation des fonctions	36
6.26.1.1	<code>major</code>	36
6.26.1.2	<code>minor</code>	36
6.26.1.3	<code>version</code>	37
6.26.2	Documentation des variables	37
6.26.2.1	<code>licence</code>	37
7	Documentation des classes	39
7.1	Référence de la classe <code>src.gestClasse.AbstractGestClasse</code>	39
7.1.1	Description détaillée	39
7.1.2	Documentation des constructeurs et destructeur	39
7.1.2.1	<code>__init__</code>	39
7.1.3	Documentation des fonctions membres	40
7.1.3.1	<code>collectClasses</code>	40
7.1.3.2	<code>elevesDeClasse</code>	40
7.1.3.3	<code>showable_name</code>	40
7.1.3.4	<code>unique_name</code>	40
7.2	Référence de la classe <code>src.usbThread.abstractThreadUSB</code>	41
7.2.1	Description détaillée	42
7.2.2	Documentation des constructeurs et destructeur	42
7.2.2.1	<code>__init__</code>	42
7.2.3	Documentation des fonctions membres	42
7.2.3.1	<code>__str__</code>	42
7.2.3.2	<code>copytree</code>	43
7.2.3.3	<code>run</code>	43
7.2.3.4	<code>threadType</code>	44
7.2.3.5	<code>todo</code>	44
7.2.3.6	<code>writeToLog</code>	45
7.2.4	Documentation des données membres	46
7.2.4.1	<code>dest</code>	46
7.2.4.2	<code>fileList</code>	46

7.2.4.3	logfile	46
7.2.4.4	parent	46
7.2.4.5	subdir	46
7.2.4.6	ud	46
7.3	Référence de la classe <code>src.ownedUsbDisk.Available</code>	47
7.3.1	Description détaillée	48
7.3.2	Documentation des constructeurs et destructeur	48
7.3.2.1	__init__	48
7.3.3	Documentation des fonctions membres	48
7.3.3.1	finishInit	48
7.3.4	Documentation des données membres	48
7.3.4.1	ownerDialog	48
7.4	Référence de la classe <code>src.usbDisk2.Available</code>	49
7.4.1	Description détaillée	50
7.4.2	Documentation des constructeurs et destructeur	50
7.4.2.1	__init__	50
7.4.3	Documentation des fonctions membres	50
7.4.3.1	__getitem__	50
7.4.3.2	__len__	51
7.4.3.3	__str__	51
7.4.3.4	__trunc__	51
7.4.3.5	compare	51
7.4.3.6	contains	52
7.4.3.7	disks	52
7.4.3.8	disks_ud	52
7.4.3.9	finishInit	53
7.4.3.10	getFirstFats	53
7.4.3.11	hasDev	53
7.4.3.12	mountFirstFats	54
7.4.3.13	parts	54
7.4.3.14	parts_ud	54
7.4.3.15	summary	55
7.4.4	Documentation des données membres	55
7.4.4.1	access	55
7.4.4.2	firstFats	55
7.5	Référence de la classe <code>src.mainWindow.CheckBoxDelegate</code>	56
7.5.1	Description détaillée	56
7.5.2	Documentation des constructeurs et destructeur	56
7.5.2.1	__init__	56
7.5.3	Documentation des fonctions membres	56

7.5.3.1	editorEvent	57
7.5.3.2	paint	57
7.6	Référence de la classe src.checkBoxDialog.CheckBoxDialog	57
7.6.1	Description détaillée	58
7.6.2	Documentation des constructeurs et destructeur	58
7.6.2.1	__init__	58
7.6.3	Documentation des fonctions membres	58
7.6.3.1	all	59
7.6.3.2	esc	59
7.6.3.3	none	59
7.6.3.4	toggle	59
7.6.4	Documentation des données membres	59
7.6.4.1	mainWindow	59
7.6.4.2	ui	59
7.7	Référence de la classe src.choixEleves.choixElevesDialog	59
7.7.1	Description détaillée	61
7.7.2	Documentation des constructeurs et destructeur	61
7.7.2.1	__init__	61
7.7.3	Documentation des fonctions membres	61
7.7.3.1	addToList	61
7.7.3.2	checkNum	62
7.7.3.3	coche	63
7.7.3.4	connecteGestionnaire	63
7.7.3.5	decoche	63
7.7.3.6	delInList	63
7.7.3.7	escape	64
7.7.3.8	fichierEleves	64
7.7.3.9	itemStrings	64
7.7.3.10	listeChoix	64
7.7.3.11	listeUnique_Names	65
7.7.3.12	pop	65
7.7.3.13	replie	66
7.7.3.14	takeItem	66
7.7.3.15	updateParentIcon	66
7.7.3.16	valid	67
7.7.4	Documentation des données membres	67
7.7.4.1	gestionnaire	67
7.7.4.2	ok	67
7.7.4.3	prefs	67
7.7.4.4	ui	67

7.8	Référence de la classe <code>src.chooseInSticks.chooseDialog</code>	68
7.8.1	Description détaillée	69
7.8.2	Documentation des constructeurs et destructeur	69
7.8.2.1	<code>__init__</code>	69
7.8.3	Documentation des fonctions membres	69
7.8.3.1	<code>activate</code>	69
7.8.3.2	<code>append</code>	69
7.8.3.3	<code>baseDir</code>	70
7.8.3.4	<code>changeWd</code>	70
7.8.3.5	<code>checkValues</code>	71
7.8.3.6	<code>checkWorkDirs</code>	71
7.8.3.7	<code>choose</code>	71
7.8.3.8	<code>choose_dir</code>	72
7.8.3.9	<code>listStorages</code>	72
7.8.3.10	<code>minus</code>	73
7.8.3.11	<code>pathList</code>	73
7.8.3.12	<code>plus</code>	73
7.8.3.13	<code>selectedDiskMountPoint</code>	74
7.8.3.14	<code>selectedDiskOwner</code>	74
7.8.4	Documentation des données membres	74
7.8.4.1	<code>mainWindow</code>	74
7.8.4.2	<code>ok</code>	74
7.8.4.3	<code>okButton</code>	75
7.8.4.4	<code>ownedUsbDictionary</code>	75
7.9	Référence de la classe <code>src.copyToDialog1.copyToDialog1</code>	75
7.9.1	Description détaillée	76
7.9.2	Documentation des fonctions membres	76
7.9.2.1	<code>cancel</code>	76
7.9.2.2	<code>cd</code>	76
7.9.2.3	<code>changeWd</code>	77
7.9.2.4	<code>cont</code>	77
7.9.2.5	<code>displaySize</code>	77
7.9.2.6	<code>remove</code>	78
7.9.2.7	<code>select</code>	78
7.9.2.8	<code>selectedList</code>	78
7.9.2.9	<code>setFromListeDir</code>	79
7.9.2.10	<code>setupFromListe</code>	79
7.9.2.11	<code>setupToListe</code>	79
7.9.3	Documentation des données membres	80
7.9.3.1	<code>mainWindow</code>	80

7.9.3.2	ok	80
7.10	Référence de la classe src.mainWindow.DiskSizeDelegate	80
7.10.1	Description détaillée	81
7.10.2	Documentation des constructeurs et destructeur	81
7.10.2.1	__init__	81
7.10.3	Documentation des fonctions membres	81
7.10.3.1	paint	81
7.10.3.2	val2txt	81
7.11	Référence de la classe src.gestclassetreeview.gestClasseTreeView	82
7.11.1	Description détaillée	82
7.11.2	Documentation des constructeurs et destructeur	83
7.11.2.1	__init__	83
7.11.3	Documentation des fonctions membres	83
7.11.3.1	allItems	83
7.11.3.2	checkedItems	83
7.11.3.3	connecteGestionnaire	83
7.11.3.4	expandedItems	83
7.11.4	Documentation des données membres	83
7.11.4.1	gest	83
7.11.4.2	root	84
7.12	Référence de la classe src.help.helpWindow	84
7.12.1	Description détaillée	85
7.12.2	Documentation des constructeurs et destructeur	85
7.12.2.1	__init__	85
7.12.3	Documentation des fonctions membres	85
7.12.3.1	loadBrowsers	85
7.12.4	Documentation des données membres	85
7.12.4.1	ui	85
7.13	Référence de la classe src.diskFull.mainWindow	85
7.13.1	Description détaillée	86
7.13.2	Documentation des constructeurs et destructeur	86
7.13.2.1	__init__	86
7.13.3	Documentation des données membres	86
7.13.3.1	total	86
7.13.3.2	ui	87
7.13.3.3	used	87
7.13.3.4	v	87
7.14	Référence de la classe src.mainWindow.mainWindow	87
7.14.1	Description détaillée	90
7.14.2	Documentation des constructeurs et destructeur	90

7.14.2.1	<code>__init__</code>	90
7.14.3	Documentation des fonctions membres	90
7.14.3.1	<code>applyPreferences</code>	90
7.14.3.2	<code>cbAdded</code>	90
7.14.3.3	<code>cbRemoved</code>	91
7.14.3.4	<code>changeWd</code>	91
7.14.3.5	<code>checkAll</code>	91
7.14.3.6	<code>checkModify</code>	91
7.14.3.7	<code>checkNone</code>	92
7.14.3.8	<code>checkToggle</code>	92
7.14.3.9	<code>connectTableModel</code>	92
7.14.3.10	<code>copyFrom</code>	93
7.14.3.11	<code>copyTo</code>	93
7.14.3.12	<code>delFiles</code>	93
7.14.3.13	<code>deviceAdded</code>	94
7.14.3.14	<code>deviceRemoved</code>	94
7.14.3.15	<code>diskFromOwner</code>	95
7.14.3.16	<code>diskSizeData</code>	96
7.14.3.17	<code>editOwner</code>	96
7.14.3.18	<code>findAllDisks</code>	97
7.14.3.19	<code>help</code>	98
7.14.3.20	<code>initRedoStuff</code>	98
7.14.3.21	<code>manageCheckBoxes</code>	98
7.14.3.22	<code>namesCmd</code>	98
7.14.3.23	<code>namingADrive</code>	98
7.14.3.24	<code>popCmd</code>	99
7.14.3.25	<code>preference</code>	99
7.14.3.26	<code>pushCmd</code>	99
7.14.3.27	<code>redoCmd</code>	100
7.14.3.28	<code>sameDiskData</code>	100
7.14.3.29	<code>setAvailableNames</code>	100
7.14.3.30	<code>setThemedIcon</code>	100
7.14.3.31	<code>tableClicked</code>	101
7.14.3.32	<code>umount</code>	102
7.14.3.33	<code>updateButtons</code>	102
7.14.4	Documentation des données membres	102
7.14.4.1	<code>availableNames</code>	103
7.14.4.2	<code>checkAllSignal</code>	103
7.14.4.3	<code>checkNoneSignal</code>	103
7.14.4.4	<code>checkToggleSignal</code>	103

7.14.4.5	copyfromIcon	103
7.14.4.6	header	103
7.14.4.7	iconRedo	103
7.14.4.8	iconStop	103
7.14.4.9	locale	103
7.14.4.10	manFileLocation	103
7.14.4.11	movefromIcon	103
7.14.4.12	mv	103
7.14.4.13	namesDialog	104
7.14.4.14	namesEmptyIcon	104
7.14.4.15	namesEmptyTip	104
7.14.4.16	namesFullIcon	104
7.14.4.17	namesFullTip	104
7.14.4.18	oldThreads	104
7.14.4.19	operations	104
7.14.4.20	popCmdSignal	104
7.14.4.21	proxy	104
7.14.4.22	pushCmdSignal	104
7.14.4.23	recentConnect	104
7.14.4.24	recentDisconnect	104
7.14.4.25	redoStatusTip	105
7.14.4.26	redoToolTip	105
7.14.4.27	schoolFile	105
7.14.4.28	shouldNameDrive	105
7.14.4.29	stopStatusTip	105
7.14.4.30	stopToolTip	105
7.14.4.31	t	105
7.14.4.32	tm	105
7.14.4.33	ui	105
7.14.4.34	visibleheader	105
7.14.4.35	workdir	105
7.15	Référence de la classe src.ownedUsbDisk.MainWindow	106
7.15.1	Description détaillée	106
7.15.2	Documentation des constructeurs et destructeur	106
7.15.2.1	__init__	106
7.16	Référence de la classe src.usbDisk2.MainWindow	107
7.16.1	Description détaillée	107
7.16.2	Documentation des constructeurs et destructeur	107
7.16.2.1	__init__	107
7.17	Référence de la classe src.mytextbrowser.myTextBrowser	108

7.17.1	Description détaillée	108
7.17.2	Documentation des fonctions membres	108
7.17.2.1	setHtml	109
7.17.2.2	setSource	110
7.18	Référence de la classe src.nameAdrive.nameAdriveDialog	110
7.18.1	Description détaillée	111
7.18.2	Documentation des constructeurs et destructeur	111
7.18.2.1	__init__	111
7.18.3	Documentation des fonctions membres	111
7.18.3.1	esc	111
7.18.3.2	makeSelection	111
7.18.3.3	ok	112
7.18.3.4	selectionChanged	112
7.18.4	Documentation des données membres	112
7.18.4.1	nameList	112
7.18.4.2	numPattern	112
7.18.4.3	oldName	112
7.18.4.4	tattoo	112
7.18.4.5	ui	112
7.19	Référence de la classe src.notification.Notification	112
7.19.1	Description détaillée	113
7.19.2	Documentation des constructeurs et destructeur	113
7.19.2.1	__init__	113
7.19.3	Documentation des fonctions membres	113
7.19.3.1	notify	113
7.19.4	Documentation des données membres	113
7.19.4.1	actions	113
7.19.4.2	app_icon	113
7.19.4.3	app_name	113
7.19.4.4	body	114
7.19.4.5	expire_timeout	114
7.19.4.6	hints	114
7.19.4.7	interface	114
7.19.4.8	replaces_id	114
7.19.4.9	summary	114
7.20	Référence de la classe src.preferences.preferenceWindow	114
7.20.1	Description détaillée	115
7.20.2	Documentation des constructeurs et destructeur	115
7.20.2.1	__init__	115
7.20.3	Documentation des fonctions membres	115

7.20.3.1	enableDelay	115
7.20.3.2	setValues	116
7.20.3.3	updateRefreshLabel	116
7.20.3.4	values	116
7.20.4	Documentation des données membres	116
7.20.4.1	ui	116
7.21	Référence de la classe QAbstractTableModel	117
7.22	Référence de la classe QDialog	118
7.23	Référence de la classe QMainWindow	119
7.24	Référence de la classe QObject	119
7.25	Référence de la classe QStyledItemDelegate	120
7.26	Référence de la classe QTextBrowser	120
7.27	Référence de la classe QTreeView	121
7.28	Référence de la classe src.gestClasse.Sconet	121
7.28.1	Description détaillée	122
7.28.2	Documentation des constructeurs et destructeur	122
7.28.2.1	__init__	122
7.28.3	Documentation des fonctions membres	123
7.28.3.1	__str__	123
7.28.3.2	collectClasses	123
7.28.3.3	collectNullTexts	123
7.28.3.4	collectOneClass	123
7.28.3.5	elementsWalk	123
7.28.3.6	eleveParID	124
7.28.3.7	elevesDeClasse	124
7.28.3.8	makeCompact	124
7.28.3.9	showable_name	124
7.28.3.10	unIDEleveDeClasse	124
7.28.3.11	unique_name	125
7.28.4	Documentation des données membres	125
7.28.4.1	classes	125
7.28.4.2	currentClassName	125
7.28.4.3	currentID	125
7.28.4.4	currentResult	125
7.28.4.5	donnees	125
7.28.4.6	nullTexts	125
7.29	Référence de la classe src.sconet.Sconet	125
7.29.1	Description détaillée	126
7.29.2	Documentation des constructeurs et destructeur	126
7.29.2.1	__init__	126

7.29.3	Documentation des fonctions membres	126
7.29.3.1	__str__	126
7.29.3.2	collectClasses	126
7.29.3.3	collectNullTexts	126
7.29.3.4	collectOneClass	126
7.29.3.5	elementsWalk	127
7.29.3.6	makeCompact	127
7.29.4	Documentation des données membres	127
7.29.4.1	classes	127
7.29.4.2	donnees	127
7.29.4.3	nullTexts	128
7.30	Référence de la classe src.usbThread.threadCopyFromUSB	128
7.30.1	Description détaillée	129
7.30.2	Documentation des constructeurs et destructeur	129
7.30.2.1	__init__	129
7.30.3	Documentation des fonctions membres	129
7.30.3.1	todo	129
7.30.4	Documentation des données membres	130
7.30.4.1	rootPath	130
7.31	Référence de la classe src.usbThread.threadCopyToUSB	130
7.31.1	Description détaillée	131
7.31.2	Documentation des constructeurs et destructeur	131
7.31.2.1	__init__	131
7.31.3	Documentation des fonctions membres	131
7.31.3.1	threadType	132
7.31.3.2	todo	132
7.32	Référence de la classe src.usbThread.threadDeleteInUSB	132
7.32.1	Description détaillée	133
7.32.2	Documentation des constructeurs et destructeur	134
7.32.2.1	__init__	134
7.32.3	Documentation des fonctions membres	134
7.32.3.1	todo	134
7.33	Référence de la classe src.usbThread.threadMoveFromUSB	134
7.33.1	Description détaillée	136
7.33.2	Documentation des constructeurs et destructeur	136
7.33.2.1	__init__	136
7.33.3	Documentation des fonctions membres	136
7.33.3.1	todo	136
7.33.4	Documentation des données membres	137
7.33.4.1	rootPath	137

7.34	Référence de la classe <code>src.usbThread.ThreadRegister</code>	137
7.34.1	Description détaillée	137
7.34.2	Documentation des constructeurs et destructeur	138
7.34.2.1	<code>__init__</code>	138
7.34.3	Documentation des fonctions membres	138
7.34.3.1	<code>__str__</code>	138
7.34.3.2	<code>busy</code>	138
7.34.3.3	<code>pop</code>	138
7.34.3.4	<code>push</code>	138
7.34.3.5	<code>threadSet</code>	138
7.34.4	Documentation des données membres	138
7.34.4.1	<code>dico</code>	138
7.35	Référence de la classe <code>src.ownedUsbDisk.uDisk2</code>	139
7.35.1	Description détaillée	140
7.35.2	Documentation des constructeurs et destructeur	140
7.35.2.1	<code>__init__</code>	140
7.35.3	Documentation des fonctions membres	141
7.35.3.1	<code>__getitem__</code>	141
7.35.3.2	<code>ensureOwner</code>	141
7.35.3.3	<code>getFat</code>	142
7.35.3.4	<code>getOwner</code>	142
7.35.3.5	<code>headers</code>	143
7.35.3.6	<code>ownerByDb</code>	143
7.35.3.7	<code>randomOwner</code>	143
7.35.3.8	<code>readQuirks</code>	144
7.35.3.9	<code>tattoo</code>	144
7.35.3.10	<code>uniqueId</code>	145
7.35.3.11	<code>valuableProperties</code>	145
7.35.3.12	<code>visibleDir</code>	145
7.35.4	Documentation des données membres	145
7.35.4.1	<code>headers</code>	145
7.35.4.2	<code>owner</code>	145
7.35.4.3	<code>visibleDirs</code>	146
7.36	Référence de la classe <code>src.usbDisk2.uDisk2</code>	146
7.36.1	Description détaillée	147
7.36.2	Documentation des constructeurs et destructeur	147
7.36.2.1	<code>__init__</code>	147
7.36.3	Documentation des fonctions membres	148
7.36.3.1	<code>__getitem__</code>	148
7.36.3.2	<code>__str__</code>	148

7.36.3.3	<code>ensureMounted</code>	149
7.36.3.4	<code>headers</code>	149
7.36.3.5	<code>isDosFat</code>	149
7.36.3.6	<code>isMounted</code>	150
7.36.3.7	<code>mountPoint</code>	150
7.36.3.8	<code>title</code>	150
7.36.3.9	<code>uniqueId</code>	150
7.36.3.10	<code>unNumberProp</code>	150
7.36.3.11	<code>valuableProperties</code>	151
7.36.4	Documentation des données membres	151
7.36.4.1	<code>capacity</code>	151
7.36.4.2	<code>devStuff</code>	151
7.36.4.3	<code>firstFat</code>	152
7.36.4.4	<code>free</code>	152
7.36.4.5	<code>fstype</code>	152
7.36.4.6	<code>headers</code>	152
7.36.4.7	<code>isUsb</code>	152
7.36.4.8	<code>model</code>	152
7.36.4.9	<code>mp</code>	152
7.36.4.10	<code>parent</code>	152
7.36.4.11	<code>path</code>	152
7.36.4.12	<code>rlock</code>	152
7.36.4.13	<code>selected</code>	152
7.36.4.14	<code>stickid</code>	152
7.36.4.15	<code>uuid</code>	153
7.36.4.16	<code>vendor</code>	153
7.37	Référence de la classe <code>src.usbDisk2.UDisksBackend</code>	153
7.37.1	Description détaillée	154
7.37.2	Documentation des constructeurs et destructeur	154
7.37.2.1	<code>__init__</code>	154
7.37.3	Documentation des fonctions membres	154
7.37.3.1	<code>addHook</code>	154
7.37.3.2	<code>detect_devices</code>	154
7.37.3.3	<code>objIsUsb</code>	155
7.37.3.4	<code>retry_mount</code>	156
7.37.4	Documentation des données membres	156
7.37.4.1	<code>bus</code>	156
7.37.4.2	<code>cbHooks</code>	156
7.37.4.3	<code>diskClass</code>	156
7.37.4.4	<code>install_thread</code>	157

7.37.4.5	logger	157
7.37.4.6	manager	157
7.37.4.7	modified	157
7.37.4.8	targets	157
7.37.4.9	udisks	157
7.38	Référence de la classe <code>src.mainWindow.UsbDiskDelegate</code>	157
7.38.1	Description détaillée	158
7.38.2	Documentation des constructeurs et destructeur	158
7.38.2.1	<code>__init__</code>	158
7.38.3	Documentation des fonctions membres	158
7.38.3.1	<code>paint</code>	158
7.38.4	Documentation des données membres	158
7.38.4.1	<code>busyPixmap</code>	158
7.38.4.2	<code>okPixmap</code>	159
7.39	Référence de la classe <code>src.mainWindow.usbTableModel</code>	159
7.39.1	Description détaillée	160
7.39.2	Documentation des constructeurs et destructeur	160
7.39.2.1	<code>__init__</code>	160
7.39.3	Documentation des fonctions membres	160
7.39.3.1	<code>columnCount</code>	160
7.39.3.2	<code>data</code>	160
7.39.3.3	<code>headerData</code>	160
7.39.3.4	<code>partition</code>	160
7.39.3.5	<code>rowCount</code>	161
7.39.3.6	<code>setData</code>	161
7.39.3.7	<code>sort</code>	161
7.39.3.8	<code>updateOwnerColumn</code>	161
7.39.4	Documentation des données membres	161
7.39.4.1	<code>donnees</code>	161
7.39.4.2	<code>header</code>	161
7.39.4.3	<code>pere</code>	161
8	Documentation des fichiers	163
8.1	Référence du fichier <code>src/__init__.py</code>	163
8.2	Référence du fichier <code>src.checkBoxDialog.py</code>	163
8.3	Référence du fichier <code>src.choixEleves.py</code>	163
8.4	Référence du fichier <code>src.chooseInSticks.py</code>	164
8.5	Référence du fichier <code>src.copyToDialog1.py</code>	164
8.6	Référence du fichier <code>src/db.py</code>	164
8.7	Référence du fichier <code>src/debug.py</code>	165

8.8	Référence du fichier src/diskFull.py	165
8.9	Référence du fichier src/gestClasse.py	166
8.10	Référence du fichier src/gestclassetreeview.py	166
8.11	Référence du fichier src/globaldef.py	166
8.12	Référence du fichier src/help.py	167
8.13	Référence du fichier src/mainWindow.py	167
8.14	Référence du fichier src/marques.py	167
8.15	Référence du fichier src/mytextbrowser.py	167
8.16	Référence du fichier src/nameAdrive.py	168
8.17	Référence du fichier src/notification.py	168
8.18	Référence du fichier src/ownedUsbDisk.py	168
8.19	Référence du fichier src/preferences.py	169
8.20	Référence du fichier src/scolasync.py	169
8.21	Référence du fichier src/sconet.py	170
8.22	Référence du fichier src/test3.py	170
8.23	Référence du fichier src/usbDisk2.py	170
8.24	Référence du fichier src/usbThread.py	171
8.25	Référence du fichier src/version.py	171

Chapitre 1

ScolaSync

1.1 But de l'application

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

1.2 CAHIER DE CHARGES DE SCOLASYNC

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

1.3 Licence

ScolaSync version 4.0 :

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

Copyright © 2010-2012 Georges Khaznadar georgesk@offset.org

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 ou plus
- la bibliothèque udisks pour interroger facilement le noyau sur le statut des disques, et pour réaliser certaines actions sur les disques et clés USB
- l'utilisation de threads pour mener en parallèle les actions qui concernent simultanément plusieurs clés USB

Chapitre 2

Index des espaces de nommage

2.1 Paquetages

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

scolasync

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

src	11
src.checkBoxDialog	11
src.choixEleves	12
src.chooseInSticks	13
src.copyToDialog1	13
src.db	14
src.debug	18
src.diskFull	20
src.gestClasse	20
src.gestclassetreeview	21
src.globaldef	21
src.help	22
src.mainWindow	23
src.marques	25
src.mytextbrowser	25
src.nameAdrive	25
src.notification	25
src.ownedUsbDisk	26
src.preferences	27
src.scolasync	28
src.sconet	29
src.test3	29
src.usbDisk2	30
src.usbThread	34
src.version	36

Chapitre 3

Index hiérarchique

3.1 Hiérarchie des classes

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

src.gestClasse.AbstractGestClasse	39
src.gestClasse.Sconet	121
src.notification.Notification	112
QAbstractTableModel	117
src.mainWindow.usbTableModel	159
QDialog	118
src.checkBoxDialog.CheckBoxDialog	57
src.choixEleves.choixElevesDialog	59
src.chooseInSticks.chooseDialog	68
src.copyToDialog1.copyToDialog1	75
src.help.helpWindow	84
src.nameAdrive.nameAdriveDialog	110
src.preferences.preferenceWindow	114
QMainWindow	119
src.diskFull.mainWindow	85
src.mainWindow.mainWindow	87
src.ownedUsbDisk.MainWindow	106
src.usbDisk2.MainWindow	107
QObject	119
src.ownedUsbDisk.uDisk2	139
QStyledItemDelegate	120
src.mainWindow.CheckBoxDelegate	56
src.mainWindow.DiskSizeDelegate	80
src.mainWindow.UsbDiskDelegate	157
QTextBrowser	120
src.mytextbrowser.myTextBrowser	108
QTreeView	121
src.gestclassetreeview.gestClasseTreeView	82
src.sconet.Sconet	125
Thread	
src.usbThread.abstractThreadUSB	41
src.usbThread.threadCopyFromUSB	128
src.usbThread.threadCopyToUSB	130
src.usbThread.threadDeletelnUSB	132
src.usbThread.threadMoveFromUSB	134

src.usbThread.ThreadRegister	137
src.usbDisk2.uDisk2	146
src.ownedUsbDisk.uDisk2	139
src.usbDisk2.UDisksBackend	153
src.usbDisk2.Available	49
src.ownedUsbDisk.Available	47

Chapitre 4

Index des classes

4.1 Liste des classes

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

src.gestClasse.AbstractGestClasse	39
src.usbThread.abstractThreadUSB	
Une classe abstraite, qui sert de creuset pour les classe servant aux copies et aux effacements	41
src.ownedUsbDisk.Available	
Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires	47
src.usbDisk2.Available	
Une classe pour représenter la collection des disques USB connectés	49
src.mainWindow.CheckBoxDelegate	56
src.checkBoxDialog.CheckBoxDialog	
Un dialogue pour gérer les cases à cocher de l'application	57
src.choixEleves.choixElevesDialog	
Implémente un dialogue permettant de choisir des élèves les propriétés importantes sont self.ok, vrai si on doit prendre en compte la liste sélectionnée, et le contenu de la liste des sélectionnés, dont on peut récupérer les élèves un par un à l'aide de self.pop()	59
src.chooseInSticks.chooseDialog	
Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB	68
src.copyToDialog1.copyToDialog1	
Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB	75
src.mainWindow.DiskSizeDelegate	
Classe pour figurer la taille de la mémoire du baladeur	80
src.gestclassetreeview.gestClasseTreeView	82
src.help.helpWindow	84
src.diskFull.mainWindow	85
src.mainWindow.mainWindow	
Defines the main window of the application	87
src.ownedUsbDisk.MainWindow	106
src.usbDisk2.MainWindow	107
src.mytextbrowser.myTextBrowser	
Une classe qui ouvre Firefox quand on clique sur un lien externe	108
src.nameAdrive.nameAdriveDialog	
Un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles	110
src.notification.Notification	
Une classe pour afficher des notifications à l'écran	112
src.preferences.preferenceWindow	114
QAbstractTableModel	117
QDialog	118
QMainWindow	119
QObject	119

QStyledItemDelegate	120
QTextBrowser	120
QTreeView	121
src.gestClasse.Sconet	
Une classe pour travailler avec des données Sconet	121
src.sconet.Sconet	
Une classe pour travailler avec des données Sconet	125
src.usbThread.threadCopyFromUSB	
Classe pour les threads copiant depuis les clés USB	128
src.usbThread.threadCopyToUSB	
Classe pour les threads copiant vers les clés USB	130
src.usbThread.threadDeleteInUSB	
Classe pour les threads effaçant des sous-arbres dans les clés USB	132
src.usbThread.threadMoveFromUSB	
Classe pour les threads déplaçant des fichiers depuis les clés USB	134
src.usbThread.ThreadRegister	
Une classe pour tenir un registre des threads concernant les baladeurs	137
src.ownedUsbDisk.uDisk2	
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	139
src.usbDisk2.uDisk2	
Une classe pour représenter un disque ou une partition	146
src.usbDisk2.UDisksBackend	
Cette classe a été inspirée par le projet USBcreator	153
src.mainWindow.UsbDiskDelegate	
Classe pour identifier le baladeur dans le tableau	157
src.mainWindow.usbTableModel	
Un modèle de table pour des séries de clés USB	159

Chapitre 5

Index des fichiers

5.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/___init___py	163
src/checkboxDialog.py	163
src/choixEleves.py	163
src/chooseInSticks.py	164
src/copyToDialog1.py	164
src/db.py	164
src/debug.py	165
src/diskFull.py	165
src/gestClasse.py	166
src/gestclassetreeview.py	166
src/globaldef.py	166
src/help.py	167
src/mainWindow.py	167
src/marques.py	167
src/mytextbrowser.py	167
src/nameAdrive.py	168
src/notification.py	168
src/ownedUsbDisk.py	168
src/preferences.py	169
src/scolasync.py	169
src/sconet.py	170
src/test3.py	170
src/usbDisk2.py	170
src/usbThread.py	171
src/version.py	171

Chapitre 6

Documentation des espaces de nommage

6.1 Référence de l'espace de nommage scolasync

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

6.1.1 Description détaillée

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

6.2 Référence de l'espace de nommage src

Espaces de nommage

- [checkBoxDialog](#)
- [choixEleves](#)
- [chooseInSticks](#)
- [copyToDialog1](#)
- [db](#)
- [debug](#)
- [diskFull](#)
- [gestClasse](#)
- [gestclassetreeview](#)
- [globaldef](#)
- [help](#)
- [mainWindow](#)
- [marques](#)
- [mytextbrowser](#)
- [nameAdrive](#)
- [notification](#)
- [ownedUsbDisk](#)
- [preferences](#)
- [scolasync](#)
- [sconet](#)
- [test3](#)
- [usbDisk2](#)
- [usbThread](#)
- [version](#)

6.3 Référence de l'espace de nommage src.checkBoxDialog

Classes

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

Variables

- string `licenceEn`

6.3.1 Documentation des variables

6.3.1.1 string `src.checkBoxDialog.licenceEn`

Valeur initiale :

```

1 = """
2     file checkBoxDialog.py
3     this file is part of the project scolasync
4
5     Copyright (C) 2010 Georges Khaznadar <georgesk@ofset.org>
6
7     This program is free software: you can redistribute it and/or modify
8     it under the terms of the GNU General Public License as published by
9     the Free Software Foundation, either version3 of the License, or
10    (at your option) any later version.
11
12    This program is distributed in the hope that it will be useful,
13    but WITHOUT ANY WARRANTY; without even the implied warranty of
14    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15    GNU General Public License for more details.
16
17    You should have received a copy of the GNU General Public License
18    along with this program. If not, see <http://www.gnu.org/licenses/>.
19 """

```

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

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

Classes

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

Variables

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

6.4.1 Documentation des variables

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

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

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

Définition à la ligne 257 du fichier choixEleves.py.

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

Définition à la ligne 260 du fichier choixEleves.py.

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

Définition à la ligne 3 du fichier choixEleves.py.

6.5 Référence de l'espace de nommage src.chooseInSticks

Classes

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

Variables

- string `licenceEn`

6.5.1 Documentation des variables

6.5.1.1 `string src.chooseInSticks.licenceEn`

Valeur initiale :

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

Définition à la ligne 3 du fichier chooseInSticks.py.

6.6 Référence de l'espace de nommage src.copyToDialog1

Classes

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

Variables

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

6.6.1 Documentation des variables

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

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

6.6.1.2 string `src.copyToDialog1.licenceEn`

Valeur initiale :

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

Définition à la ligne 3 du fichier `copyToDialog1.py`.

6.6.1.3 tuple `src.copyToDialog1.windows = copyToDialog1()`

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

6.7 Référence de l'espace de nommage `src.db`

Fonctions

- def `openDb()`
Ouverture de la base de données de l'application, et création si nécessaire.
- def `checkVersion` (major, minor)
Vérifie si la base de données reste compatible.
- def `hasStudent` (student)
vérifie qu'un étudiant est déjà connu
- def `knowsId` (stickid, uuid, tattoo)
dit si une clé USB est déjà connue
- def `tattooList()`
Renvoie la liste des tatouages connus de la base de données.
- def `readStudent` (stickid, uuid, tattoo)
renvoie l'étudiant qui possède une clé USB
- def `readPrefs()`
renvoie les préférences de ScolaSync
- def `setWd` (newDir)
définit le nouveau nom du répertoire de travail préféré.
- def `writeStudent` (stickid, uuid, tattoo, student)
inscrit un étudiant comme propriétaire d'une clé USB

— def `writePrefs` (prefs)
 inscrit les préférences

Variables

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

6.7.1 Documentation des fonctions

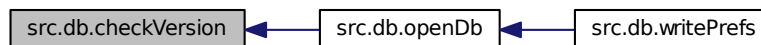
6.7.1.1 def src.db.checkVersion (major, minor)

Vérifie si la base de données reste compatible.

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

Définition à la ligne 56 du fichier db.py.

Voici le graphe des appelants de cette fonction :



6.7.1.2 def src.db.hasStudent (student)

vérifie qu'un étudiant est déjà connu

Paramètres

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

Renvoie

True si le propriétaire existe déjà

Définition à la ligne 78 du fichier db.py.

6.7.1.3 def src.db.knowsId (stickid, uuid, tattoo)

dit si une clé USB est déjà connue

Paramètres

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

Renvoie

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

Définition à la ligne 91 du fichier db.py.

Voici le graphe des appelants de cette fonction :

**6.7.1.4 def src.db.openDb ()**

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

Renvoie

une instance de base de données sqlite3

Définition à la ligne 36 du fichier db.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

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

renvoie les préférences de ScolaSync

Renvoie

un dictionnaire de préférences

Définition à la ligne 124 du fichier db.py.

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

renvoie l'étudiant qui possède une clé USB

Renvoie

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

Définition à la ligne 110 du fichier db.py.

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

définit le nouveau nom du répertoire de travail préféré.

Définition à la ligne 153 du fichier db.py.

6.7.1.8 def src.db.tattooList ()

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

Définition à la ligne 100 du fichier db.py.

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

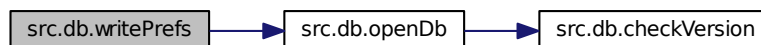
inscrit les préférences

Paramètres

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

Définition à la ligne 179 du fichier db.py.

Voici le graphe d'appel pour cette fonction :

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

inscrit un étudiant comme propriétaire d'une clé USB

Paramètres

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

Définition à la ligne 163 du fichier db.py.

Voici le graphe d'appel pour cette fonction :

**6.7.2 Documentation des variables****6.7.2.1 src.db.cursor = None**

Définition à la ligne 29 du fichier db.py.

6.7.2.2 src.db.database = None

Définition à la ligne 28 du fichier db.py.

6.7.2.3 dictionary src.db.licence = {}

Définition à la ligne 3 du fichier db.py.

6.8 Référence de l'espace de nommage src.debug**Fonctions**

- def `button` (w, cb)
ajoute un bouton de débogage dans une fenêtre
- def `listePartitionsCochees` (w)
renseigne sur la liste des partions cochées de la fenêtre principale

Variables

- dictionary `licence` = {}
Ce module facilite le debogage.
- string `licenceEn`
- string `licenceFr`

6.8.1 Documentation des fonctions**6.8.1.1 def src.debug.button (w, cb)**

ajoute un bouton de débogage dans une fenêtre

Paramètres

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

Définition à la ligne 62 du fichier debug.py.

6.8.1.2 def src.debug.listePartitionsCochees (w)

renseigne sur la liste des partions cochées de la fenêtre principale

Paramètres

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

Définition à la ligne 81 du fichier debug.py.

6.8.2 Documentation des variables**6.8.2.1 dictionary src.debug.licence = {}**

Ce module facilite le debogage.

Définition à la ligne 8 du fichier debug.py.

6.8.2.2 string src.debug.licenceEn**Valeur initiale :**

```

1 = """
2     scolasync version %s:
3
4     a program to manage file transfers between a computer and a collection
5     of USB sticks.
6
7     Copyright (C) 2010-2013 Georges Khaznadar <georgesk@debian.org>
8
9     This program is free software: you can redistribute it and/or modify
10    it under the terms of the GNU General Public License as published by
11    the Free Software Foundation, either version 3 of the License, or
12    (at your option) any later version.
13
14    This program is distributed in the hope that it will be useful,
15    but WITHOUT ANY WARRANTY; without even the implied warranty of
16    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17    GNU General Public License for more details.
18
19    You should have received a copy of the GNU General Public License
20    along with this program. If not, see <http://www.gnu.org/licenses/>.
21 """

```

Définition à la ligne 9 du fichier debug.py.

6.8.2.3 string src.debug.licenceFr**Valeur initiale :**

```

1 = """
2     scolasync version %s :
3
4     un programme pour gérer des transferts de fichiers entre un
5     ordinateur et une collection de clés USB.
6

```

```

7      Copyright (C) 2010-2013 Georges Khaznadar <georgesk@debian.org>
8
9      Ce projet est un logiciel libre : vous pouvez le redistribuer, le
10     modifier selon les terme de la GPL (GNU Public License) dans les
11     termes de la Free Software Foundation concernant la version 3 ou
12     plus de la dite licence.
13
14     Ce programme est fait avec l'espoir qu'il sera utile mais SANS
15     AUCUNE GARANTIE. Lisez la licence pour plus de détails.
16
17     <http://www.gnu.org/licenses/>.
18 """

```

Définition à la ligne 32 du fichier debug.py.

6.9 Référence de l'espace de nommage src.diskFull

Classes

— class [mainWindow](#)

Fonctions

— def [sceneWithUsage](#) (parent, rect, percent)

Variables

— dictionary [licence](#) = {}

6.9.1 Documentation des fonctions

6.9.1.1 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.

6.9.2 Documentation des variables

6.9.2.1 dictionary src.diskFull.licence = {}

Définition à la ligne 4 du fichier diskFull.py.

6.10 Référence de l'espace de nommage src.gestClasse

Classes

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

Variables

- dictionary `licence` = {}
Ce module permet de gérer des classes d'élèves.

6.10.1 Documentation des variables

6.10.1.1 dictionary `src.gestClasse.licence` = {}

Ce module permet de gérer des classes d'élèves.

La classe `AbstractGestClasse` définit les fonctions minimales à implémenter pour chaque gestionnaire de classes.

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

6.11 Référence de l'espace de nommage src.gestclasstreeview

Classes

- class `gestClasseTreeView`

Variables

- dictionary `licence` = {}

6.11.1 Documentation des variables

6.11.1.1 dictionary `src.gestclasstreeview.licence` = {}

Définition à la ligne 3 du fichier `gestclasstreeview.py`.

6.12 Référence de l'espace de nommage src.globaldef

Fonctions

- def `firstdir` (l)
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"

6.12.1 Documentation des fonctions

6.12.1.1 def `src.globaldef.firstdir` (l)

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

Paramètres

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

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

6.12.2 Documentation des variables**6.12.2.1 `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 10 du fichier `globaldef.py`.

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

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

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

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

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

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

6.13 Référence de l'espace de nommage `src.help`**Classes**

— class [helpWindow](#)

Variables

— dictionary [licence](#) = {}

6.13.1 Documentation des variables

6.13.1.1 dictionary src.help.licence = {}

Définition à la ligne 4 du fichier help.py.

6.14 Référence de l'espace de nommage src.mainWindow

Classes

- class `CheckBoxDelegate`
- class `DiskSizeDelegate`
Classe pour figurer la taille de la mémoire du baladeur.
- class `mainWindow`
defines the main window of the application.
- class `UsbDiskDelegate`
Classe pour identifier le baladeur dans le tableau.
- class `usbTableModel`
Un modèle de table pour des séries de clés USB.

Fonctions

- def `registerCmd` (cmd, partition)
enregistre la commande cmd pour la partition donnée
- def `CheckBoxRect` (view_item_style_options)

Variables

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

6.14.1 Documentation des fonctions

6.14.1.1 def src.mainWindow.CheckBoxRect (view_item_style_options)

Paramètres

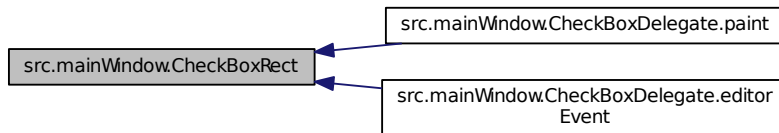
<code>view_item_↔ style_options</code>	des options permettant de décider de la taille d'un rectangle
--	---

Renvoie

un QRect dimensionné selon les bonnes options

Définition à la ligne 867 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

**6.14.1.2 def src.mainWindow.registerCmd (cmd, partition)**

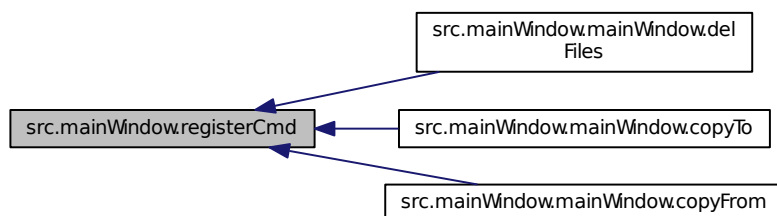
enregistre la commande cmd pour la partition donnée

Paramètres

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

Définition à la ligne 54 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

**6.14.2 Documentation des variables****6.14.2.1 dictionary src.mainWindow.activeThreads = {}**

Définition à la ligne 42 du fichier mainWindow.py.

6.14.2.2 src.mainWindow.lastCommand = None

Définition à la ligne 46 du fichier mainWindow.py.

6.14.2.3 dictionary src.mainWindow.licence = {}

Définition à la ligne 4 du fichier mainWindow.py.

6.14.2.4 dictionary src.mainWindow.pastCommands = {}

Définition à la ligne 45 du fichier mainWindow.py.

6.15 Référence de l'espace de nommage src.marques

6.16 Référence de l'espace de nommage src.mytextbrowser

Classes

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

Variables

- dictionary [licence](#) = {}

6.16.1 Documentation des variables

6.16.1.1 dictionary src.mytextbrowser.licence = {}

Définition à la ligne 4 du fichier mytextbrowser.py.

6.17 Référence de l'espace de nommage src.nameAdrive

Classes

- class [nameAdriveDialog](#)
un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles

Variables

- dictionary [licence](#) = {}

6.17.1 Documentation des variables

6.17.1.1 dictionary src.nameAdrive.licence = {}

Définition à la ligne 3 du fichier nameAdrive.py.

6.18 Référence de l'espace de nommage src.notification

Classes

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

Variables

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

6.18.1 Documentation des variables

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

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

6.18.1.2 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 74 du fichier `notification.py`.

6.19 Référence de l'espace de nommage `src.ownedUsbDisk`

Classes

- class `Available`
Une classe qui fournit une collection de disques USB connectés, avec leurs propriétaires.
- class `MainWindow`
- class `uDisk2`
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.

Fonctions

- def `tattooInDir` (mountPoint)
Renvoie le tatouage pour un point de montage donné, quitte à le créer si nécessaire.
- def `editRecord`
édition de la base de données.
- def `print_targets_if_modif` (man, obj)

Variables

- dictionary `licence` = {}
- tuple `app` = `QApplication(sys.argv)`
- tuple `main` = `MainWindow()`

6.19.1 Documentation des fonctions

6.19.1.1 def `src.ownedUsbDisk.editRecord` (*owd*, *hint* = " ")

édition de la base de données.

Paramètres

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

Définition à la ligne 69 du fichier `ownedUsbDisk.py`.

6.19.1.2 `def src.ownedUsbDisk.print_targets_if_modif (man, obj)`

Définition à la ligne 332 du fichier `ownedUsbDisk.py`.

6.19.1.3 `def src.ownedUsbDisk.tattooInDir (mountPoint)`

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

Paramètres

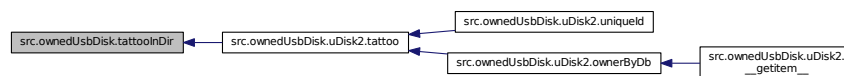
<i>mountPoint</i>	un point de montage de partition
-------------------	----------------------------------

Renvoie

le tatouage

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

Voici le graphe des appelants de cette fonction :



6.19.2 Documentation des variables

6.19.2.1 `tuple src.ownedUsbDisk.app = QApplication(sys.argv)`

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

6.19.2.2 `dictionary src.ownedUsbDisk.licence = {}`

Définition à la ligne 3 du fichier `ownedUsbDisk.py`.

6.19.2.3 `tuple src.ownedUsbDisk.main = MainWindow()`

Définition à la ligne 343 du fichier `ownedUsbDisk.py`.

6.20 Référence de l'espace de nommage src.preferences

Classes

— class [preferenceWindow](#)

Variables

— dictionary `licence` = {}

6.20.1 Documentation des variables

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

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

6.21 Référence de l'espace de nommage `src.scolasync`

Fonctions

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

Variables

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

6.21.1 Documentation des fonctions

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

Le lancement de l'application.

Paramètres

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

Définition à la ligne 147 du fichier `scolasync.py`.

6.21.2 Documentation des variables

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

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

6.21.2.2 string `src.scolasync.licenceEn`

Valeur initiale :

```

1 = """
2     scolasync version %s:
3
4     a program to manage file transfers between a computer and a collection
5     of USB sticks.
6
7     Copyright (C) 2010-2012 Georges Khaznadar <georgesk@offset.org>
8
9     This program is free software: you can redistribute it and/or modify
10    it under the terms of the GNU General Public License as published by
11    the Free Software Foundation, either version 3 of the License, or
12    (at your option) any later version.
13
```



```

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

```

Définition à la ligne 85 du fichier scolasync.py.

6.21.2.3 string src.scolasync.licenceFr

Valeur initiale :

```

1 = """
2     scolasync version %s :
3
4     un programme pour gérer des transferts de fichiers entre un
5     ordinateur et une collection de clés USB.
6
7     Copyright (C) 2010-2012 Georges Khaznadar <georgesk@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 108 du fichier scolasync.py.

6.22 Référence de l'espace de nommage src.sconet

Classes

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

Variables

- dictionary [licence](#) = {}
- tuple [s](#) = [Sconet](#)("../exemples/SCONET_test.xml")

6.22.1 Documentation des variables

6.22.1.1 dictionary src.sconet.licence = {}

Définition à la ligne 3 du fichier sconet.py.

6.22.1.2 tuple src.sconet.s = Sconet("../exemples/SCONET_test.xml")

Définition à la ligne 102 du fichier sconet.py.

6.23 Référence de l'espace de nommage src.test3

Variables

```

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

```

6.23.1 Documentation des variables

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

Définition à la ligne 8 du fichier test3.py.

6.23.1.2 `tuple src.test3.module = __import__(moduleName)`

Définition à la ligne 16 du fichier test3.py.

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

Définition à la ligne 14 du fichier test3.py.

6.23.1.4 `list src.test3.notsafe = []`

Définition à la ligne 12 du fichier test3.py.

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

Définition à la ligne 9 du fichier test3.py.

6.23.1.6 `src.test3.python3safe = True`

Définition à la ligne 5 du fichier test3.py.

6.23.1.7 `list src.test3.safe = []`

Définition à la ligne 11 du fichier test3.py.

6.24 Référence de l'espace de nommage src.usbDisk2

Classes

```

— class Available
    une classe pour représenter la collection des disques USB connectés
— class MainWindow
— class uDisk2
    une classe pour représenter un disque ou une partition.
— class UDisksBackend
    Cette classe a été inspirée par le projet USBcreator.

```

Fonctions

- def `inspectData` ()
- def `safePath` (obj)
- def `fs_size` (device)
Récupère de façon sûre le path d'une instance de `UDisksObjectProxy`.
- def `print_targets_if_modif` (man, obj)
Renvoie la taille d'un système de fichier et la place disponible.

Variables

- dictionary `licence` = {}
- string `licence_en`
- string `dependences` = "python3-dbus python3-dbus.mainloop.qt"
- `debug` = False
activate debugging #####
- tuple `no_options` = `GLib.Variant('a{sv}', {})`
la variable suivante a été recopiées à l'aveugle ##### depuis un fichier du projet USBcreator
#####
- tuple `not_interesting`
des "chemins" correspondant à des disques non débranchables #####
- tuple `app` = `QApplication(sys.argv)`
- tuple `main` = `MainWindow()`

6.24.1 Documentation des fonctions

6.24.1.1 def src.usbDisk2.fs_size (device)

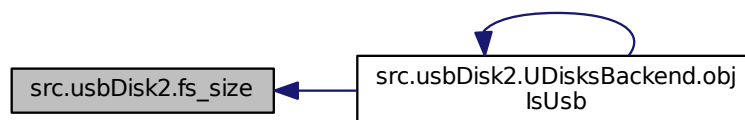
Renvoie la taille d'un système de fichier et la place disponible.

Renvoie

un tuple : taille totale et espace libre

Définition à la ligne 76 du fichier `usbDisk2.py`.

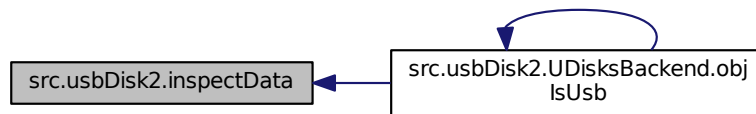
Voici le graphe des appelants de cette fonction :



6.24.1.2 def src.usbDisk2.inspectData ()

Définition à la ligne 36 du fichier `usbDisk2.py`.

Voici le graphe des appelants de cette fonction :



6.24.1.3 `def src.usbDisk2.print_targets_if_modif (man, obj)`

Définition à la ligne 802 du fichier `usbDisk2.py`.

6.24.1.4 `def src.usbDisk2.safePath (obj)`

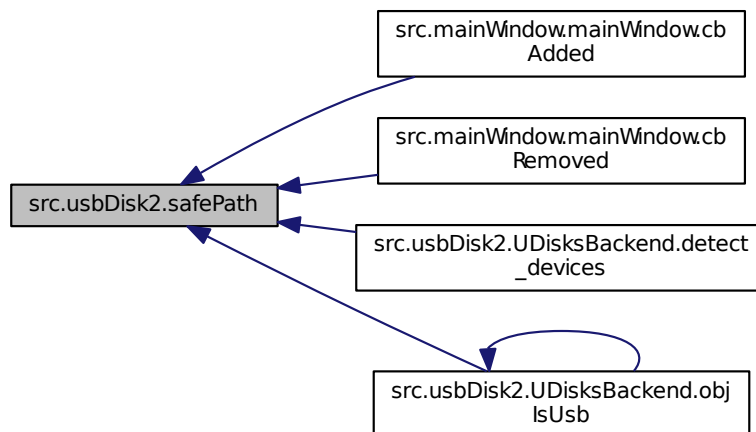
Récupère de façon sûre le path d'une instance de `UDisksObjectProxy`.

Paramètres

<i>obj</i>	instance de <code>UDisksObjectProxy</code> , ou simple chaîne
------------	---

Définition à la ligne 60 du fichier `usbDisk2.py`.

Voici le graphe des appelants de cette fonction :



6.24.2 Documentation des variables

6.24.2.1 `tuple src.usbDisk2.app = QApplication(sys.argv)`

Définition à la ligne 809 du fichier `usbDisk2.py`.

6.24.2.2 `src.usbDisk2.debug = False`

activate debugging #####

Définition à la ligne 35 du fichier usbDisk2.py.

6.24.2.3 `string src.usbDisk2.dependencies = "python3-dbus python3-dbus.mainloop.qt"`

Définition à la ligne 26 du fichier usbDisk2.py.

6.24.2.4 `dictionary src.usbDisk2.licence = {}`

Définition à la ligne 3 du fichier usbDisk2.py.

6.24.2.5 `string src.usbDisk2.licence_en`

Valeur initiale :

```
1 = """
2     file usbDisk2.py
3     this file is part of the project scolasync. It is a rewrite of
4     usbDisk.py to take in account udisks2.
5
6     Copyright (C) 2014 Georges Khaznadar <georgesk@ofset.org>
7
8     This program is free software: you can redistribute it and/or modify
9     it under the terms of the GNU General Public License as published by
10    the Free Software Foundation, either version3 of the License, or
11    (at your option) any later version.
12
13    This program is distributed in the hope that it will be useful,
14    but WITHOUT ANY WARRANTY; without even the implied warranty of
15    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16    GNU General Public License for more details.
17
18    You should have received a copy of the GNU General Public License
19    along with this program. If not, see <http://www.gnu.org/licenses/>.
20 """
```

Définition à la ligne 4 du fichier usbDisk2.py.

6.24.2.6 `tuple src.usbDisk2.main = MainWindow()`

Définition à la ligne 810 du fichier usbDisk2.py.

6.24.2.7 `tuple src.usbDisk2.no_options = GLib.Variant('a{sv}', {})`

la variable suivante a été recopiées à l'aveugle ##### depuis un fichier du projet USBcreator
#####

Définition à la ligne 88 du fichier usbDisk2.py.

6.24.2.8 `tuple src.usbDisk2.not_interesting`

Valeur initiale :

```
1 = (
2     # boucle
3     '/org/freedesktop/UDisks2/block_devices/loop',
4     # disque raid
5     '/org/freedesktop/UDisks2/block_devices/dm_',
6     # mémoire vive
7     '/org/freedesktop/UDisks2/block_devices/ram',
```

```

8      '/org/freedesktop/UDisks2/block_devices/zram',
9      # disques durs
10     '/org/freedesktop/UDisks2/drives/' ,
11     )

```

des "chemins" correspondant à des disques non débranchables #####

Définition à la ligne 93 du fichier usbDisk2.py.

6.25 Référence de l'espace de nommage src.usbThread

Classes

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

Fonctions

- def [ensureDirExists](#) (destpath)
force l'existence d'un répertoire, récursivement si nécessaire
- def [test_copytree](#) ()
Teste la fonction copytree.
- def [test_copy2](#) ()
Teste la copie d'un fichier vers une destination telle qu'elle est pratiquée dans la méthode copytree de [abstractThreadUSB](#).

Variables

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

6.25.1 Documentation des fonctions

6.25.1.1 def src.usbThread.ensureDirExists (destpath)

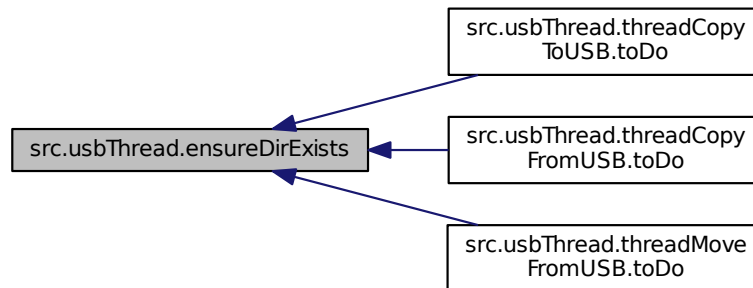
force l'existence d'un répertoire, récursivement si nécessaire

Paramètres

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

Définition à la ligne 34 du fichier usbThread.py.

Voici le graphe des appelants de cette fonction :



6.25.1.2 def src.usbThread.test_copy2 ()

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

Définition à la ligne 592 du fichier usbThread.py.

6.25.1.3 def src.usbThread.test_copytree ()

Teste la fonction copytree.

Définition à la ligne 575 du fichier usbThread.py.

6.25.2 Documentation des variables

6.25.2.1 int src.usbThread._threadNumber = 0

Définition à la ligne 27 du fichier usbThread.py.

6.25.2.2 string src.usbThread.licenceEn

Valeur initiale :

```

1 = """
2   file usbThread.py
3   this file is part of the project scolasync
4
5   Copyright (C) 2010-2012 Georges Khaznadar <georgesk@ofset.org>
6
7   This program is free software: you can redistribute it and/or modify
8   it under the terms of the GNU General Public License as published by
9   the Free Software Foundation, either version3 of the License, or
10  (at your option) any later version.
11
12  This program is distributed in the hope that it will be useful,
13  but WITHOUT ANY WARRANTY; without even the implied warranty of
14  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
15  GNU General Public License for more details.
16
17  You should have received a copy of the GNU General Public License
18  along with this program. If not, see <http://www.gnu.org/licenses/>.
19 """
  
```

Définition à la ligne 3 du fichier usbThread.py.

6.26 Référence de l'espace de nommage src.version

Fonctions

```
— def major ()  
— def minor ()  
— def version ()
```

Variables

```
— dictionary licence = {}
```

6.26.1 Documentation des fonctions

6.26.1.1 def src.version.major ()

Renvoie

le numéro majeur de version

Définition à la ligne 28 du fichier version.py.

Voici le graphe des appelants de cette fonction :



6.26.1.2 def src.version.minor ()

Renvoie

le numéro mineur de version

Définition à la ligne 35 du fichier version.py.

Voici le graphe des appelants de cette fonction :



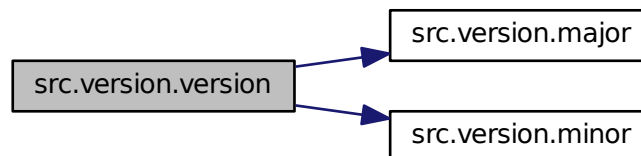
6.26.1.3 `def src.version.version ()`

Renvoie

l'identifiant de la version

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

Voici le graphe d'appel pour cette fonction :



6.26.2 Documentation des variables

6.26.2.1 `dictionary src.version.licence = {}`

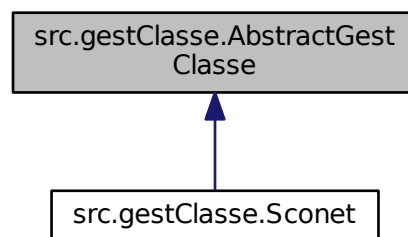
Définition à la ligne 3 du fichier `version.py`.

Chapitre 7

Documentation des classes

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

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



Fonctions membres publiques

- `def __init__ (self, f)`
le constructeur
- `def collectClasses (self)`
- `def elevesDeClasse (self, cl)`
- `def unique_name (self, el)`
- `def showable_name (self, el)`

7.1.1 Description détaillée

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

7.1.2 Documentation des constructeurs et destructeur

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

le constructeur

Paramètres

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

Définition à la ligne 41 du fichier `gestClasse.py`.

7.1.3 Documentation des fonctions membres**7.1.3.1 `def src.gestClasse.AbstractGestClasse.collectClasses (self)`****Renvoie**

une liste de noms de classes d'un établissement scolaire

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

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

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

Renvoie

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

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

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

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

Renvoie

une chaîne unicode, pour nommer l'élève

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

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

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

Renvoie

une chaîne unicode, unique dans l'établissement

Définition à la ligne 64 du fichier `gestClasse.py`.

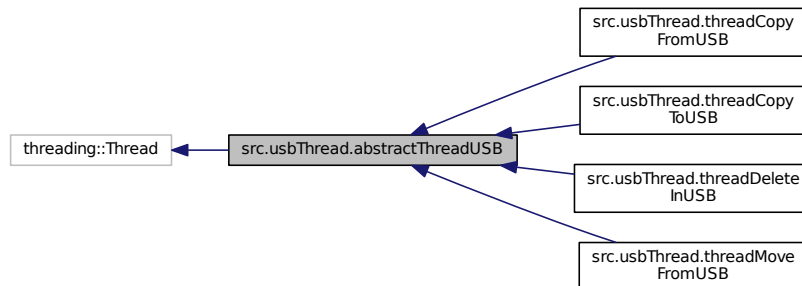
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/gestClasse.py](#)

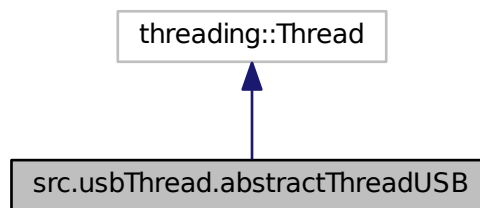
7.2 Référence de la classe src.usbThread.abstractThreadUSB

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

Graphe d'héritage de src.usbThread.abstractThreadUSB :



Graphe de collaboration de src.usbThread.abstractThreadUSB :



Fonctions membres publiques

- def `__init__`
Constructeur Crée un thread pour copier une liste de fichiers vers une clé USB.
- def `run` (self)
- def `writeToLog` (self, msg)
Écrit un message dans le fichier de journalisation.
- def `copytree`
Une version modifiée de shutil.copytree qui accepte que les répertoires destination soient déjà existants.
- def `__str__` (self)
Renvoie une chaîne informative sur le thread.
- def `threadType` (self)
information sur le thread.
- def `todo` (self, ud, fileList, subdir, dest, logfile)
La fonction abstraite pour les choses à faire.

Attributs publics

- ud
- fileList
- subdir
- dest

- logfile
- parent

7.2.1 Description détaillée

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

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

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

Définition à la ligne 149 du fichier usbThread.py.

7.2.2 Documentation des constructeurs et destructeur

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

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

Paramètres

<i>ud</i>	l'instance uDisk correspondant à une partition de clé USB
<i>fileList</i>	la liste des fichiers à traiter
<i>subdir</i>	un sous-répertoire de la clé USB
<i>dest</i>	un répertoire de destination si nécessaire, None par défaut
<i>logfile</i>	un fichier de journalisation, /dev/null par défaut
<i>parent</i>	un widget qui recevra de signaux en début et en fin d'exécution

Définition à la ligne 163 du fichier usbThread.py.

7.2.3 Documentation des fonctions membres

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

Renvoie une chaîne informative sur le thread.

Renvoie

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

Définition à la ligne 251 du fichier usbThread.py.

Voici le graphe d'appel pour cette fonction :



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

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

Cette source dérive de la documentation fournie avec Python 2.7

Paramètres

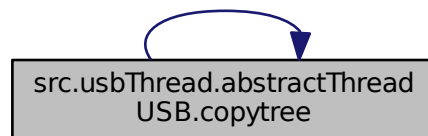
<i>src</i>	un nom de fichier ou de répertoire
<i>dst</i>	un nom de de répertoire (déjà existant ou à créer)
<i>symlinks</i>	vrai si on veut recopier les liens tels quels
<i>ignore</i>	une fonction qui construit une liste de fichiers à ignorer (profil : répertoire, liste de noms de fichiers -> liste de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

Renvoie

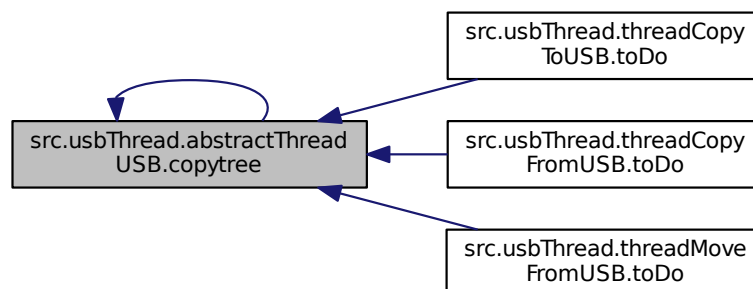
une liste d'erreurs éventuellement relevées, sinon une liste vide

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

Voici le graphe d'appel pour cette fonction :



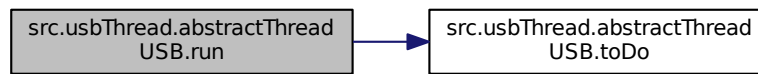
Voici le graphe des appelants de cette fonction :



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

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

Voici le graphe d'appel pour cette fonction :



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

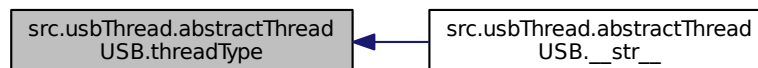
information sur le thread.

Renvoie

une chaîne courte qui informe sur le type de thread

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

Voici le graphe des appelants de cette fonction :



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

La fonction abstraite pour les choses à faire.

Paramètres

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

Définition à la ligne 278 du fichier `usbThread.py`.

Voici le graphe des appelants de cette fonction :



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

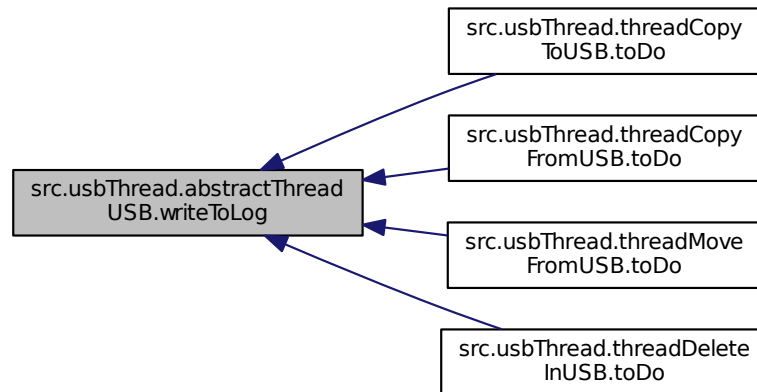
Écrit un message dans le fichier de journalisation.

Paramètres

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

Définition à la ligne 183 du fichier usbThread.py.

Voici le graphe des appelants de cette fonction :



7.2.4 Documentation des données membres

7.2.4.1 `src.usbThread.abstractThreadUSB.dest`

Définition à la ligne 170 du fichier usbThread.py.

7.2.4.2 `src.usbThread.abstractThreadUSB.fileList`

Définition à la ligne 168 du fichier usbThread.py.

7.2.4.3 `src.usbThread.abstractThreadUSB.logfile`

Définition à la ligne 171 du fichier usbThread.py.

7.2.4.4 `src.usbThread.abstractThreadUSB.parent`

Définition à la ligne 172 du fichier usbThread.py.

7.2.4.5 `src.usbThread.abstractThreadUSB.subdir`

Définition à la ligne 169 du fichier usbThread.py.

7.2.4.6 `src.usbThread.abstractThreadUSB.ud`

Définition à la ligne 166 du fichier usbThread.py.

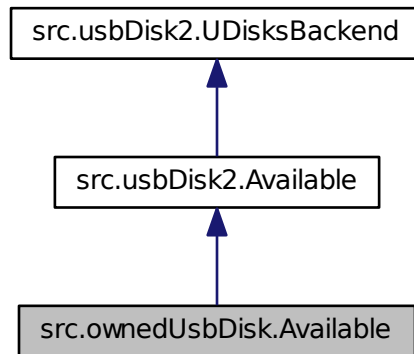
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

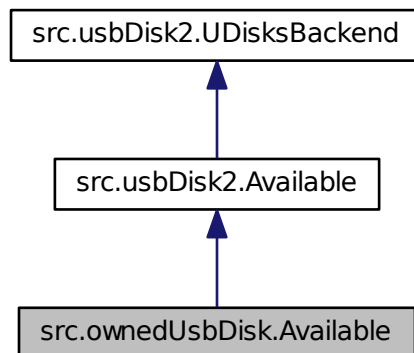
7.3 Référence de la classe src.ownedUsbDisk.Available

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

Graphe d'héritage de src.ownedUsbDisk.Available :



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

Attributs publics

- `ownerDialog`

7.3.1 Description détaillée

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

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

Définition à la ligne 288 du fichier `ownedUsbDisk.py`.

7.3.2 Documentation des constructeurs et destructeur

7.3.2.1 `def src.ownedUsbDisk.Available.__init__(self, access = "disk", diskClass = uDisk2, ownerDialog=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>access</i>	le mode d'accès : 'disk' ou 'firstFat'
<i>diskClass</i>	la classe d'objets à créer pour chaque disque
<i>ownerDialog</i>	vrai si on veut qu'il y ait un dialogue automatique pour déterminer le propriétaire des disques non reconnus

Définition à la ligne 300 du fichier `ownedUsbDisk.py`.

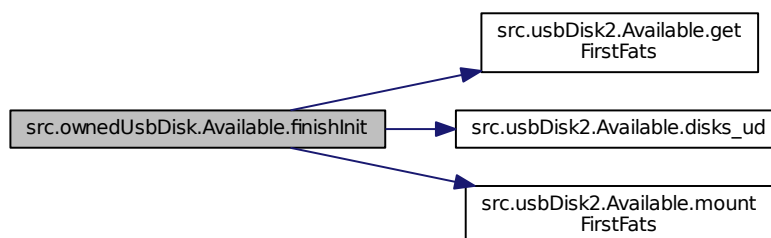
7.3.3 Documentation des fonctions membres

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

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

Définition à la ligne 310 du fichier `ownedUsbDisk.py`.

Voici le graphe d'appel pour cette fonction :



7.3.4 Documentation des données membres

7.3.4.1 `src.ownedUsbDisk.Available.ownerDialog`

Définition à la ligne 301 du fichier `ownedUsbDisk.py`.

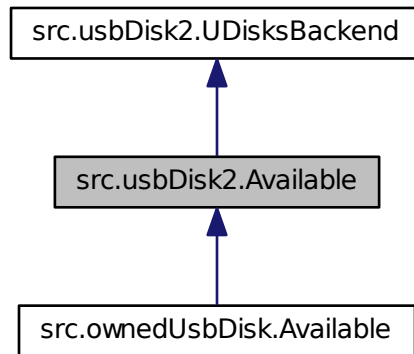
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/ownedUsbDisk.py](#)

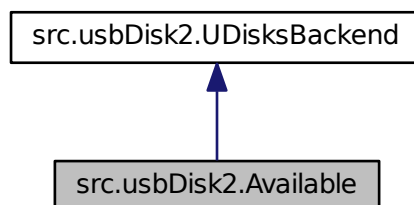
7.4 Référence de la classe src.usbDisk2.Available

une classe pour représenter la collection des disques USB connectés

Graphe d'héritage de src.usbDisk2.Available :



Graphe de collaboration de src.usbDisk2.Available :



Fonctions membres publiques

- def [__init__](#)
Le constructeur.
- def [finishInit](#) (self)
Fin de l'initialisation.
- def [mountFirstFats](#) (self)
fabrique la liste des partitions FAT, monte les partitions FAT si elles ne le sont pas
- def [__trunc__](#) (self)
- def [compare](#) (self, other)
Sert à comparer deux collections de disques, par exemple une collection passée et une collection présente.
- def [contains](#) (self, ud)
Permet de déterminer si un disque est dans la collection.
- def [disks](#) (self)
Réculte les enregistrements de niveau supérieur de self.targets.
- def [parts](#) (self, d)
Réculte les partitions d'un disque.

- `def disks_ud (self)`
Récolte les enregistrements de niveau supérieur de `self.targets`.
- `def parts_ud (self, d)`
Récolte les partitions d'un disque.
- `def summary (self)`
Fournit une représentation imprimable d'un résumé
- `def __str__ (self)`
Fournit une représentation imprimable.
- `def __getitem__ (self, n)`
Renvoie le nième disque.
- `def __len__ (self)`
Renseigne sur la longueur de la collection.
- `def getFirstFats (self)`
Facilite l'accès aux partitions de type DOS-FAT, et a des effets de bord :
- `def hasDev (self, dev)`

Attributs publics

- `access`
- `firstFats`

7.4.1 Description détaillée

une classe pour représenter la collection des disques USB connectés

les attributs publics sont :

- **access** le type d'accès qu'on veut pour les items
- **targets** 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
- **firstFats** une liste composée de la première partion DOS-FAT de chaque disque USB.

Définition à la ligne 592 du fichier `usbDisk2.py`.

7.4.2 Documentation des constructeurs et destructeur

7.4.2.1 `def src.usbDisk2.Available.__init__ (self, access = "disk", diskClass = uDisk2)`

Le constructeur.

Paramètres

<code>access</code>	définit le type d'accès souhaité. Par défaut, c'est "disk" c'est à dire qu'on veut la liste des disques USB. Autres valeurs possibles : "firstFat" pour les premières partitions vfat.
<code>diskClass</code>	la classe de disques à créer

Définition à la ligne 602 du fichier `usbDisk2.py`.

7.4.3 Documentation des fonctions membres

7.4.3.1 `def src.usbDisk2.Available.__getitem__ (self, n)`

Renvoie le nième disque.

Le fonctionnement dépend du paramètre `self.access`

Paramètres

<code>n</code>	un numéro
----------------	-----------

Renvoie

le nième disque USB connecté sous forme d'instance de [uDisk2](#)

Définition à la ligne 730 du fichier `usbDisk2.py`.

7.4.3.2 `def src.usbDisk2.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 743 du fichier `usbDisk2.py`.

7.4.3.3 `def src.usbDisk2.Available.__str__(self)`

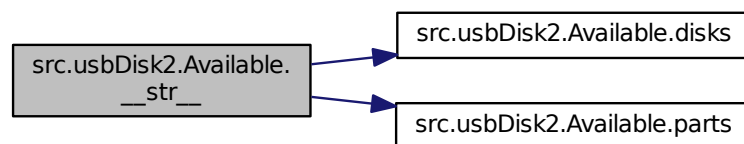
Fournit une représentation imprimable.

Renvoie

une représentation imprimable de la collection

Définition à la ligne 710 du fichier `usbDisk2.py`.

Voici le graphe d'appel pour cette fonction :

**7.4.3.4 `def src.usbDisk2.Available.__trunc__(self)`****Renvoie**

le nombre de medias connectés

Définition à la ligne 630 du fichier `usbDisk2.py`.

7.4.3.5 `def src.usbDisk2.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 Available
--------------	---

Renvoie

vrai si other semble être la même collection de disques USB

Définition à la ligne 640 du fichier usbDisk2.py.

Voici le graphe d'appel pour cette fonction :



7.4.3.6 def src.usbDisk2.Available.contains (self, ud)

Permet de déterminer si un disque est dans la collection.

Paramètres

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

Renvoie

vrai si le uDisk ud est dans la collection

Définition à la ligne 650 du fichier usbDisk2.py.

7.4.3.7 def src.usbDisk2.Available.disks (self)

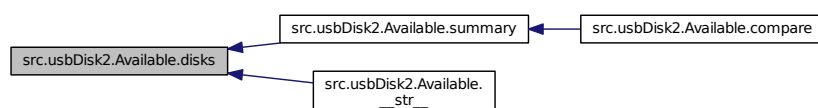
Récolte les enregistrements de niveau supérieur de self.targets.

Renvoie

la liste des chemins vers les disque USB détectés

Définition à la ligne 658 du fichier usbDisk2.py.

Voici le graphe des appelants de cette fonction :



7.4.3.8 def src.usbDisk2.Available.disks_ud (self)

Récolte les enregistrements de niveau supérieur de self.targets.

Renvoie

la liste des objets `uDisk2` détectés

Définition à la ligne 675 du fichier `usbDisk2.py`.

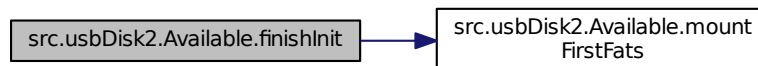
Voici le graphe des appelants de cette fonction :

**7.4.3.9 def `src.usbDisk2.Available.finishInit (self)`**

Fin de l'initialisation.

Définition à la ligne 612 du fichier `usbDisk2.py`.

Voici le graphe d'appel pour cette fonction :

**7.4.3.10 def `src.usbDisk2.Available.getFirstFats (self)`**

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

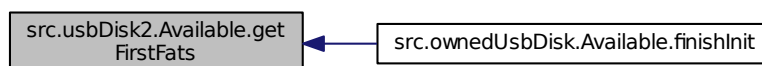
- marque la première vfat dans chaque instance de disque
- construit une liste des chemins uDisk des FATs

Renvoie

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

Définition à la ligne 758 du fichier `usbDisk2.py`.

Voici le graphe des appelants de cette fonction :

**7.4.3.11 def `src.usbDisk2.Available.hasDev (self, dev)`**

Paramètres

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

Renvoie

True si la partition est dans la liste des partions disponibles

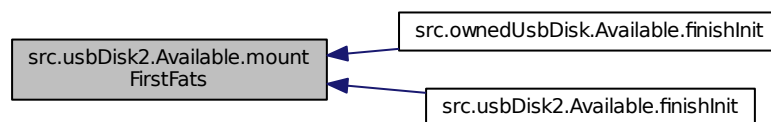
Définition à la ligne 775 du fichier usbDisk2.py.

7.4.3.12 `def src.usbDisk2.Available.mountFirstFats (self)`

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

Définition à la ligne 620 du fichier usbDisk2.py.

Voici le graphe des appelants de cette fonction :

7.4.3.13 `def src.usbDisk2.Available.parts (self, d)`

Récolte les partitions d'un disque.

Paramètres

<i>d</i>	le chemin vers un disque
----------	--------------------------

Renvoie

la liste des partitions de ce disque

Définition à la ligne 667 du fichier usbDisk2.py.

Voici le graphe des appelants de cette fonction :

7.4.3.14 `def src.usbDisk2.Available.parts_ud (self, d)`

Récolte les partitions d'un disque.

Paramètres

<code>d</code>	le chemin vers un disque
----------------	--------------------------

Renvoie

la liste des objets `uDisk2` qui sont des partitions de ce disque

Définition à la ligne 685 du fichier `usbDisk2.py`.

7.4.3.15 `def src.usbDisk2.Available.summary (self)`

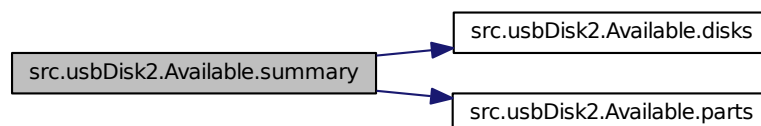
Fournit une représentation imprimable d'un résumé

Renvoie

une représentation imprimable d'un résumé de la collection

Définition à la ligne 693 du fichier `usbDisk2.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.4.4 Documentation des données membres

7.4.4.1 `src.usbDisk2.Available.access`

Définition à la ligne 604 du fichier `usbDisk2.py`.

7.4.4.2 `src.usbDisk2.Available.firstFats`

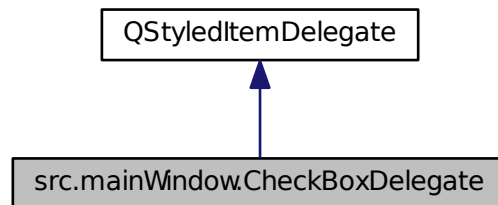
Définition à la ligne 621 du fichier `usbDisk2.py`.

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

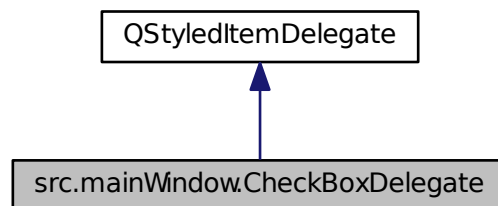
— `src/usbDisk2.py`

7.5 Référence de la classe src.mainWindow.CheckBoxDelegate

Graphe d'héritage de src.mainWindow.CheckBoxDelegate :



Graphe de collaboration de src.mainWindow.CheckBoxDelegate :



Fonctions membres publiques

- def `__init__`(self, parent)
- def `paint`(self, painter, option, index)
- def `editorEvent`(self, event, model, option, index)

7.5.1 Description détaillée

Définition à la ligne 873 du fichier `mainWindow.py`.

7.5.2 Documentation des constructeurs et destructeur

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

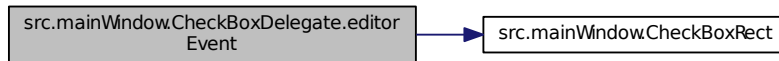
Définition à la ligne 874 du fichier `mainWindow.py`.

7.5.3 Documentation des fonctions membres

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

Définition à la ligne 888 du fichier mainWindow.py.

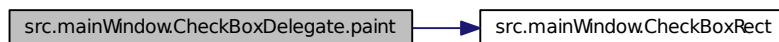
Voici le graphe d'appel pour cette fonction :



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

Définition à la ligne 877 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



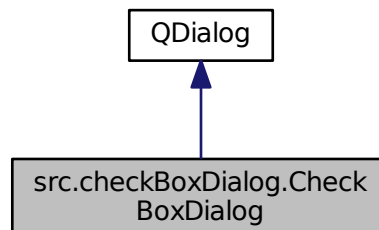
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

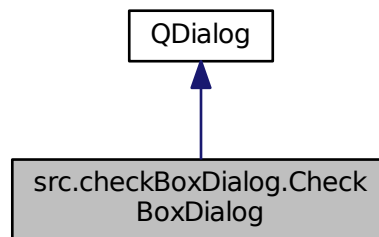
7.6 Référence de la classe src.checkBoxDialog.CheckBoxDialog

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

Graphe d'héritage de `src.checkBoxDialog.CheckBoxDialog` :



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



Fonctions membres publiques

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

Attributs publics

- `mainWindow`
- `ui`

7.6.1 Description détaillée

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

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

7.6.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

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

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

7.6.3 Documentation des fonctions membres

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

Fait cocher tous les baladeurs.

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

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

termine le dialogue sans rien faire

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

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

Fait décocher tous les boutons.

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

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

Fait inverser tous les boutons.

Définition à la ligne 58 du fichier `checkBoxDialog.py`.

7.6.4 Documentation des données membres

7.6.4.1 `src.checkBoxDialog.CheckBoxDialog.mainWindow`

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

7.6.4.2 `src.checkBoxDialog.CheckBoxDialog.ui`

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

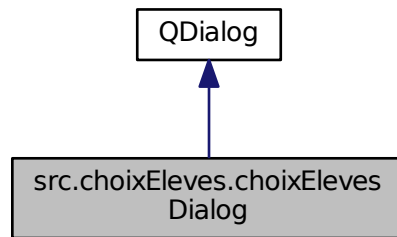
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/checkBoxDialog.py](#)

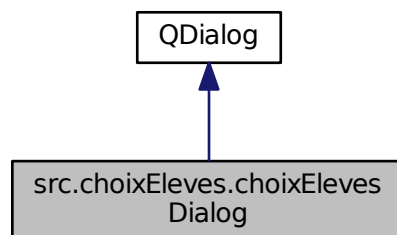
7.7 Référence de la classe `src.choixEleves.choixElevesDialog`

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

Graphe d'héritage de src.choixEleves.choixElevesDialog :



Graphe de collaboration de src.choixEleves.choixElevesDialog :



Fonctions membres publiques

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

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

Attributs publics

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

7.7.1 Description détaillée

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

Définition à la ligne 39 du fichier choixEleves.py.

7.7.2 Documentation des constructeurs et destructeur

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

le constructeur récupérer des données SCONET

Paramètres

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

Définition à la ligne 48 du fichier choixEleves.py.

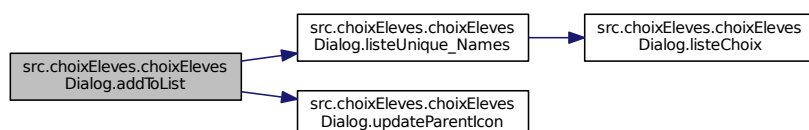
7.7.3 Documentation des fonctions membres

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

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

Définition à la ligne 154 du fichier choixEleves.py.

Voici le graphe d'appel pour cette fonction :



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

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

Paramètres

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

Définition à la ligne 107 du fichier `choixEleves.py`.

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

coche toutes les cases d'élèves visibles

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

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

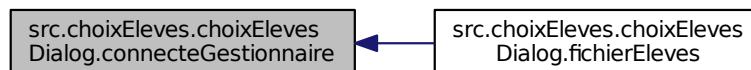
met en place l'arbre des noms d'élèves

Paramètres

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

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

Voici le graphe des appelants de cette fonction :

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

décoche toutes les cases d'élèves, visibles ou cachées

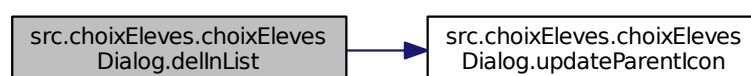
Définition à la ligne 135 du fichier `choixEleves.py`.

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

retire les élèves de la liste quand ils y sont sélectionnés

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

Voici le graphe d'appel pour cette fonction :

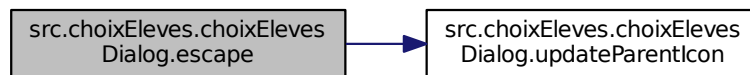


7.7.3.7 `def src.choixEleves.choixElevesDialog.escape (self)`

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

Définition à la ligne 229 du fichier choixEleves.py.

Voici le graphe d'appel pour cette fonction :



7.7.3.8 `def src.choixEleves.choixElevesDialog.fichierEleves (self)`

choisit et ouvre un nouveau fichiers d'élèves

Définition à la ligne 75 du fichier choixEleves.py.

Voici le graphe d'appel pour cette fonction :



7.7.3.9 `def src.choixEleves.choixElevesDialog.itemStrings (self)`

Renvoie

une liste des chaînes contenues dans les items

Définition à la ligne 193 du fichier choixEleves.py.

7.7.3.10 `def src.choixEleves.choixElevesDialog.listeChoix (self)`

Renvoie

la liste de `QStandardItem`s sélectionnés

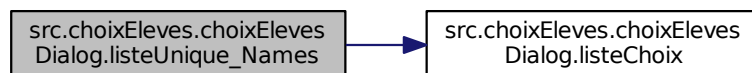
Définition à la ligne 241 du fichier `choixEleves.py`.

Voici le graphe des appelants de cette fonction :

**7.7.3.11 `def src.choixEleves.choixElevesDialog.listeUnique_Names (self)`**

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

**7.7.3.12 `def src.choixEleves.choixElevesDialog.pop (self)`**

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

Renvoie

un nom pour un baladeur, sinon None

Définition à la ligne 181 du fichier choixEleves.py.

Voici le graphe d'appel pour cette fonction :

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

replie toutes les classes du dialogue

Définition à la ligne 118 du fichier choixEleves.py.

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

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

Paramètres

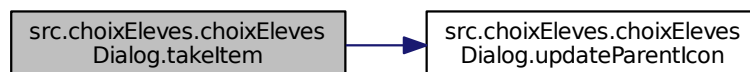
<i>une</i>	chaîne donnant le texte d'un item à trouver
------------	---

Renvoie

un nom pour un baladeur, sinon None

Définition à la ligne 205 du fichier choixEleves.py.

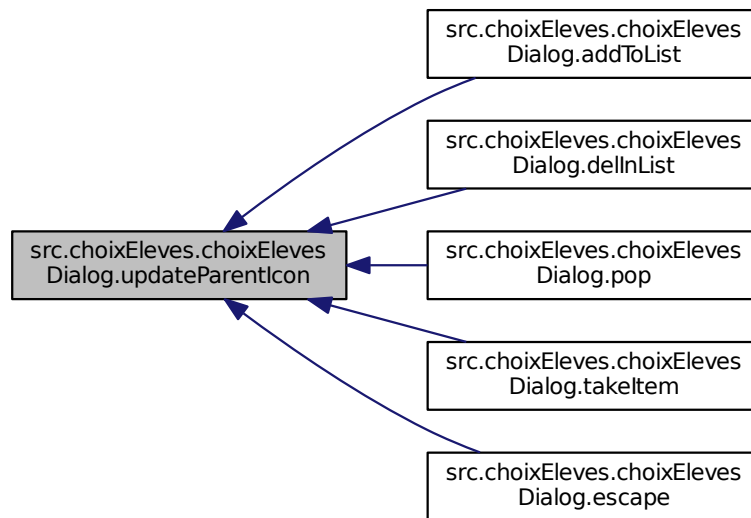
Voici le graphe d'appel pour cette fonction :

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

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

Définition à la ligne 145 du fichier choixEleves.py.

Voici le graphe des appelants de cette fonction :



7.7.3.16 `def src.choixEleves.choixElevesDialog.valid (self)`

Prend acte de la validation.

Définition à la ligne 219 du fichier `choixEleves.py`.

7.7.4 Documentation des données membres

7.7.4.1 `src.choixEleves.choixElevesDialog.gestionnaire`

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

7.7.4.2 `src.choixEleves.choixElevesDialog.ok`

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

7.7.4.3 `src.choixEleves.choixElevesDialog.prefs`

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

7.7.4.4 `src.choixEleves.choixElevesDialog.ui`

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

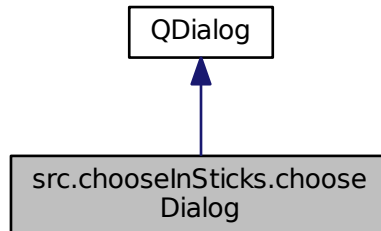
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/choixEleves.py](#)

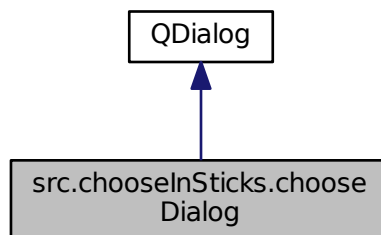
7.8 Référence de la classe src.chooseInSticks.chooseDialog

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

Graphe d'héritage de src.chooseInSticks.chooseDialog :



Graphe de collaboration de src.chooseInSticks.chooseDialog :



Fonctions membres publiques

- def `__init__`
Le constructeur.
- def `checkValues` (self)
fonction de rappel liée au bouton de validation, vérifie s'il y a bien au moins un fichier ou un répertoire sélectionné
- def `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.
- def `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.
- def `baseDir` (self)
- def `selectedDiskMountPoint` (self)
- def `selectedDiskOwner` (self)
- def `changeWd` (self)
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` (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.*
- `def activate (self, item)`
Fonction de rappel quand un item de la liste est activé
 - `def plus (self)`
Permet de choisir et d'ajouter un nouveau fichier ou répertoire à supprimer.
 - `def minus (self)`
Permet de retirer de la liste des fichiers à supprimer ceux qu'on a sélectionnés.
 - `def append (self, path)`
Ajoute un chemin avec ou sans jokers à la liste des chemins à supprimer.
 - `def 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.*

Attributs publics

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

7.8.1 Description détaillée

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

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

7.8.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

<i>parent</i>	un <code>mainWindow</code> , 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>okPrompt</i>	le texte du bouton OK

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

7.8.3 Documentation des fonctions membres

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

Fonction de rappel quand un item de la liste est activé

Paramètres

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

Définition à la ligne 250 du fichier `chooseInSticks.py`.

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

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

Paramètres

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

Définition à la ligne 284 du fichier chooseInSticks.py.

Voici le graphe des appelants de cette fonction :



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

Renvoie

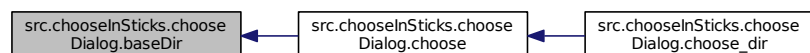
le répertoire à partir duquel on peut commencer à faire un choix de fichier ou de sous-répertoire. Il dépend du baladeur sélectionné s'il y en a un et du nom du répertoire de travail. Si on n'arrive pas à déterminer ce répertoire, renvoie None

Définition à la ligne 161 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

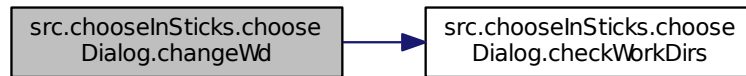


7.8.3.4 def src.chooseInSticks.chooseDialog.changeWd (self)

changement du répertoire de travail

Définition à la ligne 195 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



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

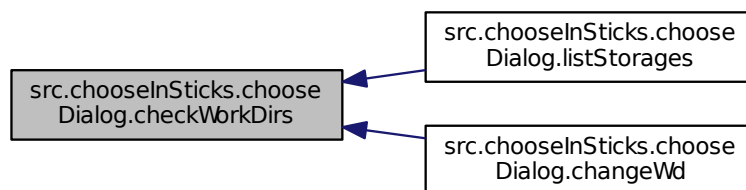
fonction de rappel liée au bouton de validation, vérifie s'il y a bien au moins un fichier ou un répertoire sélectionné
Définition à la ligne 90 du fichier `chooseInSticks.py`.

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

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

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

Voici le graphe des appelants de cette fonction :



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

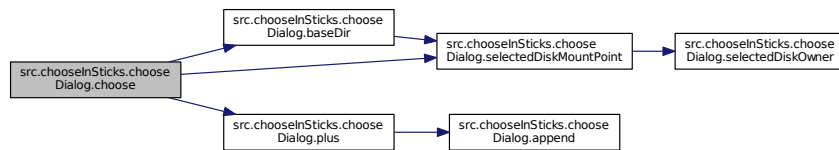
Facilite le choix de motifs de fichiers en recherchant dans les clés USB, modifie l'éditeur de ligne de texte et place le fichier choisi dans la liste.

Paramètres

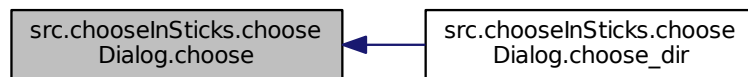
<i>kind</i>	type d'élément à choisir : "file" pour un fichier, "dir" pour un répertoire
-------------	---

Définition à la ligne 208 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

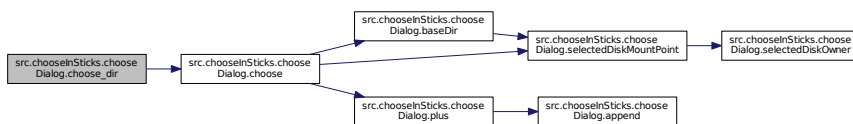


7.8.3.8 def src.chooseInSticks.chooseDialog.choose_dir (self)

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

Définition à la ligne 242 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :

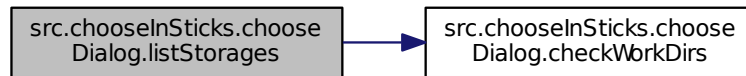


7.8.3.9 def src.chooseInSticks.chooseDialog.listStorages (self)

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

Définition à la ligne 99 du fichier chooseInSticks.py.

Voici le graphe d'appel pour cette fonction :



7.8.3.10 `def src.chooseInSticks.chooseDialog.minus (self)`

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

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

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

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

Renvoie

une liste de chemins, sous forme de `QStrings`

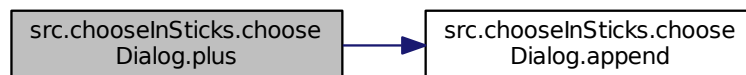
Définition à la ligne 300 du fichier `chooseInSticks.py`.

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

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

Définition à la ligne 258 du fichier `chooseInSticks.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



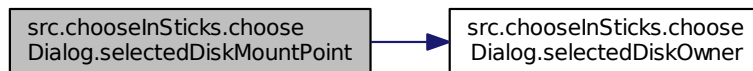
7.8.3.13 `def src.chooseInSticks.chooseDialog.selectedDiskMountPoint (self)`

Renvoie

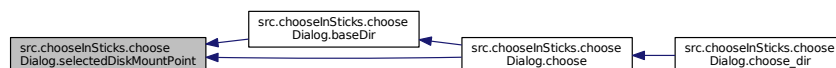
le point de montage du support sélectionné s'il y en a un

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.8.3.14 `def src.chooseInSticks.chooseDialog.selectedDiskOwner (self)`

Renvoie

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

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

Voici le graphe des appelants de cette fonction :



7.8.4 Documentation des données membres

7.8.4.1 `src.chooseInSticks.chooseDialog.mainWindow`

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

7.8.4.2 `src.chooseInSticks.chooseDialog.ok`

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

7.8.4.3 src.chooseInSticks.chooseDialog.okButton

mise en place des titres personnalisés

mise en place du bouton personnalisé

Définition à la ligne 55 du fichier chooseInSticks.py.

7.8.4.4 src.chooseInSticks.chooseDialog.ownedUsbDictionary

peuplement de la zone des noms de baladeurs

Définition à la ligne 71 du fichier chooseInSticks.py.

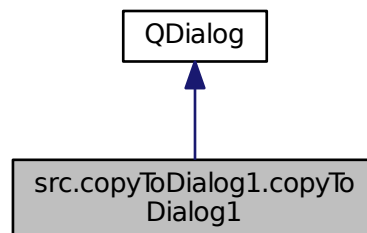
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/chooseInSticks.py](#)

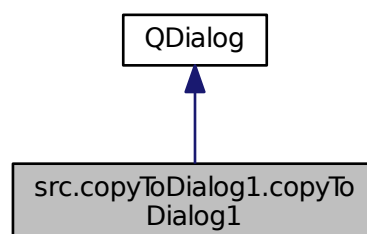
7.9 Référence de la classe src.copyToDialog1.copyToDialog1

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

Graphe d'héritage de src.copyToDialog1.copyToDialog1 :



Graphe de collaboration de src.copyToDialog1.copyToDialog1 :



Fonctions membres publiques

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

Attributs publics

- `mainWindow`
- `ok`

7.9.1 Description détaillée

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

Paramètres

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

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

7.9.2 Documentation des fonctions membres

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

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

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

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

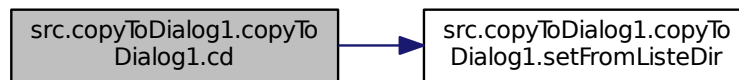
Change le répertoire courant si possible.

Paramètres

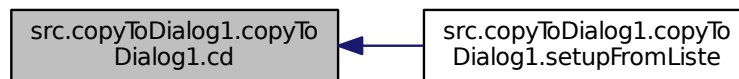
<i>ev</i>	un évènement
-----------	--------------

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



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

changement du répertoire de travail

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

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

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

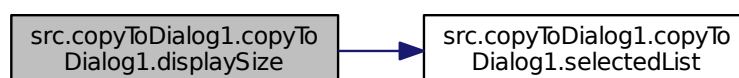
Définition à la ligne 82 du fichier `copyToDialog1.py`.

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

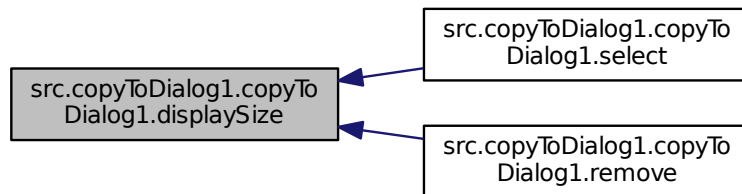
Affiche la taille de la sélection courante.

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

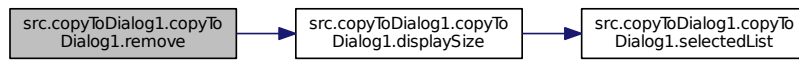


7.9.2.6 `def src.copyToDialog1.copyToDialog1.remove (self)`

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

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

Voici le graphe d'appel pour cette fonction :

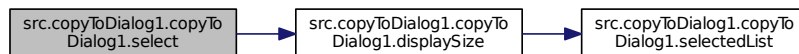


7.9.2.7 `def src.copyToDialog1.copyToDialog1.select (self)`

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

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

Voici le graphe d'appel pour cette fonction :



7.9.2.8 `def src.copyToDialog1.copyToDialog1.selectedList (self)`

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

Renvoie

une liste de QStrings

Définition à la ligne 203 du fichier copyToDialog1.py.

Voici le graphe des appelants de cette fonction :

**7.9.2.9 def src.copyToDialog1.copyToDialog1.setFromListeDir (self, directory)**

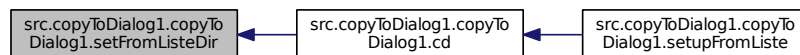
Choisit un répertoire pour la liste source.

Paramètres

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

Définition à la ligne 101 du fichier copyToDialog1.py.

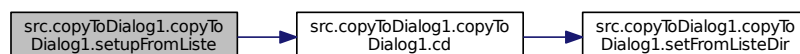
Voici le graphe des appelants de cette fonction :

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

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

Définition à la ligne 90 du fichier copyToDialog1.py.

Voici le graphe d'appel pour cette fonction :

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

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

Définition à la ligne 124 du fichier copyToDialog1.py.

7.9.3 Documentation des données membres

7.9.3.1 `src.copyToDialog1.copyToDialog1.mainWindow`

Définition à la ligne 45 du fichier `copyToDialog1.py`.

7.9.3.2 `src.copyToDialog1.copyToDialog1.ok`

Définition à la ligne 83 du fichier `copyToDialog1.py`.

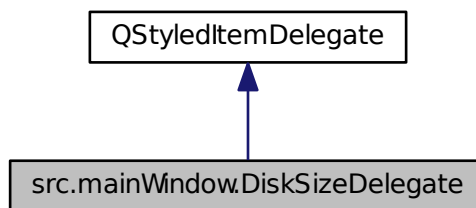
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/copyToDialog1.py](#)

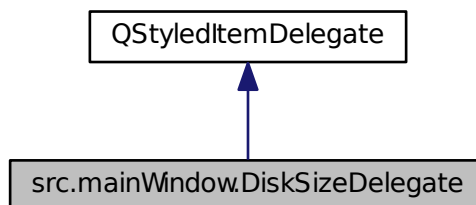
7.10 Référence de la classe `src.mainWindow.DiskSizeDelegate`

Classe pour figurer la taille de la mémoire du baladeur.

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



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



Fonctions membres publiques

- `def __init__(self, parent)`
- `def paint(self, painter, option, index)`
- `def val2txt(self, val)`

7.10.1 Description détaillée

Classe pour figurer la taille de la mémoire du baladeur.

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

Définition à la ligne 939 du fichier `mainWindow.py`.

7.10.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 940 du fichier `mainWindow.py`.

7.10.3 Documentation des fonctions membres

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

Définition à la ligne 944 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



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

Renvoie

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

Définition à la ligne 966 du fichier `mainWindow.py`.

Voici le graphe des appelants de cette fonction :

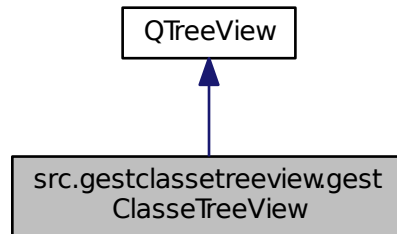


La documentation de cette classe a été générée à partir du fichier suivant :

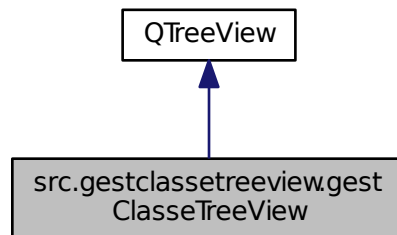
— [src/mainWindow.py](#)

7.11 Référence de la classe src.gestclassetreeview.gestClasseTreeView

Graphe d'héritage de src.gestclassetreeview.gestClasseTreeView :



Graphe de collaboration de src.gestclassetreeview.gestClasseTreeView :



Fonctions membres publiques

- def `__init__`
Le constructeur.
- def `connecteGestionnaire`
- def `expandedItems` (self)
- def `allItems` (self)
- def `checkedItems` (self)

Attributs publics

- `gest`
- `root`

7.11.1 Description détaillée

Définition à la ligne 29 du fichier gestclassetreeview.py.

7.11.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

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

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

7.11.3 Documentation des fonctions membres

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

Renvoie

la liste de tous les élèves

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

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

Renvoie

la liste de tous les élèves sélectionnés

Définition à la ligne 102 du fichier `gestclassetreeview.py`.

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

Paramètres

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

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

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

Renvoie

la liste des items non repliés (donc visibles)

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

7.11.4 Documentation des données membres

7.11.4.1 `src.gestclassetreeview.gestClasseTreeView.gest`

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

7.11.4.2 `src.gestclassetreeview.gestClasseTreeView.root`

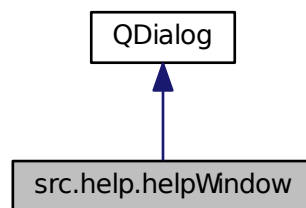
Définition à la ligne 40 du fichier `gestclassetreeview.py`.

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

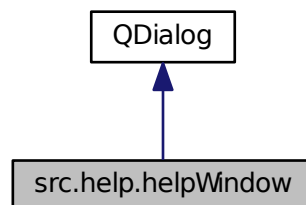
— [src/gestclassetreeview.py](#)

7.12 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` (`self`, `dir`, `locale`)
met en place les textes dans les afficheurs, en fonction de la locale.

Attributs publics

- `ui`

7.12.1 Description détaillée

Définition à la ligne 31 du fichier help.py.

7.12.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Définition à la ligne 36 du fichier help.py.

7.12.3 Documentation des fonctions membres

7.12.3.1 `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 52 du fichier help.py.

7.12.4 Documentation des données membres

7.12.4.1 `src.help.helpWindow.ui`

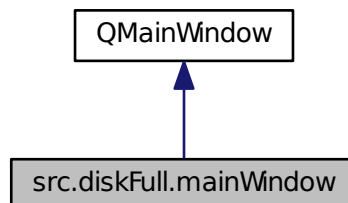
Définition à la ligne 39 du fichier help.py.

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

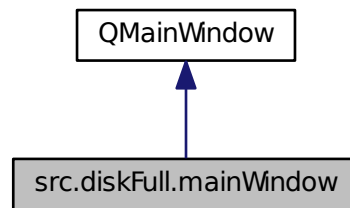
— [src/help.py](#)

7.13 Référence de la classe src.diskFull.mainWindow

Graphe d'héritage de src.diskFull.mainWindow :



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



Fonctions membres publiques

— `def __init__`
Le constructeur.

Attributs publics

— `ui`
 — `v`
 — `total`
 — `used`

7.13.1 Description détaillée

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

7.13.2 Documentation des constructeurs et destructeur

7.13.2.1 `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`.

7.13.3 Documentation des données membres

7.13.3.1 `src.diskFull.mainWindow.total`

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

7.13.3.2 `src.diskFull.mainWindow.ui`

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

7.13.3.3 `src.diskFull.mainWindow.used`

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

7.13.3.4 `src.diskFull.mainWindow.v`

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

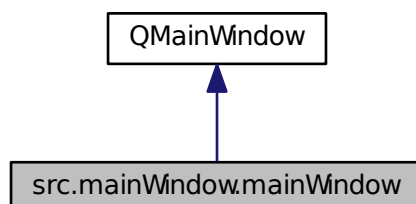
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/diskFull.py](#)

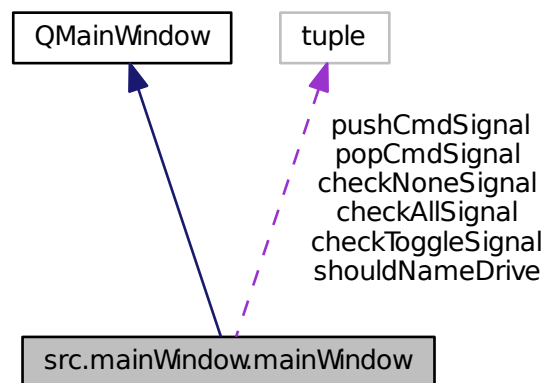
7.14 Référence de la classe `src.mainWindow.mainWindow`

defines the main window of the application.

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



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



Fonctions membres publiques

- def `__init__`
Le constructeur.
- def `setThemedIcon`
Associe une icone à un bouton, dans le thème courant.
- def `pushCmd` (self, owner, cmd)
fonction de rappel déclenchée par les threads (au commencement)
- def `popCmd` (self, owner, cmd)
fonction de rappel déclenchée par les threads (à la fin)
- def `checkModify` (self, boolFunc)
- def `checkAll` (self)
Coche tous les baladeurs.
- def `checkToggle` (self)
Inverse la coche des baladeurs.
- def `checkNone` (self)
Décoche tous les baladeurs.
- def `namingADrive` (self)
Gère un dialogue pour renommer un baladeur désigné par `self.recentConnect`.
- def `cbAdded` (self)
Renvoie une fonction de rappel pour l'abonnement aux événements de l'arrière-boutique.
- def `cbRemoved` (self)
Renvoie une fonction de rappel pour l'abonnement aux événements de l'arrière-boutique.
- def `deviceAdded` (self)
Fonction de rappel pour un medium ajouté ; se base sur la valeur de `self.recentConnect`.
- def `deviceRemoved` (self)
fonction de rappel pour un medium retiré ; se base sur la valeur de `self.recentDisconnect`
- def `initRedoStuff` (self)
Initialise des données pour le bouton central (refaire/stopper)
- def `applyPreferences` (self)
Applique les préférences et les options de ligne de commande.
- def `findAllDisks`
Initialisation du catalogue des disques USB connectés, et maintenance de l'interface graphique.
- def `changeWd` (self, newDir)
change le répertoire par défaut contenant les fichiers de travail
- def `tableClicked` (self, idx)
fonction de rappel pour un double clic sur un élément de la table
- def `manageCheckBoxes` (self)
ouvre un dialogue pour permettre de gérer les cases à cocher globalement
- def `diskSizeData` (self, rowOrDev)
- def `diskFromOwner` (self, student)
trouve le disque qui correspond à un propriétaire, ou alors renvoie le premier disque inconnu.

- def `editOwner` (self, idx)
Édition du propriétaire d'une clé.
- def `setAvailableNames` (self, available)
Met à jour l'icône qui reflète la disponibilité de noms pour renommer automatiquement des baladeurs.
- def `updateButtons` (self)
Désactive ou active les flèches selon que l'option correspondante est possible ou non.
- def `preference` (self)
lance le dialogue des préférences
- def `delFiles` (self)
Lance l'action de supprimer des fichiers ou des répertoires dans les clés USB.
- def `copyTo` (self)
Lance l'action de copier vers les clés USB.
- def `copyFrom` (self)
Lance l'action de copier depuis les clés USB.
- def `redoCmd` (self)
Relance la dernière commande, mais en l'appliquant seulement aux baladeurs nouvellement branchés.
- def `namesCmd` (self)
montre le dialogue de choix de nouveaux noms à partir d'un fichier administratif.
- def `help` (self)
Affiche le widget d'aide.
- def `umount` (self)
Démonte et détache les clés USB affichées.
- def `connectTableModel` (self, data)
Connecte le modèle de table à la table.
- def `sameDiskData` (self, one, two)

Attributs publics

- `locale`
- `ui`
- `copyfromIcon`
- `movefromIcon`
- `namesFullIcon`
- `namesEmptyIcon`
- `namesFullTip`
- `namesEmptyTip`
- `namesDialog`
- `recentConnect`
- `t`
- `proxy`
- `operations`
- `oldThreads`
- `recentDisConnect`
- `iconRedo`
- `iconStop`
- `redoToolTip`
- `redoStatusTip`
- `stopToolTip`
- `stopStatusTip`
- `schoolFile`
- `workdir`
- `manFileLocation`
- `mv`
- `header`
- `availableNames`
- `visibleheader`
- `tm`

Attributs publics statiques

- tuple `checkAllSignal` = `pyqtSignal()`
custom signals #####
- tuple `checkToggleSignal` = `pyqtSignal()`
- tuple `checkNoneSignal` = `pyqtSignal()`
- tuple `shouldNameDrive` = `pyqtSignal()`
- tuple `pushCmdSignal` = `pyqtSignal(str, str)`
- tuple `popCmdSignal` = `pyqtSignal(str, str)`

7.14.1 Description détaillée

defines the main window of the application.

Définition à la ligne 66 du fichier mainWindow.py.

7.14.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

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

Définition à la ligne 81 du fichier mainWindow.py.

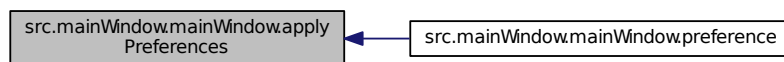
7.14.3 Documentation des fonctions membres

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

Applique les préférences et les options de ligne de commande.

Définition à la ligne 324 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



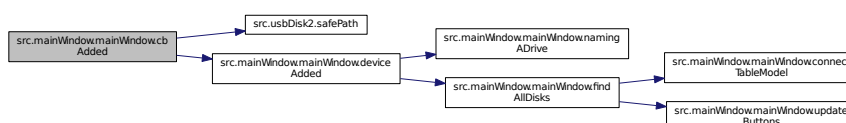
7.14.3.2 `def src.mainWindow.mainWindow.cbAdded (self)`

Renvoie une fonction de rappel pour l'abonnement aux évènements de l'arrière-boutique.

Il s'agit de la fonction pour les disques branchés

Définition à la ligne 259 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



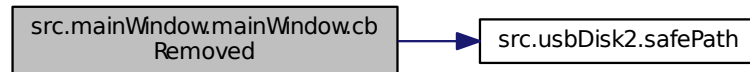
7.14.3.3 `def src.mainWindow.mainWindow.cbRemoved (self)`

Renvoie une fonction de rappel pour l'abonnement aux évènements de l'arrière-boutique.

Il s'agit de la fonction pour les disques débranchés

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

Voici le graphe d'appel pour cette fonction :

7.14.3.4 `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
---------------	------------------------------

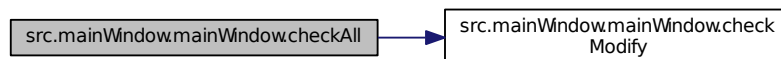
Définition à la ligne 357 du fichier `mainWindow.py`.

7.14.3.5 `def src.mainWindow.mainWindow.checkAll (self)`

Coche tous les baladeurs.

Définition à la ligne 215 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :

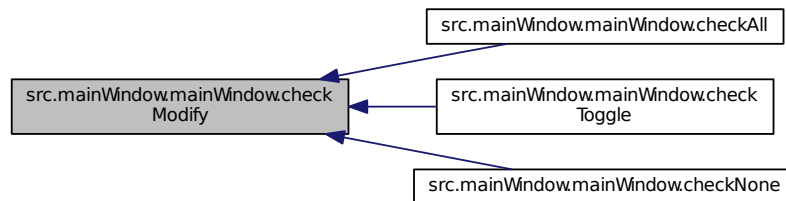
7.14.3.6 `def src.mainWindow.mainWindow.checkModify (self, boolFunc)`

Paramètres

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

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

Voici le graphe des appelants de cette fonction :

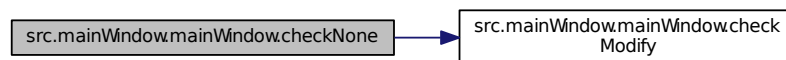


7.14.3.7 def src.mainWindow.mainWindow.checkNone (self)

Décoche tous les baladeurs.

Définition à la ligne 229 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

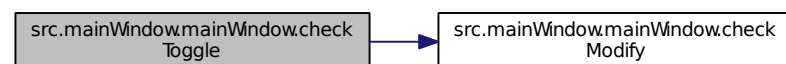


7.14.3.8 def src.mainWindow.mainWindow.checkToggle (self)

Inverse la coche des baladeurs.

Définition à la ligne 222 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



7.14.3.9 def src.mainWindow.mainWindow.connectTableModel (self, data)

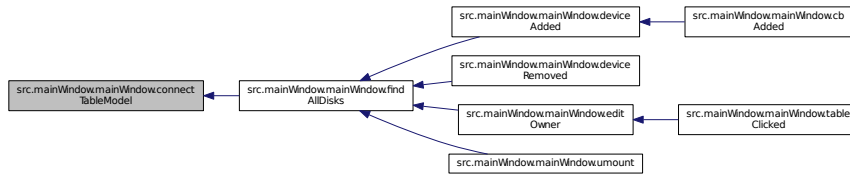
Connecte le modèle de table à la table.

Paramètres

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

Définition à la ligne 728 du fichier `mainWindow.py`.

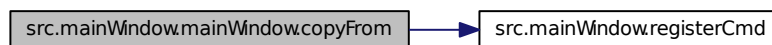
Voici le graphe des appelants de cette fonction :

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

Lance l'action de copier depuis les clés USB.

Définition à la ligne 590 du fichier `mainWindow.py`.

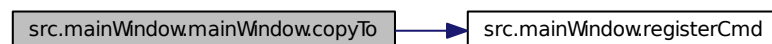
Voici le graphe d'appel pour cette fonction :

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

Lance l'action de copier vers les clés USB.

Définition à la ligne 565 du fichier `mainWindow.py`.

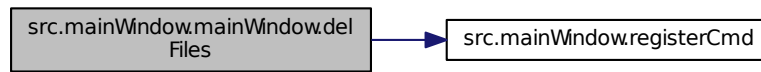
Voici le graphe d'appel pour cette fonction :

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

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

Définition à la ligne 530 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :

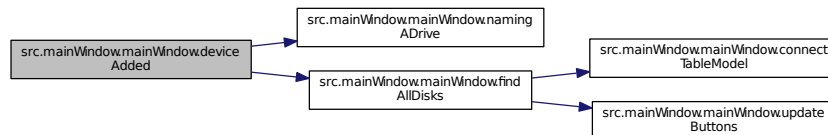


7.14.3.13 `def src.mainWindow.mainWindow.deviceAdded (self)`

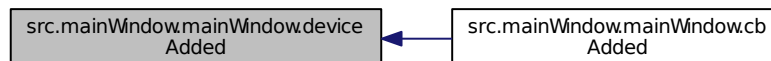
Fonction de rappel pour un medium ajouté ; se base sur la valeur de `self.recentConnect`.

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.14.3.14 `def src.mainWindow.mainWindow.deviceRemoved (self)`

fonction de rappel pour un medium retiré ; se base sur la valeur de `self.recentDisConnect`

Définition à la ligne 301 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



7.14.3.15 `def src.mainWindow.mainWindow.diskFromOwner (self, student)`

trouve le disque qui correspond à un propriétaire, ou alors renvoie le premier disque inconnu.

Paramètres

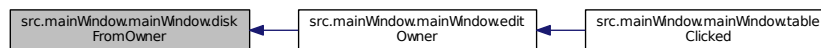
<i>student</i>	le propriétaire du disque
----------------	---------------------------

Renvoie

le disque correspondant à l'étudiant

Définition à la ligne 425 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

7.14.3.16 def src.mainWindow.mainWindow.diskSizeData (*self*, *rowOrDev*)

Paramètres

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

Renvoie

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

Définition à la ligne 406 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :

7.14.3.17 def src.mainWindow.mainWindow.editOwner (*self*, *idx*)

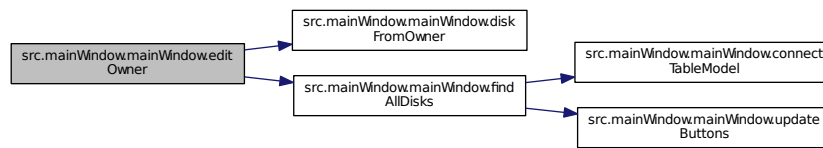
Édition du propriétaire d'une clé.

Paramètres

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

Définition à la ligne 442 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.14.3.18 `def src.mainWindow.mainWindow.findAllDisks (self, other=None)`

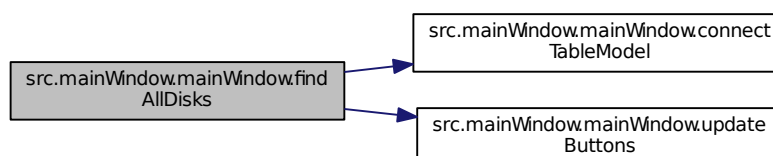
Initialisation du catalogue des disques USB connectés, et maintenance de l'interface graphique.

Paramètres

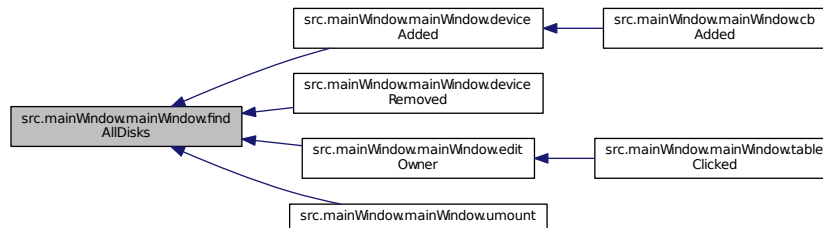
<i>other</i>	un catalogue déjà tout prêt de disques (None par défaut)
--------------	--

Définition à la ligne 340 du fichier `mainWindow.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.14.3.19 def src.mainWindow.mainWindow.help (self)

Affiche le widget d'aide.

Définition à la ligne 693 du fichier mainWindow.py.

7.14.3.20 def src.mainWindow.mainWindow.initRedoStuff (self)

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

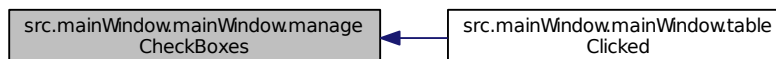
Définition à la ligne 308 du fichier mainWindow.py.

7.14.3.21 def src.mainWindow.mainWindow.manageCheckBoxes (self)

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

Définition à la ligne 395 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



7.14.3.22 def src.mainWindow.mainWindow.namesCmd (self)

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

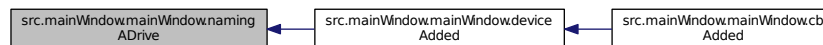
Définition à la ligne 686 du fichier mainWindow.py.

7.14.3.23 def src.mainWindow.mainWindow.namingADrive (self)

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

Définition à la ligne 237 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



7.14.3.24 def src.mainWindow.mainWindow.popCmd (self, owner, cmd)

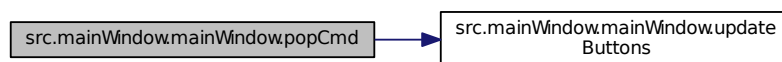
fonction de rappel déclenchée par les threads (à la fin)

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 177 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :

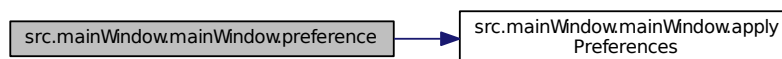


7.14.3.25 def src.mainWindow.mainWindow.preference (self)

lance le dialogue des préférences

Définition à la ligne 516 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



7.14.3.26 def src.mainWindow.mainWindow.pushCmd (self, owner, cmd)

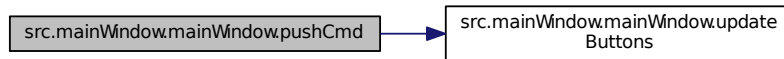
fonction de rappel déclenchée par les threads (au commencement)

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 162 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



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

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

Définition à la ligne 655 du fichier mainWindow.py.

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

Renvoie

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

Définition à la ligne 747 du fichier mainWindow.py.

7.14.3.29 `def src.mainWindow.mainWindow.setAvailableNames (self, available)`

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

Paramètres

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

Définition à la ligne 456 du fichier mainWindow.py.

7.14.3.30 `def src.mainWindow.mainWindow.setThemedIcon (self, button, name, default = None)`

Associe une icône à un bouton, dans le thème courant.

Paramètres

<i>button</i>	le bouton à décorer
<i>name</i>	le nom de l'icône
<i>default</i>	un fichier PNG ; si rien n'est donné, il aura comme valeur par défaut "images/icons32/"+name+".png"

Renvoie

l'objet de type QIcon qui a été associé au bouton

Définition à la ligne 147 du fichier mainWindow.py.

7.14.3.31 `def src.mainWindow.mainWindow.tableClicked (self, idx)`

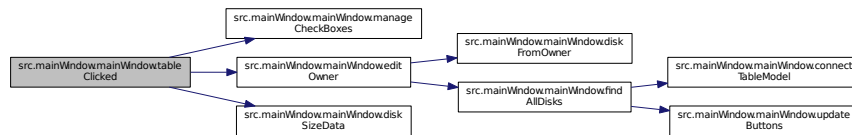
fonction de rappel pour un double clic sur un élément de la table

Paramètres

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

Définition à la ligne 366 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



7.14.3.32 def src.mainWindow.mainWindow.umount (self)

Démonte et détache les clés USB affichées.

Définition à la ligne 702 du fichier mainWindow.py.

Voici le graphe d'appel pour cette fonction :



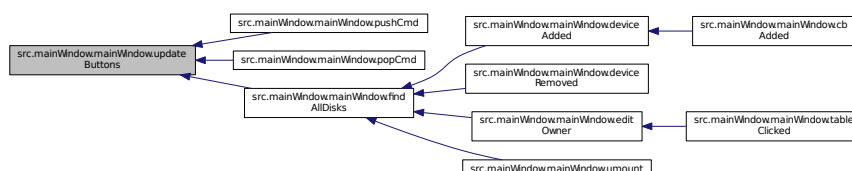
7.14.3.33 def src.mainWindow.mainWindow.updateButtons (self)

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

Pour les flèches : ça aurait du sens de préparer une opération de copie avant même de brancher des clés, donc on les active. Par contre démonter les clés quand elles sont absentes ça n'a pas d'utilité. Change l'icône du dialogue des noms selon qu'il reste ou non des noms disponibles dans le dialogue des noms.

Définition à la ligne 478 du fichier mainWindow.py.

Voici le graphe des appelants de cette fonction :



7.14.4 Documentation des données membres

7.14.4.1 `src.mainWindow.mainWindow.availableNames`

Définition à la ligne 457 du fichier `mainWindow.py`.

7.14.4.2 `tuple src.mainWindow.mainWindow.checkAllSignal = pyqtSignal()` `[static]`

custom signals #####

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

7.14.4.3 `tuple src.mainWindow.mainWindow.checkNoneSignal = pyqtSignal()` `[static]`

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

7.14.4.4 `tuple src.mainWindow.mainWindow.checkToggleSignal = pyqtSignal()` `[static]`

Définition à la ligne 69 du fichier `mainWindow.py`.

7.14.4.5 `src.mainWindow.mainWindow.copyfromIcon`

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

7.14.4.6 `src.mainWindow.mainWindow.header`

Définition à la ligne 330 du fichier `mainWindow.py`.

7.14.4.7 `src.mainWindow.mainWindow.iconRedo`

Définition à la ligne 310 du fichier `mainWindow.py`.

7.14.4.8 `src.mainWindow.mainWindow.iconStop`

Définition à la ligne 312 du fichier `mainWindow.py`.

7.14.4.9 `src.mainWindow.mainWindow.locale`

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

7.14.4.10 `src.mainWindow.mainWindow.manFileLocation`

Définition à la ligne 328 du fichier `mainWindow.py`.

7.14.4.11 `src.mainWindow.mainWindow.movefromIcon`

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

7.14.4.12 `src.mainWindow.mainWindow.mv`

Définition à la ligne 329 du fichier `mainWindow.py`.

7.14.4.13 `src.mainWindow.mainWindow.namesDialog`

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

7.14.4.14 `src.mainWindow.mainWindow.namesEmptyIcon`

Définition à la ligne 102 du fichier `mainWindow.py`.

7.14.4.15 `src.mainWindow.mainWindow.namesEmptyTip`

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

7.14.4.16 `src.mainWindow.mainWindow.namesFullIcon`

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

7.14.4.17 `src.mainWindow.mainWindow.namesFullTip`

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

7.14.4.18 `src.mainWindow.mainWindow.oldThreads`

Définition à la ligne 117 du fichier `mainWindow.py`.

7.14.4.19 `src.mainWindow.mainWindow.operations`

Définition à la ligne 116 du fichier `mainWindow.py`.

7.14.4.20 `tuple src.mainWindow.mainWindow.popCmdSignal = pyqtSignal(str, str) [static]`

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

7.14.4.21 `src.mainWindow.mainWindow.proxy`

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

7.14.4.22 `tuple src.mainWindow.mainWindow.pushCmdSignal = pyqtSignal(str, str) [static]`

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

7.14.4.23 `src.mainWindow.mainWindow.recentConnect`

Définition à la ligne 106 du fichier `mainWindow.py`.

7.14.4.24 `src.mainWindow.mainWindow.recentDisconnect`

Définition à la ligne 279 du fichier `mainWindow.py`.

7.14.4.25 `src.mainWindow.mainWindow.redoStatusTip`

Définition à la ligne 316 du fichier `mainWindow.py`.

7.14.4.26 `src.mainWindow.mainWindow.redoToolTip`

Définition à la ligne 315 du fichier `mainWindow.py`.

7.14.4.27 `src.mainWindow.mainWindow.schoolFile`

Définition à la ligne 326 du fichier `mainWindow.py`.

7.14.4.28 `tuple src.mainWindow.mainWindow.shouldNameDrive = pyqtSignal()` `[static]`

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

7.14.4.29 `src.mainWindow.mainWindow.stopStatusTip`

Définition à la ligne 318 du fichier `mainWindow.py`.

7.14.4.30 `src.mainWindow.mainWindow.stopToolTip`

Définition à la ligne 317 du fichier `mainWindow.py`.

7.14.4.31 `src.mainWindow.mainWindow.t`

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

7.14.4.32 `src.mainWindow.mainWindow.tm`

Définition à la ligne 735 du fichier `mainWindow.py`.

7.14.4.33 `src.mainWindow.mainWindow.ui`

Définition à la ligne 86 du fichier `mainWindow.py`.

7.14.4.34 `src.mainWindow.mainWindow.visibleheader`

Définition à la ligne 729 du fichier `mainWindow.py`.

7.14.4.35 `src.mainWindow.mainWindow.workdir`

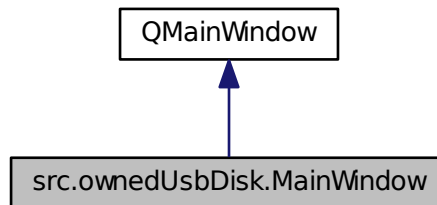
Définition à la ligne 327 du fichier `mainWindow.py`.

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

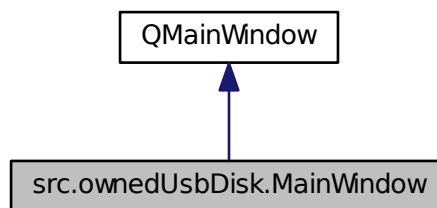
— [src/mainWindow.py](#)

7.15 Référence de la classe src.ownedUsbDisk.MainWindow

Graphe d'héritage de src.ownedUsbDisk.MainWindow :



Graphe de collaboration de src.ownedUsbDisk.MainWindow :



Fonctions membres publiques

— `def __init__(self)`

7.15.1 Description détaillée

Définition à la ligne 321 du fichier ownedUsbDisk.py.

7.15.2 Documentation des constructeurs et destructeur

7.15.2.1 `def src.ownedUsbDisk.MainWindow.__init__(self)`

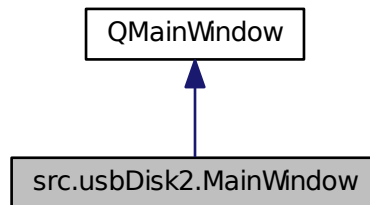
Définition à la ligne 322 du fichier ownedUsbDisk.py.

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

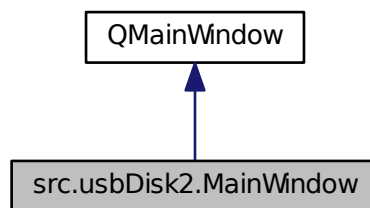
— `src/ownedUsbDisk.py`

7.16 Référence de la classe `src.usbDisk2.MainWindow`

Graphe d'héritage de `src.usbDisk2.MainWindow` :



Graphe de collaboration de `src.usbDisk2.MainWindow` :



Fonctions membres publiques

— `def __init__(self)`

7.16.1 Description détaillée

Définition à la ligne 790 du fichier `usbDisk2.py`.

7.16.2 Documentation des constructeurs et destructeur

7.16.2.1 `def src.usbDisk2.MainWindow.__init__(self)`

Définition à la ligne 791 du fichier `usbDisk2.py`.

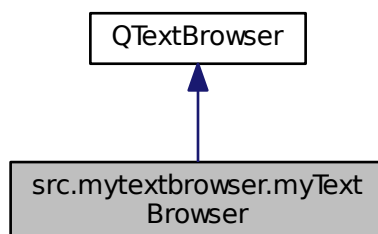
La documentation de cette classe a été générée à partir du fichier suivant :

— `src/usbDisk2.py`

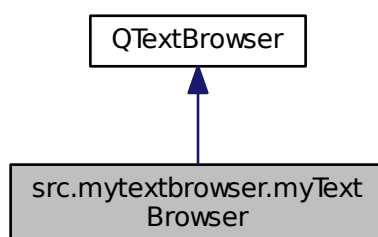
7.17 Référence de la classe src.mytextbrowser.myTextBrowser

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

Graphe d'héritage de src.mytextbrowser.myTextBrowser :



Graphe de collaboration de src.mytextbrowser.myTextBrowser :



Fonctions membres publiques

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

7.17.1 Description détaillée

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

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

7.17.2 Documentation des fonctions membres

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

lien vers la méthode `setSource` originale

Paramètres

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

Définition à la ligne 47 du fichier mytextbrowser.py.

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

lance Firefox en tâche de fond.

Paramètres

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

Définition à la ligne 39 du fichier mytextbrowser.py.

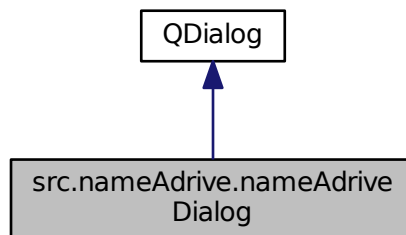
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mytextbrowser.py](#)

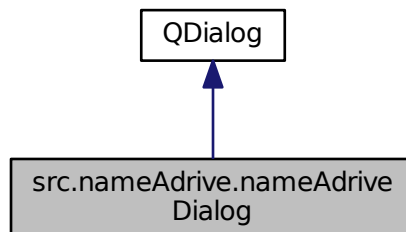
7.18 Référence de la classe src.nameAdrive.nameAdriveDialog

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

Graphe d'héritage de src.nameAdrive.nameAdriveDialog :



Graphe de collaboration de src.nameAdrive.nameAdriveDialog :



Fonctions membres publiques

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

Attributs publics

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

7.18.1 Description détaillée

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

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

7.18.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

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

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

7.18.3 Documentation des fonctions membres

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

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

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

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

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

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

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

fonction de rappel quand l'utilisateur valide le choix

Définition à la ligne 99 du fichier nameAdrive.py.

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

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

Définition à la ligne 88 du fichier nameAdrive.py.

7.18.4 Documentation des données membres

7.18.4.1 `src.nameAdrive.nameAdriveDialog.nameList`

Définition à la ligne 48 du fichier nameAdrive.py.

7.18.4.2 `src.nameAdrive.nameAdriveDialog.numPattern`

Définition à la ligne 56 du fichier nameAdrive.py.

7.18.4.3 `src.nameAdrive.nameAdriveDialog.oldName`

Définition à la ligne 47 du fichier nameAdrive.py.

7.18.4.4 `src.nameAdrive.nameAdriveDialog.tattoo`

Définition à la ligne 50 du fichier nameAdrive.py.

7.18.4.5 `src.nameAdrive.nameAdriveDialog.ui`

Définition à la ligne 51 du fichier nameAdrive.py.

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

— [src/nameAdrive.py](#)

7.19 Référence de la classe `src.notification.Notification`

Une classe pour afficher des notifications à l'écran.

Fonctions membres publiques

- `def __init__`
Le constructeur.
- `def notify (self)`

Attributs publics

- `app_name`
- `replaces_id`
- `app_icon`

- [summary](#)
- [body](#)
- [actions](#)
- [hints](#)
- [expire_timeout](#)
- [interface](#)

7.19.1 Description détaillée

Une classe pour afficher des notifications à l'écran.

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

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

7.19.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

<i>app_name</i>	nom d'une application, valeur par défaut = ""
<i>replaces_id</i>	identifiant d'une notification à remplacer valeur par défaut=0
<i>app_icon</i>	nom d'un fichier servant pour l'icône valeur par défaut=""
<i>summary</i>	description brève de la notification valeur par défaut = ""
<i>body</i>	le texte de la notification, valeur pa défaut=""
<i>actions</i>	une liste de paires représeantant des actions, valeur par défaut=[]
<i>hints</i>	un dictionnaire de suggestions, valeur par défaut={},
<i>expire_timeout</i>	durée maximale d'affichage en millisecondes, valeur par défaut=1000

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

7.19.3 Documentation des fonctions membres

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

Définition à la ligne 69 du fichier `notification.py`.

7.19.4 Documentation des données membres

7.19.4.1 `src.notification.Notification.actions`

Définition à la ligne 58 du fichier `notification.py`.

7.19.4.2 `src.notification.Notification.app_icon`

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

7.19.4.3 `src.notification.Notification.app_name`

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

7.19.4.4 `src.notification.Notification.body`

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

7.19.4.5 `src.notification.Notification.expire_timeout`

Définition à la ligne 60 du fichier `notification.py`.

7.19.4.6 `src.notification.Notification.hints`

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

7.19.4.7 `src.notification.Notification.interface`

Définition à la ligne 65 du fichier `notification.py`.

7.19.4.8 `src.notification.Notification.replaces_id`

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

7.19.4.9 `src.notification.Notification.summary`

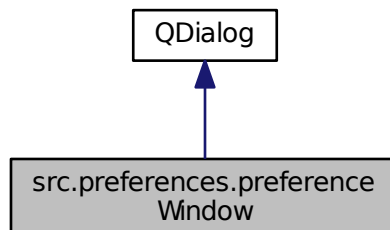
Définition à la ligne 56 du fichier `notification.py`.

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

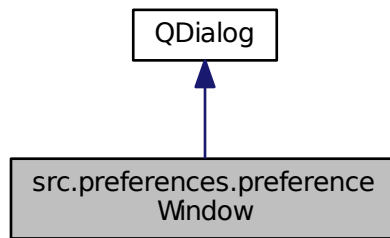
— [src/notification.py](#)

7.20 Référence de la classe `src.preferences.preferenceWindow`

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



Graphe de collaboration de src.preferences.preferenceWindow :



Fonctions membres publiques

- def `__init__`
Le constructeur.
- def `enableDelay` (self, state)
active ou désactive le glisseur pour modifier le délai de rafraichissement
- def `updateRefreshLabel` (self, val)
Met à jour l'affichage de la valeur du délai de rafraichissement.
- def `values` (self)
- def `setValues` (self, prefs)
Met en place les préférences dans le dialogue.

Attributs publics

- `ui`

7.20.1 Description détaillée

Définition à la ligne 28 du fichier preferences.py.

7.20.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Définition à la ligne 33 du fichier preferences.py.

7.20.3 Documentation des fonctions membres

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

active ou désactive le glisseur pour modifier le délai de rafraichissement

Paramètres

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

Définition à la ligne 44 du fichier preferences.py.

7.20.3.2 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 76 du fichier preferences.py.

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

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

Paramètres

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

Définition à la ligne 52 du fichier preferences.py.

7.20.3.4 def src.preferences.preferenceWindow.values (self)**Renvoie**

un dictionnaire de préférences

Définition à la ligne 62 du fichier preferences.py.

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

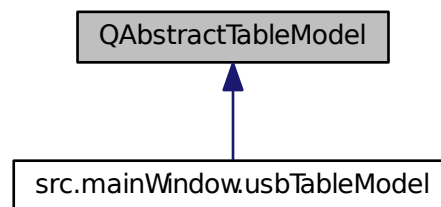
Définition à la ligne 36 du fichier preferences.py.

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

— [src/preferences.py](#)

7.21 Référence de la classe QAbstractTableModel

Graphe d'héritage de QAbstractTableModel :

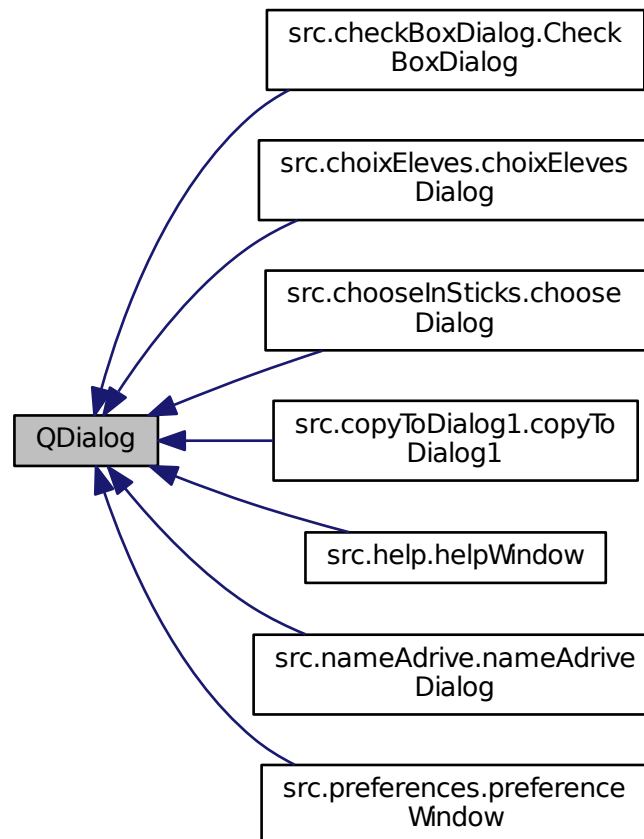


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

7.22 Référence de la classe QDialog

Graphe d'héritage de QDialog :

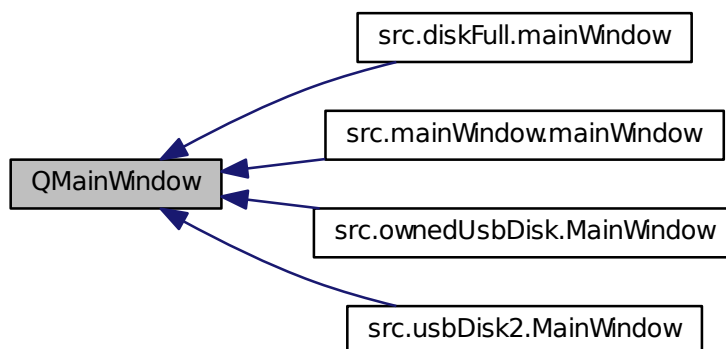


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/copyToDialog1.py](#)

7.23 Référence de la classe QMainWindow

Graphe d'héritage de QMainWindow :

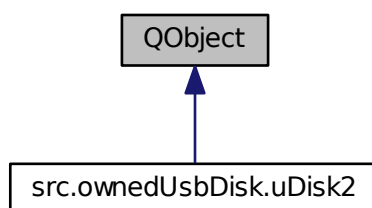


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/ownedUsbDisk.py](#)

7.24 Référence de la classe QObject

Graphe d'héritage de QObject :

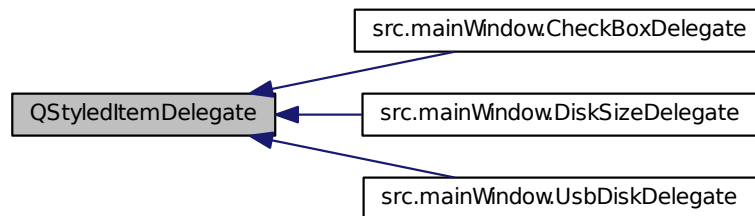


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/ownedUsbDisk.py](#)

7.25 Référence de la classe QStyledItemDelegate

Graphe d'héritage de QStyledItemDelegate :

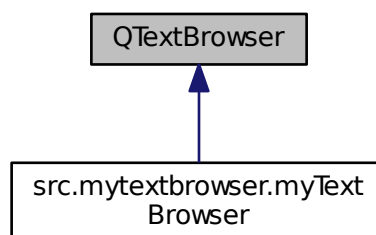


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

7.26 Référence de la classe QTextBrowser

Graphe d'héritage de QTextBrowser :

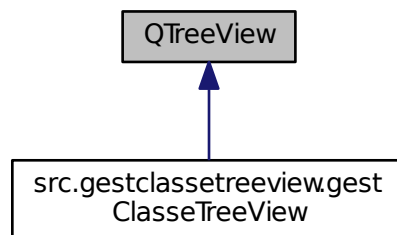


La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mytextbrowser.py](#)

7.27 Référence de la classe QTreeView

Graphe d'héritage de QTreeView :



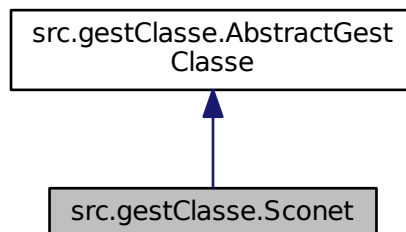
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/gestclassetreeview.py](#)

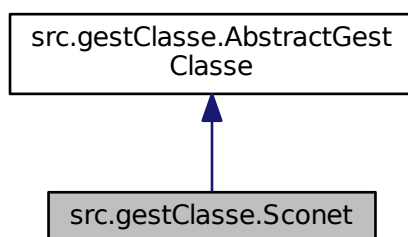
7.28 Référence de la classe src.gestClasse.Sconet

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

Graphe d'héritage de src.gestClasse.Sconet :



Graphe de collaboration de `src.gestClasse.Sconet` :



Fonctions membres publiques

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

Attributs publics

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

7.28.1 Description détaillée

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

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

7.28.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

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

Définition à la ligne 86 du fichier `gestClasse.py`.

7.28.3 Documentation des fonctions membres

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

Définition à la ligne 215 du fichier `gestClasse.py`.

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

Renvoie

the list of classes containg students

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

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

Définition à la ligne 109 du fichier `gestClasse.py`.

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

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

Paramètres

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

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

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

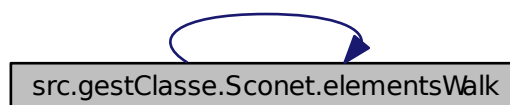
implemente un parcour des éléments d'un arbre, pour y appliquer une procédure

Paramètres

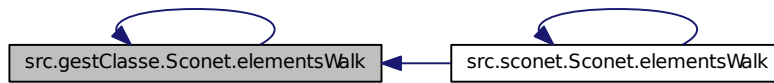
<i>el</i>	un élément
<i>proc</i>	la procédure à appliquer (paramètres : l'élément)

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.28.3.6 def src.gestClasse.Sconet.eleveParID (self, el)

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

Définition à la ligne 146 du fichier gestClasse.py.

7.28.3.7 def src.gestClasse.Sconet.elevsDeClasse (self, className)

Paramètres

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

Renvoie

list of "eleve" elements

Définition à la ligne 129 du fichier gestClasse.py.

7.28.3.8 def src.gestClasse.Sconet.makeCompact (self)

removes useless thext nodes containing only spaces.

Définition à la ligne 102 du fichier gestClasse.py.

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

Paramètres

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

Renvoie

une chaîne unicode, pour nommer l'élève

Définition à la ligne 196 du fichier gestClasse.py.

7.28.3.10 def src.gestClasse.Sconet.unIDEleveDeClasse (self, el)

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

Paramètres

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

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

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

a unique name for an "eleve", based on a few fields and on the ID

Paramètres

<i>el</i>	en "eleve" element
<i>fields</i>	the fields used to build the result a printable unique id

Définition à la ligne 183 du fichier `gestClasse.py`.

7.28.4 Documentation des données membres

7.28.4.1 `src.gestClasse.Sconet.classes`

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

7.28.4.2 `src.gestClasse.Sconet.currentClassName`

Définition à la ligne 131 du fichier `gestClasse.py`.

7.28.4.3 `src.gestClasse.Sconet.currentID`

Définition à la ligne 136 du fichier `gestClasse.py`.

7.28.4.4 `src.gestClasse.Sconet.currentResult`

Définition à la ligne 130 du fichier `gestClasse.py`.

7.28.4.5 `src.gestClasse.Sconet.donnees`

Définition à la ligne 95 du fichier `gestClasse.py`.

7.28.4.6 `src.gestClasse.Sconet.nullTexts`

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

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

— [src/gestClasse.py](#)

7.29 Référence de la classe `src.sconet.Sconet`

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

Fonctions membres publiques

- def `__init__` (self, file)
Le constructeur.
- def `makeCompact` (self)
removes useless text nodes containing only spaces.
- def `collectNullTexts` (self, el)
- def `collectClasses` (self)
- def `collectOneClass` (self, el)
- def `elementsWalk` (self, el, proc)
implemente un parcours des éléments d'un arbre, pour y appliquer une procédure
- def `__str__` (self)

Attributs publics

- `donnees`
- `nullTexts`
- `classes`

7.29.1 Description détaillée

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

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

7.29.2 Documentation des constructeurs et destructeur

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

Le constructeur.

Paramètres

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

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

7.29.3 Documentation des fonctions membres

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

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

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

Renvoie

the list of classes containg students

Définition à la ligne 69 du fichier `sconet.py`.

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

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

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

Renvoie

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

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

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

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

Paramètres

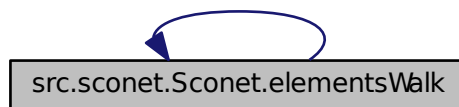
<i>el</i>	un élément
<i>proc</i>	la procédure à appliquer (paramètres : l'élément)

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

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

**7.29.3.6** `def src.sconet.Sconet.makeCompact (self)`

removes useless thext nodes containing only spaces.

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

7.29.4 Documentation des données membres**7.29.4.1** `src.sconet.Sconet.classes`

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

7.29.4.2 `src.sconet.Sconet.donnees`

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

7.29.4.3 `src.sconet.Sconet.nullTexts`

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

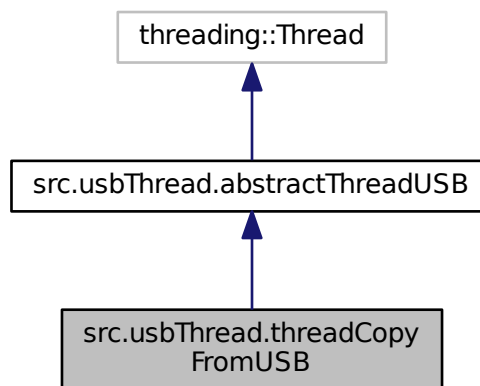
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/sconet.py](#)

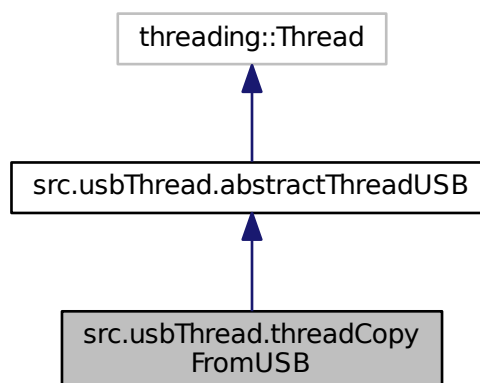
7.30 Référence de la classe `src.usbThread.threadCopyFromUSB`

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` (`self`, `ud`, `fileList`, `subdir`, `dest`, `logfile`)
Copie une liste de fichiers d'une clé USB sous un répertoire donné.

Attributs publics

- `rootPath`

7.30.1 Description détaillée

Classe pour les threads copiant depuis les clés USB.

Définition à la ligne 358 du fichier `usbThread.py`.

7.30.2 Documentation des constructeurs et destructeur

7.30.2.1 `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 373 du fichier `usbThread.py`.

7.30.3 Documentation des fonctions membres

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

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

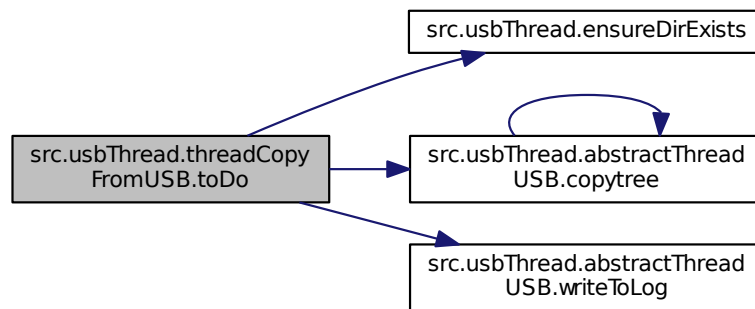
À chaque fichier ou répertoire copié, une ligne est journalisée dans le fichier de journal de l'application.

Paramètres

<code>ud</code>	l'instance <code>uDisk</code> correspondant à une partition de clé USB
<code>fileList</code>	la liste des fichiers à copier, qui peut contenir des jokers
<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

Définition à la ligne 389 du fichier `usbThread.py`.

Voici le graphe d'appel pour cette fonction :



7.30.4 Documentation des données membres

7.30.4.1 `src.usbThread.threadCopyFromUSB.rootPath`

Définition à la ligne 376 du fichier `usbThread.py`.

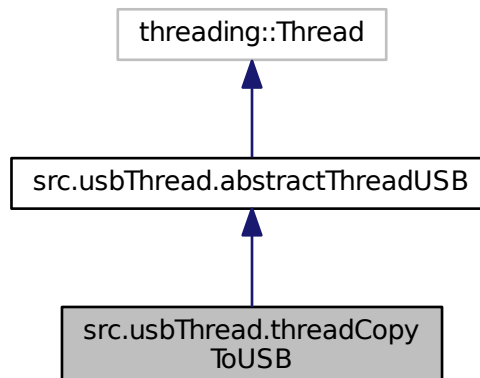
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

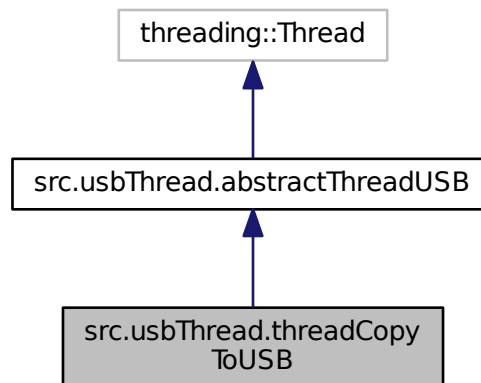
7.31 Référence de la classe `src.usbThread.threadCopyToUSB`

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` (self)
- def `todo` (self, `ud`, `fileList`, `subdir`, `dest`, `logfile`)
Copie une liste de fichiers vers une clé USB sous un répertoire donné.

Membres hérités additionnels

7.31.1 Description détaillée

Classe pour les threads copiant vers les clés USB.

Définition à la ligne 286 du fichier usbThread.py.

7.31.2 Documentation des constructeurs et destructeur

7.31.2.1 `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 uDisk 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, /dev/null par défaut
<code>parent</code>	un widget qui recevra de signaux en début et en fin d'exécution

Définition à la ligne 299 du fichier usbThread.py.

7.31.3 Documentation des fonctions membres

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

Renvoie

une chaîne courte qui informe sur le type de thread

Définition à la ligne 306 du fichier `usbThread.py`.

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

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

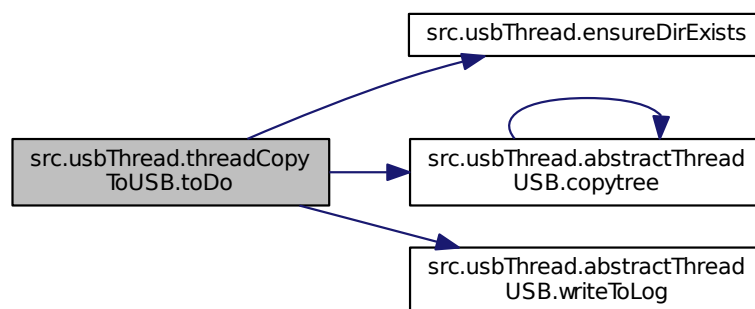
Ce répertoire est composé de `ud.visibleDir()` joint au sous-répertoire `subdir`. À chaque fichier ou répertoire copié, une ligne est journalisée dans le fichier de journal de l'application.

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

Définition à la ligne 321 du fichier `usbThread.py`.

Voici le graphe d'appel pour cette fonction :



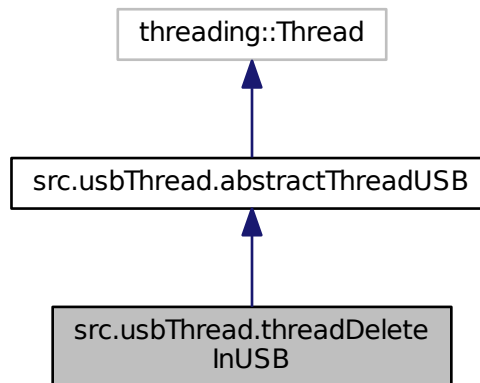
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

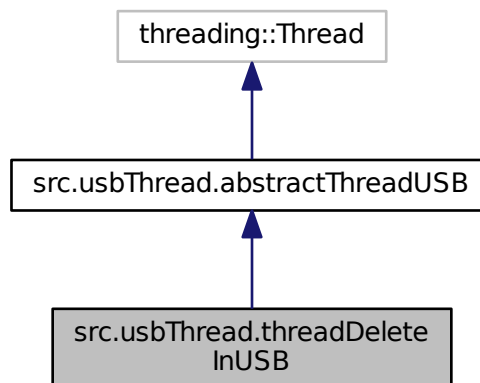
7.32 Référence de la classe `src.usbThread.threadDeleteInUSB`

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(self, ud, fileList, subdir, dest, logfile)`
Supprime une liste de fichiers dans une clé USB.

Membres hérités additionnels

7.32.1 Description détaillée

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

Définition à la ligne 506 du fichier usbThread.py.

7.32.2 Documentation des constructeurs et destructeur

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

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

Paramètres

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

Définition à la ligne 519 du fichier usbThread.py.

7.32.3 Documentation des fonctions membres

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

Supprime une liste de fichiers dans une clé USB.

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

Paramètres

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

Définition à la ligne 536 du fichier usbThread.py.

Voici le graphe d'appel pour cette fonction :



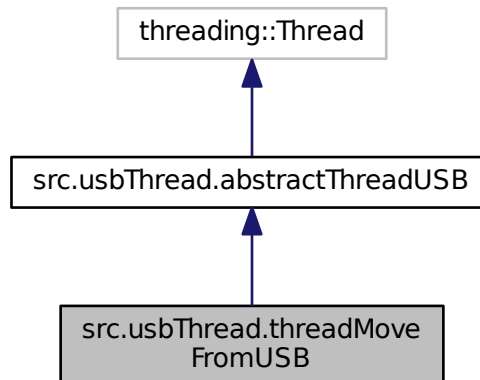
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

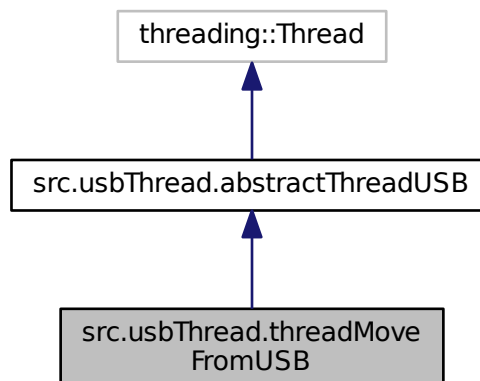
7.33 Référence de la classe src.usbThread.threadMoveFromUSB

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` (self, `ud`, `fileList`, `subdir`, `dest`, `logfile`)
Copie une liste de fichiers d'une clé USB sous un répertoire donné.

Attributs publics

- `rootPath`

7.33.1 Description détaillée

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

Définition à la ligne 429 du fichier usbThread.py.

7.33.2 Documentation des constructeurs et destructeur

7.33.2.1 `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 444 du fichier usbThread.py.

7.33.3 Documentation des fonctions membres

7.33.3.1 `def src.usbThread.threadMoveFromUSB.toDo(self, ud, fileList, subdir, dest, logfile)`

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

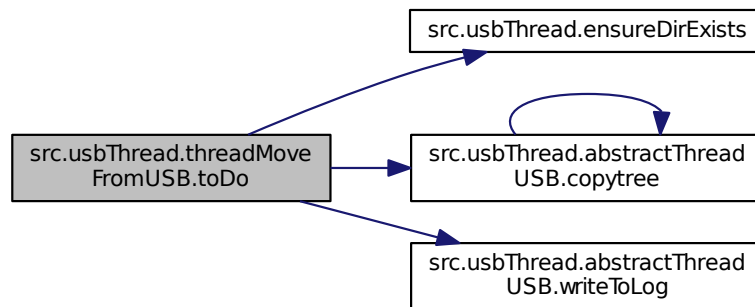
Après chaque copie réussie la source est effacée. À chaque fichier ou répertoire copié, une ligne est journalisée dans le fichier de journal de l'application.

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

Définition à la ligne 461 du fichier usbThread.py.

Voici le graphe d'appel pour cette fonction :



7.33.4 Documentation des données membres

7.33.4.1 `src.usbThread.threadMoveFromUSB.rootPath`

Définition à la ligne 447 du fichier `usbThread.py`.

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

— [src/usbThread.py](#)

7.34 Référence de la classe `src.usbThread.ThreadRegister`

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

Fonctions membres publiques

- `def __init__(self)`
Le constructeur met en place un dictionnaire.
- `def __str__(self)`
- `def push(self, ud, thread)`
- `def pop(self, ud, thread)`
- `def busy(self, owner)`
Indique si le disque est occupé par des threads.
- `def threadSet(self)`
renvoie l'ensemble des threads actifs

Attributs publics

- `dico`

7.34.1 Description détaillée

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

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

7.34.2 Documentation des constructeurs et destructeur

7.34.2.1 `def src.usbThread.ThreadRegister.__init__(self)`

Le constructeur met en place un dictionnaire.

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

7.34.3 Documentation des fonctions membres

7.34.3.1 `def src.usbThread.ThreadRegister.__str__(self)`

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

7.34.3.2 `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 81 du fichier `usbThread.py`.

7.34.3.3 `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 72 du fichier `usbThread.py`.

7.34.3.4 `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 60 du fichier `usbThread.py`.

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

renvoie l'ensemble des threads actifs

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

7.34.4 Documentation des données membres

7.34.4.1 `src.usbThread.ThreadRegister.dico`

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

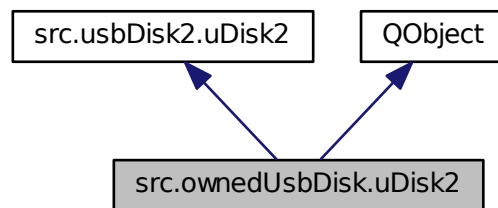
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbThread.py](#)

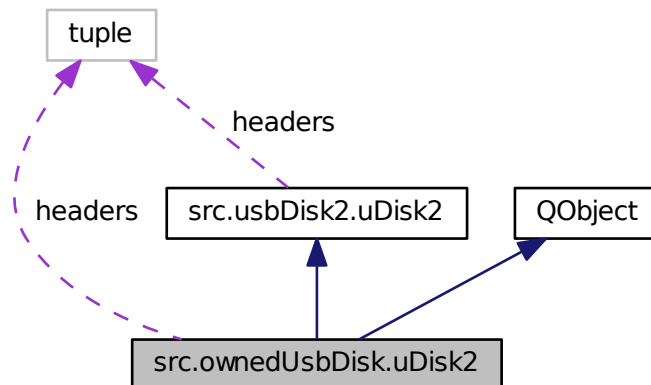
7.35 Référence de la classe src.ownedUsbDisk.uDisk2

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.uDisk2 :



Graphe de collaboration de src.ownedUsbDisk.uDisk2 :



Fonctions membres publiques

- `def __init__`
Le constructeur.
- `def getOwner (self)`
Renvoie le propriétaire.
- `def getFat (self)`
Renvoie à coup sûr la partition vfat d'un disque.
- `def valuableProperties`
Facilite l'accès aux propriétés intéressantes d'une instance.

- def `uniqueId` (self)
- def `tattoo` (self)
Renvoie un tatouage présent sur la clé, quitte à le créer.
- def `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.
- def `visibleDir` (self)
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` (self)
renvoie un nom de propriétaire dans tous les cas.
- def `__getitem__` (self, n)
renvoie un élément de listage de données internes au disque Fait en sorte que la deuxième colonne soit toujours le propriétaire
- def `ensureOwner` (self, ownerDialog)
Demande un nom de propriétaire si celui-ci n'est pas encore défini pour cette clé USB.
- def `randomOwner` (self, length)
fabrique un texte aléatoire de longueur donnée

Attributs publics

- `owner`
- `visibleDirs`

Attributs publics statiques

- tuple `headers` = staticmethod(headers)

7.35.1 Description détaillée

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

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

7.35.2 Documentation des constructeurs et destructeur

7.35.2.1 `def src.ownedUsbDisk.uDisk2.__init__(self, path, mp = '', isUsb=False, vendor = '', model = '', parent=None, fstype = '', serial = '', uuid = '', free = 0, capacity = 0, device = '', firstFat=None, selected=True)`

Le constructeur.

Paramètres

<code>path</code>	un chemin comme <code>'/org/freedesktop/UDisks2/block_devices/sdX'</code>
<code>mp</code>	point de montage ("" par défaut)
<code>isUsb</code>	en général, vrai vu qu'on se s'intéressera qu'à des périphériques USB
<code>vendor</code>	indication de vendeur
<code>model</code>	indication de modèle
<code>parent</code>	périphérique parent (None par défaut)
<code>fstype</code>	type de système de fichiers

<i>serial</i>	numéro de série
<i>uuid</i>	identifiant donné au disque lors du formatage
<i>free</i>	taille de la zone libre pour l'écriture
<i>capacity</i>	taille du périphérique
<i>device</i>	pseudo-fichier pour l'accès au périphérique
<i>firstFat</i>	une instance de <code>uDisk2</code> , de type vfat parmi les partitions
<i>selected</i>	vrai/faux selon qu'on sélectionne ou non le périphérique (vrai par défaut)

Définition à la ligne 107 du fichier `ownedUsbDisk.py`.

7.35.3 Documentation des fonctions membres

7.35.3.1 `def src.ownedUsbDisk.uDisk2.__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

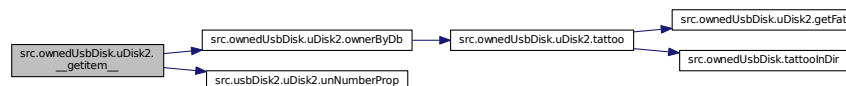
<i>n</i>	un nombre
----------	-----------

Renvoie

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

Définition à la ligne 231 du fichier `ownedUsbDisk.py`.

Voici le graphe d'appel pour cette fonction :



7.35.3.2 `def src.ownedUsbDisk.uDisk2.ensureOwner (self, ownerDialog)`

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

Enregistre au passage le nom du propriétaire dans les instances du disque et de sa partition vfat.

Paramètres

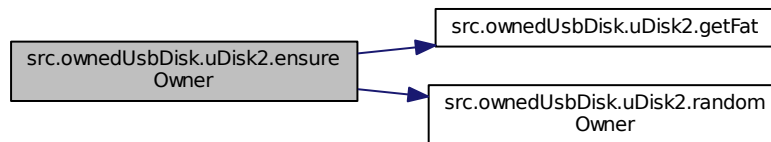
<i>ownerDialog</i>	si vrai : fait dialogue interactif
--------------------	------------------------------------

Renvoie

un nom de propriétaire

Définition à la ligne 253 du fichier ownedUsbDisk.py.

Voici le graphe d'appel pour cette fonction :

**7.35.3.3 def src.ownedUsbDisk.uDisk2.getFat (self)**

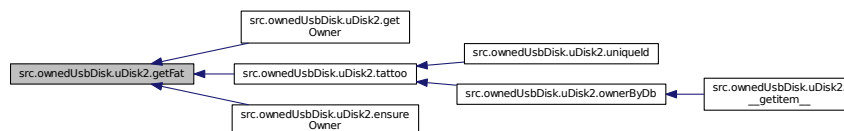
Renvoie à coup sûr la partition vfat d'un disque.

Renvoie

une instance `uDisk2` représentant une partition vfat

Définition à la ligne 129 du fichier ownedUsbDisk.py.

Voici le graphe des appelants de cette fonction :

**7.35.3.4 def src.ownedUsbDisk.uDisk2.getOwner (self)**

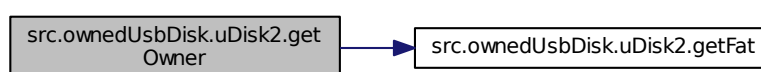
Renvoie le propriétaire.

Renvoie

le propriétaire de la clé

Définition à la ligne 121 du fichier ownedUsbDisk.py.

Voici le graphe d'appel pour cette fonction :



7.35.3.5 `def src.ownedUsbDisk.uDisk2.headers (locale = "C")`

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

Paramètres

<i>locale</i>	la locale, pour traduire les titres
---------------	-------------------------------------

Renvoie

une liste de titres de colonnes

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

7.35.3.6 `def src.ownedUsbDisk.uDisk2.ownerByDb (self)`

renvoie un nom de propriétaire dans tous les cas.

Définition à la ligne 217 du fichier `ownedUsbDisk.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

7.35.3.7 `def src.ownedUsbDisk.uDisk2.randomOwner (self, length)`

fabrique un texte aléatoire de longueur donnée

Paramètres

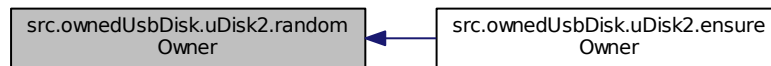
<i>length</i>	la longueur recherchée
---------------	------------------------

Renvoie

un texte pseudo-aléatoire

Définition à la ligne 276 du fichier ownedUsbDisk.py.

Voici le graphe des appelants de cette fonction :

**7.35.3.8 def src.ownedUsbDisk.uDisk2.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 176 du fichier ownedUsbDisk.py.

7.35.3.9 def src.ownedUsbDisk.uDisk2.tattoo (self)

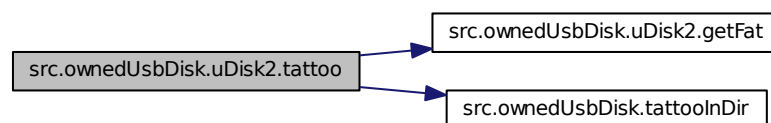
Renvoie un tatouage présent sur la clé, quitte à le créer.

Renvoie

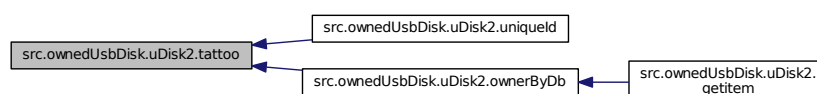
un tatouage, supposément unique.

Définition à la ligne 161 du fichier ownedUsbDisk.py.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.35.3.10 `def src.ownedUsbDisk.uDisk2.uniqueId (self)`

Renvoie

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

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

Voici le graphe d'appel pour cette fonction :

7.35.3.11 `def src.ownedUsbDisk.uDisk2.valuableProperties (self, indent = 4)`

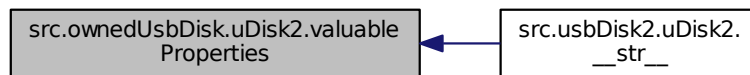
Facilite l'accès aux propriétés intéressantes d'une instance.

Renvoie

une chaîne indentée avec les propriétés intéressantes, une par ligne

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

Voici le graphe des appelants de cette fonction :

7.35.3.12 `def src.ownedUsbDisk.uDisk2.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 192 du fichier `ownedUsbDisk.py`.

7.35.4 Documentation des données membres

7.35.4.1 `tuple src.ownedUsbDisk.uDisk2.headers = staticmethod(headers)` `[static]`

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

7.35.4.2 `src.ownedUsbDisk.uDisk2.owner`

Définition à la ligne 113 du fichier `ownedUsbDisk.py`.

7.35.4.3 `src.ownedUsbDisk.uDisk2.visibleDirs`

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

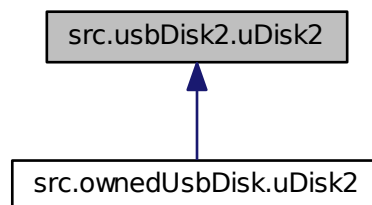
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/ownedUsbDisk.py](#)

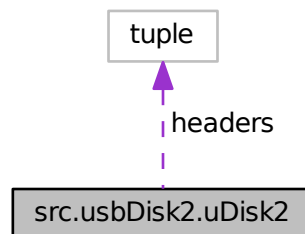
7.36 Référence de la classe `src.usbDisk2.uDisk2`

une classe pour représenter un disque ou une partition.

Graphe d'héritage de `src.usbDisk2.uDisk2` :



Graphe de collaboration de `src.usbDisk2.uDisk2` :



Fonctions membres publiques

- `def __init__`
Le constructeur.
- `def uniqueId (self)`
renvoie un identifiant unique.
- `def headers`
Méthode statique, pour avoir des titres de colonne.
- `def __str__ (self)`
Fournit une représentation imprimable.
- `def title (self)`
Permet d'obtenir un identifiant unique de disque.

- `def isDosFat (self)`
Permet de reconnaître les partitions DOS-FAT.
- `def isMounted (self)`
- `def valuableProperties`
Facilite l'accès aux propriétés intéressantes d'une instance.
- `def mountPoint (self)`
- `def unNumberProp (self, n)`
retire le numéro des en-têtes pour en faire un nom de propriété valide pour interroger dbus
- `def __getitem__ (self, n)`
Renvoie un élément de listage de données internes au disque.
- `def ensureMounted (self)`
Permet de s'assurer qu'une partition ou un disque sera bien monté

Attributs publics

- `path`
- `mp`
- `isUsb`
- `vendor`
- `model`
- `parent`
- `fstype`
- `stickid`
- `uuid`
- `free`
- `capacity`
- `devStuff`
- `firstFat`
- `selected`
- `rlock`

Attributs publics statiques

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

7.36.1 Description détaillée

une classe pour représenter un disque ou une partition.

les attributs publics sont :

- **path** le chemin dans le système dbus
- **device** l'objet dbus qui correspond à l'instance
- **device_prop** un proxy pour questionner cet objet dbus
- **selected** booléen vrai si on doit considérer cette instance comme sélectionnée. Vrai à l'initialisation
- **rlock** un verrou récursif permettant de réserver l'usage du media pour un seul thread

Définition à la ligne 395 du fichier `usbDisk2.py`.

7.36.2 Documentation des constructeurs et destructeur

7.36.2.1 `def src.usbDisk2.uDisk2.__init__(self, path, mp = '', isUsb = False, vendor = '', model = '', parent = None, fstype = '', serial = '', uuid = '', free = 0, capacity = 0, device = '', firstFat = None, selected = True)`

Le constructeur.

Paramètres

<code>path</code>	un chemin comme <code>'/org/freedesktop/UDisks2/block_devices/sdX'</code>
-------------------	---

<i>mp</i>	point de montage (" par défaut)
<i>isUsb</i>	en général, vrai vu qu'on se s'intéressera qu'à des périphériques USB
<i>vendor</i>	indication de vendeur
<i>model</i>	indication de modèle
<i>parent</i>	périphérique parent (None par défaut)
<i>fstype</i>	type de système de fichiers
<i>serial</i>	numéro de série
<i>uuid</i>	identifiant donné au disque lors du formatage
<i>free</i>	taille de la zone libre pour l'écriture
<i>capacity</i>	taille du périphérique
<i>device</i>	pseudo-fichier pour l'accès au périphérique
<i>firstFat</i>	une instance de uDisk2 , de type vfat parmi les partitions
<i>selected</i>	vrai/faux selon qu'on sélectionne ou non le périphérique (vrai par défaut)

Définition à la ligne 418 du fichier usbDisk2.py.

7.36.3 Documentation des fonctions membres

7.36.3.1 `def src.usbDisk2.uDisk2.__getitem__(self, n)`

Renvoie un élément de listage de données internes au disque.

Paramètres

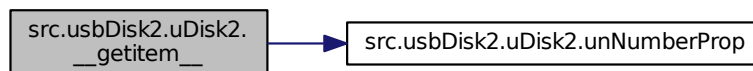
<i>n</i>	un nombre
----------	-----------

Renvoie

un élément si $n > 0$, et le drapeau `self.selected` si $n == 0$. Les noms des éléments sont dans la liste `itemNames` utilisée dans la fonction statique `headers`

Définition à la ligne 544 du fichier usbDisk2.py.

Voici le graphe d'appel pour cette fonction :



7.36.3.2 `def src.usbDisk2.uDisk2.__str__(self)`

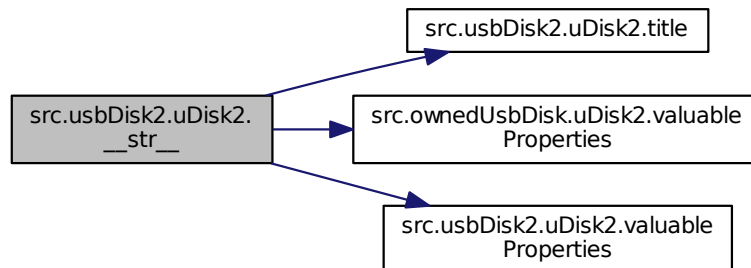
Fournit une représentation imprimable.

Renvoie

une représentation imprimable de l'instance

Définition à la ligne 476 du fichier `usbDisk2.py`.

Voici le graphe d'appel pour cette fonction :

**7.36.3.3** `def src.usbDisk2.uDisk2.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 556 du fichier `usbDisk2.py`.

7.36.3.4 `def src.usbDisk2.uDisk2.headers (locale = "C")`

Méthode statique, pour avoir des titres de colonne.

renvoie des titres pour les items obtenus par **getitem**.

Paramètres

<i>locale</i>	la locale, pour traduire les titres éventuellement. Valeur par défaut : "C"
---------------	---

Renvoie

une liste de titres de colonnes

Définition à la ligne 465 du fichier `usbDisk2.py`.

7.36.3.5 `def src.usbDisk2.uDisk2.isDosFat (self)`

Permet de reconnaître les partitions DOS-FAT.

Renvoie

True dans le cas d'une partition FAT16 ou FAT32

Définition à la ligne 492 du fichier `usbDisk2.py`.

7.36.3.6 `def src.usbDisk2.uDisk2.isMounted (self)`

Renvoie

True si le disque ou la partition est montée

Définition à la ligne 499 du fichier usbDisk2.py.

7.36.3.7 `def src.usbDisk2.uDisk2.mountPoint (self)`

Renvoie

le point de montage

Définition à la ligne 519 du fichier usbDisk2.py.

7.36.3.8 `def src.usbDisk2.uDisk2.title (self)`

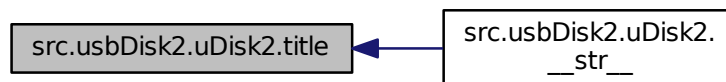
Permet d'obtenir un identifiant unique de disque.

Renvoie

le chemin dbus de l'instance

Définition à la ligne 484 du fichier usbDisk2.py.

Voici le graphe des appelants de cette fonction :



7.36.3.9 `def src.usbDisk2.uDisk2.uniqueId (self)`

renvoie un identifiant unique.

Dans cette classe, cette fonction est synonyme de file()

Renvoie

un identifiant unique, garanti par le système de fichiers

Définition à la ligne 454 du fichier usbDisk2.py.

7.36.3.10 `def src.usbDisk2.uDisk2.unNumberProp (self, n)`

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

Paramètres

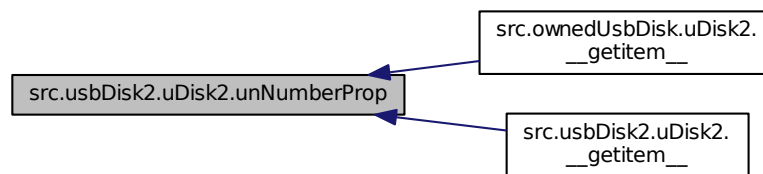
<i>n</i>	un numéro de propriété qui se réfère aux headers
----------	--

Renvoie

une propriété renvoyée par dbus, dans un format imprimable

Définition à la ligne 529 du fichier `usbDisk2.py`.

Voici le graphe des appelants de cette fonction :

7.36.3.11 `def src.usbDisk2.uDisk2.valuableProperties (self, indent = 4)`

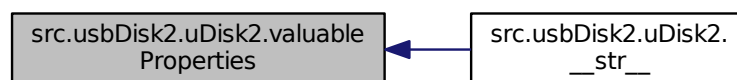
Facilite l'accès aux propriétés intéressantes d'une instance.

Renvoie

une chaîne indentée avec les propriétés intéressantes, une par ligne

Définition à la ligne 507 du fichier `usbDisk2.py`.

Voici le graphe des appelants de cette fonction :



7.36.4 Documentation des données membres

7.36.4.1 `src.usbDisk2.uDisk2.capacity`

Définition à la ligne 429 du fichier `usbDisk2.py`.

7.36.4.2 `src.usbDisk2.uDisk2.devStuff`

Définition à la ligne 430 du fichier `usbDisk2.py`.

7.36.4.3 `src.usbDisk2.uDisk2.firstFat`

Définition à la ligne 431 du fichier `usbDisk2.py`.

7.36.4.4 `src.usbDisk2.uDisk2.free`

Définition à la ligne 428 du fichier `usbDisk2.py`.

7.36.4.5 `src.usbDisk2.uDisk2.fstype`

Définition à la ligne 425 du fichier `usbDisk2.py`.

7.36.4.6 `tuple src.usbDisk2.uDisk2.headers = staticmethod(headers)` `[static]`

Définition à la ligne 469 du fichier `usbDisk2.py`.

7.36.4.7 `src.usbDisk2.uDisk2.isUsb`

Définition à la ligne 421 du fichier `usbDisk2.py`.

7.36.4.8 `src.usbDisk2.uDisk2.model`

Définition à la ligne 423 du fichier `usbDisk2.py`.

7.36.4.9 `src.usbDisk2.uDisk2.mp`

Définition à la ligne 420 du fichier `usbDisk2.py`.

7.36.4.10 `src.usbDisk2.uDisk2.parent`

Définition à la ligne 424 du fichier `usbDisk2.py`.

7.36.4.11 `src.usbDisk2.uDisk2.path`

Définition à la ligne 419 du fichier `usbDisk2.py`.

7.36.4.12 `src.usbDisk2.uDisk2.rlock`

Définition à la ligne 433 du fichier `usbDisk2.py`.

7.36.4.13 `src.usbDisk2.uDisk2.selected`

Définition à la ligne 432 du fichier `usbDisk2.py`.

7.36.4.14 `src.usbDisk2.uDisk2.stickid`

Définition à la ligne 426 du fichier `usbDisk2.py`.

7.36.4.15 src.usbDisk2.uDisk2.uuid

Définition à la ligne 427 du fichier usbDisk2.py.

7.36.4.16 src.usbDisk2.uDisk2.vendor

Définition à la ligne 422 du fichier usbDisk2.py.

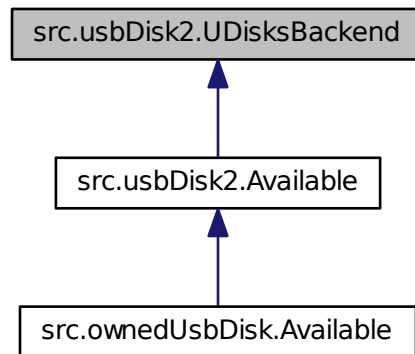
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbDisk2.py](#)

7.37 Référence de la classe src.usbDisk2.UDisksBackend

Cette classe a été inspirée par le projet USBcreator.

Graphe d'héritage de src.usbDisk2.UDisksBackend :



Fonctions membres publiques

- `def __init__`
Le constructeur.
- `def addHook` (self, signal, func)
ajoute une fonction à appeler pour un signal nommé, et enregistre cette fonction dans self.cbHooks, après vérification de sa liste de paramètres.
- `def retry_mount`
Essaie de monter un système de fichier jusqu'à ce qu'il cesse d'échouer avec "Busy", ou que l'erreur soit "déjà monté".
- `def detect_devices` (self)
Fait un inventaire des disques.
- `def objIsUsb` (self, obj)
détermine si un périphérique est de type USB

Attributs publics

- `install_thread`
- `logger`
- `diskClass`
self.targets est un dictionnaire des disques détectés les clés sont les paths et les contenus des instances de disk↔ Class

- [targets](#)
- [modified](#)
 - self.modified signifie une modification récente, à prendre en compte par une application au niveau utilisateur*
- [bus](#)
- [udisks](#)
- [manager](#)
- [cbHooks](#)

7.37.1 Description détaillée

Cette classe a été inspirée par le projet USBcreator.

Plusieurs modifications ont été faites au code original. Les fonctions de rappel ne tiennent compte que des périphériques USB

Définition à la ligne 111 du fichier usbDisk2.py.

7.37.2 Documentation des constructeurs et destructeur

7.37.2.1 `def src.usbDisk2.UDisksBackend.__init__(self, logger = logging, diskClass = object)`

Le constructeur.

Paramètres

<i>logger</i>	un objet permettant de journaliser les messages ; par défaut il se confond avec le module logging
<i>diskClass</i>	la classe à utiliser pour créer des instances de disques

Définition à la ligne 119 du fichier usbDisk2.py.

7.37.3 Documentation des fonctions membres

7.37.3.1 `def src.usbDisk2.UDisksBackend.addHook(self, signal, func)`

ajoute une fonction à appeler pour un signal nommé, et enregistre cette fonction dans self.cbHooks, après vérification de sa liste de paramètres.

Paramètres

<i>signal</i>	une chaîne
<i>func</i>	une fonction

Renvoie

le résultat de l'appel à self.manager.connect(signal,func)

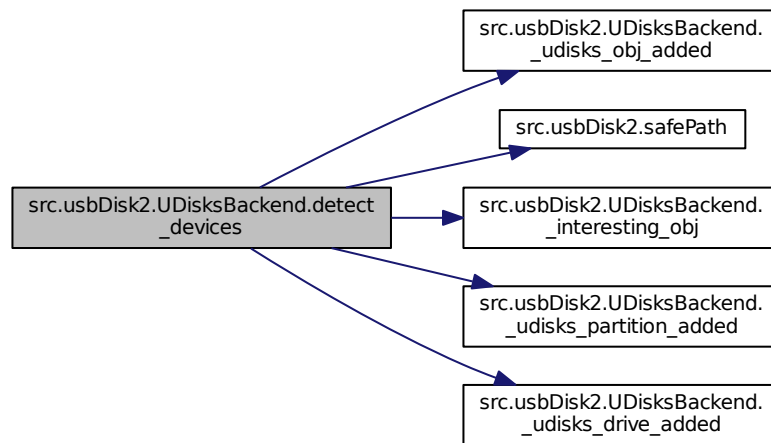
Définition à la ligne 177 du fichier usbDisk2.py.

7.37.3.2 `def src.usbDisk2.UDisksBackend.detect_devices(self)`

Fait un inventaire des disques.

Définition à la ligne 214 du fichier usbDisk2.py.

Voici le graphe d'appel pour cette fonction :



7.37.3.3 `def src.usbDisk2.UDisksBackend.objIsUsb (self, obj)`

détermine si un périphérique est de type USB

Paramètres

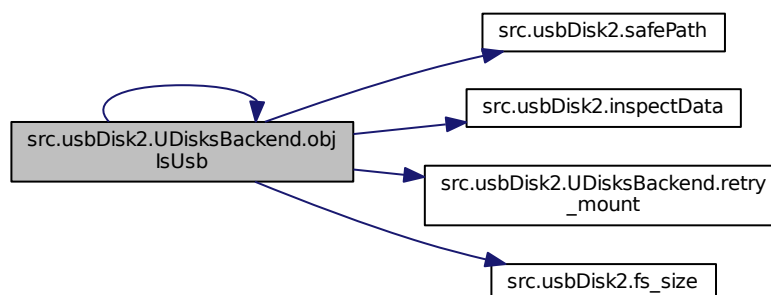
<i>obj</i>	un objet <code>UDisksObjectProxy</code>
------------	---

Renvoie

vrai si c'est un périphérique USB

Définition à la ligne 276 du fichier `usbDisk2.py`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



7.37.3.4 def src.usbDisk2.UDisksBackend.retry_mount (self, fs, timeout = 5, retryDelay = 0.3)

Essaie de monter un système de fichier jusqu'à ce qu'il cesse d'échouer avec "Busy", ou que l'erreur soit "déjà monté".

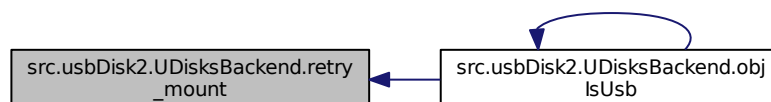
Échoue si l'erreur est autre que les deux précédentes.

Paramètres

<i>fs</i>	un système de fichier à monter
<i>timeout</i>	nombre de secondes d'attente au maximum
<i>retryDelay</i>	délai entre deux essais

Définition à la ligne 194 du fichier usbDisk2.py.

Voici le graphe des appelants de cette fonction :



7.37.4 Documentation des données membres

7.37.4.1 src.usbDisk2.UDisksBackend.bus

Définition à la ligne 131 du fichier usbDisk2.py.

7.37.4.2 src.usbDisk2.UDisksBackend.cbHooks

Définition à la ligne 134 du fichier usbDisk2.py.

7.37.4.3 src.usbDisk2.UDisksBackend.diskClass

self.targets est un dictionnaire des disques détectés les clés sont les paths et les contenus des instances de disk↔
Class

Définition à la ligne 124 du fichier `usbDisk2.py`.

7.37.4.4 `src.usbDisk2.UDisksBackend.install_thread`

Définition à la ligne 120 du fichier `usbDisk2.py`.

7.37.4.5 `src.usbDisk2.UDisksBackend.logger`

Définition à la ligne 121 du fichier `usbDisk2.py`.

7.37.4.6 `src.usbDisk2.UDisksBackend.manager`

Définition à la ligne 133 du fichier `usbDisk2.py`.

7.37.4.7 `src.usbDisk2.UDisksBackend.modified`

`self.modified` signifie une modification récente, à prendre en compte par une application au niveau utilisateur désactivé, quelquefois `drive.get_cached_property('Size').get_uint64()` renvoie des résultats erronés juste après le branchement

Définition à la ligne 128 du fichier `usbDisk2.py`.

7.37.4.8 `src.usbDisk2.UDisksBackend.targets`

Définition à la ligne 125 du fichier `usbDisk2.py`.

7.37.4.9 `src.usbDisk2.UDisksBackend.udisks`

Définition à la ligne 132 du fichier `usbDisk2.py`.

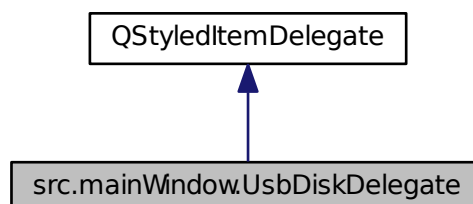
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/usbDisk2.py](#)

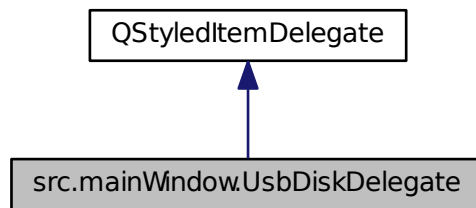
7.38 Référence de la classe `src.mainWindow.UsbDiskDelegate`

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__(self, parent)`
- `def paint(self, painter, option, index)`

Attributs publics

- `okPixmap`
- `busyPixmap`

7.38.1 Description détaillée

Classe pour identifier le baladeur dans le tableau.

La routine de rendu à l'écran trace une petite icône et le nom du propriétaire à côté.

Définition à la ligne 910 du fichier `mainWindow.py`.

7.38.2 Documentation des constructeurs et destructeur

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

Définition à la ligne 911 du fichier `mainWindow.py`.

7.38.3 Documentation des fonctions membres

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

Définition à la ligne 916 du fichier `mainWindow.py`.

7.38.4 Documentation des données membres

7.38.4.1 `src.mainWindow.UsbDiskDelegate.busyPixmap`

Définition à la ligne 914 du fichier `mainWindow.py`.

7.38.4.2 src.mainWindow.UsbDiskDelegate.okPixmap

Définition à la ligne 913 du fichier mainWindow.py.

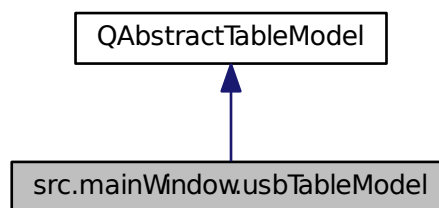
La documentation de cette classe a été générée à partir du fichier suivant :

— [src/mainWindow.py](#)

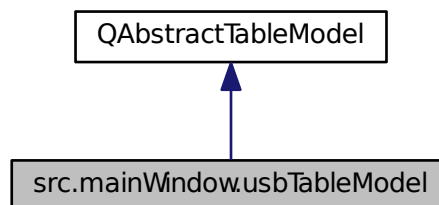
7.39 Référence de la classe src.mainWindow.usbTableModel

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 [updateOwnerColumn](#) (self)
force la mise à jour de la colonne des propriétaires
- def [rowCount](#) (self, parent)
un QModelIndex
- def [columnCount](#) (self, parent)
un QModelIndex
- def [setData](#) (self, index, value, role)
- def [partition](#) (self, index)
- def [data](#) (self, index, role)
- def [headerData](#) (self, section, orientation, role)
- def [sort](#)

Sort table by given column number.

Attributs publics

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

7.39.1 Description détaillée

Un modèle de table pour des séries de clés USB.

Définition à la ligne 755 du fichier mainWindow.py.

7.39.2 Documentation des constructeurs et destructeur

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

Paramètres

<i>parent</i>	un QObject
<i>header</i>	les en-têtes de colonnes
<i>donnees</i>	les données

Définition à la ligne 763 du fichier mainWindow.py.

7.39.3 Documentation des fonctions membres

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

un `QModelIndex`

Définition à la ligne 789 du fichier mainWindow.py.

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

Définition à la ligne 807 du fichier mainWindow.py.

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

Définition à la ligne 842 du fichier mainWindow.py.

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

Paramètres

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

Renvoie

la partition pointée par index

Définition à la ligne 804 du fichier mainWindow.py.

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

un `QModelIndex`

Définition à la ligne 782 du fichier `mainWindow.py`.

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

Définition à la ligne 792 du fichier `mainWindow.py`.

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

Sort table by given column number.

Paramètres

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

Définition à la ligne 854 du fichier `mainWindow.py`.

7.39.3.8 `def src.mainWindow.usbTableModel.updateOwnerColumn (self)`

force la mise à jour de la colonne des propriétaires

Définition à la ligne 773 du fichier `mainWindow.py`.

7.39.4 Documentation des données membres

7.39.4.1 `src.mainWindow.usbTableModel.donnees`

Définition à la ligne 766 du fichier `mainWindow.py`.

7.39.4.2 `src.mainWindow.usbTableModel.header`

Définition à la ligne 765 du fichier `mainWindow.py`.

7.39.4.3 `src.mainWindow.usbTableModel.pere`

Définition à la ligne 767 du fichier `mainWindow.py`.

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

— [src/mainWindow.py](#)

Chapitre 8

Documentation des fichiers

8.1 Référence du fichier src/__init__.py

Espaces de nommage

— [src](#)

8.2 Référence du fichier src/checkboxDialog.py

Classes

— class [src.checkboxDialog.CheckBoxDialog](#)
Un dialogue pour gérer les cases à cocher de l'application.

Espaces de nommage

— [src.checkboxDialog](#)

Variables

— string [src.checkboxDialog.licenceEn](#)

8.3 Référence du fichier src/choixEleves.py

Classes

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

Espaces de nommage

— [src.choixEleves](#)

Variables

— dictionary [src.choixEleves.licence](#) = {}

- tuple [src.choixEleves.app](#) = QApplication(sys.argv)
- tuple [src.choixEleves.d](#) = choixElevesDialog(gestionnaire=gestClasse.Sconet)
- tuple [src.choixEleves.i](#) = d.pop()

8.4 Référence du fichier src/chooseInSticks.py

Classes

- class [src.chooseInSticks.chooseDialog](#)
Un dialogue pour choisir un ensemble de fichiers à copier depuis une clé USB.

Espaces de nommage

- [src.chooseInSticks](#)

Variables

- string [src.chooseInSticks.licenceEn](#)

8.5 Référence du fichier src/copyToDialog1.py

Classes

- class [src.copyToDialog1.copyToDialog1](#)
Un dialogue pour choisir un ensemble de fichiers à transférer vers une collection de clés USB.

Espaces de nommage

- [src.copyToDialog1](#)

Variables

- string [src.copyToDialog1.licenceEn](#)
- tuple [src.copyToDialog1.app](#) = QApplication(sys.argv)
- tuple [src.copyToDialog1.windows](#) = copyToDialog1()

8.6 Référence du fichier src/db.py

Espaces de nommage

- [src.db](#)

Fonctions

- def [src.db.openDb](#) ()
Ouverture de la base de données de l'application, et création si nécessaire.
- def [src.db.checkVersion](#) (major, minor)
Vérifie si la base de données reste compatible.
- def [src.db.hasStudent](#) (student)
vérifie qu'un étudiant est déjà connu
- def [src.db.knowsId](#) (stickid, uuid, tattoo)
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](#) (stickid, uuid, tattoo)
renvoie l'étudiant qui possède une clé USB
- def [src.db.readPrefs](#) ()
renvoie les préférences de ScolaSync
- def [src.db.setWd](#) (newDir)
définit le nouveau nom du répertoire de travail préféré.
- def [src.db.writeStudent](#) (stickid, uuid, tattoo, student)
inscrit un étudiant comme propriétaire d'une clé USB
- def [src.db.writePrefs](#) (prefs)
inscrit les préférences

Variables

- dictionary [src.db.licence](#) = {}
- [src.db.database](#) = None
- [src.db.cursor](#) = None

8.7 Référence du fichier src/debug.py

Espaces de nommage

- [src.debug](#)

Fonctions

- def [src.debug.button](#) (w, cb)
ajoute un bouton de débogage dans une fenêtre
- def [src.debug.listePartitionsCochees](#) (w)
renseigne sur la liste des partions cochées de la fenêtre principale

Variables

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

8.8 Référence du fichier src/diskFull.py

Classes

- class [src.diskFull.mainWindow](#)

Espaces de nommage

- [src.diskFull](#)

Fonctions

- def [src.diskFull.sceneWithUsage](#) (parent, rect, percent)

Variables

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

8.9 Référence du fichier src/gestClasse.py

Classes

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

Espaces de nommage

- [src.gestClasse](#)

Variables

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

8.10 Référence du fichier src/gestclassetreeview.py

Classes

- class [src.gestclassetreeview.gestClasseTreeView](#)

Espaces de nommage

- [src.gestclassetreeview](#)

Variables

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

8.11 Référence du fichier src/globaldef.py

Espaces de nommage

- [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"

8.12 Référence du fichier src/help.py

Classes

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

Espaces de nommage

- [src.help](#)

Variables

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

8.13 Référence du fichier src/mainWindow.py

Classes

- class [src.mainWindow.mainWindow](#)
defines the main window of the application.
- 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.

Espaces de nommage

- [src.mainWindow](#)

Fonctions

- def [src.mainWindow.registerCmd](#) (cmd, partition)
enregistre la commande cmd pour la partition donnée
- def [src.mainWindow.CheckBoxRect](#) (view_item_style_options)

Variables

- dictionary [src.mainWindow.licence](#) = {}
- dictionary [src.mainWindow.activeThreads](#) = {}
- dictionary [src.mainWindow.pastCommands](#) = {}
- [src.mainWindow.lastCommand](#) = None

8.14 Référence du fichier src/marques.py

Espaces de nommage

- [src.marques](#)

8.15 Référence du fichier src/mytextbrowser.py

Classes

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

Espaces de nommage

- [src.mytextbrowser](#)

Variables

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

8.16 Référence du fichier src/nameAdrive.py

Classes

- class [src.nameAdrive.nameAdriveDialog](#)
un dialogue pour renommer un baladeur, compte tenu d'une liste de noms disponibles

Espaces de nommage

- [src.nameAdrive](#)

Variables

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

8.17 Référence du fichier src/notification.py

Classes

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

Espaces de nommage

- [src.notification](#)

Variables

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

8.18 Référence du fichier src/ownedUsbDisk.py

Classes

- class [src.ownedUsbDisk.uDisk2](#)
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.*
- class [src.ownedUsbDisk.MainWindow](#)

Espaces de nommage

- [src.ownedUsbDisk](#)

Fonctions

- def [src.ownedUsbDisk.tattooInDir](#) (mountPoint)
Renvoie le tatouage pour un point de montage donné, quitte à le créer si nécessaire.
- def [src.ownedUsbDisk.editRecord](#)
édition de la base de données.
- def [src.ownedUsbDisk.print_targets_if_modif](#) (man, obj)

Variables

- dictionary [src.ownedUsbDisk.licence](#) = {}
- tuple [src.ownedUsbDisk.app](#) = QApplication(sys.argv)
- tuple [src.ownedUsbDisk.main](#) = MainWindow()

8.19 Référence du fichier src/preferences.py

Classes

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

Espaces de nommage

- [src.preferences](#)

Variables

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

8.20 Référence du fichier src/scolasync.py

Espaces de nommage

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

Fonctions

- def [src.scolasync.run](#)
Le lancement de l'application.

Variables

- dictionary [src.scolasync.licence](#) = {}
- string [src.scolasync.licenceEn](#)
- string [src.scolasync.licenceFr](#)

8.21 Référence du fichier src/sconet.py

Classes

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

Espaces de nommage

- [src.sconet](#)

Variables

- dictionary [src.sconet.licence](#) = {}
- tuple [src.sconet.s](#) = [Sconet](#)("../exemples/SCONET_test.xml")

8.22 Référence du fichier src/test3.py

Espaces de nommage

- [src.test3](#)

Variables

- [src.test3.python3safe](#) = True
- tuple [src.test3.files](#) = `os.listdir(".")`
- tuple [src.test3.pattern](#) = `re.compile(".*\.py$")`
- list [src.test3.safe](#) = []
- list [src.test3.notsafe](#) = []
- tuple [src.test3.moduleName](#) = `f.replace(".py", "")`
- tuple [src.test3.module](#) = `__import__(moduleName)`

8.23 Référence du fichier src/usbDisk2.py

Classes

- class [src.usbDisk2.UDisksBackend](#)
Cette classe a été inspirée par le projet USBcreator.
- class [src.usbDisk2.uDisk2](#)
une classe pour représenter un disque ou une partition.
- class [src.usbDisk2.Available](#)
une classe pour représenter la collection des disques USB connectés
- class [src.usbDisk2.MainWindow](#)

Espaces de nommage

- [src.usbDisk2](#)

Fonctions

- def [src.usbDisk2.inspectData](#) ()
- def [src.usbDisk2.safePath](#) (obj)
Récupère de façon sûre le path d'une instance de [UDisksObjectProxy](#).
- def [src.usbDisk2.fs_size](#) (device)
Renvoie la taille d'un système de fichier et la place disponible.
- def [src.usbDisk2.print_targets_if_modif](#) (man, obj)

Variables

```

— dictionary src.usbDisk2.licence = {}
— string src.usbDisk2.licence_en
— string src.usbDisk2.dependencies = "python3-dbus python3-dbus.mainloop.qt"
— src.usbDisk2.debug = False
  activate debugging #####
— tuple src.usbDisk2.no_options = GLib.Variant('a{sv}', {})
  la variable suivante a été recopiées à l'aveugle ##### depuis un fichier du projet USBcreator
  #####
— tuple src.usbDisk2.not_interesting
  des "chemins" correspondant à des disques non débranchables #####
— tuple src.usbDisk2.app = QApplication(sys.argv)
— tuple src.usbDisk2.main = MainWindow()
```

8.24 Référence du fichier src/usbThread.py

Classes

```

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

Espaces de nommage

```

— src.usbThread
```

Fonctions

```

— def src.usbThread.ensureDirExists (destpath)
  force l'existence d'un répertoire, récursivement si nécessaire
— def src.usbThread.test_copytree ()
  Teste la fonction copytree.
— def src.usbThread.test_copy2 ()
  Teste la copie d'un fichier vers une destination telle qu'elle est pratiquée dans la méthode copytree de abstractThreadUSB.
```

Variables

```

— string src.usbThread.licenceEn
— int src.usbThread._threadNumber = 0
```

8.25 Référence du fichier src/version.py

Espaces de nommage

```

— src.version
```

Fonctions

- def `src.version.major` ()
- def `src.version.minor` ()
- def `src.version.version` ()

Variables

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