Editeur de configuration pour simulation électromagnétique

1.0

Généré par Doxygen 1.7.6.1

Mercredi Mai 29 2013 18 :48 :49

Table des matières

1 Index des classes **1**

1.1 Hiérarchie des classes 1

2 Index des classes **5**

2.1 Liste des classes 5

3 Index des fichiers **11**

3.1 Liste des fichiers 11

4 Documentation des classes **15**

4.1 Référence de la classe analyseTemporelle 15

4.1.1 Description détaillée 16

4.1.2 Documentation des constructeurs et destructeur 16

4.1.2.1 analyseTemporelle 16

4.1.2.2 ∼analyseTemporelle 16

4.1.3 Documentation des fonctions membres 17

4.1.3.1 ecrire 17

4.1.3.2 GetPeriode 17

4.1.3.3 GetTemps 17

4.1.3.4 SetPeriode 17

4.1.3.5 SetTemps 17

4.2 Référence de la classe blocCartographieTemporelle 17

4.2.1 Description détaillée 19

4.2.2 Documentation des constructeurs et destructeur 19

4.2.2.1 blocCartographieTemporelle 19

4.2.2.2 ∼blocCartographieTemporelle 19

4.2.3 Documentation des fonctions membres 19
4.2.3.1 ecrire 19
4.2.3.2 GetCarto 19
4.3 Référence de la classe blocConfiguration 19
4.3.1 Description détaillée 21
4.3.2 Documentation des constructeurs et destructeur 21
4.3.2.1 blocConfiguration 21
4.3.2.2 ∼blocConfiguration 21
4.3.3 Documentation des fonctions membres 21
4.3.3.1 ecrire 21
4.3.3.2 getHeader 21
4.3.3.3 setExtension 21
4.3.3.4 setHeader 22
4.3.4 Documentation des données membres 22
4.3.4.1 extension 22
4.3.4.2 header 22
4.4 Référence de la classe blocEchantillonnage 22
4.4.1 Description détaillée 23
4.4.2 Documentation des constructeurs et destructeur 23
4.4.2.1 blocEchantillonnage 23
4.4.2.2 ∼blocEchantillonnage 23
4.4.3 Documentation des fonctions membres 23
4.4.3.1 ecrire 24
4.4.3.2 GetFacteurEchatillonnage 24
4.4.3.3 SetFacteurEchatillonnage 24
4.5 Référence de la classe BlocElementBase 24
4.5.1 Description détaillée 25
4.5.2 Documentation des constructeurs et destructeur 25
4.5.2.1 BlocElementBase 26
4.5.3 Documentation des fonctions membres 26
4.5.3.1 addElement 26
4.5.3.2 getElement 26
4.5.3.3 getListElementPointer 26
4.5.3.4 getNbElement 26

4.5.3.5 removeElement 27
4.5.3.6 removeElement 27
4.5.3.7 setElement 27
4.5.4 Documentation des données membres 27
4.5.4.1 listElement 27
4.6 Référence de la classe blocElementLocalise 27
4.6.1 Description détaillée 29
4.6.2 Documentation des constructeurs et destructeur 29
4.6.2.1 blocElementLocalise 29
4.6.2.2 ∼blocElementLocalise 29
4.6.3 Documentation des fonctions membres 29
4.6.3.1 ecrire 29
4.6.3.2 GetElementLocalise 29
4.7 Référence de la classe blocMetallisation 29
4.7.1 Description détaillée 31
4.7.2 Documentation des constructeurs et destructeur 31
4.7.2.1 blocMetallisation 31
4.7.2.2 ∼blocMetallisation 31
4.7.3 Documentation des fonctions membres 31
4.7.3.1 ecrire 31
4.7.3.2 GetMetallisation 31
4.8 Référence de la classe blocParallelepipede 31
4.8.1 Description détaillée 33
4.8.2 Documentation des constructeurs et destructeur 33
4.8.2.1 blocParallelepipede 33
4.8.2.2 ∼blocParallelepipede 33
4.8.3 Documentation des fonctions membres 33
4.8.3.1 ecrire 33
4.8.3.2 getParallelepipede 33
4.9 Référence de la classe blocPortExcitation 33
4.9.1 Description détaillée 35
4.9.2 Documentation des constructeurs et destructeur 35
4.9.2.1 blocPortExcitation 35
4.9.2.2 ∼blocPortExcitation 35

4.9.3 Documentation des fonctions membres 35
4.9.3.1 ecrire 35
4.9.3.2 GetPortExcitation 35
4.10 Référence de la classe blocSonde 35
4.10.1 Description détaillée 37
4.10.2 Documentation des constructeurs et destructeur 37
4.10.2.1 blocSonde 37
4.10.2.2 ∼blocSonde 37
4.10.3 Documentation des fonctions membres 37
4.10.3.1 ecrire 37
4.10.3.2 GetSonde 37
4.11 Référence de la classe blocSurfacePrelevement 37
4.11.1 Description détaillée 39
4.11.2 Documentation des constructeurs et destructeur 39
4.11.2.1 blocSurfacePrelevement 39
4.11.2.2 ∼blocSurfacePrelevement 39
4.11.3 Documentation des fonctions membres 39
4.11.3.1 ecrire 39
4.11.3.2 GetNbSurfacesDG 39
4.11.3.3 GetSurface 39
4.11.3.4 SetNbSurfacesDG 39
4.12 Référence de la classe cageExcitation 40
4.12.1 Description détaillée 41
4.12.2 Documentation des constructeurs et destructeur 41
4.12.2.1 cageExcitation 41
4.12.2.2 ∼cageExcitation 41
4.12.3 Documentation des fonctions membres 41
4.12.3.1 ecrire 42
4.12.3.2 GetHauteur 42
4.12.3.3 GetInsideOutside 42
4.12.3.4 GetModeExcitation 42
4.12.3.5 GetModulationAmplitude 42
4.12.3.6 GetModulationPhase 42
4.12.3.7 GetNbFaces 42

4.12.3.8 GetNomFichier 42
4.12.3.9 GetPhi0 42
4.12.3.10 GetPointReference 42
4.12.3.11 GetPsi0 43
4.12.3.12 GetSensPropagation 43
4.12.3.13 GetTheta0 43
4.12.3.14 GetTypeExcitation 43
4.12.3.15 isCreate 43
4.12.3.16 setCreate 43
4.12.3.17 SetHauteur 43
4.12.3.18 SetInsideOutside 43
4.12.3.19 SetModeExcitation 43
4.12.3.20 SetModulationAmplitude 43
4.12.3.21 SetModulationPhase 44
4.12.3.22 SetNbFaces 44
4.12.3.23 SetNomFichier 44
4.12.3.24 SetPhi0 44
4.12.3.25 SetPointReference 44
4.12.3.26 SetPointReference 44
4.12.3.27 SetPsi0 44
4.12.3.28 SetSensPropagation 44
4.12.3.29 SetTheta0 44
4.12.3.30 SetTypeExcitation 45
4.13 Référence de la classe CalculChampsLointainWidget 45
4.13.1 Description détaillée 46
4.13.2 Documentation des constructeurs et destructeur 46
4.13.2.1 CalculChampsLointainWidget 46
4.13.3 Documentation des fonctions membres 46
4.13.3.1 init 46
4.14 Référence de la classe calculDirectivite 47
4.14.1 Description détaillée 48
4.14.2 Documentation des constructeurs et destructeur 48
4.14.2.1 calculDirectivite 48
4.14.2.2 ∼calculDirectivite 48

4.14.3 Documentation des fonctions membres 48
4.14.3.1 ecrire 48
4.14.3.2 GetPasPhi 48
4.14.3.3 GetPasTheta 48
4.14.3.4 IsCalcul 49
4.14.3.5 SetCalcul 49
4.14.3.6 SetPasPhi 49
4.14.3.7 SetPasTheta 49
4.15 Référence de la classe cartographieTemporelle 49
4.15.1 Description détaillée 50
4.15.2 Documentation des constructeurs et destructeur 50
4.15.2.1 cartographieTemporelle 51
4.15.2.2 ∼cartographieTemporelle 51
4.15.3 Documentation des fonctions membres 51
4.15.3.1 ecrire 51
4.15.3.2 GetIntervalle 51
4.15.3.3 GetIterationDebut 51
4.15.3.4 GetIterationFin 51
4.15.3.5 IsChampE 51
4.15.3.6 IsChampH 51
4.15.3.7 SetChampE 51
4.15.3.8 SetChampH 51
4.15.3.9 SetIntervalle 52
4.15.3.10 SetIterationDebut 52
4.15.3.11 SetIterationFin 52
4.16 Référence de la classe champLointain 52
4.16.1 Description détaillée 53
4.16.2 Documentation des constructeurs et destructeur 53
4.16.2.1 champLointain 54
4.16.2.2 ∼champLointain 54
4.16.3 Documentation des fonctions membres 54
4.16.3.1 ecrire 54
4.16.3.2 GetFrequenceMax 54
4.16.3.3 GetFrequenceMin 54

4.16.3.4 GetHauteurPlanMasse 54
4.16.3.5 GetNomFichier 54
4.16.3.6 GetOrigineRepere 54
4.16.3.7 GetPasDiscrPhi 54
4.16.3.8 GetPasDiscrTeta 54
4.16.3.9 GetPasFrequence 55
4.16.3.10 IsCalculImage 55
4.16.3.11 SetCalculImage 55
4.16.3.12 SetFrequenceMax 55
4.16.3.13 SetFrequenceMin 55
4.16.3.14 SetHauteurPlanMasse 55
4.16.3.15 SetNomFichier 55
4.16.3.16 SetOrigineRepere 55
4.16.3.17 SetOrigineRepere 55
4.16.3.18 SetPasDiscrPhi 56
4.16.3.19 SetPasDiscrTeta 56
4.16.3.20 SetPasFrequence 56
4.17 Référence de la classe ColorPicker 56
4.17.1 Description détaillée 57
4.17.2 Documentation des constructeurs et destructeur 57
4.17.2.1 ColorPicker 57
4.17.3 Documentation des fonctions membres 57
4.17.3.1 colorChanged 57
4.17.3.2 ouvreColorDialog 57
4.17.3.3 paintEvent 57
4.17.3.4 setColor 57
4.18 Référence de la classe ComboxDecaleeDelegate 58
4.18.1 Description détaillée 58
4.18.2 Documentation des constructeurs et destructeur 58
4.18.2.1 ComboxDecaleeDelegate 58
4.18.3 Documentation des fonctions membres 58
4.18.3.1 createEditor 58
4.18.3.2 paint 58
4.18.3.3 setEditorData 59

4.18.3.4 setModelData 59
4.18.3.5 updateEditorGeometry 59
4.19 Référence de la classe compressionHuygens 59
4.19.1 Description détaillée 60
4.19.2 Documentation des constructeurs et destructeur 60
4.19.2.1 compressionHuygens 60
4.19.2.2 ∼compressionHuygens 60
4.19.3 Documentation des fonctions membres 61
4.19.3.1 ecrire 61
4.19.3.2 GetCompressionHauteur 61
4.19.3.3 GetCompressionLargeur 61
4.19.3.4 GetCompressionLongueur 61
4.19.3.5 GetFacteurMultiplicatif 61
4.19.3.6 SetCompressionHauteur 61
4.19.3.7 SetCompressionLargeur 61
4.19.3.8 SetCompressionLongueur 61
4.19.3.9 SetFacteurMultiplicatif 62
4.20 Référence de la classe coordonnee 62
4.20.1 Description détaillée 62
4.20.2 Documentation des constructeurs et destructeur 62
4.20.2.1 coordonnee 62
4.20.2.2 coordonnee 62
4.20.2.3 coordonnee 63
4.20.2.4 ∼coordonnee 63
4.20.3 Documentation des fonctions membres 63
4.20.3.1 ecrire 63
4.20.3.2 GetX 63
4.20.3.3 GetY 63
4.20.3.4 GetZ 63
4.20.3.5 operator== 63
4.20.3.6 SetX 63
4.20.3.7 SetY 63
4.20.3.8 SetZ 63
4.20.3.9 toQString 64

4.21 Référence de la classe CoordonneeDelegate 64
4.21.1 Description détaillée 64
4.21.2 Documentation des constructeurs et destructeur 64
4.21.2.1 CoordonneeDelegate 64
4.21.3 Documentation des fonctions membres 64
4.21.3.1 createEditor 65
4.21.3.2 paint 65
4.21.3.3 setEditorData 65
4.21.3.4 setModelData 65
4.21.3.5 updateEditorGeometry 65
4.22 Référence de la classe Cube 65
4.22.1 Description détaillée 66
4.22.2 Documentation des constructeurs et destructeur 66
4.22.2.1 Cube 66
4.22.2.2 Cube 66
4.22.3 Documentation des fonctions membres 66
4.22.3.1 dessineCube 66
4.22.3.2 drawCube 66
4.23 Référence de la classe Data 66
4.23.1 Description détaillée 67
4.23.2 Documentation des constructeurs et destructeur 67
4.23.2.1 Data 67
4.23.2.2 Data 67
4.23.3 Documentation des fonctions membres 67
4.23.3.1 ecrire 67
4.23.3.2 getAnalyseTempo 67
4.23.3.3 getBlocCartoTempo 67
4.23.3.4 getBlocEchantillonnage 68
4.23.3.5 getBlocElementLocalise 68
4.23.3.6 getBlocMetallisations 68
4.23.3.7 getBlocParallelepipede 68
4.23.3.8 getBlocPortExcitation 68
4.23.3.9 getBlocSonde 68
4.23.3.10 getBlocSurfacePrelev 68

4.23.3.11 getCageExcitation 68
4.23.3.12 getCalculDirective 68
4.23.3.13 getChampLointain 68
4.23.3.14 getCompressionHuygens 69
4.23.3.15 getExcitation 69
4.23.3.16 getFormatStockage 69
4.23.3.17 getParamExcitations 69
4.23.3.18 getParoi 69
4.23.3.19 getPML 69
4.23.3.20 getPortExcitations 69
4.23.3.21 getVolumeCalcul 69
4.24 Référence de la classe DescriptionGeoWidget 69
4.24.1 Description détaillée 71
4.24.2 Documentation des constructeurs et destructeur 71
4.24.2.1 DescriptionGeoWidget 71
4.24.3 Documentation des fonctions membres 71
4.24.3.1 init 71
4.24.3.2 newElemLocCreated 71
4.24.3.3 newMetalCreated 71
4.24.3.4 newParaCreated 72
4.24.3.5 openElementLocaliseWindows 72
4.25 Référence de la classe Ecriture 72
4.25.1 Description détaillée 72
4.25.2 Documentation des fonctions membres 72
4.25.2.1 Ecrire 72
4.25.2.2 GetNomFichier 72
4.25.2.3 SetNomFichier 73
4.25.2.4 setNumStru 73
4.26 Référence de la classe elementBase 73
4.26.1 Description détaillée 75
4.26.2 Documentation des constructeurs et destructeur 76
4.26.2.1 elementBase 76
4.26.2.2 ∼elementBase 76
4.26.3 Documentation des fonctions membres 76

4.26.3.1 ecrireElementBase 76
4.26.3.2 GetArriereDroit 76
4.26.3.3 GetAvantGauche 76
4.26.3.4 getCouleur 76
4.26.3.5 getInvertColor 76
4.26.3.6 invertColor 76
4.26.3.7 operator== 76
4.26.3.8 setAlpha 76
4.26.3.9 SetArriereDroit 77
4.26.3.10 SetArriereDroit 77
4.26.3.11 SetAvantGauche 77
4.26.3.12 SetAvantGauche 77
4.26.3.13 setCouleur 77
4.26.4 Documentation des données membres 77
4.26.4.1 arriereDroit 77
4.26.4.2 avantGauche 77
4.26.4.3 couleur 77
4.27 Référence de la classe elementLocalise 77
4.27.1 Description détaillée 79
4.27.2 Documentation des constructeurs et destructeur 79
4.27.2.1 elementLocalise 79
4.27.2.2 ∼elementLocalise 79
4.27.3 Documentation des fonctions membres 79
4.27.3.1 ecrire 79
4.27.3.2 GetDirection 79
4.27.3.3 GetType 79
4.27.3.4 GetValeur 79
4.27.3.5 SetDirection 79
4.27.3.6 SetType 80
4.27.3.7 SetValeur 80
4.28 Référence de la classe Excitation 80
4.28.1 Description détaillée 81
4.28.2 Documentation des constructeurs et destructeur 81
4.28.2.1 Excitation 81

4.28.2.2 ∼Excitation 81
4.28.3 Documentation des fonctions membres 81
4.28.3.1 ecrire 82
4.28.3.2 GetAmplitudeV0 82
4.28.3.3 SetAmplitudeV0 82
4.29 Référence de la classe FilterWidget 82
4.29.1 Description détaillée 82
4.29.2 Documentation des constructeurs et destructeur 83
4.29.2.1 FilterWidget 83
4.29.3 Documentation des fonctions membres 83
4.29.3.1 clear 83
4.29.3.2 columnChanged 83
4.29.3.3 filterChanged 83
4.29.3.4 setCurrentModel 83
4.29.3.5 setListCombox 83
4.30 Référence de la classe FormAnalyseTempo 83
4.30.1 Description détaillée 83
4.30.2 Documentation des constructeurs et destructeur 84
4.30.2.1 FormAnalyseTempo 84
4.30.3 Documentation des fonctions membres 84
4.30.3.1 init 84
4.30.3.2 setAnalyseTempo 84
4.31 Référence de la classe formatStockage 84
4.31.1 Description détaillée 85
4.31.2 Documentation des constructeurs et destructeur 85
4.31.2.1 formatStockage 85
4.31.2.2 ∼formatStockage 85
4.31.3 Documentation des fonctions membres 86
4.31.3.1 ecrire 86
4.31.3.2 GetFormat 86
4.31.3.3 GetFormatFichierPrelevement 86
4.31.3.4 SetFormat 86
4.31.3.5 SetFormatFichierPrelevement 86
4.32 Référence de la classe FormCageExcitation 86

4.32.1 Description détaillée 88
4.32.2 Documentation des constructeurs et destructeur 88
4.32.2.1 FormCageExcitation 88
4.32.3 Documentation des fonctions membres 88
4.32.3.1 actualiseFormulaire 88
4.32.3.2 changeCreate 88
4.32.3.3 manageFormulaire 88
4.32.3.4 setCageExcitation 88
4.32.3.5 valider 88
4.33 Référence de la classe FormCalculDirective 89
4.33.1 Description détaillée 89
4.33.2 Documentation des constructeurs et destructeur 89
4.33.2.1 FormCalculDirective 89
4.33.3 Documentation des fonctions membres 89
4.33.3.1 init 89
4.33.3.2 setCaculDirective 89
4.34 Référence de la classe FormCartographieTemporelle 90
4.34.1 Description détaillée 91
4.34.2 Documentation des constructeurs et destructeur 91
4.34.2.1 FormCartographieTemporelle 91
4.34.3 Documentation des fonctions membres 91
4.34.3.1 reset 91
4.34.3.2 valider 91
4.35 Référence de la classe FormChampLointain 92
4.35.1 Description détaillée 92
4.35.2 Documentation des constructeurs et destructeur 92
4.35.2.1 FormChampLointain 92
4.35.3 Documentation des fonctions membres 92
4.35.3.1 init 92
4.35.3.2 setChampLointain 92
4.36 Référence de la classe FormCompressionHuygens 92
4.36.1 Description détaillée 93
4.36.2 Documentation des constructeurs et destructeur 93
4.36.2.1 FormCompressionHuygens 93

4.36.3 Documentation des fonctions membres 93
4.36.3.1 init 93
4.36.3.2 setCompressHuy 93
4.37 Référence de la classe FormCoordonnees 93
4.37.1 Description détaillée 94
4.37.2 Documentation des constructeurs et destructeur 94
4.37.2.1 FormCoordonnees 94
4.37.2.2 FormCoordonnees 94
4.37.3 Documentation des fonctions membres 94
4.37.3.1 changeX 94
4.37.3.2 changeY 94
4.37.3.3 changeZ 94
4.37.3.4 coordonneeChanged 95
4.37.3.5 getCoord 95
4.37.3.6 setCoordonnee 95
4.38 Référence de la classe FormEchantillonage 95
4.38.1 Description détaillée 95
4.38.2 Documentation des constructeurs et destructeur 95
4.38.2.1 FormEchantillonage 95
4.38.3 Documentation des fonctions membres 95
4.38.3.1 init 95
4.38.3.2 setEchantillonnage 96
4.39 Référence de la classe FormElementBase 96
4.39.1 Description détaillée 98
4.39.2 Documentation des constructeurs et destructeur 98
4.39.2.1 FormElementBase 98
4.39.3 Documentation des fonctions membres 98
4.39.3.1 changeCouleur 98
4.39.3.2 elementValide 98
4.39.3.3 getWidgetElementBase 98
4.39.3.4 init 98
4.39.3.5 reset 99
4.39.3.6 setElement 99
4.39.3.7 valider 99

4.39.4 Documentation des données membres 99
4.39.4.1 boutonValider 99
4.39.4.2 couleurPick 99
4.39.4.3 formCoord1 99
4.39.4.4 formCoord2 99
4.40 Référence de la classe FormElementLocalise 100
4.40.1 Description détaillée 101
4.40.2 Documentation des constructeurs et destructeur 101
4.40.2.1 FormElementLocalise 102
4.40.3 Documentation des fonctions membres 102
4.40.3.1 reset 102
4.40.3.2 valider 102
4.41 Référence de la classe FormExcitation 102
4.41.1 Description détaillée 102
4.41.2 Documentation des constructeurs et destructeur 102
4.41.2.1 FormExcitation 103
4.41.3 Documentation des fonctions membres 103
4.41.3.1 init 103
4.41.3.2 setExcitation 103
4.42 Référence de la classe FormFormatStockage 103
4.42.1 Description détaillée 103
4.42.2 Documentation des constructeurs et destructeur 103
4.42.2.1 FormFormatStockage 103
4.42.3 Documentation des fonctions membres 104
4.42.3.1 init 104
4.42.3.2 setFormatStockage 104
4.43 Référence de la classe FormMetallisation 104
4.43.1 Description détaillée 105
4.43.2 Documentation des constructeurs et destructeur 105
4.43.2.1 FormMetallisation 106
4.43.3 Documentation des fonctions membres 106
4.43.3.1 getMetallisation 106
4.43.3.2 reset 106
4.43.3.3 valider 106

4.44 Référence de la classe FormNouveau 106
4.44.1 Description détaillée 107
4.44.2 Documentation des constructeurs et destructeur 107
4.44.2.1 FormNouveau 107
4.44.3 Documentation des fonctions membres 107
4.44.3.1 getNum 107
4.44.3.2 setNum 107
4.44.3.3 testFichierExistant 107
4.45 Référence de la classe FormOuvrir 107
4.45.1 Description détaillée 108
4.45.2 Documentation des constructeurs et destructeur 108
4.45.2.1 FormOuvrir 108
4.45.3 Documentation des fonctions membres 108
4.45.3.1 getNum 108
4.45.3.2 setNum 108
4.45.3.3 testFichierExistant 108
4.46 Référence de la classe FormParallelepipede 108
4.46.1 Description détaillée 110
4.46.2 Documentation des constructeurs et destructeur 110
4.46.2.1 FormParallelepipede 110
4.46.3 Documentation des fonctions membres 110
4.46.3.1 reset 110
4.46.3.2 valider 110
4.47 Référence de la classe FormParamExcitation 110
4.47.1 Description détaillée 111
4.47.2 Documentation des constructeurs et destructeur 111
4.47.2.1 FormParamExcitation 111
4.47.3 Documentation des fonctions membres 111
4.47.3.1 afficheSinus 111
4.47.3.2 init 111
4.47.3.3 setParamExcit 111
4.48 Référence de la classe FormParoi 111
4.48.1 Description détaillée 112
4.48.2 Documentation des constructeurs et destructeur 112

4.48.2.1 FormParoi 112
4.48.3 Documentation des fonctions membres 112
4.48.3.1 init 112
4.48.3.2 setParoi 112
4.49 Référence de la classe FormPML 112
4.49.1 Description détaillée 113
4.49.2 Documentation des constructeurs et destructeur 113
4.49.2.1 FormPML 113
4.49.3 Documentation des fonctions membres 113
4.49.3.1 init 113
4.49.3.2 setPML 113
4.50 Référence de la classe FormPortExcitation 113
4.50.1 Description détaillée 115
4.50.2 Documentation des constructeurs et destructeur 115
4.50.2.1 FormPortExcitation 115
4.50.3 Documentation des fonctions membres 115
4.50.3.1 reset 115
4.50.3.2 valider 115
4.51 Référence de la classe FormSettings 115
4.51.1 Description détaillée 116
4.51.2 Documentation des constructeurs et destructeur 116
4.51.2.1 FormSettings 116
4.51.3 Documentation des fonctions membres 116
4.51.3.1 appliquer 116
4.51.3.2 ouvreDialogDossier 116
4.52 Référence de la classe FormSonde 116
4.52.1 Description détaillée 118
4.52.2 Documentation des constructeurs et destructeur 118
4.52.2.1 FormSonde 118
4.52.3 Documentation des fonctions membres 118
4.52.3.1 reset 118
4.52.3.2 valider 118
4.53 Référence de la classe FormSurfacePrelevement 118
4.53.1 Description détaillée 120

4.53.2 Documentation des constructeurs et destructeur 120
4.53.2.1 FormSurfacePrelevement 120
4.53.3 Documentation des fonctions membres 120
4.53.3.1 activeResteForm 120
4.53.3.2 reset 120
4.53.3.3 valider 120
4.54 Référence de la classe FormVolumeCalcul 121
4.54.1 Description détaillée 121
4.54.2 Documentation des constructeurs et destructeur 121
4.54.2.1 FormVolumeCalcul 121
4.54.3 Documentation des fonctions membres 121
4.54.3.1 init 121
4.54.3.2 setVolumeCalcul 121
4.55 Référence de la classe MainWindow 121
4.55.1 Description détaillée 122
4.55.2 Documentation des constructeurs et destructeur 122
4.55.2.1 MainWindow 122
4.55.2.2 ∼MainWindow 122
4.55.3 Documentation des fonctions membres 122
4.55.3.1 changeFilter 122
4.55.3.2 ecrire 122
4.55.3.3 nouvelleSimulation 123
4.55.3.4 ouvreParametres 123
4.55.3.5 ouvrirSimulation 123
4.56 Référence de la classe metallisation 123
4.56.1 Description détaillée 124
4.56.2 Documentation des constructeurs et destructeur 124
4.56.2.1 metallisation 124
4.56.2.2 ∼metallisation 124
4.56.3 Documentation des fonctions membres 125
4.56.3.1 ecrire 125
4.56.3.2 GetConductivite 125
4.56.3.3 SetConductivite 125
4.57 Référence de la classe MyGLWidget 125

4.57.1 Description détaillée 126
4.57.2 Documentation des constructeurs et destructeur 126
4.57.2.1 MyGLWidget 126
4.57.3 Documentation des fonctions membres 126
4.57.3.1 initializeGL 126
4.57.3.2 paintGL 126
4.57.3.3 resizeGL 126
4.57.3.4 sizeHint 126
4.57.3.5 timeOutSlot 126
4.58 Référence de la classe MySortFilterProxyModel 127
4.58.1 Description détaillée 127
4.58.2 Documentation des constructeurs et destructeur 127
4.58.2.1 MySortFilterProxyModel 127
4.58.3 Documentation des fonctions membres 127
4.58.3.1 filterAcceptsRow 127
4.58.3.2 lessThan 127
4.58.3.3 setFilterKeyColumn 127
4.59 Référence de la classe ObjetFDTDWidget 128
4.59.1 Description détaillée 129
4.59.2 Documentation des constructeurs et destructeur 129
4.59.2.1 ObjetFDTDWidget 130
4.59.3 Documentation des fonctions membres 130
4.59.3.1 init 130
4.59.3.2 newCartoTempoCreated 130
4.59.3.3 newSondeCreated 130
4.59.3.4 newSurfacePrelCreated 130
4.59.3.5 openCartoTempoWindows 130
4.60 Référence de la classe parallelepipede 130
4.60.1 Description détaillée 132
4.60.2 Documentation des constructeurs et destructeur 132
4.60.2.1 parallelepipede 132
4.60.2.2 ∼parallelepipede 132
4.60.3 Documentation des fonctions membres 132
4.60.3.1 ecrire 132

4.60.3.2 GetConductivite 132
4.60.3.3 GetPermeRelative 132
4.60.3.4 GetPermitRelative 132
4.60.3.5 SetConductivite 132
4.60.3.6 SetPermeRelative 133
4.60.3.7 SetPermitRelative 133
4.61 Référence de la classe ParamAvanceWidget 133
4.61.1 Description détaillée 134
4.61.2 Documentation des constructeurs et destructeur 134
4.61.2.1 ParamAvanceWidget 134
4.61.3 Documentation des fonctions membres 134
4.61.3.1 init 134
4.62 Référence de la classe ParamExcitations 135
4.62.1 Description détaillée 136
4.62.2 Documentation des constructeurs et destructeur 136
4.62.2.1 ParamExcitations 136
4.62.2.2 ∼ParamExcitations 136
4.62.3 Documentation des fonctions membres 136
4.62.3.1 ecrire 136
4.62.3.2 GetFrequenceMax 136
4.62.3.3 GetFrequenceSinus 136
4.62.3.4 GetType 137
4.62.3.5 SetFrequenceMax 137
4.62.3.6 SetFrequenceSinus 137
4.62.3.7 SetType 137
4.63 Référence de la classe ParamSimuWidget 137
4.63.1 Description détaillée 138
4.63.2 Documentation des constructeurs et destructeur 138
4.63.2.1 ParamSimuWidget 138
4.63.3 Documentation des fonctions membres 139
4.63.3.1 init 139
4.63.3.2 newPortExcitationCreated 139
4.64 Référence de la classe paroi 139
4.64.1 Description détaillée 140

4.64.2 Documentation des constructeurs et destructeur 140
4.64.2.1 paroi 141
4.64.2.2 ∼paroi 141
4.64.3 Documentation des fonctions membres 141
4.64.3.1 ecrire 141
4.64.3.2 GetParoiInferieure 141
4.64.3.3 GetParoiSuperieure 141
4.64.3.4 GetParoiX 141
4.64.3.5 GetParoiY 141
4.64.3.6 paroiInfChanged 141
4.64.3.7 paroiSupChanged 141
4.64.3.8 paroiXChanged 141
4.64.3.9 paroiYChanged 141
4.64.3.10 SetParoiInferieure 141
4.64.3.11 SetParoiSuperieure 142
4.64.3.12 SetParoiX 142
4.64.3.13 SetParoiY 142
4.65 Référence de la classe Parser 142
4.65.1 Description détaillée 142
4.65.2 Documentation des constructeurs et destructeur 142
4.65.2.1 Parser 142
4.65.3 Documentation des fonctions membres 143
4.65.3.1 getEmplacementFichier 143
4.65.3.2 getNumStru 143
4.65.3.3 parse 143
4.65.3.4 setEmplacementFichier 143
4.65.3.5 setNumFichier 143
4.66 Référence de la classe pml 143
4.66.1 Description détaillée 145
4.66.2 Documentation des constructeurs et destructeur 145
4.66.2.1 pml 145
4.66.2.2 ∼pml 145
4.66.3 Documentation des fonctions membres 145
4.66.3.1 ecrire 145

4.66.3.2 GetEpaisseurCouche 145
4.66.3.3 GetKMax 145
4.66.3.4 GetOrdreVariation 145
4.66.3.5 GetSigmaMax 145
4.66.3.6 SetEpaisseurCouche 145
4.66.3.7 SetKMax 146
4.66.3.8 SetOrdreVariation 146
4.66.3.9 SetSigmaMax 146
4.67 Référence de la classe Point 146
4.67.1 Description détaillée 146
4.67.2 Documentation des constructeurs et destructeur 146
4.67.2.1 Point 146
4.67.2.2 Point 147
4.67.3 Documentation des données membres 147
4.67.3.1 x 147
4.67.3.2 y 147
4.67.3.3 z 147
4.68 Référence de la classe portExcitation 147
4.68.1 Description détaillée 148
4.68.2 Documentation des constructeurs et destructeur 149
4.68.2.1 portExcitation 149
4.68.2.2 ∼portExcitation 149
4.68.3 Documentation des fonctions membres 149
4.68.3.1 ecrire 149
4.68.3.2 GetDirection 149
4.68.3.3 GetImpedance 149
4.68.3.4 GetPonderationAmplitude 149
4.68.3.5 GetPonderationPhase 149
4.68.3.6 GetTypePort 149
4.68.3.7 GetTypeSource 149
4.68.3.8 IsPortActif 149
4.68.3.9 IsPortInfinitesimal 150
4.68.3.10 SetDirection 150
4.68.3.11 SetImpedance 150

4.68.3.12 SetPonderationAmplitude 150
4.68.3.13 SetPonderationPhase 150
4.68.3.14 SetPortActif 150
4.68.3.15 SetPortInfinitesimal 150
4.68.3.16 SetTypePort 150
4.68.3.17 SetTypeSource 150
4.69 Référence de la classe sonde 151
4.69.1 Description détaillée 152
4.69.2 Documentation des constructeurs et destructeur 152
4.69.2.1 sonde 152
4.69.2.2 ∼sonde 152
4.69.3 Documentation des fonctions membres 152
4.69.3.1 ecrire 152
4.69.3.2 GetPointApplication 152
4.69.3.3 IsEx 153
4.69.3.4 IsEy 153
4.69.3.5 IsEz 153
4.69.3.6 IsHx 153
4.69.3.7 IsHy 153
4.69.3.8 IsHz 153
4.69.3.9 IsValeurAuCentre 153
4.69.3.10 SetEx 153
4.69.3.11 SetEy 153
4.69.3.12 SetEz 153
4.69.3.13 SetHx 154
4.69.3.14 SetHy 154
4.69.3.15 SetHz 154
4.69.3.16 SetPointApplication 154
4.69.3.17 SetPointApplication 154
4.69.3.18 SetValeurAuCentre 154
4.70 Référence de la classe surfacePrelevement 154
4.70.1 Description détaillée 156
4.70.2 Documentation des constructeurs et destructeur 156
4.70.2.1 surfacePrelevement 156

4.70.2.2 surfacePrelevement 156
4.70.2.3 ∼surfacePrelevement 156
4.70.3 Documentation des fonctions membres 156
4.70.3.1 ecrire 156
4.70.3.2 GetCompressionHauteur 156
4.70.3.3 GetCompressionLargeur 156
4.70.3.4 GetCompressionLongueur 157
4.70.3.5 GetInsideOutside 157
4.70.3.6 GetNbFaces 157
4.70.3.7 GetNomFichier 157
4.70.3.8 GetTypeSurface 157
4.70.3.9 SetCompressionHauteur 157
4.70.3.10 SetCompressionLargeur 157
4.70.3.11 SetCompressionLongueur 157
4.70.3.12 SetInsideOutside 157
4.70.3.13 SetNbFaces 158
4.70.3.14 SetNomFichier 158
4.70.3.15 SetTypeSurface 158
4.71 Référence de la classe TableModel 158
4.71.1 Description détaillée 160
4.71.2 Documentation des constructeurs et destructeur 160
4.71.2.1 TableModel 160
4.71.3 Documentation des fonctions membres 160
4.71.3.1 addElement 160
4.71.3.2 flags 160
4.71.3.3 getElement 160
4.71.3.4 getHeaders 161
4.71.3.5 headerData 161
4.71.3.6 initHeaders 161
4.71.3.7 removeElement 161
4.71.3.8 rowCount 161
4.72 Référence de la classe TableModelCartoTempo 161
4.72.1 Description détaillée 163
4.72.2 Documentation des énumérations membres 163

4.72.2.1 Columns 163
4.72.3 Documentation des constructeurs et destructeur 163
4.72.3.1 TableModelCartoTempo 163
4.72.4 Documentation des fonctions membres 163
4.72.4.1 columnCount 163
4.72.4.2 data 163
4.72.4.3 setData 163
4.73 Référence de la classe TableModelElementLocalise 164
4.73.1 Description détaillée 165
4.73.2 Documentation des énumérations membres 165
4.73.2.1 Columns 165
4.73.3 Documentation des constructeurs et destructeur 165
4.73.3.1 TableModelElementLocalise 165
4.73.4 Documentation des fonctions membres 165
4.73.4.1 columnCount 165
4.73.4.2 data 165
4.73.4.3 setData 166
4.74 Référence de la classe TableModelMetallisation 166
4.74.1 Description détaillée 167
4.74.2 Documentation des énumérations membres 167
4.74.2.1 Columns 167
4.74.3 Documentation des constructeurs et destructeur 167
4.74.3.1 TableModelMetallisation 167
4.74.4 Documentation des fonctions membres 167
4.74.4.1 columnCount 167
4.74.4.2 data 167
4.74.4.3 setData 168
4.75 Référence de la classe TableModelParallelepipede 168
4.75.1 Description détaillée 169
4.75.2 Documentation des énumérations membres 169
4.75.2.1 Columns 169
4.75.3 Documentation des constructeurs et destructeur 169
4.75.3.1 TableModelParallelepipede 169
4.75.4 Documentation des fonctions membres 169

4.75.4.1 columnCount 169
4.75.4.2 data 170
4.75.4.3 setData 170
4.76 Référence de la classe TableModelPortExcitation 170
4.76.1 Description détaillée 171
4.76.2 Documentation des énumérations membres 171
4.76.2.1 Columns 171
4.76.3 Documentation des constructeurs et destructeur 172
4.76.3.1 TableModelPortExcitation 172
4.76.4 Documentation des fonctions membres 172
4.76.4.1 columnCount 172
4.76.4.2 data 172
4.76.4.3 setData 172
4.77 Référence de la classe TableModelSonde 172
4.77.1 Description détaillée 174
4.77.2 Documentation des énumérations membres 174
4.77.2.1 Columns 174
4.77.3 Documentation des constructeurs et destructeur 174
4.77.3.1 TableModelSonde 174
4.77.4 Documentation des fonctions membres 174
4.77.4.1 columnCount 174
4.77.4.2 data 174
4.77.4.3 setData 175
4.78 Référence de la classe TableModelSurfacePrelev 175
4.78.1 Description détaillée 176
4.78.2 Documentation des énumérations membres 176
4.78.2.1 Columns 176
4.78.3 Documentation des constructeurs et destructeur 176
4.78.3.1 TableModelSurfacePrelev 176
4.78.4 Documentation des fonctions membres 177
4.78.4.1 columnCount 177
4.78.4.2 data 177
4.78.4.3 flags 177
4.78.4.4 setData 177

4.79 Référence de la classe TableView 177
4.79.1 Description détaillée 179
4.79.2 Documentation des constructeurs et destructeur 179
4.79.2.1 TableView 179
4.79.3 Documentation des fonctions membres 179
4.79.3.1 addElement 179
4.79.3.2 addNewElement 179
4.79.3.3 contextMenuEvent 180
4.79.3.4 deleteElement 180
4.79.3.5 editColor 180
4.79.3.6 getSourceModel 180
4.79.3.7 keyPressEvent 180
4.79.3.8 selectedIndexes 180
4.79.3.9 selectionChanged 181
4.79.3.10 setSourceModel 181
4.80 Référence de la classe TableViewCartTempo 181
4.80.1 Description détaillée 182
4.80.2 Documentation des constructeurs et destructeur 182
4.80.2.1 TableViewCartTempo 182
4.80.3 Documentation des fonctions membres 182
4.80.3.1 addNewElement 182
4.81 Référence de la classe TableViewElementLocalise 183
4.81.1 Description détaillée 184
4.81.2 Documentation des constructeurs et destructeur 184
4.81.2.1 TableViewElementLocalise 184
4.81.3 Documentation des fonctions membres 184
4.81.3.1 addNewElement 184
4.82 Référence de la classe TableViewMetallisation 184
4.82.1 Description détaillée 185
4.82.2 Documentation des constructeurs et destructeur 186
4.82.2.1 TableViewMetallisation 186
4.82.3 Documentation des fonctions membres 186
4.82.3.1 addNewElement 186
4.83 Référence de la classe TableViewParallelepipede 186

4.83.1 Description détaillée 187
4.83.2 Documentation des constructeurs et destructeur 187
4.83.2.1 TableViewParallelepipede 187
4.83.3 Documentation des fonctions membres 187
4.83.3.1 addNewElement 187
4.84 Référence de la classe TableViewPortExcitation 188
4.84.1 Description détaillée 189
4.84.2 Documentation des constructeurs et destructeur 189
4.84.2.1 TableViewPortExcitation 189
4.84.3 Documentation des fonctions membres 189
4.84.3.1 addNewElement 189
4.85 Référence de la classe TableViewSonde 189
4.85.1 Description détaillée 190
4.85.2 Documentation des constructeurs et destructeur 191
4.85.2.1 TableViewSonde 191
4.85.3 Documentation des fonctions membres 191
4.85.3.1 addNewElement 191
4.86 Référence de la classe TableViewSurfacePrelev 191
4.86.1 Description détaillée 192
4.86.2 Documentation des constructeurs et destructeur 192
4.86.2.1 TableViewSurfacePrelev 192
4.86.3 Documentation des fonctions membres 192
4.86.3.1 addNewElement 192
4.87 Référence de la classe ToolBoxWidget 193
4.87.1 Description détaillée 194
4.87.2 Documentation des constructeurs et destructeur 194
4.87.2.1 ToolBoxWidget 194
4.87.3 Documentation des fonctions membres 194
4.87.3.1 init 195
4.87.3.2 setData 195
4.87.4 Documentation des données membres 195
4.87.4.1 data 195
4.88 Référence de la classe Visualisation3D 195
4.88.1 Description détaillée 197

4.88.2 Documentation des constructeurs et destructeur 197
4.88.2.1 Visualisation3D 197
4.88.3 Documentation des fonctions membres 197
4.88.3.1 ajoutListElement 197
4.88.3.2 clearScene 197
4.88.3.3 coordonneeToPoint 197
4.88.3.4 dessineCageExcitation 198
4.88.3.5 dessineScene 198
4.88.3.6 dessineVolumeCalcul 198
4.88.3.7 init 198
4.88.3.8 initializeGL 198
4.88.3.9 keyPressEvent 198
4.88.3.10 mouseMoveEvent 198
4.88.3.11 mousePressEvent 199
4.88.3.12 mouseReleaseEvent 199
4.88.3.13 normalizeAngle 199
4.88.3.14 paintGL 199
4.88.3.15 reInit 199
4.88.3.16 resizeGL 199
4.88.3.17 setCageExcitation 199
4.88.3.18 setVolumeCalcul 199
4.88.3.19 setXRotation 199
4.88.3.20 setYRotation 200
4.88.3.21 setZoom 200
4.88.3.22 startToZoom11 200
4.88.3.23 stopTimer 200
4.88.3.24 toZero 200
4.88.3.25 toZeroAngle 200
4.88.3.26 wheelEvent 201
4.88.3.27 zoomChanged 201
4.89 Référence de la classe volumeCalcul 201
4.89.1 Description détaillée 202
4.89.2 Documentation des constructeurs et destructeur 203
4.89.2.1 volumeCalcul 203

4.89.2.2 ∼volumeCalcul 203
4.89.3 Documentation des fonctions membres 203
4.89.3.1 ecrire 203
4.89.3.2 GetHauteur 203
4.89.3.3 GetLargeur 203
4.89.3.4 GetLongueur 203
4.89.3.5 GetNombreX 203
4.89.3.6 GetNombreY 203
4.89.3.7 GetNombreZ 203
4.89.3.8 hauteurChanged 204
4.89.3.9 largeurChanged 204
4.89.3.10 longueurChanged 204
4.89.3.11 nbXChanged 204
4.89.3.12 nbYChanged 204
4.89.3.13 nbZChanged 204
4.89.3.14 SetHauteur 204
4.89.3.15 SetLargeur 204
4.89.3.16 SetLongueur 204
4.89.3.17 SetNombreX 204
4.89.3.18 SetNombreY 204
4.89.3.19 SetNombreZ 204

5 Documentation des fichiers 205

5.1 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/analyseTemporelle.cpp 205
5.2 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/analyseTemporelle.h 206
5.3 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocCartographieTemporelle.cpp 206
5.4 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocCartographieTemporelle.h 207
5.5 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocConfiguration.cpp 208
5.6 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocConfiguration.h 208

5.7 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocEchantillonnage.cpp 209
5.8 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocEchantillonnage.h 210
5.9 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/BlocElementBase.cpp 211
5.10 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/BlocElementBase.h 212
5.11 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocElementLocalise.cpp 213
5.12 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocElementLocalise.h 213
5.13 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocMetallisation.cpp 214
5.14 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocMetallisation.h 215
5.15 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocParallelepipede.cpp 216
5.16 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocParallelepipede.h 216
5.17 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocPortExcitation.cpp 217
5.18 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocPortExcitation.h 218
5.19 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSonde.cpp 219
5.20 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSonde.h . 219
5.21 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSurfacePrelevement.cpp 220
5.22 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSurfacePrelevement.h 221
5.23 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cageExcitation.cpp 222
5.24 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cageExcitation.h 223
5.25 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/CalculChampsLointainWidget.cpp 224
5.26 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/CalculChampsLointainWidget.h 224
5.27 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/calculDirectivite.cpp 225

5.28 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/calcul-Directivite.h 226
5.29 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cartographie-Temporelle.cpp 226
5.30 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cartographie-Temporelle.h 227
5.31 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/champ-Lointain.cpp 228
5.32 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/champ-Lointain.h 229
5.33 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ColorPicker.cpp 230
5.34 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ColorPicker.h . 231
5.35 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Combox-DecaleeDelegate.cpp 232
5.36 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Combox-DecaleeDelegate.h 232
5.37 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/compression-Huygens.cpp 233
5.38 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/compression-Huygens.h 234
5.39 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/coordonnee.cpp235
5.40 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/coordonnee.h . 236
5.41 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Coordonnee-Delegate.cpp 237
5.42 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Coordonnee-Delegate.h 237
5.43 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cube.cpp . . . 239
5.44 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cube.h 239
5.45 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Data.cpp . . . 240
5.46 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Data.h 240
5.47 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Description-GeoWidget.cpp 241
5.48 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Description-GeoWidget.h 242
5.49 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Ecriture.cpp . 242
5.50 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Ecriture.h . . . 243
5.51 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/element-Base.cpp 244

5.52 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/elementBase.h 244

5.53 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/elementLocalise.cpp 245

5.54 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/elementLocalise.h 245

5.55 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Excitation.cpp 246

5.56 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/Excitation.h . . 247

5.57 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FilterWidget.cpp248

5.58 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FilterWidget.h 249

5.59 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormAnalyseTempo.cpp 250

5.60 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormAnalyseTempo.h 251

5.61 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/formatStockage.cpp 252

5.62 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/formatStockage.h 253

5.63 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCageExcitation.cpp 254

5.64 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCageExcitation.h 255

5.65 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCalculDirective.cpp 256

5.66 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCalculDirective.h 257

5.67 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCartographieTemporelle.cpp 258

5.68 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCartographieTemporelle.h 259

5.69 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormChampLointain.cpp 260

5.70 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormChampLointain.h 261

5.71 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCompressionHuygens.cpp 262

5.72 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCompressionHuygens.h 262

5.73 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCoordonnees.cpp 263

5.74 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormCoordonnees.h . . . 264
5.75 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormEchantillonage.cpp . . . 265
5.76 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormEchantillonage.h . . . 266
5.77 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormElementBase.cpp . . . 267
5.78 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormElementBase.h . . . 268
5.79 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormElementLocalise.cpp . . . 269
5.80 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormElementLocalise.h . . . 269
5.81 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormExcitation.cpp . . . 270
5.82 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormExcitation.h . . . 271
5.83 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormFormatStockage.cpp . . . 272
5.84 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormFormatStockage.h . . . 273
5.85 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormMetallisation.cpp . . . 274
5.86 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormMetallisation.h . . . 275
5.87 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormNouveau.cpp . . . 276
5.88 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormNouveau.h 276
5.89 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormOuvrir.cpp 277
5.90 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormOuvrir.h . 278
5.91 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormParallelepipede.cpp . . . 279
5.92 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormParallelepipede.h . . . 280
5.93 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormParamExcitation.cpp . . . 280
5.94 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormParamExcitation.h . . . 281
5.95 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormParoi.cpp 282

5.96 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormParoi.h . 283
5.97 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormPML.cpp 285
5.98 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormPML.h . . 285
5.99 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormPortExcitation.cpp 287
5.100 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormPortExcitation.h 287
5.101 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormSettings.cpp 288
5.102 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormSettings.h 289
5.103 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormSonde.cpp289
5.104 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormSonde.h . 290
5.105 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormSurfacePrelevement.cpp 291
5.106 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormSurfacePrelevement.h 291
5.107 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormVolumeCalcul.cpp 292
5.108 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/FormVolumeCalcul.h 293
5.109 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/main.cpp . . . 294
5.109.1 Documentation des fonctions 295
5.109.1.1 main 295
5.110 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/mainwindow.cpp295
5.111 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/mainwindow.h 295
5.112 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/metallisation.cpp296
5.113 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/metallisation.h 297
5.114 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/myglwidget.cpp 297
5.115 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/myglwidget.h . 298
5.116 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/MySortFilterProxyModel.cpp 299
5.117 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/MySortFilterProxyModel.h 300
5.118 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ObjetFDTDWidget.cpp 300
5.119 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ObjetFDTDWidget.h 301

5.120 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parallelepipede.cpp 302

5.121 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parallelepipede.h 302

5.122 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ParamAvanceWidget.cpp . . . 303

5.123 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ParamAvanceWidget.h . . . 304

5.124 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ParamExcitations.cpp . . . 305

5.125 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ParamExcitations.h . . . 305

5.126 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ParamSimuWidget.cpp . . . 306

5.127 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ParamSimuWidget.h . . . 307

5.128 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/paroi.cpp . . . 308

5.129 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/paroi.h . . . 308

5.130 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parser.cpp . . . 309

5.131 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parser.h . . . 310

5.131.1 Documentation des macros . . . 310

5.131.1.1 TAILLE_CHAINE . . . 310

5.131.1.2 TAILLE_NOM_FICHIER . . . 310

5.132 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/pml.cpp . . . 311

5.133 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/pml.h . . . 312

5.134 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/point.cpp . . . 313

5.135 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/point.h . . . 313

5.136 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/portExcitation.cpp . . . 314

5.137 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/portExcitation.h 315

5.138 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/sonde.cpp . . . 316

5.139 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/sonde.h . . . 316

5.140 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/surfacePrelevement.cpp . . . 317

5.141 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/surfacePrelevement.h . . . 318

5.142 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModel.cpp 319

5.143 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModel.h . 320

5.144 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelCartoTempo.cpp 320
5.145 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelCartoTempo.h 321
5.146 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelElementLocalise.cpp 322
5.147 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelElementLocalise.h 323
5.148 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelMetallisation.cpp 324
5.149 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelMetallisation.h 325
5.150 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelParallelepipede.cpp 326
5.151 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelParallelepipede.h 327
5.152 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelPortExcitation.cpp 328
5.153 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelPortExcitation.h 328
5.154 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelSonde.cpp 329
5.155 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelSonde.h 330
5.156 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelSurfacePrelev.cpp 331
5.157 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableModelSurfacePrelev.h 332
5.158 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView.cpp 333
5.159 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView.h . . 334
5.160 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableViewCartTempo.cpp 335
5.161 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableViewCartTempo.h 335
5.162 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableViewElementLocalise.cpp 336
5.163 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableViewElementLocalise.h 336
5.164 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableViewMetallisation.cpp 337

5.165 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-Metallisation.h 338

5.166 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-Parallelepipede.cpp 338

5.167 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-Parallelepipede.h 339

5.168 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-PortExcitation.cpp 340

5.169 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-PortExcitation.h 340

5.170 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-Sonde.cpp 341

5.171 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-Sonde.h 341

5.172 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-SurfacePrelev.cpp 342

5.173 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/TableView-SurfacePrelev.h 343

5.174 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ToolBox-Widget.cpp 344

5.175 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/ToolBox-Widget.h 344

5.176 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/visualisation3d.cpp 345

5.177 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/visualisation3d.h 345

5.178 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/volume-Calcul.cpp 346

5.179 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/volumeCalcul.h 347

Chapitre 1

Index des classes

1.1 Hiérarchie des classes

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

- blocConfiguration 19
  - analyseTemporelle 15
  - blocEchantillonnage 22
  - BlocElementBase 24
    - blocCartographieTemporelle 17
    - blocElementLocalise 27
    - blocMetallisation 29
    - blocParallelepipede 31
    - blocPortExcitation 33
    - blocSonde 35
    - blocSurfacePrelevement 37
  - cageExcitation 40
  - calculDirectivite 47
  - champLointain 52
  - compressionHuygens 59
  - Excitation 80
  - formatStockage 84
  - ParamExcitations 135
  - paroi 139
  - pml 143
  - volumeCalcul 201
- ColorPicker 56
- ComboxDecaleeDelegate 58
- coordonnee 62
- CoordonneeDelegate 64
- Cube 65
- Data 66
- Ecriture 72
- elementBase 73

cageExcitation . . . 40
cartographieTemporelle . . . 49
elementLocalise . . . 77
metallisation . . . 123
parallelepipede . . . 130
portExcitation . . . 147
sonde . . . 151
surfacePrelevement . . . 154
FilterWidget . . . 82
FormAnalyseTempo . . . 83
FormCalculDirective . . . 89
FormChampLointain . . . 92
FormCompressionHuygens . . . 92
FormCoordonnees . . . 93
FormEchantillonage . . . 95
FormElementBase . . . 96
FormCageExcitation . . . 86
FormCartographieTemporelle . . . 90
FormElementLocalise . . . 100
FormMetallisation . . . 104
FormParallelepipede . . . 108
FormPortExcitation . . . 113
FormSonde . . . 116
FormSurfacePrelevement . . . 118
FormExcitation . . . 102
FormFormatStockage . . . 103
FormNouveau . . . 106
FormOuvrir . . . 107
FormParamExcitation . . . 110
FormParoi . . . 111
FormPML . . . 112
FormSettings . . . 115
FormVolumeCalcul . . . 121
MainWindow . . . 121
MyGLWidget . . . 125
Visualisation3D . . . 195
MySortFilterProxyModel . . . 127
Parser . . . 142
Point . . . 146
TableModel . . . 158
TableModelCartoTempo . . . 161
TableModelElementLocalise . . . 164
TableModelMetallisation . . . 166
TableModelParallelepipede . . . 168
TableModelPortExcitation . . . 170
TableModelSonde . . . 172
TableModelSurfacePrelev . . . 175
TableView . . . 177
TableViewCartTempo . . . 181

TableViewElementLocalise 183
TableViewMetallisation 184
TableViewParallelepipede 186
TableViewPortExcitation 188
TableViewSonde 189
TableViewSurfacePrelev 191
ToolBoxWidget 193
CalculChampsLointainWidget 45
DescriptionGeoWidget 69
ObjetFDTDWidget 128
ParamAvanceWidget 133
ParamSimuWidget 137

Chapitre 2

Index des classes

2.1 Liste des classes

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

analyseTemporelle
Cette classe représente la variable **analyseTemporelle** (p. 15) d'une simulation 15

blocCartographieTemporelle
Cette classe représente une liste de **cartographieTemporelle** (p. 49) 17

blocConfiguration
Classe de base pour les données de configuration d'une simulation 19

blocEchantillonnage
Cette classe représente la variable echantillonnage d'une simulation 22

BlocElementBase
Représente une liste d'**elementBase** (p. 73) 24

blocElementLocalise
Cette classe représente une liste de **elementLocalise** (p. 77) 27

blocMetallisation
Cette classe représente une liste de metallisation 29

blocParallelepipede
Cette classe représente une liste de parallelepipede 31

blocPortExcitation
Cette classe représente une liste de **portExcitation** (p. 147) 33

blocSonde
Cette classe représente une liste de sonde 35

blocSurfacePrelevement
Cette classe représente une liste de **surfacePrelevement** (p. 154) . 37

cageExcitation
Cette classe représente la variable cageExcitations d'une simulation 40

CalculChampsLointainWidget
Cette classe représente le formulaire pour modifier un Champs-Lointain 45

calculDirectivite
Cette classe représente la variable **calculDirectivite** (p. 47) d'une simulation 47
cartographieTemporelle
Cette classe représente la variable **cartographieTemporelle** (p. 49) d'une simulation 49
champLointain
Cette classe représente la variable **champLointain** (p. 52) d'une simulation 52
ColorPicker
Cette classe est un widget sous forme de bouton qui permet de sélectionner une couleur dans une boite de dialogue qui s'ouvre quand le bouton est pressé. Un carré de la couleur sélectionnée est présent sur le bouton 56
ComboxDecaleeDelegate
Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur sous forme de liste déroulante 58
compressionHuygens
Cette classe représente la variable **compressionHuygens** (p. 59) d'une simulation 59
coordonnee
Cette classe représente une coordonnee en 3 dimensions. Elle sert d'unité de position dans la scène 3D 62
CoordonneeDelegate
Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur pour modifier une coordonnees. Cette éditeur est en faite un **FormCoordonnees** (p. 93) . . . 64
Cube 65
Data
Contient toutes les données utiles pour la simulation 66
DescriptionGeoWidget
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie 69
Ecriture 72
elementBase
The **elementBase** (p. 73) class 73
elementLocalise
Cette classe représente la variable **elementLocalise** (p. 77) d'une simulation 77
Excitation
Cette classe représente la variable **Excitation** (p. 80) d'une simulation 80
FilterWidget
The **FilterWidget** (p. 82) class 82
FormAnalyseTempo
The **FormAnalyseTempo** (p. 83) class 83

formatStockage
Cette classe représente la variable **formatStockage** (p. 84) d'une simulation 84
FormCageExcitation
Formulaire permettant d'ajouter un objet **cageExcitation** (p. 40) . . 86
FormCalculDirective
Formulaire permettant de modifier l'objet **calculDirectivite** (p. 47) . 89
FormCartographieTemporelle
Formulaire permettant d'ajouter un nouvelle objet **cartographie-Temporelle** (p. 49) dans la scène 3D 90
FormChampLointain
Cette classe fournit un formulaire pour modifier l'objet **champ-Lointain** (p. 52) 92
FormCompressionHuygens
Cette classe fournit un formulaire pour modifier l'objet **compression-Huygens** (p. 59) 92
FormCoordonnees
Cette classe fournit un widget pour modifier l'objet coordonnee . . . 93
FormEchantillonage
Cette classe fournit un formulaire pour modifier l'objet **bloc-Echantillonnage** (p. 22) 95
FormElementBase
Cette classe sert de base aux formulaires qui modifie des objets **elementBase** (p. 73) 96
FormElementLocalise
Formulaire permettant d'ajouter un nouvelle objet **elementLocalise** (p. 77) dans la scène 3D 100
FormExcitation
Cette classe fournit un formulaire pour modifier l'objet **Excitation** (p. 80) 102
FormFormatStockage
Cette classe fournit un formulaire pour modifier l'objet **format-Stockage** (p. 84) 103
FormMetallisation
Formulaire permettant d'ajouter un nouvelle objet metallisation dans la scène 3D 104
FormNouveau
Cette classe est le formulaire pour ajouter une nouvelle simulation . 106
FormOuvrir
Cette classe est le formulaire pour ouvrir une nouvelle simulation . . 107
FormParallelepipede
Formulaire permettant d'ajouter un nouvelle objet parallelepipede dans la scène 3D 108
FormParamExcitation
Cette classe fournit un formulaire pour modifier l'objet **Param-Excitations** (p. 135) 110
FormParoi
Cette classe fournit un formulaire pour modifier l'objet paroi 111
FormPML
Cette classe fournit un formulaire pour modifier l'objet pml 112

FormPortExcitation
Formulaire permettant d'ajouter un nouvelle objet **portExcitation** (p. 147) dans la scène 3D 113
FormSettings
Cette classe est le formulaire pour modifier les paramètres du logiciel 115
FormSonde
Formulaire permettant d'ajouter un nouvelle objet sonde dans la scène 3D 116
FormSurfacePrelevement
Formulaire permettant d'ajouter un nouvelle objet **surface-Prelevement** (p. 154) dans la scène 3D 118
FormVolumeCalcul
Cette classe fournit un formulaire pour modifier l'objet **volumeCalcul** (p. 201) 121
MainWindow
The **MainWindow** (p. 121) class Cette classe représente la fenêtre principale du logiciel 121
metallisation
Cette classe représente la variable metallisation d'une simulation . . 123
MyGLWidget
Cette classe sert de base à la classe qui affiche le rendu en 3D avec OpenGL 125
MySortFilterProxyModel
Cette classe permet de personnalisé le tri et le filtrage dans les listes 127
ObjetFDTDWidget
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie 128
parallelepipede
Cette classe représente la variable parallelepipede d'une simulation 130
ParamAvanceWidget
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie 133
ParamExcitations
Cette classe représente la variable **ParamExcitations** (p. 135) d'une simulation 135
ParamSimuWidget
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie 137
paroi
Cette classe représente la variable paroi d'une simulation 139
Parser
Le parser de fichier de configuration 142
pml
Cette classe représente la variable pml d'une simulation 143
Point
Cette classe représente un point dans un espace en 3D. Elle est utilisé pour afficher les **Cube** (p. 65) dans la vue 3D 146

portExcitation
Cette classe représente la variable **portExcitation** (p. 147) d'une simulation 147
sonde
Cette classe représente la variable sonde d'une simulation 151
surfacePrelevement
Cette classe représente la variable **surfacePrelevement** (p. 154) d'une simulation 154
TableModel
Cette classe sert de model de base pour les éléments des tableaux 158
TableModelCartoTempo 161
TableModelElementLocalise 164
TableModelMetallisation 166
TableModelParallelepipede 168
TableModelPortExcitation 170
TableModelSonde 172
TableModelSurfacePrelev 175
TableView
Cette classe sert de vue de base pour les tableaux de données. Elle affiche les données contenu dans son model sous forme de tableau 177
TableViewCartTempo 181
TableViewElementLocalise 183
TableViewMetallisation 184
TableViewParallelepipede 186
TableViewPortExcitation 188
TableViewSonde 189
TableViewSurfacePrelev 191
ToolBoxWidget
Classe de base pour les widgets conteneurs de la ToolBox 193
Visualisation3D
Affiche et gère la scène 3D 195
volumeCalcul
Cette classe représente la variable **volumeCalcul** (p. 201) d'une simulation 201

Chapitre 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

/home/kuiderlix/Prog/cpp/pskl-etprat/**analyseTemporelle.cpp** 205
/home/kuiderlix/Prog/cpp/pskl-etprat/**analyseTemporelle.h** 206
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocCartographieTemporelle.cpp** . . . 206
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocCartographieTemporelle.h** 207
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocConfiguration.cpp** 208
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocConfiguration.h** 208
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocEchantillonnage.cpp** 209
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocEchantillonnage.h** 210
/home/kuiderlix/Prog/cpp/pskl-etprat/**BlocElementBase.cpp** 211
/home/kuiderlix/Prog/cpp/pskl-etprat/**BlocElementBase.h** 212
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocElementLocalise.cpp** 213
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocElementLocalise.h** 213
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocMetallisation.cpp** 214
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocMetallisation.h** 215
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocParallelepipede.cpp** 216
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocParallelepipede.h** 216
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocPortExcitation.cpp** 217
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocPortExcitation.h** 218
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocSonde.cpp** 219
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocSonde.h** 219
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocSurfacePrelevement.cpp** 220
/home/kuiderlix/Prog/cpp/pskl-etprat/**blocSurfacePrelevement.h** 221
/home/kuiderlix/Prog/cpp/pskl-etprat/**cageExcitation.cpp** 222
/home/kuiderlix/Prog/cpp/pskl-etprat/**cageExcitation.h** 223
/home/kuiderlix/Prog/cpp/pskl-etprat/**CalculChampsLointainWidget.cpp** . . 224
/home/kuiderlix/Prog/cpp/pskl-etprat/**CalculChampsLointainWidget.h** 224
/home/kuiderlix/Prog/cpp/pskl-etprat/**calculDirectivite.cpp** 225
/home/kuiderlix/Prog/cpp/pskl-etprat/**calculDirectivite.h** 226
/home/kuiderlix/Prog/cpp/pskl-etprat/**cartographieTemporelle.cpp** 226

/home/kuiderlix/Prog/cpp/pskl-etprat/**cartographieTemporelle.h** 227
/home/kuiderlix/Prog/cpp/pskl-etprat/**champLointain.cpp** 228
/home/kuiderlix/Prog/cpp/pskl-etprat/**champLointain.h** 229
/home/kuiderlix/Prog/cpp/pskl-etprat/**ColorPicker.cpp** 230
/home/kuiderlix/Prog/cpp/pskl-etprat/**ColorPicker.h** 231
/home/kuiderlix/Prog/cpp/pskl-etprat/**ComboxDecaleeDelegate.cpp** 232
/home/kuiderlix/Prog/cpp/pskl-etprat/**ComboxDecaleeDelegate.h** 232
/home/kuiderlix/Prog/cpp/pskl-etprat/**compressionHuygens.cpp** 233
/home/kuiderlix/Prog/cpp/pskl-etprat/**compressionHuygens.h** 234
/home/kuiderlix/Prog/cpp/pskl-etprat/**coordonnee.cpp** 235
/home/kuiderlix/Prog/cpp/pskl-etprat/**coordonnee.h** 236
/home/kuiderlix/Prog/cpp/pskl-etprat/**CoordonneeDelegate.cpp** 237
/home/kuiderlix/Prog/cpp/pskl-etprat/**CoordonneeDelegate.h** 237
/home/kuiderlix/Prog/cpp/pskl-etprat/**cube.cpp** 239
/home/kuiderlix/Prog/cpp/pskl-etprat/**cube.h** 239
/home/kuiderlix/Prog/cpp/pskl-etprat/**Data.cpp** 240
/home/kuiderlix/Prog/cpp/pskl-etprat/**Data.h** 240
/home/kuiderlix/Prog/cpp/pskl-etprat/**DescriptionGeoWidget.cpp** 241
/home/kuiderlix/Prog/cpp/pskl-etprat/**DescriptionGeoWidget.h** 242
/home/kuiderlix/Prog/cpp/pskl-etprat/**Ecriture.cpp** 242
/home/kuiderlix/Prog/cpp/pskl-etprat/**Ecriture.h** 243
/home/kuiderlix/Prog/cpp/pskl-etprat/**elementBase.cpp** 244
/home/kuiderlix/Prog/cpp/pskl-etprat/**elementBase.h** 244
/home/kuiderlix/Prog/cpp/pskl-etprat/**elementLocalise.cpp** 245
/home/kuiderlix/Prog/cpp/pskl-etprat/**elementLocalise.h** 245
/home/kuiderlix/Prog/cpp/pskl-etprat/**Excitation.cpp** 246
/home/kuiderlix/Prog/cpp/pskl-etprat/**Excitation.h** 247
/home/kuiderlix/Prog/cpp/pskl-etprat/**FilterWidget.cpp** 248
/home/kuiderlix/Prog/cpp/pskl-etprat/**FilterWidget.h** 249
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormAnalyseTempo.cpp** 250
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormAnalyseTempo.h** 251
/home/kuiderlix/Prog/cpp/pskl-etprat/**formatStockage.cpp** 252
/home/kuiderlix/Prog/cpp/pskl-etprat/**formatStockage.h** 253
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCageExcitation.cpp** 254
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCageExcitation.h** 255
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCalculDirective.cpp** 256
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCalculDirective.h** 257
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCartographieTemporelle.cpp** . . 258
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCartographieTemporelle.h** 259
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormChampLointain.cpp** 260
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormChampLointain.h** 261
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCompressionHuygens.cpp** . . . 262
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCompressionHuygens.h** 262
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCoordonnees.cpp** 263
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormCoordonnees.h** 264
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormEchantillonage.cpp** 265
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormEchantillonage.h** 266
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementBase.cpp** 267
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementBase.h** 268
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementLocalise.cpp** 269

/home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementLocalise.h** 269
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormExcitation.cpp** 270
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormExcitation.h** 271
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormFormatStockage.cpp** 272
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormFormatStockage.h** 273
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormMetallisation.cpp** 274
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormMetallisation.h** 275
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormNouveau.cpp** 276
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormNouveau.h** 276
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormOuvrir.cpp** 277
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormOuvrir.h** 278
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormParallelepipede.cpp** 279
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormParallelepipede.h** 280
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormParamExcitation.cpp** 280
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormParamExcitation.h** 281
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormParoi.cpp** 282
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormParoi.h** 283
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormPML.cpp** 285
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormPML.h** 285
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormPortExcitation.cpp** 287
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormPortExcitation.h** 287
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormSettings.cpp** 288
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormSettings.h** 289
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormSonde.cpp** 289
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormSonde.h** 290
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormSurfacePrelevement.cpp** 291
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormSurfacePrelevement.h** 291
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormVolumeCalcul.cpp** 292
/home/kuiderlix/Prog/cpp/pskl-etprat/**FormVolumeCalcul.h** 293
/home/kuiderlix/Prog/cpp/pskl-etprat/**main.cpp** 294
/home/kuiderlix/Prog/cpp/pskl-etprat/**mainwindow.cpp** 295
/home/kuiderlix/Prog/cpp/pskl-etprat/**mainwindow.h** 295
/home/kuiderlix/Prog/cpp/pskl-etprat/**metallisation.cpp** 296
/home/kuiderlix/Prog/cpp/pskl-etprat/**metallisation.h** 297
/home/kuiderlix/Prog/cpp/pskl-etprat/**myglwidget.cpp** 297
/home/kuiderlix/Prog/cpp/pskl-etprat/**myglwidget.h** 298
/home/kuiderlix/Prog/cpp/pskl-etprat/**MySortFilterProxyModel.cpp** 299
/home/kuiderlix/Prog/cpp/pskl-etprat/**MySortFilterProxyModel.h** 300
/home/kuiderlix/Prog/cpp/pskl-etprat/**ObjetFDTDWidget.cpp** 300
/home/kuiderlix/Prog/cpp/pskl-etprat/**ObjetFDTDWidget.h** 301
/home/kuiderlix/Prog/cpp/pskl-etprat/**parallelepipede.cpp** 302
/home/kuiderlix/Prog/cpp/pskl-etprat/**parallelepipede.h** 302
/home/kuiderlix/Prog/cpp/pskl-etprat/**ParamAvanceWidget.cpp** 303
/home/kuiderlix/Prog/cpp/pskl-etprat/**ParamAvanceWidget.h** 304
/home/kuiderlix/Prog/cpp/pskl-etprat/**ParamExcitations.cpp** 305
/home/kuiderlix/Prog/cpp/pskl-etprat/**ParamExcitations.h** 305
/home/kuiderlix/Prog/cpp/pskl-etprat/**ParamSimuWidget.cpp** 306
/home/kuiderlix/Prog/cpp/pskl-etprat/**ParamSimuWidget.h** 307
/home/kuiderlix/Prog/cpp/pskl-etprat/**paroi.cpp** 308
/home/kuiderlix/Prog/cpp/pskl-etprat/**paroi.h** 308

/home/kuiderlix/Prog/cpp/pskl-etprat/**parser.cpp** 309
/home/kuiderlix/Prog/cpp/pskl-etprat/**parser.h** 310
/home/kuiderlix/Prog/cpp/pskl-etprat/**pml.cpp** 311
/home/kuiderlix/Prog/cpp/pskl-etprat/**pml.h** 312
/home/kuiderlix/Prog/cpp/pskl-etprat/**point.cpp** 313
/home/kuiderlix/Prog/cpp/pskl-etprat/**point.h** 313
/home/kuiderlix/Prog/cpp/pskl-etprat/**portExcitation.cpp** 314
/home/kuiderlix/Prog/cpp/pskl-etprat/**portExcitation.h** 315
/home/kuiderlix/Prog/cpp/pskl-etprat/**sonde.cpp** 316
/home/kuiderlix/Prog/cpp/pskl-etprat/**sonde.h** 316
/home/kuiderlix/Prog/cpp/pskl-etprat/**surfacePrelevement.cpp** 317
/home/kuiderlix/Prog/cpp/pskl-etprat/**surfacePrelevement.h** 318
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModel.cpp** 319
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModel.h** 320
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelCartoTempo.cpp** 320
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelCartoTempo.h** 321
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelElementLocalise.cpp** . . . 322
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelElementLocalise.h** 323
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelMetallisation.cpp** 324
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelMetallisation.h** 325
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelParallelepipede.cpp** 326
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelParallelepipede.h** 327
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelPortExcitation.cpp** 328
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelPortExcitation.h** 328
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSonde.cpp** 329
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSonde.h** 330
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSurfacePrelev.cpp** 331
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSurfacePrelev.h** 332
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableView.cpp** 333
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableView.h** 334
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewCartTempo.cpp** 335
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewCartTempo.h** 335
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewElementLocalise.cpp** 336
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewElementLocalise.h** 336
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewMetallisation.cpp** 337
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewMetallisation.h** 338
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewParallelepipede.cpp** 338
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewParallelepipede.h** 339
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewPortExcitation.cpp** 340
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewPortExcitation.h** 340
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSonde.cpp** 341
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSonde.h** 341
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSurfacePrelev.cpp** 342
/home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSurfacePrelev.h** 343
/home/kuiderlix/Prog/cpp/pskl-etprat/**ToolBoxWidget.cpp** 344
/home/kuiderlix/Prog/cpp/pskl-etprat/**ToolBoxWidget.h** 344
/home/kuiderlix/Prog/cpp/pskl-etprat/**visualisation3d.cpp** 345
/home/kuiderlix/Prog/cpp/pskl-etprat/**visualisation3d.h** 345
/home/kuiderlix/Prog/cpp/pskl-etprat/**volumeCalcul.cpp** 346
/home/kuiderlix/Prog/cpp/pskl-etprat/**volumeCalcul.h** 347

Chapitre 4

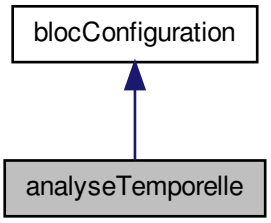
Documentation des classes

4.1 Référence de la classe analyseTemporelle

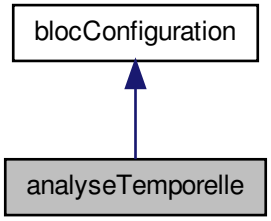
Cette classe représente la variable **analyseTemporelle** (p. 15) d'une simulation.

```
#include <analyseTemporelle.h>
```

Graphe d'héritage de analyseTemporelle :



Graphe de collaboration de analyseTemporelle :



Connecteurs publics

- void **SetPeriode** (double periode)
- void **SetTemps** (double temps)
- void **ecrire** ()

Fonctions membres publiques

- **analyseTemporelle** ()
- virtual ~**analyseTemporelle** ()
- double **GetPeriode** () const
- double **GetTemps** () const

4.1.1 Description détaillée

Cette classe représente la variable **analyseTemporelle** (p. 15) d'une simulation.

Définition à la ligne 16 du fichier analyseTemporelle.h.

4.1.2 Documentation des constructeurs et destructeur

4.1.2.1 analyseTemporelle : :analyseTemporelle ( )

Définition à la ligne 10 du fichier analyseTemporelle.cpp.

4.1.2.2 analyseTemporelle : :~analyseTemporelle ( ) [virtual]

Définition à la ligne 16 du fichier analyseTemporelle.cpp.

4.1.3 Documentation des fonctions membres

4.1.3.1 void analyseTemporelle : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 19 du fichier analyseTemporelle.cpp.

4.1.3.2 double analyseTemporelle : :GetPeriode ( ) const [inline]

Définition à la ligne 22 du fichier analyseTemporelle.h.

4.1.3.3 double analyseTemporelle : :GetTemps ( ) const [inline]

Définition à la ligne 26 du fichier analyseTemporelle.h.

4.1.3.4 void analyseTemporelle : :SetPeriode (double *periode*) [inline, slot]

Définition à la ligne 32 du fichier analyseTemporelle.h.

4.1.3.5 void analyseTemporelle : :SetTemps (double *temps*) [inline, slot]

Définition à la ligne 36 du fichier analyseTemporelle.h.

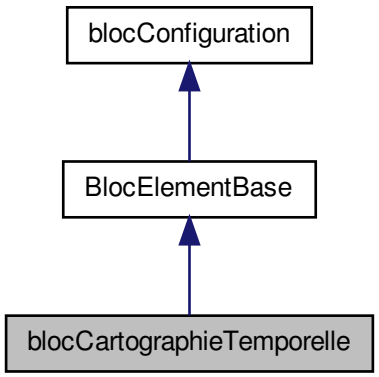
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**analyseTemporelle.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**analyseTemporelle.cpp**

4.2 Référence de la classe blocCartographieTemporelle

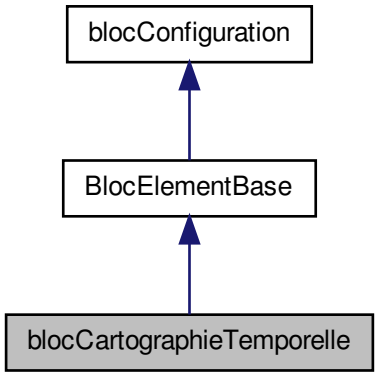
Cette classe représente une liste de **cartographieTemporelle** (p. 49).

```
#include <blocCartographieTemporelle.h>
```

Graphe d'héritage de blocCartographieTemporelle :



Graphe de collaboration de blocCartographieTemporelle :



Fonctions membres publiques

- **blocCartographieTemporelle** ()
- virtual ∼**blocCartographieTemporelle** ()
- **cartographieTemporelle** ∗ **GetCarto** (int i)

– void **ecrire** ()

4.2.1 Description détaillée

Cette classe représente une liste de **cartographieTemporelle** (p. 49).

Définition à la ligne 17 du fichier blocCartographieTemporelle.h.

4.2.2 Documentation des constructeurs et destructeur

4.2.2.1 blocCartographieTemporelle : :blocCartographieTemporelle ( )

Définition à la ligne 13 du fichier blocCartographieTemporelle.cpp.

4.2.2.2 blocCartographieTemporelle : :∼blocCartographieTemporelle ( ) [virtual]

Définition à la ligne 17 du fichier blocCartographieTemporelle.cpp.

4.2.3 Documentation des fonctions membres

4.2.3.1 void blocCartographieTemporelle : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 20 du fichier blocCartographieTemporelle.cpp.

4.2.3.2 cartographieTemporelle∗ blocCartographieTemporelle : :GetCarto (int *i*) [inline]

Définition à la ligne 24 du fichier blocCartographieTemporelle.h.

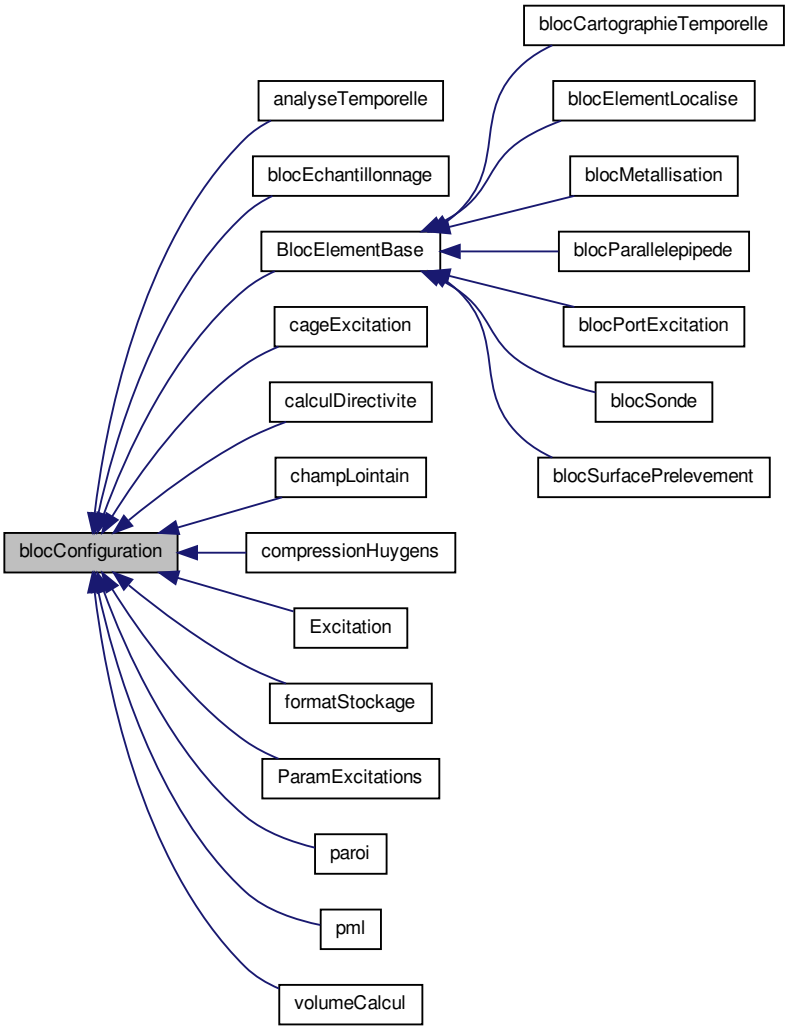
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocCartographieTemporelle.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocCartographieTemporelle.cpp**

4.3 Référence de la classe blocConfiguration

Classe de base pour les données de configuration d'une simulation.

```
#include <blocConfiguration.h>
```

Graphe d'héritage de blocConfiguration :



Fonctions membres publiques

- **blocConfiguration** ()
- virtual ~**blocConfiguration** ()
- std : :string **getHeader** () const
- void **setHeader** (std : :string **header**)
- void **setExtension** (std : :string **extension**)
- void **ecrire** ()

Attributs protégés

– std : :string **header**
– std : :string **extension**

4.3.1 Description détaillée

Classe de base pour les données de configuration d'une simulation.

Définition à la ligne 20 du fichier blocConfiguration.h.

4.3.2 Documentation des constructeurs et destructeur

4.3.2.1 blocConfiguration : :blocConfiguration ( )

Définition à la ligne 10 du fichier blocConfiguration.cpp.

4.3.2.2 blocConfiguration : :~blocConfiguration ( ) [virtual]

Définition à la ligne 14 du fichier blocConfiguration.cpp.

4.3.3 Documentation des fonctions membres

4.3.3.1 void blocConfiguration : :ecrire ( ) [inline]

Réimplémentée dans **cageExcitation** (p. 42), **champLointain** (p. 54), **volumeCalcul** (p. 203), **paroi** (p. 141), **pml** (p. 145), **compressionHuygens** (p. 61), **calculDirectivite** (p. 48), **ParamExcitations** (p. 136), **analyseTemporelle** (p. 17), **formatStockage** (p. 86), **blocSurfacePrelevement** (p. 39), **blocEchantillonnage** (p. 24), **Excitation** (p. 82), **blocParallelepipede** (p. 33), **blocCartographieTemporelle** (p. 19), **blocSonde** (p. 37), **blocMetallisation** (p. 31), **blocPortExcitation** (p. 35), et **blocElementLocalise** (p. 29).

Définition à la ligne 38 du fichier blocConfiguration.h.

4.3.3.2 std : :string blocConfiguration : :getHeader ( ) const [inline]

Définition à la ligne 26 du fichier blocConfiguration.h.

4.3.3.3 void blocConfiguration : :setExtension (std : :string *extension*) [inline]

Définition à la ligne 34 du fichier blocConfiguration.h.

4.3.3.4 void blocConfiguration : :setHeader (std : :string *header*) [inline]

Définition à la ligne 30 du fichier blocConfiguration.h.

4.3.4 Documentation des données membres

4.3.4.1 std : :string blocConfiguration : :extension [protected]

Définition à la ligne 50 du fichier blocConfiguration.h.

4.3.4.2 std : :string blocConfiguration : :header [protected]

Définition à la ligne 49 du fichier blocConfiguration.h.

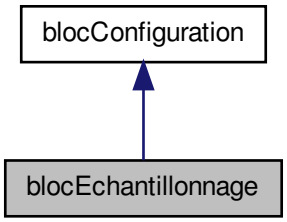
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**blocConfiguration.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**blocConfiguration.cpp**

4.4 Référence de la classe blocEchantillonnage

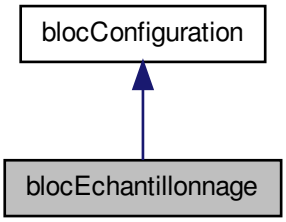
Cette classe représente la variable echantillonnage d'une simulation.

`#include <blocEchantillonnage.h>`

Graphe d'héritage de blocEchantillonnage :



Graphe de collaboration de blocEchantillonnage :



Connecteurs publics

- void **SetFacteurEchatillonnage** (int facteurEchatillonnage)
- void **ecrire** ()

Fonctions membres publiques

- **blocEchantillonnage** ()
- virtual ~**blocEchantillonnage** ()
- int **GetFacteurEchatillonnage** () const

4.4.1 Description détaillée

Cette classe représente la variable echantillonnage d'une simulation.

Définition à la ligne 16 du fichier blocEchantillonnage.h.

4.4.2 Documentation des constructeurs et destructeur

4.4.2.1 blocEchantillonnage : :blocEchantillonnage ( )

Définition à la ligne 10 du fichier blocEchantillonnage.cpp.

4.4.2.2 blocEchantillonnage : :~blocEchantillonnage ( ) [virtual]

Définition à la ligne 16 du fichier blocEchantillonnage.cpp.

4.4.3 Documentation des fonctions membres

4.4.3.1 void blocEchantillonnage : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 19 du fichier blocEchantillonnage.cpp.

4.4.3.2 int blocEchantillonnage : :GetFacteurEchatillonnage ( ) const [inline]

Définition à la ligne 22 du fichier blocEchantillonnage.h.

4.4.3.3 void blocEchantillonnage : :SetFacteurEchatillonnage (int *facteurEchatillonnage*) [inline, slot]

Définition à la ligne 28 du fichier blocEchantillonnage.h.

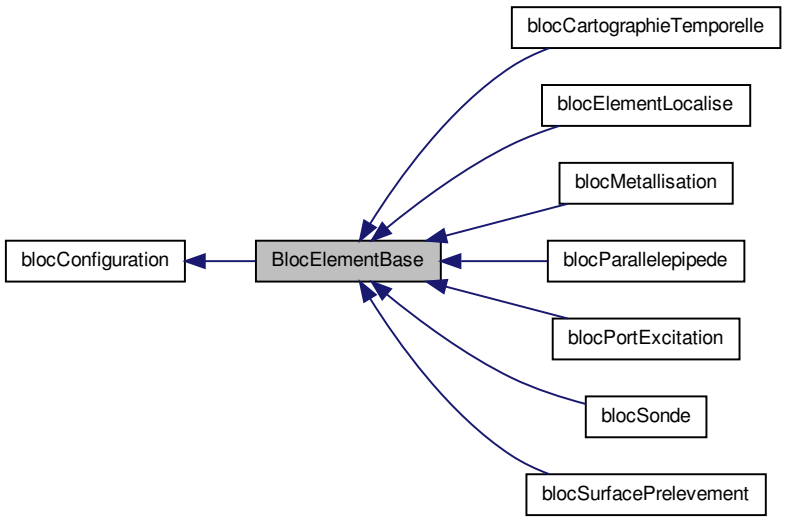
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**blocEchantillonnage.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**blocEchantillonnage.cpp**

4.5 Référence de la classe BlocElementBase

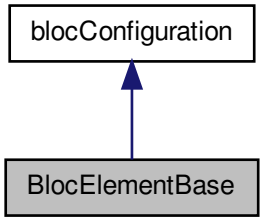
Représente une liste d'**elementBase** (p. 73).

`#include <BlocElementBase.h>`

Graphe d'héritage de BlocElementBase :



Graphe de collaboration de BlocElementBase :



Fonctions membres publiques

- **BlocElementBase** ()
- int **getNbElement** () const
  Retourne le nombre d'élément dans la liste.
- void **setElement** (int i, **elementBase** ∗elem)
  Remplace l'élément à l'indice i par elem.
- void **removeElement** (**elementBase** ∗elem)
  Supprime l'élément elem du tableau.
- void **removeElement** (int i)
  Supprime l'élément à l'indice i.
- void **addElement** (**elementBase** ∗elem)
  Ajoute l'élément elem à la fin de la liste.
- **elementBase** ∗ **getElement** (int i)
  *Retourne un **elementBase** (p. 73) contenu à l'indice i.*
- QList< **elementBase** ∗ > ∗ **getListElementPointer** ()
  Retourne un pointeur sur la liste des éléments.

Attributs protégés

- QList< **elementBase** ∗ > **listElement**

4.5.1 Description détaillée

Représente une liste d'**elementBase** (p. 73).

Définition à la ligne 10 du fichier BlocElementBase.h.

4.5.2 Documentation des constructeurs et destructeur

4.5.2.1 BlocElementBase : :BlocElementBase ( )

Définition à la ligne 3 du fichier BlocElementBase.cpp.

4.5.3 Documentation des fonctions membres

4.5.3.1 void BlocElementBase : :addElement (elementBase ∗ *elem*) [inline]

Ajoute l'élément elem à la fin de la liste.

Paramètres

elem	

Définition à la ligne 55 du fichier BlocElementBase.h.

4.5.3.2 elementBase∗ BlocElementBase : :getElement (int *i*) [inline]

Retourne un **elementBase** (p. 73) contenu à l'indice i.

Paramètres

i	

Renvoie

Définition à la ligne 63 du fichier BlocElementBase.h.

4.5.3.3 QList<elementBase∗>∗ BlocElementBase : :getListElementPointer ( ) [inline]

Retourne un pointeur sur la liste des éléments.

Renvoie

Définition à la ligne 72 du fichier BlocElementBase.h.

4.5.3.4 int BlocElementBase : :getNbElement ( ) const [inline]

Retourne le nombre d'élément dans la liste.

Renvoie

Définition à la ligne 19 du fichier BlocElementBase.h.

4.5.3.5 void BlocElementBase : :removeElement (elementBase ∗ *elem*) [inline]

Supprime l'élément elem du tableau.

Paramètres

elem	l'élément à supprimé

Définition à la ligne 37 du fichier BlocElementBase.h.

4.5.3.6 void BlocElementBase : :removeElement (int *i*) [inline]

Supprime l'élément à l'indice i.

Paramètres

i	

Définition à la ligne 46 du fichier BlocElementBase.h.

4.5.3.7 void BlocElementBase : :setElement (int *i*, elementBase ∗ *elem*) [inline]

Remplace l'élément à l'indice i par elem.

Paramètres

i	l'indice
elem	le nouveau élément

Définition à la ligne 28 du fichier BlocElementBase.h.

4.5.4 Documentation des données membres

4.5.4.1 QList<elementBase∗> BlocElementBase : :listElement [protected]

Définition à la ligne 78 du fichier BlocElementBase.h.

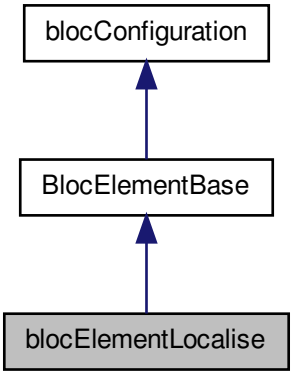
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**BlocElementBase.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**BlocElementBase.cpp**

4.6 Référence de la classe blocElementLocalise

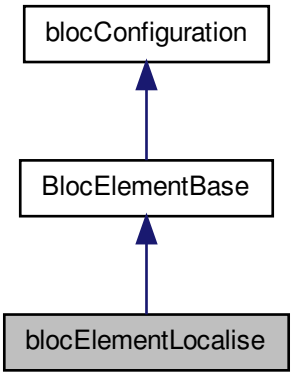
Cette classe représente une liste de **elementLocalise** (p. 77).

```
#include <blocElementLocalise.h>
```

Graphe d'héritage de blocElementLocalise :



Graphe de collaboration de blocElementLocalise :



Fonctions membres publiques

- **blocElementLocalise** ()
- virtual ~**blocElementLocalise** ()
- **elementLocalise** * **GetElementLocalise** (int i)

– void **ecrire** ()

4.6.1 Description détaillée

Cette classe représente une liste de **elementLocalise** (p. 77).

Définition à la ligne 17 du fichier blocElementLocalise.h.

4.6.2 Documentation des constructeurs et destructeur

4.6.2.1 blocElementLocalise : :blocElementLocalise ( )

Définition à la ligne 13 du fichier blocElementLocalise.cpp.

4.6.2.2 blocElementLocalise : :~blocElementLocalise ( ) [virtual]

Définition à la ligne 17 du fichier blocElementLocalise.cpp.

4.6.3 Documentation des fonctions membres

4.6.3.1 void blocElementLocalise : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 20 du fichier blocElementLocalise.cpp.

4.6.3.2 elementLocalise* blocElementLocalise : :GetElementLocalise (int *i*) [inline]

Définition à la ligne 22 du fichier blocElementLocalise.h.

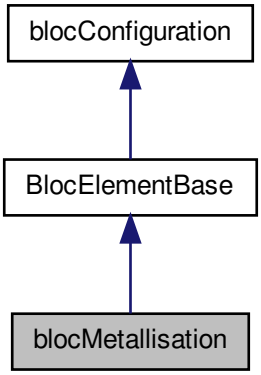
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocElementLocalise.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocElementLocalise.cpp**

4.7 Référence de la classe blocMetallisation

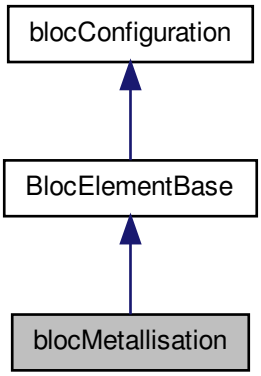
Cette classe représente une liste de metallisation.

```
#include <blocMetallisation.h>
```

Graphe d'héritage de blocMetallisation :



Graphe de collaboration de blocMetallisation :



Fonctions membres publiques

- **blocMetallisation** ()
- virtual ∼**blocMetallisation** ()
- **metallisation** ∗ **GetMetallisation** (int i)

– void **ecrire** ()

4.7.1 Description détaillée

Cette classe représente une liste de metallisation.

Définition à la ligne 17 du fichier blocMetallisation.h.

4.7.2 Documentation des constructeurs et destructeur

4.7.2.1 blocMetallisation : :blocMetallisation ( )

Définition à la ligne 12 du fichier blocMetallisation.cpp.

4.7.2.2 blocMetallisation : :∼blocMetallisation ( ) [virtual]

Définition à la ligne 15 du fichier blocMetallisation.cpp.

4.7.3 Documentation des fonctions membres

4.7.3.1 void blocMetallisation : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 18 du fichier blocMetallisation.cpp.

4.7.3.2 metallisation∗ blocMetallisation : :GetMetallisation (int *i*) [inline]

Définition à la ligne 22 du fichier blocMetallisation.h.

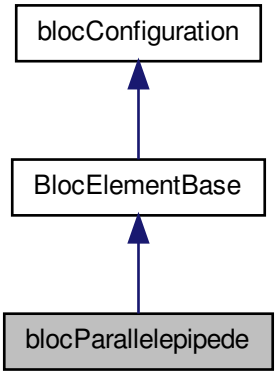
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocMetallisation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocMetallisation.cpp**

4.8 Référence de la classe blocParallelepipede

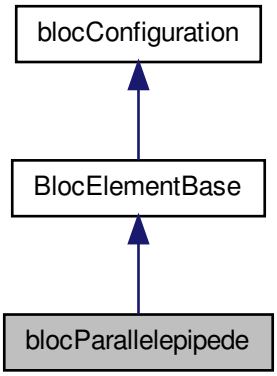
Cette classe représente une liste de parallelepipede.

```
#include <blocParallelepipede.h>
```

Graphe d'héritage de blocParallelepipede :



Graphe de collaboration de blocParallelepipede :



Fonctions membres publiques

- **blocParallelepipede** ()
- virtual ~**blocParallelepipede** ()
- **parallelepipede** * **getParallelepipede** (int i)

– void **ecrire** ()

4.8.1 Description détaillée

Cette classe représente une liste de parallelepipede.

Définition à la ligne 18 du fichier blocParallelepipede.h.

4.8.2 Documentation des constructeurs et destructeur

4.8.2.1 blocParallelepipede : :blocParallelepipede ( )

Définition à la ligne 10 du fichier blocParallelepipede.cpp.

4.8.2.2 blocParallelepipede : :∼blocParallelepipede ( ) [virtual]

Définition à la ligne 13 du fichier blocParallelepipede.cpp.

4.8.3 Documentation des fonctions membres

4.8.3.1 void blocParallelepipede : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 16 du fichier blocParallelepipede.cpp.

4.8.3.2 parallelepipede∗ blocParallelepipede : :getParallelepipede (int *i*) [inline]

Définition à la ligne 24 du fichier blocParallelepipede.h.

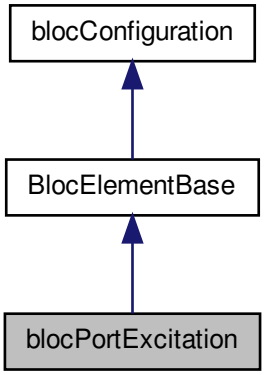
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocParallelepipede.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocParallelepipede.cpp**

4.9 Référence de la classe blocPortExcitation

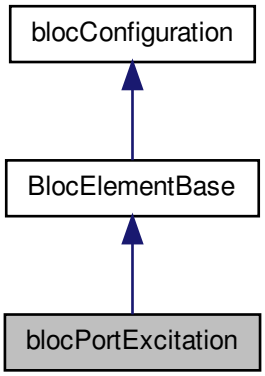
Cette classe représente une liste de **portExcitation** (p. 147).

```
#include <blocPortExcitation.h>
```

Graphe d'héritage de blocPortExcitation :



Graphe de collaboration de blocPortExcitation :



Fonctions membres publiques

- **blocPortExcitation** ()
- virtual ~**blocPortExcitation** ()
- **portExcitation** * **GetPortExcitation** (int i)

– void **ecrire** ()

4.9.1 Description détaillée

Cette classe représente une liste de **portExcitation** (p. 147).

Définition à la ligne 17 du fichier blocPortExcitation.h.

4.9.2 Documentation des constructeurs et destructeur

4.9.2.1 blocPortExcitation : :blocPortExcitation ( )

Définition à la ligne 12 du fichier blocPortExcitation.cpp.

4.9.2.2 blocPortExcitation : :~blocPortExcitation ( ) [virtual]

Définition à la ligne 15 du fichier blocPortExcitation.cpp.

4.9.3 Documentation des fonctions membres

4.9.3.1 void blocPortExcitation : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 18 du fichier blocPortExcitation.cpp.

4.9.3.2 portExcitation* blocPortExcitation : :GetPortExcitation (int *i*) [inline]

Définition à la ligne 23 du fichier blocPortExcitation.h.

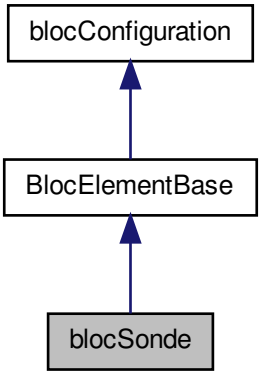
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocPortExcitation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocPortExcitation.cpp**

4.10 Référence de la classe blocSonde

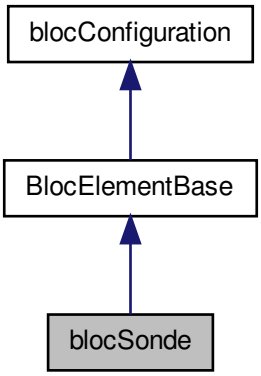
Cette classe représente une liste de sonde.

```
#include <blocSonde.h>
```

Graphe d'héritage de blocSonde :



Graphe de collaboration de blocSonde :



Fonctions membres publiques

- **blocSonde** ()
- virtual ∼**blocSonde** ()
- **sonde** ∗ **GetSonde** (int i)

– void **ecrire** ()

4.10.1 Description détaillée

Cette classe représente une liste de sonde.

Définition à la ligne 17 du fichier blocSonde.h.

4.10.2 Documentation des constructeurs et destructeur

4.10.2.1 blocSonde : :blocSonde ( )

Définition à la ligne 12 du fichier blocSonde.cpp.

4.10.2.2 blocSonde : :∼blocSonde ( ) [virtual]

Définition à la ligne 16 du fichier blocSonde.cpp.

4.10.3 Documentation des fonctions membres

4.10.3.1 void blocSonde : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 19 du fichier blocSonde.cpp.

4.10.3.2 sonde∗ blocSonde : :GetSonde (int *i*) [inline]

Définition à la ligne 23 du fichier blocSonde.h.

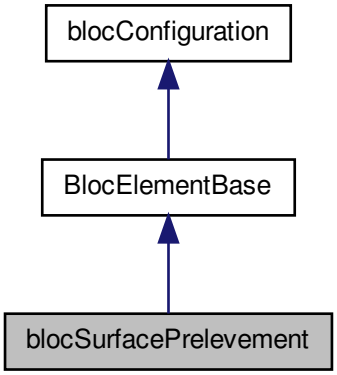
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocSonde.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocSonde.cpp**

4.11 Référence de la classe blocSurfacePrelevement

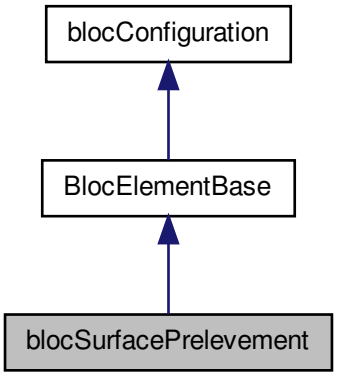
Cette classe représente une liste de **surfacePrelevement** (p. 154).

```
#include <blocSurfacePrelevement.h>
```

Graphe d'héritage de blocSurfacePrelevement :



Graphe de collaboration de blocSurfacePrelevement :



Fonctions membres publiques

- **blocSurfacePrelevement** ()
- virtual ~**blocSurfacePrelevement** ()
- int **GetNbSurfacesDG** () const

– void **SetNbSurfacesDG** (int nbSurfacesDG)
– **surfacePrelevement** ∗ **GetSurface** (int i)
– void **ecrire** ()

4.11.1 Description détaillée

Cette classe représente une liste de **surfacePrelevement** (p. 154).

Définition à la ligne 17 du fichier blocSurfacePrelevement.h.

4.11.2 Documentation des constructeurs et destructeur

4.11.2.1 blocSurfacePrelevement : :blocSurfacePrelevement ( )

Définition à la ligne 12 du fichier blocSurfacePrelevement.cpp.

4.11.2.2 blocSurfacePrelevement : :∼blocSurfacePrelevement ( ) [virtual]

Définition à la ligne 16 du fichier blocSurfacePrelevement.cpp.

4.11.3 Documentation des fonctions membres

4.11.3.1 void blocSurfacePrelevement : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 19 du fichier blocSurfacePrelevement.cpp.

4.11.3.2 int blocSurfacePrelevement : :GetNbSurfacesDG ( ) const [inline]

Définition à la ligne 22 du fichier blocSurfacePrelevement.h.

4.11.3.3 surfacePrelevement∗ blocSurfacePrelevement : :GetSurface (int *i*) [inline]

Définition à la ligne 31 du fichier blocSurfacePrelevement.h.

4.11.3.4 void blocSurfacePrelevement : :SetNbSurfacesDG (int *nbSurfacesDG*) [inline]

Définition à la ligne 26 du fichier blocSurfacePrelevement.h.

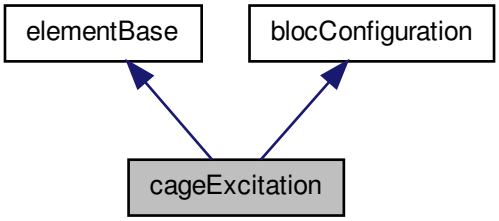
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocSurfacePrelevement.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**blocSurfacePrelevement.cpp**

4.12 Référence de la classe cageExcitation

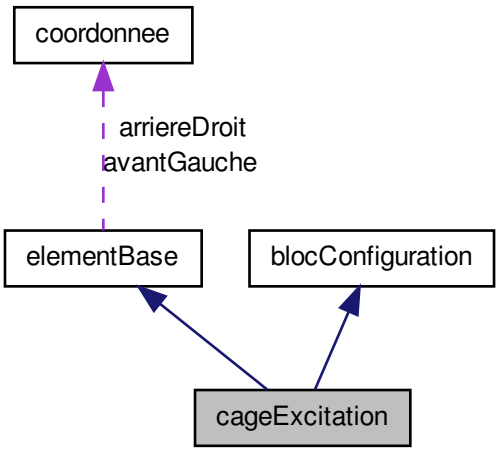
Cette classe représente la variable cageExcitations d'une simulation.

```
#include <cageExcitation.h>
```

Graphe d'héritage de cageExcitation :



Graphe de collaboration de cageExcitation :



Fonctions membres publiques

- **cageExcitation** ()
- virtual ∼**cageExcitation** ()
- int **GetHauteur** () const
- void **SetHauteur** (int hauteur)
- int **GetInsideOutside** () const
- void **SetInsideOutside** (int insideOutside)
- int **GetModeExcitation** () const
- void **SetModeExcitation** (int modeExcitation)
- float **GetModulationAmplitude** () const
- void **SetModulationAmplitude** (float modulationAmplitude)
- float **GetModulationPhase** () const
- void **SetModulationPhase** (float modulationPhase)
- int **GetNbFaces** () const
- void **SetNbFaces** (int nbFaces)
- float **GetPhi0** () const
- void **SetPhi0** (float phi0)
- **coordonnee GetPointReference** () const
- void **SetPointReference** (int x, int y, int z)
- void **SetPointReference** (**coordonnee** c)
- float **GetPsi0** () const
- void **SetPsi0** (float psi0)
- int **GetSensPropagation** () const
- void **SetSensPropagation** (int sensPropagation)
- float **GetTheta0** () const
- void **SetTheta0** (float theta0)
- int **GetTypeExcitation** () const
- void **SetTypeExcitation** (int typeExcitation)
- std : :string **GetNomFichier** () const
- void **SetNomFichier** (std : :string nomFichier)
- void **ecrire** ()
- bool **isCreate** ()
- void **setCreate** (bool c)

4.12.1 Description détaillée

Cette classe représente la variable cageExcitations d'une simulation.

Définition à la ligne 19 du fichier cageExcitation.h.

4.12.2 Documentation des constructeurs et destructeur

4.12.2.1 cageExcitation : :cageExcitation ( )

Définition à la ligne 10 du fichier cageExcitation.cpp.

4.12.2.2 cageExcitation : :∼cageExcitation ( ) [virtual]

Définition à la ligne 36 du fichier cageExcitation.cpp.

4.12.3 Documentation des fonctions membres

4.12.3.1 void cageExcitation : :ecrire ( )

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 39 du fichier cageExcitation.cpp.

4.12.3.2 int cageExcitation : :GetHauteur ( ) const [inline]

Définition à la ligne 25 du fichier cageExcitation.h.

4.12.3.3 int cageExcitation : :GetInsideOutside ( ) const [inline]

Définition à la ligne 33 du fichier cageExcitation.h.

4.12.3.4 int cageExcitation : :GetModeExcitation ( ) const [inline]

Définition à la ligne 41 du fichier cageExcitation.h.

4.12.3.5 float cageExcitation : :GetModulationAmplitude ( ) const [inline]

Définition à la ligne 49 du fichier cageExcitation.h.

4.12.3.6 float cageExcitation : :GetModulationPhase ( ) const [inline]

Définition à la ligne 57 du fichier cageExcitation.h.

4.12.3.7 int cageExcitation : :GetNbFaces ( ) const [inline]

Définition à la ligne 65 du fichier cageExcitation.h.

4.12.3.8 std : :string cageExcitation : :GetNomFichier ( ) const [inline]

Définition à la ligne 124 du fichier cageExcitation.h.

4.12.3.9 float cageExcitation : :GetPhi0 ( ) const [inline]

Définition à la ligne 73 du fichier cageExcitation.h.

4.12.3.10 coordonnee cageExcitation : :GetPointReference ( ) const [inline]

Définition à la ligne 81 du fichier cageExcitation.h.

4.12.3.11 float cageExcitation : :GetPsi0 ( ) const [inline]

Définition à la ligne 92 du fichier cageExcitation.h.

4.12.3.12 int cageExcitation : :GetSensPropagation ( ) const [inline]

Définition à la ligne 100 du fichier cageExcitation.h.

4.12.3.13 float cageExcitation : :GetTheta0 ( ) const [inline]

Définition à la ligne 108 du fichier cageExcitation.h.

4.12.3.14 int cageExcitation : :GetTypeExcitation ( ) const [inline]

Définition à la ligne 116 du fichier cageExcitation.h.

4.12.3.15 bool cageExcitation : :isCreate ( ) [inline]

Définition à la ligne 135 du fichier cageExcitation.h.

4.12.3.16 void cageExcitation : :setCreate (bool *c*) [inline]

Définition à la ligne 140 du fichier cageExcitation.h.

4.12.3.17 void cageExcitation : :SetHauteur (int *hauteur*) [inline]

Définition à la ligne 29 du fichier cageExcitation.h.

4.12.3.18 void cageExcitation : :SetInsideOutside (int *insideOutside*) [inline]

Définition à la ligne 37 du fichier cageExcitation.h.

4.12.3.19 void cageExcitation : :SetModeExcitation (int *modeExcitation*) [inline]

Définition à la ligne 45 du fichier cageExcitation.h.

4.12.3.20 void cageExcitation : :SetModulationAmplitude (float *modulationAmplitude*) [inline]

Définition à la ligne 53 du fichier cageExcitation.h.

4.12.3.21 void cageExcitation : :SetModulationPhase (float *modulationPhase*) [inline]

Définition à la ligne 61 du fichier cageExcitation.h.

4.12.3.22 void cageExcitation : :SetNbFaces (int *nbFaces*) [inline]

Définition à la ligne 69 du fichier cageExcitation.h.

4.12.3.23 void cageExcitation : :SetNomFichier (std : :string *nomFichier*) [inline]

Définition à la ligne 128 du fichier cageExcitation.h.

4.12.3.24 void cageExcitation : :SetPhi0 (float *phi0*) [inline]

Définition à la ligne 77 du fichier cageExcitation.h.

4.12.3.25 void cageExcitation : :SetPointReference (int *x*, int *y*, int *z*) [inline]

Définition à la ligne 85 du fichier cageExcitation.h.

4.12.3.26 void cageExcitation : :SetPointReference (coordonnee *c*) [inline]

Définition à la ligne 88 du fichier cageExcitation.h.

4.12.3.27 void cageExcitation : :SetPsi0 (float *psi0*) [inline]

Définition à la ligne 96 du fichier cageExcitation.h.

4.12.3.28 void cageExcitation : :SetSensPropagation (int *sensPropagation*) [inline]

Définition à la ligne 104 du fichier cageExcitation.h.

4.12.3.29 void cageExcitation : :SetTheta0 (float *theta0*) [inline]

Définition à la ligne 112 du fichier cageExcitation.h.

4.12.3.30 void cageExcitation : :SetTypeExcitation (int *typeExcitation*) [inline]

Définition à la ligne 120 du fichier cageExcitation.h.

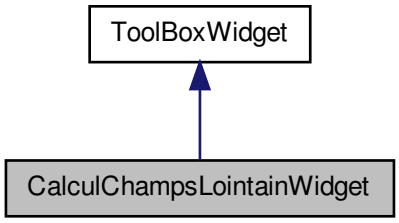
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**cageExcitation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**cageExcitation.cpp**

4.13 Référence de la classe CalculChampsLointainWidget

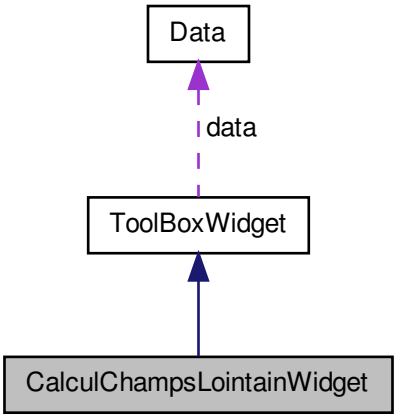
Cette classe représente le formulaire pour modifier un ChampsLointain.

```
#include <CalculChampsLointainWidget.h>
```

Graphe d'héritage de CalculChampsLointainWidget :



Graphe de collaboration de CalculChampsLointainWidget :



Fonctions membres publiques

- **CalculChampsLointainWidget** (**Data** ∗**data**, QWidget ∗parent=0)
- void **init** ()

  Cette fonction initialise les formulaires avec les données.

4.13.1 Description détaillée

Cette classe représente le formulaire pour modifier un ChampsLointain.

Définition à la ligne 14 du fichier CalculChampsLointainWidget.h.

4.13.2 Documentation des constructeurs et destructeur

4.13.2.1 CalculChampsLointainWidget : :CalculChampsLointainWidget (Data ∗ *data,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier CalculChampsLointainWidget.cpp.

4.13.3 Documentation des fonctions membres

4.13.3.1 void CalculChampsLointainWidget : :init ( ) [virtual]

Cette fonction initialise les formulaires avec les données.

Implémente **ToolBoxWidget** (p. 195).

Définition à la ligne 23 du fichier CalculChampsLointainWidget.cpp.

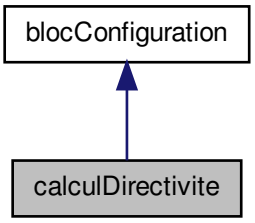
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**CalculChampsLointainWidget.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**CalculChampsLointainWidget.cpp**

4.14 Référence de la classe calculDirectivite

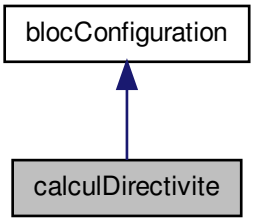
Cette classe représente la variable **calculDirectivite** (p. 47) d'une simulation.

```
#include <calculDirectivite.h>
```

Graphe d'héritage de calculDirectivite :



Graphe de collaboration de calculDirectivite :



Connecteurs publics

- void **SetCalcul** (bool calcul)
- void **SetPasPhi** (int pasPhi)
- void **SetPasTheta** (int pasTheta)
- void **ecrire** ()

Fonctions membres publiques

- **calculDirectivite** ()
- virtual ∼**calculDirectivite** ()
- bool **IsCalcul** () const
- int **GetPasPhi** () const
- int **GetPasTheta** () const

4.14.1 Description détaillée

Cette classe représente la variable **calculDirectivite** (p. 47) d'une simulation.

Définition à la ligne 16 du fichier calculDirectivite.h.

4.14.2 Documentation des constructeurs et destructeur

4.14.2.1 calculDirectivite : :calculDirectivite ( )

Définition à la ligne 10 du fichier calculDirectivite.cpp.

4.14.2.2 calculDirectivite : :∼calculDirectivite ( ) [virtual]

Définition à la ligne 16 du fichier calculDirectivite.cpp.

4.14.3 Documentation des fonctions membres

4.14.3.1 void calculDirectivite : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 19 du fichier calculDirectivite.cpp.

4.14.3.2 int calculDirectivite : :GetPasPhi ( ) const [inline]

Définition à la ligne 25 du fichier calculDirectivite.h.

4.14.3.3 int calculDirectivite : :GetPasTheta ( ) const [inline]

Définition à la ligne 28 du fichier calculDirectivite.h.

4.14.3.4 bool calculDirectivite : :IsCalcul ( ) const [inline]

Définition à la ligne 22 du fichier calculDirectivite.h.

4.14.3.5 void calculDirectivite : :SetCalcul (bool *calcul*) [inline, slot]

Définition à la ligne 35 du fichier calculDirectivite.h.

4.14.3.6 void calculDirectivite : :SetPasPhi (int *pasPhi*) [inline, slot]

Définition à la ligne 39 du fichier calculDirectivite.h.

4.14.3.7 void calculDirectivite : :SetPasTheta (int *pasTheta*) [inline, slot]

Définition à la ligne 43 du fichier calculDirectivite.h.

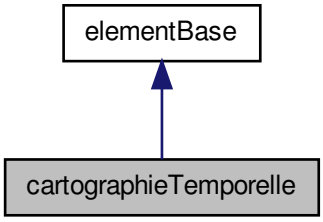
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**calculDirectivite.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**calculDirectivite.cpp**

4.15 Référence de la classe cartographieTemporelle

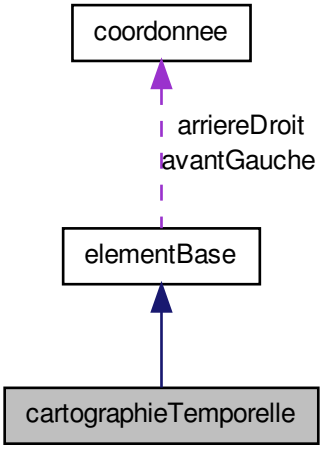
Cette classe représente la variable **cartographieTemporelle** (p. 49) d'une simulation.

```
#include <cartographieTemporelle.h>
```

Graphe d'héritage de cartographieTemporelle :



Graphe de collaboration de cartographieTemporelle :



Fonctions membres publiques

- **cartographieTemporelle** ()
- virtual ~**cartographieTemporelle** ()
- bool **IsChampE** () const
- void **SetChampE** (bool champE)
- bool **IsChampH** () const
- void **SetChampH** (bool champH)
- int **GetIntervalle** () const
- void **SetIntervalle** (int intervalle)
- int **GetIterationDebut** () const
- void **SetIterationDebut** (int iterationDebut)
- int **GetIterationFin** () const
- void **SetIterationFin** (int iterationFin)
- void **ecrire** (int nb)

4.15.1 Description détaillée

Cette classe représente la variable **cartographieTemporelle** (p. 49) d'une simulation.

Définition à la ligne 16 du fichier cartographieTemporelle.h.

4.15.2 Documentation des constructeurs et destructeur

4.15.2.1 cartographieTemporelle : :cartographieTemporelle ( )

Définition à la ligne 13 du fichier cartographieTemporelle.cpp.

4.15.2.2 cartographieTemporelle : :~cartographieTemporelle ( ) [virtual]

Définition à la ligne 21 du fichier cartographieTemporelle.cpp.

4.15.3 Documentation des fonctions membres

4.15.3.1 void cartographieTemporelle : :ecrire (int *nb*)

Définition à la ligne 24 du fichier cartographieTemporelle.cpp.

4.15.3.2 int cartographieTemporelle : :GetIntervalle ( ) const [inline]

Définition à la ligne 37 du fichier cartographieTemporelle.h.

4.15.3.3 int cartographieTemporelle : :GetIterationDebut ( ) const [inline]

Définition à la ligne 45 du fichier cartographieTemporelle.h.

4.15.3.4 int cartographieTemporelle : :GetIterationFin ( ) const [inline]

Définition à la ligne 53 du fichier cartographieTemporelle.h.

4.15.3.5 bool cartographieTemporelle : :IsChampE ( ) const [inline]

Définition à la ligne 21 du fichier cartographieTemporelle.h.

4.15.3.6 bool cartographieTemporelle : :IsChampH ( ) const [inline]

Définition à la ligne 29 du fichier cartographieTemporelle.h.

4.15.3.7 void cartographieTemporelle : :SetChampE (bool *champE*) [inline]

Définition à la ligne 25 du fichier cartographieTemporelle.h.

4.15.3.8 void cartographieTemporelle : :SetChampH (bool *champH*) [inline]

Définition à la ligne 33 du fichier cartographieTemporelle.h.

4.15.3.9 void cartographieTemporelle : :SetIntervalle (int *intervalle*) [inline]

Définition à la ligne 41 du fichier cartographieTemporelle.h.

4.15.3.10 void cartographieTemporelle : :SetIterationDebut (int *iterationDebut*) [inline]

Définition à la ligne 49 du fichier cartographieTemporelle.h.

4.15.3.11 void cartographieTemporelle : :SetIterationFin (int *iterationFin*) [inline]

Définition à la ligne 57 du fichier cartographieTemporelle.h.

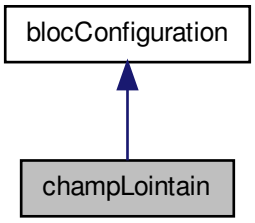
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**cartographieTemporelle.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**cartographieTemporelle.cpp**

4.16 Référence de la classe champLointain

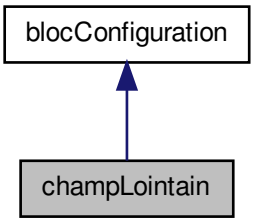
Cette classe représente la variable **champLointain** (p. 52) d'une simulation.

```
#include <champLointain.h>
```

Graphe d'héritage de champLointain :



Graphe de collaboration de champLointain :



Connecteurs publics

- void **SetCalculImage** (bool calculImage)
- void **SetFrequenceMax** (double frequenceMax)
- void **SetFrequenceMin** (double frequenceMin)
- void **SetHauteurPlanMasse** (int hauteurPlanMasse)
- void **SetOrigineRepere** (int x, int y, int z)
- void **SetOrigineRepere** (**coordonnee** c)
- void **SetPasDiscrPhi** (int pasDiscrPhi)
- void **SetPasDiscrTeta** (int pasDiscrTeta)
- void **SetPasFrequence** (double pasFrequence)
- void **SetNomFichier** (std : :string nomFichier)
- void **ecrire** ()

Fonctions membres publiques

- **champLointain** ()
- virtual ∼**champLointain** ()
- bool **IsCalculImage** () const
- double **GetFrequenceMax** () const
- double **GetFrequenceMin** () const
- int **GetHauteurPlanMasse** () const
- **coordonnee GetOrigineRepere** () const
- int **GetPasDiscrPhi** () const
- int **GetPasDiscrTeta** () const
- double **GetPasFrequence** () const
- std : :string **GetNomFichier** () const

4.16.1 Description détaillée

Cette classe représente la variable **champLointain** (p. 52) d'une simulation.

Définition à la ligne 17 du fichier champLointain.h.

4.16.2 Documentation des constructeurs et destructeur

4.16.2.1 champLointain : :champLointain ( )

Définition à la ligne 10 du fichier champLointain.cpp.

4.16.2.2 champLointain : :~champLointain ( ) [virtual]

Définition à la ligne 20 du fichier champLointain.cpp.

4.16.3 Documentation des fonctions membres

4.16.3.1 void champLointain : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 23 du fichier champLointain.cpp.

4.16.3.2 double champLointain : :GetFrequenceMax ( ) const [inline]

Définition à la ligne 26 du fichier champLointain.h.

4.16.3.3 double champLointain : :GetFrequenceMin ( ) const [inline]

Définition à la ligne 29 du fichier champLointain.h.

4.16.3.4 int champLointain : :GetHauteurPlanMasse ( ) const [inline]

Définition à la ligne 32 du fichier champLointain.h.

4.16.3.5 std : :string champLointain : :GetNomFichier ( ) const [inline]

Définition à la ligne 47 du fichier champLointain.h.

4.16.3.6 coordonnee champLointain : :GetOrigineRepere ( ) const [inline]

Définition à la ligne 35 du fichier champLointain.h.

4.16.3.7 int champLointain : :GetPasDiscrPhi ( ) const [inline]

Définition à la ligne 38 du fichier champLointain.h.

4.16.3.8 int champLointain : :GetPasDiscrTeta ( ) const [inline]

Définition à la ligne 41 du fichier champLointain.h.

4.16.3.9 double champLointain : :GetPasFrequence ( ) const [inline]

Définition à la ligne 44 du fichier champLointain.h.

4.16.3.10 bool champLointain : :IsCalculImage ( ) const [inline]

Définition à la ligne 23 du fichier champLointain.h.

4.16.3.11 void champLointain : :SetCalculImage (bool *calculImage*) [inline, slot]

Définition à la ligne 53 du fichier champLointain.h.

4.16.3.12 void champLointain : :SetFrequenceMax (double *frequenceMax*) [inline, slot]

Définition à la ligne 56 du fichier champLointain.h.

4.16.3.13 void champLointain : :SetFrequenceMin (double *frequenceMin*) [inline, slot]

Définition à la ligne 60 du fichier champLointain.h.

4.16.3.14 void champLointain : :SetHauteurPlanMasse (int *hauteurPlanMasse*) [inline, slot]

Définition à la ligne 64 du fichier champLointain.h.

4.16.3.15 void champLointain : :SetNomFichier (std : :string *nomFichier*) [inline, slot]

Définition à la ligne 91 du fichier champLointain.h.

4.16.3.16 void champLointain : :SetOrigineRepere (int *x*, int *y*, int *z*) [inline, slot]

Définition à la ligne 68 du fichier champLointain.h.

4.16.3.17 void champLointain : :SetOrigineRepere (coordonnee *c*) [inline, slot]

Définition à la ligne 71 du fichier champLointain.h.

4.16.3.18 void champLointain : :SetPasDiscrPhi (int *pasDiscrPhi*) [inline, slot]

Définition à la ligne 76 du fichier champLointain.h.

4.16.3.19 void champLointain : :SetPasDiscrTeta (int *pasDiscrTeta*) [inline, slot]

Définition à la ligne 81 du fichier champLointain.h.

4.16.3.20 void champLointain : :SetPasFrequence (double *pasFrequence*) [inline, slot]

Définition à la ligne 86 du fichier champLointain.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**champLointain.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**champLointain.cpp**

4.17 Référence de la classe ColorPicker

Cette classe est un widget sous forme de bouton qui permet de sélectionner une couleur dans une boite de dialogue qui s'ouvre quand le bouton est pressé. Un carré de la couleur sélectionnée est présent sur le bouton.

```
#include <ColorPicker.h>
```

Connecteurs publics

- void **ouvreColorDialog** ()
  Cette fonction ouvre la boite de dialogue pour changer de couleur et met à jour la couleur courante.

Signaux

- void **colorChanged** (QColor)
  Ce signal est émis quand la couleur change.

Fonctions membres publiques

- **ColorPicker** (QColor c, QWidget ∗parent=0)
- void **paintEvent** (QPaintEvent ∗)
  Dessine le carré avec la couleur sélectionnée sur le bouton.
- void **setColor** (QColor c)
  Change la couleur courante.

4.17.1 Description détaillée

Cette classe est un widget sous forme de bouton qui permet de sélectionner une couleur dans une boite de dialogue qui s'ouvre quand le bouton est pressé. Un carré de la couleur sélectionnée est présent sur le bouton.

Définition à la ligne 14 du fichier ColorPicker.h.

4.17.2 Documentation des constructeurs et destructeur

4.17.2.1 ColorPicker : :ColorPicker (QColor *c*, QWidget * *parent* = 0) [explicit]

Définition à la ligne 3 du fichier ColorPicker.cpp.

4.17.3 Documentation des fonctions membres

4.17.3.1 void ColorPicker : :colorChanged (QColor) [signal]

Ce signal est émis quand la couleur change.

4.17.3.2 void ColorPicker : :ouvreColorDialog ( ) [slot]

Cette fonction ouvre la boite de dialogue pour changer de couleur et met à jour la couleur courante.

Définition à la ligne 25 du fichier ColorPicker.cpp.

4.17.3.3 void ColorPicker : :paintEvent (QPaintEvent * *e*)

Dessine le carré avec la couleur sélectionnée sur le bouton.

Définition à la ligne 11 du fichier ColorPicker.cpp.

4.17.3.4 void ColorPicker : :setColor (QColor *c*) [inline]

Change la couleur courante.

Paramètres

c	la nouvelle couleur

Définition à la ligne 29 du fichier ColorPicker.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ColorPicker.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ColorPicker.cpp**

4.18 Référence de la classe ComboxDecaleeDelegate

Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur sous forme de liste déroulante.

```
#include <ComboxDecaleeDelegate.h>
```

Fonctions membres publiques

- **ComboxDecaleeDelegate** (QStringList list, int decalage=0, QObject ∗parent=0)
- void **paint** (QPainter ∗painter, const QStyleOptionViewItem &option, const QModelIndex &index) const
- QWidget ∗ **createEditor** (QWidget ∗parent, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **setEditorData** (QWidget ∗editor, const QModelIndex &index) const
- void **setModelData** (QWidget ∗editor, QAbstractItemModel ∗model, const QModelIndex &index) const
- void **updateEditorGeometry** (QWidget ∗editor, const QStyleOptionViewItem &option, const QModelIndex &index) const

4.18.1 Description détaillée

Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur sous forme de liste déroulante.

Définition à la ligne 11 du fichier ComboxDecaleeDelegate.h.

4.18.2 Documentation des constructeurs et destructeur

4.18.2.1 ComboxDecaleeDelegate : :ComboxDecaleeDelegate (QStringList *list,* int *decalage =* 0*,* QObject ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier ComboxDecaleeDelegate.cpp.

4.18.3 Documentation des fonctions membres

4.18.3.1 QWidget ∗ ComboxDecaleeDelegate : :createEditor (QWidget ∗ *parent,* const QStyleOptionViewItem & *option,* const QModelIndex & *index*) const

Définition à la ligne 28 du fichier ComboxDecaleeDelegate.cpp.

4.18.3.2 void ComboxDecaleeDelegate : :paint (QPainter ∗ *painter,* const QStyleOptionViewItem & *option,* const QModelIndex & *index*) const

Définition à la ligne 10 du fichier ComboxDecaleeDelegate.cpp.

4.18.3.3 void ComboxDecaleeDelegate : :setEditorData (QWidget ∗ *editor,* const QModelIndex & *index*) const

Définition à la ligne 39 du fichier ComboxDecaleeDelegate.cpp.

4.18.3.4 void ComboxDecaleeDelegate : :setModelData (QWidget ∗ *editor,* QAbstractItemModel ∗ *model,* const QModelIndex & *index*) const

Définition à la ligne 53 du fichier ComboxDecaleeDelegate.cpp.

4.18.3.5 void ComboxDecaleeDelegate : :updateEditorGeometry (QWidget ∗ *editor,* const QStyleOptionViewItem & *option,* const QModelIndex & *index*) const

Définition à la ligne 64 du fichier ComboxDecaleeDelegate.cpp.

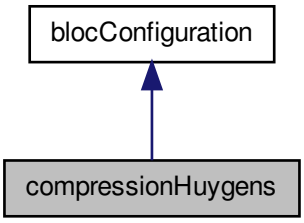
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**ComboxDecaleeDelegate.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**ComboxDecaleeDelegate.cpp**

4.19 Référence de la classe compressionHuygens

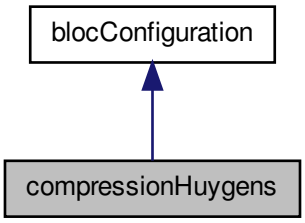
Cette classe représente la variable **compressionHuygens** (p. 59) d'une simulation.

```
#include <compressionHuygens.h>
```

Graphe d'héritage de compressionHuygens :



Graphe de collaboration de compressionHuygens :



Connecteurs publics

- void **SetCompressionHauteur** (int compressionHauteur)
- void **SetCompressionLargeur** (int compressionLargeur)
- void **SetCompressionLongueur** (int compressionLongueur)
- void **SetFacteurMultiplicatif** (int facteurMultiplicatif)
- void **ecrire** ()

Fonctions membres publiques

- **compressionHuygens** ()
- virtual ∼**compressionHuygens** ()
- int **GetCompressionHauteur** () const
- int **GetCompressionLargeur** () const
- int **GetCompressionLongueur** () const
- int **GetFacteurMultiplicatif** () const

4.19.1 Description détaillée

Cette classe représente la variable **compressionHuygens** (p. 59) d'une simulation.

Définition à la ligne 16 du fichier compressionHuygens.h.

4.19.2 Documentation des constructeurs et destructeur

4.19.2.1 compressionHuygens : :compressionHuygens ( )

Définition à la ligne 10 du fichier compressionHuygens.cpp.

4.19.2.2 compressionHuygens : :∼compressionHuygens ( ) [virtual]

Définition à la ligne 17 du fichier compressionHuygens.cpp.

4.19.3 Documentation des fonctions membres

4.19.3.1 void compressionHuygens : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 20 du fichier compressionHuygens.cpp.

4.19.3.2 int compressionHuygens : :GetCompressionHauteur ( ) const [inline]

Définition à la ligne 22 du fichier compressionHuygens.h.

4.19.3.3 int compressionHuygens : :GetCompressionLargeur ( ) const [inline]

Définition à la ligne 25 du fichier compressionHuygens.h.

4.19.3.4 int compressionHuygens : :GetCompressionLongueur ( ) const [inline]

Définition à la ligne 28 du fichier compressionHuygens.h.

4.19.3.5 int compressionHuygens : :GetFacteurMultiplicatif ( ) const [inline]

Définition à la ligne 31 du fichier compressionHuygens.h.

4.19.3.6 void compressionHuygens : :SetCompressionHauteur (int *compressionHauteur*) [inline, slot]

Définition à la ligne 37 du fichier compressionHuygens.h.

4.19.3.7 void compressionHuygens : :SetCompressionLargeur (int *compressionLargeur*) [inline, slot]

Définition à la ligne 41 du fichier compressionHuygens.h.

4.19.3.8 void compressionHuygens : :SetCompressionLongueur (int *compressionLongueur*) [inline, slot]

Définition à la ligne 45 du fichier compressionHuygens.h.

4.19.3.9 void compressionHuygens : :SetFacteurMultiplicatif (int *facteurMultiplicatif*) [inline, slot]

Définition à la ligne 49 du fichier compressionHuygens.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**compressionHuygens.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**compressionHuygens.cpp**

4.20 Référence de la classe coordonnee

Cette classe représente une coordonnee en 3 dimensions. Elle sert d'unité de position dans la scène 3D.

```
#include <coordonnee.h>
```

Fonctions membres publiques

- **coordonnee** ()
- **coordonnee** (int x, int y, int z)
- **coordonnee** (const **coordonnee** &orig)
- virtual ~**coordonnee** ()
- int **GetX** () const
- void **SetX** (int x)
- int **GetY** () const
- void **SetY** (int y)
- int **GetZ** () const
- void **SetZ** (int z)
- QString **toQString** ()
- bool **operator==** (**coordonnee** const &b)
- void **ecrire** ()

4.20.1 Description détaillée

Cette classe représente une coordonnee en 3 dimensions. Elle sert d'unité de position dans la scène 3D.

Définition à la ligne 17 du fichier coordonnee.h.

4.20.2 Documentation des constructeurs et destructeur

4.20.2.1 coordonnee : :coordonnee ( )

Définition à la ligne 12 du fichier coordonnee.cpp.

4.20.2.2 coordonnee : :coordonnee (int *x*, int *y*, int *z*)

Définition à la ligne 18 du fichier coordonnee.cpp.

4.20.2.3 coordonnee : :coordonnee (const coordonnee & *orig*)

Définition à la ligne 24 du fichier coordonnee.cpp.

4.20.2.4 coordonnee : :~coordonnee ( ) [virtual]

Définition à la ligne 30 du fichier coordonnee.cpp.

4.20.3 Documentation des fonctions membres

4.20.3.1 void coordonnee : :ecrire ( )

Définition à la ligne 38 du fichier coordonnee.cpp.

4.20.3.2 int coordonnee : :GetX ( ) const [inline]

Définition à la ligne 24 du fichier coordonnee.h.

4.20.3.3 int coordonnee : :GetY ( ) const [inline]

Définition à la ligne 32 du fichier coordonnee.h.

4.20.3.4 int coordonnee : :GetZ ( ) const [inline]

Définition à la ligne 40 du fichier coordonnee.h.

4.20.3.5 bool coordonnee : :operator== (coordonnee const & *b*)

Définition à la ligne 33 du fichier coordonnee.cpp.

4.20.3.6 void coordonnee : :SetX (int *x*) [inline]

Définition à la ligne 28 du fichier coordonnee.h.

4.20.3.7 void coordonnee : :SetY (int *y*) [inline]

Définition à la ligne 36 du fichier coordonnee.h.

4.20.3.8 void coordonnee : :SetZ (int *z*) [inline]

Définition à la ligne 44 du fichier coordonnee.h.

4.20.3.9 QString coordonnee : :toQString ( ) [inline]

Définition à la ligne 48 du fichier coordonnee.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**coordonnee.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**coordonnee.cpp**

4.21 Référence de la classe CoordonneeDelegate

Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur pour modifier une coordonnees. Cette éditeur est en faite un **FormCoordonnees** (p. 93).

```
#include <CoordonneeDelegate.h>
```

Fonctions membres publiques

- **CoordonneeDelegate** ()
- void **paint** (QPainter ∗painter, const QStyleOptionViewItem &option, const QModelIndex &index) const
- QWidget ∗ **createEditor** (QWidget ∗parent, const QStyleOptionViewItem &option, const QModelIndex &index) const
- void **setEditorData** (QWidget ∗editor, const QModelIndex &index) const
- void **setModelData** (QWidget ∗editor, QAbstractItemModel ∗model, const QModelIndex &index) const
- void **updateEditorGeometry** (QWidget ∗editor, const QStyleOptionViewItem &option, const QModelIndex &index) const

4.21.1 Description détaillée

Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur pour modifier une coordonnees. Cette éditeur est en faite un **FormCoordonnees** (p. 93).

Définition à la ligne 11 du fichier CoordonneeDelegate.h.

4.21.2 Documentation des constructeurs et destructeur

4.21.2.1 CoordonneeDelegate : :CoordonneeDelegate ( )

Définition à la ligne 5 du fichier CoordonneeDelegate.cpp.

4.21.3 Documentation des fonctions membres

4.21.3.1 QWidget * CoordonneeDelegate : :createEditor (QWidget * *parent,* const QStyleOptionViewItem & *option,* const QModelIndex & *index*) const

Définition à la ligne 19 du fichier CoordonneeDelegate.cpp.

4.21.3.2 void CoordonneeDelegate : :paint (QPainter * *painter,* const QStyleOptionViewItem & *option,* const QModelIndex & *index*) const

Définition à la ligne 9 du fichier CoordonneeDelegate.cpp.

4.21.3.3 void CoordonneeDelegate : :setEditorData (QWidget * *editor,* const QModelIndex & *index*) const

Définition à la ligne 29 du fichier CoordonneeDelegate.cpp.

4.21.3.4 void CoordonneeDelegate : :setModelData (QWidget * *editor,* QAbstractItemModel * *model,* const QModelIndex & *index*) const

Définition à la ligne 39 du fichier CoordonneeDelegate.cpp.

4.21.3.5 void CoordonneeDelegate : :updateEditorGeometry (QWidget * *editor,* const QStyleOptionViewItem & *option,* const QModelIndex & *index*) const

Définition à la ligne 50 du fichier CoordonneeDelegate.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**CoordonneeDelegate.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**CoordonneeDelegate.cpp**

4.22 Référence de la classe Cube

```
#include <cube.h>
```

Fonctions membres publiques

- **Cube** ()
- **Cube** (**Point**, **Point**)
- void **dessineCube** ()

Fonctions membres publiques statiques

- static void **drawCube** (**Point** topLeftCorner, **Point** bottomRightCorner)

4.22.1 Description détaillée

Définition à la ligne 7 du fichier cube.h.

4.22.2 Documentation des constructeurs et destructeur

4.22.2.1 Cube : :Cube ( )

Définition à la ligne 3 du fichier cube.cpp.

4.22.2.2 Cube : :Cube (Point *p1,* Point *p2*)

Définition à la ligne 15 du fichier cube.cpp.

4.22.3 Documentation des fonctions membres

4.22.3.1 void Cube : :dessineCube ( )

Définition à la ligne 22 du fichier cube.cpp.

4.22.3.2 void Cube : :drawCube (Point *topLeftCorner,* Point *bottomRightCorner*) [static]

Définition à la ligne 74 du fichier cube.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**cube.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**cube.cpp**

4.23 Référence de la classe Data

Contient toutes les données utiles pour la simulation.

```
#include <Data.h>
```

Fonctions membres publiques

- **Data** ()
- **Data** (**Parser** ∗p)
- **volumeCalcul** ∗ **getVolumeCalcul** ()
- **paroi** ∗ **getParoi** ()
- **blocMetallisation** ∗ **getBlocMetallisations** ()
- **blocParallelepipede** ∗ **getBlocParallelepipede** ()
- **blocSonde** ∗ **getBlocSonde** ()
- **blocElementLocalise** ∗ **getBlocElementLocalise** ()
- **blocPortExcitation** ∗ **getBlocPortExcitation** ()
- **blocCartographieTemporelle** ∗ **getBlocCartoTempo** ()

- **blocSurfacePrelevement** ∗ **getBlocSurfacePrelev** ()
- **pml** ∗ **getPML** ()
- **ParamExcitations** ∗ **getParamExcitations** ()
- **analyseTemporelle** ∗ **getAnalyseTempo** ()
- **blocPortExcitation** ∗ **getPortExcitations** ()
- **cageExcitation** ∗ **getCageExcitation** ()
- **champLointain** ∗ **getChampLointain** ()
- **Excitation** ∗ **getExcitation** ()
- **formatStockage** ∗ **getFormatStockage** ()
- **blocEchantillonnage** ∗ **getBlocEchantillonnage** ()
- **compressionHuygens** ∗ **getCompressionHuygens** ()
- **calculDirectivite** ∗ **getCalculDirective** ()
- void **ecrire** (int numstru)

4.23.1 Description détaillée

Contient toutes les données utiles pour la simulation.

Définition à la ligne 29 du fichier Data.h.

4.23.2 Documentation des constructeurs et destructeur

4.23.2.1 Data : :Data ( )

Définition à la ligne 5 du fichier Data.cpp.

4.23.2.2 Data : :Data (Parser ∗ *p*)

Définition à la ligne 9 du fichier Data.cpp.

4.23.3 Documentation des fonctions membres

4.23.3.1 void Data : :ecrire (int *numstru*)

Définition à la ligne 14 du fichier Data.cpp.

4.23.3.2 analyseTemporelle∗ Data : :getAnalyseTempo ( ) [inline]

Définition à la ligne 49 du fichier Data.h.

4.23.3.3 blocCartographieTemporelle∗ Data : :getBlocCartoTempo ( ) [inline]

Définition à la ligne 44 du fichier Data.h.

4.23.3.4 blocEchantillonnage* Data : :getBlocEchantillonnage ( ) [inline]

Définition à la ligne 55 du fichier Data.h.

4.23.3.5 blocElementLocalise* Data : :getBlocElementLocalise ( ) [inline]

Définition à la ligne 42 du fichier Data.h.

4.23.3.6 blocMetallisation* Data : :getBlocMetallisations ( ) [inline]

Définition à la ligne 39 du fichier Data.h.

4.23.3.7 blocParallelepipede* Data : :getBlocParallelepipede ( ) [inline]

Définition à la ligne 40 du fichier Data.h.

4.23.3.8 blocPortExcitation* Data : :getBlocPortExcitation ( ) [inline]

Définition à la ligne 43 du fichier Data.h.

4.23.3.9 blocSonde* Data : :getBlocSonde ( ) [inline]

Définition à la ligne 41 du fichier Data.h.

4.23.3.10 blocSurfacePrelevement* Data : :getBlocSurfacePrelev ( ) [inline]

Définition à la ligne 45 du fichier Data.h.

4.23.3.11 cageExcitation* Data : :getCageExcitation ( ) [inline]

Définition à la ligne 51 du fichier Data.h.

4.23.3.12 calculDirectivite* Data : :getCalculDirective ( ) [inline]

Définition à la ligne 57 du fichier Data.h.

4.23.3.13 champLointain* Data : :getChampLointain ( ) [inline]

Définition à la ligne 52 du fichier Data.h.

4.23.3.14 compressionHuygens* Data : :getCompressionHuygens ( ) [inline]

Définition à la ligne 56 du fichier Data.h.

4.23.3.15 Excitation* Data : :getExcitation ( ) [inline]

Définition à la ligne 53 du fichier Data.h.

4.23.3.16 formatStockage* Data : :getFormatStockage ( ) [inline]

Définition à la ligne 54 du fichier Data.h.

4.23.3.17 ParamExcitations* Data : :getParamExcitations ( ) [inline]

Définition à la ligne 48 du fichier Data.h.

4.23.3.18 paroi* Data : :getParoi ( ) [inline]

Définition à la ligne 37 du fichier Data.h.

4.23.3.19 pml* Data : :getPML ( ) [inline]

Définition à la ligne 47 du fichier Data.h.

4.23.3.20 blocPortExcitation* Data : :getPortExcitations ( ) [inline]

Définition à la ligne 50 du fichier Data.h.

4.23.3.21 volumeCalcul* Data : :getVolumeCalcul ( ) [inline]

Définition à la ligne 36 du fichier Data.h.

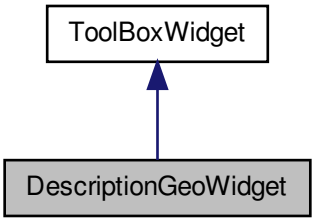
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**Data.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**Data.cpp**

4.24 Référence de la classe DescriptionGeoWidget

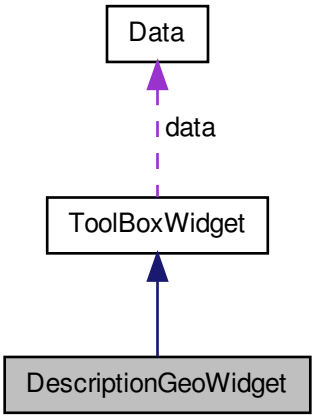
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

```
#include <DescriptionGeoWidget.h>
```

Graphe d'héritage de DescriptionGeoWidget :



Graphe de collaboration de DescriptionGeoWidget :



Connecteurs publics

– void **openElementLocaliseWindows** ()
  *Cette fonction ouvre un formulaire pour ajouter un objet **elementLocalise** (p. 77).*

Signaux

– void **newMetalCreated** (**elementBase** ∗)

Ce signal est émis quand un nouvelle objet metallisation est créé

– void **newParaCreated** (**elementBase** ∗)

Ce signal est émis quand un nouvelle objet parallelepipede est créé

– void **newElemLocCreated** (**elementBase** ∗)

*Ce signal est émis quand un nouvelle objet **elementLocalise** (p. 77) est créé*

Fonctions membres publiques

– **DescriptionGeoWidget** (**Data** ∗**data**, QWidget ∗parent=0)

– void **init** ()

*Initialise les formulaire avec les valeurs de l'objet **Data** (p. 66).*

4.24.1 Description détaillée

Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

Définition à la ligne 16 du fichier DescriptionGeoWidget.h.

4.24.2 Documentation des constructeurs et destructeur

4.24.2.1 DescriptionGeoWidget : :DescriptionGeoWidget (Data ∗ *data,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier DescriptionGeoWidget.cpp.

4.24.3 Documentation des fonctions membres

4.24.3.1 void DescriptionGeoWidget : :init ( ) [virtual]

Initialise les formulaire avec les valeurs de l'objet **Data** (p. 66).

Implémente **ToolBoxWidget** (p. 195).

Définition à la ligne 55 du fichier DescriptionGeoWidget.cpp.

4.24.3.2 void DescriptionGeoWidget : :newElemLocCreated (elementBase ∗) [signal]

Ce signal est émis quand un nouvelle objet **elementLocalise** (p. 77) est créé

4.24.3.3 void DescriptionGeoWidget : :newMetalCreated (elementBase ∗) [signal]

Ce signal est émis quand un nouvelle objet metallisation est créé

4.24.3.4 void DescriptionGeoWidget : :newParaCreated (elementBase *) [signal]

Ce signal est émis quand un nouvelle objet parallelepipede est créé

4.24.3.5 void DescriptionGeoWidget : :openElementLocaliseWindows ( ) [slot]

Cette fonction ouvre un formulaire pour ajouter un objet **elementLocalise** (p. 77).

Définition à la ligne 45 du fichier DescriptionGeoWidget.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**DescriptionGeoWidget.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**DescriptionGeoWidget.cpp**

4.25 Référence de la classe Ecriture

```
#include <Ecriture.h>
```

Fonctions membres publiques

- std : :string **GetNomFichier** () const

Fonctions membres publiques statiques

- static void **SetNomFichier** (std : :string extension)
- static void **Ecrire** (std : :string s)
- static void **setNumStru** (int num)

4.25.1 Description détaillée

Définition à la ligne 17 du fichier Ecriture.h.

4.25.2 Documentation des fonctions membres

4.25.2.1 static void Ecriture : :Ecrire (std : :string *s*) [inline, static]

Définition à la ligne 46 du fichier Ecriture.h.

4.25.2.2 std : :string Ecriture : :GetNomFichier ( ) const [inline]

Définition à la ligne 20 du fichier Ecriture.h.

4.25.2.3 static void Ecriture : :SetNomFichier (std : :string *extension*) [inline, static]

Définition à la ligne 24 du fichier Ecriture.h.

4.25.2.4 static void Ecriture : :setNumStru (int *num*) [inline, static]

Définition à la ligne 51 du fichier Ecriture.h.

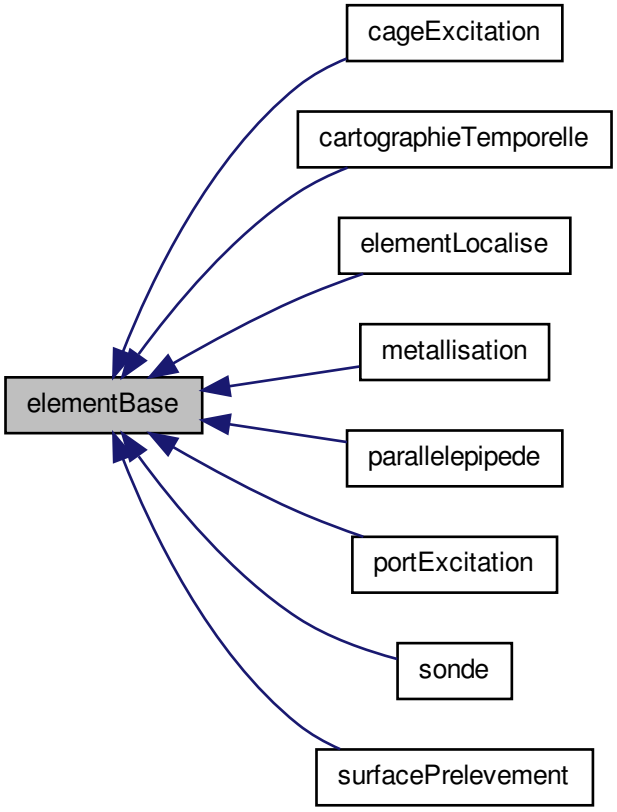
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**Ecriture.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**Ecriture.cpp**

4.26 Référence de la classe elementBase

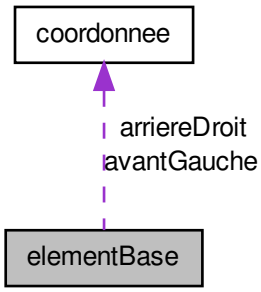
The **elementBase** (p. 73) class.

```
#include <elementBase.h>
```

Graphe d'héritage de elementBase :



Graphe de collaboration de elementBase :



Fonctions membres publiques

- **elementBase** ()
- virtual ~**elementBase** ()
- **coordonnee GetArriereDroit** () const
- **coordonnee GetAvantGauche** () const
- QColor **getCouleur** () const
- void **ecrireElementBase** ()
- bool **operator==** (**elementBase** const &b)
- void **invertColor** ()
- QColor **getInvertColor** ()
- void **setCouleur** (QColor c)
- void **setAlpha** (int a)
- void **SetAvantGauche** (int x, int y, int z)
- void **SetAvantGauche** (**coordonnee** copie)
- void **SetArriereDroit** (int x, int y, int z)
- void **SetArriereDroit** (**coordonnee** copie)

Attributs protégés

- QColor **couleur**
- **coordonnee avantGauche**
- **coordonnee arriereDroit**

4.26.1 Description détaillée

The **elementBase** (p. 73) class.

Cette classe représente un élement de base affichable dans la scène 3D et contenu dans un tableau

Définition à la ligne 19 du fichier elementBase.h.

4.26.2 Documentation des constructeurs et destructeur

4.26.2.1 elementBase : :elementBase ( )

Définition à la ligne 11 du fichier elementBase.cpp.

4.26.2.2 elementBase : :~elementBase ( ) [virtual]

Définition à la ligne 18 du fichier elementBase.cpp.

4.26.3 Documentation des fonctions membres

4.26.3.1 void elementBase : :ecrireElementBase ( )

Définition à la ligne 21 du fichier elementBase.cpp.

4.26.3.2 coordonnee elementBase : :GetArriereDroit ( ) const [inline]

Définition à la ligne 24 du fichier elementBase.h.

4.26.3.3 coordonnee elementBase : :GetAvantGauche ( ) const [inline]

Définition à la ligne 28 du fichier elementBase.h.

4.26.3.4 QColor elementBase : :getCouleur ( ) const [inline]

Définition à la ligne 32 du fichier elementBase.h.

4.26.3.5 QColor elementBase : :getInvertColor ( ) [inline]

Définition à la ligne 49 du fichier elementBase.h.

4.26.3.6 void elementBase : :invertColor ( ) [inline]

Définition à la ligne 42 du fichier elementBase.h.

4.26.3.7 bool elementBase : :operator== (elementBase const & *b*)

Définition à la ligne 29 du fichier elementBase.cpp.

4.26.3.8 void elementBase : :setAlpha (int *a*) [inline]

Définition à la ligne 63 du fichier elementBase.h.

4.26.3.9 void elementBase : :SetArriereDroit (int *x*, int *y*, int *z*) [inline]

Définition à la ligne 74 du fichier elementBase.h.

4.26.3.10 void elementBase : :SetArriereDroit (coordonnee *copie*) [inline]

Définition à la ligne 78 du fichier elementBase.h.

4.26.3.11 void elementBase : :SetAvantGauche (int *x*, int *y*, int *z*) [inline]

Définition à la ligne 67 du fichier elementBase.h.

4.26.3.12 void elementBase : :SetAvantGauche (coordonnee *copie*) [inline]

Définition à la ligne 71 du fichier elementBase.h.

4.26.3.13 void elementBase : :setCouleur (QColor *c*) [inline]

Définition à la ligne 59 du fichier elementBase.h.

4.26.4 Documentation des données membres

4.26.4.1 coordonnee elementBase : :arriereDroit [protected]

Définition à la ligne 89 du fichier elementBase.h.

4.26.4.2 coordonnee elementBase : :avantGauche [protected]

Définition à la ligne 88 du fichier elementBase.h.

4.26.4.3 QColor elementBase : :couleur [protected]

Définition à la ligne 86 du fichier elementBase.h.

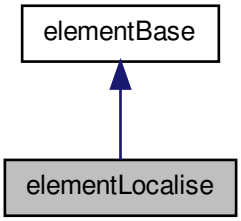
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**elementBase.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**elementBase.cpp**

4.27 Référence de la classe elementLocalise

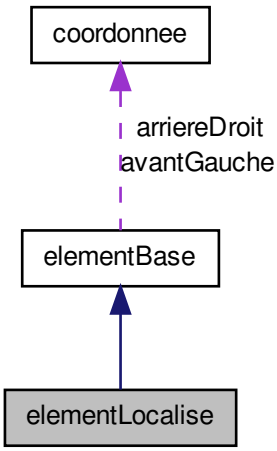
Cette classe représente la variable **elementLocalise** (p. 77) d'une simulation.

```
#include <elementLocalise.h>
```

Graphe d'héritage de elementLocalise :



Graphe de collaboration de elementLocalise :



Fonctions membres publiques

- **elementLocalise** ()
- virtual ~**elementLocalise** ()
- int **GetDirection** () const
- void **SetDirection** (int direction)
- int **GetType** () const
- void **SetType** (int type)

– float **GetValeur** () const
– void **SetValeur** (float valeur)
– void **ecrire** (int nb)

4.27.1 Description détaillée

Cette classe représente la variable **elementLocalise** (p. 77) d'une simulation.

Définition à la ligne 16 du fichier elementLocalise.h.

4.27.2 Documentation des constructeurs et destructeur

4.27.2.1 elementLocalise : :elementLocalise ( )

Définition à la ligne 13 du fichier elementLocalise.cpp.

4.27.2.2 elementLocalise : :~elementLocalise ( ) [virtual]

Définition à la ligne 20 du fichier elementLocalise.cpp.

4.27.3 Documentation des fonctions membres

4.27.3.1 void elementLocalise : :ecrire (int *nb*)

Définition à la ligne 23 du fichier elementLocalise.cpp.

4.27.3.2 int elementLocalise : :GetDirection ( ) const [inline]

Définition à la ligne 21 du fichier elementLocalise.h.

4.27.3.3 int elementLocalise : :GetType ( ) const [inline]

Définition à la ligne 29 du fichier elementLocalise.h.

4.27.3.4 float elementLocalise : :GetValeur ( ) const [inline]

Définition à la ligne 37 du fichier elementLocalise.h.

4.27.3.5 void elementLocalise : :SetDirection (int *direction*) [inline]

Définition à la ligne 25 du fichier elementLocalise.h.

4.27.3.6 void elementLocalise : :SetType (int *type*) [inline]

Définition à la ligne 33 du fichier elementLocalise.h.

4.27.3.7 void elementLocalise : :SetValeur (float *valeur*) [inline]

Définition à la ligne 41 du fichier elementLocalise.h.

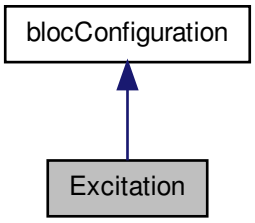
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**elementLocalise.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**elementLocalise.cpp**

4.28 Référence de la classe Excitation

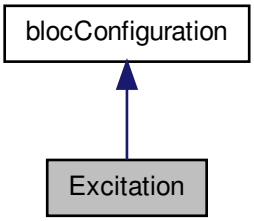
Cette classe représente la variable **Excitation** (p. 80) d'une simulation.

`#include <Excitation.h>`

Graphe d'héritage de Excitation :



Graphe de collaboration de Excitation :



Connecteurs publics

- void **SetAmplitudeV0** (double amplitudeV0)
- void **ecrire** ()

Fonctions membres publiques

- **Excitation** ()
- virtual ~**Excitation** ()
- double **GetAmplitudeV0** () const

4.28.1 Description détaillée

Cette classe représente la variable **Excitation** (p. 80) d'une simulation.

Définition à la ligne 16 du fichier Excitation.h.

4.28.2 Documentation des constructeurs et destructeur

4.28.2.1 Excitation : :Excitation ( )

Définition à la ligne 10 du fichier Excitation.cpp.

4.28.2.2 Excitation : :~Excitation ( ) [virtual]

Définition à la ligne 14 du fichier Excitation.cpp.

4.28.3 Documentation des fonctions membres

4.28.3.1 void Excitation : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 17 du fichier Excitation.cpp.

4.28.3.2 double Excitation : :GetAmplitudeV0 ( ) const [inline]

Définition à la ligne 22 du fichier Excitation.h.

4.28.3.3 void Excitation : :SetAmplitudeV0 (double *amplitudeV0*) [inline, slot]

Définition à la ligne 28 du fichier Excitation.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**Excitation.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**Excitation.cpp**

4.29 Référence de la classe FilterWidget

The **FilterWidget** (p. 82) class.

```
#include <FilterWidget.h>
```

Connecteurs publics

- void **setListCombox** (QStringList)
- void **setCurrentModel** (**TableModel** ∗)
- void **clear** ()

Signaux

- void **columnChanged** (int)
- void **filterChanged** (QString)

Fonctions membres publiques

- **FilterWidget** (QWidget ∗parent=0)

4.29.1 Description détaillée

The **FilterWidget** (p. 82) class.

Ce widget est un formulaire qui permet de filtrer le tableau courant en fonction de sa colonne et d'une expression régulière

Définition à la ligne 13 du fichier FilterWidget.h.

4.29.2 Documentation des constructeurs et destructeur

4.29.2.1 FilterWidget : :FilterWidget (QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FilterWidget.cpp.

4.29.3 Documentation des fonctions membres

4.29.3.1 void FilterWidget : :clear ( ) [slot]

Définition à la ligne 37 du fichier FilterWidget.cpp.

4.29.3.2 void FilterWidget : :columnChanged (int) [signal]

4.29.3.3 void FilterWidget : :filterChanged (QString) [signal]

4.29.3.4 void FilterWidget : :setCurrentModel (TableModel ∗ *t*) [slot]

Définition à la ligne 31 du fichier FilterWidget.cpp.

4.29.3.5 void FilterWidget : :setListCombox (QStringList /) [slot]

Définition à la ligne 23 du fichier FilterWidget.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FilterWidget.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FilterWidget.cpp**

4.30 Référence de la classe FormAnalyseTempo

The **FormAnalyseTempo** (p. 83) class.

```
#include <FormAnalyseTempo.h>
```

Fonctions membres publiques

– **FormAnalyseTempo** (**analyseTemporelle** ∗anaTempo, QWidget ∗parent=0)
– void **setAnalyseTempo** (**analyseTemporelle** ∗temp)
– void **init** ()

4.30.1 Description détaillée

The **FormAnalyseTempo** (p. 83) class.

Formulaire de modification de l'objet **analyseTemporelle** (p. 15)

Définition à la ligne 11 du fichier FormAnalyseTempo.h.

4.30.2 Documentation des constructeurs et destructeur

4.30.2.1 FormAnalyseTempo : :FormAnalyseTempo (analyseTemporelle * *anaTempo,* QWidget * *parent* = 0)

Définition à la ligne 3 du fichier FormAnalyseTempo.cpp.

4.30.3 Documentation des fonctions membres

4.30.3.1 void FormAnalyseTempo : :init ( )

Définition à la ligne 31 du fichier FormAnalyseTempo.cpp.

4.30.3.2 void FormAnalyseTempo : :setAnalyseTempo (analyseTemporelle * *temp*) [inline]

Définition à la ligne 16 du fichier FormAnalyseTempo.h.

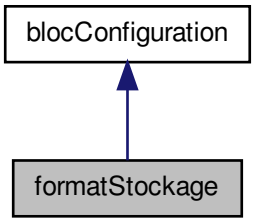
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormAnalyseTempo.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormAnalyseTempo.cpp**

4.31 Référence de la classe formatStockage

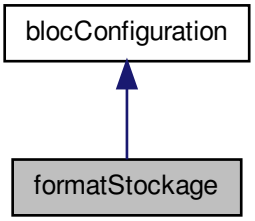
Cette classe représente la variable **formatStockage** (p. 84) d'une simulation.

```
#include <formatStockage.h>
```

Graphe d'héritage de formatStockage :



Graphe de collaboration de formatStockage :



Connecteurs publics

- void **SetFormat** (int format)
- void **SetFormatFichierPrelevement** (int formatFichierPrelevement)
- void **ecrire** ()

Fonctions membres publiques

- **formatStockage** ()
- virtual ~**formatStockage** ()
- int **GetFormat** () const
- int **GetFormatFichierPrelevement** () const

4.31.1 Description détaillée

Cette classe représente la variable **formatStockage** (p. 84) d'une simulation.

Définition à la ligne 16 du fichier formatStockage.h.

4.31.2 Documentation des constructeurs et destructeur

4.31.2.1 formatStockage : :formatStockage ( )

Définition à la ligne 10 du fichier formatStockage.cpp.

4.31.2.2 formatStockage : :~formatStockage ( ) [virtual]

Définition à la ligne 15 du fichier formatStockage.cpp.

4.31.3 Documentation des fonctions membres

4.31.3.1 void formatStockage : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 18 du fichier formatStockage.cpp.

4.31.3.2 int formatStockage : :GetFormat ( ) const [inline]

Définition à la ligne 22 du fichier formatStockage.h.

4.31.3.3 int formatStockage : :GetFormatFichierPrelevement ( ) const [inline]

Définition à la ligne 26 du fichier formatStockage.h.

4.31.3.4 void formatStockage : :SetFormat (int *format*) [inline, slot]

Définition à la ligne 32 du fichier formatStockage.h.

4.31.3.5 void formatStockage : :SetFormatFichierPrelevement (int *formatFichierPrelevement*) [inline, slot]

Définition à la ligne 36 du fichier formatStockage.h.

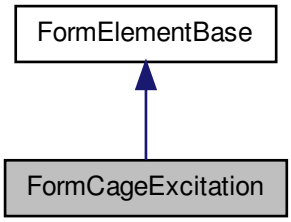
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**formatStockage.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**formatStockage.cpp**

4.32 Référence de la classe FormCageExcitation

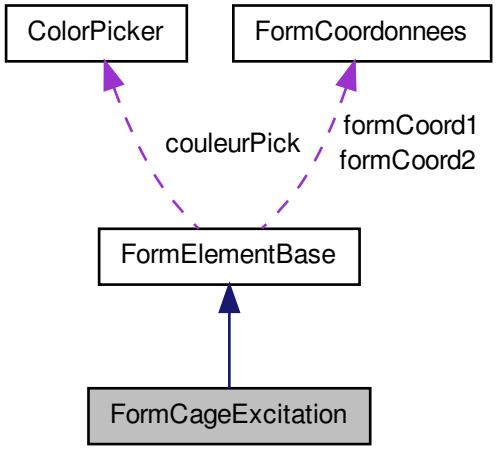
Formulaire permettant d'ajouter un objet **cageExcitation** (p. 40).

```
#include <FormCageExcitation.h>
```

Graphe d'héritage de FormCageExcitation :



Graphe de collaboration de FormCageExcitation :



Connecteurs publics

- void **valider** ()
  *Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).*
- void **manageFormulaire** (int val)
- void **actualiseFormulaire** (int)
- void **changeCreate** (bool)

Fonctions membres publiques

- **FormCageExcitation** (**cageExcitation** ∗cageExci, QWidget ∗parent=0)
- void **setCageExcitation** (**cageExcitation** ∗c)

4.32.1 Description détaillée

Formulaire permettant d'ajouter un objet **cageExcitation** (p. 40).

Définition à la ligne 10 du fichier FormCageExcitation.h.

4.32.2 Documentation des constructeurs et destructeur

4.32.2.1 FormCageExcitation : :FormCageExcitation (cageExcitation ∗ *cageExci,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 5 du fichier FormCageExcitation.cpp.

4.32.3 Documentation des fonctions membres

4.32.3.1 void FormCageExcitation : :actualiseFormulaire (int *val*) [slot]

Définition à la ligne 153 du fichier FormCageExcitation.cpp.

4.32.3.2 void FormCageExcitation : :changeCreate (bool *c*) [slot]

Définition à la ligne 159 du fichier FormCageExcitation.cpp.

4.32.3.3 void FormCageExcitation : :manageFormulaire (int *val*) [slot]

Définition à la ligne 128 du fichier FormCageExcitation.cpp.

4.32.3.4 void FormCageExcitation : :setCageExcitation (cageExcitation ∗ *c*) [inline]

Définition à la ligne 16 du fichier FormCageExcitation.h.

4.32.3.5 void FormCageExcitation : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 106 du fichier FormCageExcitation.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCageExcitation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCageExcitation.cpp**

4.33 Référence de la classe FormCalculDirective

Formulaire permettant de modifier l'objet **calculDirectivite** (p. 47).

```
#include <FormCalculDirective.h>
```

Fonctions membres publiques

– **FormCalculDirective** (**calculDirectivite** ∗calDir, QWidget ∗parent=0)
– void **setCaculDirective** (**calculDirectivite** ∗c)
– void **init** ()

4.33.1 Description détaillée

Formulaire permettant de modifier l'objet **calculDirectivite** (p. 47).

Définition à la ligne 10 du fichier FormCalculDirective.h.

4.33.2 Documentation des constructeurs et destructeur

4.33.2.1 FormCalculDirective : :FormCalculDirective (calculDirectivite ∗ *calDir,* QWidget ∗ *parent =* 0)

Définition à la ligne 3 du fichier FormCalculDirective.cpp.

4.33.3 Documentation des fonctions membres

4.33.3.1 void FormCalculDirective : :init ( )

Définition à la ligne 38 du fichier FormCalculDirective.cpp.

4.33.3.2 void FormCalculDirective : :setCaculDirective (calculDirectivite ∗ *c*) [inline]

Définition à la ligne 15 du fichier FormCalculDirective.h.

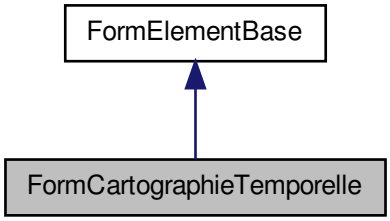
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCalculDirective.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCalculDirective.cpp**

4.34 Référence de la classe FormCartographieTemporelle

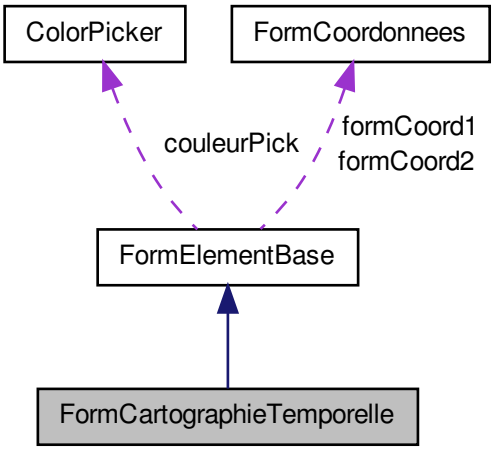
Formulaire permettant d'ajouter un nouvelle objet **cartographieTemporelle** (p. 49) dans la scène 3D.

```
#include <FormCartographieTemporelle.h>
```

Graphe d'héritage de FormCartographieTemporelle :



Graphe de collaboration de FormCartographieTemporelle :



Connecteurs publics

– void **valider** ()
  *Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).*
– void **reset** ()
  Place des valeurs par défaut dans le formulaire.

Fonctions membres publiques

– **FormCartographieTemporelle** (**cartographieTemporelle** ∗carto, QWidget ∗parent=0)

4.34.1 Description détaillée

Formulaire permettant d'ajouter un nouvelle objet **cartographieTemporelle** (p. 49) dans la scène 3D.

Définition à la ligne 10 du fichier FormCartographieTemporelle.h.

4.34.2 Documentation des constructeurs et destructeur

4.34.2.1 FormCartographieTemporelle : :FormCartographieTemporelle (cartographieTemporelle ∗ *carto,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormCartographieTemporelle.cpp.

4.34.3 Documentation des fonctions membres

4.34.3.1 void FormCartographieTemporelle : :reset ( ) [virtual, slot]

Place des valeurs par défaut dans le formulaire.

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 74 du fichier FormCartographieTemporelle.cpp.

4.34.3.2 void FormCartographieTemporelle : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 63 du fichier FormCartographieTemporelle.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCartographieTemporelle.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCartographieTemporelle.cpp**

4.35 Référence de la classe FormChampLointain

Cette classe fournit un formulaire pour modifier l'objet **champLointain** (p. 52).

```
#include <FormChampLointain.h>
```

Fonctions membres publiques

- **FormChampLointain** (**champLointain** ∗champ, QWidget ∗parent=0)
- void **setChampLointain** (**champLointain** ∗c)
- void **init** ()

4.35.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet **champLointain** (p. 52).

Définition à la ligne 11 du fichier FormChampLointain.h.

4.35.2 Documentation des constructeurs et destructeur

4.35.2.1 FormChampLointain : :FormChampLointain (champLointain ∗ *champ,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormChampLointain.cpp.

4.35.3 Documentation des fonctions membres

4.35.3.1 void FormChampLointain : :init ( )

Définition à la ligne 58 du fichier FormChampLointain.cpp.

4.35.3.2 void FormChampLointain : :setChampLointain (champLointain ∗ *c*) [inline]

Définition à la ligne 17 du fichier FormChampLointain.h.

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

- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormChampLointain.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormChampLointain.cpp**

4.36 Référence de la classe FormCompressionHuygens

Cette classe fournit un formulaire pour modifier l'objet **compressionHuygens** (p. 59).

```
#include <FormCompressionHuygens.h>
```

Fonctions membres publiques

- **FormCompressionHuygens** (**compressionHuygens** ∗compreHuy, QWidget ∗parent=0)
- void **setCompressHuy** (**compressionHuygens** ∗c)
- void **init** ()

4.36.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet **compressionHuygens** (p. 59).

Définition à la ligne 10 du fichier FormCompressionHuygens.h.

4.36.2 Documentation des constructeurs et destructeur

4.36.2.1 FormCompressionHuygens : :FormCompressionHuygens (compressionHuygens ∗ *compreHuy,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormCompressionHuygens.cpp.

4.36.3 Documentation des fonctions membres

4.36.3.1 void FormCompressionHuygens : :init ( )

Définition à la ligne 48 du fichier FormCompressionHuygens.cpp.

4.36.3.2 void FormCompressionHuygens : :setCompressHuy (compressionHuygens ∗ *c*) [inline]

Définition à la ligne 16 du fichier FormCompressionHuygens.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCompressionHuygens.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCompressionHuygens.cpp**

4.37 Référence de la classe FormCoordonnees

Cette classe fournit un widget pour modifier l'objet coordonnee.

```
#include <FormCoordonnees.h>
```

Connecteurs publics

- void **changeX** (int)
- void **changeY** (int)
- void **changeZ** (int)

Signaux

– void **coordonneeChanged** (**coordonnee**)

Fonctions membres publiques

– **FormCoordonnees** (QWidget ∗parent=0)
– **FormCoordonnees** (**coordonnee** coord, QWidget ∗parent=0)
– void **setCoordonnee** (**coordonnee** c)
– **coordonnee getCoord** ()

4.37.1 Description détaillée

Cette classe fournit un widget pour modifier l'objet coordonnee.

Définition à la ligne 10 du fichier FormCoordonnees.h.

4.37.2 Documentation des constructeurs et destructeur

4.37.2.1 FormCoordonnees : :FormCoordonnees (QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormCoordonnees.cpp.

4.37.2.2 FormCoordonnees : :FormCoordonnees (coordonnee *coord,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 10 du fichier FormCoordonnees.cpp.

4.37.3 Documentation des fonctions membres

4.37.3.1 void FormCoordonnees : :changeX (int *v*) [slot]

Définition à la ligne 30 du fichier FormCoordonnees.cpp.

4.37.3.2 void FormCoordonnees : :changeY (int *v*) [slot]

Définition à la ligne 36 du fichier FormCoordonnees.cpp.

4.37.3.3 void FormCoordonnees : :changeZ (int *v*) [slot]

Définition à la ligne 42 du fichier FormCoordonnees.cpp.

4.37.3.4 void FormCoordonnees : :coordonneeChanged (coordonnee) [signal]

4.37.3.5 coordonnee FormCoordonnees : :getCoord ( )

Définition à la ligne 25 du fichier FormCoordonnees.cpp.

4.37.3.6 void FormCoordonnees : :setCoordonnee (coordonnee *c*)

Définition à la ligne 17 du fichier FormCoordonnees.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCoordonnees.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormCoordonnees.cpp**

4.38 Référence de la classe FormEchantillonage

Cette classe fournit un formulaire pour modifier l'objet **blocEchantillonnage** (p. 22).

```
#include <FormEchantillonage.h>
```

Fonctions membres publiques

- **FormEchantillonage** (**blocEchantillonnage** ∗echanti, QWidget ∗parent=0)
- void **setEchantillonnage** (**blocEchantillonnage** ∗e)
- void **init** ()

4.38.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet **blocEchantillonnage** (p. 22).

Définition à la ligne 9 du fichier FormEchantillonage.h.

4.38.2 Documentation des constructeurs et destructeur

4.38.2.1 FormEchantillonage : :FormEchantillonage (blocEchantillonnage ∗ *echanti,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormEchantillonage.cpp.

4.38.3 Documentation des fonctions membres

4.38.3.1 void FormEchantillonage : :init ( )

Définition à la ligne 29 du fichier FormEchantillonage.cpp.

4.38.3.2 void FormEchantillonage : :setEchantillonnage (blocEchantillonnage * e) [inline]

Définition à la ligne 15 du fichier FormEchantillonage.h.

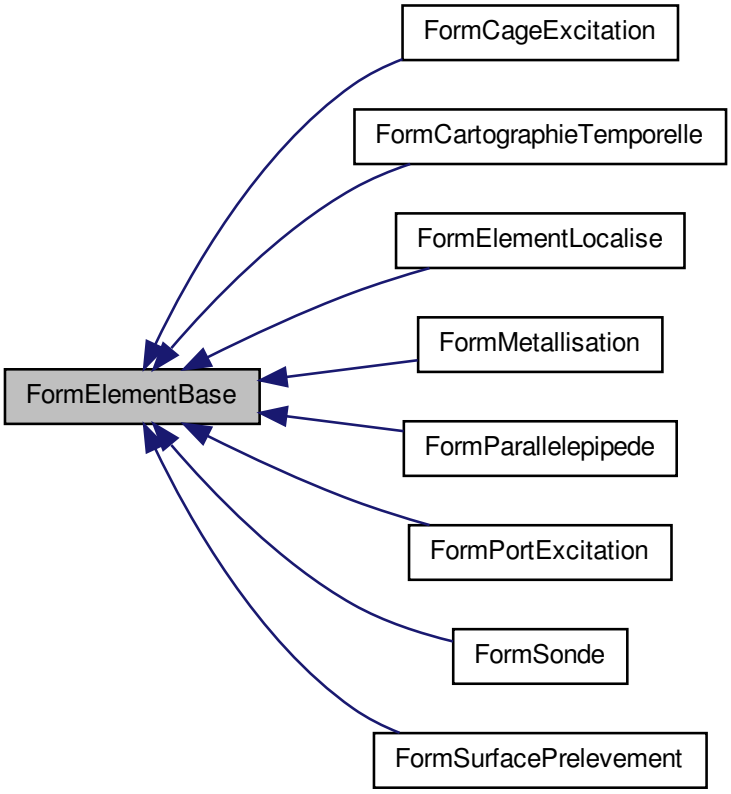
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormEchantillonage.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormEchantillonage.cpp**

4.39 Référence de la classe FormElementBase

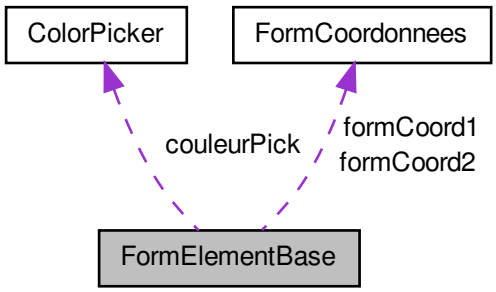
Cette classe sert de base aux formulaires qui modifie des objets **elementBase** (p. 73).

```
#include <FormElementBase.h>
```

Graphe d'héritage de FormElementBase :



Graphe de collaboration de FormElementBase :



Connecteurs publics

- virtual void **valider** ()
  *Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).*
- void **changeCouleur** (QColor)

Signaux

- void **elementValide** (**elementBase** ∗)
  *Ce signal est émis quand le bouton valider est pressé. Cela signifie que l'**elementBase** (p. 73) a bien était modifié.*

Fonctions membres publiques

- **FormElementBase** (**elementBase** ∗element, QWidget ∗parent=0)
- QWidget ∗ **getWidgetElementBase** ()
  *Cette fonction retourne les champs qui sont communs à tous les **elementBase** (p. 73). C'est à dire : le formulaire pour modifier les coordonnees ainsi que le bouton de validation.*

Fonctions membres protégées

- virtual void **reset** ()
  Place des valeurs par défaut dans le formulaire.
- virtual void **init** ()
  *Initialise le formulaire avec les données de l'objet **elementBase** (p. 73).*
- void **setElement** (**elementBase** ∗)
  *Modifie l'**elementBase** (p. 73) que contient le formulaire.*

Attributs protégés

- **FormCoordonnees** ∗ **formCoord1**
- **FormCoordonnees** ∗ **formCoord2**
- QPushButton ∗ **boutonValider**
- **ColorPicker** ∗ **couleurPick**

4.39.1 Description détaillée

Cette classe sert de base aux formulaires qui modifie des objets **elementBase** (p. 73).

Définition à la ligne 12 du fichier FormElementBase.h.

4.39.2 Documentation des constructeurs et destructeur

4.39.2.1 FormElementBase : :FormElementBase (elementBase ∗ *element,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormElementBase.cpp.

4.39.3 Documentation des fonctions membres

4.39.3.1 void FormElementBase : :changeCouleur (QColor *c*) [slot]

Définition à la ligne 48 du fichier FormElementBase.cpp.

4.39.3.2 void FormElementBase : :elementValide (elementBase ∗) [signal]

Ce signal est émis quand le bouton valider est pressé. Cela signifie que l'**elementBase** (p. 73) a bien était modifié.

4.39.3.3 QWidget ∗ FormElementBase : :getWidgetElementBase ( )

Cette fonction retourne les champs qui sont communs à tous les **elementBase** (p. 73). C'est à dire : le formulaire pour modifier les coordonnees ainsi que le bouton de validation.

Renvoie

Un widget qui contient un formulaire et un bouton valider

Définition à la ligne 9 du fichier FormElementBase.cpp.

4.39.3.4 void FormElementBase : :init ( ) [protected, virtual]

Initialise le formulaire avec les données de l'objet **elementBase** (p. 73).

Définition à la ligne 60 du fichier FormElementBase.cpp.

4.39.3.5 void FormElementBase : :reset ( ) [protected, virtual]

Place des valeurs par défaut dans le formulaire.

Réimplémentée dans **FormMetallisation** (p. 106), **FormElementLocalise** (p. 102), **FormParallelepipede** (p. 110), **FormCartographieTemporelle** (p. 91), **FormPortExcitation** (p. 115), **FormSonde** (p. 118), et **FormSurfacePrelevement** (p. 120).

Définition à la ligne 53 du fichier FormElementBase.cpp.

4.39.3.6 void FormElementBase : :setElement (elementBase ∗ *elem*) [protected]

Modifie l'**elementBase** (p. 73) que contient le formulaire.

Définition à la ligne 67 du fichier FormElementBase.cpp.

4.39.3.7 void FormElementBase : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée dans **FormCageExcitation** (p. 88), **FormMetallisation** (p. 106), **FormElementLocalise** (p. 102), **FormCartographieTemporelle** (p. 91), **FormParallelepipede** (p. 110), **FormPortExcitation** (p. 115), **FormSonde** (p. 118), et **FormSurfacePrelevement** (p. 120).

Définition à la ligne 42 du fichier FormElementBase.cpp.

4.39.4 Documentation des données membres

4.39.4.1 QPushButton∗ FormElementBase : :boutonValider [protected]

Définition à la ligne 58 du fichier FormElementBase.h.

4.39.4.2 ColorPicker∗ FormElementBase : :couleurPick [protected]

Définition à la ligne 60 du fichier FormElementBase.h.

4.39.4.3 FormCoordonnees∗ FormElementBase : :formCoord1 [protected]

Définition à la ligne 44 du fichier FormElementBase.h.

4.39.4.4 FormCoordonnees ∗ FormElementBase : :formCoord2 [protected]

Définition à la ligne 44 du fichier FormElementBase.h.

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

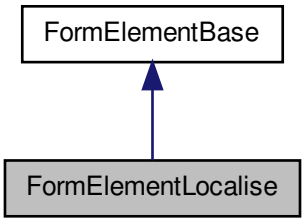
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementBase.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementBase.cpp**

4.40 Référence de la classe FormElementLocalise

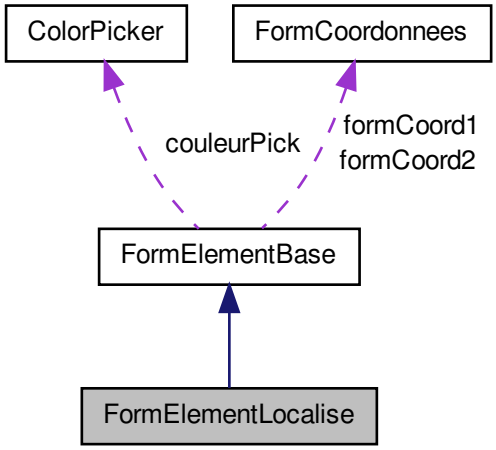
Formulaire permettant d'ajouter un nouvelle objet **elementLocalise** (p. 77) dans la scène 3D.

```
#include <FormElementLocalise.h>
```

Graphe d'héritage de FormElementLocalise :



Graphe de collaboration de FormElementLocalise :



Connecteurs publics

– void **valider** ()
  *Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).*
– void **reset** ()
  Place des valeurs par défaut dans le formulaire.

Fonctions membres publiques

– **FormElementLocalise** (**elementLocalise** ∗localise, QWidget ∗parent=0)

4.40.1 Description détaillée

Formulaire permettant d'ajouter un nouvelle objet **elementLocalise** (p. 77) dans la scène 3D.

Définition à la ligne 11 du fichier FormElementLocalise.h.

4.40.2 Documentation des constructeurs et destructeur

4.40.2.1 FormElementLocalise : :FormElementLocalise (elementLocalise * *localise,* QWidget * *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormElementLocalise.cpp.

4.40.3 Documentation des fonctions membres

4.40.3.1 void FormElementLocalise : :reset ( ) [virtual, slot]

Place des valeurs par défaut dans le formulaire.

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 60 du fichier FormElementLocalise.cpp.

4.40.3.2 void FormElementLocalise : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase *)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 51 du fichier FormElementLocalise.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementLocalise.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormElementLocalise.cpp**

4.41 Référence de la classe FormExcitation

Cette classe fournit un formulaire pour modifier l'objet **Excitation** (p. 80).

```
#include <FormExcitation.h>
```

Fonctions membres publiques

- **FormExcitation** (**Excitation** *excitation, QWidget *parent=0)
- void **setExcitation** (**Excitation** *e)
- void **init** ()

4.41.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet **Excitation** (p. 80).

Définition à la ligne 10 du fichier FormExcitation.h.

4.41.2 Documentation des constructeurs et destructeur

4.41.2.1 FormExcitation : :FormExcitation (Excitation ∗ *excitation,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormExcitation.cpp.

4.41.3 Documentation des fonctions membres

4.41.3.1 void FormExcitation : :init ( )

Définition à la ligne 29 du fichier FormExcitation.cpp.

4.41.3.2 void FormExcitation : :setExcitation (Excitation ∗ *e*) [inline]

Définition à la ligne 16 du fichier FormExcitation.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormExcitation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormExcitation.cpp**

4.42 Référence de la classe FormFormatStockage

Cette classe fournit un formulaire pour modifier l'objet **formatStockage** (p. 84).

```
#include <FormFormatStockage.h>
```

Fonctions membres publiques

– **FormFormatStockage** (**formatStockage** ∗format, QWidget ∗parent=0)
– void **setFormatStockage** (**formatStockage** ∗f)
– void **init** ()

4.42.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet **formatStockage** (p. 84).

Définition à la ligne 10 du fichier FormFormatStockage.h.

4.42.2 Documentation des constructeurs et destructeur

4.42.2.1 FormFormatStockage : :FormFormatStockage (formatStockage ∗ *format,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormFormatStockage.cpp.

4.42.3 Documentation des fonctions membres

4.42.3.1 void FormFormatStockage : :init ( )

Définition à la ligne 34 du fichier FormFormatStockage.cpp.

4.42.3.2 void FormFormatStockage : :setFormatStockage (formatStockage * *f*) [inline]

Définition à la ligne 16 du fichier FormFormatStockage.h.

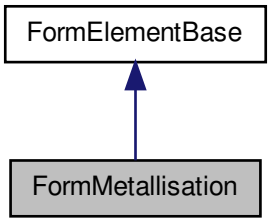
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormFormatStockage.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormFormatStockage.cpp**

4.43 Référence de la classe FormMetallisation

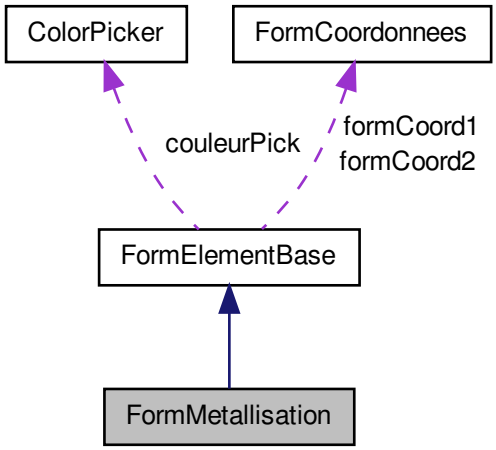
Formulaire permettant d'ajouter un nouvelle objet metallisation dans la scène 3D.

```
#include <FormMetallisation.h>
```

Graphe d'héritage de FormMetallisation :



Graphe de collaboration de FormMetallisation :



Connecteurs publics

– void **valider** ()
  *Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).*
– void **reset** ()
  Place des valeurs par défaut dans le formulaire.

Fonctions membres publiques

– **FormMetallisation** (**metallisation** ∗metal, QWidget ∗parent=0)
– **metallisation** ∗ **getMetallisation** ()

4.43.1 Description détaillée

Formulaire permettant d'ajouter un nouvelle objet metallisation dans la scène 3D.

Définition à la ligne 11 du fichier FormMetallisation.h.

4.43.2 Documentation des constructeurs et destructeur

4.43.2.1 FormMetallisation : :FormMetallisation (metallisation ∗ *metal,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormMetallisation.cpp.

4.43.3 Documentation des fonctions membres

4.43.3.1 metallisation∗ FormMetallisation : :getMetallisation ( )

4.43.3.2 void FormMetallisation : :reset ( ) [virtual, slot]

Place des valeurs par défaut dans le formulaire.

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 39 du fichier FormMetallisation.cpp.

4.43.3.3 void FormMetallisation : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 32 du fichier FormMetallisation.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormMetallisation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormMetallisation.cpp**

4.44 Référence de la classe FormNouveau

Cette classe est le formulaire pour ajouter une nouvelle simulation.

```
#include <FormNouveau.h>
```

Connecteurs publics

– void **testFichierExistant** (int n)
– void **setNum** (int i)

Fonctions membres publiques

– **FormNouveau** (QWidget ∗parent=0)
– int **getNum** ()

4.44.1 Description détaillée

Cette classe est le formulaire pour ajouter une nouvelle simulation.

Définition à la ligne 9 du fichier FormNouveau.h.

4.44.2 Documentation des constructeurs et destructeur

4.44.2.1 FormNouveau : :FormNouveau (QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormNouveau.cpp.

4.44.3 Documentation des fonctions membres

4.44.3.1 int FormNouveau : :getNum ( ) [inline]

Définition à la ligne 15 du fichier FormNouveau.h.

4.44.3.2 void FormNouveau : :setNum (int *i*) [slot]

Définition à la ligne 71 du fichier FormNouveau.cpp.

4.44.3.3 void FormNouveau : :testFichierExistant (int *n*) [slot]

Définition à la ligne 55 du fichier FormNouveau.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormNouveau.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormNouveau.cpp**

4.45 Référence de la classe FormOuvrir

Cette classe est le formulaire pour ouvrir une nouvelle simulation.

```
#include <FormOuvrir.h>
```

Connecteurs publics

- void **testFichierExistant** (int n)
- void **setNum** (int i)

Fonctions membres publiques

- **FormOuvrir** (QWidget ∗parent=0)
- int **getNum** ()

4.45.1 Description détaillée

Cette classe est le formulaire pour ouvrir une nouvelle simulation.

Définition à la ligne 9 du fichier FormOuvrir.h.

4.45.2 Documentation des constructeurs et destructeur

4.45.2.1 FormOuvrir : :FormOuvrir (QWidget * *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormOuvrir.cpp.

4.45.3 Documentation des fonctions membres

4.45.3.1 int FormOuvrir : :getNum ( ) [inline]

Définition à la ligne 15 du fichier FormOuvrir.h.

4.45.3.2 void FormOuvrir : :setNum (int *i*) [slot]

Définition à la ligne 68 du fichier FormOuvrir.cpp.

4.45.3.3 void FormOuvrir : :testFichierExistant (int *n*) [slot]

Définition à la ligne 52 du fichier FormOuvrir.cpp.

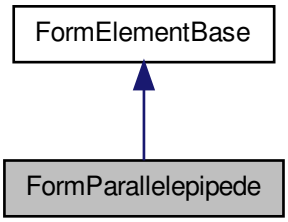
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormOuvrir.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormOuvrir.cpp**

4.46 Référence de la classe FormParallelepipede

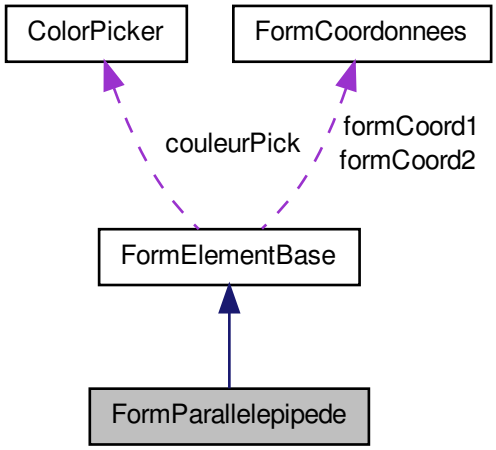
Formulaire permettant d'ajouter un nouvelle objet parallelepipede dans la scène 3D.

```
#include <FormParallelepipede.h>
```

Graphe d'héritage de FormParallelepipede :



Graphe de collaboration de FormParallelepipede :



Connecteurs publics

– void **valider** ()

*Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase*** (p. *73) et émet le signal **elementValide(elementBase ∗)*** (p. *98).*

– void **reset** ()

Place des valeurs par défaut dans le formulaire.

Fonctions membres publiques

– **FormParallelepipede** (**parallelepipede** ∗paralle, QWidget ∗parent=0)

4.46.1 Description détaillée

Formulaire permettant d'ajouter un nouvelle objet parallelepipede dans la scène 3D.

Définition à la ligne 10 du fichier FormParallelepipede.h.

4.46.2 Documentation des constructeurs et destructeur

4.46.2.1 FormParallelepipede : :FormParallelepipede (parallelepipede ∗ *paralle,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormParallelepipede.cpp.

4.46.3 Documentation des fonctions membres

4.46.3.1 void FormParallelepipede : :reset ( ) [virtual, slot]

Place des valeurs par défaut dans le formulaire.

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 48 du fichier FormParallelepipede.cpp.

4.46.3.2 void FormParallelepipede : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 39 du fichier FormParallelepipede.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormParallelepipede.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormParallelepipede.cpp**

4.47 Référence de la classe FormParamExcitation

Cette classe fournit un formulaire pour modifier l'objet **ParamExcitations** (p. 135).

```
#include <FormParamExcitation.h>
```

Connecteurs publics

– void **afficheSinus** (int)

Fonctions membres publiques

– **FormParamExcitation** (**ParamExcitations** ∗excit, QWidget ∗parent=0)
– void **setParamExcit** (**ParamExcitations** ∗ex)
– void **init** ()

4.47.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet **ParamExcitations** (p. 135).

Définition à la ligne 10 du fichier FormParamExcitation.h.

4.47.2 Documentation des constructeurs et destructeur

4.47.2.1 FormParamExcitation : :FormParamExcitation (ParamExcitations ∗ *excit,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormParamExcitation.cpp.

4.47.3 Documentation des fonctions membres

4.47.3.1 void FormParamExcitation : :afficheSinus (int *i*) [slot]

Définition à la ligne 64 du fichier FormParamExcitation.cpp.

4.47.3.2 void FormParamExcitation : :init ( )

Définition à la ligne 52 du fichier FormParamExcitation.cpp.

4.47.3.3 void FormParamExcitation : :setParamExcit (ParamExcitations ∗ *ex*) [inline]

Définition à la ligne 16 du fichier FormParamExcitation.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormParamExcitation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormParamExcitation.cpp**

4.48 Référence de la classe FormParoi

Cette classe fournit un formulaire pour modifier l'objet paroi.

```
#include <FormParoi.h>
```

Fonctions membres publiques

- **FormParoi** (**paroi** ∗m_paroi, QWidget ∗parent=0)
- void **setParoi** (**paroi** ∗p)
- void **init** ()

4.48.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet paroi.

Définition à la ligne 10 du fichier FormParoi.h.

4.48.2 Documentation des constructeurs et destructeur

4.48.2.1 FormParoi : :FormParoi (paroi ∗ *m_paroi,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormParoi.cpp.

4.48.3 Documentation des fonctions membres

4.48.3.1 void FormParoi : :init ( )

Définition à la ligne 42 du fichier FormParoi.cpp.

4.48.3.2 void FormParoi : :setParoi (paroi ∗ *p*) [inline]

Définition à la ligne 16 du fichier FormParoi.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormParoi.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormParoi.cpp**

4.49 Référence de la classe FormPML

Cette classe fournit un formulaire pour modifier l'objet pml.

```
#include <FormPML.h>
```

Fonctions membres publiques

- **FormPML** (**pml** ∗m_pml, QWidget ∗parent=0)
- void **setPML** (**pml** ∗p)
- void **init** ()

4.49.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet pml.

Définition à la ligne 10 du fichier FormPML.h.

4.49.2 Documentation des constructeurs et destructeur

4.49.2.1 FormPML : :FormPML (pml ∗ *m_pml,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormPML.cpp.

4.49.3 Documentation des fonctions membres

4.49.3.1 void FormPML : :init ( )

Définition à la ligne 39 du fichier FormPML.cpp.

4.49.3.2 void FormPML : :setPML (pml ∗ *p*) [inline]

Définition à la ligne 16 du fichier FormPML.h.

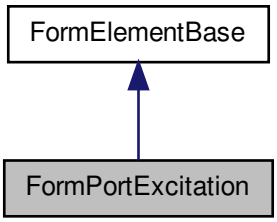
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormPML.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormPML.cpp**

4.50 Référence de la classe FormPortExcitation

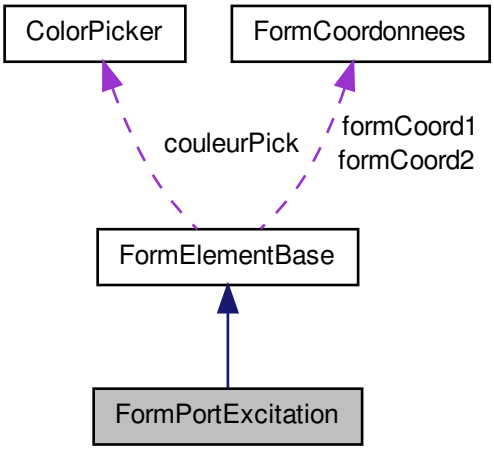
Formulaire permettant d'ajouter un nouvelle objet **portExcitation** (p. 147) dans la scène 3D.

```
#include <FormPortExcitation.h>
```

Graphe d'héritage de FormPortExcitation :



Graphe de collaboration de FormPortExcitation :



Connecteurs publics

– void **valider** ()

*Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).*

– void **reset** ()

Place des valeurs par défaut dans le formulaire.

Fonctions membres publiques

– **FormPortExcitation** (**portExcitation** ∗port, QWidget ∗parent=0)

4.50.1 Description détaillée

Formulaire permettant d'ajouter un nouvelle objet **portExcitation** (p. 147) dans la scène 3D.

Définition à la ligne 10 du fichier FormPortExcitation.h.

4.50.2 Documentation des constructeurs et destructeur

4.50.2.1 FormPortExcitation : :FormPortExcitation (portExcitation ∗ *port,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormPortExcitation.cpp.

4.50.3 Documentation des fonctions membres

4.50.3.1 void FormPortExcitation : :reset ( ) [virtual, slot]

Place des valeurs par défaut dans le formulaire.

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 101 du fichier FormPortExcitation.cpp.

4.50.3.2 void FormPortExcitation : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 87 du fichier FormPortExcitation.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormPortExcitation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormPortExcitation.cpp**

4.51 Référence de la classe FormSettings

Cette classe est le formulaire pour modifier les paramètres du logiciel.

```
#include <FormSettings.h>
```

Connecteurs publics

– void **ouvreDialogDossier** ()
– void **appliquer** ()

Fonctions membres publiques

– **FormSettings** (QWidget ∗parent=0)

4.51.1 Description détaillée

Cette classe est le formulaire pour modifier les paramètres du logiciel.

Définition à la ligne 9 du fichier FormSettings.h.

4.51.2 Documentation des constructeurs et destructeur

4.51.2.1 FormSettings : :FormSettings (QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier FormSettings.cpp.

4.51.3 Documentation des fonctions membres

4.51.3.1 void FormSettings : :appliquer ( ) [slot]

Définition à la ligne 59 du fichier FormSettings.cpp.

4.51.3.2 void FormSettings : :ouvreDialogDossier ( ) [slot]

Définition à la ligne 54 du fichier FormSettings.cpp.

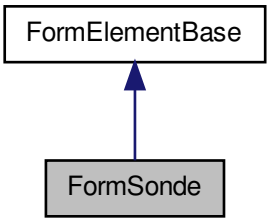
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormSettings.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormSettings.cpp**

4.52 Référence de la classe FormSonde

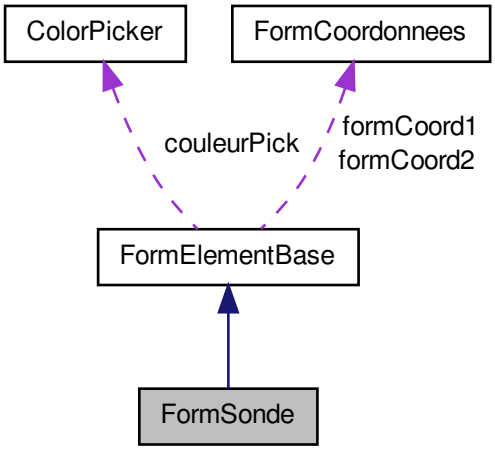
Formulaire permettant d'ajouter un nouvelle objet sonde dans la scène 3D.

```
#include <FormSonde.h>
```

Graphe d'héritage de FormSonde :



Graphe de collaboration de FormSonde :



Connecteurs publics

– void **valider** ()

*Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase*** (p. *73) et émet le signal **elementValide(elementBase ∗)*** (p. *98).*

– void **reset** ()

Place des valeurs par défaut dans le formulaire.

Fonctions membres publiques

– **FormSonde** (**sonde** ∗m_sonde, QWidget ∗parent=0)

4.52.1 Description détaillée

Formulaire permettant d'ajouter un nouvelle objet sonde dans la scène 3D.

Définition à la ligne 10 du fichier FormSonde.h.

4.52.2 Documentation des constructeurs et destructeur

4.52.2.1 FormSonde : :FormSonde (sonde ∗ *m_sonde,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormSonde.cpp.

4.52.3 Documentation des fonctions membres

4.52.3.1 void FormSonde : :reset ( ) [virtual, slot]

Place des valeurs par défaut dans le formulaire.

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 71 du fichier FormSonde.cpp.

4.52.3.2 void FormSonde : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 52 du fichier FormSonde.cpp.

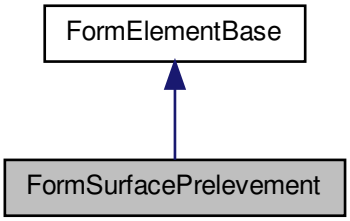
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormSonde.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormSonde.cpp**

4.53 Référence de la classe FormSurfacePrelevement

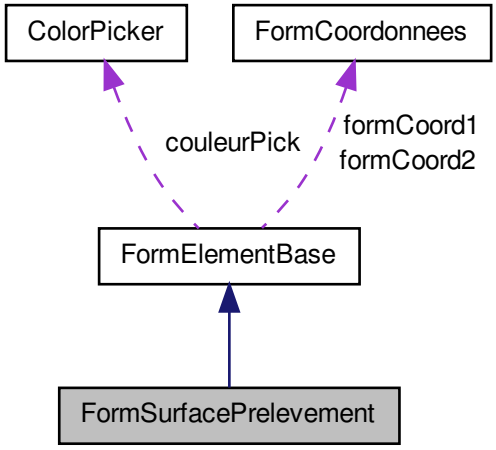
Formulaire permettant d'ajouter un nouvelle objet **surfacePrelevement** (p. 154) dans la scène 3D.

```
#include <FormSurfacePrelevement.h>
```

Graphe d'héritage de FormSurfacePrelevement :



Graphe de collaboration de FormSurfacePrelevement :



Connecteurs publics

– void **valider** ()
  *Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase*** (p. *73) et émet le signal **elementValide(elementBase ∗)*** (p. *98).*
– void **reset** ()
  Place des valeurs par défaut dans le formulaire.
– void **activeResteForm** (int val)

Fonctions membres publiques

– **FormSurfacePrelevement** (**surfacePrelevement** ∗surface, QWidget ∗parent=0)

4.53.1 Description détaillée

Formulaire permettant d'ajouter un nouvelle objet **surfacePrelevement** (p. 154) dans la scène 3D.

Définition à la ligne 10 du fichier FormSurfacePrelevement.h.

4.53.2 Documentation des constructeurs et destructeur

4.53.2.1 FormSurfacePrelevement : :FormSurfacePrelevement (surfacePrelevement ∗ *surface,* QWidget ∗ *parent =* 0)

Définition à la ligne 3 du fichier FormSurfacePrelevement.cpp.

4.53.3 Documentation des fonctions membres

4.53.3.1 void FormSurfacePrelevement : :activeResteForm (int *val*) [slot]

Définition à la ligne 87 du fichier FormSurfacePrelevement.cpp.

4.53.3.2 void FormSurfacePrelevement : :reset ( ) [virtual, slot]

Place des valeurs par défaut dans le formulaire.

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 79 du fichier FormSurfacePrelevement.cpp.

4.53.3.3 void FormSurfacePrelevement : :valider ( ) [virtual, slot]

Cette fonction est appelé quand le bouton valider est pressé. Elle met à jour les données de l'**elementBase** (p. 73) et émet le signal **elementValide(elementBase ∗)** (p. 98).

Réimplémentée à partir de **FormElementBase** (p. 99).

Définition à la ligne 64 du fichier FormSurfacePrelevement.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormSurfacePrelevement.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**FormSurfacePrelevement.cpp**

4.54 Référence de la classe FormVolumeCalcul

Cette classe fournit un formulaire pour modifier l'objet **volumeCalcul** (p. 201).

```
#include <FormVolumeCalcul.h>
```

Fonctions membres publiques

- **FormVolumeCalcul** (**volumeCalcul** ∗volCal, QWidget ∗parent=0)
- void **setVolumeCalcul** (**volumeCalcul** ∗vol)
- void **init** ()

4.54.1 Description détaillée

Cette classe fournit un formulaire pour modifier l'objet **volumeCalcul** (p. 201).

Définition à la ligne 10 du fichier FormVolumeCalcul.h.

4.54.2 Documentation des constructeurs et destructeur

4.54.2.1 FormVolumeCalcul : :FormVolumeCalcul (volumeCalcul ∗ *volCal,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier FormVolumeCalcul.cpp.

4.54.3 Documentation des fonctions membres

4.54.3.1 void FormVolumeCalcul : :init ( )

Définition à la ligne 80 du fichier FormVolumeCalcul.cpp.

4.54.3.2 void FormVolumeCalcul : :setVolumeCalcul (volumeCalcul ∗ *vol*) [inline]

Définition à la ligne 16 du fichier FormVolumeCalcul.h.

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

- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormVolumeCalcul.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**FormVolumeCalcul.cpp**

4.55 Référence de la classe MainWindow

The **MainWindow** (p. 121) class Cette classe représente la fenêtre principale du logiciel.

```
#include <mainwindow.h>
```

Connecteurs publics

- void **changeFilter** ()
  changeFilter
- void **nouvelleSimulation** ()
  initialise l'interface graphique avec des données vierges
- void **ouvrirSimulation** ()
  Charge les fichiers de configuration et initialise l'interface graphique.
- void **ouvreParametres** ()
  Ouvre la fenetre des paramètres du logiciel.
- void **ecrire** ()

Fonctions membres publiques

- **MainWindow** (QWidget ∗parent=0)
- ∼**MainWindow** ()

4.55.1 Description détaillée

The **MainWindow** (p. 121) class Cette classe représente la fenêtre principale du logiciel.

Définition à la ligne 29 du fichier mainwindow.h.

4.55.2 Documentation des constructeurs et destructeur

4.55.2.1 MainWindow : :MainWindow (QWidget ∗ *parent* = `0`)

Définition à la ligne 3 du fichier mainwindow.cpp.

4.55.2.2 MainWindow : :∼MainWindow ( )

Définition à la ligne 36 du fichier mainwindow.cpp.

4.55.3 Documentation des fonctions membres

4.55.3.1 void MainWindow : :changeFilter ( ) `[slot]`

changeFilter

Cette fonction est appelé dès que l'on change l'onglet courant (et donc le tableau a affiché). Elle modifie les champs du filtre dans le Widget **FilterWidget** (p. 82)

Définition à la ligne 41 du fichier mainwindow.cpp.

4.55.3.2 void MainWindow : :ecrire ( ) `[slot]`

Définition à la ligne 90 du fichier mainwindow.cpp.

4.55.3.3 void MainWindow : :nouvelleSimulation ( ) [slot]

initialise l'interface graphique avec des données vierges

Définition à la ligne 54 du fichier mainwindow.cpp.

4.55.3.4 void MainWindow : :ouvreParametres ( ) [slot]

Ouvre la fenetre des paramètres du logiciel.

Définition à la ligne 83 du fichier mainwindow.cpp.

4.55.3.5 void MainWindow : :ouvrirSimulation ( ) [slot]

Charge les fichiers de configuration et initialise l'interface graphique.

Définition à la ligne 69 du fichier mainwindow.cpp.

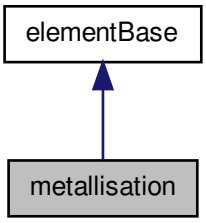
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**mainwindow.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**mainwindow.cpp**

4.56 Référence de la classe metallisation

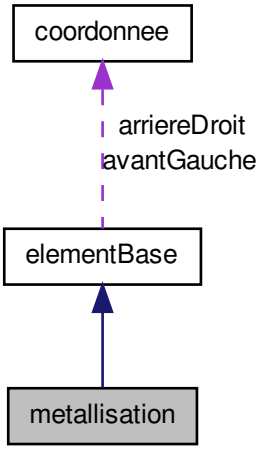
Cette classe représente la variable metallisation d'une simulation.

```
#include <metallisation.h>
```

Graphe d'héritage de metallisation :



Graphe de collaboration de metallisation :



Fonctions membres publiques

- **metallisation** ()
- virtual ~**metallisation** ()
- float **GetConductivite** () const
- void **SetConductivite** (float conductivite)
- void **ecrire** (int nb)

4.56.1 Description détaillée

Cette classe représente la variable metallisation d'une simulation.

Définition à la ligne 18 du fichier metallisation.h.

4.56.2 Documentation des constructeurs et destructeur

4.56.2.1 metallisation : :metallisation ( )

Définition à la ligne 10 du fichier metallisation.cpp.

4.56.2.2 metallisation : :~metallisation ( ) [virtual]

Définition à la ligne 14 du fichier metallisation.cpp.

4.56.3 Documentation des fonctions membres

4.56.3.1 void metallisation : :ecrire (int *nb*)

Définition à la ligne 17 du fichier metallisation.cpp.

4.56.3.2 float metallisation : :GetConductivite ( ) const [inline]

Définition à la ligne 23 du fichier metallisation.h.

4.56.3.3 void metallisation : :SetConductivite (float *conductivite*) [inline]

Définition à la ligne 27 du fichier metallisation.h.

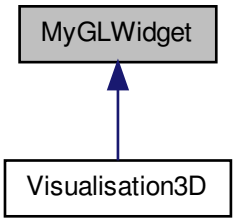
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**metallisation.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**metallisation.cpp**

4.57 Référence de la classe MyGLWidget

Cette classe sert de base à la classe qui affiche le rendu en 3D avec OpenGL.

```
#include <myglwidget.h>
```

Graphe d'héritage de MyGLWidget :



Connecteurs publics

- virtual void **timeOutSlot** ()

Fonctions membres publiques

- **MyGLWidget** (QWidget *parent=0)
- virtual void **initializeGL** ()=0
- virtual void **resizeGL** (int width, int height)=0
- virtual void **paintGL** ()=0
- virtual QSize **sizeHint** () const

4.57.1 Description détaillée

Cette classe sert de base à la classe qui affiche le rendu en 3D avec OpenGL.

Définition à la ligne 9 du fichier myglwidget.h.

4.57.2 Documentation des constructeurs et destructeur

4.57.2.1 MyGLWidget : :MyGLWidget (QWidget * *parent* = 0) [explicit]

Définition à la ligne 3 du fichier myglwidget.cpp.

4.57.3 Documentation des fonctions membres

4.57.3.1 virtual void MyGLWidget : :initializeGL ( ) [pure virtual]

Implémenté dans **Visualisation3D** (p. 198).

4.57.3.2 virtual void MyGLWidget : :paintGL ( ) [pure virtual]

Implémenté dans **Visualisation3D** (p. 199).

4.57.3.3 virtual void MyGLWidget : :resizeGL (int *width*, int *height*) [pure virtual]

Implémenté dans **Visualisation3D** (p. 199).

4.57.3.4 QSize MyGLWidget : :sizeHint ( ) const [virtual]

Définition à la ligne 25 du fichier myglwidget.cpp.

4.57.3.5 void MyGLWidget : :timeOutSlot ( ) [virtual, slot]

Définition à la ligne 20 du fichier myglwidget.cpp.

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

- /home/kuiderlix/Prog/cpp/pskl-etprat/**myglwidget.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**myglwidget.cpp**

4.58 Référence de la classe MySortFilterProxyModel

Cette classe permet de personnalisé le tri et le filtrage dans les listes.

```
#include <MySortFilterProxyModel.h>
```

Connecteurs publics

– void **setFilterKeyColumn** (int column)

Fonctions membres publiques

– **MySortFilterProxyModel** (QObject ∗parent=0)
– bool **lessThan** (const QModelIndex &left, const QModelIndex &right) const
– bool **filterAcceptsRow** (int sourceRow, const QModelIndex &sourceParent) const

4.58.1 Description détaillée

Cette classe permet de personnalisé le tri et le filtrage dans les listes.

Définition à la ligne 10 du fichier MySortFilterProxyModel.h.

4.58.2 Documentation des constructeurs et destructeur

4.58.2.1 MySortFilterProxyModel : :MySortFilterProxyModel (QObject ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier MySortFilterProxyModel.cpp.

4.58.3 Documentation des fonctions membres

4.58.3.1 bool MySortFilterProxyModel : :filterAcceptsRow (int *sourceRow,* const QModelIndex & *sourceParent*) const

Définition à la ligne 30 du fichier MySortFilterProxyModel.cpp.

4.58.3.2 bool MySortFilterProxyModel : :lessThan (const QModelIndex & *left,* const QModelIndex & *right*) const

Définition à la ligne 8 du fichier MySortFilterProxyModel.cpp.

4.58.3.3 void MySortFilterProxyModel : :setFilterKeyColumn (int *column*) [slot]

Définition à la ligne 25 du fichier MySortFilterProxyModel.cpp.

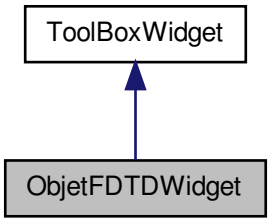
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**MySortFilterProxyModel.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**MySortFilterProxyModel.cpp**

4.59 Référence de la classe ObjetFDTDWidget

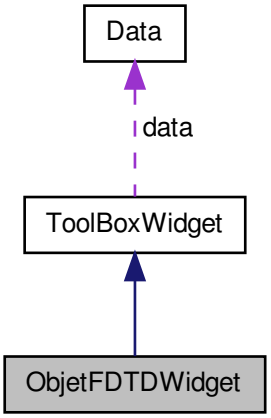
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

```
#include <ObjetFDTDWidget.h>
```

Graphe d'héritage de ObjetFDTDWidget :



Graphe de collaboration de ObjetFDTDWidget :



Connecteurs publics

– void **openCartoTempoWindows** ()

Signaux

– void **newSondeCreated** (**elementBase** ∗)
– void **newSurfacePrelCreated** (**elementBase** ∗)
– void **newCartoTempoCreated** (**elementBase** ∗)

Fonctions membres publiques

– **ObjetFDTDWidget** (**Data** ∗**data**, QWidget ∗parent=0)
– void **init** ()

4.59.1 Description détaillée

Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

Définition à la ligne 15 du fichier ObjetFDTDWidget.h.

4.59.2 Documentation des constructeurs et destructeur

4.59.2.1 ObjetFDTDWidget : :ObjetFDTDWidget (Data * *data,* QWidget * *parent* = 0) [explicit]

Définition à la ligne 3 du fichier ObjetFDTDWidget.cpp.

4.59.3 Documentation des fonctions membres

4.59.3.1 void ObjetFDTDWidget : :init ( ) [virtual]

Implémente **ToolBoxWidget** (p. 195).

Définition à la ligne 42 du fichier ObjetFDTDWidget.cpp.

4.59.3.2 void ObjetFDTDWidget : :newCartoTempoCreated (elementBase *) [signal]

4.59.3.3 void ObjetFDTDWidget : :newSondeCreated (elementBase *) [signal]

4.59.3.4 void ObjetFDTDWidget : :newSurfacePrelCreated (elementBase *) [signal]

4.59.3.5 void ObjetFDTDWidget : :openCartoTempoWindows ( ) [slot]

Définition à la ligne 47 du fichier ObjetFDTDWidget.cpp.

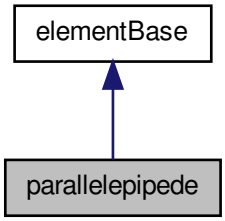
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ObjetFDTDWidget.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ObjetFDTDWidget.cpp**

4.60 Référence de la classe parallelepipede

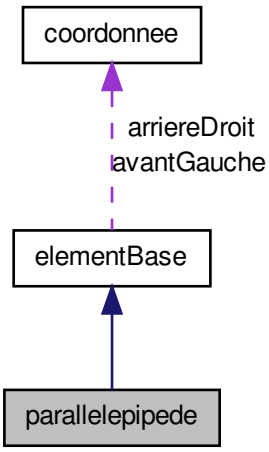
Cette classe représente la variable parallelepipede d'une simulation.

```
#include <parallelepipede.h>
```

Graphe d'héritage de parallelepipede :



Graphe de collaboration de parallelepipede :



Fonctions membres publiques

- **parallelepipede** ()
- virtual ~**parallelepipede** ()
- float **GetConductivite** () const
- void **SetConductivite** (float conductivite)
- float **GetPermeRelative** () const
- void **SetPermeRelative** (float permeRelative)

– float **GetPermitRelative** () const
– void **SetPermitRelative** (float permitRelative)
– void **ecrire** (int nb)

4.60.1 Description détaillée

Cette classe représente la variable parallelepipede d'une simulation.

Définition à la ligne 17 du fichier parallelepipede.h.

4.60.2 Documentation des constructeurs et destructeur

4.60.2.1 parallelepipede : :parallelepipede ( )

Définition à la ligne 13 du fichier parallelepipede.cpp.

4.60.2.2 parallelepipede : :~parallelepipede ( ) [virtual]

Définition à la ligne 20 du fichier parallelepipede.cpp.

4.60.3 Documentation des fonctions membres

4.60.3.1 void parallelepipede : :ecrire (int *nb*)

Définition à la ligne 23 du fichier parallelepipede.cpp.

4.60.3.2 float parallelepipede : :GetConductivite ( ) const [inline]

Définition à la ligne 22 du fichier parallelepipede.h.

4.60.3.3 float parallelepipede : :GetPermeRelative ( ) const [inline]

Définition à la ligne 30 du fichier parallelepipede.h.

4.60.3.4 float parallelepipede : :GetPermitRelative ( ) const [inline]

Définition à la ligne 38 du fichier parallelepipede.h.

4.60.3.5 void parallelepipede : :SetConductivite (float *conductivite*) [inline]

Définition à la ligne 26 du fichier parallelepipede.h.

4.60.3.6 void parallelepipede : :SetPermeRelative (float *permeRelative*) [inline]

Définition à la ligne 34 du fichier parallelepipede.h.

4.60.3.7 void parallelepipede : :SetPermitRelative (float *permitRelative*) [inline]

Définition à la ligne 42 du fichier parallelepipede.h.

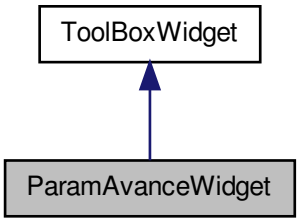
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**parallelepipede.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**parallelepipede.cpp**

4.61 Référence de la classe ParamAvanceWidget

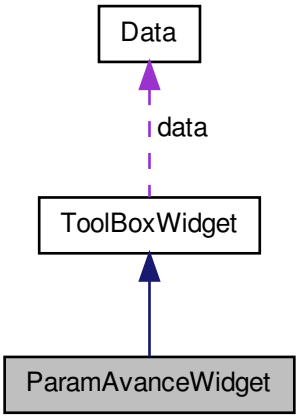
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

```
#include <ParamAvanceWidget.h>
```

Graphe d'héritage de ParamAvanceWidget :



Graphe de collaboration de ParamAvanceWidget :



Fonctions membres publiques

– **ParamAvanceWidget** (**Data** ∗**data**, QWidget ∗parent=0)
– void **init** ()

4.61.1 Description détaillée

Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

Définition à la ligne 15 du fichier ParamAvanceWidget.h.

4.61.2 Documentation des constructeurs et destructeur

4.61.2.1 ParamAvanceWidget : :ParamAvanceWidget (Data ∗ *data,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier ParamAvanceWidget.cpp.

4.61.3 Documentation des fonctions membres

4.61.3.1 void ParamAvanceWidget : :init ( ) [virtual]

Implémente **ToolBoxWidget** (p. 195).

Définition à la ligne 24 du fichier ParamAvanceWidget.cpp.

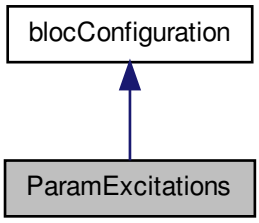
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**ParamAvanceWidget.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**ParamAvanceWidget.cpp**

4.62 Référence de la classe ParamExcitations

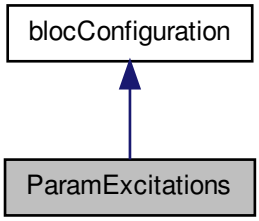
Cette classe représente la variable **ParamExcitations** (p. 135) d'une simulation.

```
#include <ParamExcitations.h>
```

Graphe d'héritage de ParamExcitations :



Graphe de collaboration de ParamExcitations :



Connecteurs publics

- void **SetFrequenceMax** (double frequenceMax)
- void **SetFrequenceSinus** (double frequenceSinus)
- void **SetType** (int type)
- void **ecrire** ()

Fonctions membres publiques

- **ParamExcitations** ()
- virtual ~**ParamExcitations** ()
- double **GetFrequenceMax** () const
- double **GetFrequenceSinus** () const
- int **GetType** () const

4.62.1 Description détaillée

Cette classe représente la variable **ParamExcitations** (p. 135) d'une simulation.

Définition à la ligne 16 du fichier ParamExcitations.h.

4.62.2 Documentation des constructeurs et destructeur

4.62.2.1 ParamExcitations : :ParamExcitations ( )

Définition à la ligne 10 du fichier ParamExcitations.cpp.

4.62.2.2 ParamExcitations : :~ParamExcitations ( ) [virtual]

Définition à la ligne 17 du fichier ParamExcitations.cpp.

4.62.3 Documentation des fonctions membres

4.62.3.1 void ParamExcitations : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 20 du fichier ParamExcitations.cpp.

4.62.3.2 double ParamExcitations : :GetFrequenceMax ( ) const [inline]

Définition à la ligne 23 du fichier ParamExcitations.h.

4.62.3.3 double ParamExcitations : :GetFrequenceSinus ( ) const [inline]

Définition à la ligne 27 du fichier ParamExcitations.h.

4.62.3.4 int ParamExcitations : :GetType ( ) const [inline]

Définition à la ligne 31 du fichier ParamExcitations.h.

4.62.3.5 void ParamExcitations : :SetFrequenceMax (double *frequenceMax*) [inline, slot]

Définition à la ligne 36 du fichier ParamExcitations.h.

4.62.3.6 void ParamExcitations : :SetFrequenceSinus (double *frequenceSinus*) [inline, slot]

Définition à la ligne 39 du fichier ParamExcitations.h.

4.62.3.7 void ParamExcitations : :SetType (int *type*) [inline, slot]

Définition à la ligne 42 du fichier ParamExcitations.h.

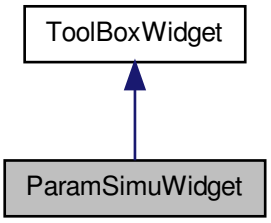
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ParamExcitations.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ParamExcitations.cpp**

4.63 Référence de la classe ParamSimuWidget

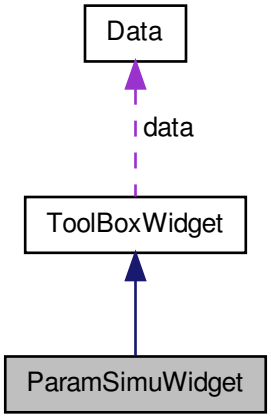
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

```
#include <ParamSimuWidget.h>
```

Graphe d'héritage de ParamSimuWidget :



Graphe de collaboration de ParamSimuWidget :



Signaux

– void **newPortExcitationCreated** (**elementBase** ∗)

Fonctions membres publiques

– **ParamSimuWidget** (**Data** ∗**data**, QWidget ∗parent=0)
– void **init** ()

4.63.1 Description détaillée

Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

Définition à la ligne 14 du fichier ParamSimuWidget.h.

4.63.2 Documentation des constructeurs et destructeur

4.63.2.1 ParamSimuWidget : :ParamSimuWidget (Data ∗ *data,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier ParamSimuWidget.cpp.

4.63.3 Documentation des fonctions membres

4.63.3.1 void ParamSimuWidget : :init ( ) [virtual]

Implémente **ToolBoxWidget** (p. 195).

Définition à la ligne 26 du fichier ParamSimuWidget.cpp.

4.63.3.2 void ParamSimuWidget : :newPortExcitationCreated (elementBase *) [signal]

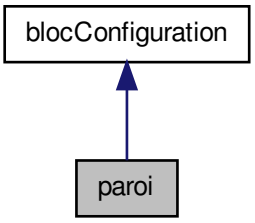
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ParamSimuWidget.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ParamSimuWidget.cpp**

4.64 Référence de la classe paroi

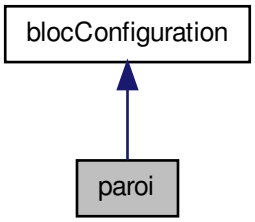
Cette classe représente la variable paroi d'une simulation.

`#include <paroi.h>`

Graphe d'héritage de paroi :



Graphe de collaboration de paroi :



Connecteurs publics

- void **SetParoiInferieure** (int paroiInferieure)
- void **SetParoiSuperieure** (int paroiSuperieure)
- void **SetParoiX** (int paroiX)
- void **SetParoiY** (int paroiY)
- void **ecrire** ()

Signaux

- void **paroiInfChanged** ()
- void **paroiSupChanged** ()
- void **paroiXChanged** ()
- void **paroiYChanged** ()

Fonctions membres publiques

- **paroi** ()
- virtual ∼**paroi** ()
- int **GetParoiInferieure** () const
- int **GetParoiSuperieure** () const
- int **GetParoiX** () const
- int **GetParoiY** () const

4.64.1 Description détaillée

Cette classe représente la variable paroi d'une simulation.

Définition à la ligne 17 du fichier paroi.h.

4.64.2 Documentation des constructeurs et destructeur

4.64.2.1 paroi : :paroi ( )

Définition à la ligne 10 du fichier paroi.cpp.

4.64.2.2 paroi : :~paroi ( ) [virtual]

Définition à la ligne 18 du fichier paroi.cpp.

4.64.3 Documentation des fonctions membres

4.64.3.1 void paroi : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 21 du fichier paroi.cpp.

4.64.3.2 int paroi : :GetParoiInferieure ( ) const [inline]

Définition à la ligne 24 du fichier paroi.h.

4.64.3.3 int paroi : :GetParoiSuperieure ( ) const [inline]

Définition à la ligne 27 du fichier paroi.h.

4.64.3.4 int paroi : :GetParoiX ( ) const [inline]

Définition à la ligne 30 du fichier paroi.h.

4.64.3.5 int paroi : :GetParoiY ( ) const [inline]

Définition à la ligne 33 du fichier paroi.h.

4.64.3.6 void paroi : :paroiInfChanged ( ) [signal]

4.64.3.7 void paroi : :paroiSupChanged ( ) [signal]

4.64.3.8 void paroi : :paroiXChanged ( ) [signal]

4.64.3.9 void paroi : :paroiYChanged ( ) [signal]

4.64.3.10 void paroi : :SetParoiInferieure (int *paroiInferieure*) [inline, slot]

Définition à la ligne 45 du fichier paroi.h.

4.64.3.11 void paroi : :SetParoiSuperieure (int *paroiSuperieure*) [inline, slot]

Définition à la ligne 48 du fichier paroi.h.

4.64.3.12 void paroi : :SetParoiX (int *paroiX*) [inline, slot]

Définition à la ligne 51 du fichier paroi.h.

4.64.3.13 void paroi : :SetParoiY (int *paroiY*) [inline, slot]

Définition à la ligne 54 du fichier paroi.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**paroi.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**paroi.cpp**

4.65 Référence de la classe Parser

Le parser de fichier de configuration.

```
#include <parser.h>
```

Fonctions membres publiques

- **Parser** ()
- void **parse** (**Data** ∗data)
  Fonction principale. Elle démarre le parsing.
- void **setNumFichier** (int num)
- QString **getEmplacementFichier** ()
- void **setEmplacementFichier** (QString e)
- int **getNumStru** ()

4.65.1 Description détaillée

Le parser de fichier de configuration.

Définition à la ligne 31 du fichier parser.h.

4.65.2 Documentation des constructeurs et destructeur

4.65.2.1 Parser : :Parser ( )

Définition à la ligne 3 du fichier parser.cpp.

4.65.3 Documentation des fonctions membres

4.65.3.1 QString Parser : :getEmplacementFichier ( ) [inline]

Définition à la ligne 47 du fichier parser.h.

4.65.3.2 int Parser : :getNumStru ( )

Définition à la ligne 547 du fichier parser.cpp.

4.65.3.3 void Parser : :parse (Data * *data*)

Fonction principale. Elle démarre le parsing.

Paramètres

data	Le conteneur des données récupérées.

>Fonction principale

Définition à la ligne 535 du fichier parser.cpp.

4.65.3.4 void Parser : :setEmplacementFichier (QString *e*) [inline]

Définition à la ligne 52 du fichier parser.h.

4.65.3.5 void Parser : :setNumFichier (int *num*) [inline]

Définition à la ligne 42 du fichier parser.h.

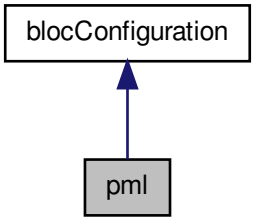
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**parser.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**parser.cpp**

4.66 Référence de la classe pml

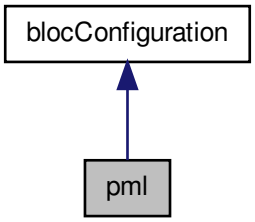
Cette classe représente la variable pml d'une simulation.

```
#include <pml.h>
```

Graphe d'héritage de pml :



Graphe de collaboration de pml :



Connecteurs publics

- void **SetEpaisseurCouche** (int epaisseurCouche)
- void **SetKMax** (double kMax)
- void **SetOrdreVariation** (int ordreVariation)
- void **SetSigmaMax** (double sigmaMax)
- void **ecrire** ()

Fonctions membres publiques

- **pml** ()
- virtual ~**pml** ()
- int **GetEpaisseurCouche** () const
- double **GetKMax** () const
- int **GetOrdreVariation** () const
- double **GetSigmaMax** () const

4.66.1 Description détaillée

Cette classe représente la variable pml d'une simulation.

Définition à la ligne 16 du fichier pml.h.

4.66.2 Documentation des constructeurs et destructeur

4.66.2.1 pml : :pml ( )

Définition à la ligne 10 du fichier pml.cpp.

4.66.2.2 pml : :~pml ( ) [virtual]

Définition à la ligne 17 du fichier pml.cpp.

4.66.3 Documentation des fonctions membres

4.66.3.1 void pml : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 20 du fichier pml.cpp.

4.66.3.2 int pml : :GetEpaisseurCouche ( ) const [inline]

Définition à la ligne 23 du fichier pml.h.

4.66.3.3 double pml : :GetKMax ( ) const [inline]

Définition à la ligne 27 du fichier pml.h.

4.66.3.4 int pml : :GetOrdreVariation ( ) const [inline]

Définition à la ligne 31 du fichier pml.h.

4.66.3.5 double pml : :GetSigmaMax ( ) const [inline]

Définition à la ligne 35 du fichier pml.h.

4.66.3.6 void pml : :SetEpaisseurCouche (int *epaisseurCouche*) [inline, slot]

Définition à la ligne 41 du fichier pml.h.

4.66.3.7 void pml : :SetKMax (double *kMax*) [inline, slot]

Définition à la ligne 45 du fichier pml.h.

4.66.3.8 void pml : :SetOrdreVariation (int *ordreVariation*) [inline, slot]

Définition à la ligne 49 du fichier pml.h.

4.66.3.9 void pml : :SetSigmaMax (double *sigmaMax*) [inline, slot]

Définition à la ligne 53 du fichier pml.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**pml.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**pml.cpp**

4.67 Référence de la classe Point

Cette classe représente un point dans un espace en 3D. Elle est utilisé pour afficher les **Cube** (p. 65) dans la vue 3D.

```
#include <point.h>
```

Fonctions membres publiques

- **Point** ()
- **Point** (float _x, float _y, float _z)

Attributs publics

- float **x**
- float **y**
- float **z**

4.67.1 Description détaillée

Cette classe représente un point dans un espace en 3D. Elle est utilisé pour afficher les **Cube** (p. 65) dans la vue 3D.

Définition à la ligne 8 du fichier point.h.

4.67.2 Documentation des constructeurs et destructeur

4.67.2.1 Point : :Point ( )

Définition à la ligne 3 du fichier point.cpp.

4.67.2.2 Point : :Point (float _x, float _y, float _z)

Définition à la ligne 10 du fichier point.cpp.

4.67.3 Documentation des données membres

4.67.3.1 float Point : :x

Définition à la ligne 14 du fichier point.h.

4.67.3.2 float Point : :y

Définition à la ligne 14 du fichier point.h.

4.67.3.3 float Point : :z

Définition à la ligne 14 du fichier point.h.

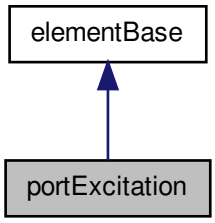
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**point.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**point.cpp**

4.68 Référence de la classe portExcitation

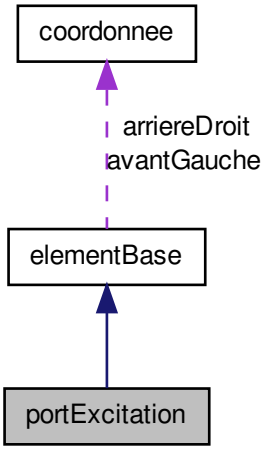
Cette classe représente la variable **portExcitation** (p. 147) d'une simulation.

```
#include <portExcitation.h>
```

Graphe d'héritage de portExcitation :



Graphe de collaboration de portExcitation :



Fonctions membres publiques

- **portExcitation** ()
- virtual ~**portExcitation** ()
- int **GetDirection** () const
- void **SetDirection** (int direction)
- float **GetImpedance** () const
- void **SetImpedance** (float impedance)
- float **GetPonderationAmplitude** () const
- void **SetPonderationAmplitude** (float ponderationAmplitude)
- float **GetPonderationPhase** () const
- void **SetPonderationPhase** (float ponderationPhase)
- bool **IsPortActif** () const
- void **SetPortActif** (bool portActif)
- bool **IsPortInfinitesimal** () const
- void **SetPortInfinitesimal** (bool portInfinitesimal)
- int **GetTypePort** () const
- void **SetTypePort** (int typePort)
- int **GetTypeSource** () const
- void **SetTypeSource** (int typeSource)
- void **ecrire** (int nb)

4.68.1 Description détaillée

Cette classe représente la variable **portExcitation** (p. 147) d'une simulation.

Définition à la ligne 16 du fichier portExcitation.h.

4.68.2 Documentation des constructeurs et destructeur

4.68.2.1 portExcitation : :portExcitation ( )

Définition à la ligne 13 du fichier portExcitation.cpp.

4.68.2.2 portExcitation : :~portExcitation ( ) [virtual]

Définition à la ligne 26 du fichier portExcitation.cpp.

4.68.3 Documentation des fonctions membres

4.68.3.1 void portExcitation : :ecrire (int *nb*)

Définition à la ligne 29 du fichier portExcitation.cpp.

4.68.3.2 int portExcitation : :GetDirection ( ) const [inline]

Définition à la ligne 21 du fichier portExcitation.h.

4.68.3.3 float portExcitation : :GetImpedance ( ) const [inline]

Définition à la ligne 29 du fichier portExcitation.h.

4.68.3.4 float portExcitation : :GetPonderationAmplitude ( ) const [inline]

Définition à la ligne 37 du fichier portExcitation.h.

4.68.3.5 float portExcitation : :GetPonderationPhase ( ) const [inline]

Définition à la ligne 45 du fichier portExcitation.h.

4.68.3.6 int portExcitation : :GetTypePort ( ) const [inline]

Définition à la ligne 69 du fichier portExcitation.h.

4.68.3.7 int portExcitation : :GetTypeSource ( ) const [inline]

Définition à la ligne 77 du fichier portExcitation.h.

4.68.3.8 bool portExcitation : :IsPortActif ( ) const [inline]

Définition à la ligne 53 du fichier portExcitation.h.

4.68.3.9 bool portExcitation : :IsPortInfinitesimal ( ) const [inline]

Définition à la ligne 61 du fichier portExcitation.h.

4.68.3.10 void portExcitation : :SetDirection (int *direction*) [inline]

Définition à la ligne 25 du fichier portExcitation.h.

4.68.3.11 void portExcitation : :SetImpedance (float *impedance*) [inline]

Définition à la ligne 33 du fichier portExcitation.h.

4.68.3.12 void portExcitation : :SetPonderationAmplitude (float *ponderationAmplitude*) [inline]

Définition à la ligne 41 du fichier portExcitation.h.

4.68.3.13 void portExcitation : :SetPonderationPhase (float *ponderationPhase*) [inline]

Définition à la ligne 49 du fichier portExcitation.h.

4.68.3.14 void portExcitation : :SetPortActif (bool *portActif*) [inline]

Définition à la ligne 57 du fichier portExcitation.h.

4.68.3.15 void portExcitation : :SetPortInfinitesimal (bool *portInfinitesimal*) [inline]

Définition à la ligne 65 du fichier portExcitation.h.

4.68.3.16 void portExcitation : :SetTypePort (int *typePort*) [inline]

Définition à la ligne 73 du fichier portExcitation.h.

4.68.3.17 void portExcitation : :SetTypeSource (int *typeSource*) [inline]

Définition à la ligne 81 du fichier portExcitation.h.

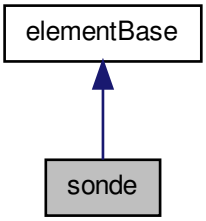
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**portExcitation.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**portExcitation.cpp**

4.69 Référence de la classe sonde

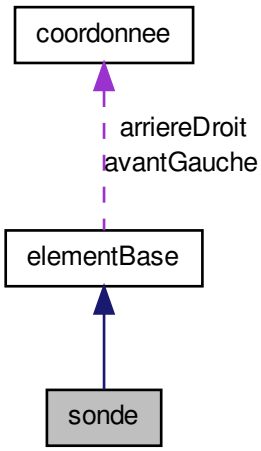
Cette classe représente la variable sonde d'une simulation.

`#include <sonde.h>`

Graphe d'héritage de sonde :



Graphe de collaboration de sonde :



Fonctions membres publiques

- **sonde** ()
- virtual ∼**sonde** ()
- bool **IsEx** () const
- void **SetEx** (bool ex)
- bool **IsEy** () const
- void **SetEy** (bool ey)
- bool **IsEz** () const
- void **SetEz** (bool ez)
- bool **IsHx** () const
- void **SetHx** (bool hx)
- bool **IsHy** () const
- void **SetHy** (bool hy)
- bool **IsHz** () const
- void **SetHz** (bool hz)
- **coordonnee GetPointApplication** () const
- void **SetPointApplication** (int x, int y, int z)
- void **SetPointApplication** (**coordonnee** c)
- bool **IsValeurAuCentre** () const
- void **SetValeurAuCentre** (bool valeurAuCentre)
- void **ecrire** (int nb)

4.69.1 Description détaillée

Cette classe représente la variable sonde d'une simulation.

Définition à la ligne 17 du fichier sonde.h.

4.69.2 Documentation des constructeurs et destructeur

4.69.2.1 sonde : :sonde ( )

Définition à la ligne 13 du fichier sonde.cpp.

4.69.2.2 sonde : :∼sonde ( ) [virtual]

Définition à la ligne 24 du fichier sonde.cpp.

4.69.3 Documentation des fonctions membres

4.69.3.1 void sonde : :ecrire (int *nb*)

Définition à la ligne 27 du fichier sonde.cpp.

4.69.3.2 coordonnee sonde : :GetPointApplication ( ) const [inline]

Définition à la ligne 70 du fichier sonde.h.

4.69.3.3 bool sonde : :IsEx ( ) const [inline]

Définition à la ligne 22 du fichier sonde.h.

4.69.3.4 bool sonde : :IsEy ( ) const [inline]

Définition à la ligne 30 du fichier sonde.h.

4.69.3.5 bool sonde : :IsEz ( ) const [inline]

Définition à la ligne 38 du fichier sonde.h.

4.69.3.6 bool sonde : :IsHx ( ) const [inline]

Définition à la ligne 46 du fichier sonde.h.

4.69.3.7 bool sonde : :IsHy ( ) const [inline]

Définition à la ligne 54 du fichier sonde.h.

4.69.3.8 bool sonde : :IsHz ( ) const [inline]

Définition à la ligne 62 du fichier sonde.h.

4.69.3.9 bool sonde : :IsValeurAuCentre ( ) const [inline]

Définition à la ligne 81 du fichier sonde.h.

4.69.3.10 void sonde : :SetEx (bool *ex*) [inline]

Définition à la ligne 26 du fichier sonde.h.

4.69.3.11 void sonde : :SetEy (bool *ey*) [inline]

Définition à la ligne 34 du fichier sonde.h.

4.69.3.12 void sonde : :SetEz (bool *ez*) [inline]

Définition à la ligne 42 du fichier sonde.h.

4.69.3.13 void sonde : :SetHx (bool *hx*) [inline]

Définition à la ligne 50 du fichier sonde.h.

4.69.3.14 void sonde : :SetHy (bool *hy*) [inline]

Définition à la ligne 58 du fichier sonde.h.

4.69.3.15 void sonde : :SetHz (bool *hz*) [inline]

Définition à la ligne 66 du fichier sonde.h.

4.69.3.16 void sonde : :SetPointApplication (int *x*, int *y*, int *z*) [inline]

Définition à la ligne 74 du fichier sonde.h.

4.69.3.17 void sonde : :SetPointApplication (coordonnee *c*) [inline]

Définition à la ligne 77 du fichier sonde.h.

4.69.3.18 void sonde : :SetValeurAuCentre (bool *valeurAuCentre*) [inline]

Définition à la ligne 85 du fichier sonde.h.

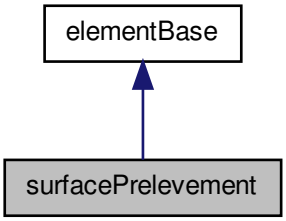
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**sonde.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**sonde.cpp**

4.70 Référence de la classe surfacePrelevement

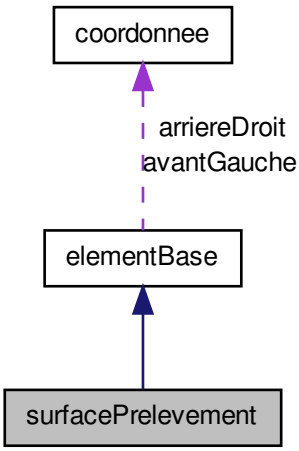
Cette classe représente la variable **surfacePrelevement** (p. 154) d'une simulation.

```
#include <surfacePrelevement.h>
```

Graphe d'héritage de surfacePrelevement :



Graphe de collaboration de surfacePrelevement :



Fonctions membres publiques

- **surfacePrelevement** ()
- **surfacePrelevement** (const **surfacePrelevement** &orig)
- virtual ~**surfacePrelevement** ()
- int **GetCompressionHauteur** () const
- void **SetCompressionHauteur** (int compressionHauteur)
- int **GetCompressionLargeur** () const

- void **SetCompressionLargeur** (int compressionLargeur)
- int **GetCompressionLongueur** () const
- void **SetCompressionLongueur** (int compressionLongueur)
- int **GetInsideOutside** () const
- void **SetInsideOutside** (int insideOutside)
- int **GetTypeSurface** () const
- void **SetTypeSurface** (int typeSurface)
- std : :string **GetNomFichier** () const
- void **SetNomFichier** (std : :string nomFichier)
- int **GetNbFaces** () const
- void **SetNbFaces** (int nbFaces)
- void **ecrire** (int type)

4.70.1 Description détaillée

Cette classe représente la variable **surfacePrelevement** (p. 154) d'une simulation.

Définition à la ligne 17 du fichier surfacePrelevement.h.

4.70.2 Documentation des constructeurs et destructeur

4.70.2.1 surfacePrelevement : :surfacePrelevement ( )

Définition à la ligne 13 du fichier surfacePrelevement.cpp.

4.70.2.2 surfacePrelevement : :surfacePrelevement (const surfacePrelevement & *orig*)

4.70.2.3 surfacePrelevement : :~surfacePrelevement ( ) [virtual]

Définition à la ligne 24 du fichier surfacePrelevement.cpp.

4.70.3 Documentation des fonctions membres

4.70.3.1 void surfacePrelevement : :ecrire (int *type*)

Définition à la ligne 27 du fichier surfacePrelevement.cpp.

4.70.3.2 int surfacePrelevement : :GetCompressionHauteur ( ) const [inline]

Définition à la ligne 23 du fichier surfacePrelevement.h.

4.70.3.3 int surfacePrelevement : :GetCompressionLargeur ( ) const [inline]

Définition à la ligne 31 du fichier surfacePrelevement.h.

4.70.3.4 int surfacePrelevement : :GetCompressionLongueur ( ) const [inline]

Définition à la ligne 39 du fichier surfacePrelevement.h.

4.70.3.5 int surfacePrelevement : :GetInsideOutside ( ) const [inline]

Définition à la ligne 47 du fichier surfacePrelevement.h.

4.70.3.6 int surfacePrelevement : :GetNbFaces ( ) const [inline]

Définition à la ligne 71 du fichier surfacePrelevement.h.

4.70.3.7 std : :string surfacePrelevement : :GetNomFichier ( ) const [inline]

Définition à la ligne 63 du fichier surfacePrelevement.h.

4.70.3.8 int surfacePrelevement : :GetTypeSurface ( ) const [inline]

Définition à la ligne 55 du fichier surfacePrelevement.h.

4.70.3.9 void surfacePrelevement : :SetCompressionHauteur (int *compressionHauteur*) [inline]

Définition à la ligne 27 du fichier surfacePrelevement.h.

4.70.3.10 void surfacePrelevement : :SetCompressionLargeur (int *compressionLargeur*) [inline]

Définition à la ligne 35 du fichier surfacePrelevement.h.

4.70.3.11 void surfacePrelevement : :SetCompressionLongueur (int *compressionLongueur*) [inline]

Définition à la ligne 43 du fichier surfacePrelevement.h.

4.70.3.12 void surfacePrelevement : :SetInsideOutside (int *insideOutside*) [inline]

Définition à la ligne 51 du fichier surfacePrelevement.h.

4.70.3.13 void surfacePrelevement : :SetNbFaces (int *nbFaces*) [inline]

Définition à la ligne 75 du fichier surfacePrelevement.h.

4.70.3.14 void surfacePrelevement : :SetNomFichier (std : :string *nomFichier*) [inline]

Définition à la ligne 67 du fichier surfacePrelevement.h.

4.70.3.15 void surfacePrelevement : :SetTypeSurface (int *typeSurface*) [inline]

Définition à la ligne 59 du fichier surfacePrelevement.h.

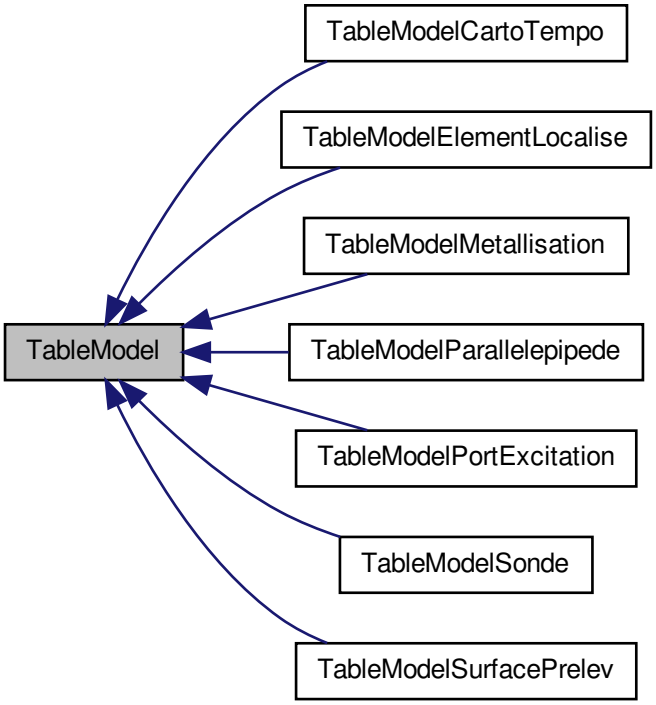
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**surfacePrelevement.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**surfacePrelevement.cpp**

4.71 Référence de la classe TableModel

Cette classe sert de model de base pour les éléments des tableaux.

```
#include <TableModel.h>
```

Graphe d'héritage de TableModel :



Connecteurs publics

- void **addElement** (**elementBase** ∗element)
  ajout un élément à la fin du tableau
- void **removeElement** (int i)
  supprime un élément du tableau

Fonctions membres publiques

- **TableModel** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
- int **rowCount** (const QModelIndex &parent=QModelIndex()) const
- Qt : :ItemFlags **flags** (const QModelIndex &index) const
- QVariant **headerData** (int section, Qt : :Orientation orientation, int role=Qt : :Display-Role) const
- void **initHeaders** (const QStringList l)
  Initialise les entetes du tableau.
- QStringList **getHeaders** () const
  Retourne la liste des entetes du tableau.

– **elementBase** ∗ **getElement** (int row)
Retourne un élément contenu dans le tableau au rang row.

4.71.1 Description détaillée

Cette classe sert de model de base pour les éléments des tableaux.

Définition à la ligne 11 du fichier TableModel.h.

4.71.2 Documentation des constructeurs et destructeur

4.71.2.1 TableModel : :TableModel (BlocElementBase ∗ *bloc,* QObject ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableModel.cpp.

4.71.3 Documentation des fonctions membres

4.71.3.1 void TableModel : :addElement (elementBase ∗ *element*) [slot]

ajout un élément à la fin du tableau

Paramètres

element	l'élément a ajouté

Définition à la ligne 20 du fichier TableModel.cpp.

4.71.3.2 Qt : :ItemFlags TableModel : :flags (const QModelIndex & *index*) const

Réimplémentée dans **TableModelSurfacePrelev** (p. 177).

Définition à la ligne 10 du fichier TableModel.cpp.

4.71.3.3 elementBase ∗ TableModel : :getElement (int *row*)

Retourne un élément contenu dans le tableau au rang row.

Paramètres

row	

Renvoie

l'élément à l'indice row

Définition à la ligne 50 du fichier TableModel.cpp.

4.71.3.4 QStringList TableModel : :getHeaders ( ) const [inline]

Retourne la liste des entetes du tableau.

Renvoie

Définition à la ligne 32 du fichier TableModel.h.

4.71.3.5 QVariant TableModel : :headerData (int *section,* Qt : :Orientation *orientation,* int *role* = Qt : : DisplayRole) const

Définition à la ligne 35 du fichier TableModel.cpp.

4.71.3.6 void TableModel : :initHeaders (const QStringList *l*)

Initialise les entetes du tableau.

Paramètres

l	la liste des entetes

Définition à la ligne 45 du fichier TableModel.cpp.

4.71.3.7 void TableModel : :removeElement (int *i*) [slot]

supprime un élément du tableau

Paramètres

i	le numéro dans le tableau de l'élément à supprimé

Définition à la ligne 28 du fichier TableModel.cpp.

4.71.3.8 int TableModel : :rowCount (const QModelIndex & *parent* = QModelIndex()) const

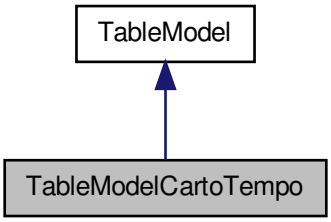
Définition à la ligne 16 du fichier TableModel.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModel.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModel.cpp**

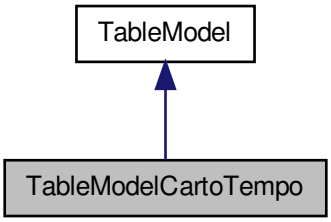
4.72 Référence de la classe TableModelCartoTempo

```
#include <TableModelCartoTempo.h>
```

Graphe d'héritage de TableModelCartoTempo :



Graphe de collaboration de TableModelCartoTempo :



Types publics

– enum **Columns** { **champE** = 0, **champH** = 1, **iterationDebut** = 2, **iterationFin** = 3, **intervalle** = 4, **AvantGauche** = 5, **ArriereDroit** = 6 }

Fonctions membres publiques

– **TableModelCartoTempo** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
– QVariant **data** (const QModelIndex &index, int role=Qt : :DisplayRole) const
– int **columnCount** (const QModelIndex &parent=QModelIndex()) const
– bool **setData** (const QModelIndex &index, const QVariant &value, int role)

4.72.1 Description détaillée

Définition à la ligne 8 du fichier TableModelCartoTempo.h.

4.72.2 Documentation des énumérations membres

4.72.2.1 enum TableModelCartoTempo : :Columns

Valeurs énumérées :

champE

champH

iterationDebut

iterationFin

intervalle

AvantGauche

ArriereDroit

Définition à la ligne 11 du fichier TableModelCartoTempo.h.

4.72.3 Documentation des constructeurs et destructeur

4.72.3.1 TableModelCartoTempo : :TableModelCartoTempo (BlocElementBase * *bloc,* QObject * *parent =* `0`) `[explicit]`

Définition à la ligne 3 du fichier TableModelCartoTempo.cpp.

4.72.4 Documentation des fonctions membres

4.72.4.1 int TableModelCartoTempo : :columnCount (const QModelIndex & *parent =* `QModelIndex()`) const

Définition à la ligne 59 du fichier TableModelCartoTempo.cpp.

4.72.4.2 QVariant TableModelCartoTempo : :data (const QModelIndex & *index,* int *role =* `Qt::DisplayRole`) const

Définition à la ligne 10 du fichier TableModelCartoTempo.cpp.

4.72.4.3 bool TableModelCartoTempo : :setData (const QModelIndex & *index,* const QVariant & *value,* int *role*)

Définition à la ligne 64 du fichier TableModelCartoTempo.cpp.

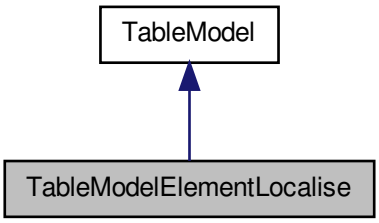
La documentation de cette classe a été générée à partir des fichiers suivants :

– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelCartoTempo.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelCartoTempo.cpp**

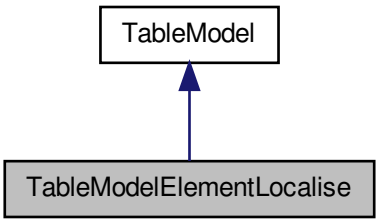
4.73 Référence de la classe TableModelElementLocalise

```
#include <TableModelElementLocalise.h>
```

Graphe d'héritage de TableModelElementLocalise :



Graphe de collaboration de TableModelElementLocalise :



Types publics

– enum **Columns** { **Type** = 0, **Valeur** = 1, **Direction** = 2, **AvantGauche** = 3, **Arriere-Droit** = 4 }

Fonctions membres publiques

- **TableModelElementLocalise** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
- QVariant **data** (const QModelIndex &index, int role=Qt : :DisplayRole) const
- int **columnCount** (const QModelIndex &parent=QModelIndex()) const
- bool **setData** (const QModelIndex &index, const QVariant &value, int role)

4.73.1 Description détaillée

Définition à la ligne 7 du fichier TableModelElementLocalise.h.

4.73.2 Documentation des énumérations membres

4.73.2.1 enum TableModelElementLocalise : :Columns

Valeurs énumérées :

Type

Valeur

Direction

AvantGauche

ArriereDroit

Définition à la ligne 11 du fichier TableModelElementLocalise.h.

4.73.3 Documentation des constructeurs et destructeur

4.73.3.1 TableModelElementLocalise : :TableModelElementLocalise (BlocElementBase ∗ *bloc,* QObject ∗ *parent =* `0`) `[explicit]`

Définition à la ligne 4 du fichier TableModelElementLocalise.cpp.

4.73.4 Documentation des fonctions membres

4.73.4.1 int TableModelElementLocalise : :columnCount (const QModelIndex & *parent* = `QModelIndex()`) const

Définition à la ligne 54 du fichier TableModelElementLocalise.cpp.

4.73.4.2 QVariant TableModelElementLocalise : :data (const QModelIndex & *index,* int *role =* `Qt : :DisplayRole`) const

Définition à la ligne 12 du fichier TableModelElementLocalise.cpp.

4.73.4.3 bool TableModelElementLocalise : :setData (const QModelIndex & *index,* const QVariant & *value,* int *role*)

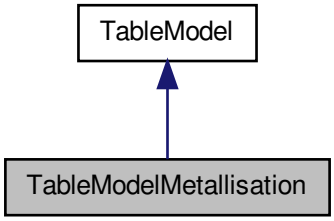
Définition à la ligne 59 du fichier TableModelElementLocalise.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelElementLocalise.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelElementLocalise.cpp**

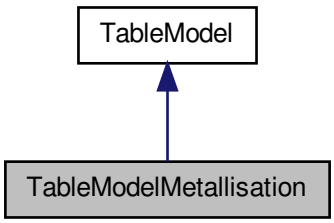
4.74 Référence de la classe TableModelMetallisation

```
#include <TableModelMetallisation.h>
```

Graphe d'héritage de TableModelMetallisation :



Graphe de collaboration de TableModelMetallisation :



Types publics

– enum **Columns** { **Conductivite** = 0, **Coord1** = 1, **Coord2** = 2 }

Fonctions membres publiques

– **TableModelMetallisation** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
– QVariant **data** (const QModelIndex &index, int role=Qt : :DisplayRole) const
– int **columnCount** (const QModelIndex &parent=QModelIndex()) const
– bool **setData** (const QModelIndex &index, const QVariant &value, int role)

4.74.1 Description détaillée

Définition à la ligne 7 du fichier TableModelMetallisation.h.

4.74.2 Documentation des énumérations membres

4.74.2.1 enum TableModelMetallisation : :Columns

Valeurs énumérées :

Conductivite

Coord1

Coord2

Définition à la ligne 11 du fichier TableModelMetallisation.h.

4.74.3 Documentation des constructeurs et destructeur

4.74.3.1 TableModelMetallisation : :TableModelMetallisation (BlocElementBase ∗ *bloc,* QObject ∗ *parent* = `0`) `[explicit]`

Définition à la ligne 3 du fichier TableModelMetallisation.cpp.

4.74.4 Documentation des fonctions membres

4.74.4.1 int TableModelMetallisation : :columnCount (const QModelIndex & *parent* = `QModelIndex()`) const

Définition à la ligne 12 du fichier TableModelMetallisation.cpp.

4.74.4.2 QVariant TableModelMetallisation : :data (const QModelIndex & *index,* int *role* = `Qt::DisplayRole`) const

Définition à la ligne 17 du fichier TableModelMetallisation.cpp.

4.74.4.3 bool TableModelMetallisation : :setData (const QModelIndex & *index,* const QVariant & *value,* int *role*)

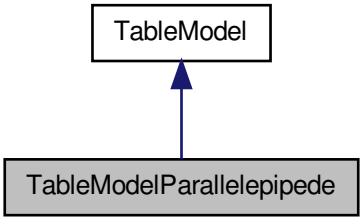
Définition à la ligne 52 du fichier TableModelMetallisation.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelMetallisation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelMetallisation.cpp**

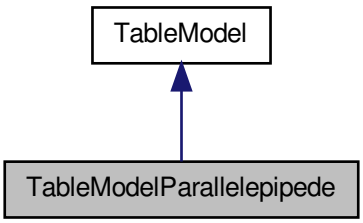
4.75 Référence de la classe TableModelParallelepipede

```
#include <TableModelParallelepipede.h>
```

Graphe d'héritage de TableModelParallelepipede :



Graphe de collaboration de TableModelParallelepipede :



Types publics

– enum **Columns** { **PermitRelative** = 0, **PermeRelative** = 1, **Conductivite** = 2, **Coord1** = 3, **Coord2** = 4 }

Fonctions membres publiques

– **TableModelParallelepipede** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
– QVariant **data** (const QModelIndex &index, int role=Qt : :DisplayRole) const
– int **columnCount** (const QModelIndex &parent=QModelIndex()) const
– bool **setData** (const QModelIndex &index, const QVariant &value, int role)

4.75.1 Description détaillée

Définition à la ligne 7 du fichier TableModelParallelepipede.h.

4.75.2 Documentation des énumérations membres

4.75.2.1 enum TableModelParallelepipede : :Columns

Valeurs énumérées :

PermitRelative

PermeRelative

Conductivite

Coord1

Coord2

Définition à la ligne 12 du fichier TableModelParallelepipede.h.

4.75.3 Documentation des constructeurs et destructeur

4.75.3.1 TableModelParallelepipede : :TableModelParallelepipede (BlocElementBase ∗ *bloc,* QObject ∗ *parent* = 0)

Définition à la ligne 3 du fichier TableModelParallelepipede.cpp.

4.75.4 Documentation des fonctions membres

4.75.4.1 int TableModelParallelepipede : :columnCount (const QModelIndex & *parent* = QModelIndex()) const

Définition à la ligne 55 du fichier TableModelParallelepipede.cpp.

4.75.4.2 QVariant TableModelParallelepipede : :data (const QModelIndex & *index,* int *role* = `Qt : : DisplayRole`) const

Définition à la ligne 11 du fichier TableModelParallelepipede.cpp.

4.75.4.3 bool TableModelParallelepipede : :setData (const QModelIndex & *index,* const QVariant & *value,* int *role*)

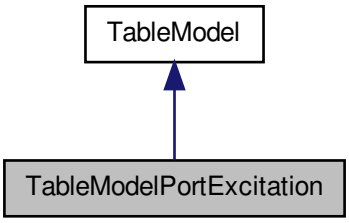
Définition à la ligne 60 du fichier TableModelParallelepipede.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelParallelepipede.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelParallelepipede.cpp**

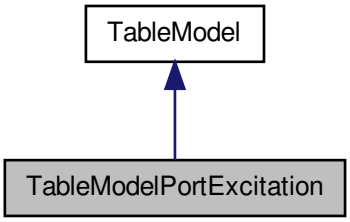
4.76 Référence de la classe TableModelPortExcitation

```
#include <TableModelPortExcitation.h>
```

Graphe d'héritage de TableModelPortExcitation :



Graphe de collaboration de TableModelPortExcitation :



Types publics

– enum **Columns** { **portActif** = 0, **portInfinitesimal** = 1, **type** = 2, **direction** = 3, **impedance** = 4, **typeSource** = 5, **ponderationAmplitude** = 6, **ponderationPhase** = 7, **AvantGauche** = 8, **ArriereDroit** = 9 }

Fonctions membres publiques

– **TableModelPortExcitation** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
– QVariant **data** (const QModelIndex &index, int role=Qt : :DisplayRole) const
– int **columnCount** (const QModelIndex &parent=QModelIndex()) const
– bool **setData** (const QModelIndex &index, const QVariant &value, int role)

4.76.1 Description détaillée

Définition à la ligne 7 du fichier TableModelPortExcitation.h.

4.76.2 Documentation des énumérations membres

4.76.2.1 enum TableModelPortExcitation : :Columns

Valeurs énumérées :

portActif

portInfinitesimal

type

direction

impedance

typeSource

ponderationAmplitude

ponderationPhase

AvantGauche

ArriereDroit

Définition à la ligne 11 du fichier TableModelPortExcitation.h.

4.76.3 Documentation des constructeurs et destructeur

4.76.3.1 TableModelPortExcitation : :TableModelPortExcitation (BlocElementBase * *bloc,* QObject * *parent* = `0`) `[explicit]`

Définition à la ligne 3 du fichier TableModelPortExcitation.cpp.

4.76.4 Documentation des fonctions membres

4.76.4.1 int TableModelPortExcitation : :columnCount (const QModelIndex & *parent* = `QModelIndex()`) const

Définition à la ligne 72 du fichier TableModelPortExcitation.cpp.

4.76.4.2 QVariant TableModelPortExcitation : :data (const QModelIndex & *index,* int *role* = `Qt::DisplayRole`) const

Définition à la ligne 11 du fichier TableModelPortExcitation.cpp.

4.76.4.3 bool TableModelPortExcitation : :setData (const QModelIndex & *index,* const QVariant & *value,* int *role*)

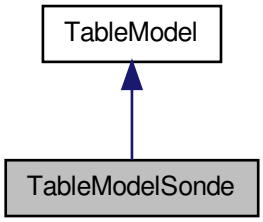
Définition à la ligne 77 du fichier TableModelPortExcitation.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelPortExcitation.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelPortExcitation.cpp**

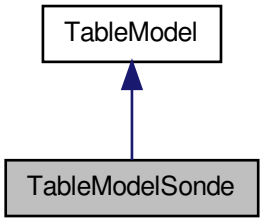
4.77 Référence de la classe TableModelSonde

```
#include <TableModelSonde.h>
```

Graphe d'héritage de TableModelSonde :



Graphe de collaboration de TableModelSonde :



Types publics

– enum **Columns** { **ex** = 0, **ey** = 1, **ez** = 2, **hx** = 3, **hy** = 4, **hz** = 5, **PointAppli** = 6, **ValCentre** = 7, **AvantGauche** = 8, **ArriereDroit** = 9 }

Fonctions membres publiques

– **TableModelSonde** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
– QVariant **data** (const QModelIndex &index, int role=Qt : :DisplayRole) const
– int **columnCount** (const QModelIndex &parent=QModelIndex()) const
– bool **setData** (const QModelIndex &index, const QVariant &value, int role)

4.77.1 Description détaillée

Définition à la ligne 7 du fichier TableModelSonde.h.

4.77.2 Documentation des énumérations membres

4.77.2.1 enum TableModelSonde : :Columns

Valeurs énumérées :

ex

ey

ez

hx

hy

hz

PointAppli

ValCentre

AvantGauche

ArriereDroit

Définition à la ligne 11 du fichier TableModelSonde.h.

4.77.3 Documentation des constructeurs et destructeur

4.77.3.1 TableModelSonde : :TableModelSonde (BlocElementBase ∗ *bloc,* QObject ∗ *parent =* `0`)

Définition à la ligne 3 du fichier TableModelSonde.cpp.

4.77.4 Documentation des fonctions membres

4.77.4.1 int TableModelSonde : :columnCount (const QModelIndex & *parent =* `QModelIndex()`) const

Définition à la ligne 74 du fichier TableModelSonde.cpp.

4.77.4.2 QVariant TableModelSonde : :data (const QModelIndex & *index,* int *role =* `Qt::DisplayRole`) const

Définition à la ligne 11 du fichier TableModelSonde.cpp.

4.77.4.3 bool TableModelSonde : :setData (const QModelIndex & *index,* const QVariant & *value,* int *role*)

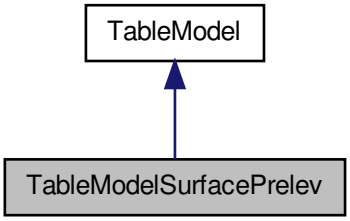
Définition à la ligne 79 du fichier TableModelSonde.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSonde.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSonde.cpp**

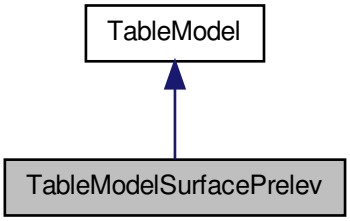
4.78 Référence de la classe TableModelSurfacePrelev

```
#include <TableModelSurfacePrelev.h>
```

Graphe d'héritage de TableModelSurfacePrelev :



Graphe de collaboration de TableModelSurfacePrelev :



Types publics

– enum **Columns** { **TypeSurface** = 0, **nomFichier** = 1, **insideOutside** = 2, **nbFace** = 3, **compressionLongueur** = 4, **compressionLargeur** = 5, **compressionHauteur** = 6, **AvantGauche** = 7, **ArriereDroit** = 8 }

Fonctions membres publiques

– **TableModelSurfacePrelev** (**BlocElementBase** ∗bloc, QObject ∗parent=0)
– QVariant **data** (const QModelIndex &index, int role=Qt : :DisplayRole) const
– int **columnCount** (const QModelIndex &parent=QModelIndex()) const
– Qt : :ItemFlags **flags** (const QModelIndex &index) const
– bool **setData** (const QModelIndex &index, const QVariant &value, int role)

4.78.1 Description détaillée

Définition à la ligne 7 du fichier TableModelSurfacePrelev.h.

4.78.2 Documentation des énumérations membres

4.78.2.1 enum TableModelSurfacePrelev : :Columns

Valeurs énumérées :

TypeSurface

nomFichier

insideOutside

nbFace

compressionLongueur

compressionLargeur

compressionHauteur

AvantGauche

ArriereDroit

Définition à la ligne 11 du fichier TableModelSurfacePrelev.h.

4.78.3 Documentation des constructeurs et destructeur

4.78.3.1 TableModelSurfacePrelev : :TableModelSurfacePrelev (BlocElementBase ∗ *bloc,* QObject ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableModelSurfacePrelev.cpp.

4.78.4 Documentation des fonctions membres

4.78.4.1 int TableModelSurfacePrelev : :columnCount (const QModelIndex & *parent* = `QModelIndex()`) const

Définition à la ligne 90 du fichier TableModelSurfacePrelev.cpp.

4.78.4.2 QVariant TableModelSurfacePrelev : :data (const QModelIndex & *index,* int *role* = `Qt::DisplayRole`) const

Définition à la ligne 32 du fichier TableModelSurfacePrelev.cpp.

4.78.4.3 Qt : :ItemFlags TableModelSurfacePrelev : :flags (const QModelIndex & *index*) const

Réimplémentée à partir de **TableModel** (p. 160).

Définition à la ligne 11 du fichier TableModelSurfacePrelev.cpp.

4.78.4.4 bool TableModelSurfacePrelev : :setData (const QModelIndex & *index,* const QVariant & *value,* int *role*)

Définition à la ligne 95 du fichier TableModelSurfacePrelev.cpp.

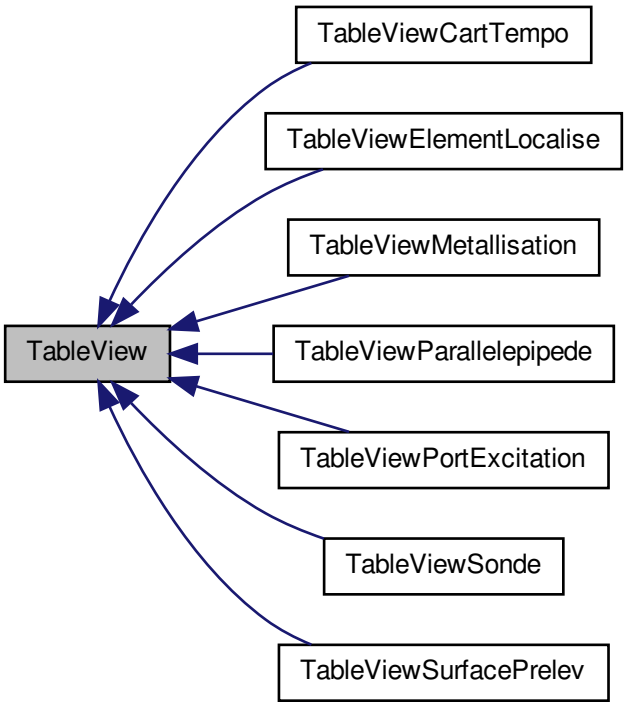
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSurfacePrelev.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableModelSurfacePrelev.cpp**

4.79 Référence de la classe TableView

Cette classe sert de vue de base pour les tableaux de données. Elle affiche les données contenu dans son model sous forme de tableau.

```
#include <TableView.h>
```

Graphe d'héritage de TableView :



Connecteurs publics

- void **addElement** (**elementBase** ∗e)
  Ajoute un élément au model.
- virtual void **addNewElement** ()
  Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.
- void **deleteElement** ()
  Supprime un élément du model.
- void **editColor** ()
  Ouvre une fenetre de dialogue pour changer de couleur les éléments séléctionnés. La couleur choisi sera appliquée à tous les éléments séléctionnés.

Fonctions membres publiques

- **TableView** (QWidget ∗parent=0)
- void **contextMenuEvent** (QContextMenuEvent ∗e)
  affiche un menu contextuel

- void **keyPressEvent** (QKeyEvent ∗event)
  Gère les entrées au clavier. Cette fonction appelle simplement la fonction de suppression des éléments quand la touche SUPPR est enfoncée.
- QModelIndexList **selectedIndexes** () const
  Modifie la liste des éléments séléctionnés pour prendre en compte seulement les lignes.
- virtual void **selectionChanged** (const QItemSelection &selected, const QItemSelection &deselected)
  Change la couleur des éléments séléctionnés.
- **TableModel** ∗ **getSourceModel** ()
  Retourne le model associé.
- void **setSourceModel** (**TableModel** ∗m)

4.79.1 Description détaillée

Cette classe sert de vue de base pour les tableaux de données. Elle affiche les données contenu dans son model sous forme de tableau.

Définition à la ligne 18 du fichier TableView.h.

4.79.2 Documentation des constructeurs et destructeur

4.79.2.1 TableView : :TableView (QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableView.cpp.

4.79.3 Documentation des fonctions membres

4.79.3.1 void TableView : :addElement (elementBase ∗ *e*) [slot]

Ajoute un élément au model.

Paramètres

e	l'élément a ajouté

Définition à la ligne 86 du fichier TableView.cpp.

4.79.3.2 void TableView : :addNewElement ( ) [virtual, slot]

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Réimplémentée dans **TableViewMetallisation** (p. 186), **TableViewElementLocalise** (p. 184), **TableViewParallelepipede** (p. 187), **TableViewPortExcitation** (p. 189), **TableViewCartTempo** (p. 182), **TableViewSurfacePrelev** (p. 192), et **TableViewSonde** (p. 191).

Définition à la ligne 91 du fichier TableView.cpp.

4.79.3.3 void TableView : :contextMenuEvent (QContextMenuEvent * *e*)

affiche un menu contextuel

Définition à la ligne 13 du fichier TableView.cpp.

4.79.3.4 void TableView : :deleteElement ( ) [slot]

Supprime un élément du model.

TableView : :deleteMetallisation Supprime les éléments séléctionnés.

Définition à la ligne 100 du fichier TableView.cpp.

4.79.3.5 void TableView : :editColor ( ) [slot]

Ouvre une fenetre de dialogue pour changer de couleur les éléments séléctionnés. La couleur choisi sera appliquée à tous les éléments séléctionnés.

Définition à la ligne 110 du fichier TableView.cpp.

4.79.3.6 TableModel * TableView : :getSourceModel ( )

Retourne le model associé.

Renvoie

le model **TableModel** (p. 158) associé à la vue.

Définition à la ligne 81 du fichier TableView.cpp.

4.79.3.7 void TableView : :keyPressEvent (QKeyEvent * *event*)

Gère les entrées au clavier. Cette fonction appelle simplement la fonction de suppression des éléments quand la touche SUPPR est enfoncée.

Définition à la ligne 23 du fichier TableView.cpp.

4.79.3.8 QModelIndexList TableView : :selectedIndexes ( ) const

Modifie la liste des éléments séléctionnés pour prendre en compte seulement les lignes.

TableView : :selectedIndexes (p. 180) Retourne la liste des lignes séléctionnés et trié sur le numéro de ligne.

Renvoie

la liste des éléments séléctionnés.

Définition à la ligne 36 du fichier TableView.cpp.

4.79.3.9 void TableView : :selectionChanged (const QItemSelection & *selected,* const QItemSelection & *deselected*) [virtual]

Change la couleur des éléments séléctionnés.

Définition à la ligne 57 du fichier TableView.cpp.

4.79.3.10 void TableView : :setSourceModel (TableModel * *m*) [inline]

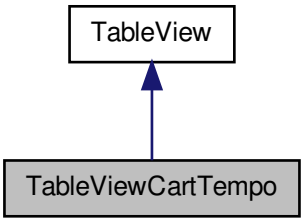
Définition à la ligne 51 du fichier TableView.h.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableView.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableView.cpp**

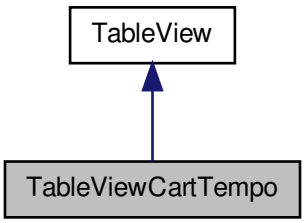
4.80 Référence de la classe TableViewCartTempo

```
#include <TableViewCartTempo.h>
```

Graphe d'héritage de TableViewCartTempo :



Graphe de collaboration de TableViewCartTempo :



Connecteurs publics

– void **addNewElement** ()
Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Fonctions membres publiques

– **TableViewCartTempo** (**blocCartographieTemporelle** ∗bloc, QWidget ∗parent=0)

4.80.1 Description détaillée

Définition à la ligne 10 du fichier TableViewCartTempo.h.

4.80.2 Documentation des constructeurs et destructeur

4.80.2.1 TableViewCartTempo : :TableViewCartTempo (blocCartographieTemporelle ∗ *bloc,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableViewCartTempo.cpp.

4.80.3 Documentation des fonctions membres

4.80.3.1 void TableViewCartTempo : :addNewElement ( ) [virtual, slot]

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Réimplémentée à partir de **TableView** (p. 179).

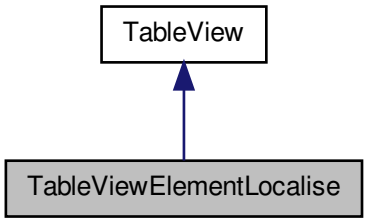
Définition à la ligne 14 du fichier TableViewCartTempo.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewCartTempo.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewCartTempo.cpp**

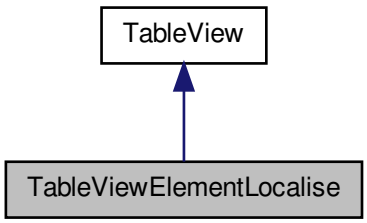
4.81 Référence de la classe TableViewElementLocalise

```
#include <TableViewElementLocalise.h>
```

Graphe d'héritage de TableViewElementLocalise :



Graphe de collaboration de TableViewElementLocalise :



Connecteurs publics

– void **addNewElement** ()

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Fonctions membres publiques

– **TableViewElementLocalise** (**blocElementLocalise** ∗bloc, QWidget ∗parent=0)

4.81.1 Description détaillée

Définition à la ligne 11 du fichier TableViewElementLocalise.h.

4.81.2 Documentation des constructeurs et destructeur

4.81.2.1 TableViewElementLocalise : :TableViewElementLocalise (blocElementLocalise ∗ *bloc,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier TableViewElementLocalise.cpp.

4.81.3 Documentation des fonctions membres

4.81.3.1 void TableViewElementLocalise : :addNewElement ( ) [virtual, slot]

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Réimplémentée à partir de **TableView** (p. 179).

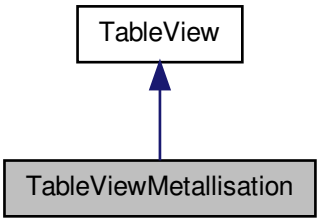
Définition à la ligne 16 du fichier TableViewElementLocalise.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewElementLocalise.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewElementLocalise.cpp**

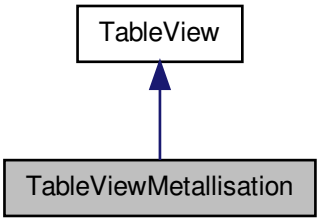
4.82 Référence de la classe TableViewMetallisation

```
#include <TableViewMetallisation.h>
```

Graphe d'héritage de TableViewMetallisation :



Graphe de collaboration de TableViewMetallisation :



Connecteurs publics

– void **addNewElement** ()
  TableViewMetallisation : :addNewMetallisation Créer un nouvel objet metallisation et ouvre la fenetre d'edition de l'objet.

Fonctions membres publiques

– **TableViewMetallisation** (**blocMetallisation** ∗blocMetal, QWidget ∗parent=0)

4.82.1 Description détaillée

Définition à la ligne 10 du fichier TableViewMetallisation.h.

4.82.2 Documentation des constructeurs et destructeur

4.82.2.1 TableViewMetallisation : :TableViewMetallisation (blocMetallisation * *blocMetal,* QWidget * *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableViewMetallisation.cpp.

4.82.3 Documentation des fonctions membres

4.82.3.1 void TableViewMetallisation : :addNewElement ( ) [virtual, slot]

TableViewMetallisation : :addNewMetallisation Créer un nouvel objet metallisation et ouvre la fenetre d'edition de l'objet.

Réimplémentée à partir de **TableView** (p. 179).

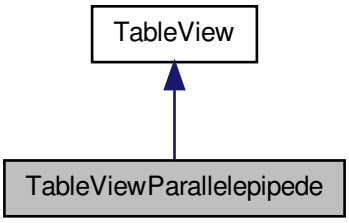
Définition à la ligne 18 du fichier TableViewMetallisation.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewMetallisation.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewMetallisation.cpp**

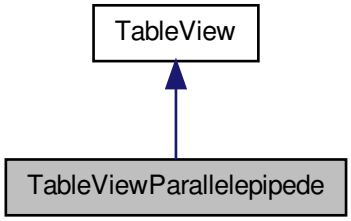
4.83 Référence de la classe TableViewParallelepipede

```
#include <TableViewParallelepipede.h>
```

Graphe d'héritage de TableViewParallelepipede :



Graphe de collaboration de TableViewParallelepipede :



Connecteurs publics

– void **addNewElement** ()
Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Fonctions membres publiques

– **TableViewParallelepipede** (**blocParallelepipede** ∗blocParal, QWidget ∗parent=0)

4.83.1 Description détaillée

Définition à la ligne 11 du fichier TableViewParallelepipede.h.

4.83.2 Documentation des constructeurs et destructeur

4.83.2.1 TableViewParallelepipede : :TableViewParallelepipede (blocParallelepipede ∗ *blocParal,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableViewParallelepipede.cpp.

4.83.3 Documentation des fonctions membres

4.83.3.1 void TableViewParallelepipede : :addNewElement ( ) [virtual, slot]

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Réimplémentée à partir de **TableView** (p. 179).

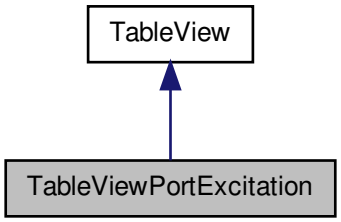
Définition à la ligne 13 du fichier TableViewParallelepipede.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewParallelepipede.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewParallelepipede.cpp**

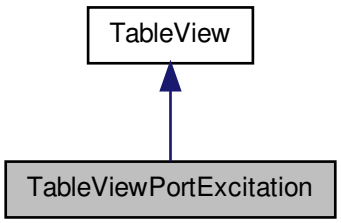
4.84 Référence de la classe TableViewPortExcitation

```
#include <TableViewPortExcitation.h>
```

Graphe d'héritage de TableViewPortExcitation :



Graphe de collaboration de TableViewPortExcitation :



Connecteurs publics

– void **addNewElement** ()
  Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Fonctions membres publiques

– **TableViewPortExcitation** (**blocPortExcitation** ∗bloc, QWidget ∗parent=0)

4.84.1 Description détaillée

Définition à la ligne 11 du fichier TableViewPortExcitation.h.

4.84.2 Documentation des constructeurs et destructeur

4.84.2.1 TableViewPortExcitation : :TableViewPortExcitation (blocPortExcitation ∗ *bloc,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableViewPortExcitation.cpp.

4.84.3 Documentation des fonctions membres

4.84.3.1 void TableViewPortExcitation : :addNewElement ( ) [virtual, slot]

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Réimplémentée à partir de **TableView** (p. 179).

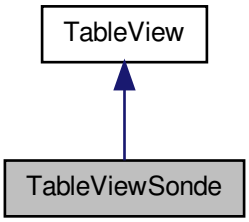
Définition à la ligne 17 du fichier TableViewPortExcitation.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewPortExcitation.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewPortExcitation.cpp**

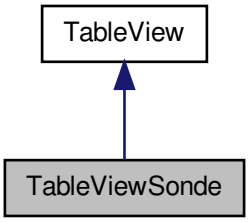
4.85 Référence de la classe TableViewSonde

```
#include <TableViewSonde.h>
```

Graphe d'héritage de TableViewSonde :



Graphe de collaboration de TableViewSonde :



Connecteurs publics

– void **addNewElement** ()
  Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Fonctions membres publiques

– **TableViewSonde** (**blocSonde** ∗bloc, QWidget ∗parent=0)

4.85.1 Description détaillée

Définition à la ligne 10 du fichier TableViewSonde.h.

4.85.2 Documentation des constructeurs et destructeur

4.85.2.1 TableViewSonde : :TableViewSonde (blocSonde * *bloc,* QWidget * *parent* = 0) [explicit]

Définition à la ligne 3 du fichier TableViewSonde.cpp.

4.85.3 Documentation des fonctions membres

4.85.3.1 void TableViewSonde : :addNewElement ( ) [virtual, slot]

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Réimplémentée à partir de **TableView** (p. 179).

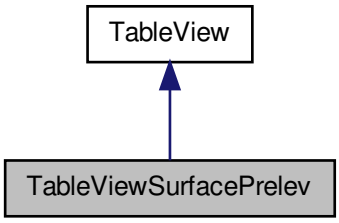
Définition à la ligne 14 du fichier TableViewSonde.cpp.

La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSonde.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSonde.cpp**

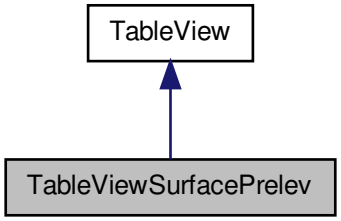
4.86 Référence de la classe TableViewSurfacePrelev

```
#include <TableViewSurfacePrelev.h>
```

Graphe d'héritage de TableViewSurfacePrelev :



Graphe de collaboration de TableViewSurfacePrelev :



Connecteurs publics

– void **addNewElement** ()
Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Fonctions membres publiques

– **TableViewSurfacePrelev** (**blocSurfacePrelevement** ∗bloc, QWidget ∗parent=0)

4.86.1 Description détaillée

Définition à la ligne 12 du fichier TableViewSurfacePrelev.h.

4.86.2 Documentation des constructeurs et destructeur

4.86.2.1 TableViewSurfacePrelev : :TableViewSurfacePrelev (blocSurfacePrelevement ∗ *bloc,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 3 du fichier TableViewSurfacePrelev.cpp.

4.86.3 Documentation des fonctions membres

4.86.3.1 void TableViewSurfacePrelev : :addNewElement ( ) [virtual, slot]

Fonction virtual qui est appelé quand l'utilisateur veut ajouter un nouveau élément avec le menu contextuel du tableau. Par défaut, cette fonction ne fait rien.

Réimplémentée à partir de **TableView** (p. 179).

Définition à la ligne 17 du fichier TableViewSurfacePrelev.cpp.

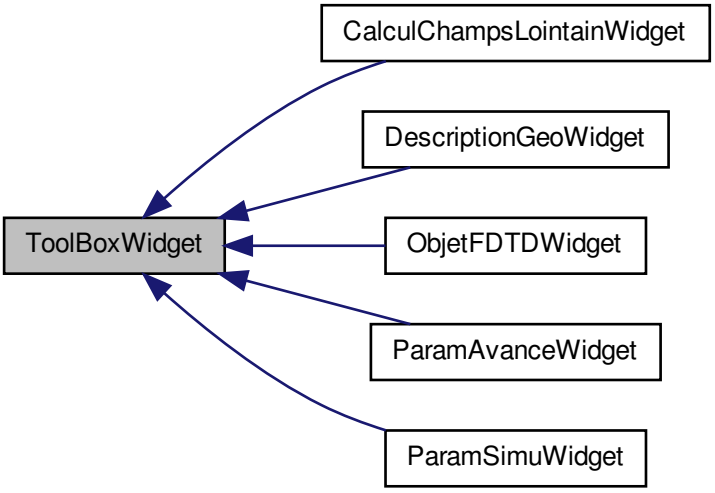
La documentation de cette classe a été générée à partir des fichiers suivants :
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSurfacePrelev.h**
– /home/kuiderlix/Prog/cpp/pskl-etprat/**TableViewSurfacePrelev.cpp**

4.87 Référence de la classe ToolBoxWidget

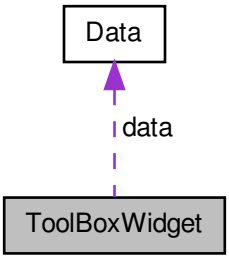
Classe de base pour les widgets conteneurs de la ToolBox.

```
#include <ToolBoxWidget.h>
```

Graphe d'héritage de ToolBoxWidget :



Graphe de collaboration de ToolBoxWidget :



Fonctions membres publiques

- **ToolBoxWidget** (**Data** ∗**data**, QWidget ∗parent=0)
- void **setData** (**Data** ∗**data**)
- virtual void **init** ()=0

Attributs protégés

- **Data** ∗ **data**

4.87.1 Description détaillée

Classe de base pour les widgets conteneurs de la ToolBox.

Définition à la ligne 10 du fichier ToolBoxWidget.h.

4.87.2 Documentation des constructeurs et destructeur

4.87.2.1 ToolBoxWidget : :ToolBoxWidget (Data ∗ *data,* QWidget ∗ *parent* = 0) [explicit]

Définition à la ligne 3 du fichier ToolBoxWidget.cpp.

4.87.3 Documentation des fonctions membres

4.87.3.1 virtual void ToolBoxWidget : :init ( ) [pure virtual]

Implémenté dans **DescriptionGeoWidget** (p. 71), **CalculChampsLointainWidget** (p. 46), **ObjetFDTDWidget** (p. 130), **ParamAvanceWidget** (p. 134), et **ParamSimuWidget** (p. 139).

4.87.3.2 void ToolBoxWidget : :setData (Data * *data*) [inline]

Définition à la ligne 16 du fichier ToolBoxWidget.h.

4.87.4 Documentation des données membres

4.87.4.1 Data* ToolBoxWidget : :data [protected]

Définition à la ligne 29 du fichier ToolBoxWidget.h.

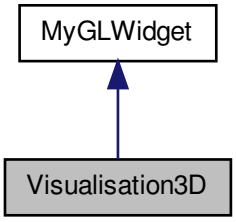
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ToolBoxWidget.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**ToolBoxWidget.cpp**

4.88 Référence de la classe Visualisation3D

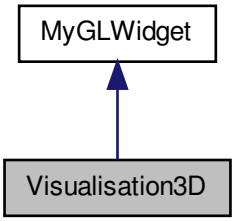
Affiche et gère la scène 3D.

```
#include <visualisation3d.h>
```

Graphe d'héritage de Visualisation3D :



Graphe de collaboration de Visualisation3D :



Connecteurs publics

- void **setZoom** (int z)
  Modifie le zoom sur la scène.
- void **relnit** ()
  Cette fonction fait tendre progressivement (à chaque appel) les variables qui gèrent la scène dans le état initial.
- void **startToZoom11** ()
  Cette fonction démarre le timer qui permet de remettre la scène dans son état initial.

Signaux

- void **zoomChanged** (int)
  Signal émis quand le zoom change.
- void **stopTimer** ()
  Signal émis quand le timer doit s'arreter.

Fonctions membres publiques

- **Visualisation3D** (**volumeCalcul** ∗volume, QWidget ∗parent=0)
- void **initializeGL** ()
- void **resizeGL** (int width, int height)
- void **paintGL** ()
- void **keyPressEvent** (QKeyEvent ∗keyEvent)
- void **mousePressEvent** (QMouseEvent ∗event)
- void **mouseMoveEvent** (QMouseEvent ∗event)
- void **mouseReleaseEvent** (QMouseEvent ∗event)
- void **wheelEvent** (QWheelEvent ∗event)
- void **dessineVolumeCalcul** ()
  Desinne dans la scène le volume de calcul.
- void **dessineScene** ()
  Dessine dans la scène tout les éléments du scénario.
- void **dessineCageExcitation** ()
  Dessine la cage d'excitation du scénario de simulation s'elle existe. La cage s'affiche par défault en transparence.

- **Point coordonneeToPoint** (**coordonnee**)
  *Tranforme une coordonnee en un **Point** (p. 146).*
- void **ajoutListElement** (**BlocElementBase** ∗)
  Ajoute une liste d'élément à la scène.
- void **setXRotation** (int angle)
- void **setYRotation** (int angle)
- void **normalizeAngle** (int ∗angle)
- void **toZero** (int ∗var, int step)
  Fait tendre la variable var vers 0 avec un pas de step à chaque appel. Exemple : si var=50 et step = 1 La fonction va décrémenté de 1 la variable à chaque appel.
- void **toZeroAngle** (int ∗angle, int step)
  *Meme fonction que **toZero()** (p. 200) mais appliqué à un angle.*
- void **init** ()
  Initialise les variables de la scène (zoom, xRot, yMov, ...)
- void **setVolumeCalcul** (**volumeCalcul** ∗vol)
- void **setCageExcitation** (**cageExcitation** ∗c)
- void **clearScene** ()
  Vide la scène.

4.88.1 Description détaillée

Affiche et gère la scène 3D.

Définition à la ligne 21 du fichier visualisation3d.h.

4.88.2 Documentation des constructeurs et destructeur

4.88.2.1 Visualisation3D : :Visualisation3D (volumeCalcul ∗ *volume,* QWidget ∗ *parent =* 0) [explicit]

Définition à la ligne 4 du fichier visualisation3d.cpp.

4.88.3 Documentation des fonctions membres

4.88.3.1 void Visualisation3D : :ajoutListElement (BlocElementBase ∗ *l*)

Ajoute une liste d'élément à la scène.

Définition à la ligne 207 du fichier visualisation3d.cpp.

4.88.3.2 void Visualisation3D : :clearScene ( ) [inline]

Vide la scène.

Définition à la ligne 104 du fichier visualisation3d.h.

4.88.3.3 Point Visualisation3D : :coordonneeToPoint (coordonnee *c*)

Tranforme une coordonnee en un **Point** (p. 146).

Paramètres

c	la coordonnee a transformée

Renvoie

le **Point** (p. 146) transformé

Définition à la ligne 192 du fichier visualisation3d.cpp.

4.88.3.4 void Visualisation3D : :dessineCageExcitation ( )

Dessine la cage d'excitation du scénario de simulation s'elle existe. La cage s'affiche par défault en transparence.

Définition à la ligne 182 du fichier visualisation3d.cpp.

4.88.3.5 void Visualisation3D : :dessineScene ( )

Dessine dans la scène tout les éléments du scénario.

Définition à la ligne 166 du fichier visualisation3d.cpp.

4.88.3.6 void Visualisation3D : :dessineVolumeCalcul ( )

Desinne dans la scène le volume de calcul.

Définition à la ligne 160 du fichier visualisation3d.cpp.

4.88.3.7 void Visualisation3D : :init ( )

Initialise les variables de la scène (zoom, xRot, yMov, ...)

Définition à la ligne 217 du fichier visualisation3d.cpp.

4.88.3.8 void Visualisation3D : :initializeGL ( ) [virtual]

Implémente **MyGLWidget** (p. 126).

Définition à la ligne 22 du fichier visualisation3d.cpp.

4.88.3.9 void Visualisation3D : :keyPressEvent (QKeyEvent * *keyEvent*)

Définition à la ligne 75 du fichier visualisation3d.cpp.

4.88.3.10 void Visualisation3D : :mouseMoveEvent (QMouseEvent * *event*)

Définition à la ligne 105 du fichier visualisation3d.cpp.

4.88.3.11 void Visualisation3D : :mousePressEvent (QMouseEvent ∗ *event*)

Définition à la ligne 99 du fichier visualisation3d.cpp.

4.88.3.12 void Visualisation3D : :mouseReleaseEvent (QMouseEvent ∗ *event*)

Définition à la ligne 122 du fichier visualisation3d.cpp.

4.88.3.13 void Visualisation3D : :normalizeAngle (int ∗ *angle*)

Définition à la ligne 249 du fichier visualisation3d.cpp.

4.88.3.14 void Visualisation3D : :paintGL ( ) [virtual]

Implémente **MyGLWidget** (p. 126).

Définition à la ligne 47 du fichier visualisation3d.cpp.

4.88.3.15 void Visualisation3D : :reInit ( ) [slot]

Cette fonction fait tendre progressivement (à chaque appel) les variables qui gèrent la scène dans le état initial.

Définition à la ligne 227 du fichier visualisation3d.cpp.

4.88.3.16 void Visualisation3D : :resizeGL (int *width,* int *height*) [virtual]

Implémente **MyGLWidget** (p. 126).

Définition à la ligne 34 du fichier visualisation3d.cpp.

4.88.3.17 void Visualisation3D : :setCageExcitation (cageExcitation ∗ *c*) [inline]

Définition à la ligne 96 du fichier visualisation3d.h.

4.88.3.18 void Visualisation3D : :setVolumeCalcul (volumeCalcul ∗ *vol*) [inline]

Définition à la ligne 91 du fichier visualisation3d.h.

4.88.3.19 void Visualisation3D : :setXRotation (int *angle*)

Définition à la ligne 141 du fichier visualisation3d.cpp.

4.88.3.20 void Visualisation3D : :setYRotation (int *angle*)

Définition à la ligne 150 du fichier visualisation3d.cpp.

4.88.3.21 void Visualisation3D : :setZoom (int *z*) [slot]

Modifie le zoom sur la scène.

Paramètres

z	

Définition à la ligne 212 du fichier visualisation3d.cpp.

4.88.3.22 void Visualisation3D : :startToZoom11 ( ) [slot]

Cette fonction démarre le timer qui permet de remettre la scène dans son état initial.

Définition à la ligne 244 du fichier visualisation3d.cpp.

4.88.3.23 void Visualisation3D : :stopTimer ( ) [signal]

Signal émis quand le timer doit s'arreter.

4.88.3.24 void Visualisation3D : :toZero (int * *var,* int *step*)

Fait tendre la variable var vers 0 avec un pas de step à chaque appel. Exemple : si var=50 et step = 1 La fonction va décrémenté de 1 la variable à chaque appel.

Paramètres

var	
step	

Définition à la ligne 257 du fichier visualisation3d.cpp.

4.88.3.25 void Visualisation3D : :toZeroAngle (int * *angle,* int *step*)

Meme fonction que **toZero()** (p. 200) mais appliqué à un angle.

Paramètres

angle	
step	

Définition à la ligne 273 du fichier visualisation3d.cpp.

4.88.3.26 void Visualisation3D : :wheelEvent (QWheelEvent * *event*)

Définition à la ligne 128 du fichier visualisation3d.cpp.

4.88.3.27 void Visualisation3D : :zoomChanged (int) [signal]

Signal émis quand le zoom change.

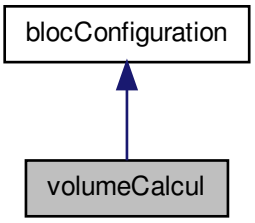
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**visualisation3d.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**visualisation3d.cpp**

4.89 Référence de la classe volumeCalcul

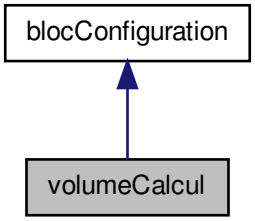
Cette classe représente la variable **volumeCalcul** (p. 201) d'une simulation.

```
#include <volumeCalcul.h>
```

Graphe d'héritage de volumeCalcul :



Graphe de collaboration de volumeCalcul :



Connecteurs publics

- void **SetHauteur** (double hauteur)
- void **SetLargeur** (double largeur)
- void **SetLongueur** (double longueur)
- void **SetNombreX** (int nombreX)
- void **SetNombreY** (int nombreY)
- void **SetNombreZ** (int nombreZ)
- void **ecrire** ()

Signaux

- void **hauteurChanged** ()
- void **largeurChanged** ()
- void **longueurChanged** ()
- void **nbXChanged** ()
- void **nbYChanged** ()
- void **nbZChanged** ()

Fonctions membres publiques

- **volumeCalcul** ()
- virtual ~**volumeCalcul** ()
- double **GetHauteur** () const
- double **GetLargeur** () const
- double **GetLongueur** () const
- int **GetNombreX** () const
- int **GetNombreY** () const
- int **GetNombreZ** () const

4.89.1 Description détaillée

Cette classe représente la variable **volumeCalcul** (p. 201) d'une simulation.

Définition à la ligne 17 du fichier volumeCalcul.h.

4.89.2 Documentation des constructeurs et destructeur

4.89.2.1 volumeCalcul : :volumeCalcul ( )

Définition à la ligne 12 du fichier volumeCalcul.cpp.

4.89.2.2 volumeCalcul : :~volumeCalcul ( ) [virtual]

Définition à la ligne 21 du fichier volumeCalcul.cpp.

4.89.3 Documentation des fonctions membres

4.89.3.1 void volumeCalcul : :ecrire ( ) [slot]

Réimplémentée à partir de **blocConfiguration** (p. 21).

Définition à la ligne 24 du fichier volumeCalcul.cpp.

4.89.3.2 double volumeCalcul : :GetHauteur ( ) const [inline]

Définition à la ligne 24 du fichier volumeCalcul.h.

4.89.3.3 double volumeCalcul : :GetLargeur ( ) const [inline]

Définition à la ligne 27 du fichier volumeCalcul.h.

4.89.3.4 double volumeCalcul : :GetLongueur ( ) const [inline]

Définition à la ligne 30 du fichier volumeCalcul.h.

4.89.3.5 int volumeCalcul : :GetNombreX ( ) const [inline]

Définition à la ligne 33 du fichier volumeCalcul.h.

4.89.3.6 int volumeCalcul : :GetNombreY ( ) const [inline]

Définition à la ligne 36 du fichier volumeCalcul.h.

4.89.3.7 int volumeCalcul : :GetNombreZ ( ) const [inline]

Définition à la ligne 39 du fichier volumeCalcul.h.

4.89.3.8 void volumeCalcul : :hauteurChanged ( ) [signal]

4.89.3.9 void volumeCalcul : :largeurChanged ( ) [signal]

4.89.3.10 void volumeCalcul : :longueurChanged ( ) [signal]

4.89.3.11 void volumeCalcul : :nbXChanged ( ) [signal]

4.89.3.12 void volumeCalcul : :nbYChanged ( ) [signal]

4.89.3.13 void volumeCalcul : :nbZChanged ( ) [signal]

4.89.3.14 void volumeCalcul : :SetHauteur (double *hauteur*) [inline, slot]

Définition à la ligne 55 du fichier volumeCalcul.h.

4.89.3.15 void volumeCalcul : :SetLargeur (double *largeur*) [inline, slot]

Définition à la ligne 58 du fichier volumeCalcul.h.

4.89.3.16 void volumeCalcul : :SetLongueur (double *longueur*) [inline, slot]

Définition à la ligne 61 du fichier volumeCalcul.h.

4.89.3.17 void volumeCalcul : :SetNombreX (int *nombreX*) [inline, slot]

Définition à la ligne 64 du fichier volumeCalcul.h.

4.89.3.18 void volumeCalcul : :SetNombreY (int *nombreY*) [inline, slot]

Définition à la ligne 67 du fichier volumeCalcul.h.

4.89.3.19 void volumeCalcul : :SetNombreZ (int *nombreZ*) [inline, slot]

Définition à la ligne 70 du fichier volumeCalcul.h.

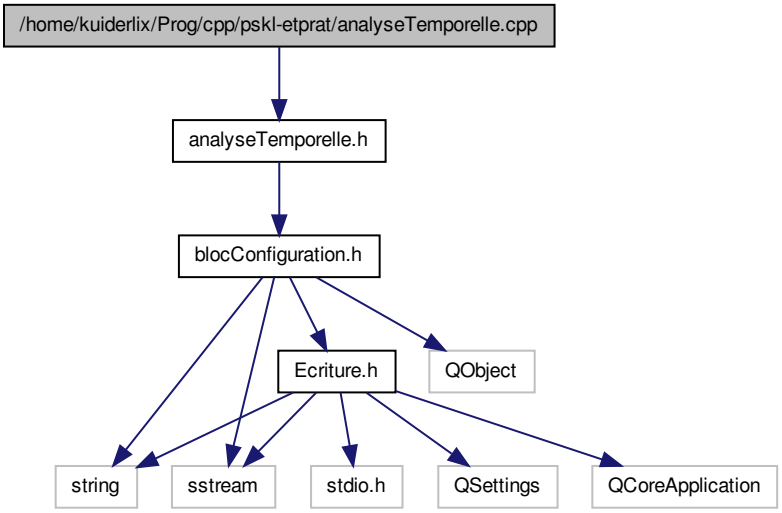
La documentation de cette classe a été générée à partir des fichiers suivants :
- /home/kuiderlix/Prog/cpp/pskl-etprat/**volumeCalcul.h**
- /home/kuiderlix/Prog/cpp/pskl-etprat/**volumeCalcul.cpp**

Chapitre 5

Documentation des fichiers

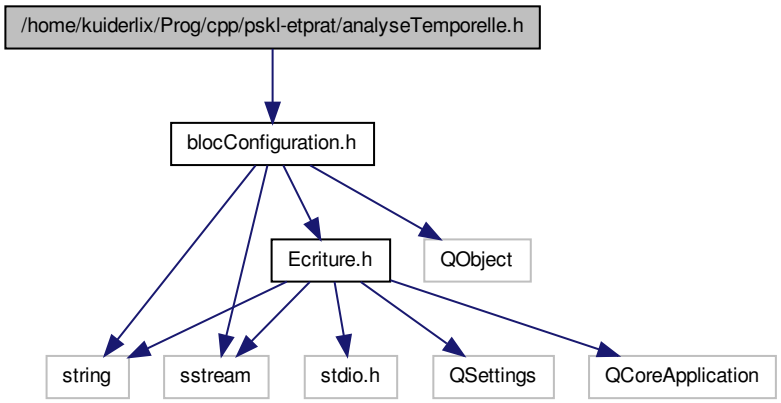
5.1 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/analyseTemporelle.cpp

`#include "analyseTemporelle.h"` Graphe des dépendances par inclusion de analyseTemporelle.cpp :

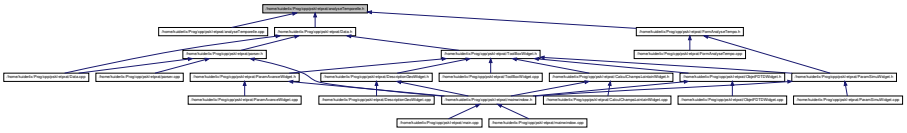


5.2 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/analyseTemporelle.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion de analyseTemporelle.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



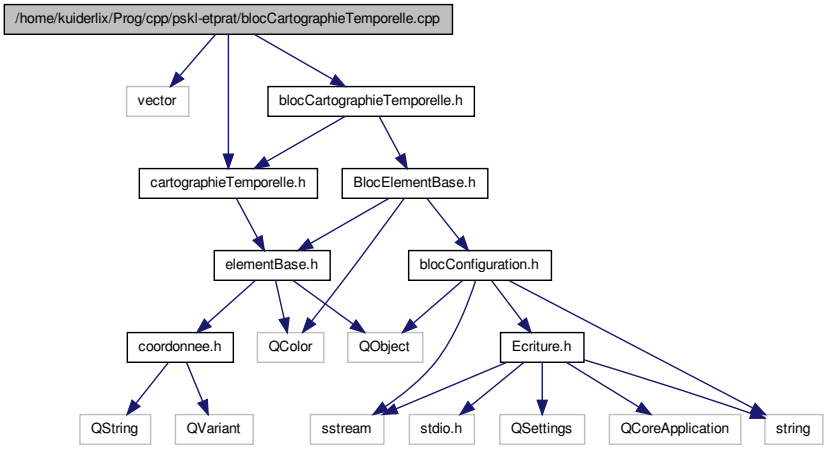
Classes

– class **analyseTemporelle**
*Cette classe représente la variable **analyseTemporelle** (p. 15) d'une simulation.*

5.3 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocCartographieTemporelle.cpp

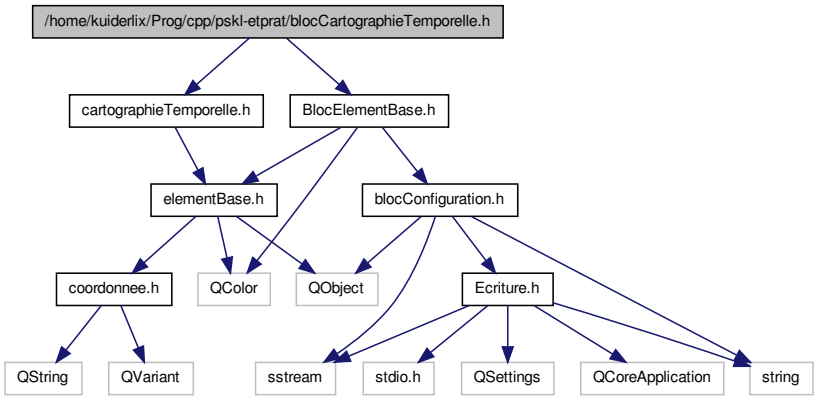
`#include <vector>` `#include "blocCartographieTemporelle.h"` `#include "cartographieTemporelle.h"` Graphe des dépendances par

inclusion de blocCartographieTemporelle.cpp :

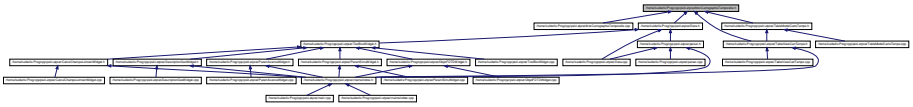


5.4 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocCartographieTemporelle.h

`#include "cartographieTemporelle.h"` `#include "BlocElementBase.h"` Graphe des dépendances par inclusion de blocCartographieTemporelle.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

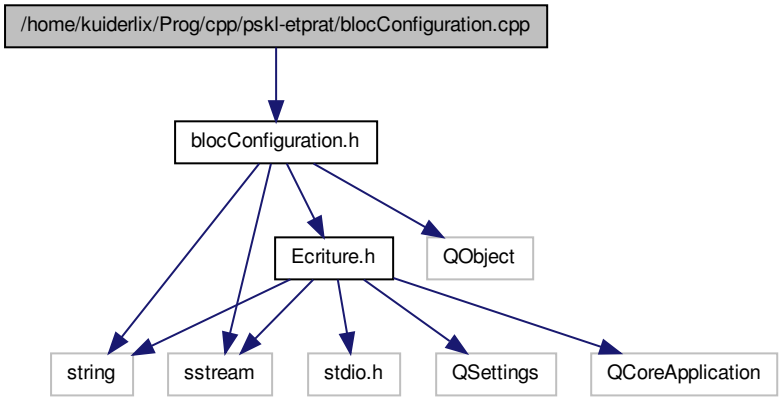


Classes

– class **blocCartographieTemporelle**
Cette classe représente une liste de ***cartographieTemporelle*** *(p. 49).*

5.5 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocConfiguration.cpp

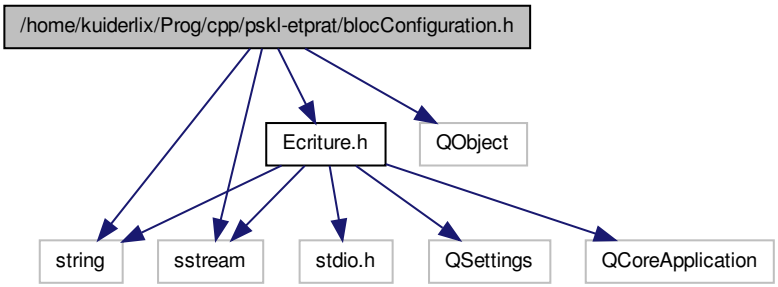
`#include "blocConfiguration.h"` Graphe des dépendances par inclusion de blocConfiguration.cpp :



5.6 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocConfiguration.h

`#include <string>` `#include <sstream>` `#include <QObject>` `#include "Ecriture.h"` Graphe des dépendances par inclusion de bloc-

Configuration.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

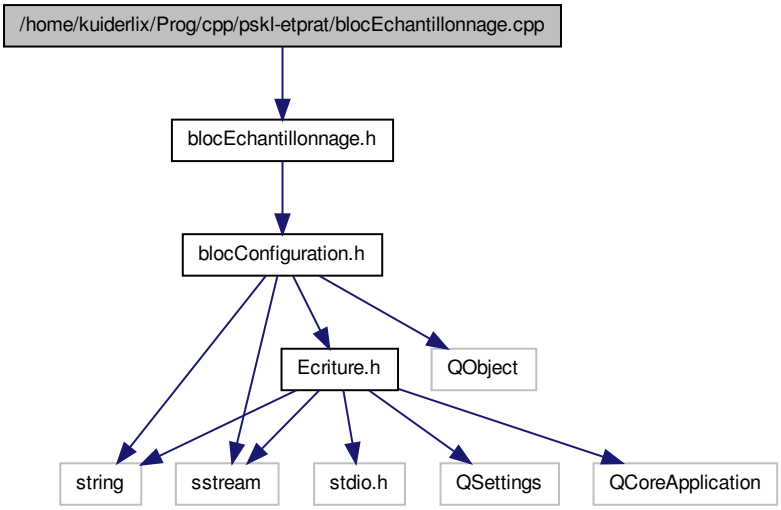
Classes

– class **blocConfiguration**
Classe de base pour les données de configuration d'une simulation.

5.7 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocEchantillonnage.cpp

`#include "blocEchantillonnage.h"` Graphe des dépendances par inclu-

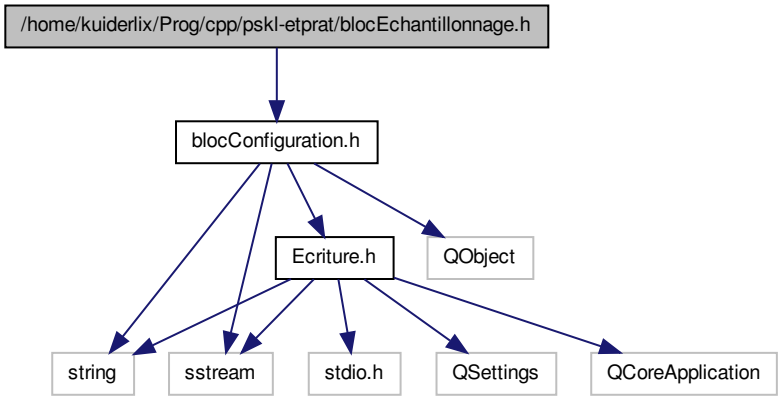
sion de blocEchantillonnage.cpp :



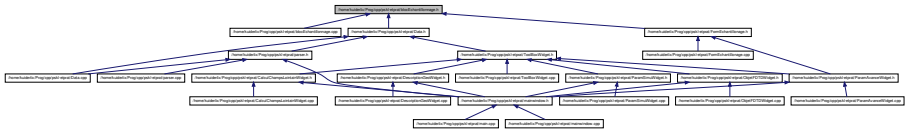
5.8 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocEchantillonnage.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion

de blocEchantillonnage.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



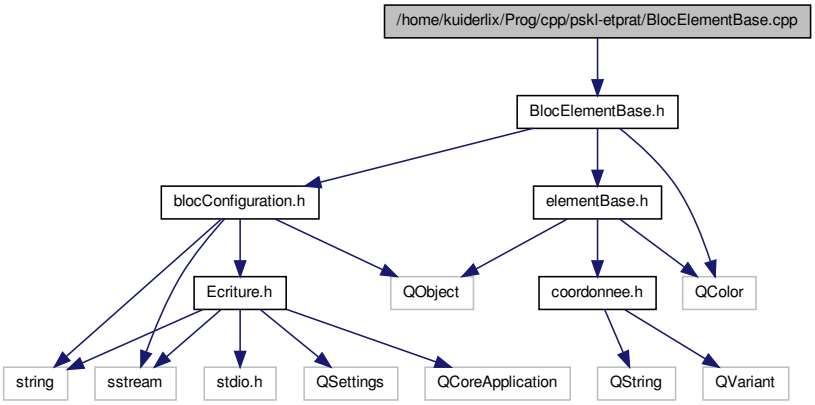
Classes

– class **blocEchantillonnage**
Cette classe représente la variable echantillonnage d'une simulation.

5.9 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/BlocElementBase.cpp

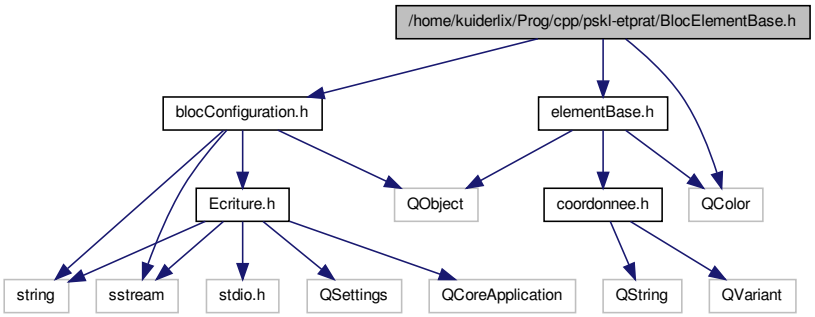
`#include "BlocElementBase.h"` Graphe des dépendances par inclusion de

BlocElementBase.cpp :



5.10 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-BlocElementBase.h

`#include "blocConfiguration.h"` `#include "elementBase.h"` × `#include <QColor>` Graphe des dépendances par inclusion de BlocElement-Base.h :



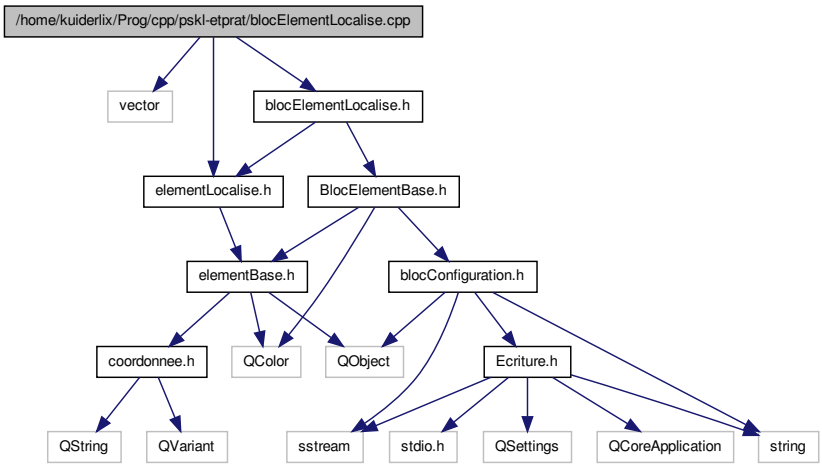
Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

– class **BlocElementBase**
*Représente une liste d'**elementBase*** (p. 73).

5.11 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocElementLocalise.cpp

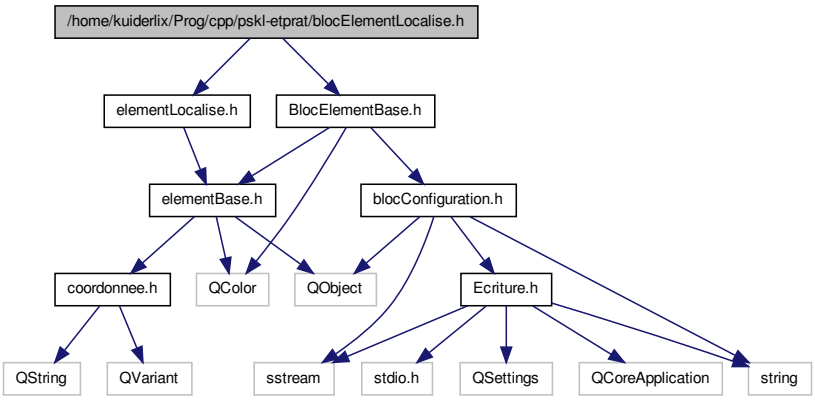
`#include <vector>` `#include "blocElementLocalise.h"` `#include "elementLocalise.h"` Graphe des dépendances par inclusion de blocElementLocalise.cpp :



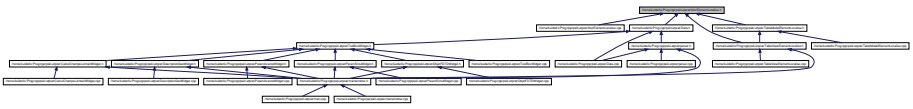
5.12 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocElementLocalise.h

`#include "elementLocalise.h"` `#include "BlocElementBase.-`

`h"` Graphe des dépendances par inclusion de blocElementLocalise.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



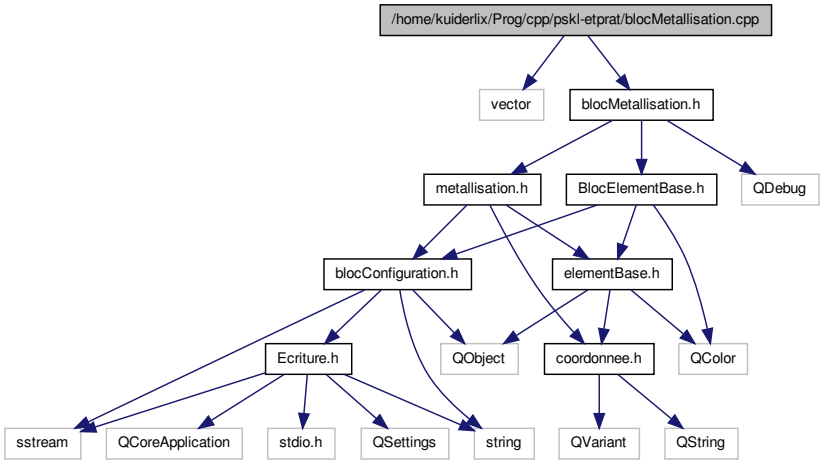
Classes

– class **blocElementLocalise**
Cette classe représente une liste de ***elementLocalise*** *(p. 77).*

5.13 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocMetallisation.cpp

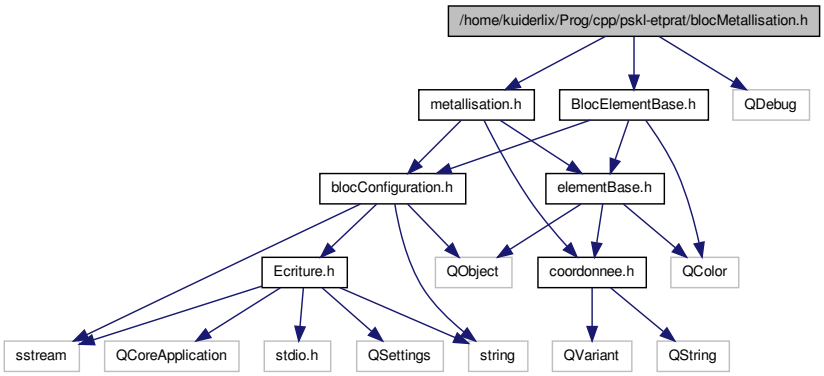
`#include <vector> #include "blocMetallisation.h"` Graphe des

dépendances par inclusion de blocMetallisation.cpp :

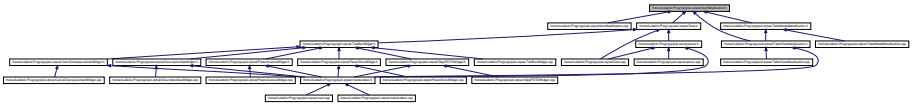


5.14 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocMetallisation.h

`#include "BlocElementBase.h"` `#include "metallisation.h"` `#include <QDebug>` Graphe des dépendances par inclusion de blocMetallisation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

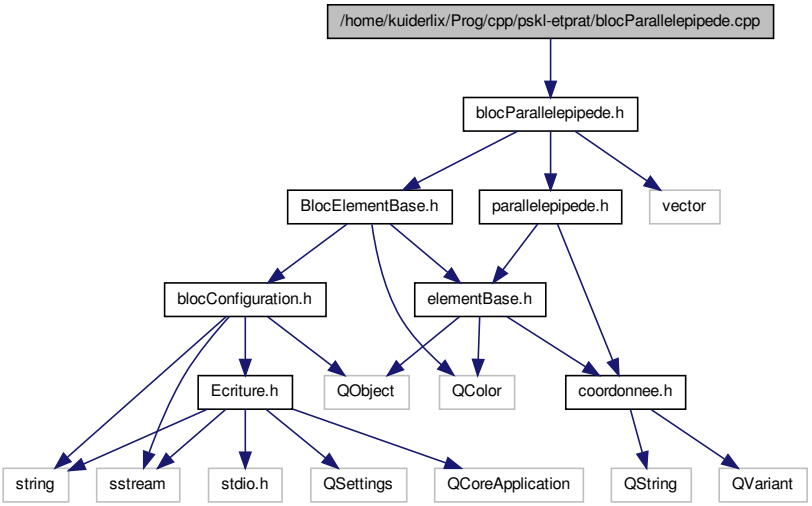


Classes

– class **blocMetallisation**
Cette classe représente une liste de metallisation.

5.15 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocParallelepipede.cpp

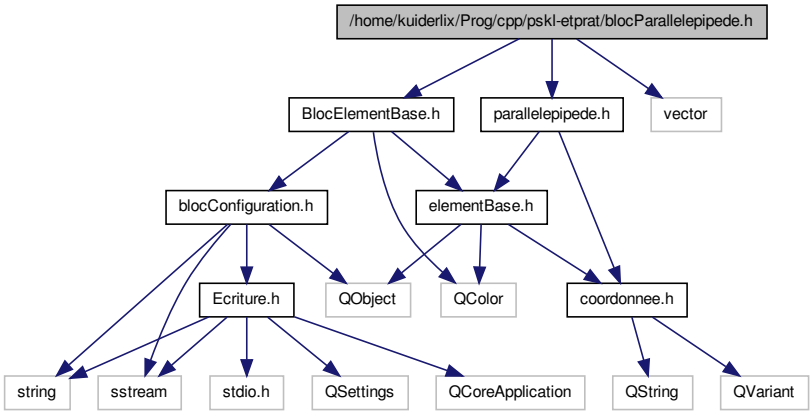
`#include "blocParallelepipede.h"` Graphe des dépendances par inclusion de blocParallelepipede.cpp :



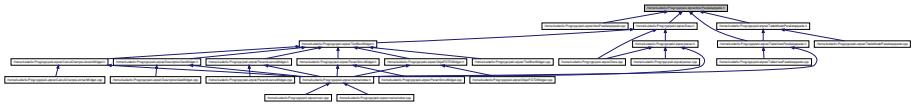
5.16 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocParallelepipede.h

`#include "BlocElementBase.h"` `#include "parallelepipede.-`

`h" #include <vector>` Graphe des dépendances par inclusion de bloc-Parallelepipede.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



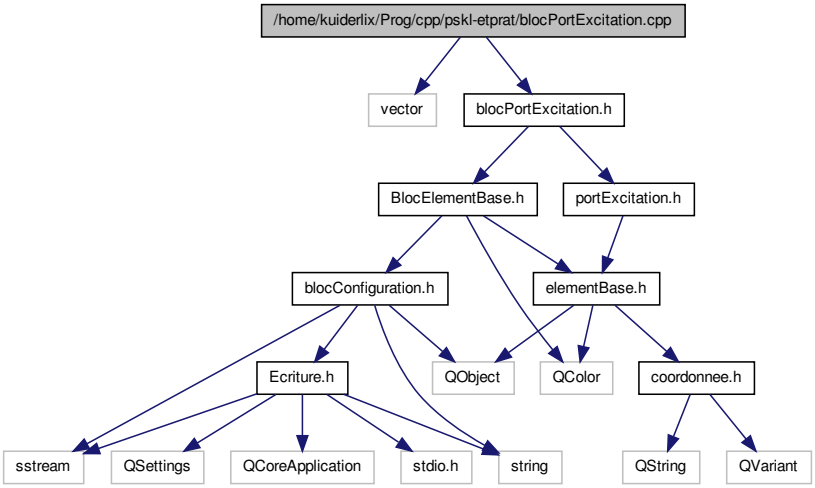
Classes

– class **blocParallelepipede**
Cette classe représente une liste de parallelepipede.

5.17 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/bloc-PortExcitation.cpp

`#include <vector>` `#include "blocPortExcitation.h"` Graphe des

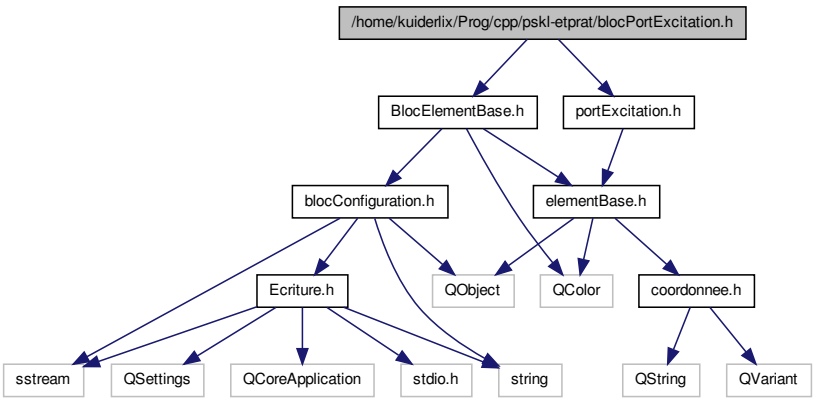
dépendances par inclusion de blocPortExcitation.cpp :



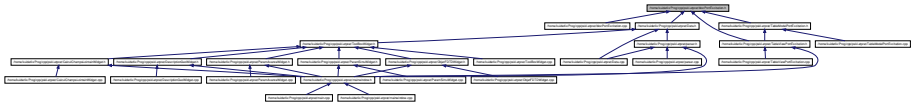
5.18 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocPortExcitation.h

`#include "BlocElementBase.h"` `#include "portExcitation.h"`

Graphe des dépendances par inclusion de blocPortExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

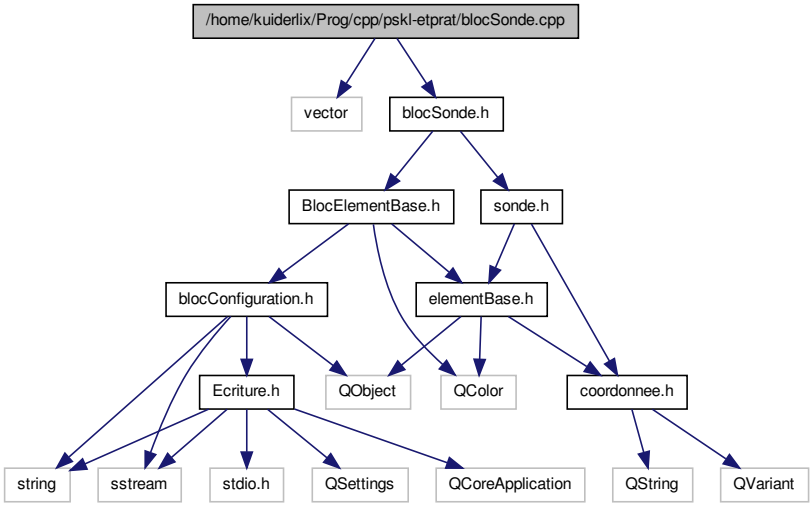


Classes

– class **blocPortExcitation**
  *Cette classe représente une liste de **portExcitation*** (p. *147*).

5.19 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSonde.cpp

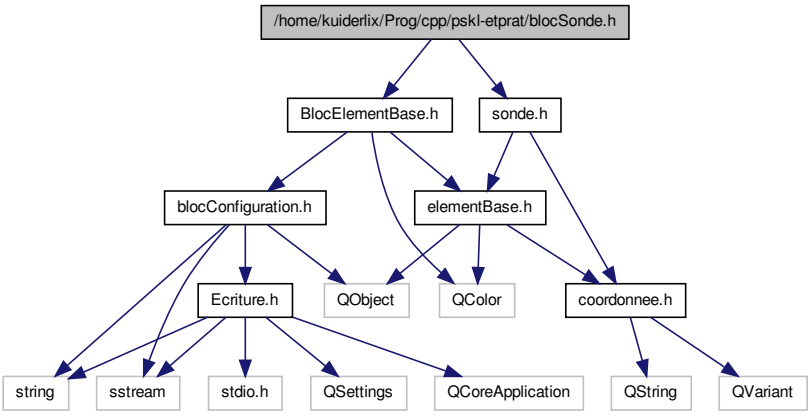
`#include <vector>` `#include "blocSonde.h"` Graphe des dépendances par inclusion de blocSonde.cpp :



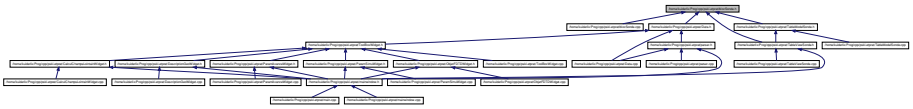
5.20 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSonde.h

`#include "BlocElementBase.h"` `#include "sonde.h"` Graphe des

dépendances par inclusion de blocSonde.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



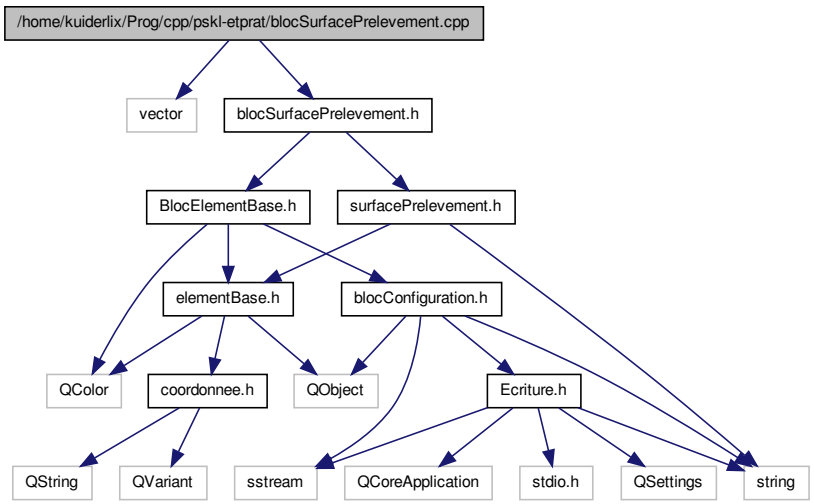
Classes

– class **blocSonde**
Cette classe représente une liste de sonde.

5.21 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSurfacePrelevement.cpp

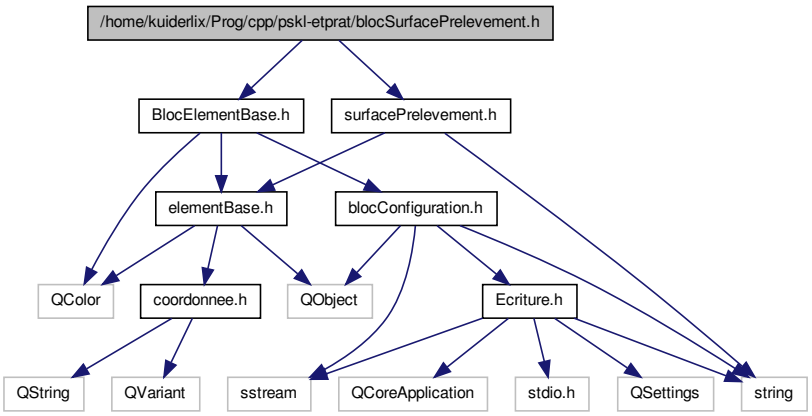
```
#include <vector>    #include "blocSurfacePrelevement.h" ×
```

Graphe des dépendances par inclusion de blocSurfacePrelevement.cpp :

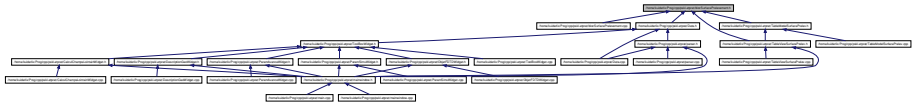


5.22 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/blocSurfacePrelevement.h

`#include "surfacePrelevement.h"` `#include "BlocElementBase.h"` Graphe des dépendances par inclusion de blocSurfacePrelevement.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



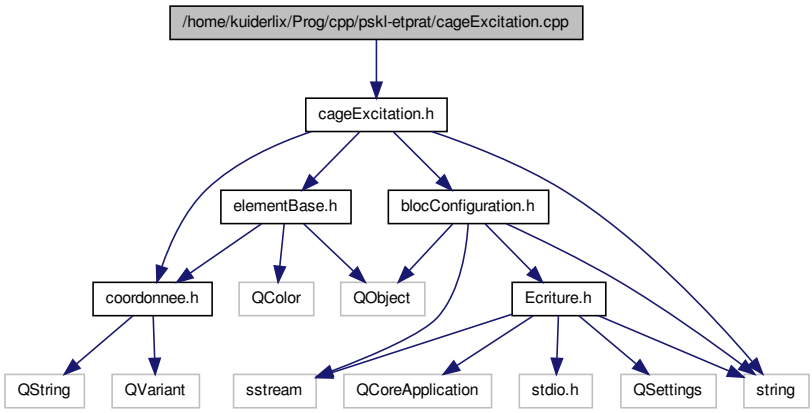
Classes

– class **blocSurfacePrelevement**
Cette classe représente une liste de ***surfacePrelevement*** *(p. 154).*

5.23 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cageExcitation.cpp

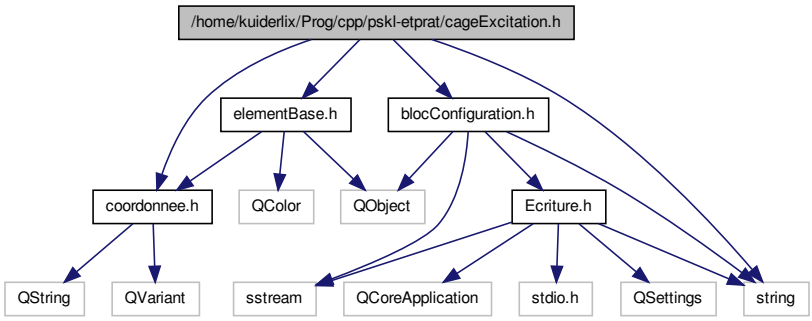
`#include "cageExcitation.h"` Graphe des dépendances par inclusion de

cageExcitation.cpp :

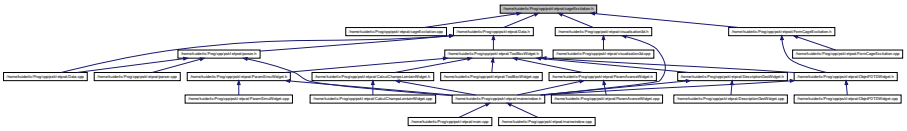


5.24 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cageExcitation.h

`#include "elementBase.h"` `#include "coordonnee.h"` `#include "blocConfiguration.h"` `#include <string>` Graphe des dépendances par inclusion de cageExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

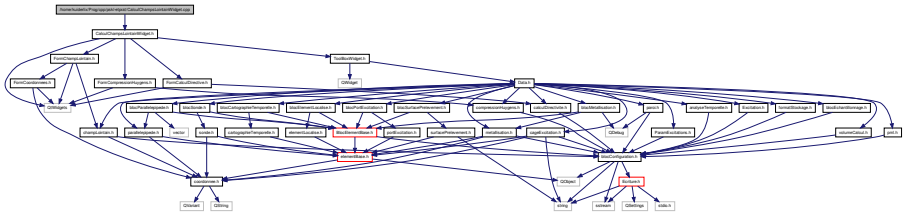


Classes

– class **cageExcitation**
Cette classe représente la variable cageExcitations d'une simulation.

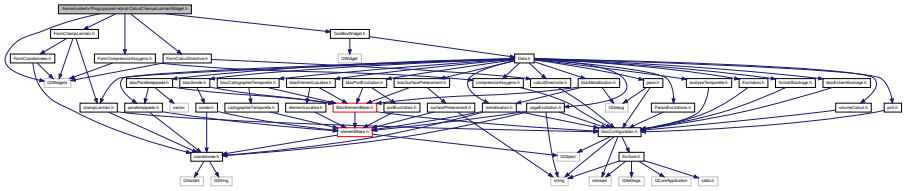
5.25 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-CalculChampsLointainWidget.cpp

`#include "CalculChampsLointainWidget.h"` Graphe des dépendances par inclusion de CalculChampsLointainWidget.cpp :

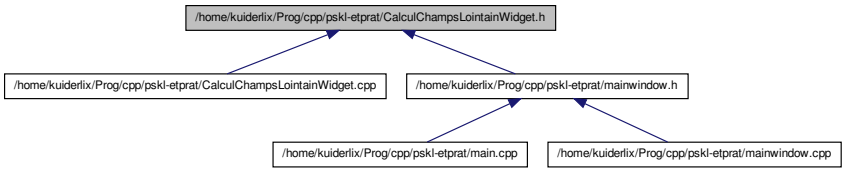


5.26 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-CalculChampsLointainWidget.h

`#include <QtWidgets>` `#include "FormCompressionHuygens.-h"` `#include "FormCalculDirective.h"` `#include "FormChamp-Lointain.h"` `#include "ToolBoxWidget.h"` Graphe des dépendances par inclusion de CalculChampsLointainWidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

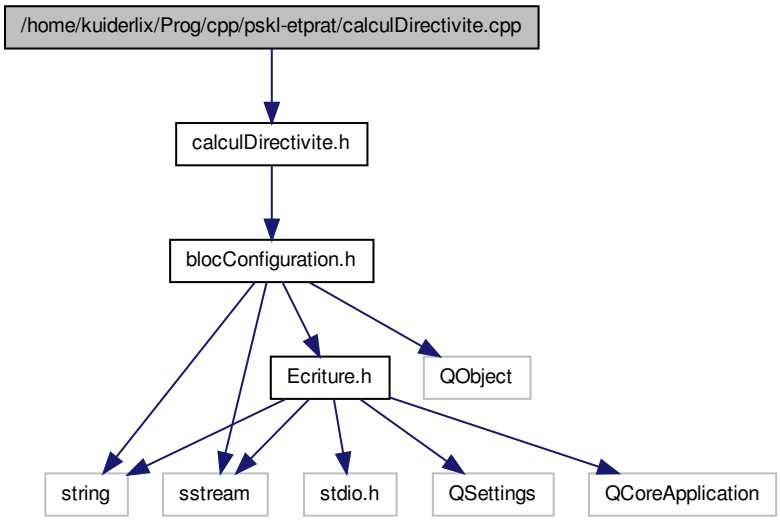


Classes

– class **CalculChampsLointainWidget**
Cette classe représente le formulaire pour modifier un ChampsLointain.

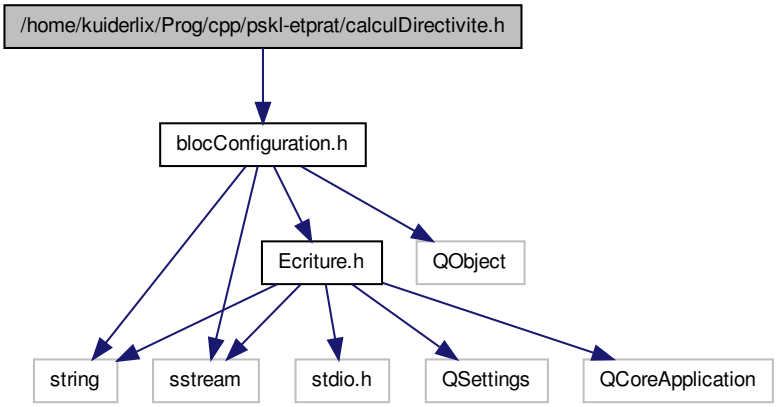
5.27 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/calculDirectivite.cpp

`#include "calculDirectivite.h"` Graphe des dépendances par inclusion de calculDirectivite.cpp :



5.28 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/calculDirectivite.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion de calculDirectivite.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

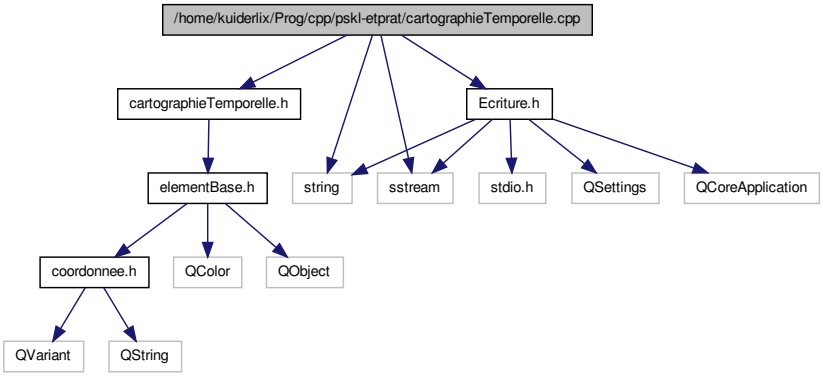
Classes

– class **calculDirectivite**
*Cette classe représente la variable **calculDirectivite*** (p. 47) *d'une simulation.*

5.29 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cartographieTemporelle.cpp

`#include "cartographieTemporelle.h"` `#include <string>` × `#include <sstream>` `#include "Ecriture.h"` Graphe des dépendances

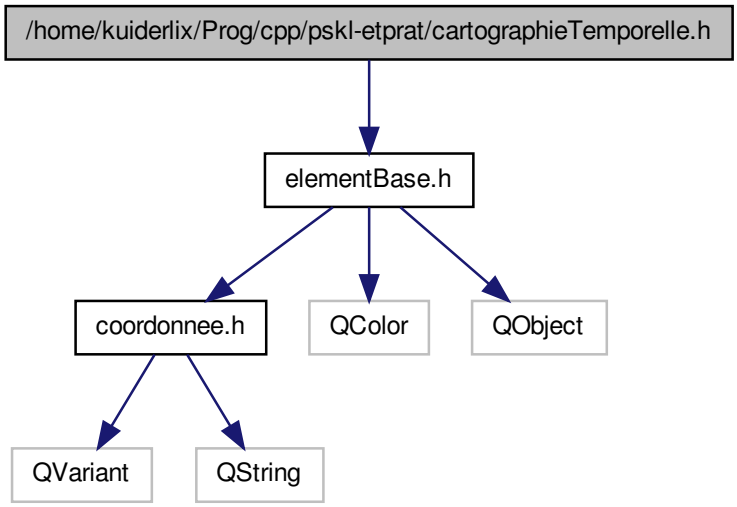
par inclusion de cartographieTemporelle.cpp :



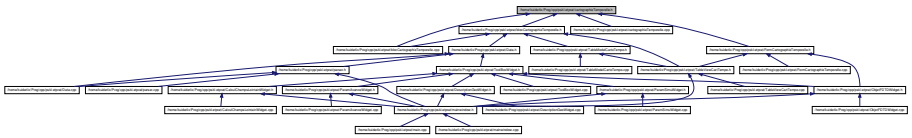
5.30 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cartographie-Temporelle.h

`#include "elementBase.h"` Graphe des dépendances par inclusion de

cartographieTemporelle.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



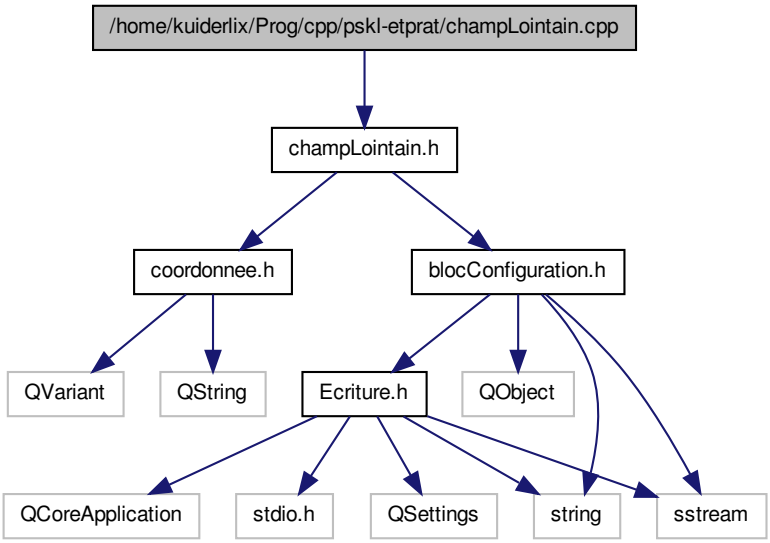
Classes

– class **cartographieTemporelle**
Cette classe représente la variable ***cartographieTemporelle*** *(p. 49) d'une simulation.*

5.31 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/champLointain.cpp

`#include "champLointain.h"` Graphe des dépendances par inclusion de

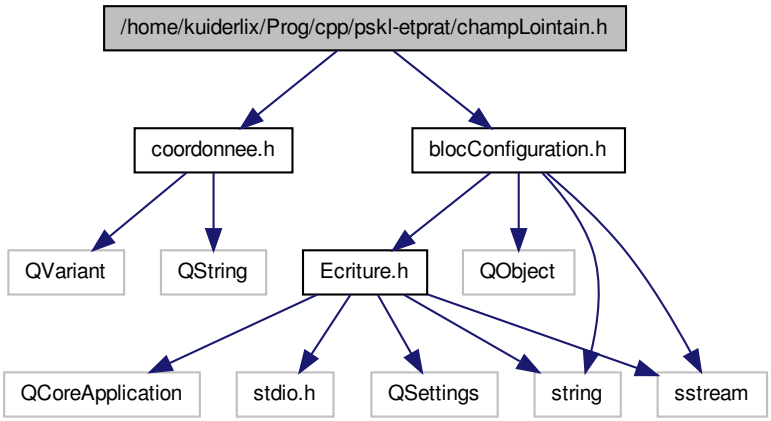
champLointain.cpp :



5.32 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/champLointain.h

```
#include "coordonnee.h"   #include "blocConfiguration.h" ×
```

Graphe des dépendances par inclusion de champLointain.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

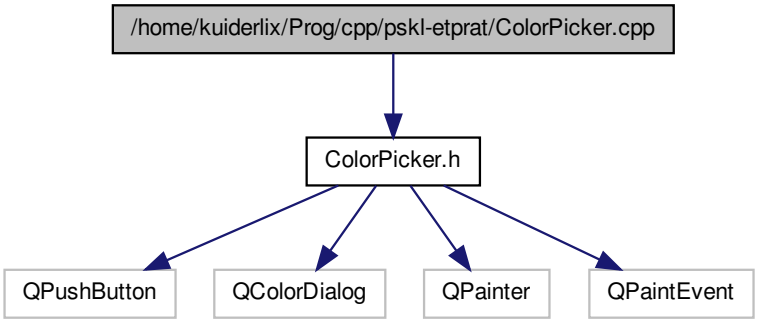
Classes

– class **champLointain**
*Cette classe représente la variable **champLointain** (p. 52) d'une simulation.*

5.33 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ColorPicker.cpp

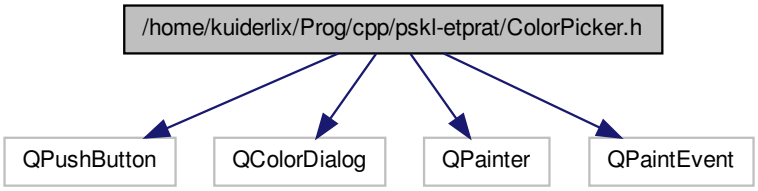
`#include "ColorPicker.h"` Graphe des dépendances par inclusion de Color-

Picker.cpp :

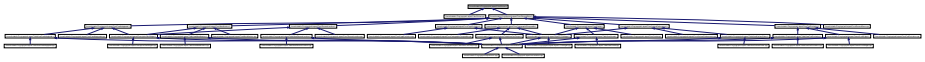


5.34 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ColorPicker.h

`#include <QPushButton>` `#include <QColorDialog>` `#include <QPainter>` `#include <QPaintEvent>` Graphe des dépendances par inclusion de ColorPicker.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

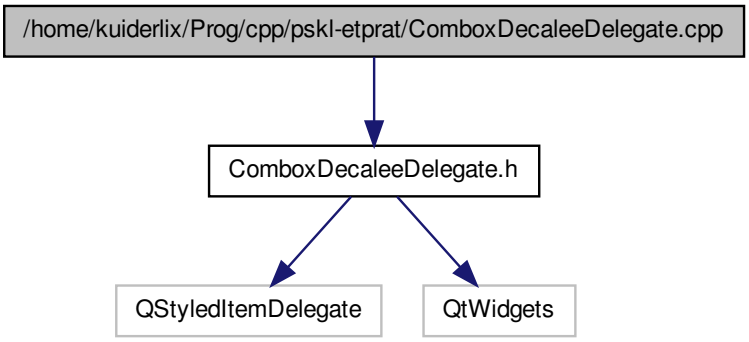


Classes

– class **ColorPicker**
Cette classe est un widget sous forme de bouton qui permet de sélectionner une couleur dans une boite de dialogue qui s'ouvre quand le bouton est pressé. Un carré de la couleur sélectionnée est présent sur le bouton.

5.35 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ComboxDecaleeDelegate.cpp

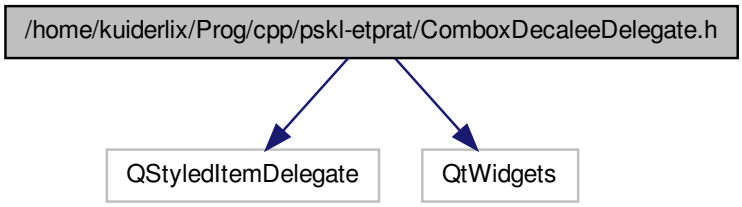
`#include "ComboxDecaleeDelegate.h"` Graphe des dépendances par inclusion de ComboxDecaleeDelegate.cpp :



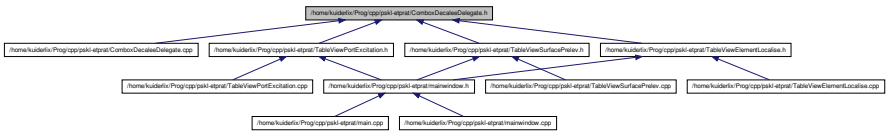
5.36 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ComboxDecaleeDelegate.h

`#include <QStyledItemDelegate>` `#include <QtWidgets>` Graphe

des dépendances par inclusion de ComboxDecaleeDelegate.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



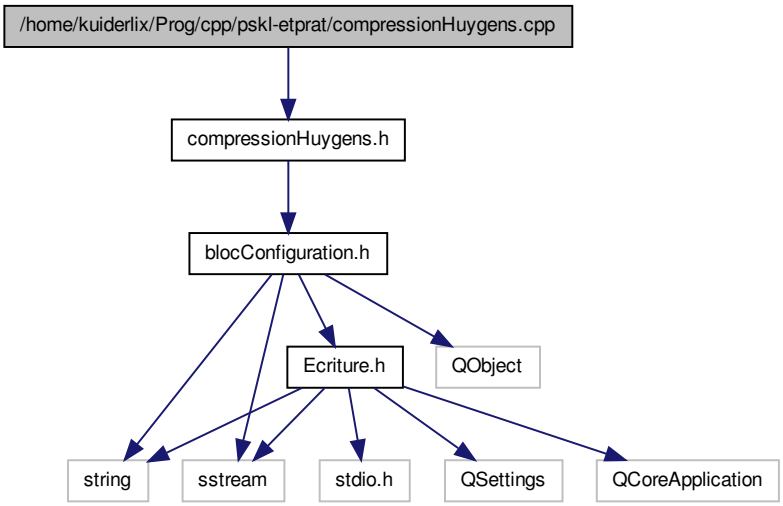
Classes

– class **ComboxDecaleeDelegate**
Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur sous forme de liste déroulante.

5.37 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/compression-Huygens.cpp

`#include "compressionHuygens.h"` Graphe des dépendances par inclusion

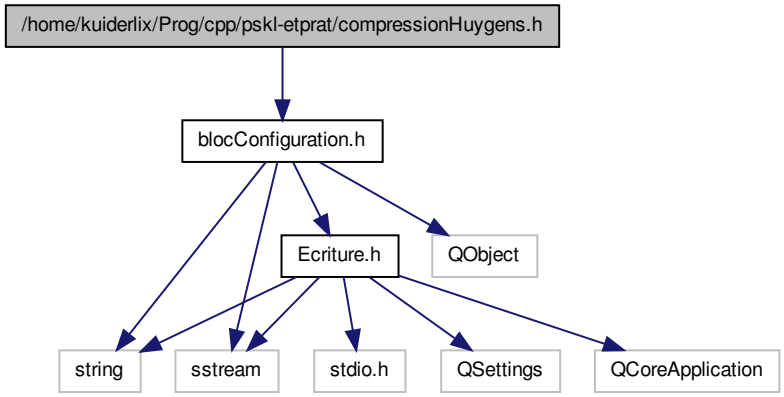
de compressionHuygens.cpp :



5.38 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/compressionHuygens.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion

de compressionHuygens.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

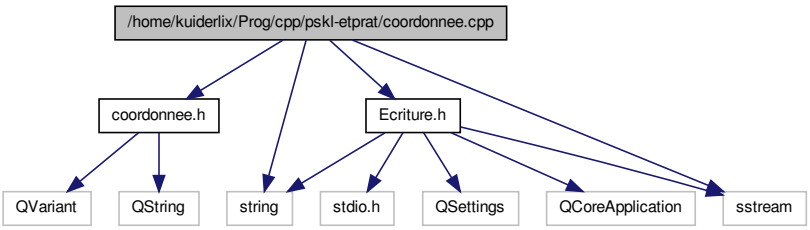
Classes

– class **compressionHuygens**
Cette classe représente la variable ***compressionHuygens*** *(p. 59) d'une simulation.*

5.39 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/coordonnee.cpp

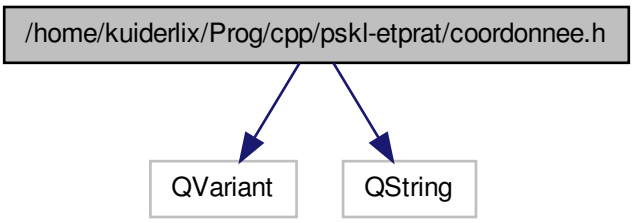
`#include "coordonnee.h"` `#include "Ecriture.h"` `#include <string>` `#include <sstream>` Graphe des dépendances par inclusion

de coordonnee.cpp :



5.40 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/coordonnee.h

`#include <QVariant>` `#include <QString>` Graphe des dépendances par inclusion de coordonnee.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

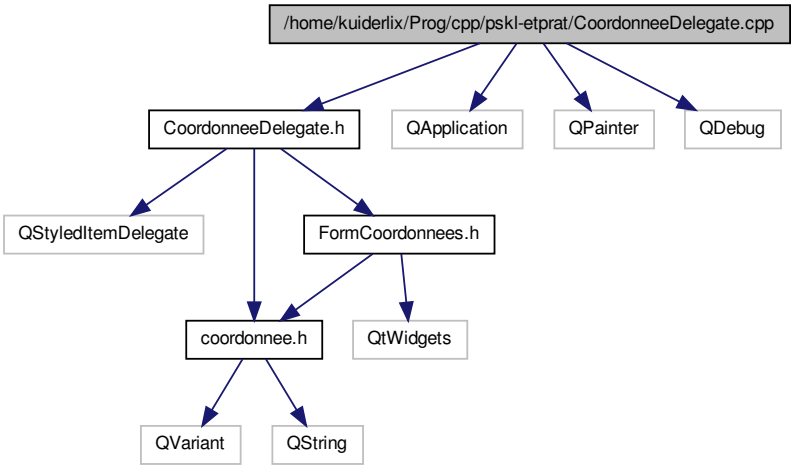
Classes

– class **coordonnee**

Cette classe représente une coordonnee en 3 dimensions. Elle sert d'unité de position dans la scène 3D.

5.41 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-CoordonneeDelegate.cpp

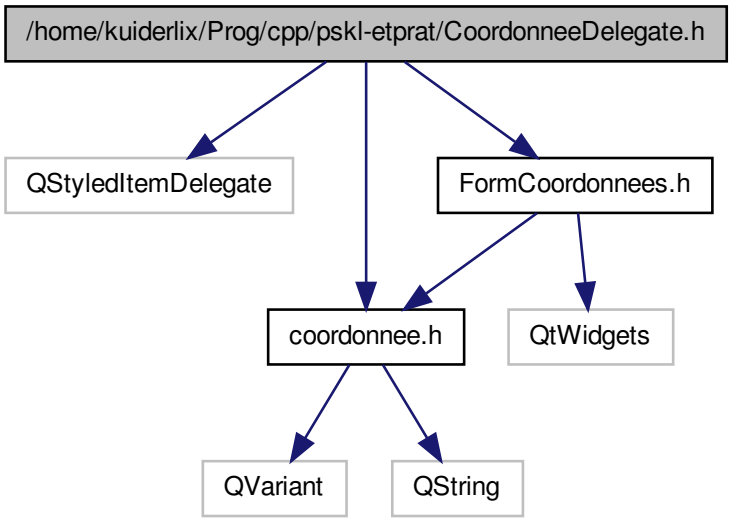
`#include "CoordonneeDelegate.h"` `#include <QApplication>` × `#include <QPainter>` `#include <QDebug>` Graphe des dépendances par inclusion de CoordonneeDelegate.cpp :



5.42 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-CoordonneeDelegate.h

`#include <QStyledItemDelegate>` `#include "coordonnee.h"` × `#include "FormCoordonnees.h"` Graphe des dépendances par inclusion de

CoordonneeDelegate.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

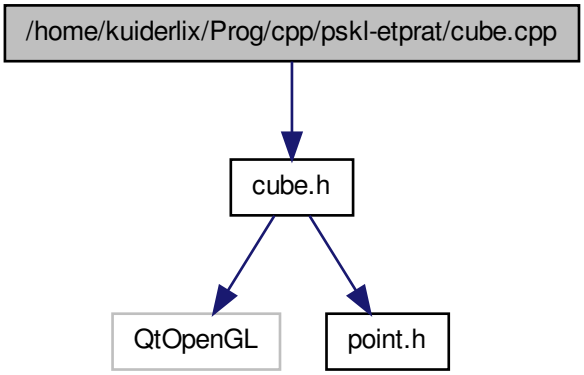
Classes

– class **CoordonneeDelegate**

Cette classe permet de personnalisé l'éditeur d'un tableau pour une certaines colonnes. Elle fourni un éditeur pour modifier une coordonnees. Cette éditeur est en faite un ***FormCoordonnees*** *(p. 93).*

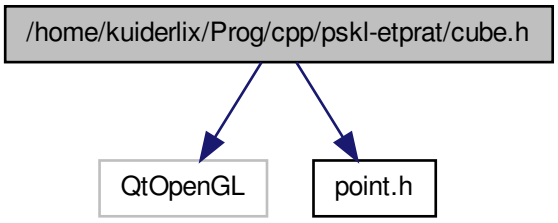
5.43 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cube.cpp

`#include "cube.h"` Graphe des dépendances par inclusion de cube.cpp :

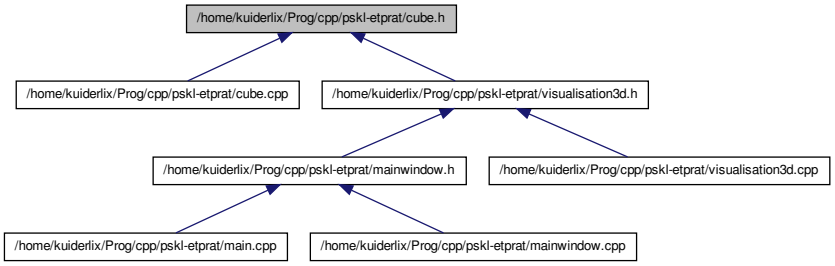


5.44 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/cube.h

`#include <QtOpenGL>` `#include "point.h"` Graphe des dépendances par inclusion de cube.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

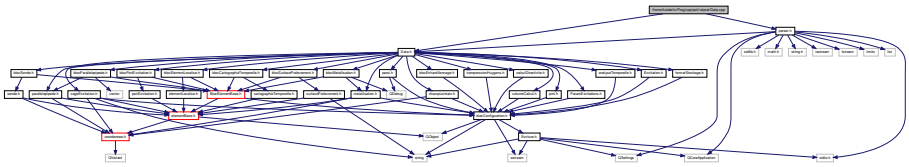


Classes

– class **Cube**

5.45 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-Data.cpp

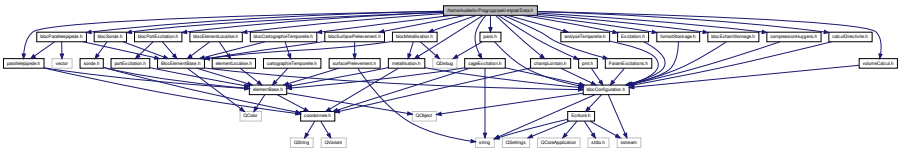
`#include "Data.h"` `#include "parser.h"` Graphe des dépendances par inclusion de Data.cpp :



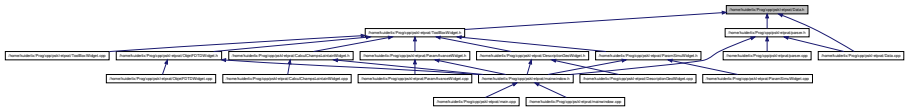
5.46 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-Data.h

`#include "volumeCalcul.h"` `#include "paroi.h"` `#include "metallisation.h"` `#include "blocMetallisation.h"` `#include "parallelepipede.h"` `#include "blocParallelepipede.h"` × `#include "blocElementLocalise.h"` `#include "pml.h"` `#include "ParamExcitations.h"` `#include "analyseTemporelle.h"` `#include "blocPortExcitation.h"` `#include "cageExcitation.h"` `#include "blocSonde.h"` `#include "blocCartographieTemporelle.h"`

`#include "blocSurfacePrelevement.h"` `#include "champLointain.h"` `#include "Excitation.h"` `#include "formatStockage.h"` × `#include "blocEchantillonnage.h"` `#include "compressionHuygens.h"` `#include "calculDirectivite.h"` Graphe des dépendances par inclusion de Data.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

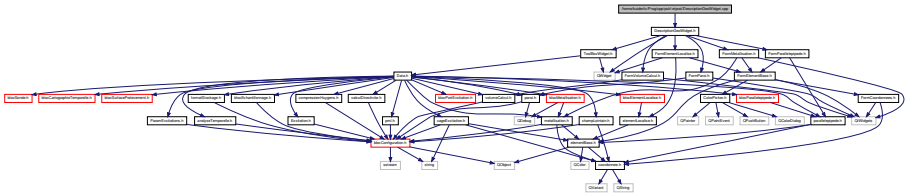


Classes

– class **Data**
  Contient toutes les données utiles pour la simulation.

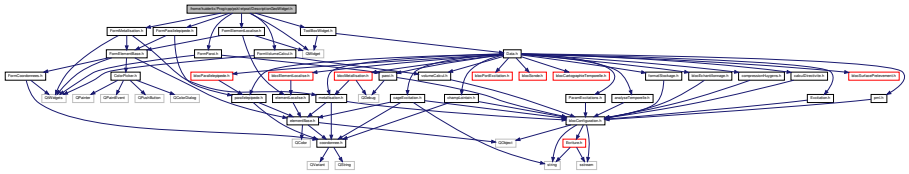
5.47 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-DescriptionGeoWidget.cpp

`#include "DescriptionGeoWidget.h"` Graphe des dépendances par inclusion de DescriptionGeoWidget.cpp :

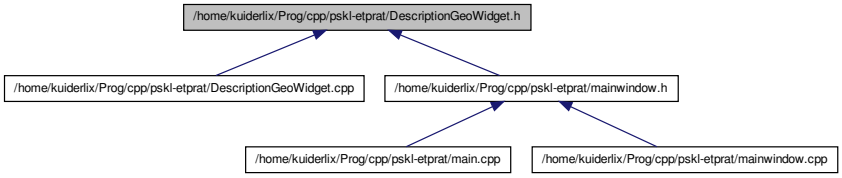


5.48 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-DescriptionGeoWidget.h

`#include <QWidget>` `#include "FormParallelepipede.h"` × `#include "FormMetallisation.h"` `#include "FormVolumeCalcul.-h"` `#include "FormElementLocalise.h"` `#include "FormParoi.-h"` `#include "ToolBoxWidget.h"` Graphe des dépendances par inclusion de DescriptionGeoWidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



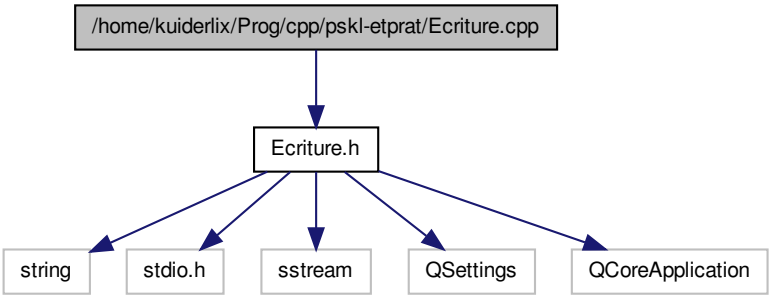
Classes

– class **DescriptionGeoWidget**
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

5.49 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-Ecriture.cpp

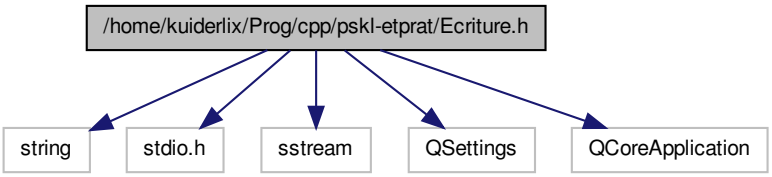
`#include "Ecriture.h"` Graphe des dépendances par inclusion de Ecriture.-

cpp :



5.50 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-Ecriture.h

`#include <string>` `#include <stdio.h>` `#include <sstream>` × `#include <QSettings>` `#include <QCoreApplication>` Graphe des dépendances par inclusion de Ecriture.h :



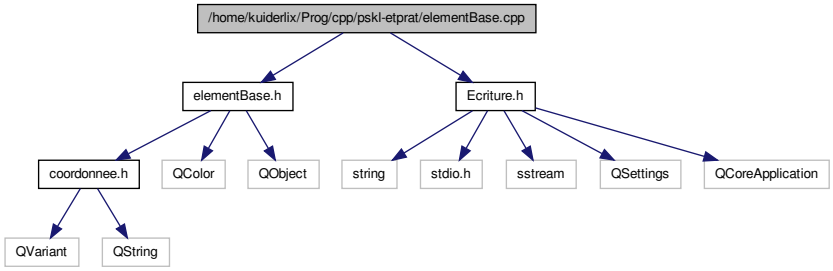
Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

– class **Ecriture**

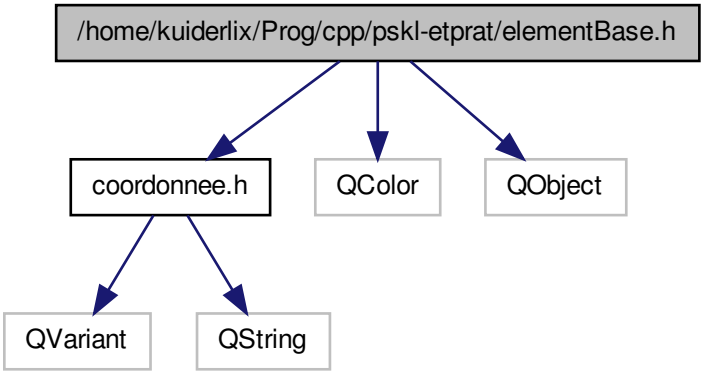
5.51 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/elementBase.cpp

`#include "elementBase.h"` `#include "Ecriture.h"` Graphe des dépendances par inclusion de elementBase.cpp :



5.52 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/elementBase.h

`#include "coordonnee.h"` `#include <QColor>` `#include <QObject>` Graphe des dépendances par inclusion de elementBase.h :



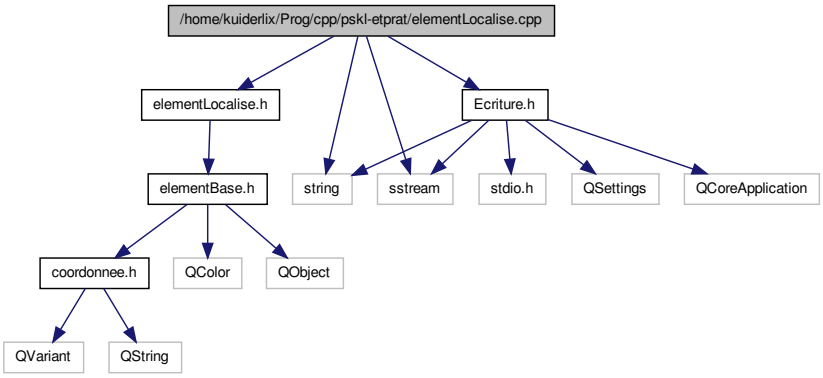
Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

– class **elementBase**
*The **elementBase** (p. 73) class.*

5.53 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/elementLocalise.cpp

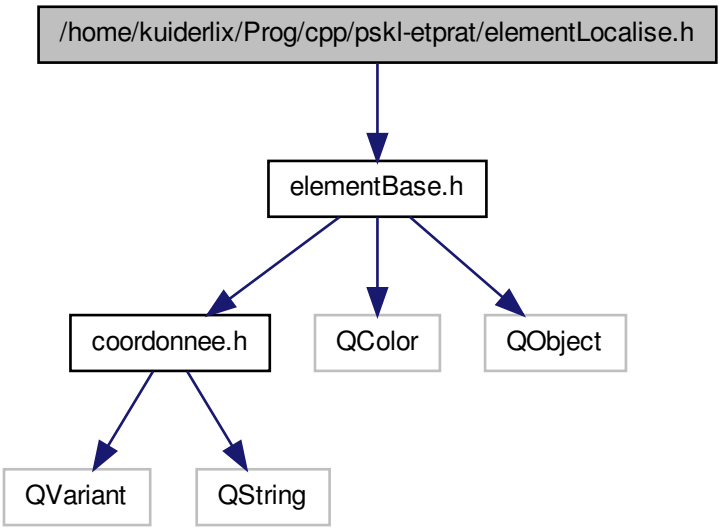
`#include "elementLocalise.h"` `#include <string>` `#include <sstream>` `#include "Ecriture.h"` Graphe des dépendances par inclusion de elementLocalise.cpp :



5.54 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/elementLocalise.h

`#include "elementBase.h"` Graphe des dépendances par inclusion de

elementLocalise.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

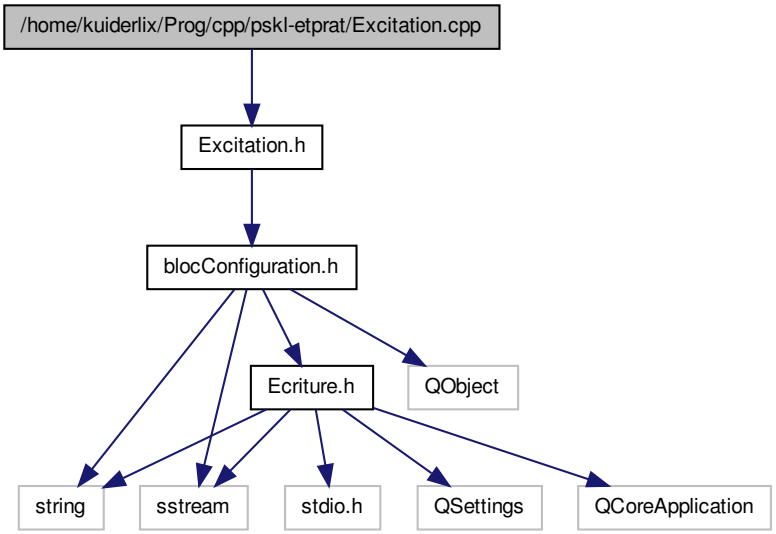
Classes

– class **elementLocalise**
Cette classe représente la variable ***elementLocalise*** *(p. 77) d'une simulation.*

5.55 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-Excitation.cpp

`#include "Excitation.h"` Graphe des dépendances par inclusion de -

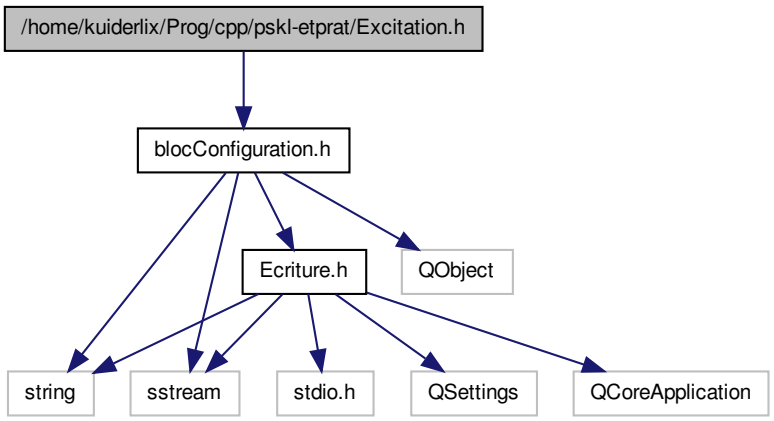
Excitation.cpp :



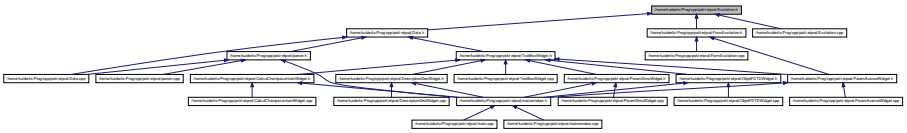
5.56 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-Excitation.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion

de Excitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



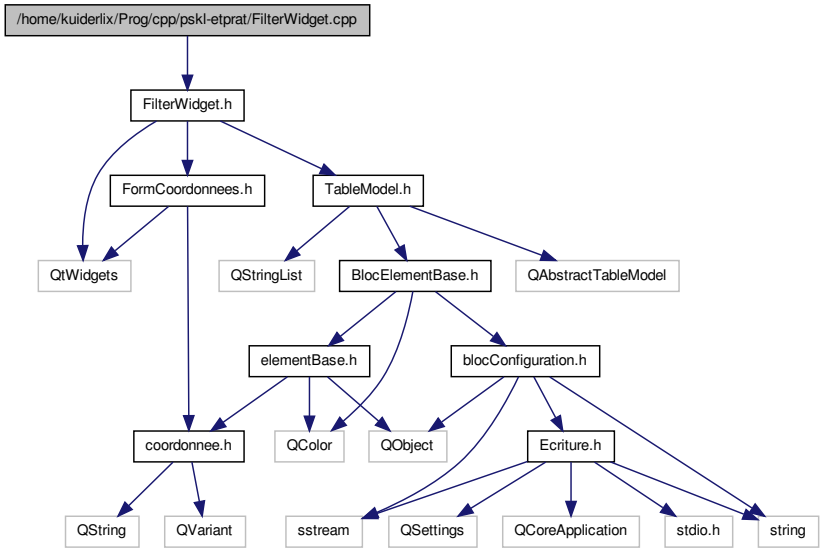
Classes

– class **Excitation**
*Cette classe représente la variable **Excitation** (p. 80) d'une simulation.*

5.57 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FilterWidget.cpp

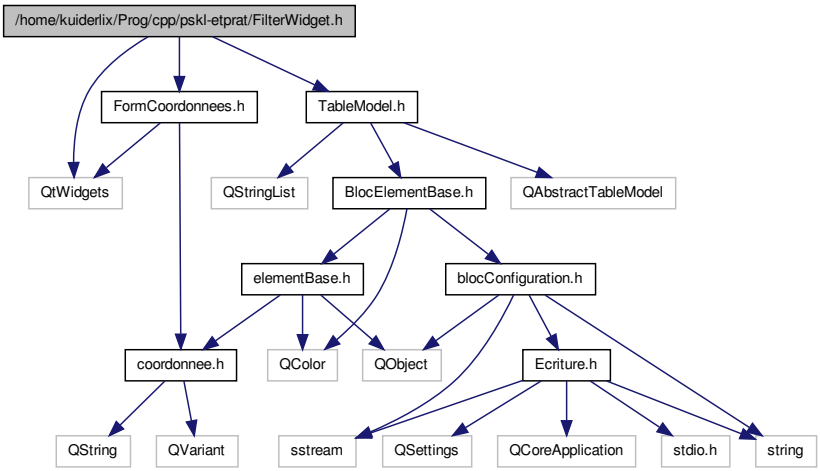
`#include "FilterWidget.h"` Graphe des dépendances par inclusion de Filter-

Widget.cpp :

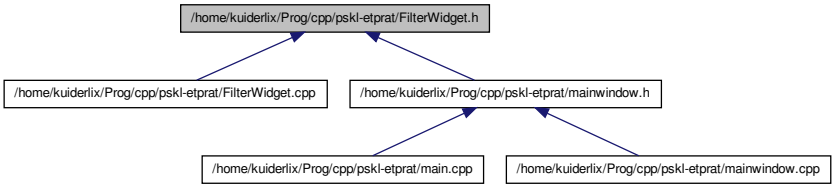


5.58 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FilterWidget.h

`#include <QtWidgets> #include "TableModel.h" #include "-FormCoordonnees.h"` Graphe des dépendances par inclusion de FilterWidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



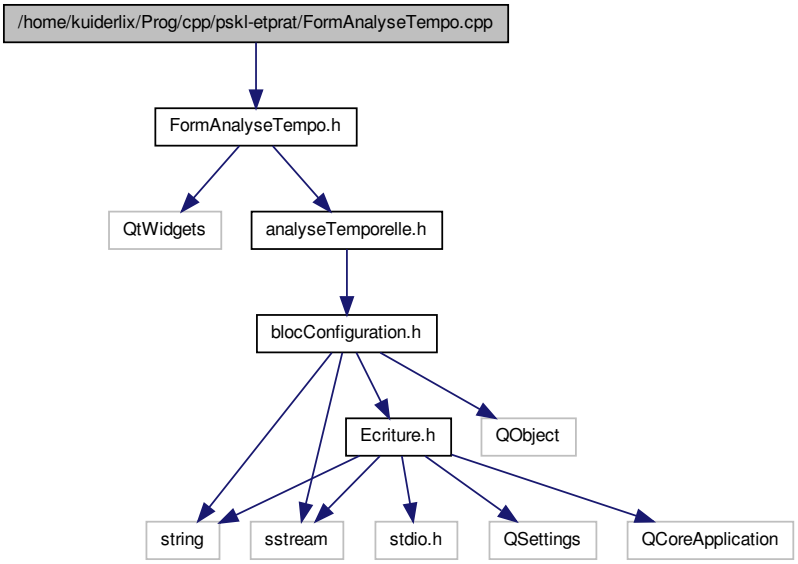
Classes

– class **FilterWidget**
*The **FilterWidget** (p. 82) class.*

5.59 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormAnalyseTempo.cpp

`#include "FormAnalyseTempo.h"` Graphe des dépendances par inclusion de

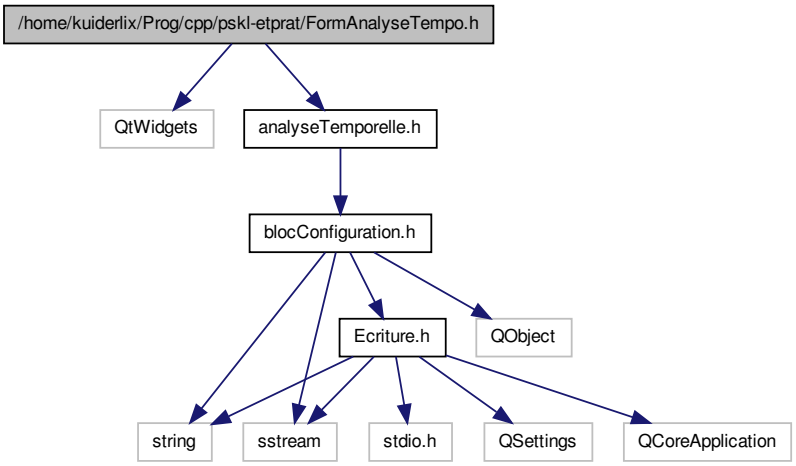
FormAnalyseTempo.cpp :



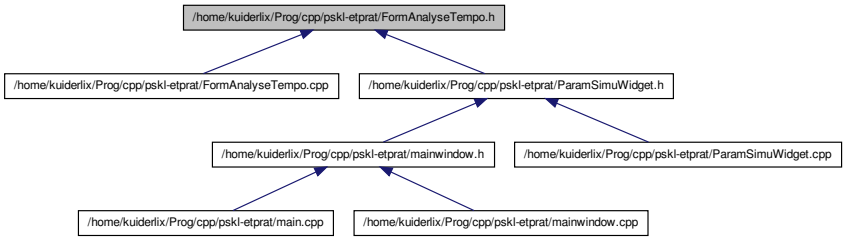
5.60 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormAnalyseTempo.h

`#include <QtWidgets>` `#include "analyseTemporelle.h"` Graphe

des dépendances par inclusion de FormAnalyseTempo.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



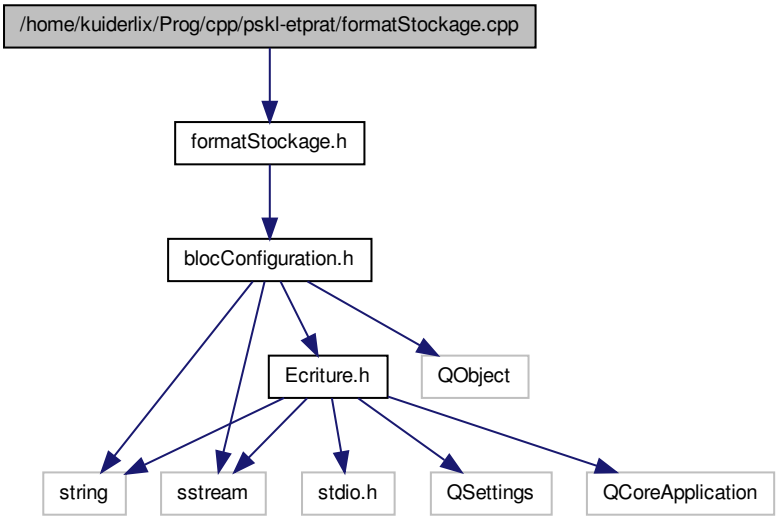
Classes

– class **FormAnalyseTempo**
*The **FormAnalyseTempo** (p. 83) class.*

5.61 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/formatStockage.cpp

`#include "formatStockage.h"` Graphe des dépendances par inclusion de

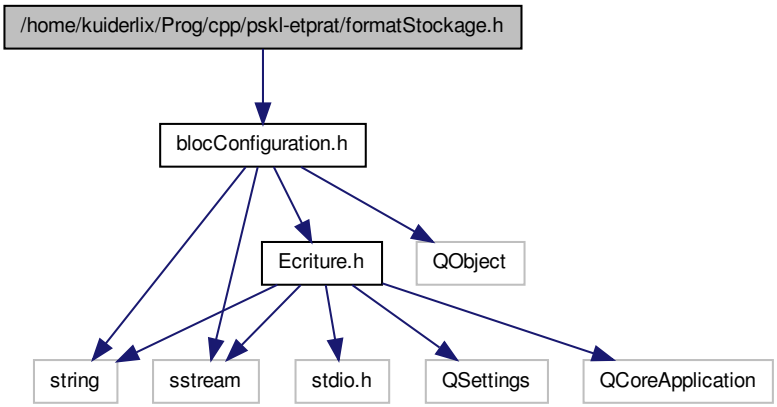
formatStockage.cpp :



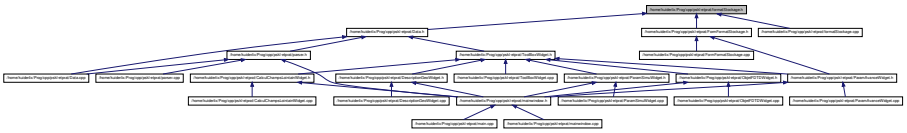
5.62 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/formatStockage.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion

de formatStockage.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



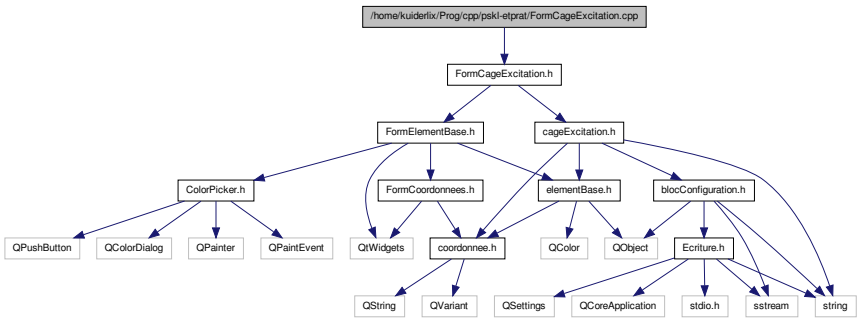
Classes

– class **formatStockage**
*Cette classe représente la variable **formatStockage*** (p. *84*) *d'une simulation.*

5.63 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCageExcitation.cpp

`#include "FormCageExcitation.h"` Graphe des dépendances par inclusion

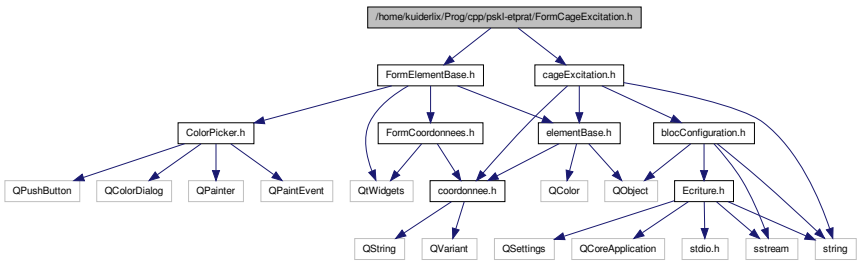
de FormCageExcitation.cpp :



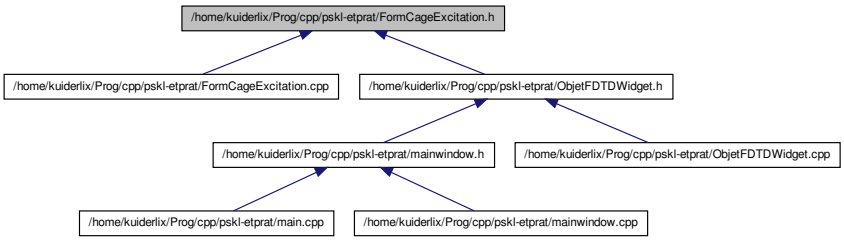
5.64 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCageExcitation.h

`#include "FormElementBase.h"` `#include "cageExcitation.h"`

Graphe des dépendances par inclusion de FormCageExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



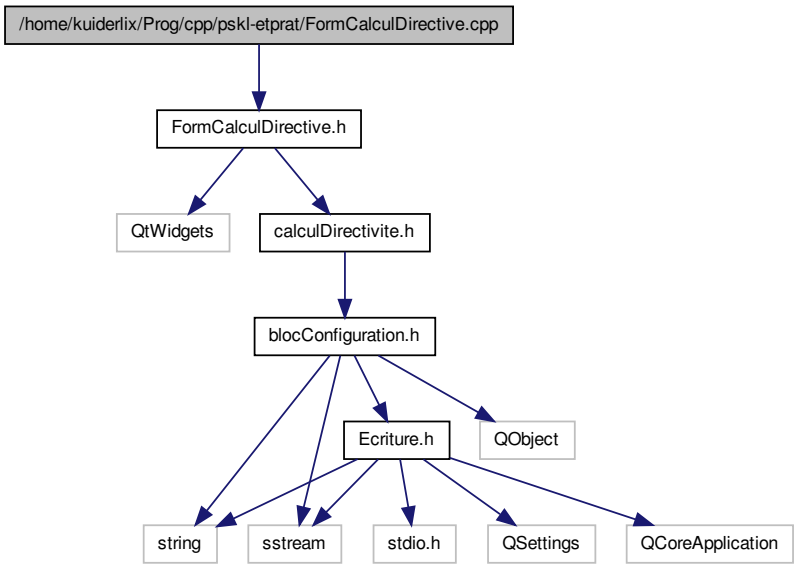
Classes

– class **FormCageExcitation**
*Formulaire permettant d'ajouter un objet **cageExcitation*** (p. *40*).

5.65 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCalculDirective.cpp

`#include "FormCalculDirective.h"` Graphe des dépendances par inclu-

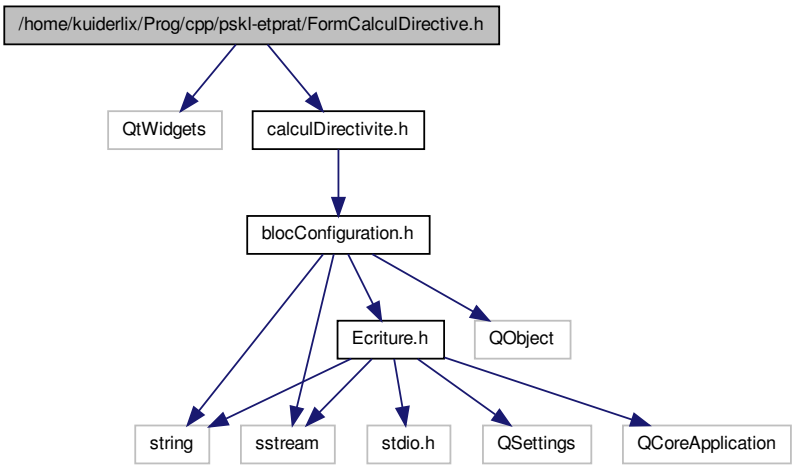
sion de FormCalculDirective.cpp :



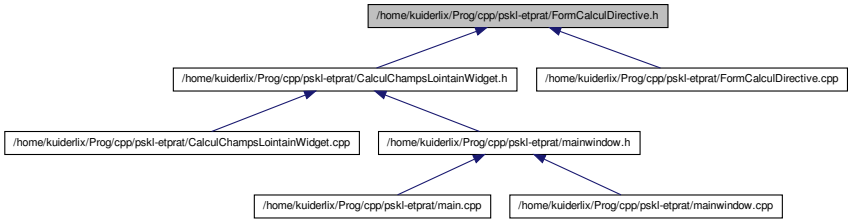
5.66 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCalculDirective.h

`#include <QtWidgets>` `#include "calculDirectivite.h"` Graphe

des dépendances par inclusion de FormCalculDirective.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



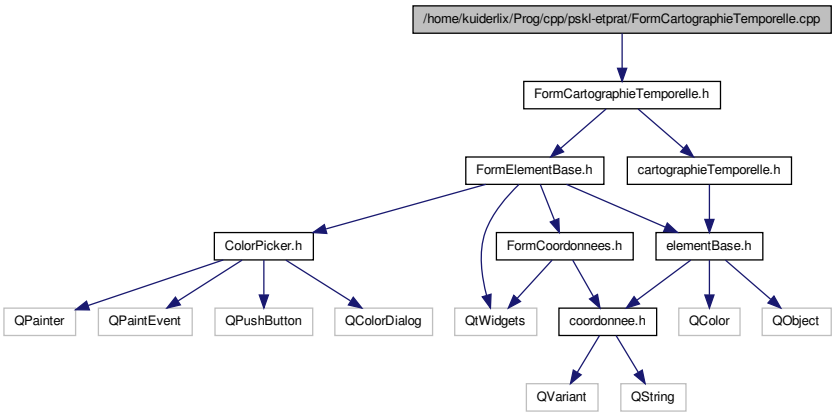
Classes

– class **FormCalculDirective**
*Formulaire permettant de modifier l'objet **calculDirectivite** (p. 47).*

5.67 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCartographieTemporelle.cpp

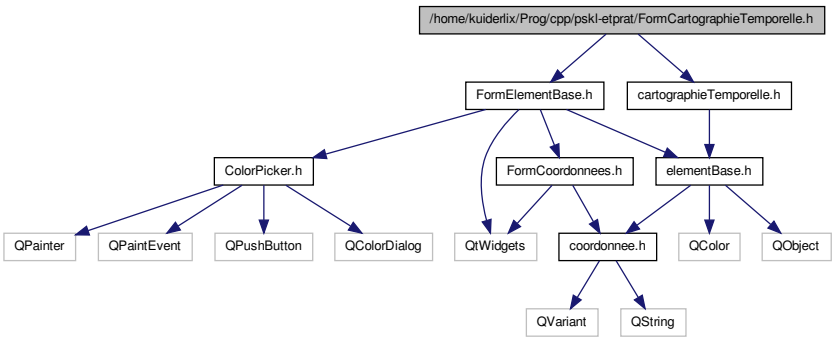
`#include "FormCartographieTemporelle.h"` Graphe des dépendances

par inclusion de FormCartographieTemporelle.cpp :

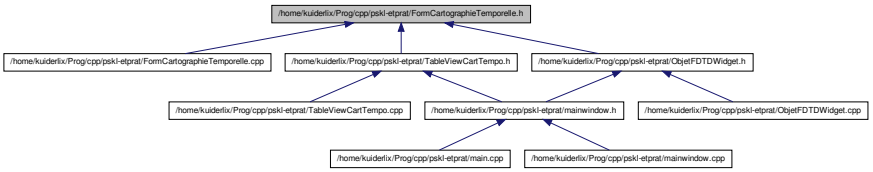


5.68 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCartographieTemporelle.h

`#include "FormElementBase.h"` `#include "cartographieTemporelle.-h"` Graphe des dépendances par inclusion de FormCartographieTemporelle.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

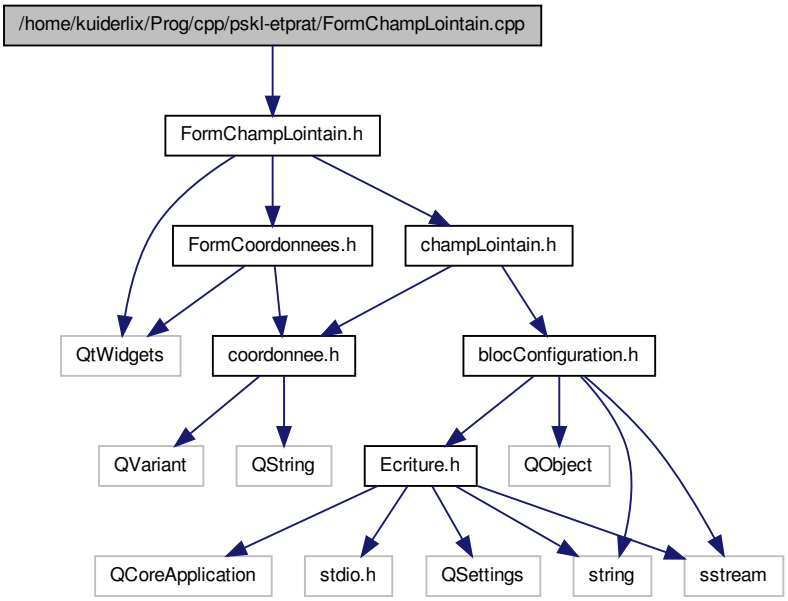


Classes

– class **FormCartographieTemporelle**
Formulaire permettant d'ajouter un nouvelle objet ***cartographieTemporelle*** *(p. 49) dans la scène 3D.*

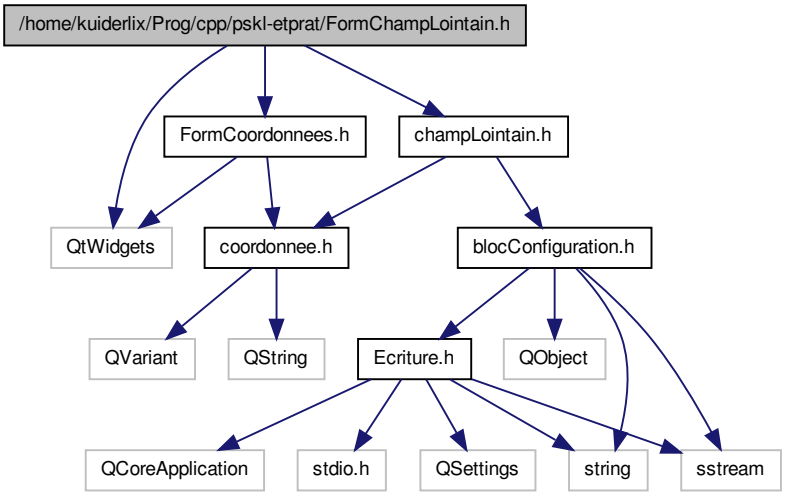
5.69 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormChampLointain.cpp

`#include "FormChampLointain.h"` Graphe des dépendances par inclusion de FormChampLointain.cpp :

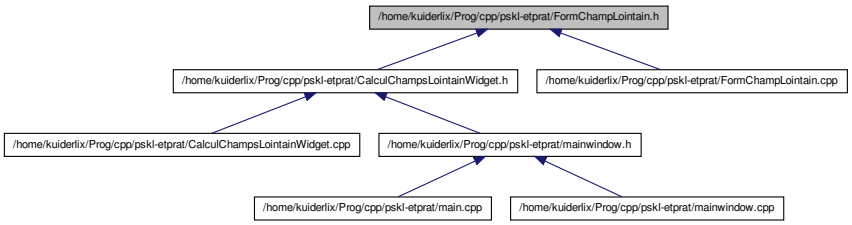


5.70 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormChampLointain.h

`#include <QtWidgets>` `#include "champLointain.h"` `#include "FormCoordonnees.h"` Graphe des dépendances par inclusion de FormChampLointain.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

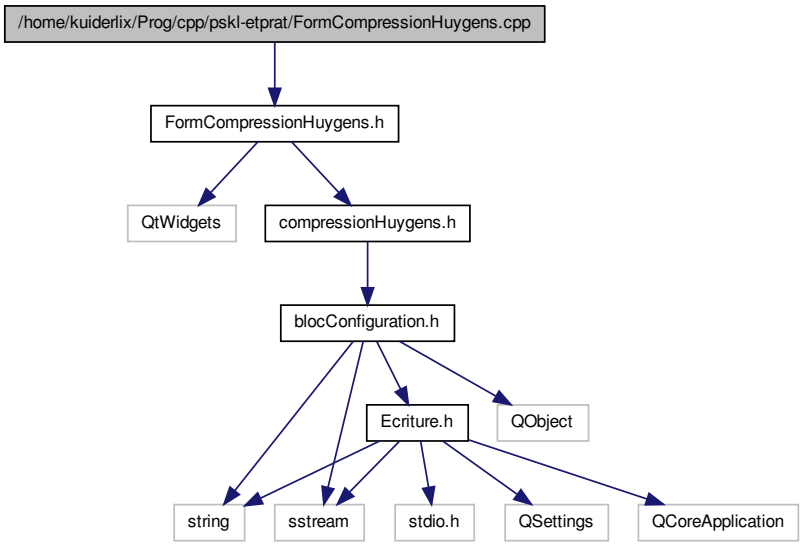


Classes

– class **FormChampLointain**
*Cette classe fournit un formulaire pour modifier l'objet **champLointain** (p. 52).*

5.71 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCompressionHuygens.cpp

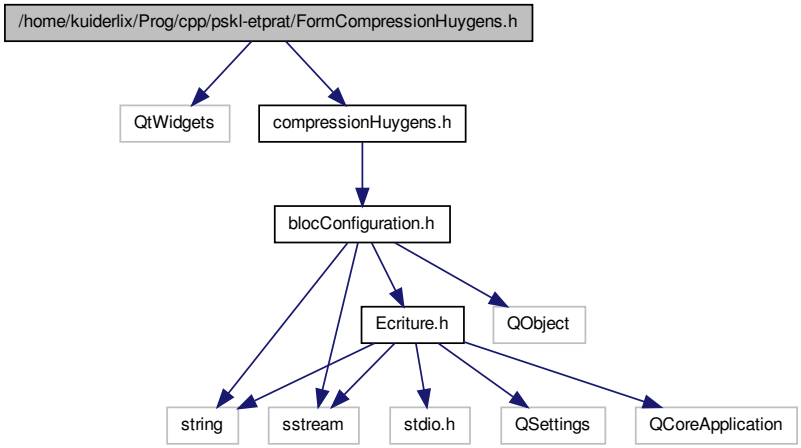
`#include "FormCompressionHuygens.h"` Graphe des dépendances par inclusion de FormCompressionHuygens.cpp :



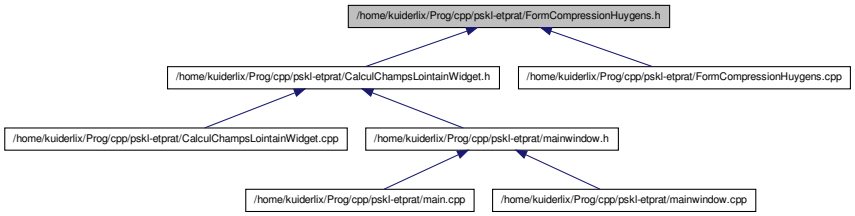
5.72 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCompressionHuygens.h

`#include <QtWidgets>` `#include "compressionHuygens.h"` Graphe

des dépendances par inclusion de FormCompressionHuygens.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



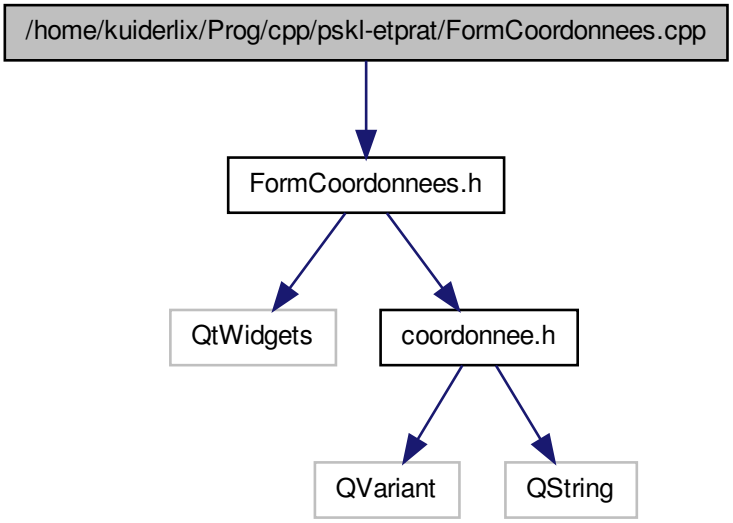
Classes

– class **FormCompressionHuygens**
Cette classe fournit un formulaire pour modifier l'objet ***compressionHuygens*** *(p. 59).*

5.73 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCoordonnees.cpp

`#include "FormCoordonnees.h"` Graphe des dépendances par inclusion de

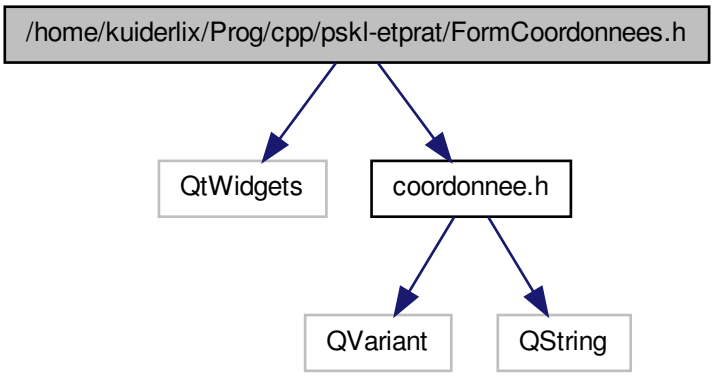
FormCoordonnees.cpp :



5.74 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormCoordonnees.h

`#include <QtWidgets>` `#include "coordonnee.h"` Graphe des dépen-

dances par inclusion de FormCoordonnees.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

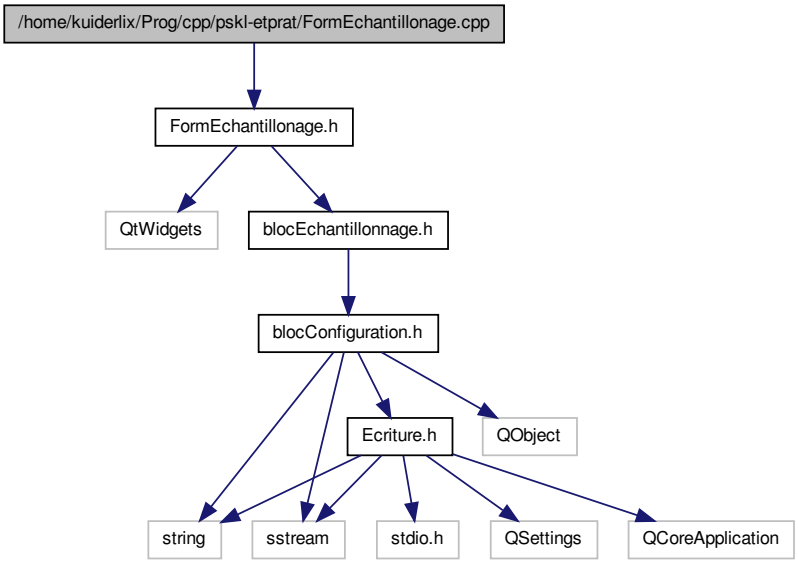
Classes

– class **FormCoordonnees**
Cette classe fournit un widget pour modifier l'objet coordonnee.

5.75 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormEchantillonage.cpp

`#include "FormEchantillonage.h"` Graphe des dépendances par inclusion

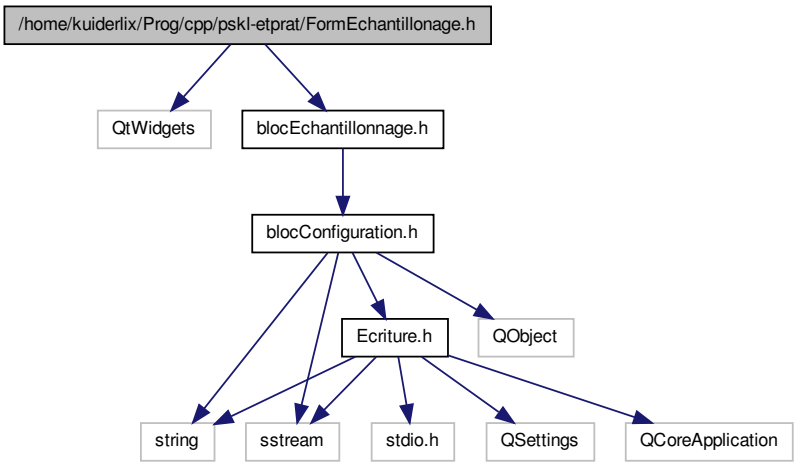
de FormEchantillonage.cpp :



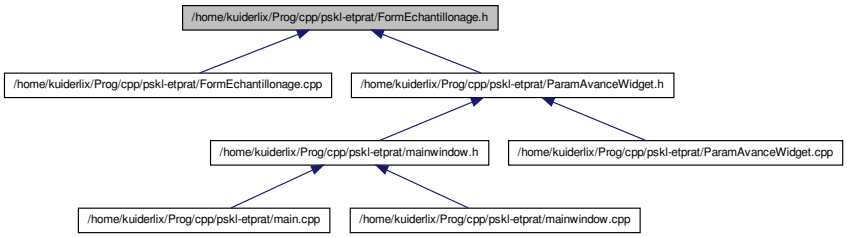
5.76 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormEchantillonage.h

```
#include <QtWidgets>    #include "blocEchantillonnage.h" ×
```

Graphe des dépendances par inclusion de FormEchantillonage.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



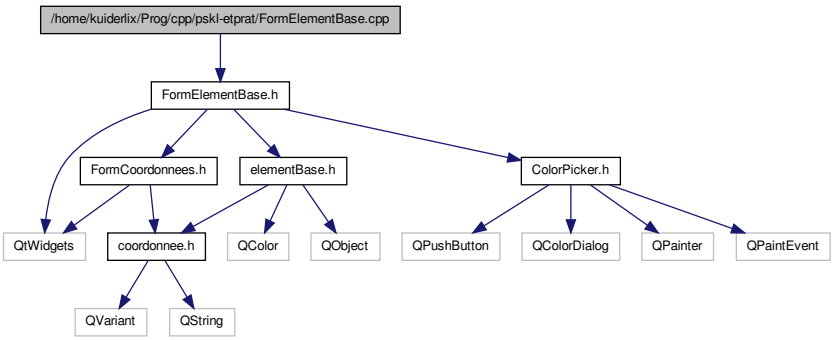
Classes

– class **FormEchantillonage**
Cette classe fournit un formulaire pour modifier l'objet ***blocEchantillonnage*** *(p. 22).*

5.77 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormElementBase.cpp

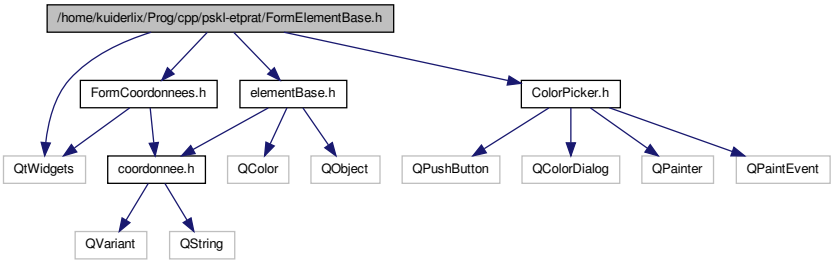
`#include "FormElementBase.h"` Graphe des dépendances par inclusion de

FormElementBase.cpp :

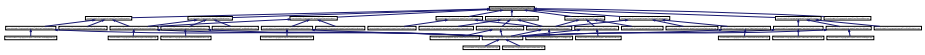


5.78 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormElementBase.h

`#include <QtWidgets>` `#include "FormCoordonnees.h"` `#include "elementBase.h"` `#include "ColorPicker.h"` Graphe des dépendances par inclusion de FormElementBase.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



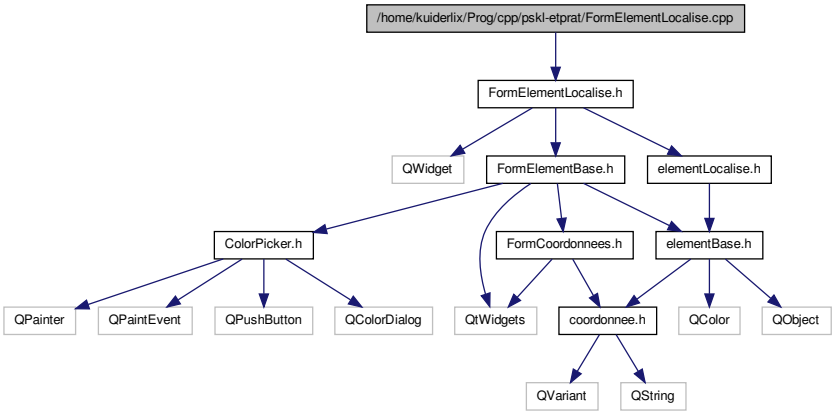
Classes

– class **FormElementBase**

Cette classe sert de base aux formulaires qui modifie des objets ***elementBase*** *(p. 73).*

5.79 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormElementLocalise.cpp

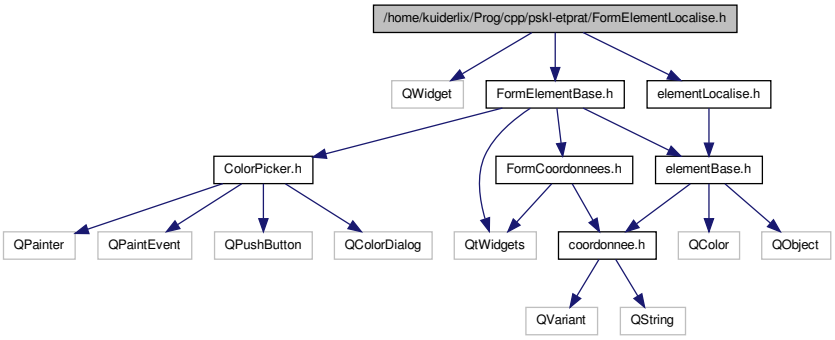
`#include "FormElementLocalise.h"` Graphe des dépendances par inclusion de FormElementLocalise.cpp :



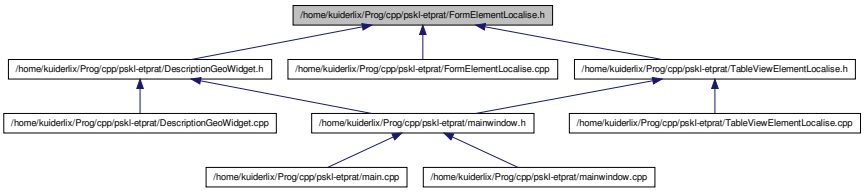
5.80 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormElementLocalise.h

`#include <QWidget>` `#include "FormElementBase.h"` `#include "elementLocalise.h"` Graphe des dépendances par inclusion de FormElement-

Localise.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



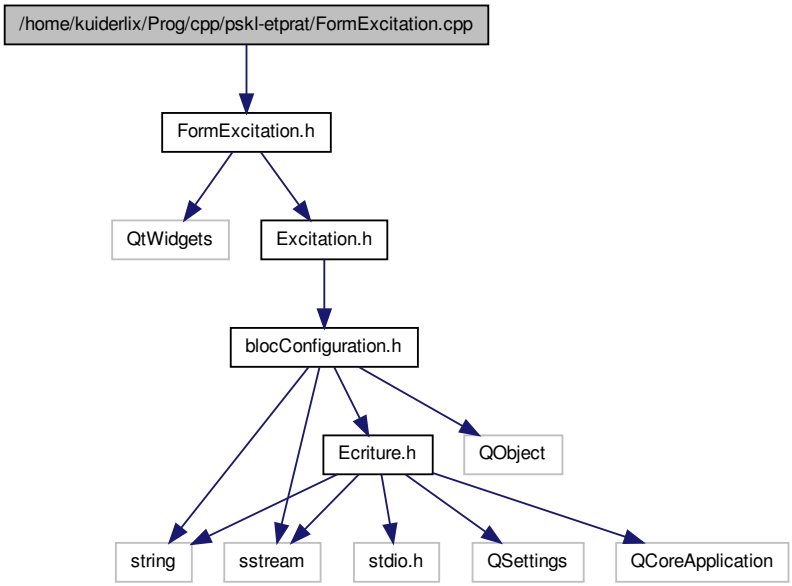
Classes

– class **FormElementLocalise**
*Formulaire permettant d'ajouter un nouvelle objet **elementLocalise*** (p. *77*) *dans la scène 3D.*

5.81 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormExcitation.cpp

`#include "FormExcitation.h"` Graphe des dépendances par inclusion de -

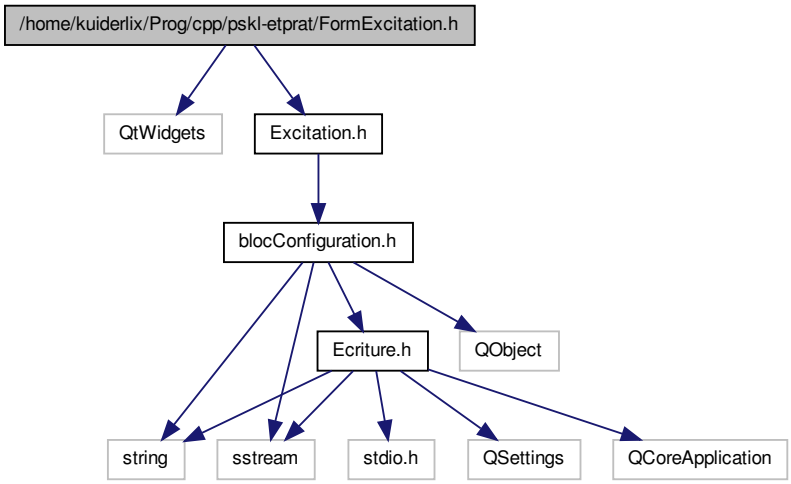
FormExcitation.cpp :



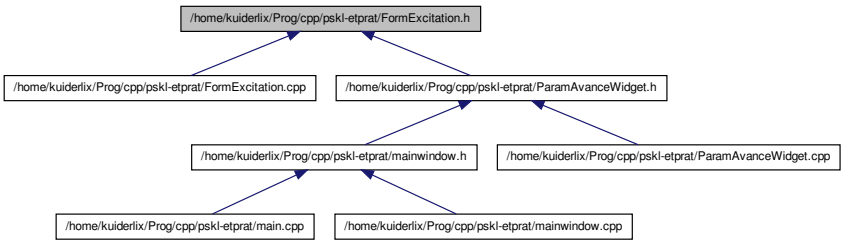
5.82 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormExcitation.h

`#include <QtWidgets>` `#include "Excitation.h"` Graphe des dépen-

dances par inclusion de FormExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



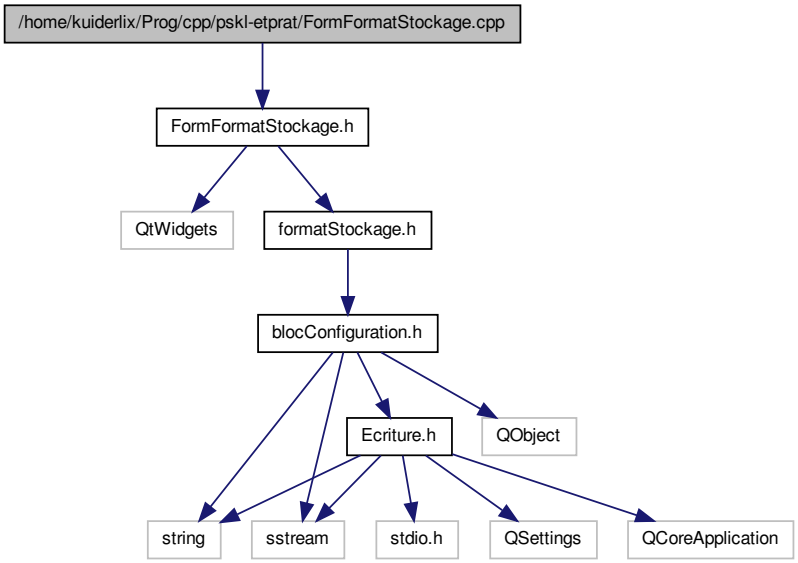
Classes

– class **FormExcitation**
*Cette classe fournit un formulaire pour modifier l'objet **Excitation** (p. 80).*

5.83 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormFormatStockage.cpp

`#include "FormFormatStockage.h"` Graphe des dépendances par inclusion

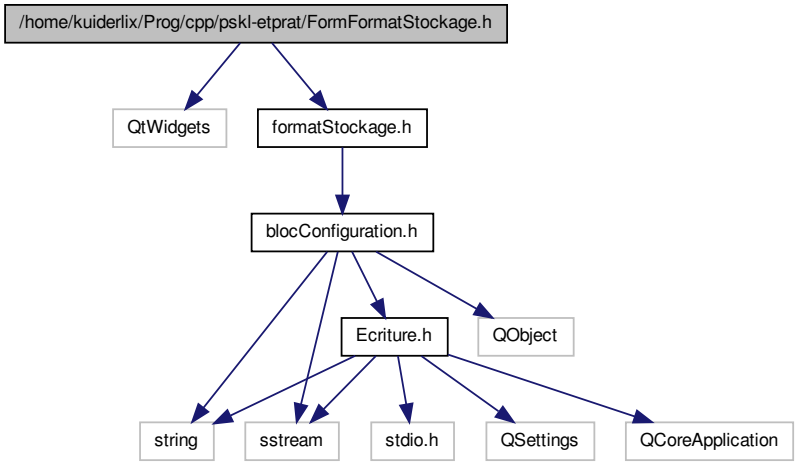
de FormFormatStockage.cpp :



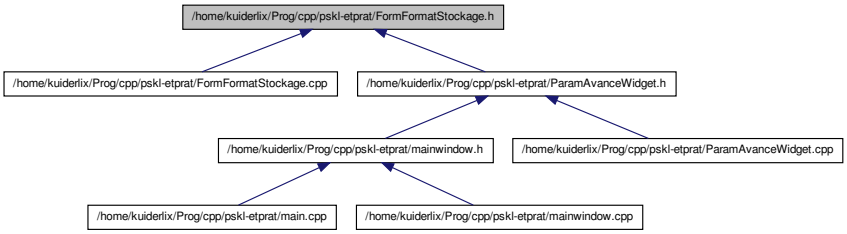
5.84 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormFormatStockage.h

`#include <QtWidgets>` `#include "formatStockage.h"` Graphe des

dépendances par inclusion de FormFormatStockage.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



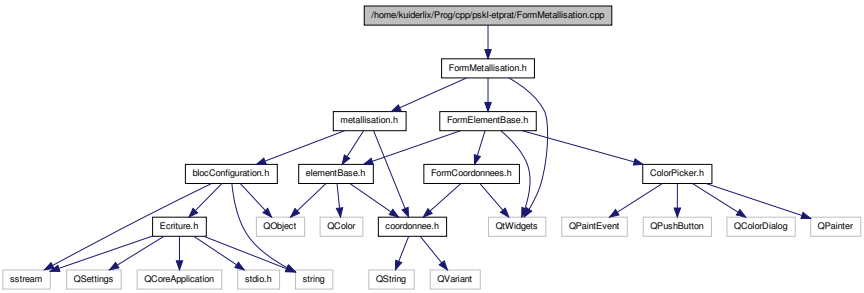
Classes

– class **FormFormatStockage**
*Cette classe fournit un formulaire pour modifier l'objet **formatStockage** (p. 84).*

5.85 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormMetallisation.cpp

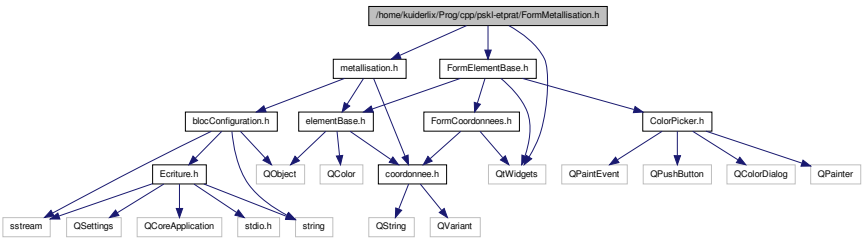
`#include "FormMetallisation.h"` Graphe des dépendances par inclusion

de FormMetallisation.cpp :

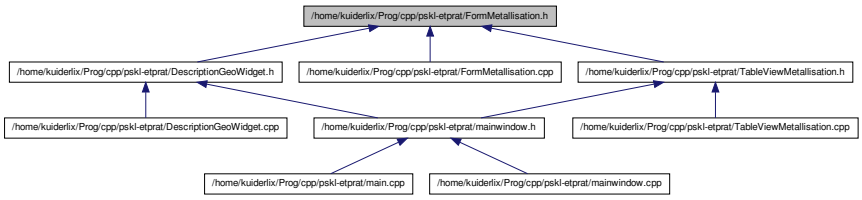


5.86 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormMetallisation.h

`#include <QtWidgets>` `#include "metallisation.h"` `#include "FormElementBase.h"` Graphe des dépendances par inclusion de Form-Metallisation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

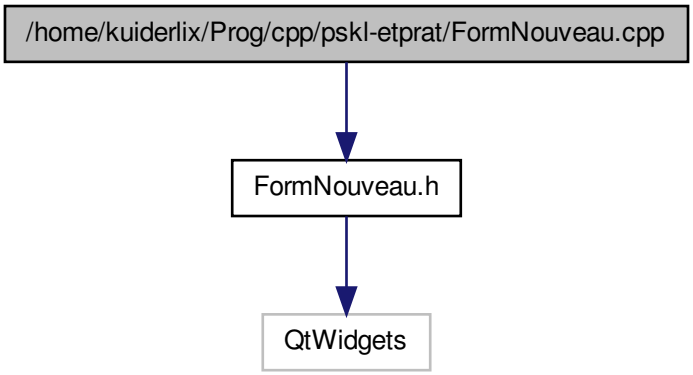


Classes

– class **FormMetallisation**
Formulaire permettant d'ajouter un nouvelle objet metallisation dans la scène 3D.

5.87 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormNouveau.cpp

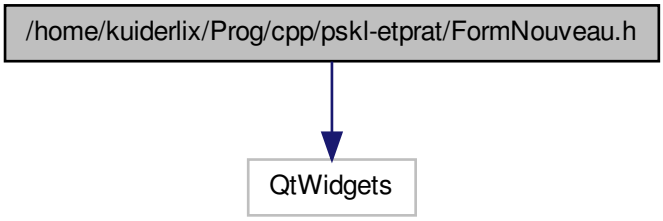
`#include "FormNouveau.h"` Graphe des dépendances par inclusion de Form-Nouveau.cpp :



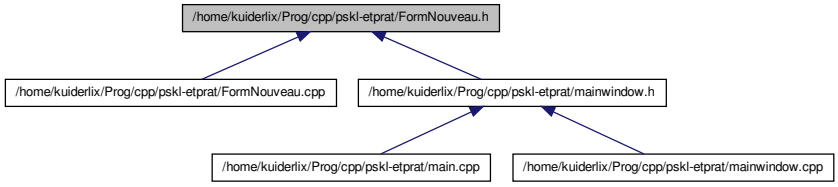
5.88 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormNouveau.h

`#include <QtWidgets>` Graphe des dépendances par inclusion de Form-

Nouveau.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



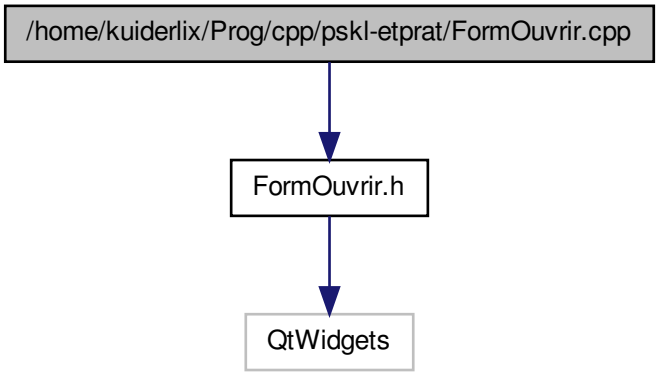
Classes

– class **FormNouveau**
Cette classe est le formulaire pour ajouter une nouvelle simulation.

5.89 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormOuvrir.cpp

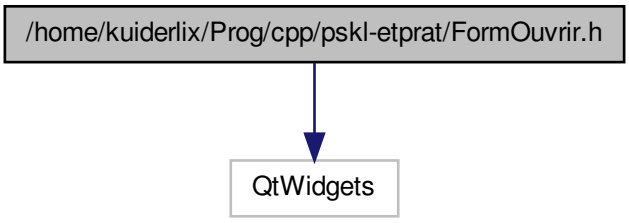
`#include "FormOuvrir.h"` Graphe des dépendances par inclusion de Form-

Ouvrir.cpp :

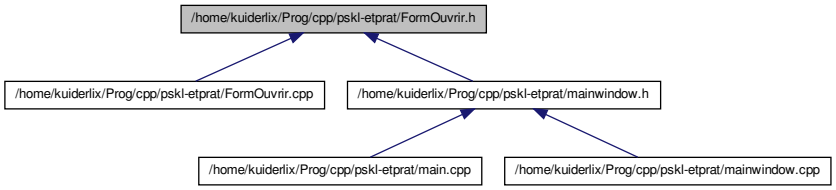


5.90 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormOuvrir.h

`#include <QtWidgets>` Graphe des dépendances par inclusion de FormOuvrir.-h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

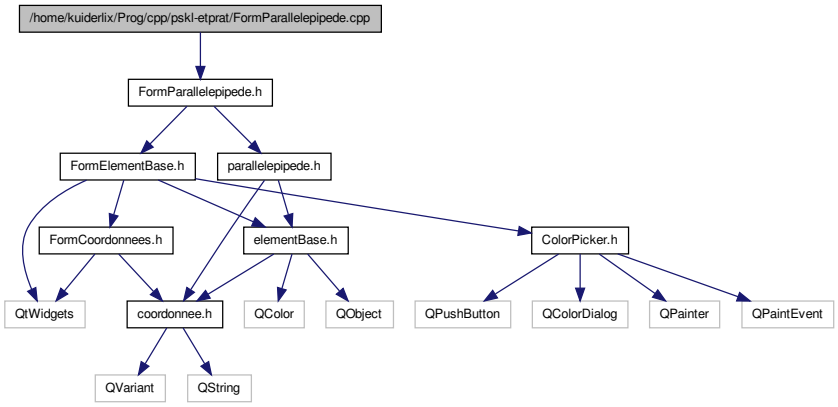


Classes

– class **FormOuvrir**
Cette classe est le formulaire pour ouvrir une nouvelle simulation.

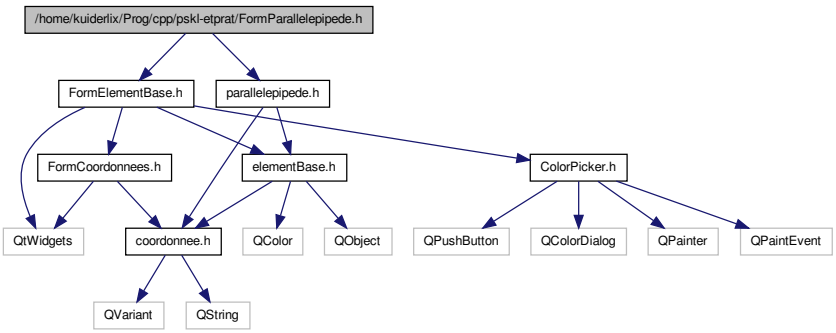
5.91 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormParallelepipede.cpp

`#include "FormParallelepipede.h"` Graphe des dépendances par inclusion de FormParallelepipede.cpp :

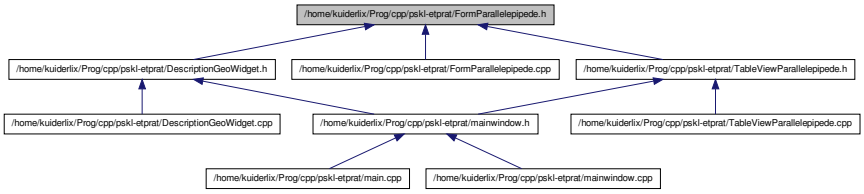


5.92 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormParallelepipede.h

`#include "FormElementBase.h"` `#include "parallelepipede.-h"` Graphe des dépendances par inclusion de FormParallelepipede.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



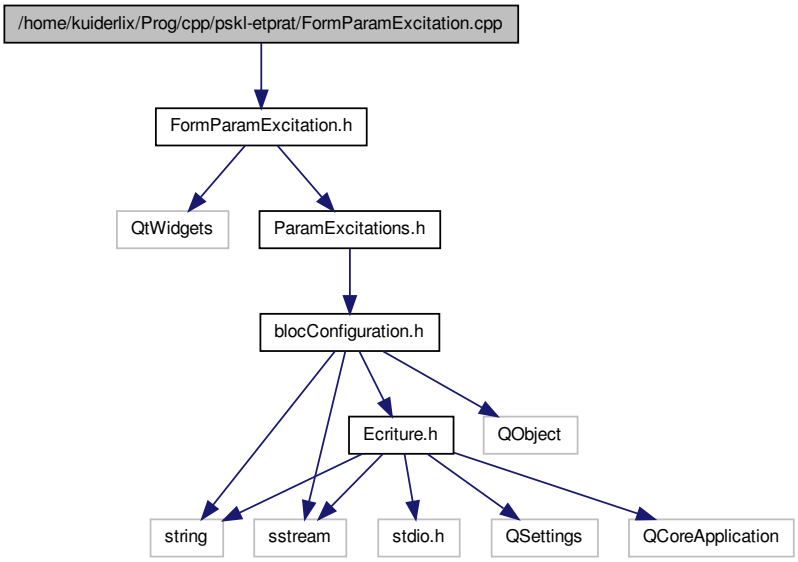
Classes

– class **FormParallelepipede**
Formulaire permettant d'ajouter un nouvelle objet parallelepipede dans la scène 3D.

5.93 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormParamExcitation.cpp

`#include "FormParamExcitation.h"` Graphe des dépendances par inclu-

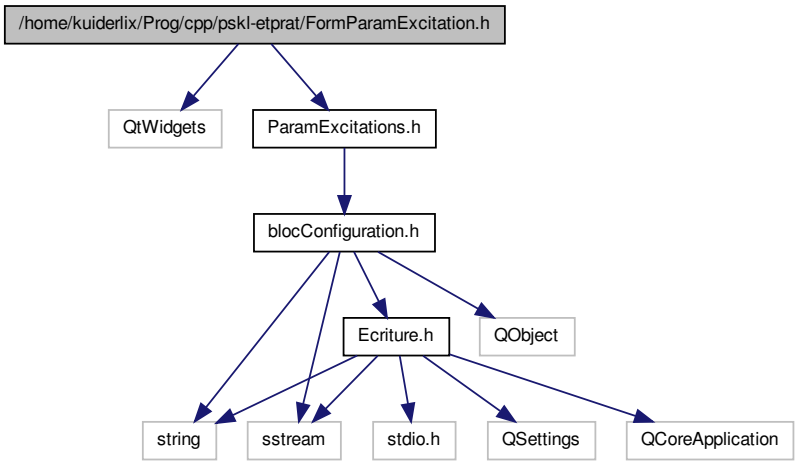
sion de FormParamExcitation.cpp :



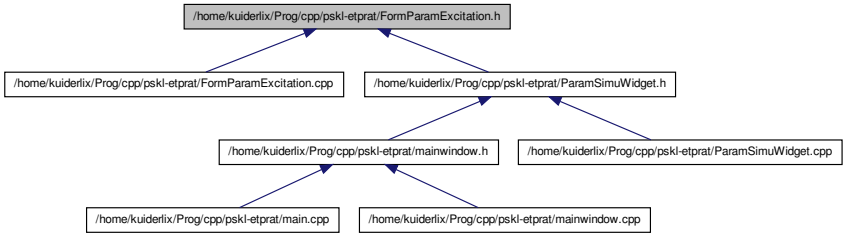
5.94 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormParamExcitation.h

`#include <QtWidgets>` `#include "ParamExcitations.h"` Graphe

des dépendances par inclusion de FormParamExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



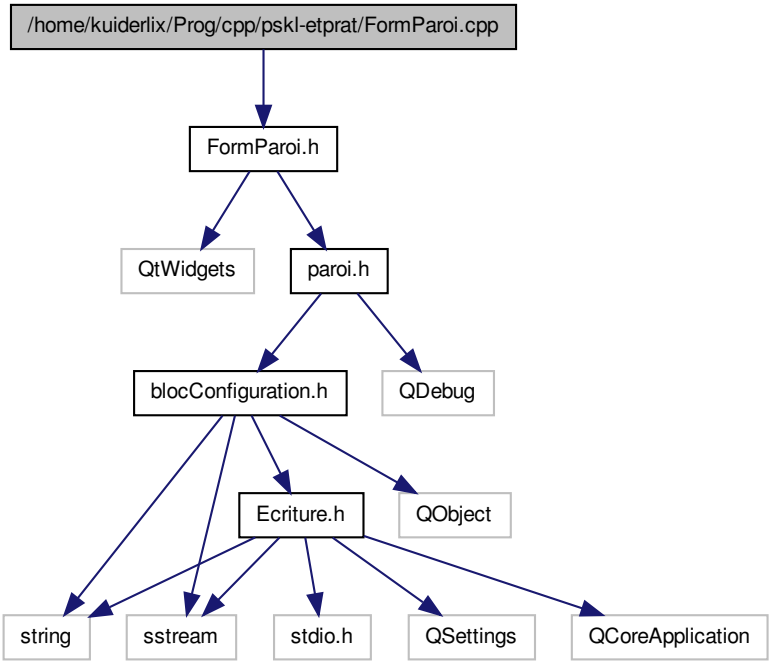
Classes

– class **FormParamExcitation**
Cette classe fournit un formulaire pour modifier l'objet ***ParamExcitations*** *(p. 135).*

5.95 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormParoi.cpp

`#include "FormParoi.h"` Graphe des dépendances par inclusion de Form-

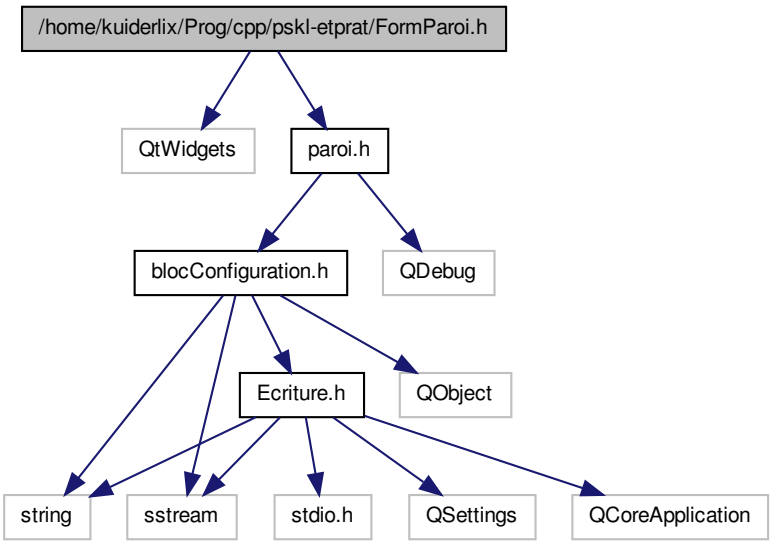
Paroi.cpp :



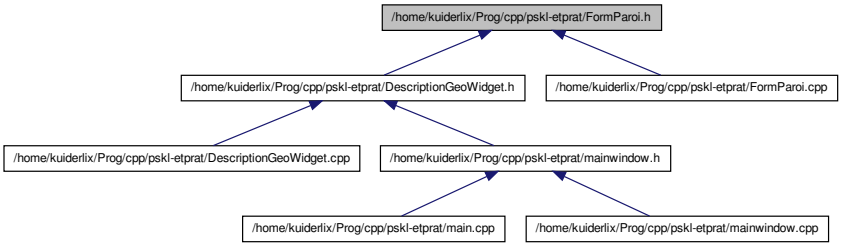
5.96 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormParoi.h

`#include <QtWidgets>` `#include "paroi.h"` Graphe des dépendances

par inclusion de FormParoi.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

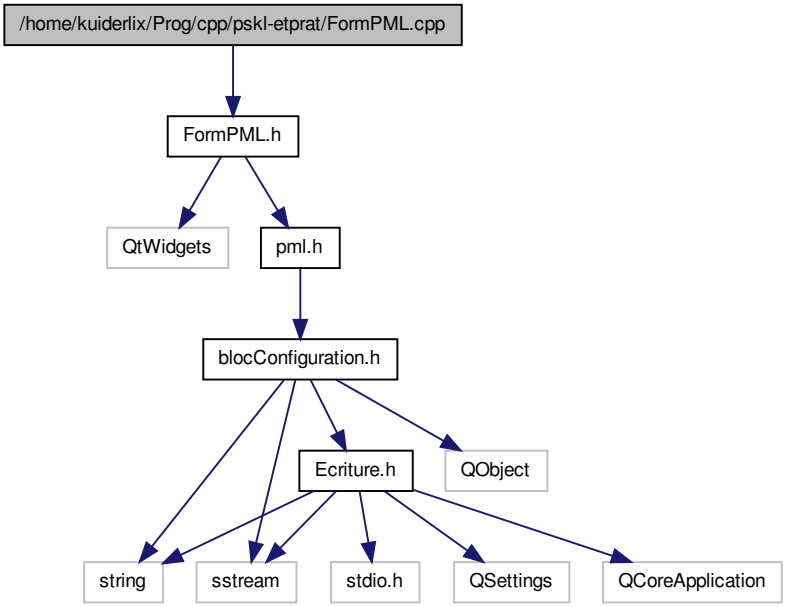


Classes

– class **FormParoi**
Cette classe fournit un formulaire pour modifier l'objet paroi.

5.97 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormPML.cpp

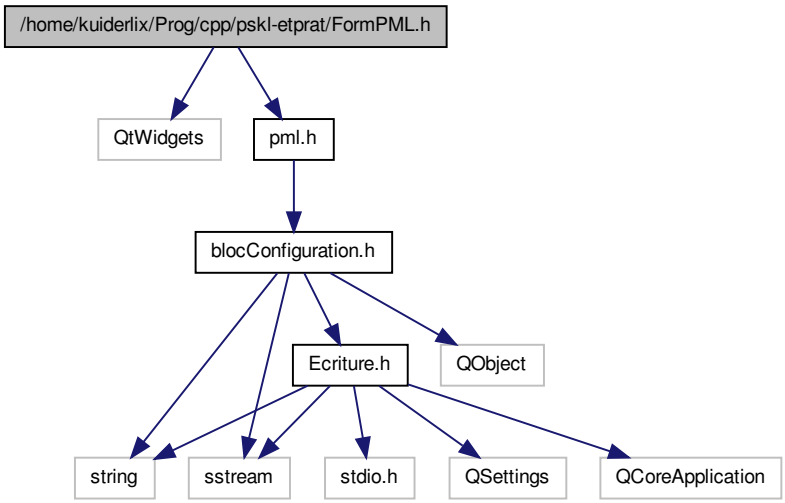
`#include "FormPML.h"` Graphe des dépendances par inclusion de FormPML.-cpp :



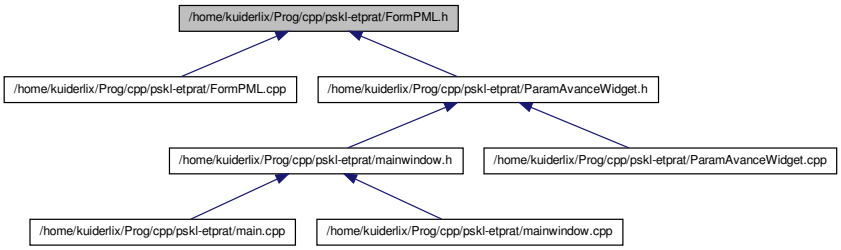
5.98 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormPML.h

`#include <QtWidgets>` `#include "pml.h"` Graphe des dépendances par

inclusion de FormPML.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



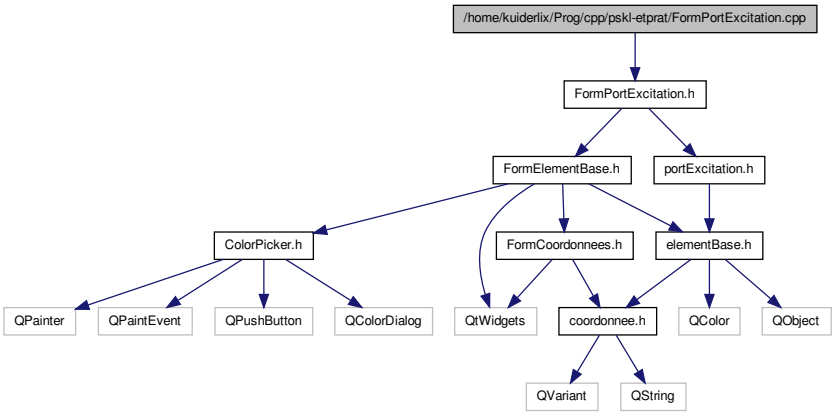
Classes

– class **FormPML**

Cette classe fournit un formulaire pour modifier l'objet pml.

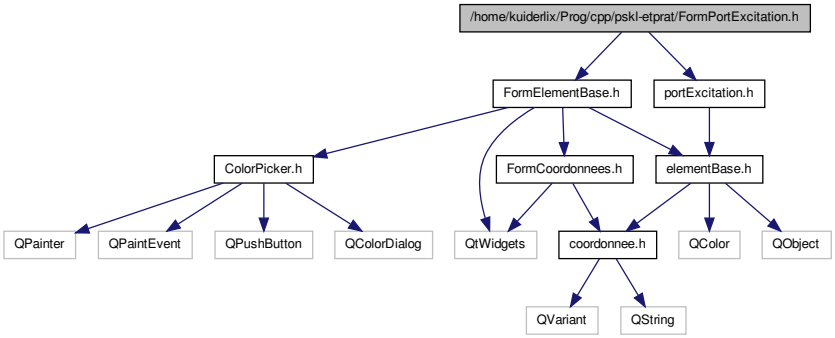
5.99 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormPortExcitation.cpp

`#include "FormPortExcitation.h"` Graphe des dépendances par inclusion de FormPortExcitation.cpp :

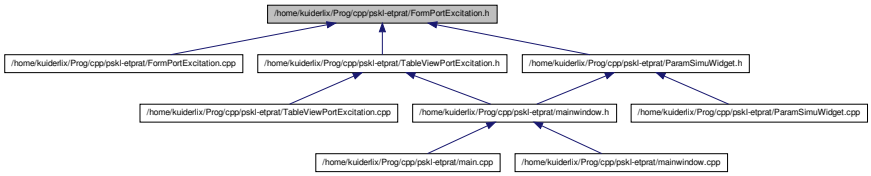


5.100 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormPortExcitation.h

`#include "FormElementBase.h"` `#include "portExcitation.h"` Graphe des dépendances par inclusion de FormPortExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

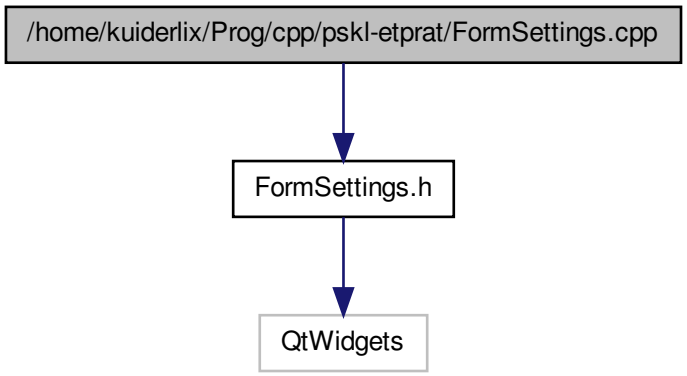


Classes

– class **FormPortExcitation**
*Formulaire permettant d'ajouter un nouvelle objet **portExcitation*** (p. *147) dans la scène 3D.*

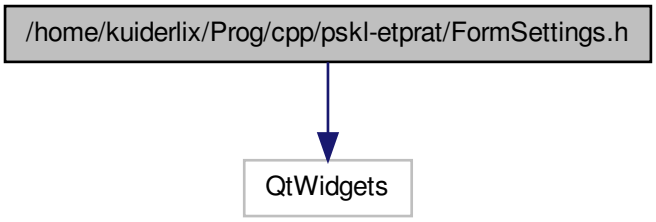
5.101 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormSettings.cpp

`#include "FormSettings.h"` Graphe des dépendances par inclusion de Form-Settings.cpp :

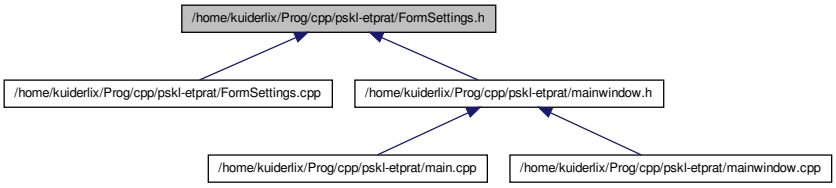


5.102 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormSettings.h

`#include <QtWidgets>` Graphe des dépendances par inclusion de Form-Settings.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



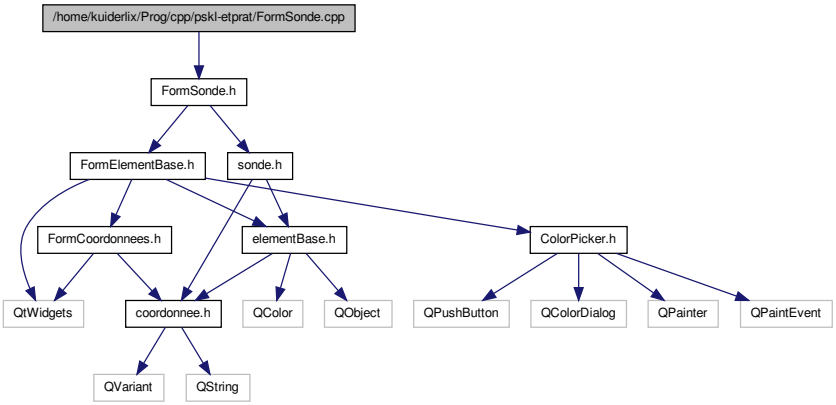
Classes

– class **FormSettings**
Cette classe est le formulaire pour modifier les paramètres du logiciel.

5.103 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormSonde.cpp

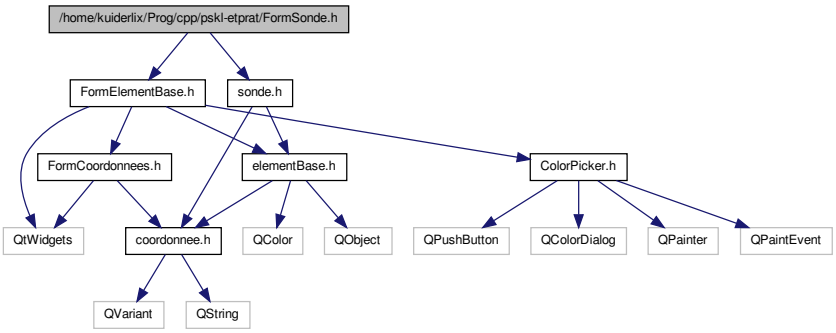
`#include "FormSonde.h"` Graphe des dépendances par inclusion de Form-

Sonde.cpp :

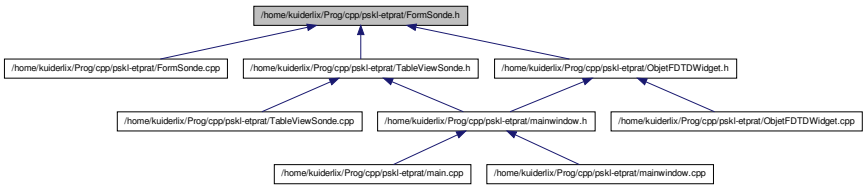


5.104 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormSonde.h

`#include "FormElementBase.h"` `#include "sonde.h"` Graphe des dépendances par inclusion de FormSonde.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

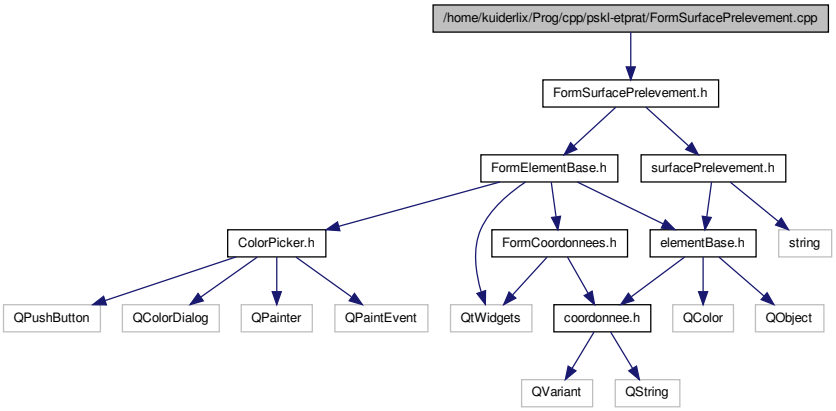


Classes

– class **FormSonde**
Formulaire permettant d'ajouter un nouvelle objet sonde dans la scène 3D.

5.105 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormSurfacePrelevement.cpp

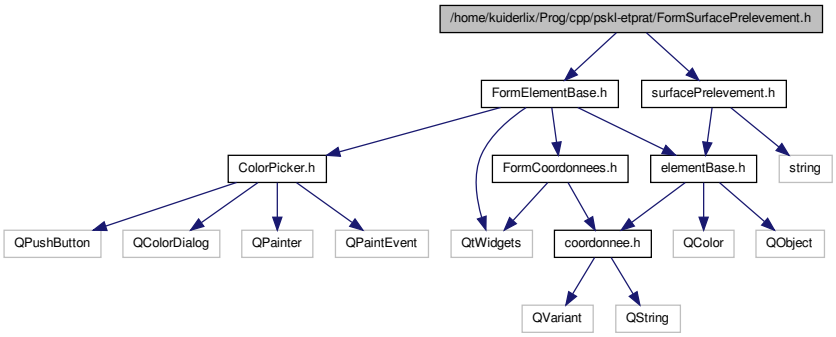
`#include "FormSurfacePrelevement.h"` Graphe des dépendances par inclusion de FormSurfacePrelevement.cpp :



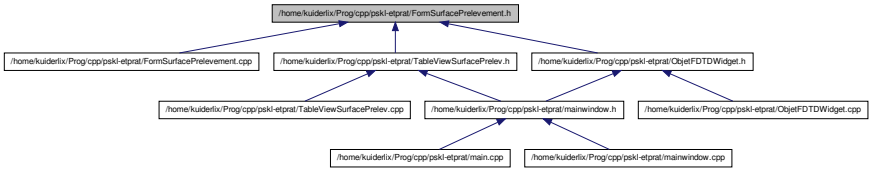
5.106 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormSurfacePrelevement.h

`#include "FormElementBase.h"` `#include "surfacePrelevement.-`

`h"` Graphe des dépendances par inclusion de FormSurfacePrelevement.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

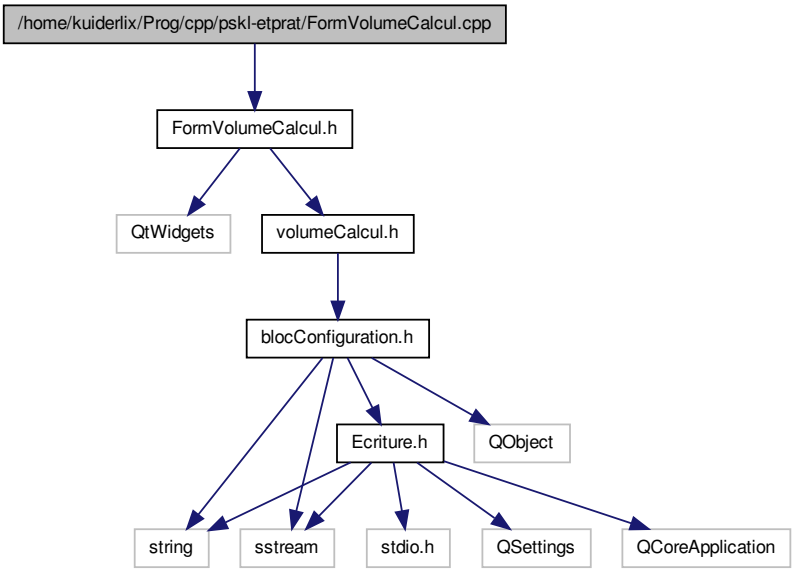
– class **FormSurfacePrelevement**

Formulaire permettant d'ajouter un nouvelle objet ***surfacePrelevement*** *(p. 154) dans la scène 3D.*

5.107 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormVolumeCalcul.cpp

`#include "FormVolumeCalcul.h"` Graphe des dépendances par inclusion de

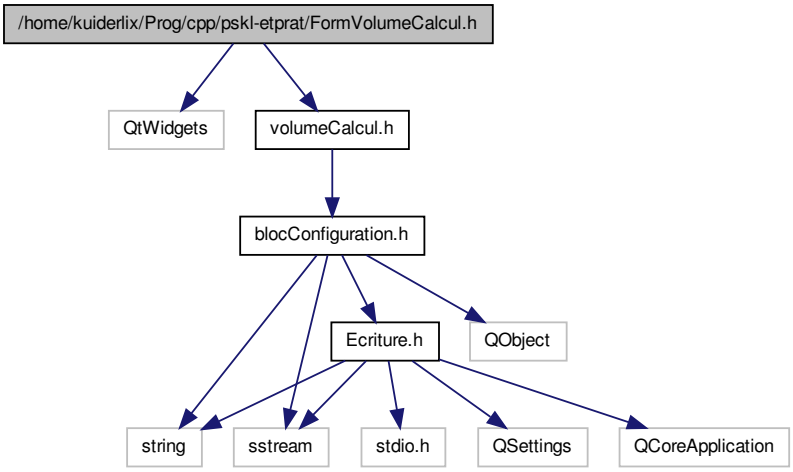
FormVolumeCalcul.cpp :



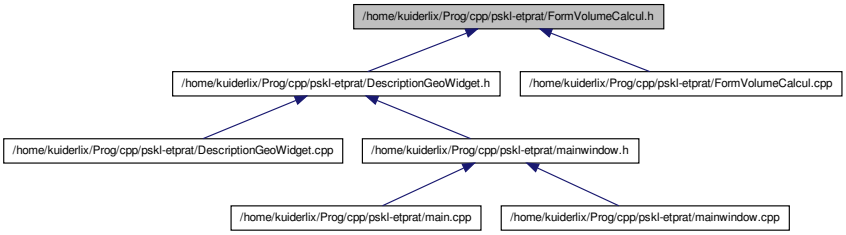
5.108 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-FormVolumeCalcul.h

`#include <QtWidgets>` `#include "volumeCalcul.h"` Graphe des

dépendances par inclusion de FormVolumeCalcul.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



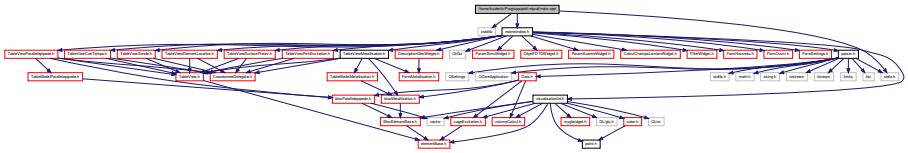
Classes

– class **FormVolumeCalcul**
Cette classe fournit un formulaire pour modifier l'objet ***volumeCalcul*** *(p. 201).*

5.109 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/main.cpp

```
#include <cstdlib> #include <stdio.h> #include "mainwindow.-
```

`h"` Graphe des dépendances par inclusion de main.cpp :



Fonctions

– int **main** (int argc, char ∗∗argv)

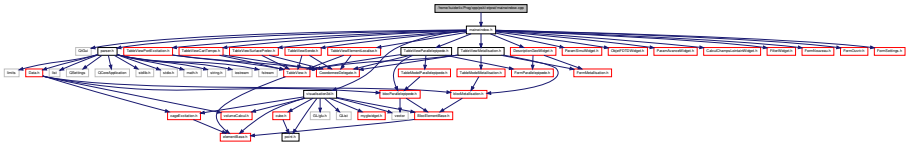
5.109.1 Documentation des fonctions

5.109.1.1 int main (int *argc*, char ∗∗ *argv*)

Définition à la ligne 19 du fichier main.cpp.

5.110 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/mainwindow.cpp

`#include "mainwindow.h"` Graphe des dépendances par inclusion de mainwindow.cpp :

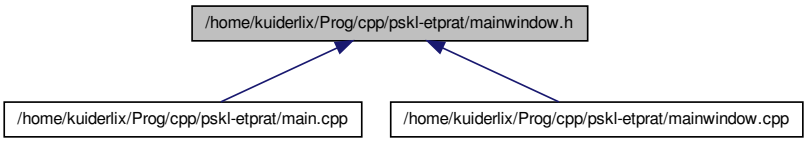


5.111 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/mainwindow.h

```
#include <QtGui> #include "parser.h" #include "visualisation3d.-
h"  #include "TableViewMetallisation.h"  #include "Table-
ViewParallelepipede.h" #include "TableViewSonde.h" #include
"TableViewElementLocalise.h"     #include "TableViewPort-
Excitation.h" #include "TableViewCartTempo.h" #include "-
TableViewSurfacePrelev.h" #include "DescriptionGeoWidget.-
h" #include "ParamSimuWidget.h" #include "ObjetFDTDWidget.-
h" #include "ParamAvanceWidget.h" #include "CalculChamps-
LointainWidget.h"  #include "FilterWidget.h"  #include "-
```

`FormNouveau.h"` `#include "FormOuvrir.h"` `#include "FormSettings.h"` Graphe des dépendances par inclusion de mainwindow.h :

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

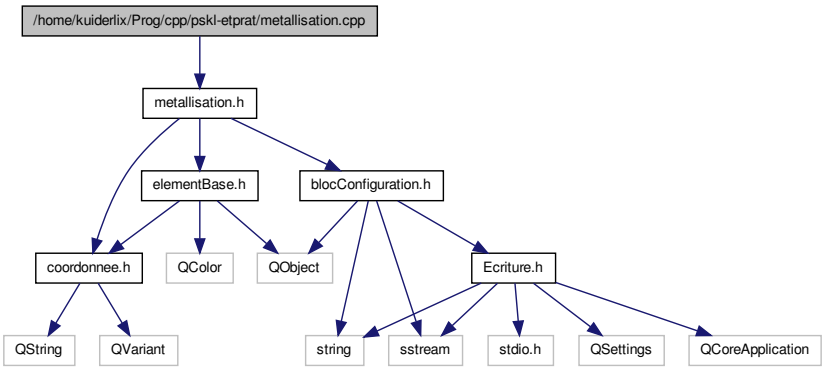


Classes

– class **MainWindow**
*The **MainWindow*** (p. *121) class Cette classe représente la fenêtre principale du logiciel.*

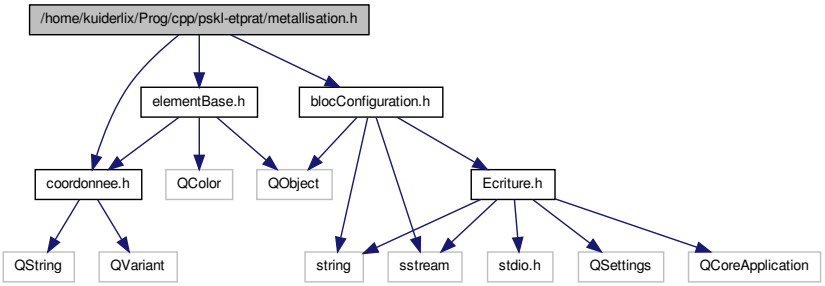
5.112 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/metallisation.cpp

`#include "metallisation.h"` Graphe des dépendances par inclusion de metallisation.cpp :

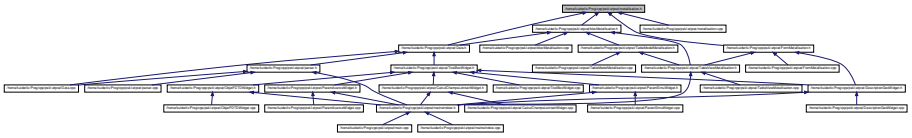


5.113 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/metallisation.h

`#include "coordonnee.h"` `#include "elementBase.h"` `#include "blocConfiguration.h"` Graphe des dépendances par inclusion de metallisation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



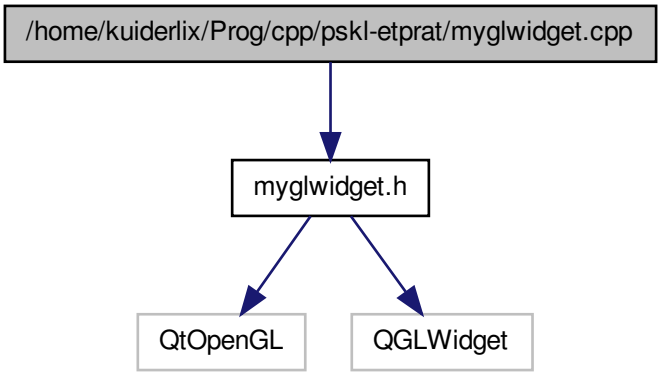
Classes

– class **metallisation**
Cette classe représente la variable metallisation d'une simulation.

5.114 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/myglwidget.cpp

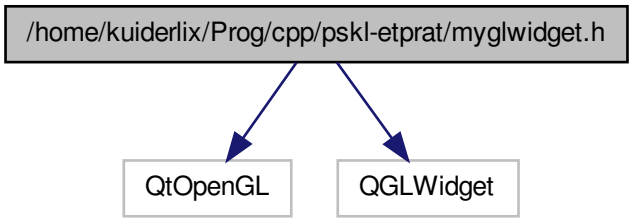
`#include "myglwidget.h"` Graphe des dépendances par inclusion de

myglwidget.cpp :

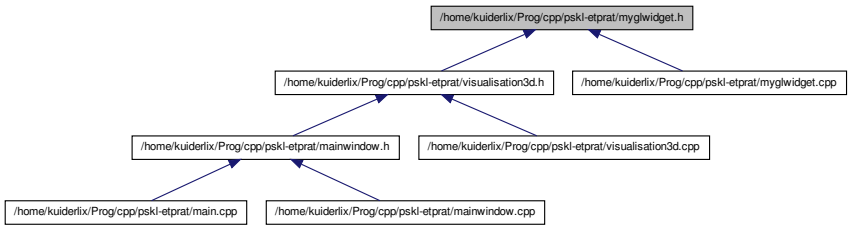


5.115 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/myglwidget.h

`#include <QtOpenGL>` `#include <QGLWidget>` Graphe des dépendances par inclusion de myglwidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

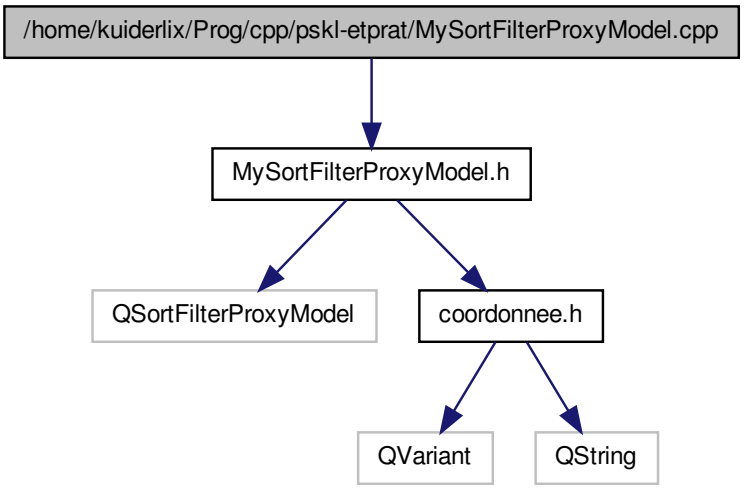


Classes

– class **MyGLWidget**
Cette classe sert de base à la classe qui affiche le rendu en 3D avec OpenGL.

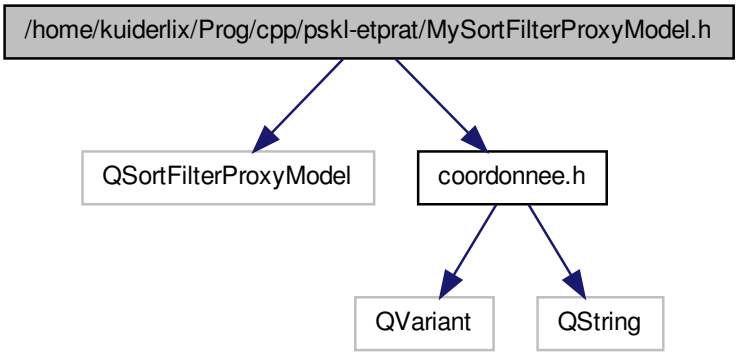
5.116 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-MySortFilterProxyModel.cpp

`#include "MySortFilterProxyModel.h"` Graphe des dépendances par inclusion de MySortFilterProxyModel.cpp :



5.117 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-MySortFilterProxyModel.h

`#include <QSortFilterProxyModel>` `#include "coordonnee.h"` ×
Graphe des dépendances par inclusion de MySortFilterProxyModel.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

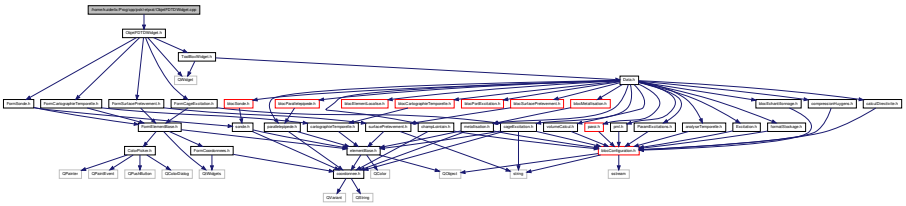
Classes

– class **MySortFilterProxyModel**
Cette classe permet de personnalisé le tri et le filtrage dans les listes.

5.118 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ObjetFDTDWidget.cpp

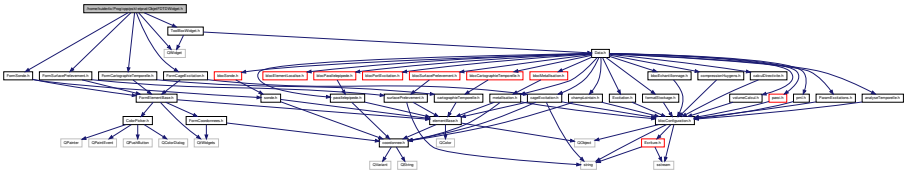
`#include "ObjetFDTDWidget.h"` Graphe des dépendances par inclusion de

ObjetFDTDWidget.cpp :

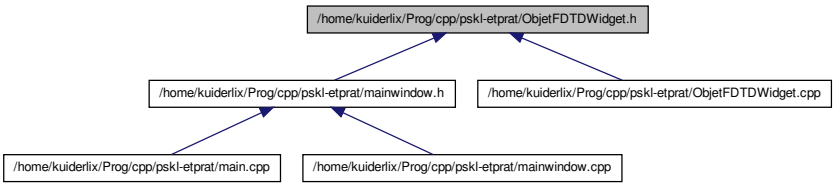


5.119 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ObjetFDTDWidget.h

`#include <QWidget>` `#include "FormSurfacePrelevement.h"` × `#include "FormCageExcitation.h"` `#include "FormSonde.h"` × `#include "FormCartographieTemporelle.h"` `#include "ToolBoxWidget.h"` Graphe des dépendances par inclusion de ObjetFDTDWidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

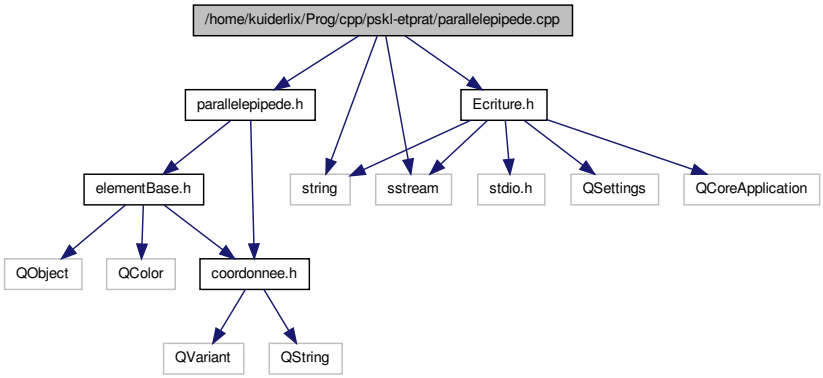


Classes

– class **ObjetFDTDWidget**
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

5.120 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parallelepipede.cpp

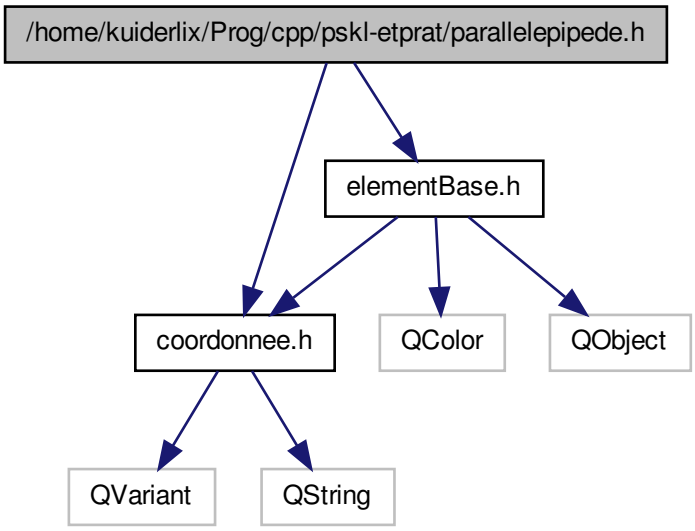
`#include "parallelepipede.h"` `#include <string>` `#include <sstream>` `#include "Ecriture.h"` Graphe des dépendances par inclusion de parallelepipede.cpp :



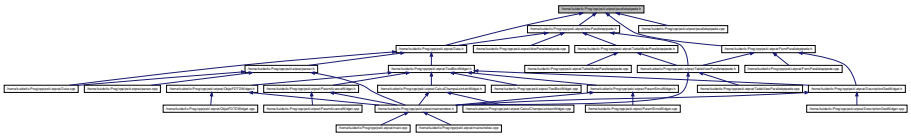
5.121 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parallelepipede.h

`#include "coordonnee.h"` `#include "elementBase.h"` Graphe des

dépendances par inclusion de parallelepipede.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



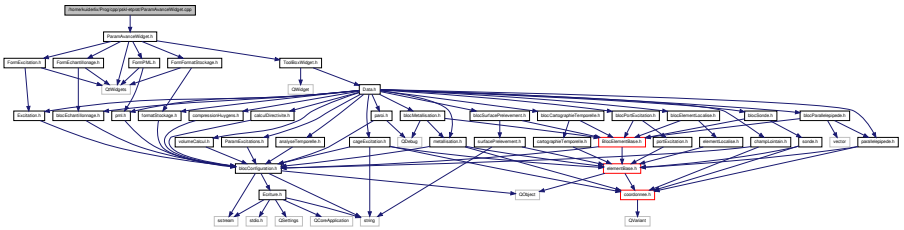
Classes

– class **parallelepipede**
Cette classe représente la variable parallelepipede d'une simulation.

5.122 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ParamAvanceWidget.cpp

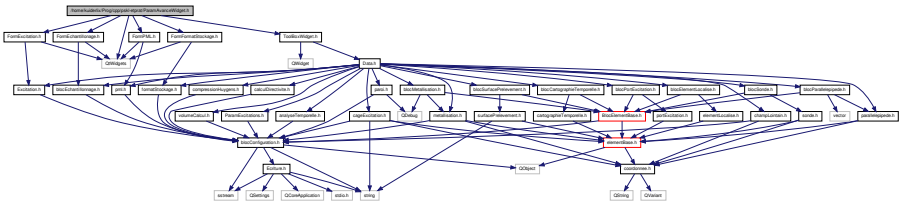
`#include "ParamAvanceWidget.h"` Graphe des dépendances par inclusion

de ParamAvanceWidget.cpp :

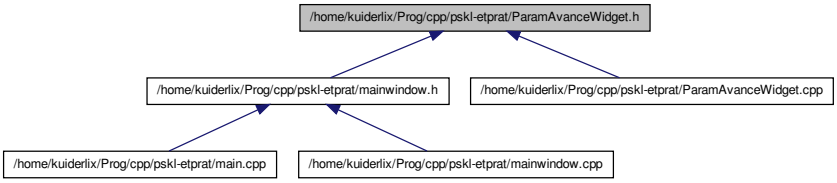


5.123 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ParamAvanceWidget.h

`#include <QtWidgets>` `#include "FormFormatStockage.h"` × `#include "FormExcitation.h"` `#include "FormEchantillonage.h"` `#include "FormPML.h"` `#include "ToolBoxWidget.h"` Graphe des dépendances par inclusion de ParamAvanceWidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



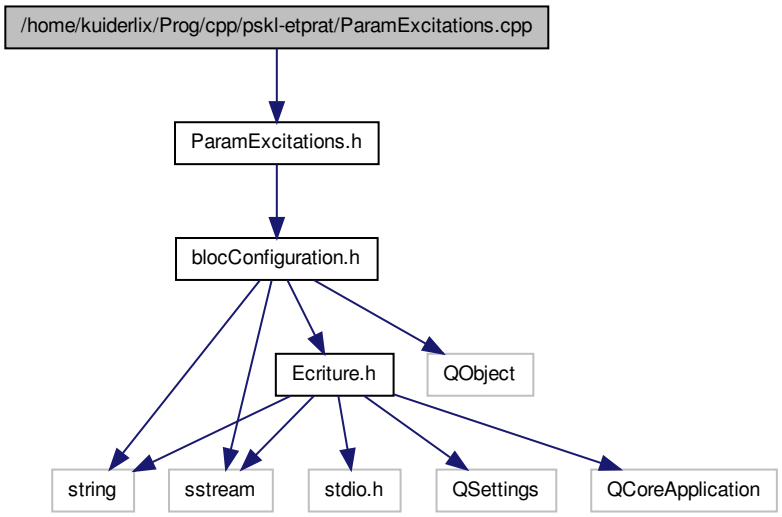
Classes

– class **ParamAvanceWidget**

Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

5.124 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ParamExcitations.cpp

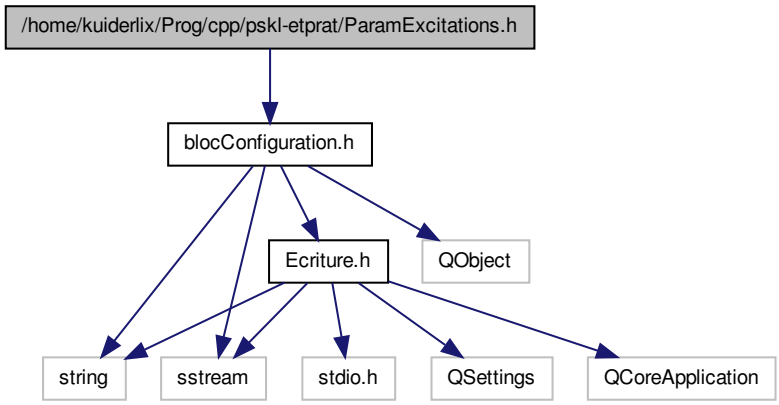
`#include "ParamExcitations.h"` Graphe des dépendances par inclusion de ParamExcitations.cpp :



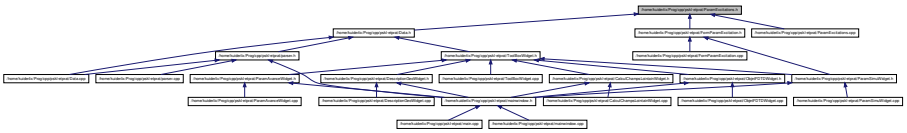
5.125 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ParamExcitations.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion

de ParamExcitations.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



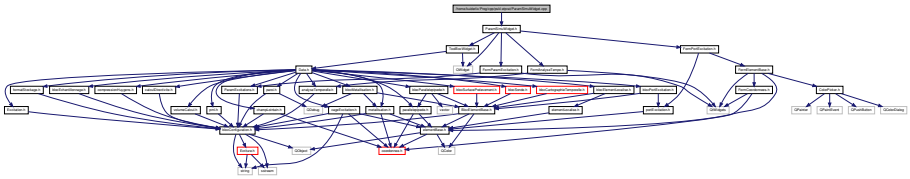
Classes

– class **ParamExcitations**
*Cette classe représente la variable **ParamExcitations*** (p. *135*) *d'une simulation.*

5.126 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ParamSimuWidget.cpp

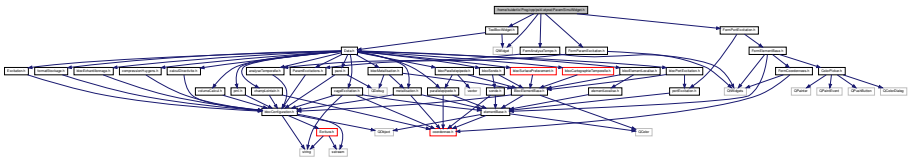
`#include "ParamSimuWidget.h"` Graphe des dépendances par inclusion de

ParamSimuWidget.cpp :

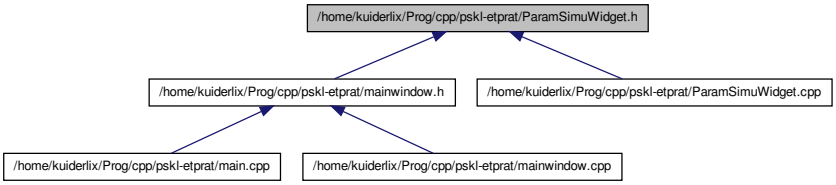


5.127 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ParamSimuWidget.h

`#include <QWidget>` `#include "FormPortExcitation.h"` `#include "FormAnalyseTempo.h"` `#include "FormParamExcitation.h"` × `#include "ToolBoxWidget.h"` Graphe des dépendances par inclusion de ParamSimuWidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

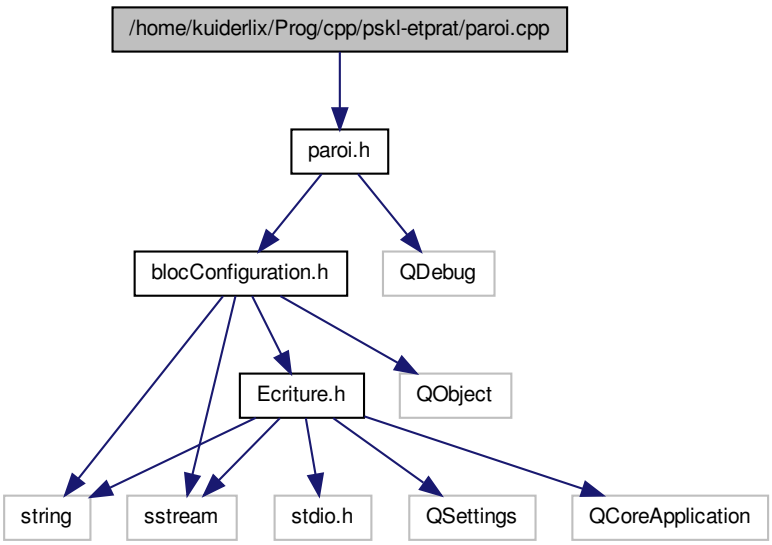


Classes

– class **ParamSimuWidget**
Cette classe représente un widget qui se trouve dans la toolbox. Il agit en faite comme un conteneur des autres formulaire de la meme catégorie.

5.128 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/paroi.cpp

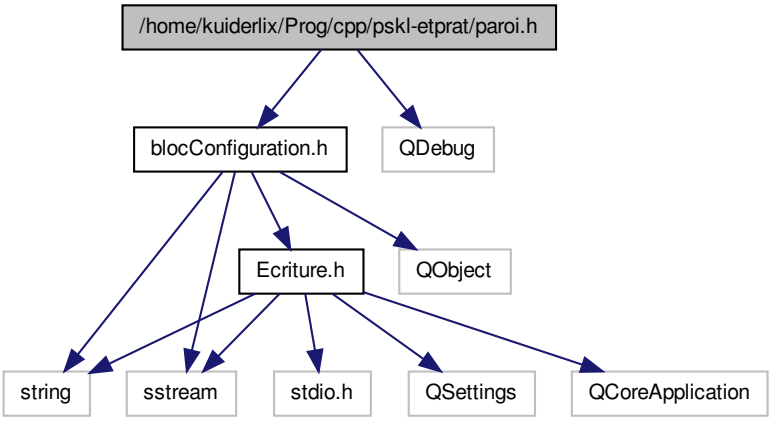
`#include "paroi.h"` Graphe des dépendances par inclusion de paroi.cpp :



5.129 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/paroi.h

`#include "blocConfiguration.h"` `#include <QDebug>` Graphe des

dépendances par inclusion de paroi.h :



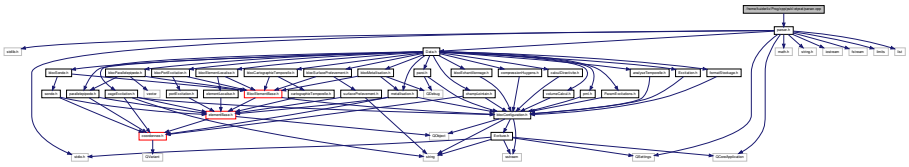
Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

– class **paroi**
Cette classe représente la variable paroi d'une simulation.

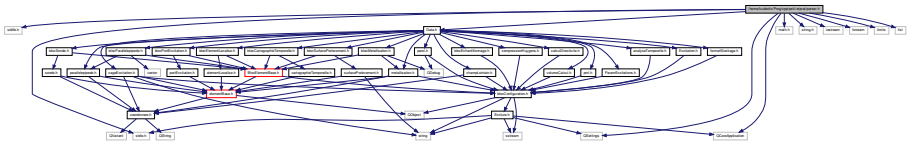
5.130 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parser.cpp

`#include "parser.h"` Graphe des dépendances par inclusion de parser.cpp :

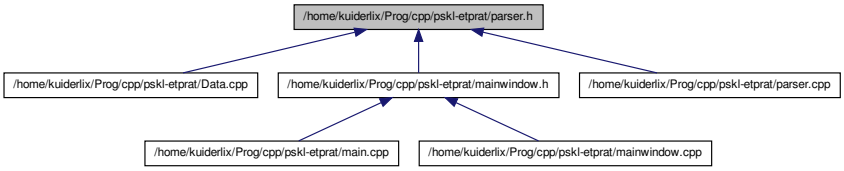


5.131 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/parser.h

`#include <stdlib.h>` `#include <stdio.h>` `#include <math.h>` `#include <string.h>` `#include <iostream>` `#include <fstream>` × `#include <limits>` `#include <list>` `#include <QSettings>` `#include <QCoreApplication>` `#include "Data.h"` Graphe des dépendances par inclusion de parser.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

– class **Parser**
  Le parser de fichier de configuration.

Macros

– #define **TAILLE_NOM_FICHIER** 10
– #define **TAILLE_CHAINE** 100

5.131.1 Documentation des macros

5.131.1.1 #define TAILLE_CHAINE 100

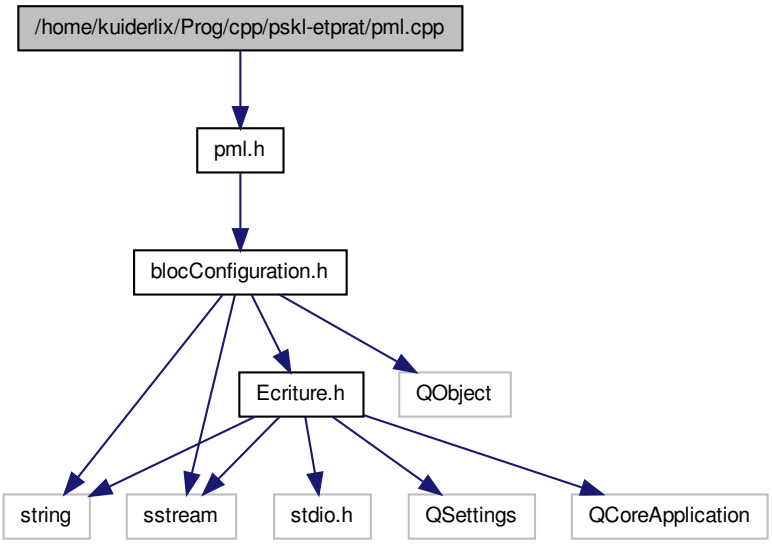
Définition à la ligne 26 du fichier parser.h.

5.131.1.2 #define TAILLE_NOM_FICHIER 10

Définition à la ligne 25 du fichier parser.h.

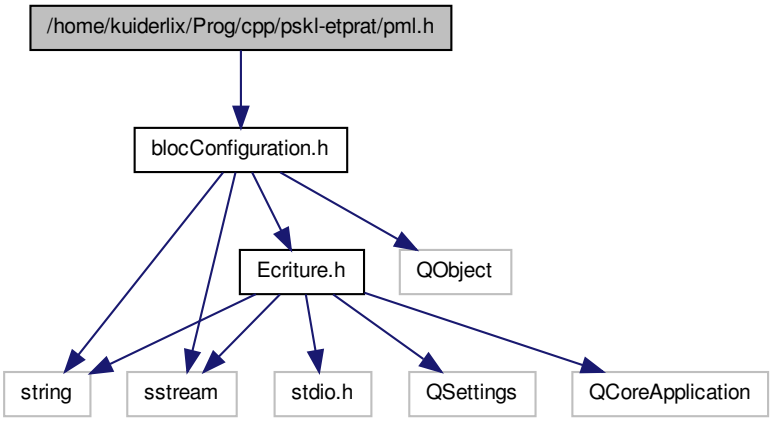
5.132 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/pml.cpp

`#include "pml.h"` Graphe des dépendances par inclusion de pml.cpp :

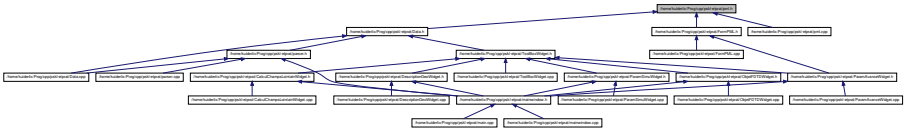


5.133 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/pml.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion de pml.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

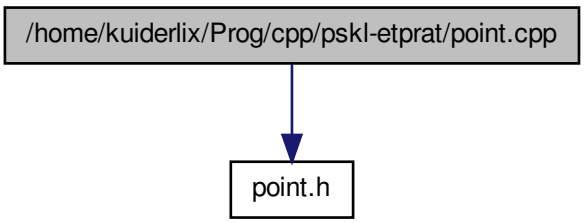


Classes

– class **pml**
Cette classe représente la variable pml d'une simulation.

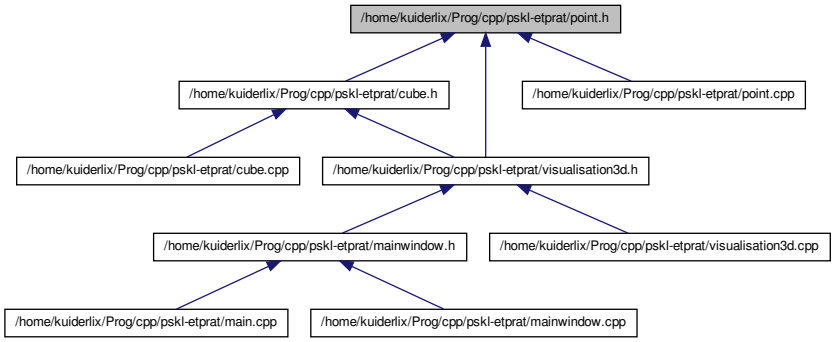
5.134 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/point.cpp

`#include "point.h"` Graphe des dépendances par inclusion de point.cpp :



5.135 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/point.h

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



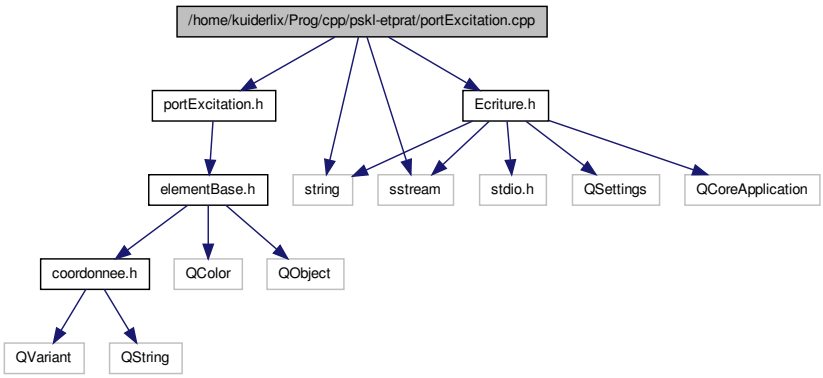
Classes

– class **Point**

Cette classe représente un point dans un espace en 3D. Elle est utilisé pour afficher les ***Cube*** *(p. 65) dans la vue 3D.*

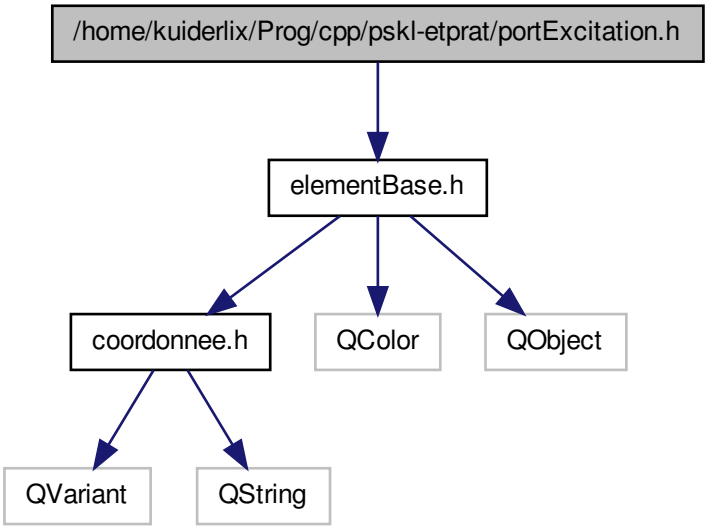
5.136 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/portExcitation.cpp

`#include "portExcitation.h"` `#include <string>` `#include <sstream>` `#include "Ecriture.h"` Graphe des dépendances par inclusion de portExcitation.cpp :



5.137 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/portExcitation.h

`#include "elementBase.h"` Graphe des dépendances par inclusion de portExcitation.h :



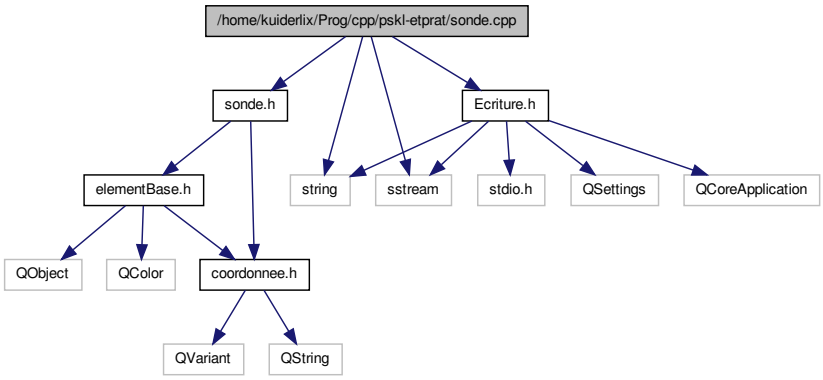
Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

Classes

– class **portExcitation**
*Cette classe représente la variable **portExcitation*** (p. *147*) *d'une simulation.*

5.138 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/sonde.cpp

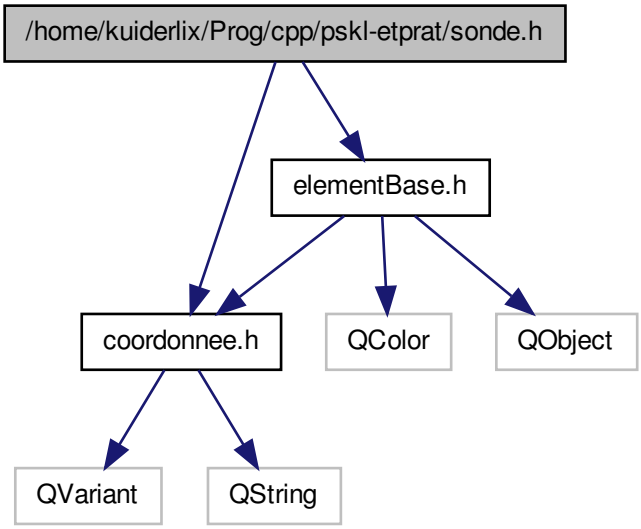
`#include "sonde.h"` `#include <string>` `#include <sstream>` × `#include "Ecriture.h"` Graphe des dépendances par inclusion de sonde.cpp :



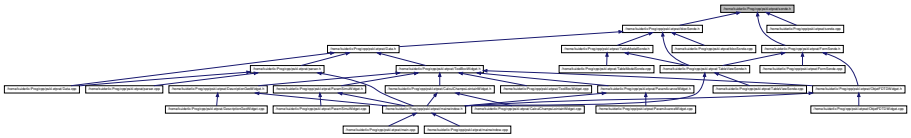
5.139 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/sonde.h

`#include "elementBase.h"` `#include "coordonnee.h"` Graphe des

dépendances par inclusion de sonde.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



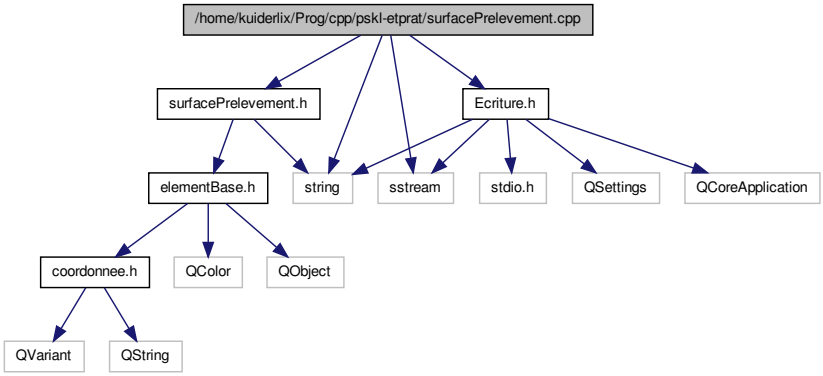
Classes

– class **sonde**
Cette classe représente la variable sonde d'une simulation.

5.140 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/surface-Prelevement.cpp

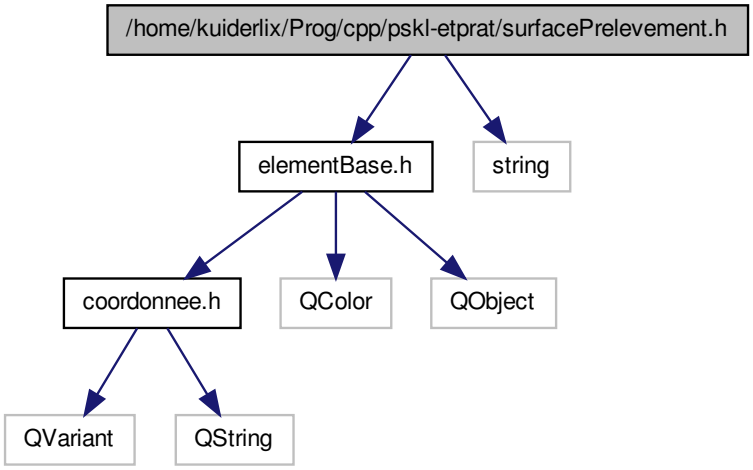
`#include "surfacePrelevement.h"` `#include <string>` `#include <sstream>` `#include "Ecriture.h"` Graphe des dépendances par inclusion

de surfacePrelevement.cpp :

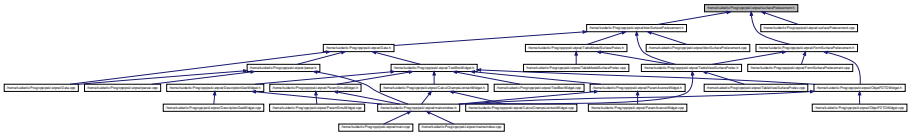


5.141 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/surface-Prelevement.h

`#include "elementBase.h"` `#include <string>` Graphe des dépendances par inclusion de surfacePrelevement.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

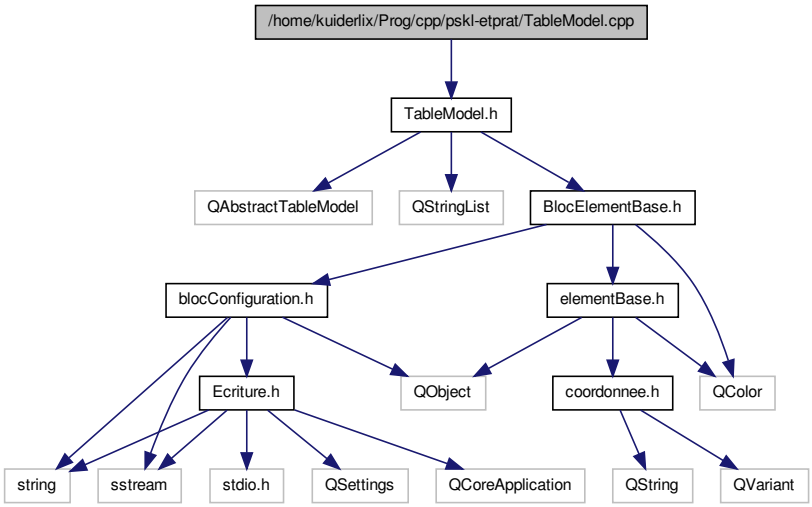


Classes

– class **surfacePrelevement**
*Cette classe représente la variable **surfacePrelevement** (p. 154) d'une simulation.*

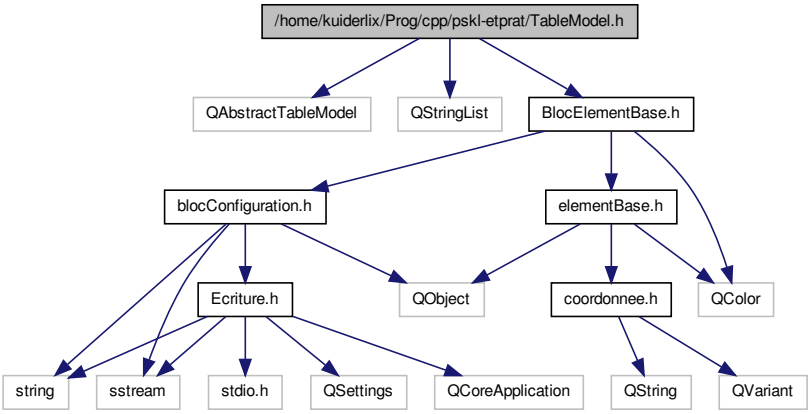
5.142 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModel.cpp

`#include "TableModel.h"` Graphe des dépendances par inclusion de TableModel.cpp :

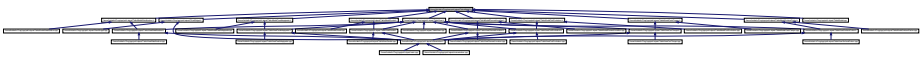


5.143 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModel.h

`#include <QAbstractTableModel>` `#include <QStringList>` × `#include "BlocElementBase.h"` Graphe des dépendances par inclusion de TableModel.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



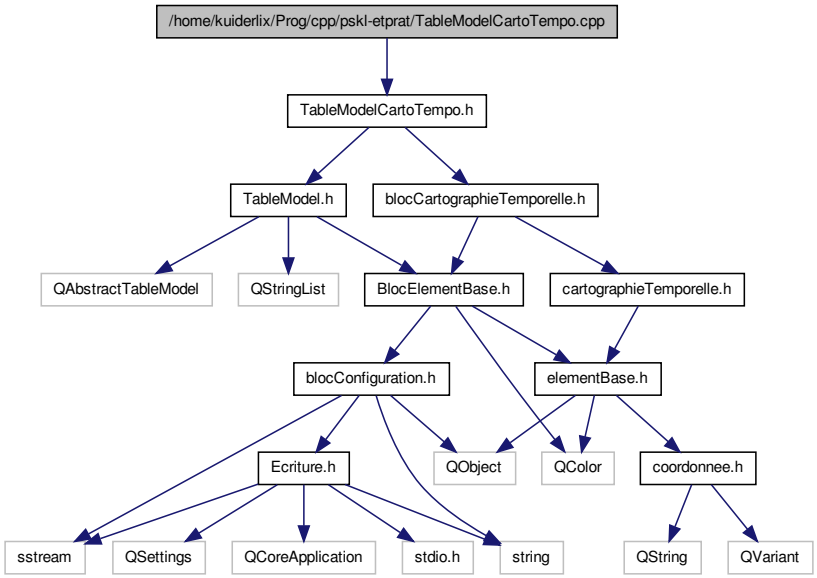
Classes

– class **TableModel**
Cette classe sert de model de base pour les éléments des tableaux.

5.144 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelCartoTempo.cpp

`#include "TableModelCartoTempo.h"` Graphe des dépendances par inclu-

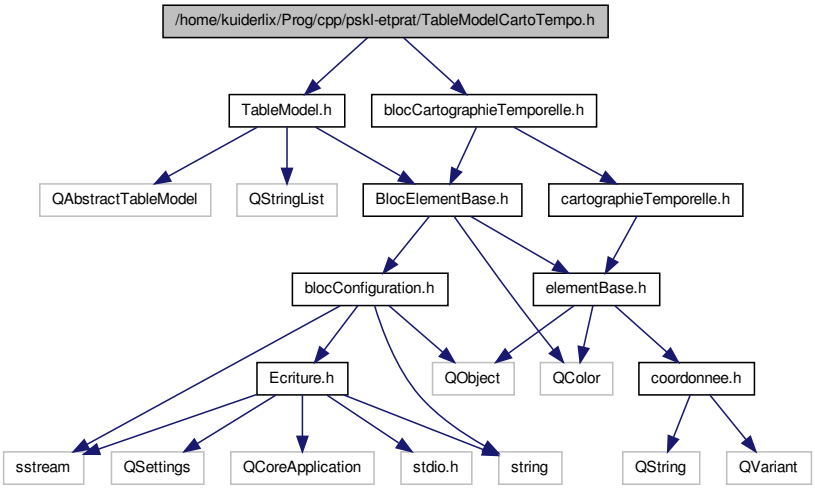
sion de TableModelCartoTempo.cpp :



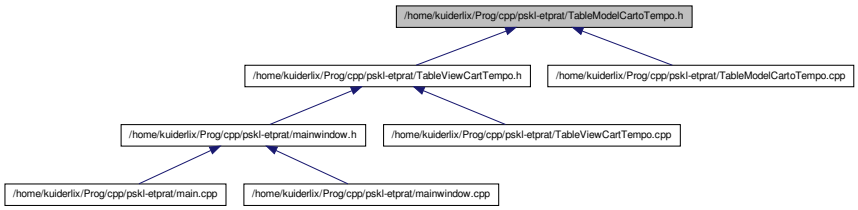
5.145 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelCartoTempo.h

```
#include "TableModel.h" #include "blocCartographieTemporelle.-
```

`h"` Graphe des dépendances par inclusion de TableModelCartoTempo.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



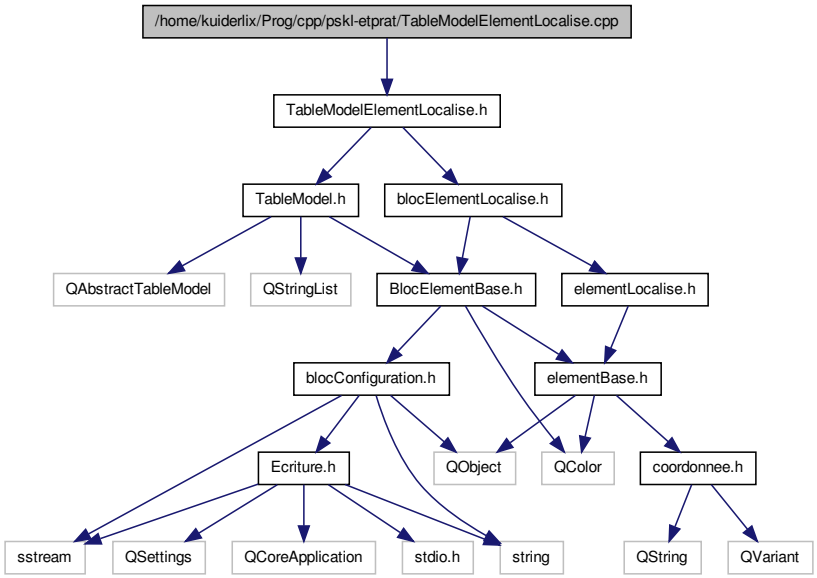
Classes

– class **TableModelCartoTempo**

5.146 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelElementLocalise.cpp

`#include "TableModelElementLocalise.h"` Graphe des dépendances

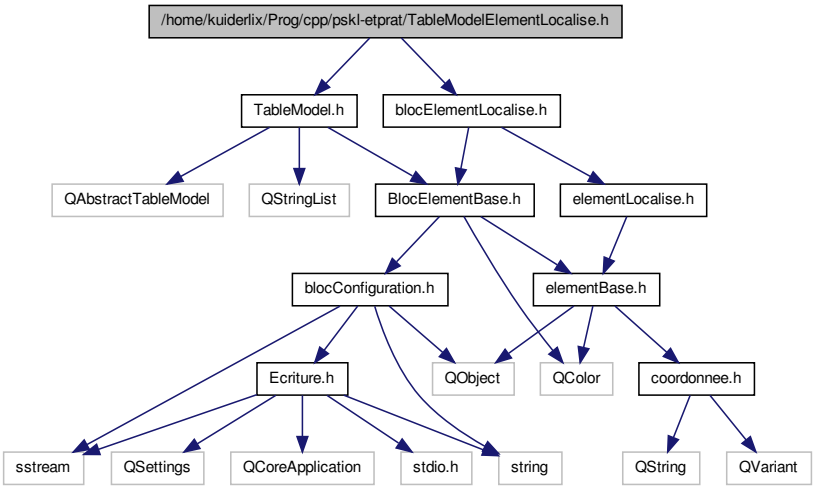
par inclusion de TableModelElementLocalise.cpp :



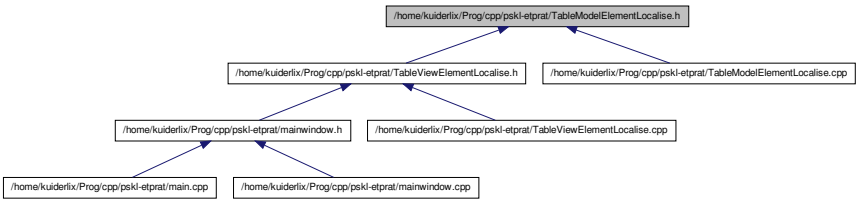
5.147 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelElementLocalise.h

```
#include "TableModel.h"   #include "blocElementLocalise.h"
```

Graphe des dépendances par inclusion de TableModelElementLocalise.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



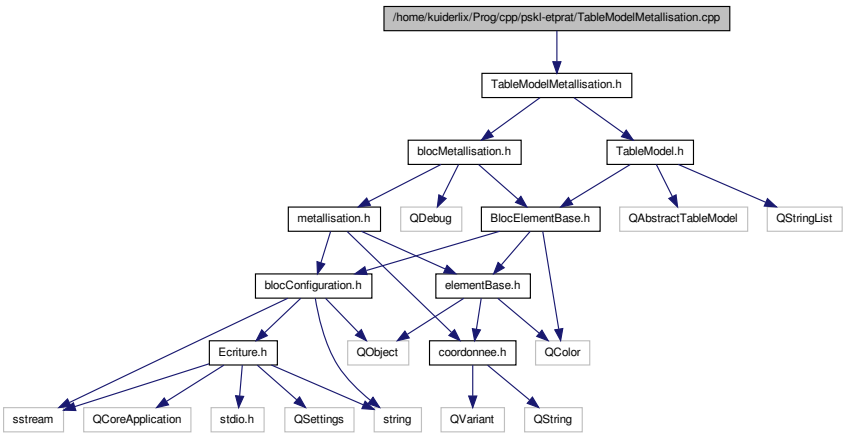
Classes

– class **TableModelElementLocalise**

5.148 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelMetallisation.cpp

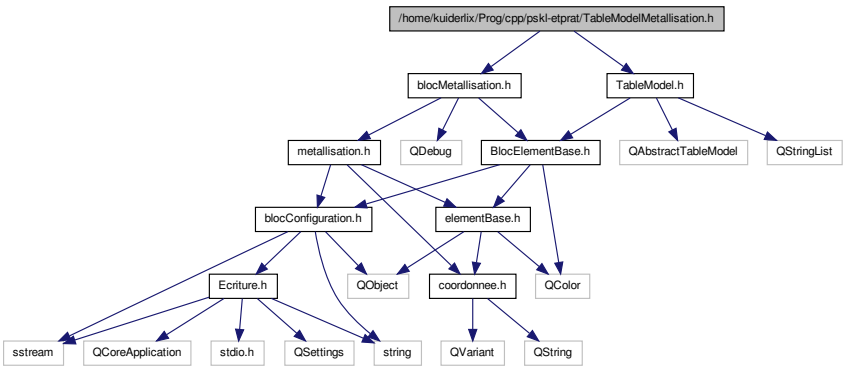
`#include "TableModelMetallisation.h"` Graphe des dépendances par

inclusion de TableModelMetallisation.cpp :

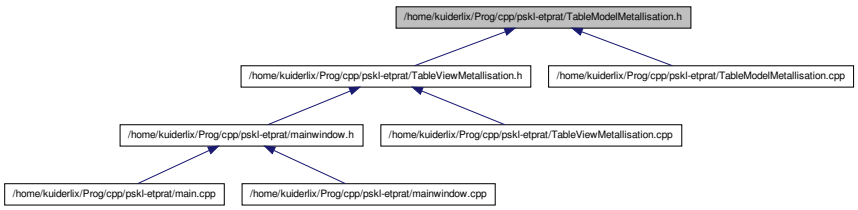


5.149 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelMetallisation.h

`#include "blocMetallisation.h"` `#include "TableModel.h"` ×
Graphe des dépendances par inclusion de TableModelMetallisation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

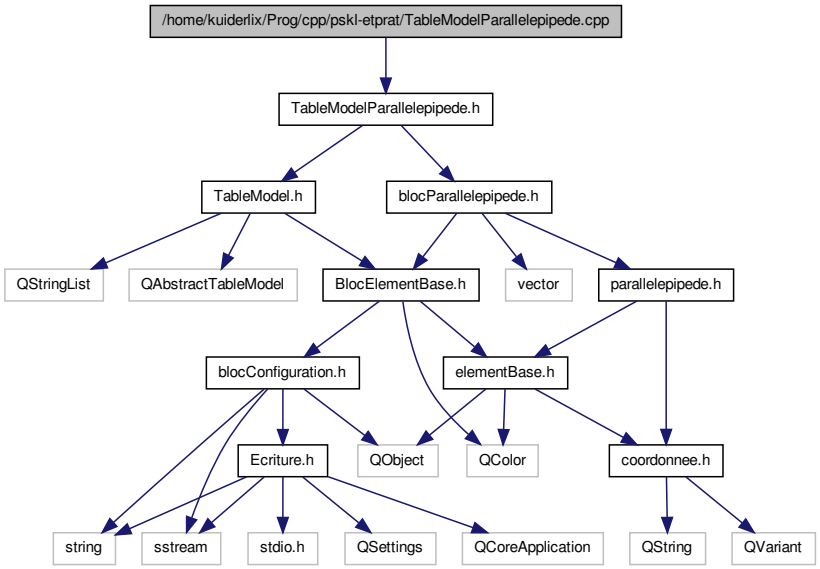


Classes

– class **TableModelMetallisation**

5.150 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelParallelepipede.cpp

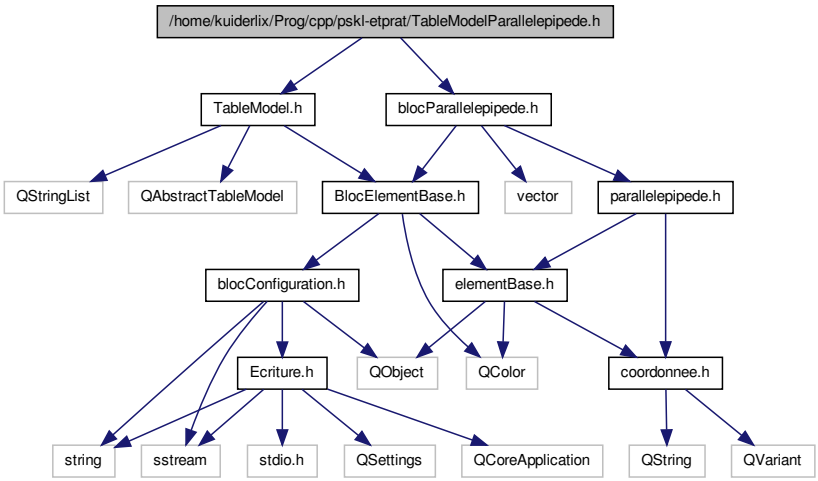
`#include "TableModelParallelepipede.h"` Graphe des dépendances par inclusion de TableModelParallelepipede.cpp :



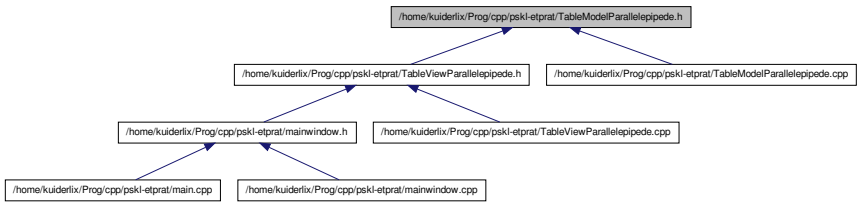
5.151 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelParallelepipede.h

`#include "TableModel.h"` `#include "blocParallelepipede.h"`

Graphe des dépendances par inclusion de TableModelParallelepipede.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

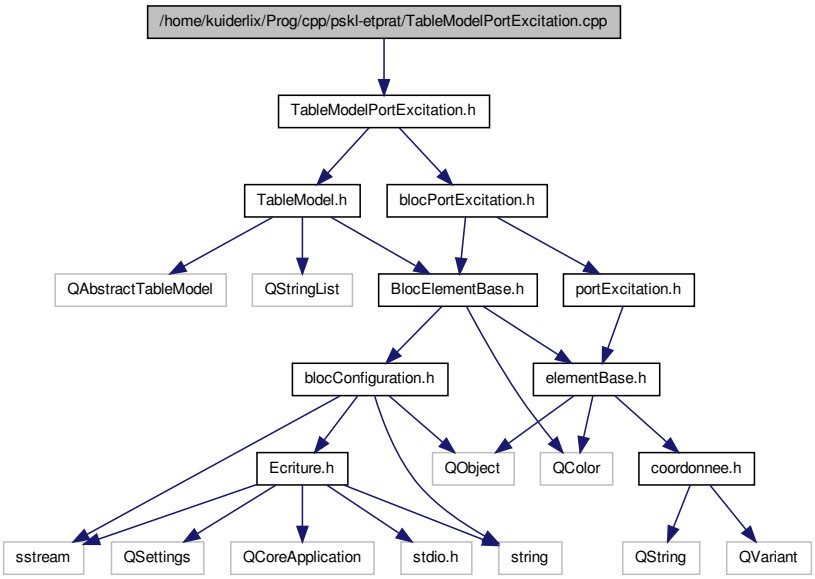


Classes

– class **TableModelParallelepipede**

5.152 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelPortExcitation.cpp

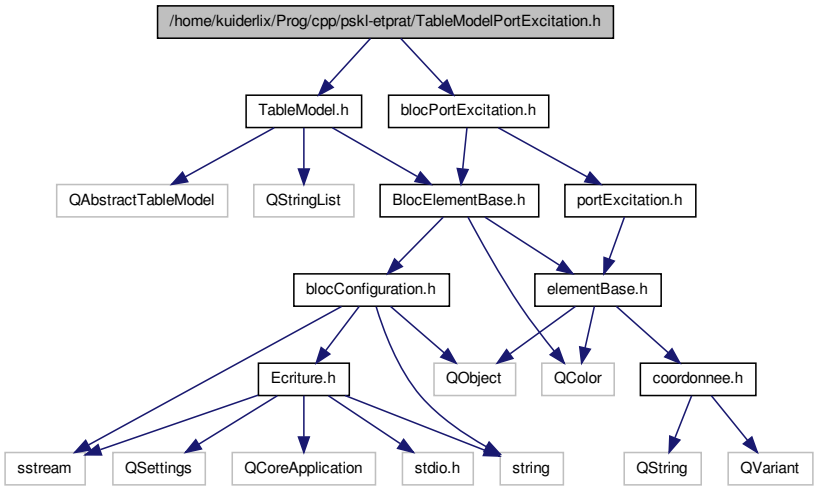
`#include "TableModelPortExcitation.h"` Graphe des dépendances par inclusion de TableModelPortExcitation.cpp :



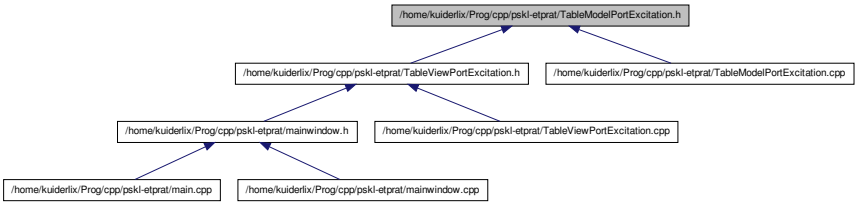
5.153 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelPortExcitation.h

`#include "TableModel.h"` `#include "blocPortExcitation.h"` ×

Graphe des dépendances par inclusion de TableModelPortExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



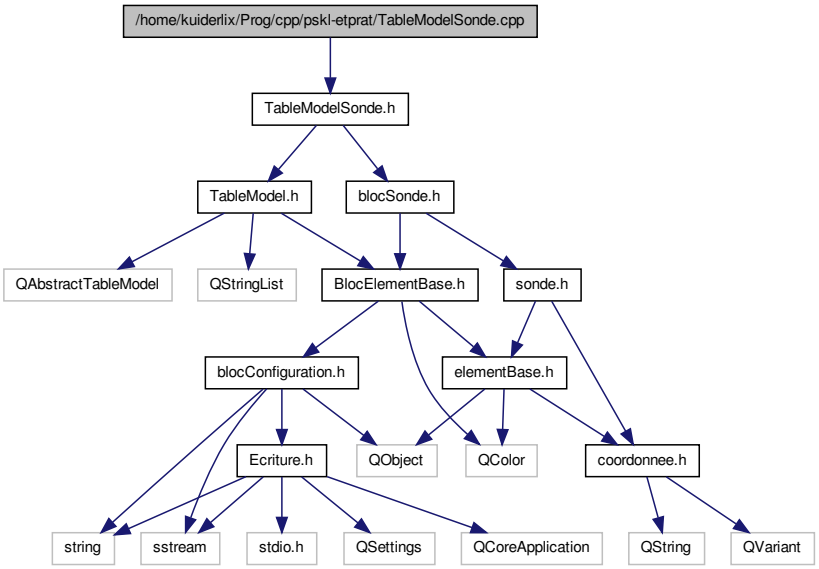
Classes

– class **TableModelPortExcitation**

5.154 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelSonde.cpp

`#include "TableModelSonde.h"` Graphe des dépendances par inclusion de

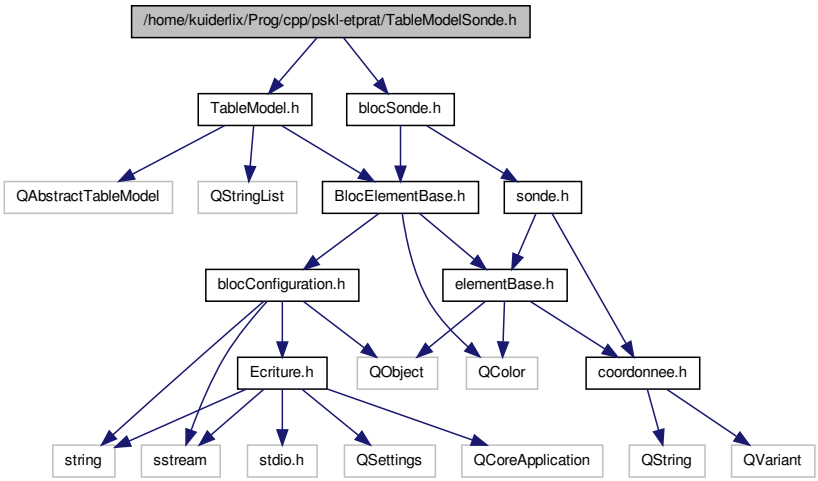
TableModelSonde.cpp :



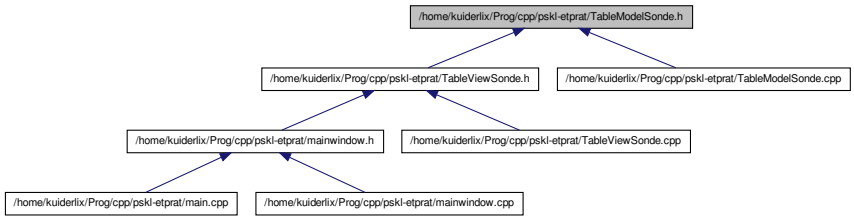
5.155 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelSonde.h

`#include "TableModel.h"` `#include "blocSonde.h"` Graphe des

dépendances par inclusion de TableModelSonde.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



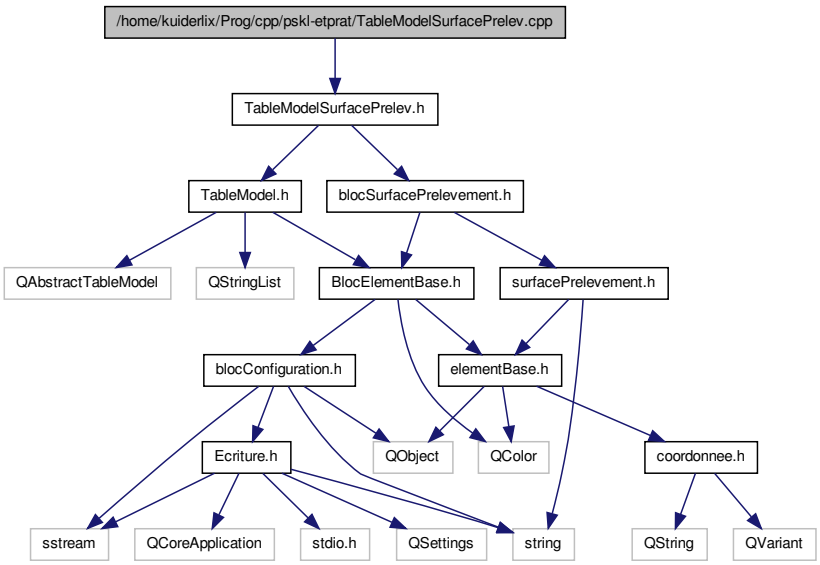
Classes

– class **TableModelSonde**

5.156 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelSurfacePrelev.cpp

`#include "TableModelSurfacePrelev.h"` Graphe des dépendances par

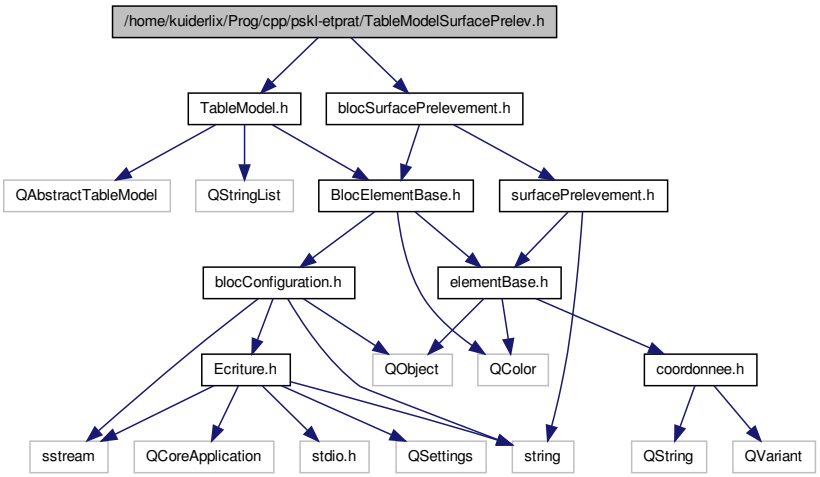
inclusion de TableModelSurfacePrelev.cpp :



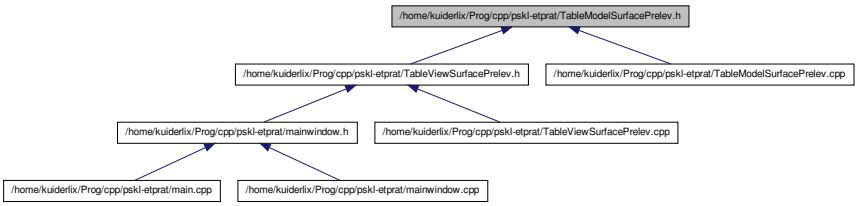
5.157 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableModelSurfacePrelev.h

```
#include "TableModel.h" #include "blocSurfacePrelevement.-
```

`h"` Graphe des dépendances par inclusion de TableModelSurfacePrelev.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



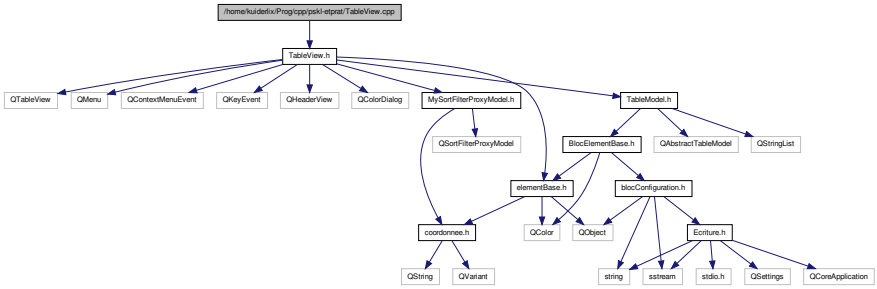
Classes

– class **TableModelSurfacePrelev**

5.158 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableView.cpp

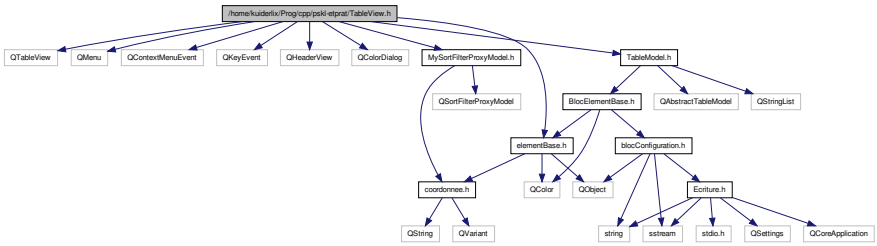
`#include "TableView.h"` Graphe des dépendances par inclusion de Table-

View.cpp :

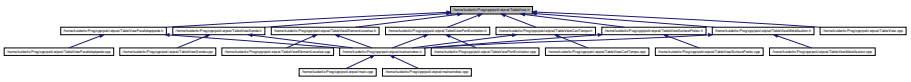


5.159 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableView.h

`#include <QTableView>` `#include <QMenu>` `#include <QContext-MenuEvent>` `#include <QKeyEvent>` `#include <QHeaderView>` × `#include <QColorDialog>` `#include "elementBase.h"` `#include "TableModel.h"` `#include "MySortFilterProxyModel.h"` Graphe des dépendances par inclusion de TableView.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



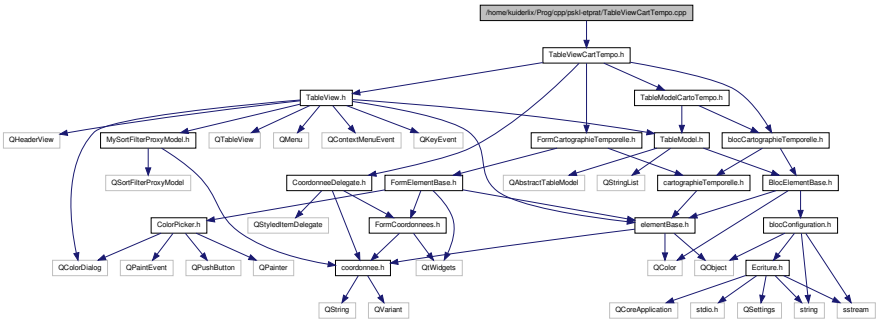
Classes

– class **TableView**

Cette classe sert de vue de base pour les tableaux de données. Elle affiche les données contenu dans son model sous forme de tableau.

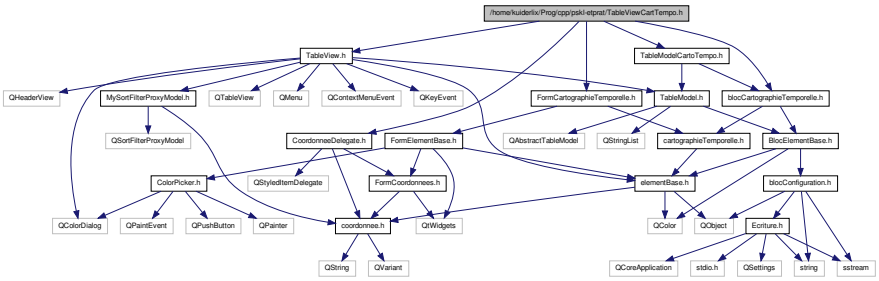
5.160 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewCartTempo.cpp

`#include "TableViewCartTempo.h"` Graphe des dépendances par inclusion de TableViewCartTempo.cpp :

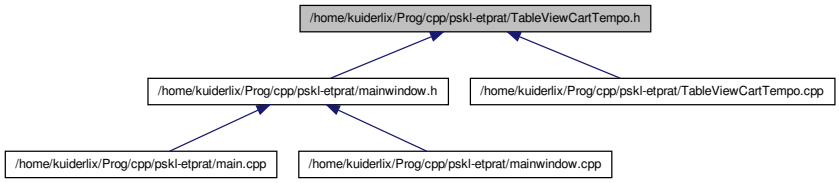


5.161 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewCartTempo.h

`#include "TableView.h"` `#include "blocCartographieTemporelle.-h"` `#include "TableModelCartoTempo.h"` `#include "Coordonnee-Delegate.h"` `#include "FormCartographieTemporelle.h"` Graphe des dépendances par inclusion de TableViewCartTempo.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

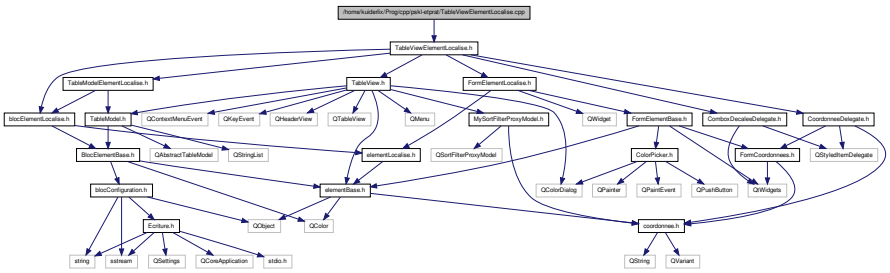


Classes

– class **TableViewCartTempo**

5.162 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewElementLocalise.cpp

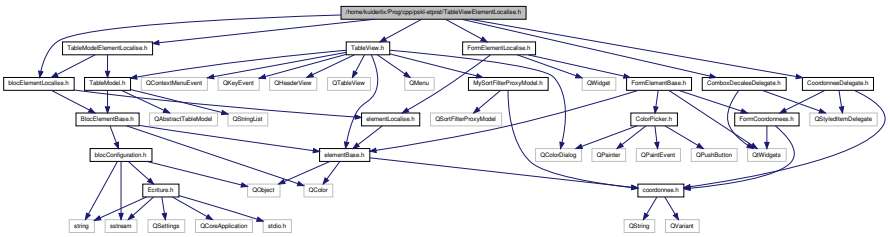
`#include "TableViewElementLocalise.h"` Graphe des dépendances par inclusion de TableViewElementLocalise.cpp :



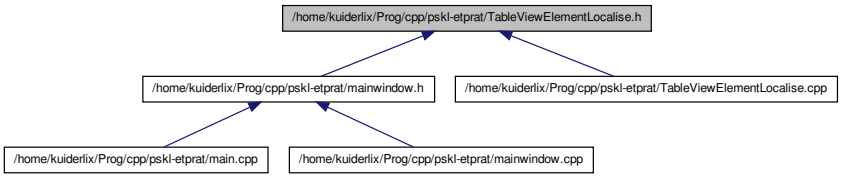
5.163 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewElementLocalise.h

`#include "TableView.h"` `#include "blocElementLocalise.-h"` `#include "TableModelElementLocalise.h"` `#include "-FormElementLocalise.h"` `#include "CoordonneeDelegate.h"` × `#include "ComboxDecaleeDelegate.h"` Graphe des dépendances par

inclusion de TableViewElementLocalise.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

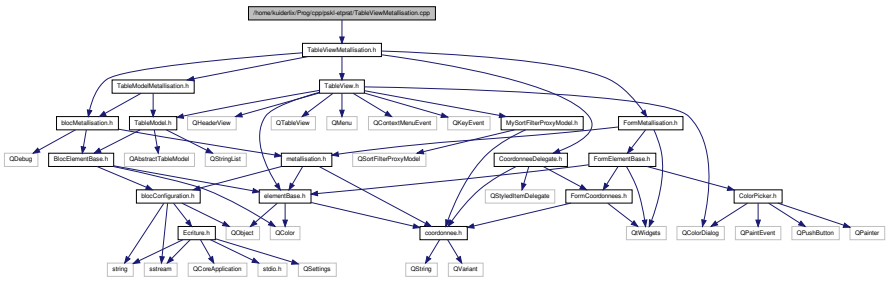


Classes

– class **TableViewElementLocalise**

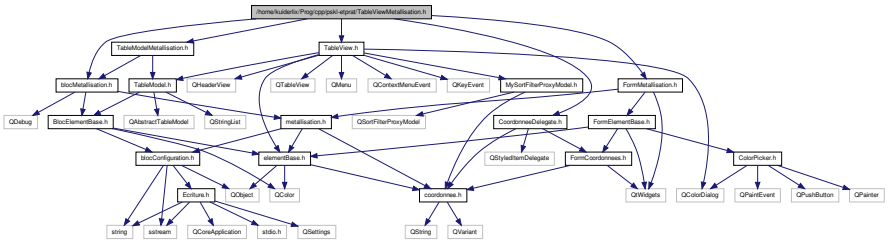
5.164 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewMetallisation.cpp

`#include "TableViewMetallisation.h"` Graphe des dépendances par inclusion de TableViewMetallisation.cpp :

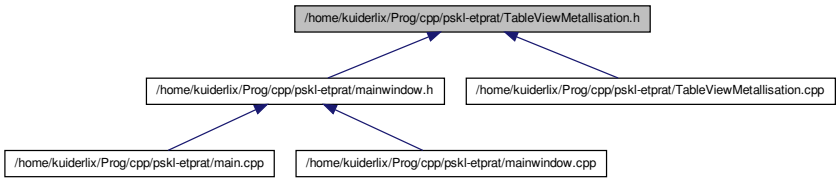


5.165 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewMetallisation.h

`#include "TableView.h"` `#include "TableModelMetallisation.-h"` `#include "CoordonneeDelegate.h"` `#include "blocMetallisation.-h"` `#include "FormMetallisation.h"` Graphe des dépendances par inclusion de TableViewMetallisation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



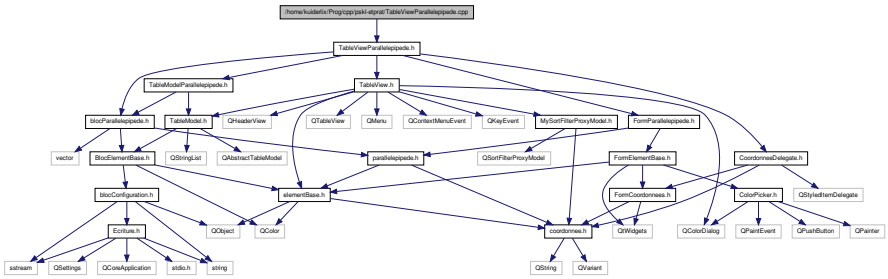
Classes

– class **TableViewMetallisation**

5.166 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewParallelepipede.cpp

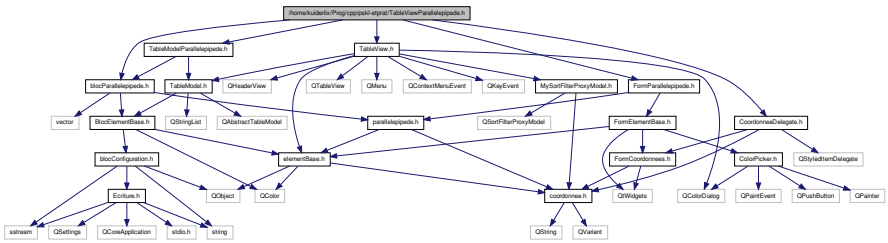
`#include "TableViewParallelepipede.h"` Graphe des dépendances par

inclusion de TableViewParallelepipede.cpp :

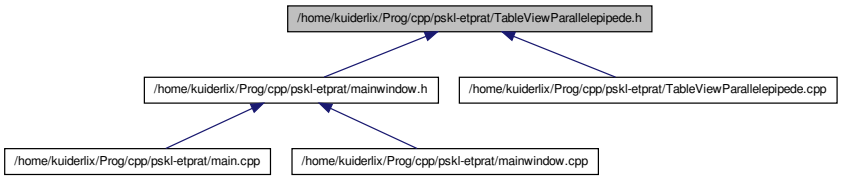


5.167 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewParallelepipede.h

`#include "TableView.h"` `#include "TableModelParallelepipede.-h"` `#include "CoordonneeDelegate.h"` `#include "blocParallelepipede.-h"` `#include "FormParallelepipede.h"` Graphe des dépendances par inclusion de TableViewParallelepipede.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

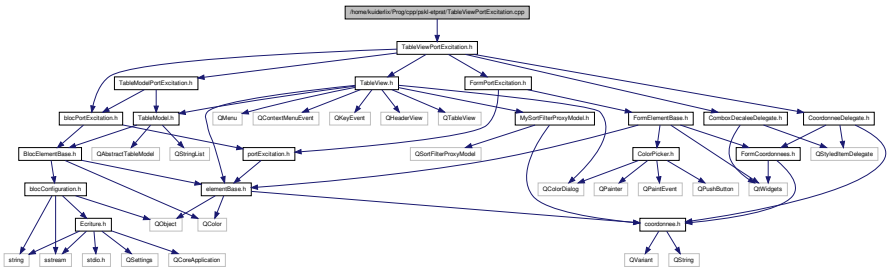


Classes

– class **TableViewParallelepipede**

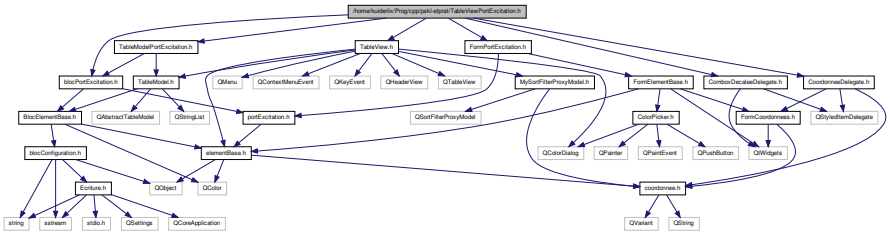
5.168 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewPortExcitation.cpp

`#include "TableViewPortExcitation.h"` Graphe des dépendances par inclusion de TableViewPortExcitation.cpp :

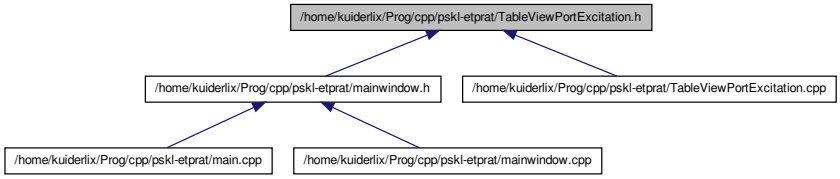


5.169 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewPortExcitation.h

`#include "TableView.h"` `#include "blocPortExcitation.h"` `#include "TableModelPortExcitation.h"` `#include "CoordonneeDelegate.h"` `#include "ComboxDecaleeDelegate.h"` `#include "FormPortExcitation.h"` Graphe des dépendances par inclusion de TableViewPortExcitation.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

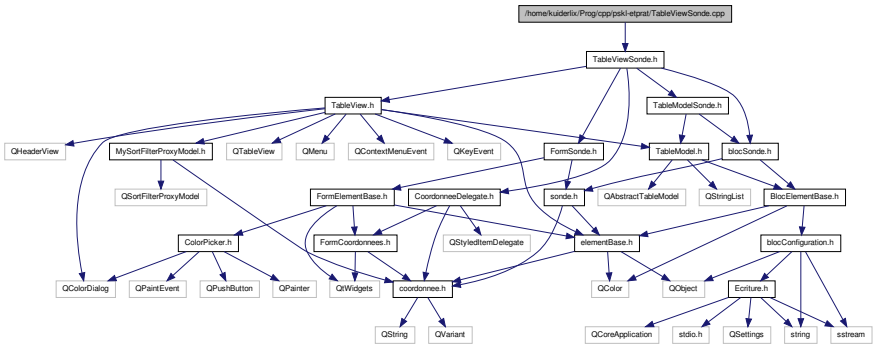


Classes

– class **TableViewPortExcitation**

5.170 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewSonde.cpp

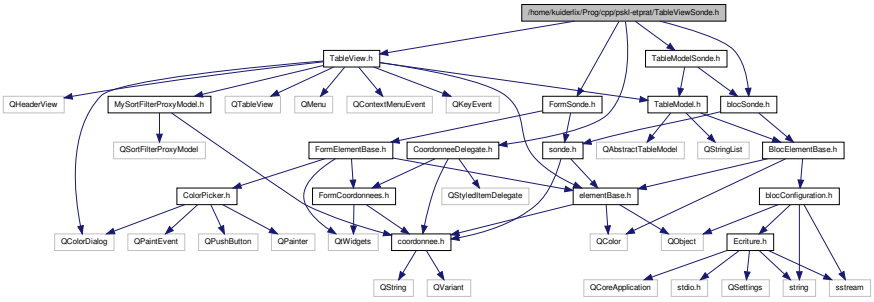
`#include "TableViewSonde.h"` Graphe des dépendances par inclusion de -TableViewSonde.cpp :



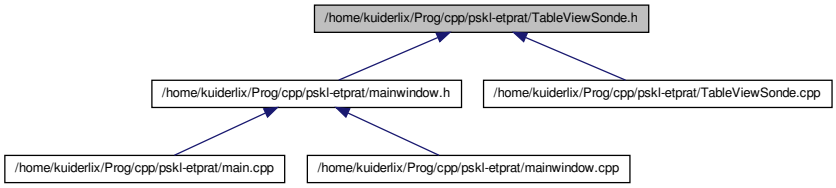
5.171 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewSonde.h

`#include "TableView.h"` `#include "blocSonde.h"` `#include "-FormSonde.h"` `#include "CoordonneeDelegate.h"` `#include "-TableModelSonde.h"` Graphe des dépendances par inclusion de TableView-

Sonde.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



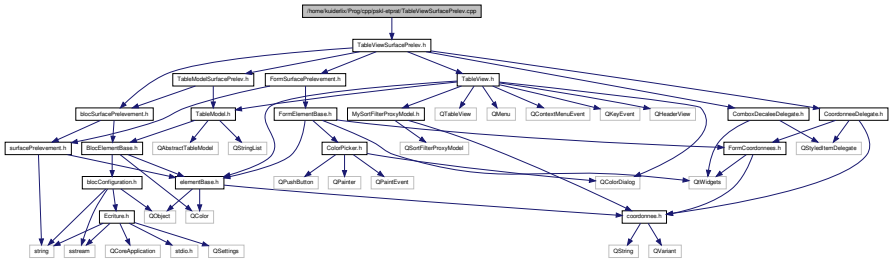
Classes

– class **TableViewSonde**

5.172 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewSurfacePrelev.cpp

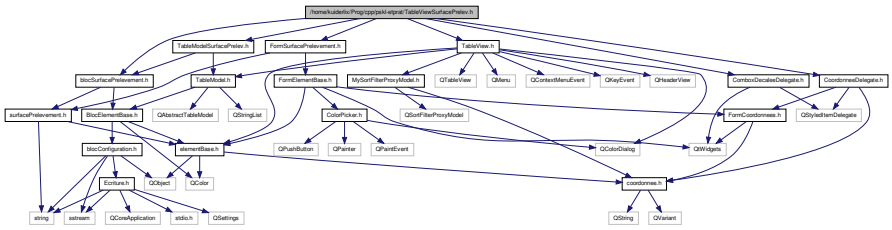
`#include "TableViewSurfacePrelev.h"` Graphe des dépendances par in-

clusion de TableViewSurfacePrelev.cpp :

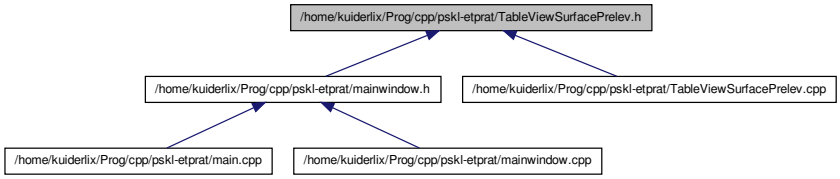


5.173 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-TableViewSurfacePrelev.h

`#include "TableView.h"` `#include "blocSurfacePrelevement.-h"` `#include "FormSurfacePrelevement.h"` `#include "Coordonnee-Delegate.h"` `#include "TableModelSurfacePrelev.h"` `#include "ComboxDecaleeDelegate.h"` Graphe des dépendances par inclusion de -TableViewSurfacePrelev.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

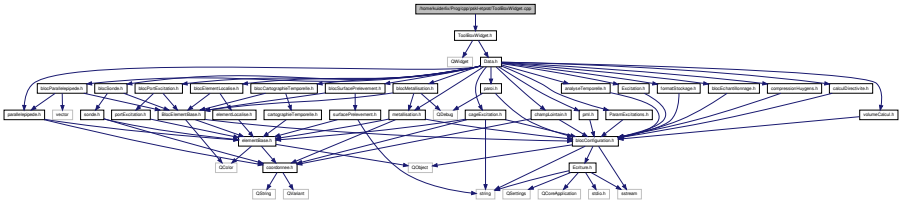


Classes

– class **TableViewSurfacePrelev**

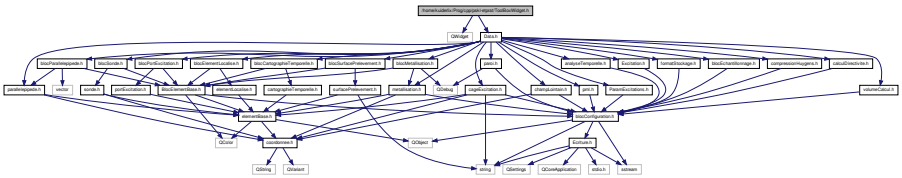
5.174 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ToolBoxWidget.cpp

`#include "ToolBoxWidget.h"` Graphe des dépendances par inclusion de -ToolBoxWidget.cpp :



5.175 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/-ToolBoxWidget.h

`#include <QWidget>` `#include "Data.h"` Graphe des dépendances par inclusion de ToolBoxWidget.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

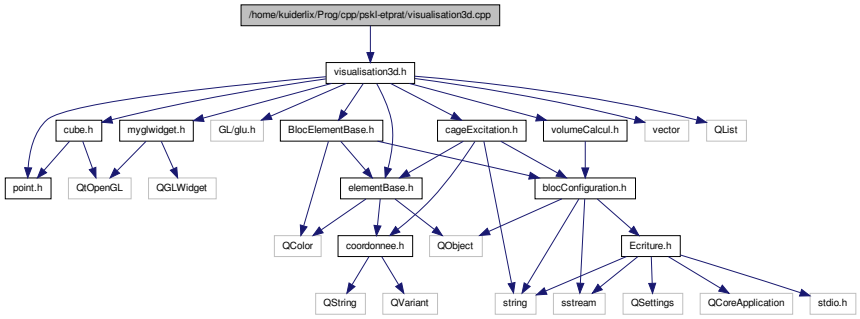
Classes

– class **ToolBoxWidget**

Classe de base pour les widgets conteneurs de la ToolBox.

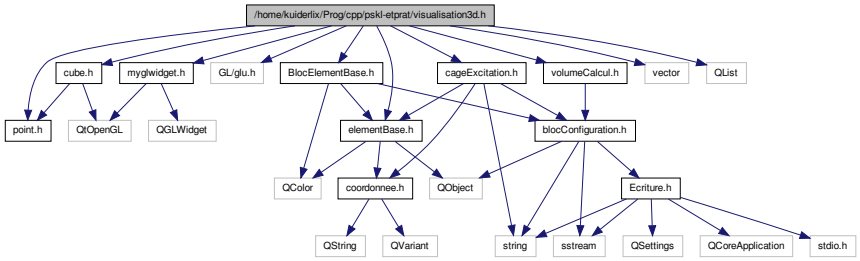
5.176 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/visualisation3d.cpp

`#include "visualisation3d.h"` Graphe des dépendances par inclusion de visualisation3d.cpp :

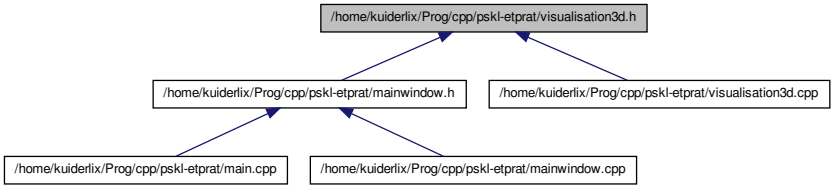


5.177 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/visualisation3d.h

`#include "myglwidget.h"` `#include <GL/glu.h>` `#include "cube.h"` `#include "elementBase.h"` `#include "point.h"` `#include <vector>` `#include "volumeCalcul.h"` `#include <QList>` × `#include "BlocElementBase.h"` `#include "cageExcitation.h"` Graphe des dépendances par inclusion de visualisation3d.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :

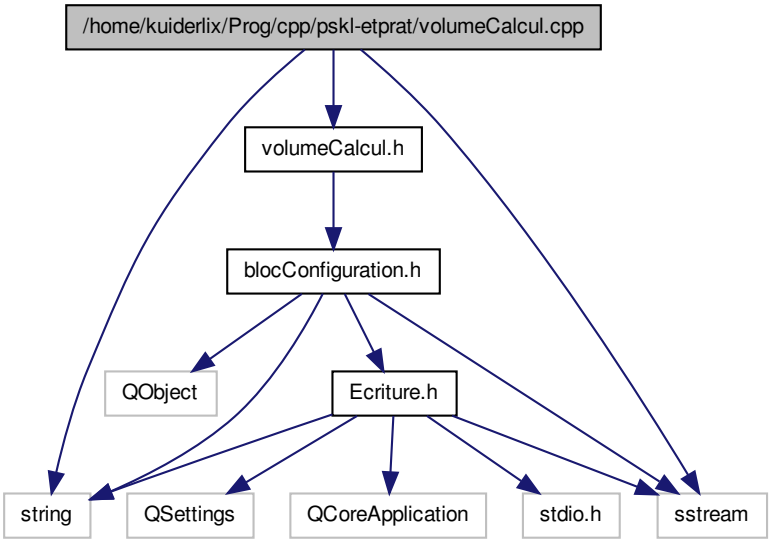


Classes

– class **Visualisation3D**
Affiche et gère la scène 3D.

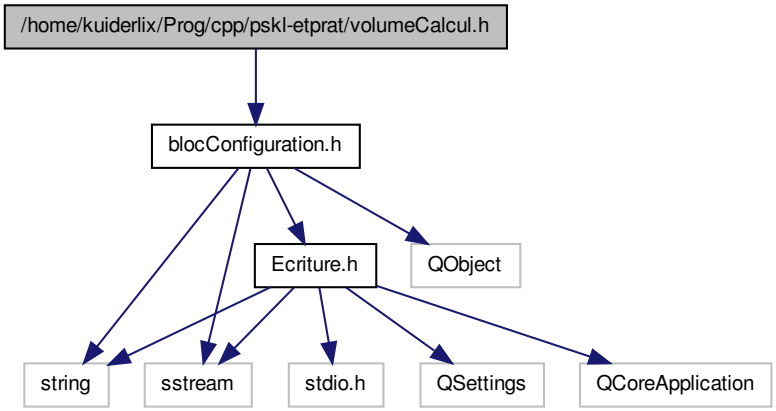
5.178 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/volumeCalcul.cpp

`#include "volumeCalcul.h"` `#include <sstream>` `#include <string>` Graphe des dépendances par inclusion de volumeCalcul.cpp :

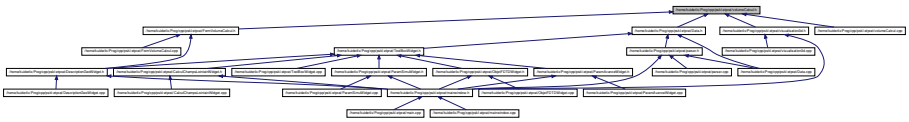


5.179 Référence du fichier /home/kuiderlix/Prog/cpp/pskl-etprat/volumeCalcul.h

`#include "blocConfiguration.h"` Graphe des dépendances par inclusion de volumeCalcul.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



Classes

– class **volumeCalcul**

*Cette classe représente la variable **volumeCalcul** (p. 201) d'une simulation.*