

Engauge Digitizer

2

Generated by Doxygen 1.8.11

Contents

1 Engauge Digitizer II	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	9
3.1 Class List	9
4 File Index	21
4.1 File List	21
5 Class Documentation	33
5.1 BackgroundStateAbstractBase Class Reference	33
5.1.1 Detailed Description	34
5.2 BackgroundStateContext Class Reference	34
5.2.1 Detailed Description	35
5.2.2 Member Function Documentation	35
5.2.2.1 setCurveSelected(const Transformation &transformation, const Document↵ ModelGridRemoval &modelGridRemoval, const DocumentModelColorFilter &modelColorFilter, const QString &curveSelected)	35
5.3 BackgroundStateCurve Class Reference	36
5.3.1 Detailed Description	36
5.4 BackgroundStateNone Class Reference	37
5.4.1 Detailed Description	37
5.5 BackgroundStateOriginal Class Reference	38
5.5.1 Detailed Description	38

5.6	BackgroundStateUnloaded Class Reference	39
5.6.1	Detailed Description	39
5.7	CallbackAddPointsInCurvesGraphs Class Reference	40
5.7.1	Detailed Description	40
5.8	CallbackAxesCheckerFromAxesPoints Class Reference	40
5.8.1	Detailed Description	40
5.9	CallbackAxisPointsAbstract Class Reference	41
5.9.1	Detailed Description	42
5.9.2	Member Function Documentation	42
5.9.2.1	isError() const	42
5.9.2.2	matrixGraph() const	42
5.9.2.3	matrixScreen() const	42
5.10	CallbackBoundingRects Class Reference	43
5.10.1	Detailed Description	43
5.11	CallbackCheckAddPointAxis Class Reference	43
5.11.1	Detailed Description	44
5.12	CallbackCheckEditPointAxis Class Reference	44
5.12.1	Detailed Description	44
5.13	CallbackDocumentHash Class Reference	45
5.13.1	Detailed Description	45
5.14	CallbackGatherXThetaValuesFunctions Class Reference	45
5.14.1	Detailed Description	46
5.15	CallbackNextOrdinal Class Reference	46
5.15.1	Detailed Description	46
5.16	CallbackPointOrdinal Class Reference	46
5.16.1	Detailed Description	47
5.17	CallbackRemovePointsInCurvesGraphs Class Reference	47
5.17.1	Detailed Description	47
5.18	CallbackSceneUpdateAfterCommand Class Reference	48
5.18.1	Detailed Description	48

5.19 CallbackUpdateTransform Class Reference	48
5.19.1 Detailed Description	49
5.19.2 Member Function Documentation	49
5.19.2.1 transformIsDefined() const	49
5.20 Checker Class Reference	49
5.20.1 Detailed Description	50
5.20.2 Member Function Documentation	50
5.20.2.1 prepareForDisplay(const QPolygonF &polygon, int pointRadius, const DocumentModelAxesChecker &modelAxesChecker, const DocumentModel↵ Coords &modelCoords, DocumentAxesPointsRequired documentAxesPoints↵ Required)	50
5.20.2.2 prepareForDisplay(const QList< Point > &Points, int pointRadius, const DocumentModelAxesChecker &modelAxesChecker, const DocumentModel↵ Coords &modelCoords, const Transformation &transformation, DocumentAxes↵ PointsRequired documentAxesPointsRequired)	50
5.20.2.3 updateModelAxesChecker(const DocumentModelAxesChecker &modelAxes↵ Checker)	50
5.21 ChecklistGuide Class Reference	51
5.21.1 Detailed Description	51
5.22 ChecklistGuideBrowser Class Reference	52
5.22.1 Detailed Description	52
5.23 ChecklistGuidePage Class Reference	52
5.23.1 Detailed Description	53
5.24 ChecklistGuidePageConclusion Class Reference	53
5.24.1 Detailed Description	54
5.25 ChecklistGuidePageCurves Class Reference	54
5.25.1 Detailed Description	55
5.26 ChecklistGuidePageIntro Class Reference	55
5.26.1 Detailed Description	55
5.27 ChecklistGuideWizard Class Reference	56
5.27.1 Detailed Description	56
5.28 ChecklistLineEdit Class Reference	56
5.28.1 Detailed Description	57

5.29	CmdAbstract Class Reference	57
5.29.1	Detailed Description	59
5.29.2	Member Function Documentation	59
5.29.2.1	resetSelection(const PointIdentifiers &pointIdentifiersToSelect)	59
5.29.2.2	saveOrCheckPostCommandDocumentStateHash(const Document &document)	59
5.29.2.3	saveOrCheckPreCommandDocumentStateHash(const Document &document)	59
5.30	CmdAddPointAxis Class Reference	60
5.30.1	Detailed Description	60
5.31	CmdAddPointGraph Class Reference	61
5.31.1	Detailed Description	61
5.32	CmdAddPointsGraph Class Reference	62
5.32.1	Detailed Description	62
5.33	CmdCopy Class Reference	63
5.33.1	Detailed Description	63
5.34	CmdCut Class Reference	64
5.34.1	Detailed Description	64
5.35	CmdDelete Class Reference	65
5.35.1	Detailed Description	65
5.36	CmdEditPointAxis Class Reference	66
5.36.1	Detailed Description	66
5.37	CmdEditPointGraph Class Reference	67
5.37.1	Detailed Description	67
5.38	CmdFactory Class Reference	68
5.38.1	Detailed Description	68
5.39	CmdMediator Class Reference	68
5.39.1	Detailed Description	70
5.39.2	Member Function Documentation	70
5.39.2.1	isModified() const	70
5.39.2.2	setDocumentAxesPointsRequired(DocumentAxesPointsRequired document↔AxesPointsRequired)	70
5.40	CmdMoveBy Class Reference	70

5.40.1 Detailed Description	71
5.41 CmdPaste Class Reference	71
5.41.1 Detailed Description	72
5.42 CmdPointChangeBase Class Reference	72
5.42.1 Detailed Description	73
5.43 CmdRedoForTest Class Reference	74
5.43.1 Detailed Description	74
5.44 CmdSelectCoordSystem Class Reference	75
5.44.1 Detailed Description	75
5.45 CmdSettingsAxesChecker Class Reference	76
5.45.1 Detailed Description	76
5.46 CmdSettingsColorFilter Class Reference	77
5.46.1 Detailed Description	77
5.47 CmdSettingsCoords Class Reference	78
5.47.1 Detailed Description	78
5.48 CmdSettingsCurveAddRemove Class Reference	79
5.48.1 Detailed Description	79
5.49 CmdSettingsCurveProperties Class Reference	80
5.49.1 Detailed Description	80
5.50 CmdSettingsDigitizeCurve Class Reference	81
5.50.1 Detailed Description	81
5.51 CmdSettingsExportFormat Class Reference	82
5.51.1 Detailed Description	82
5.52 CmdSettingsGeneral Class Reference	83
5.52.1 Detailed Description	83
5.53 CmdSettingsGridDisplay Class Reference	84
5.53.1 Detailed Description	84
5.54 CmdSettingsGridRemoval Class Reference	85
5.54.1 Detailed Description	85
5.55 CmdSettingsPointMatch Class Reference	86

5.55.1 Detailed Description	86
5.56 CmdSettingsSegments Class Reference	87
5.56.1 Detailed Description	87
5.57 CmdStackShadow Class Reference	88
5.57.1 Detailed Description	88
5.58 CmdUndoForTest Class Reference	89
5.58.1 Detailed Description	89
5.59 ColorFilter Class Reference	90
5.59.1 Detailed Description	90
5.59.2 Member Function Documentation	90
5.59.2.1 marginColor(const QImage *image) const	90
5.59.2.2 pixelToZeroToOneOrMinusOne(ColorFilterMode colorFilterMode, const QColor &pixel, QRgb rgbBackground) const	91
5.60 ColorFilterEntry Struct Reference	91
5.60.1 Detailed Description	91
5.61 ColorFilterHistogram Class Reference	91
5.61.1 Detailed Description	92
5.61.2 Member Function Documentation	92
5.61.2.1 generate(const ColorFilter &filter, double histogramBins[], ColorFilterMode colorFilterMode, const QImage &image, int &maxBinCount) const	92
5.62 ColorFilterSettings Class Reference	92
5.62.1 Detailed Description	94
5.62.2 Member Function Documentation	94
5.62.2.1 high() const	94
5.62.2.2 low() const	94
5.63 ColorFilterSettingsStrategyAbstractBase Class Reference	95
5.63.1 Detailed Description	95
5.64 ColorFilterSettingsStrategyForeground Class Reference	95
5.64.1 Detailed Description	96
5.65 ColorFilterSettingsStrategyHue Class Reference	96
5.65.1 Detailed Description	97

5.66	ColorFilterSettingsStrategyIntensity Class Reference	97
5.66.1	Detailed Description	97
5.67	ColorFilterSettingsStrategySaturation Class Reference	98
5.67.1	Detailed Description	98
5.68	ColorFilterSettingsStrategyValue Class Reference	98
5.68.1	Detailed Description	99
5.69	ColorFilterStrategyAbstractBase Class Reference	99
5.69.1	Detailed Description	100
5.70	ColorFilterStrategyForeground Class Reference	100
5.70.1	Detailed Description	100
5.71	ColorFilterStrategyHue Class Reference	101
5.71.1	Detailed Description	101
5.72	ColorFilterStrategyIntensity Class Reference	101
5.72.1	Detailed Description	102
5.73	ColorFilterStrategySaturation Class Reference	102
5.73.1	Detailed Description	103
5.74	ColorFilterStrategyValue Class Reference	103
5.74.1	Detailed Description	103
5.75	CoordSystem Class Reference	104
5.75.1	Detailed Description	107
5.75.2	Member Function Documentation	107
5.75.2.1	addPointAxisWithGeneratedIdentifier(const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)	107
5.75.2.2	addPointAxisWithSpecifiedIdentifier(const QPointF &posScreen, const QPointF &posGraph, const QString &identifier, double ordinal, bool isXOnly)	107
5.75.2.3	isXOnly(const QString &pointIdentifier) const	108
5.75.2.4	updatePointOrdinals(const Transformation &transformation)	108
5.76	CoordSystemContext Class Reference	108
5.76.1	Detailed Description	112
5.76.2	Member Function Documentation	112
5.76.2.1	addPointAxisWithGeneratedIdentifier(const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)	112

5.76.2.2	<code>addPointAxisWithSpecifiedIdentifier(const QPointF &posScreen, const QPoint↵ F &posGraph, const QString &identifier, double ordinal, bool isXOnly)</code>	112
5.76.2.3	<code>updatePointOrdinals(const Transformation &transformation)</code>	113
5.77	CoordSystemInterface Class Reference	113
5.77.1	Detailed Description	116
5.77.2	Member Function Documentation	116
5.77.2.1	<code>addPointAxisWithGeneratedIdentifier(const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)=0</code>	116
5.77.2.2	<code>addPointAxisWithSpecifiedIdentifier(const QPointF &posScreen, const QPoint↵ F &posGraph, const QString &identifier, double ordinal, bool isXOnly)=0</code>	117
5.77.2.3	<code>updatePointOrdinals(const Transformation &transformation)=0</code>	117
5.78	Correlation Class Reference	117
5.78.1	Detailed Description	118
5.78.2	Member Function Documentation	118
5.78.2.1	<code>correlateWithoutShift(int N, const double function1[], const double function2[], double &corrMax) const</code>	118
5.78.2.2	<code>correlateWithShift(int N, const double function1[], const double function2[], int &binStartMax, double &corrMax, double correlations[]) const</code>	118
5.79	CursorFactory Class Reference	118
5.79.1	Detailed Description	119
5.80	Curve Class Reference	119
5.80.1	Detailed Description	120
5.80.2	Member Function Documentation	120
5.80.2.1	<code>updatePointOrdinals(const Transformation &transformation)</code>	120
5.81	CurveNameList Class Reference	121
5.81.1	Detailed Description	121
5.82	CurveNameListEntry Class Reference	122
5.82.1	Detailed Description	123
5.83	CurveSettingsInt Class Reference	123
5.83.1	Detailed Description	123
5.84	CurvesGraphs Class Reference	124
5.84.1	Detailed Description	125
5.85	CurveStyle Class Reference	125

5.85.1 Detailed Description	126
5.86 CurveStyles Class Reference	126
5.86.1 Detailed Description	127
5.87 DigitizeStateAbstractBase Class Reference	127
5.87.1 Detailed Description	128
5.87.2 Member Function Documentation	129
5.87.2.1 begin(CmdMediator *cmdMediator, DigitizeState previousState)=0	129
5.88 DigitizeStateAxis Class Reference	129
5.88.1 Detailed Description	130
5.88.2 Member Function Documentation	130
5.88.2.1 begin(CmdMediator *cmdMediator, DigitizeState previousState)	130
5.89 DigitizeStateColorPicker Class Reference	131
5.89.1 Detailed Description	132
5.89.2 Member Function Documentation	132
5.89.2.1 begin(CmdMediator *cmdMediator, DigitizeState previousState)	132
5.90 DigitizeStateContext Class Reference	132
5.90.1 Detailed Description	134
5.91 DigitizeStateCurve Class Reference	134
5.91.1 Detailed Description	135
5.91.2 Member Function Documentation	135
5.91.2.1 begin(CmdMediator *cmdMediator, DigitizeState previousState)	135
5.92 DigitizeStateEmpty Class Reference	135
5.92.1 Detailed Description	136
5.92.2 Member Function Documentation	137
5.92.2.1 begin(CmdMediator *cmdMediator, DigitizeState previousState)	137
5.93 DigitizeStatePointMatch Class Reference	137
5.93.1 Detailed Description	138
5.93.2 Member Function Documentation	138
5.93.2.1 begin(CmdMediator *cmdMediator, DigitizeState previousState)	138
5.94 DigitizeStateSegment Class Reference	139

5.103.1 Detailed Description	148
5.104DlgImportAdvanced Class Reference	148
5.104.1 Detailed Description	149
5.105DlgImportCroppingNonPdf Class Reference	149
5.105.1 Detailed Description	150
5.106DlgImportCroppingPdf Class Reference	150
5.106.1 Detailed Description	151
5.107DlgRequiresTransform Class Reference	151
5.107.1 Detailed Description	151
5.108DlgSettingsAbstractBase Class Reference	152
5.108.1 Detailed Description	153
5.108.2 Member Function Documentation	153
5.108.2.1 enableOk(bool enable)	153
5.109DlgSettingsAxesChecker Class Reference	154
5.109.1 Detailed Description	154
5.110DlgSettingsColorFilter Class Reference	155
5.110.1 Detailed Description	156
5.111DlgSettingsCoords Class Reference	156
5.111.1 Detailed Description	157
5.112DlgSettingsCurveAddRemove Class Reference	157
5.112.1 Detailed Description	158
5.113DlgSettingsCurveProperties Class Reference	158
5.113.1 Detailed Description	159
5.114DlgSettingsDigitizeCurve Class Reference	159
5.114.1 Detailed Description	160
5.115DlgSettingsExportFormat Class Reference	160
5.115.1 Detailed Description	161
5.116DlgSettingsGeneral Class Reference	161
5.116.1 Detailed Description	162
5.117DlgSettingsGridDisplay Class Reference	162

5.117.1 Detailed Description	163
5.118DlgSettingsGridRemoval Class Reference	163
5.118.1 Detailed Description	164
5.119DlgSettingsMainWindow Class Reference	164
5.119.1 Detailed Description	165
5.120DlgSettingsPointMatch Class Reference	165
5.120.1 Detailed Description	166
5.121DlgSettingsSegments Class Reference	166
5.121.1 Detailed Description	167
5.122DlgValidatorAbstract Class Reference	167
5.122.1 Detailed Description	167
5.123DlgValidatorDateTime Class Reference	168
5.123.1 Detailed Description	168
5.124DlgValidatorDegreesMinutesSeconds Class Reference	168
5.124.1 Detailed Description	169
5.125DlgValidatorFactory Class Reference	169
5.125.1 Detailed Description	170
5.126DlgValidatorNumber Class Reference	170
5.126.1 Detailed Description	170
5.127Document Class Reference	171
5.127.1 Detailed Description	174
5.127.2 Member Function Documentation	174
5.127.2.1 addCoordSystems(unsigned int numberCoordSystemToAdd)	174
5.127.2.2 addPointAxisWithGeneratedIdentifier(const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)	174
5.127.2.3 addPointAxisWithSpecifiedIdentifier(const QPointF &posScreen, const QPoint↵ F &posGraph, const QString &identifier, double ordinal, bool isXOnly)	175
5.127.2.4 setDocumentAxesPointsRequired(DocumentAxesPointsRequired document↵ AxesPointsRequired)	175
5.127.2.5 updatePointOrdinals(const Transformation &transformation)	175
5.128DocumentHashGenerator Class Reference	175
5.128.1 Detailed Description	176

5.129DocumentModelAbstractBase Class Reference	176
5.129.1 Detailed Description	177
5.130DocumentModelAxesChecker Class Reference	177
5.130.1 Detailed Description	178
5.131DocumentModelColorFilter Class Reference	178
5.131.1 Detailed Description	180
5.131.2 Member Function Documentation	180
5.131.2.1 high(const QString &curveName) const	180
5.131.2.2 low(const QString &curveName) const	180
5.132DocumentModelCoords Class Reference	181
5.132.1 Detailed Description	182
5.133DocumentModelDigitizeCurve Class Reference	182
5.133.1 Detailed Description	183
5.134DocumentModelExportFormat Class Reference	184
5.134.1 Detailed Description	185
5.135DocumentModelGeneral Class Reference	185
5.135.1 Detailed Description	186
5.136DocumentModelGridDisplay Class Reference	186
5.136.1 Detailed Description	188
5.136.2 Member Function Documentation	188
5.136.2.1 stable() const	188
5.137DocumentModelGridRemoval Class Reference	188
5.137.1 Detailed Description	190
5.137.2 Member Function Documentation	190
5.137.2.1 stable() const	190
5.138DocumentModelPointMatch Class Reference	191
5.138.1 Detailed Description	192
5.139DocumentModelSegments Class Reference	192
5.139.1 Detailed Description	193
5.140ExportAlignLinear Class Reference	193

5.140.1 Detailed Description	194
5.141 ExportAlignLog Class Reference	194
5.141.1 Detailed Description	194
5.142 ExportFileAbstractBase Class Reference	195
5.142.1 Detailed Description	195
5.143 ExportFileFunctions Class Reference	196
5.143.1 Detailed Description	196
5.143.2 Member Function Documentation	196
5.143.2.1 exportToFile(const DocumentModelExportFormat &modelExportOverride, const Document &document, const MainWindowModel &modelMainWindow, const Transformation &transformation, QTextStream &str) const	196
5.144 ExportFileRelations Class Reference	197
5.144.1 Detailed Description	197
5.144.2 Member Function Documentation	197
5.144.2.1 exportToFile(const DocumentModelExportFormat &modelExportOverride, const Document &document, const MainWindowModel &modelMainWindow, const Transformation &transformation, QTextStream &str) const	197
5.145 ExportImageForRegression Class Reference	198
5.145.1 Detailed Description	198
5.146 ExportOrdinalsSmooth Class Reference	198
5.146.1 Detailed Description	199
5.147 ExportOrdinalsStraight Class Reference	199
5.147.1 Detailed Description	199
5.148 ExportToClipboard Class Reference	199
5.148.1 Detailed Description	200
5.148.2 Member Function Documentation	200
5.148.2.1 exportToClipboard(const QStringList &selected, const Transformation &transformation, QTextStream &strCsv, QTextStream &strHtml, const Curve &curveAxis, const CurvesGraphs &curvesGraphsAll, CurvesGraphs &curvesGraphsSelected) const	200
5.149 ExportToFile Class Reference	200
5.149.1 Detailed Description	201
5.149.2 Member Function Documentation	201

5.149.2.1 exportToFile(const DocumentModelExportFormat &modelExport, const Document &document, const MainWindowModel &modelMainWindow, const Transformation &transformation, QTextStream &str) const	201
5.150ExportXThetaValuesMergedFunctions Class Reference	201
5.150.1 Detailed Description	202
5.151FileCmdAbstract Class Reference	202
5.151.1 Detailed Description	203
5.152FileCmdClose Class Reference	203
5.152.1 Detailed Description	204
5.153FileCmdExport Class Reference	204
5.153.1 Detailed Description	204
5.154FileCmdFactory Class Reference	205
5.154.1 Detailed Description	205
5.155FileCmdImport Class Reference	205
5.155.1 Detailed Description	206
5.156FileCmdOpen Class Reference	206
5.156.1 Detailed Description	206
5.157FileCmdScript Class Reference	207
5.157.1 Detailed Description	207
5.158FilterImage Class Reference	207
5.158.1 Detailed Description	208
5.159FormatCoordsUnits Class Reference	208
5.159.1 Detailed Description	208
5.160FormatCoordsUnitsStrategyAbstractBase Class Reference	208
5.160.1 Detailed Description	209
5.160.2 Member Function Documentation	209
5.160.2.1 precisionDigitsForRawNumber(double valueUnformatted, double valueUnformattedOther, bool isXTheta, const Transformation &transformation) const	209
5.161FormatCoordsUnitsStrategyNonPolarTheta Class Reference	209
5.161.1 Detailed Description	210
5.162FormatCoordsUnitsStrategyPolarTheta Class Reference	210

5.162.1 Detailed Description	211
5.163FormatDateTime Class Reference	211
5.163.1 Detailed Description	211
5.163.2 Member Function Documentation	211
5.163.2.1 parseInput(CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, const QString &stringUntrimmed, double &value) const	211
5.164FormatDegreesMinutesSecondsBase Class Reference	212
5.164.1 Detailed Description	212
5.164.2 Member Function Documentation	212
5.164.2.1 parseInput(const QString &stringUntrimmed, double &value) const	212
5.165FormatDegreesMinutesSecondsNonPolarTheta Class Reference	213
5.165.1 Detailed Description	213
5.166FormatDegreesMinutesSecondsPolarTheta Class Reference	213
5.166.1 Detailed Description	214
5.167GeometryModel Class Reference	214
5.167.1 Detailed Description	215
5.168GeometryStrategyAbstractBase Class Reference	215
5.168.1 Detailed Description	216
5.168.2 Member Function Documentation	216
5.168.2.1 insertSubintervalsAndLoadDistances(int subintervalsPerInterval, const QVector< QPointF > &positionsGraph, QVector< QPointF > &positionsGraphWith↔ Subintervals, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const	216
5.168.2.2 polygonAreaForSimplyConnected(const QVector< QPointF > &points) const	216
5.169GeometryStrategyContext Class Reference	216
5.169.1 Detailed Description	217
5.170GeometryStrategyFunctionSmooth Class Reference	217
5.170.1 Detailed Description	218
5.171GeometryStrategyFunctionStraight Class Reference	218
5.171.1 Detailed Description	218
5.172GeometryStrategyRelationSmooth Class Reference	219
5.172.1 Detailed Description	219

5.173GeometryStrategyRelationStraight Class Reference	219
5.173.1 Detailed Description	220
5.174GeometryWindow Class Reference	220
5.174.1 Detailed Description	221
5.175GhostEllipse Class Reference	221
5.175.1 Detailed Description	222
5.176GhostPath Class Reference	222
5.176.1 Detailed Description	223
5.177GhostPolygon Class Reference	223
5.177.1 Detailed Description	223
5.178Ghosts Class Reference	223
5.178.1 Detailed Description	224
5.179GraphicsArcItem Class Reference	224
5.179.1 Detailed Description	225
5.180GraphicsItemsExtractor Class Reference	225
5.180.1 Detailed Description	225
5.181GraphicsLinesForCurve Class Reference	226
5.181.1 Detailed Description	226
5.181.2 Member Function Documentation	227
5.181.2.1 addPoint(const QString &pointIdentifier, double ordinal, GraphicsPoint &point)	227
5.181.2.2 removeTemporaryPointIfExists()	227
5.182GraphicsLinesForCurves Class Reference	227
5.182.1 Detailed Description	228
5.182.2 Member Function Documentation	228
5.182.2.1 addPoint(const QString &curveName, const QString &pointIdentifier, double ordinal, GraphicsPoint &point)	228
5.182.2.2 removeTemporaryPointIfExists()	228
5.183GraphicsPoint Class Reference	229
5.183.1 Detailed Description	230
5.184GraphicsPointAbstractBase Class Reference	230
5.184.1 Detailed Description	230

5.185GraphicsPointEllipse Class Reference	231
5.185.1 Detailed Description	231
5.186GraphicsPointFactory Class Reference	232
5.186.1 Detailed Description	232
5.187GraphicsPointPolygon Class Reference	232
5.187.1 Detailed Description	233
5.188GraphicsScene Class Reference	233
5.188.1 Detailed Description	234
5.188.2 Member Function Documentation	235
5.188.2.1 removeTemporaryPointIfExists()	235
5.188.2.2 updateAfterCommand(CmdMediator &cmdMediator, double highlightOpacity, GeometryWindow *geometryWindow)	235
5.188.2.3 updateGraphicsLinesToMatchGraphicsPoints(const CurveStyles &modelCurve↵ Styles, const Transformation &transformation)	235
5.189GraphicsView Class Reference	235
5.189.1 Detailed Description	237
5.190GridClassifier Class Reference	237
5.190.1 Detailed Description	237
5.191GridHealer Class Reference	238
5.191.1 Detailed Description	238
5.191.2 Member Function Documentation	238
5.191.2.1 erasePixel(int xCol, int yRow)	238
5.192GridInitializer Class Reference	238
5.192.1 Detailed Description	239
5.193GridLine Class Reference	239
5.193.1 Detailed Description	240
5.194GridLineFactory Class Reference	240
5.194.1 Detailed Description	241
5.194.2 Member Function Documentation	241
5.194.2.1 createGridLine(double xFrom, double yFrom, double xTo, double yTo, const Trans- formation &transformation)	241
5.195GridLineLimiter Class Reference	241

5.195.1 Detailed Description	242
5.196GridLines Class Reference	242
5.196.1 Detailed Description	242
5.197GridRemoval Class Reference	242
5.197.1 Detailed Description	243
5.198HelpBrowser Class Reference	243
5.198.1 Detailed Description	243
5.199HelpWindow Class Reference	244
5.199.1 Detailed Description	244
5.200ImportCroppingUtilBase Class Reference	244
5.200.1 Detailed Description	245
5.201ImportCroppingUtilNonPdf Class Reference	245
5.201.1 Detailed Description	246
5.202ImportCroppingUtilPdf Class Reference	246
5.202.1 Detailed Description	246
5.202.2 Member Function Documentation	247
5.202.2.1 applyImportCropping(bool isRegression, const QString &fileName, ImportCropping importCropping, Poppler::Document *&document) const	247
5.203Jpeg2000 Class Reference	247
5.203.1 Detailed Description	247
5.204LineStyle Class Reference	247
5.204.1 Detailed Description	248
5.205LoadFileInfo Class Reference	249
5.205.1 Detailed Description	249
5.206LoadImageFromUrl Class Reference	249
5.206.1 Detailed Description	250
5.207LoggerUpload Class Reference	250
5.207.1 Detailed Description	250
5.207.2 Member Function Documentation	250
5.207.2.1 loggerAssert(const char *condition, const char *file, int line) NO_RETURN_VALLUE	250

5.208MainWindow Class Reference	251
5.208.1 Detailed Description	253
5.208.2 Constructor & Destructor Documentation	253
5.208.2.1 MainWindow(const QString &errorReportFile, const QString &fileCmdScriptFile, bool isRegressionTest, bool isGnuplot, bool isReset, QStringList loadStartup↵ Files, QWidget *parent=0)	253
5.208.3 Member Function Documentation	253
5.208.3.1 selectOriginal(BackgroundImage backgroundImage)	253
5.208.3.2 updateGraphicsLinesToMatchGraphicsPoints()	253
5.209MainWindowModel Class Reference	254
5.209.1 Detailed Description	255
5.210MigrateToVersion6 Class Reference	255
5.210.1 Detailed Description	256
5.211MimePoints Class Reference	256
5.211.1 Detailed Description	257
5.212NetworkClient Class Reference	257
5.212.1 Detailed Description	257
5.213NonPdf Class Reference	258
5.213.1 Detailed Description	258
5.214NonPdfCropping Class Reference	258
5.214.1 Detailed Description	259
5.215NonPdfFrameHandle Class Reference	259
5.215.1 Detailed Description	260
5.216OrdinalGenerator Class Reference	260
5.216.1 Detailed Description	260
5.217Pdf Class Reference	260
5.217.1 Detailed Description	261
5.218PdfCropping Class Reference	261
5.218.1 Detailed Description	262
5.219PdfFrameHandle Class Reference	262
5.219.1 Detailed Description	263

5.220	Point Class Reference	263
5.220.1	Detailed Description	264
5.220.2	Constructor & Destructor Documentation	264
5.220.2.1	Point(const QString &curveName, const QPointF &posScreen)	264
5.220.2.2	Point(const QString &curveName, const QPointF &posScreen, const QPointF &posGraph, bool isXOnly)	265
5.221	PointComparator Struct Reference	265
5.221.1	Detailed Description	265
5.222	PointIdentifiers Class Reference	265
5.222.1	Detailed Description	266
5.222.2	Member Function Documentation	266
5.222.2.1	getKey(int i) const	266
5.223	PointMatchAlgorithm Class Reference	266
5.223.1	Detailed Description	267
5.224	PointMatchPixel Class Reference	267
5.224.1	Detailed Description	268
5.225	PointMatchTriplet Class Reference	268
5.225.1	Detailed Description	268
5.226	PointStyle Class Reference	268
5.226.1	Detailed Description	270
5.227	Segment Class Reference	270
5.227.1	Detailed Description	271
5.227.2	Member Function Documentation	271
5.227.2.1	firstPoint() const	271
5.227.2.2	removeUnneededLines(int *foldedLines)	271
5.228	SegmentFactory Class Reference	271
5.228.1	Detailed Description	272
5.229	SegmentLine Class Reference	272
5.229.1	Detailed Description	273
5.230	SettingsForGraph Class Reference	273
5.230.1	Detailed Description	273

5.231 Spline Class Reference	273
5.231.1 Detailed Description	274
5.231.2 Constructor & Destructor Documentation	274
5.231.2.1 Spline(const std::vector< double > &t, const std::vector< SplinePair > &xy)	274
5.231.3 Member Function Documentation	274
5.231.3.1 findSplinePairForFunctionX(double x, int numIterations) const	274
5.231.3.2 interpolateCoeff(double t) const	275
5.231.3.3 interpolateControlPoints(double t) const	275
5.232 SplineCoeff Class Reference	275
5.232.1 Detailed Description	276
5.233 SplinePair Class Reference	276
5.233.1 Detailed Description	276
5.234 StatusBar Class Reference	277
5.234.1 Detailed Description	277
5.235 TestCorrelation Class Reference	278
5.235.1 Detailed Description	278
5.236 TestFormats Class Reference	278
5.236.1 Detailed Description	279
5.237 TestGraphCoords Class Reference	279
5.237.1 Detailed Description	279
5.238 TestProjectedPoint Class Reference	280
5.238.1 Detailed Description	280
5.239 TestSegmentFill Class Reference	280
5.239.1 Detailed Description	281
5.240 TestSpline Class Reference	281
5.240.1 Detailed Description	281
5.241 TestTransformation Class Reference	282
5.241.1 Detailed Description	282
5.242 TestValidators Class Reference	282
5.242.1 Detailed Description	283

5.243	Transformation Class Reference	283
5.243.1	Detailed Description	284
5.243.2	Member Function Documentation	285
5.243.2.1	calculateTransformFromLinearCartesianPoints(const QPointF &posFrom0, const QPointF &posFrom1, const QPointF &posFrom2, const QPointF &posTo0, const QPointF &posTo1, const QPointF &posTo2)	285
5.244	TransformationStateAbstractBase Class Reference	285
5.244.1	Detailed Description	286
5.245	TransformationStateContext Class Reference	286
5.245.1	Detailed Description	286
5.246	TransformationStateDefined Class Reference	287
5.246.1	Detailed Description	287
5.247	TransformationStateUndefined Class Reference	287
5.247.1	Detailed Description	288
5.248	TranslatorContainer Class Reference	288
5.248.1	Detailed Description	288
5.249	TutorialButton Class Reference	289
5.249.1	Detailed Description	289
5.250	TutorialButtonRect Class Reference	290
5.250.1	Detailed Description	290
5.251	TutorialButtonText Class Reference	290
5.251.1	Detailed Description	291
5.252	TutorialDlg Class Reference	291
5.252.1	Detailed Description	292
5.253	TutorialStateAbstractBase Class Reference	292
5.253.1	Detailed Description	293
5.254	TutorialStateAxisPoints Class Reference	293
5.254.1	Detailed Description	294
5.255	TutorialStateChecklistWizardAbstract Class Reference	294
5.255.1	Detailed Description	295
5.256	TutorialStateChecklistWizardLines Class Reference	295

5.256.1 Detailed Description	296
5.257 TutorialStateChecklistWizardPoints Class Reference	296
5.257.1 Detailed Description	297
5.258 TutorialStateColorFilter Class Reference	297
5.258.1 Detailed Description	298
5.259 TutorialStateContext Class Reference	298
5.259.1 Detailed Description	299
5.259.2 Member Function Documentation	299
5.259.2.1 requestDelayedStateTransition(TutorialState tutorialState)	299
5.259.2.2 requestImmediateStateTransition(TutorialState tutorialState)	299
5.260 TutorialStateCurveSelection Class Reference	299
5.260.1 Detailed Description	300
5.261 TutorialStateCurveType Class Reference	300
5.261.1 Detailed Description	301
5.262 TutorialStateIntroduction Class Reference	301
5.262.1 Detailed Description	302
5.263 TutorialStatePointMatch Class Reference	302
5.263.1 Detailed Description	303
5.264 TutorialStateSegmentFill Class Reference	303
5.264.1 Detailed Description	304
5.265 ViewPointStyle Class Reference	304
5.265.1 Detailed Description	305
5.266 ViewPreview Class Reference	305
5.266.1 Detailed Description	306
5.267 ViewProfile Class Reference	306
5.267.1 Detailed Description	306
5.268 ViewProfileDivider Class Reference	307
5.268.1 Detailed Description	307
5.269 ViewProfileScale Class Reference	308
5.269.1 Detailed Description	308
5.270 ViewSegmentFilter Class Reference	308
5.270.1 Detailed Description	309
6 File Documentation	311
6.1 Callback/CallbackSearchReturn.h File Reference	311
6.1.1 Detailed Description	311
6.1.2 Enumeration Type Documentation	311
6.1.2.1 CallbackSearchReturn	311
Index	313

Chapter 1

Engauge Digitizer II

The Engauge Digitizer II application quickly extracts numeric data from images containing graphs with curves and two axes drawn. Converting an image into data may be described as doing the opposite of graphing - which converts numeric data into graph images.

For users

Major features added since version Engauge Digitizer 5.2 include:

- Sub-pixel point placement increases accuracy
- Undo/redo makes recovering from mistakes easy
- Easier and more powerful zooming
- Improved drag-and-drop
- Wizard provides an interactive tutorial to explain the basic steps
- Wizard creates a checklist guide to interactively leads user through steps from file import to file export
- MSI installer for Windows operating system

For developers

Engauge Digitizer Version 2 uses the new [Qt5](#) library, rather than the old [Qt3](#) library used by Version 1. The Qt3 library is disappearing from most operating systems, but Qt5 should be available for many years.

The code takes advantage of some powerful open source toolkits:

- [CMake](#) provides important metrics to identify possible problem areas (run `doccmake` in `src` directory)
- [Doxygen](#) documents all C++ classes (run `doxygen` in `src` directory)
- [FFTW](#) provides a fast-fourier transform (FFT) for faster image processing, especially cross-correlations
- [Log4cpp](#) provides configurable logging
- [OpenJPEG](#) supports JPEG2000 images on systems without support for that format

The code is architected with some important design patterns:

- **Command pattern** provides Undo/Redo using commands and a command stack, and also provides robust data transfer between threads using commands and a FIFO command queue
- **Factory pattern** generates points with the details encapsulated in the factory class
- **Functor pattern** provides efficient processing of [Curve](#) and [Point](#) data from outside the [Document](#) class, without violating encapsulation of performed by generic iteration through the Curves with functors
- **Model/View pattern** separates graphical object management in the [GraphicsScene](#), and graphical presentation and interaction in the [GraphicsView](#). Delegates, representing the document, interact with the Model and View
- **State pattern** isolates each digitizing mode into one state, with a context class acting as a container and single class for interfacing across the state machine boundary
- **Strategy pattern** encapsulates code chunks when the chunk to be used depends on the current context

Code development has been moved from sourceforge.net to github.com, and Doxygen documentation has been added.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

BackgroundStateAbstractBase	33
BackgroundStateCurve	36
BackgroundStateNone	37
BackgroundStateOriginal	38
BackgroundStateUnloaded	39
BackgroundStateContext	34
CallbackAddPointsInCurvesGraphs	40
CallbackAxesCheckerFromAxesPoints	40
CallbackAxisPointsAbstract	41
CallbackCheckAddPointAxis	43
CallbackCheckEditPointAxis	44
CallbackUpdateTransform	48
CallbackBoundingRects	43
CallbackDocumentHash	45
CallbackGatherXThetaValuesFunctions	45
CallbackNextOrdinal	46
CallbackPointOrdinal	46
CallbackRemovePointsInCurvesGraphs	47
CallbackSceneUpdateAfterCommand	48
Checker	49
CmdFactory	68
ColorFilter	90
ColorFilterEntry	91
ColorFilterHistogram	91
ColorFilterSettings	92
ColorFilterSettingsStrategyAbstractBase	95
ColorFilterSettingsStrategyForeground	95
ColorFilterSettingsStrategyHue	96
ColorFilterSettingsStrategyIntensity	97
ColorFilterSettingsStrategySaturation	98
ColorFilterSettingsStrategyValue	98
ColorFilterStrategyAbstractBase	99
ColorFilterStrategyForeground	100
ColorFilterStrategyHue	101

ColorFilterStrategyIntensity	101
ColorFilterStrategySaturation	102
ColorFilterStrategyValue	103
CoordSystemInterface	113
CoordSystem	104
CoordSystemContext	108
Correlation	117
CursorFactory	118
Curve	119
CurveNameListEntry	122
CurveSettingsInt	123
CurvesGraphs	124
CurveStyle	125
CurveStyles	126
DigitizeStateAbstractBase	127
DigitizeStateAxis	129
DigitizeStateColorPicker	131
DigitizeStateCurve	134
DigitizeStateEmpty	135
DigitizeStatePointMatch	137
DigitizeStateSegment	139
DigitizeStateSelect	140
DlgFilterCommand	146
DlgValidatorFactory	169
Document	171
DocumentHashGenerator	175
DocumentModelAbstractBase	176
DocumentModelAxesChecker	177
DocumentModelColorFilter	178
DocumentModelCoords	181
DocumentModelDigitizeCurve	182
DocumentModelExportFormat	184
DocumentModelGeneral	185
DocumentModelGridDisplay	186
DocumentModelGridRemoval	188
DocumentModelPointMatch	191
DocumentModelSegments	192
MainWindowModel	254
ExportAlignLinear	193
ExportAlignLog	194
ExportFileAbstractBase	195
ExportFileFunctions	196
ExportFileRelations	197
ExportImageForRegression	198
ExportOrdinalsSmooth	198
ExportOrdinalsStraight	199
ExportToClipboard	199
ExportToFile	200
ExportXThetaValuesMergedFunctions	201
FileCmdAbstract	202
FileCmdClose	203
FileCmdExport	204
FileCmdImport	205
FileCmdOpen	206
FileCmdFactory	205
FileCmdScript	207
FilterImage	207

FormatCoordsUnits	208
FormatCoordsUnitsStrategyAbstractBase	208
FormatCoordsUnitsStrategyNonPolarTheta	209
FormatCoordsUnitsStrategyPolarTheta	210
FormatDateTime	211
FormatDegreesMinutesSecondsBase	212
FormatDegreesMinutesSecondsNonPolarTheta	213
FormatDegreesMinutesSecondsPolarTheta	213
GeometryStrategyAbstractBase	215
GeometryStrategyFunctionSmooth	217
GeometryStrategyFunctionStraight	218
GeometryStrategyRelationSmooth	219
GeometryStrategyRelationStraight	219
GeometryStrategyContext	216
GhostEllipse	221
GhostPath	222
GhostPolygon	223
Ghosts	223
GraphicsItemsExtractor	225
GraphicsLinesForCurves	227
GraphicsPointAbstractBase	230
GraphicsPoint	229
GraphicsPointFactory	232
GridClassifier	237
GridHealer	238
GridInitializer	238
GridLine	239
GridLineFactory	240
GridLineLimiter	241
GridLines	242
GridRemoval	242
ImportCroppingUtilBase	244
ImportCroppingUtilNonPdf	245
ImportCroppingUtilPdf	246
Jpeg2000	247
LineStyle	247
LoadFileInfo	249
LoggerUpload	250
MigrateToVersion6	255
NonPdf	258
NonPdfCropping	258
OrdinalGenerator	260
Pdf	260
PdfCropping	261
Point	263
PointComparator	265
PointIdentifiers	265
PointMatchAlgorithm	266
PointMatchPixel	267
PointMatchTriplet	268
PointStyle	268
QAbstractTableModel	
CurveNameList	121
QDialog	
DlgEditPointAxis	143
DlgEditPointGraph	144
DlgErrorReport	145

DlgImportCroppingNonPdf	149
DlgImportCroppingPdf	150
DlgSettingsAbstractBase	152
DlgImportAdvanced	148
DlgSettingsAxesChecker	154
DlgSettingsColorFilter	155
DlgSettingsCoords	156
DlgSettingsCurveAddRemove	157
DlgSettingsCurveProperties	158
DlgSettingsDigitizeCurve	159
DlgSettingsExportFormat	160
DlgSettingsGeneral	161
DlgSettingsGridDisplay	162
DlgSettingsGridRemoval	163
DlgSettingsMainWindow	164
DlgSettingsPointMatch	165
DlgSettingsSegments	166
TutorialDlg	291
QDockWidget	
ChecklistGuide	51
GeometryWindow	220
HelpWindow	244
QDoubleValidator	
DlgValidatorAbstract	167
DlgValidatorDateTime	168
DlgValidatorDegreesMinutesSeconds	168
DlgValidatorNumber	170
QGraphicsEllipseItem	
GraphicsArcItem	224
GraphicsPointEllipse	231
QGraphicsLineItem	
SegmentLine	272
QGraphicsPathItem	
GraphicsLinesForCurve	226
QGraphicsPolygonItem	
GraphicsPointPolygon	232
QGraphicsRectItem	
NonPdfFrameHandle	259
PdfFrameHandle	262
TutorialButtonRect	290
ViewProfileDivider	307
QGraphicsScene	
GraphicsScene	233
QGraphicsTextItem	
TutorialButtonText	290
QGraphicsView	
GraphicsView	235
ViewPreview	305
ViewProfile	306
QLabel	
ViewPointStyle	304
ViewProfileScale	308
ViewSegmentFilter	308
QLineEdit	
ChecklistLineEdit	56
DlgEditPointGraphLineEdit	145
QMainWindow	
MainWindow	251

QMessageBox	
DlgAbout	142
DlgRequiresTransform	151
QMimeData	
MimePoints	256
QNetworkAccessManager	
NetworkClient	257
QObject	
CmdStackShadow	88
DigitizeStateContext	132
DigitizeStateSegment	139
DlgFilterWorker	147
GraphicsPointEllipse	231
GraphicsPointPolygon	232
LoadImageFromUrl	249
Segment	270
SegmentLine	272
StatusBar	277
TestCorrelation	278
TestFormats	278
TestGraphCoords	279
TestProjectedPoint	280
TestSegmentFill	280
TestSpline	281
TestTransformation	282
TestValidators	282
TransformationStateDefined	287
TutorialButton	289
TutorialStateAbstractBase	292
TutorialStateAxisPoints	293
TutorialStateChecklistWizardAbstract	294
TutorialStateChecklistWizardLines	295
TutorialStateChecklistWizardPoints	296
TutorialStateColorFilter	297
TutorialStateCurveSelection	299
TutorialStateCurveType	300
TutorialStateIntroduction	301
TutorialStatePointMatch	302
TutorialStateSegmentFill	303
TutorialStateContext	298
ViewProfileDivider	307
QStandardItemModel	
GeometryModel	214
QTextBrowser	
ChecklistGuideBrowser	52
HelpBrowser	243
QThread	
DlgFilterThread	147
QUndoCommand	
CmdAbstract	57
CmdCopy	63
CmdPointChangeBase	72
CmdAddPointAxis	60
CmdAddPointGraph	61
CmdAddPointsGraph	62
CmdCut	64
CmdDelete	65
CmdEditPointAxis	66

CmdEditPointGraph	67
CmdMoveBy	70
CmdPaste	71
CmdRedoForTest	74
CmdSelectCoordSystem	75
CmdSettingsAxesChecker	76
CmdSettingsColorFilter	77
CmdSettingsCoords	78
CmdSettingsCurveAddRemove	79
CmdSettingsCurveProperties	80
CmdSettingsDigitizeCurve	81
CmdSettingsExportFormat	82
CmdSettingsGeneral	83
CmdSettingsGridDisplay	84
CmdSettingsGridRemoval	85
CmdSettingsPointMatch	86
CmdSettingsSegments	87
CmdUndoForTest	89
QUndoStack	
CmdMediator	68
QWizard	
ChecklistGuideWizard	56
QWizardPage	
ChecklistGuidePage	52
ChecklistGuidePageConclusion	53
ChecklistGuidePageCurves	54
ChecklistGuidePageIntro	55
SegmentFactory	271
SettingsForGraph	273
Spline	273
SplineCoeff	275
SplinePair	276
Transformation	283
TransformationStateAbstractBase	285
TransformationStateDefined	287
TransformationStateUndefined	287
TransformationStateContext	286
TranslatorContainer	288

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

BackgroundStateAbstractBase	
Background image state machine state base class	33
BackgroundStateContext	
Context class that manages the background image state machine	34
BackgroundStateCurve	
Background image state for showing filter image from current curve	36
BackgroundStateNone	
Background image state for showing no image	37
BackgroundStateOriginal	
Background image state for showing original (=unfiltered) image	38
BackgroundStateUnloaded	
Background image state for interval between startup and loading of the image	39
CallbackAddPointsInCurvesGraphs	
Callback that is used when iterating through a read-only CurvesGraphs to add corresponding points in Document	40
CallbackAxesCheckerFromAxesPoints	
Callback for positioning the axes indicator according to the axes points	40
CallbackAxisPointsAbstract	
Callback for collecting axis points and then performing common calculations on those axis points	41
CallbackBoundingRects	
Callback for computing the bounding rectangles of the screen and graph coordinates of the points in the Document	43
CallbackCheckAddPointAxis	
Callback for sanity checking the screen and graph coordinates of an axis point, before it is added to the axes curve	43
CallbackCheckEditPointAxis	
Callback for sanity checking the screen and graph coordinates of an axis point that is in the axes curve, before changing its graph coordinates	44
CallbackDocumentHash	
Callback for DocumentHash value for a Document	45
CallbackGatherXThetaValuesFunctions	
Callback for collecting X/Theta independent variables, for functions, in preparation for exporting	45
CallbackNextOrdinal	
Callback for computing the next ordinal for a new point	46

CallbackPointOrdinal	Callback for computing the ordinal for a specified point, as a function of the LineStyle and curve geometry	46
CallbackRemovePointsInCurvesGraphs	Callback that is used when iterating through a read-only CurvesGraphs to remove corresponding points in Document	47
CallbackSceneUpdateAfterCommand	Callback for updating the QGraphicsItems in the scene after a command may have modified Points in Curves	48
CallbackUpdateTransform	Callback for collecting axis points and then calculating the current transform from those axis points	48
Checker	Box shape that is drawn through the three axis points, to temporarily (usually) or permanently (rarely) highlight the local up/down/left/right directions when all axis points have been defined	49
ChecklistGuide	Dockable text window containing checklist guide	51
ChecklistGuideBrowser	Class that adds rudimentary tree collapse/expand functionality to QTextBrowser	52
ChecklistGuidePage	This class customizes QWizardPage for ChecklistGuideWizard	52
ChecklistGuidePageConclusion	This class uses the validation method of the Conclusion page to perform final processing for ChecklistGuideWizard	53
ChecklistGuidePageCurves	This class adds validation to the Curves page	54
ChecklistGuidePageIntro	This class sets up the introduction page	55
ChecklistGuideWizard	Wizard for setting up the checklist guide	56
ChecklistLineEdit	Adds key event handling to QLineEdit	56
CmdAbstract	Wrapper around QUndoCommand. This simplifies the more complicated feature set of QUndoCommand	57
CmdAddPointAxis	Command for adding one axis point	60
CmdAddPointGraph	Command for adding one graph point	61
CmdAddPointsGraph	Command for adding one or more graph points. This is for Segment Fill mode	62
CmdCopy	Command for moving all selected Points by a specified translation	63
CmdCut	Command for cutting all selected Points	64
CmdDelete	Command for deleting all selected Points	65
CmdEditPointAxis	Command for editing the graph coordinates one axis point	66
CmdEditPointGraph	Command for editing the graph coordinates of one or more graph points	67
CmdFactory	Factory for CmdAbstractBase objects	68
CmdMediator	Command queue stack	68
CmdMoveBy	Command for moving all selected Points by a specified translation	70

CmdPaste	Command for moving all selected Points by a specified translation	71
CmdPointChangeBase	Base class for CmdBase leaf subclasses that involve point additions, deletions and/or modifications	72
CmdRedoForTest	Command for performing Redo during testing	74
CmdSelectCoordSystem	Command for changing the currently selected CoordSystem	75
CmdSettingsAxesChecker	Command for DlgSettingsAxesChecker	76
CmdSettingsColorFilter	Command for DlgSettingsColorFilter	77
CmdSettingsCoords	Command for DlgSettingsCoords	78
CmdSettingsCurveAddRemove	Command for DlgSettingsCurveAddRemove	79
CmdSettingsCurveProperties	Command for DlgSettingsCurveProperties	80
CmdSettingsDigitizeCurve	Command for DlgSettingsDigitizeCurve	81
CmdSettingsExportFormat	Command for DlgSettingsExportFormat	82
CmdSettingsGeneral	Command for DlgSettingsGeneral	83
CmdSettingsGridDisplay	Command for DlgSettingsGridDisplay	84
CmdSettingsGridRemoval	Command for DlgSettingsGridRemoval	85
CmdSettingsPointMatch	Command for DlgSettingsPointMatch	86
CmdSettingsSegments	Command for DlgSettingsSegments	87
CmdStackShadow	Command stack that shadows the CmdMediator command stack at startup when reading commands from an error report file	88
CmdUndoForTest	Command for performing Undo during testing	89
ColorFilter	Class for filtering image to remove unimportant information	90
ColorFilterEntry	Helper class so ColorFilter class can compute the background color	91
ColorFilterHistogram	Class that generates a histogram according to the current filter	91
ColorFilterSettings	Color filter parameters for one curve. For a class, this is handled the same as LineStyle and PointStyle	92
ColorFilterSettingsStrategyAbstractBase	Base class for strategy pattern whose subclasses process the different color filter settings modes (one strategy per mode)	95
ColorFilterSettingsStrategyForeground	Leaf class for foreground strategy for ColorFilterSettings	95
ColorFilterSettingsStrategyHue	Leaf class for hue strategy for ColorFilterSettings	96
ColorFilterSettingsStrategyIntensity	Leaf class for intensity strategy for ColorFilterSettings	97
ColorFilterSettingsStrategySaturation	Leaf class for saturation strategy for ColorFilterSettings	98

ColorFilterSettingsStrategyValue	
Leaf class for value strategy for ColorFilterSettings	98
ColorFilterStrategyAbstractBase	
Base class for strategy pattern whose subclasses process the different color filter settings modes (one strategy per mode)	99
ColorFilterStrategyForeground	
Leaf class for foreground strategy for ColorFilter	100
ColorFilterStrategyHue	
Leaf class for hue strategy for ColorFilter	101
ColorFilterStrategyIntensity	
Leaf class for intensity strategy for ColorFilter	101
ColorFilterStrategySaturation	
Leaf class for saturation strategy for ColorFilter	102
ColorFilterStrategyValue	
Leaf class for value strategy for ColorFilter	103
CoordSystem	
Storage of data belonging to one coordinate system	104
CoordSystemContext	
This class plays the role of context class in a state machine, although the 'states' are actually different instantiations of the CoordSystem class	108
CoordSystemInterface	
Interface common to CoordSystemContext and CoordSystem classes	113
Correlation	
Fast cross correlation between two functions	117
CursorFactory	
Create standard cross cursor, or custom cursor, according to settings	118
Curve	
Container for one set of digitized Points	119
CurveNameList	
Model for DlgSettingsCurveAddRemove and CmdSettingsCurveAddRemove	121
CurveNameListEntry	
Utility class for converting the QVariant in CurveNameList to/from the curve names as QStrings, for use by the CurveNameList model class	122
CurveSettingsInt	
Internal settings for one curve, such as LineStyle , PointStyle and CurveFilter	123
CurvesGraphs	
Container for all graph curves. The axes point curve is external to this class	124
CurveStyle	
Container for LineStyle and PointStyle for one Curve	125
CurveStyles	
Model for DlgSettingsCurveProperties and CmdSettingsCurveProperties	126
DigitizeStateAbstractBase	
Base class for all digitizing states. This serves as an interface to DigitizeStateContext	127
DigitizeStateAxis	
Digitizing state for digitizing one axis point at a time	129
DigitizeStateColorPicker	
Digitizing state for selecting a color for DigitizeStateSegment	131
DigitizeStateContext	
Container for all DigitizeStateAbstractBase subclasses. This functions as the context class in a standard state machine implementation	132
DigitizeStateCurve	
Digitizing state for creating Curve Points, one at a time	134
DigitizeStateEmpty	
Digitizing state before a Document has been created. In this state, the cursor is Qt::ArrowCursor	135
DigitizeStatePointMatch	
Digitizing state for matching Curve Points, one at a time	137
DigitizeStateSegment	
Digitizing state for creating multiple Points along a highlighted segment	139

DigitizeStateSelect	
Digitizing state for selecting one or more Points in the Document	140
DlgAbout	
About Engauge dialog. This provides a hidden shortcut for triggering <code>ENGAGE_ASSERT</code>	142
DlgEditPointAxis	
Dialog box for editing the information of one axis point	143
DlgEditPointGraph	
Dialog box for editing the information of one or more points	144
DlgEditPointGraphLineEdit	
Adds hover highlighting to <code>QLineEdit</code>	145
DlgErrorReport	
Dialog for sending error report	145
DlgFilterCommand	
Command pattern object for receiving new parameters in DlgFilterWorker from GUI thread	146
DlgFilterThread	
Class for processing new filter settings. This is based on http://blog.debao.me/2013/08/how-to-use-qthread-in-the-right-way-part-1/	147
DlgFilterWorker	
Class for processing new filter settings. This is based on http://blog.debao.me/2013/08/how-to-use-qworker-in-the-right-way-part-1/	147
DlgImportAdvanced	
Dialog for setting the advanced parameters in a newly imported Document	148
DlgImportCroppingNonPdf	
Dialog for selecting a page and frame on that page when importing an image from a non-pdf file	149
DlgImportCroppingPdf	
Dialog for selecting a page and frame on that page when importing an image from a pdf file	150
DlgRequiresTransform	
Dialog to be displayed whenever some operation or processing cannot be performed since the axis points are not defined	151
DlgSettingsAbstractBase	
Abstract base class for all Settings dialogs	152
DlgSettingsAxesChecker	
Dialog for editing axes checker settings	154
DlgSettingsColorFilter	
Dialog for editing filtering settings	155
DlgSettingsCoords	
Dialog for editing coordinates settings	156
DlgSettingsCurveAddRemove	
Dialog for editing curve names settings	157
DlgSettingsCurveProperties	
Dialog for editing curve properties settings	158
DlgSettingsDigitizeCurve	
Dialog for editing DigitizeStateCurve settings	159
DlgSettingsExportFormat	
Dialog for editing exporting settings	160
DlgSettingsGeneral	
Dialog for editing general settings	161
DlgSettingsGridDisplay	
Dialog for editing grid display settings	162
DlgSettingsGridRemoval	
Dialog for editing grid removal settings	163
DlgSettingsMainWindow	
Dialog for editing main window settings, which are entirely independent of all documents	164
DlgSettingsPointMatch	
Dialog for editing point match settings, for DigitizeStatePointMatch	165
DlgSettingsSegments	
Dialog for editing Segments settings, for DigitizeStateSegment	166

DlgValidatorAbstract	
Abstract validator for all numeric formats	167
DlgValidatorDateTime	
Validator for numeric value expressed as date and/or time	168
DlgValidatorDegreesMinutesSeconds	
Validator for angles in real degrees, integer degrees and real minutes, or integer degrees with integer minutes with real seconds	168
DlgValidatorFactory	
Validator factory	169
DlgValidatorNumber	
Validator for generic (=simple) numbers	170
Document	
Storage of one imported image and the data attached to that image	171
DocumentHashGenerator	
Generates a DocumentHash value representing the state of the entire Document	175
DocumentModelAbstractBase	
Abstract base class for document models. This class enforces a common interface for the leaf subclasses	176
DocumentModelAxesChecker	
Model for DlgSettingsAxesChecker and CmdSettingsAxesChecker	177
DocumentModelColorFilter	
Model for DlgSettingsColorFilter and CmdSettingsColorFilter	178
DocumentModelCoords	
Model for DlgSettingsCoords and CmdSettingsCoords	181
DocumentModelDigitizeCurve	
Model for DlgSettingsDigitizeCurve and CmdSettingsDigitizeCurve	182
DocumentModelExportFormat	
Model for DlgSettingsExportFormat and CmdSettingsExportFormat	184
DocumentModelGeneral	
Model for DlgSettingsGeneral and CmdSettingsGeneral	185
DocumentModelGridDisplay	
Model for DlgSettingsGridDisplay and CmdSettingsGridDisplay	186
DocumentModelGridRemoval	
Model for DlgSettingsGridRemoval and CmdSettingsGridRemoval . The settings are unstable until the user approves	188
DocumentModelPointMatch	
Model for DlgSettingsPointMatch and CmdSettingsPointMatch	191
DocumentModelSegments	
Model for DlgSettingsSegments and CmdSettingsSegments	192
ExportAlignLinear	
Pick first simplest x value between specified min and max, for linear scaling	193
ExportAlignLog	
Pick first simplest x value between specified min and max, for log scaling	194
ExportFileAbstractBase	
Strategy base class for exporting to a file. This class provides common methods	195
ExportFileFunctions	
Strategy class for exporting to a file. This strategy is external to the Document class so that class is simpler	196
ExportFileRelations	
Strategy class for exporting to a file. This strategy is external to the Document class so that class is simpler	197
ExportImageForRegression	
Class for exporting during regression, when the Transformation has not yet been defined . . .	198
ExportOrdinalsSmooth	
Utility class to interpolate points spaced evenly along a piecewise defined curve with fitted spline	198
ExportOrdinalsStraight	
Utility class to interpolate points spaced evenly along a piecewise defined curve with line segments between points	199

ExportToClipboard	
Strategy class for exporting to the clipboard. This strategy is external to the Document class so that class is simpler	199
ExportToFile	
Strategy class for exporting to a file. This strategy is external to the Document class so that class is simpler	200
ExportXThetaValuesMergedFunctions	
Creates the set of merged x/theta values for exporting functions, using interpolation	201
FileCmdAbstract	
Base class for 'file commands' that are used specifically for regression testing of file import/open/export features	202
FileCmdClose	
Command for closing a file	203
FileCmdExport	
Command for exporting a file	204
FileCmdFactory	
Factory that creates FileCmds from a file cmd script file, in xml format	205
FileCmdImport	
Command for importing a file	205
FileCmdOpen	
Command for opening a file	206
FileCmdScript	
File that manages a command stack for regression testing of file import/open/export/close . . .	207
FilterImage	
Filters an image using a combination of color filtering and grid removal	207
FormatCoordsUnits	
Highest-level wrapper around other Formats classes	208
FormatCoordsUnitsStrategyAbstractBase	
Common methods for formatting strategies	208
FormatCoordsUnitsStrategyNonPolarTheta	
Format conversions between unformatted and formatted for CoordUnitsNonPolarTheta	209
FormatCoordsUnitsStrategyPolarTheta	
Format conversions between unformatted and formatted for CoordUnitsStrategyPolarTheta . .	210
FormatDateTime	
Input parsing and output formatting for date/time values	211
FormatDegreesMinutesSecondsBase	
Common input parsing and output formatting for degrees/minutes/seconds values	212
FormatDegreesMinutesSecondsNonPolarTheta	
Angular units according to CoordUnitsNonPolarTheta	213
FormatDegreesMinutesSecondsPolarTheta	
Angular units according to CoordUnitsPolarTheta	213
GeometryModel	
Model that adds row highlighting according to the currently select point identifier	214
GeometryStrategyAbstractBase	
Base class for all geometry strategies	215
GeometryStrategyContext	
Class for that manages geometry strategies	216
GeometryStrategyFunctionSmooth	
Calculate for line through the points that is smoothly connected as a function	217
GeometryStrategyFunctionStraight	
Calculate for line through the points that is straightly connected as a function	218
GeometryStrategyRelationSmooth	
Calculate for line through the points that is smoothly connected as a relation	219
GeometryStrategyRelationStraight	
Calculate for line through the points that is straightly connected as a relation	219
GeometryWindow	
Window that displays the geometry information, as a table, for the current curve	220

GhostEllipse	
Ghost for a QGraphicsEllipseItem	221
GhostPath	
Ghost for a QGraphicsPathItem	222
GhostPolygon	
Ghost for a QGraphicsPolygonItem	223
Ghosts	
Class for showing points and lines for all coordinate systems simultaneously, even though the code normally only allows graphical items for once coordinate system to be visible at a time	223
GraphicsArcItem	
Draw an arc as an ellipse but without lines from the center to the start and end points	224
GraphicsItemsExtractor	
This class consolidates utility routines that deal with graphics items that are getting extracted from the scene	225
GraphicsLinesForCurve	
This class stores the GraphicsLine objects for one Curve	226
GraphicsLinesForCurves	
This class stores the GraphicsLinesForCurves objects, one per Curve	227
GraphicsPoint	
Graphics item for drawing a circular or polygonal Point	229
GraphicsPointAbstractBase	
Base class for adding identifiers to graphics items that represent Points	230
GraphicsPointEllipse	
This class add event handling to QGraphicsEllipseItem	231
GraphicsPointFactory	
Factor for generating GraphicsPointAbstractBase class objects	232
GraphicsPointPolygon	
This class add event handling to QGraphicsPolygonItem	232
GraphicsScene	
Add point and line handling to generic QGraphicsScene	233
GraphicsView	
QGraphicsView class with event handling added. Typically the events are sent to the active digitizing state	235
GridClassifier	
Classify the grid pattern in an original image	237
GridHealer	
Class that 'heals' the curves after grid lines have been removed	238
GridInitializer	
This class initializes the count, start, step and stop parameters for one coordinate (either x/theta or y/range)	238
GridLine	
Single grid line drawn a straight or curved line	239
GridLineFactory	
Factory class for generating the points, composed of QGraphicsItem objects, along a GridLine	240
GridLineLimiter	
Limit the number of grid lines so a bad combination of start/step/stop value will not lead to extremely long delays when the step size is much too small for the start/stop values	241
GridLines	
Container class for GridLine objects	242
GridRemoval	
Strategy class for grid removal	242
HelpBrowser	
Text browser with resource loading enhanced for use as help text browser	243
HelpWindow	
Dockable help window	244
ImportCroppingUtilBase	
Utility class for import cropping options	244

ImportCroppingUtilNonPdf	
Import of non-pdf files	245
ImportCroppingUtilPdf	
Import of pdf files	246
Jpeg2000	
Wrapper around OpenJPEG library, in C, for opening jpeg2000 files	247
LineStyle	
Details for a specific Line	247
LoadFileInfo	
Returns information about files	249
LoadImageFromUrl	
Load QImage from url. This is trivial for a file, but requires an asynchronous download step for http urls	249
LoggerUpload	
Upload logging information to website for developer support	250
MainWindow	
Main window consisting of menu, graphics scene, status bar and optional toolbars as a Single Document Interface	251
MainWindowModel	
Model for DlgSettingsMainWindow	254
MigrateToVersion6	
Converts old (=pre version 6) enums to new (=version 6) enums, for reading of old document files	255
MimePoints	
Custom mime type for separate treatment of graph coordinates and, when there is no transform, points coordinates	256
NetworkClient	
Client for interacting with Engauge server	257
NonPdf	
Wrapper around the QImage class for read and importing non-PDF files	258
NonPdfCropping	
This class shows a frame around the selected portion of the import preview window	258
NonPdfFrameHandle	
This class acts as a single handle for the NonPdfCropping class	259
OrdinalGenerator	
Utility class for generating ordinal numbers	260
Pdf	
Wrapper around the Poppler library	260
PdfCropping	
This class shows a frame around the selected portion of the pdf import preview window	261
PdfFrameHandle	
This class acts as a single handle for the PdfCropping class	262
Point	
Class that represents one digitized point. The screen-to-graph coordinate transformation is al- ways external to this class	263
PointComparator	
Comparator for sorting Point class	265
PointIdentifiers	
Hash table class that tracks point identifiers as the key, with a corresponding boolean value . . .	265
PointMatchAlgorithm	
Algorithm returning a list of points that match the specified point	266
PointMatchPixel	
Single on or off pixel out of the pixels that define the point match mode's candidate point . . .	267
PointMatchTriplet	
Representation of one matched point as produced from the point match algorithm	268
PointStyle	
Details for a specific Point	268
Segment	
Selectable piecewise-defined line that follows a filtered line in the image	270

SegmentFactory	Factory class for Segment objects	271
SegmentLine	This class is a special case of the standard <code>QGraphicsLineItem</code> for segments	272
SettingsForGraph	Manage storage and retrieval of the settings for the curves	273
Spline	Cubic interpolation given independent and dependent value vectors	273
SplineCoeff	Four element vector of a,b,c,d coefficients and the associated x value, for one interval of a set of piecewise-defined intervals	275
SplinePair	Single X/Y pair for cubic spline interpolation initialization and calculations	276
StatusBar	Wrapper around <code>QStatusBar</code> to manage permanent widgets	277
TestCorrelation	Unit tests of fast correlation algorithm	278
TestFormats	Unit tests of formats	278
TestGraphCoords	Unit tests of graph coordinate sanity checking	279
TestProjectedPoint	Unit test of spline library	280
TestSegmentFill	Unit test of segment fill feature	280
TestSpline	Unit test of spline library	281
TestTransformation	Unit test of transformation class. Checking mostly involves verifying forward/reverse are inverses of each other	282
TestValidators	Unit tests of validators	282
Transformation	Affine transformation between screen and graph coordinates, based on digitized axis points . .	283
TransformationStateAbstractBase	Base class for all transformation states. This serves as an interface to TransformationState ← Context	285
TransformationStateContext	Context class for transformation state machine	286
TransformationStateDefined	Class to show transformation since transformation is defined	287
TransformationStateUndefined	Class to not show transformation since transformation is undefined	287
TranslatorContainer	Class that stores <code>QTranslator</code> objects for the duration of application execution	288
TutorialButton	Show a button with text for clicking ion. The button is implemented using layering of two graphics items (text and rectangle)	289
TutorialButtonRect	This class customizes <code>QGraphicsRectItem</code> so it performs a callback after a mouse event . . .	290
TutorialButtonText	This class customizes <code>QGraphicsTextItem</code> so it performs a callback after a mouse event	290
TutorialDlg	Tutorial using a strategy like a comic strip with decision points deciding which panels appear .	291
TutorialStateAbstractBase	One state manages one panel of the tutorial	292
TutorialStateAxisPoints	Axis points panel discusses axis point digitization	293

TutorialStateChecklistWizardAbstract	
Abstract class that supports checklist wizard panels	294
TutorialStateChecklistWizardLines	
Checklist wizard panel for lines discusses the checklist wizard, and returns to TRANSITION_↔	
STATE_SEGMENT_FILL	295
TutorialStateChecklistWizardPoints	
Checklist wizard panel for points discusses the checklist wizard, and returns to TRANSITION_↔	
_STATE_POINT_MATCH	296
TutorialStateColorFilter	
Color filter panel discusses the curve-specific color filtering	297
TutorialStateContext	
Context class for tutorial state machine	298
TutorialStateCurveSelection	
Curve selection panel discusses how to select a curve, and perform setup on the selected curve	299
TutorialStateCurveType	
Curve type state/panel lets user select the curve type (lines or points)	300
TutorialStateIntroduction	
Introduction state/panel is the first panel the user sees	301
TutorialStatePointMatch	
Point match panel discusses the matching of points in curves without lines	302
TutorialStateSegmentFill	
Segment fill panel discusses the digitization of points along curve lines	303
ViewPointStyle	
Class that displays a view of the current Curve's point style	304
ViewPreview	
Class that modifies QGraphicsView to automatically expand/shrink the view to fit the window,	
after resize events	305
ViewProfile	
Class that modifies QGraphicsView to present a two-dimensional profile, with movable dividers	
for selecting a range	306
ViewProfileDivider	
Divider that can be dragged, in a dialog QGraphicsView	307
ViewProfileScale	
Linear horizontal scale, with the spectrum reflecting the active filter parameter	308
ViewSegmentFilter	
Class that displays the current Segment Filter in a MainWindow toolbar	308

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Background/ BackgroundImage.cpp	??
Background/ BackgroundImage.h	??
Background/ BackgroundStateAbstractBase.cpp	??
Background/ BackgroundStateAbstractBase.h	??
Background/ BackgroundStateContext.cpp	??
Background/ BackgroundStateContext.h	??
Background/ BackgroundStateCurve.cpp	??
Background/ BackgroundStateCurve.h	??
Background/ BackgroundStateNone.cpp	??
Background/ BackgroundStateNone.h	??
Background/ BackgroundStateOriginal.cpp	??
Background/ BackgroundStateOriginal.h	??
Background/ BackgroundStateUnloaded.cpp	??
Background/ BackgroundStateUnloaded.h	??
Callback/ CallbackAddPointsInCurvesGraphs.cpp	??
Callback/ CallbackAddPointsInCurvesGraphs.h	??
Callback/ CallbackAxesCheckerFromAxesPoints.cpp	??
Callback/ CallbackAxesCheckerFromAxesPoints.h	??
Callback/ CallbackAxisPointsAbstract.cpp	??
Callback/ CallbackAxisPointsAbstract.h	??
Callback/ CallbackBoundingRects.cpp	??
Callback/ CallbackBoundingRects.h	??
Callback/ CallbackCheckAddPointAxis.cpp	??
Callback/ CallbackCheckAddPointAxis.h	??
Callback/ CallbackCheckEditPointAxis.cpp	??
Callback/ CallbackCheckEditPointAxis.h	??
Callback/ CallbackDocumentHash.cpp	??
Callback/ CallbackDocumentHash.h	??
Callback/ CallbackGatherXThetaValuesFunctions.cpp	??
Callback/ CallbackGatherXThetaValuesFunctions.h	??
Callback/ CallbackNextOrdinal.cpp	??
Callback/ CallbackNextOrdinal.h	??
Callback/ CallbackPointOrdinal.cpp	??
Callback/ CallbackPointOrdinal.h	??
Callback/ CallbackRemovePointsInCurvesGraphs.cpp	??

Callback/ CallbackRemovePointsInCurvesGraphs.h	??
Callback/ CallbackSceneUpdateAfterCommand.cpp	??
Callback/ CallbackSceneUpdateAfterCommand.h	??
Callback/ CallbackSearchReturn.h	311
Callback/ CallbackUpdateTransform.cpp	??
Callback/ CallbackUpdateTransform.h	??
Checker/ Checker.cpp	??
Checker/ Checker.h	??
Checker/ CheckerMode.cpp	??
Checker/ CheckerMode.h	??
Checklist/ ChecklistGuide.cpp	??
Checklist/ ChecklistGuide.h	??
Checklist/ ChecklistGuideBrowser.cpp	??
Checklist/ ChecklistGuideBrowser.h	??
Checklist/ ChecklistGuidePage.cpp	??
Checklist/ ChecklistGuidePage.h	??
Checklist/ ChecklistGuidePageConclusion.cpp	??
Checklist/ ChecklistGuidePageConclusion.h	??
Checklist/ ChecklistGuidePageCurves.cpp	??
Checklist/ ChecklistGuidePageCurves.h	??
Checklist/ ChecklistGuidePageIntro.cpp	??
Checklist/ ChecklistGuidePageIntro.h	??
Checklist/ ChecklistGuideWizard.cpp	??
Checklist/ ChecklistGuideWizard.h	??
Checklist/ ChecklistLineEdit.cpp	??
Checklist/ ChecklistLineEdit.h	??
Checklist/ ChecklistTemplate.cpp	??
Checklist/ ChecklistTemplate.h	??
Cmd/ CmdAbstract.cpp	??
Cmd/ CmdAbstract.h	??
Cmd/ CmdAddPointAxis.cpp	??
Cmd/ CmdAddPointAxis.h	??
Cmd/ CmdAddPointGraph.cpp	??
Cmd/ CmdAddPointGraph.h	??
Cmd/ CmdAddPointsGraph.cpp	??
Cmd/ CmdAddPointsGraph.h	??
Cmd/ CmdCopy.cpp	??
Cmd/ CmdCopy.h	??
Cmd/ CmdCut.cpp	??
Cmd/ CmdCut.h	??
Cmd/ CmdDelete.cpp	??
Cmd/ CmdDelete.h	??
Cmd/ CmdEditPointAxis.cpp	??
Cmd/ CmdEditPointAxis.h	??
Cmd/ CmdEditPointGraph.cpp	??
Cmd/ CmdEditPointGraph.h	??
Cmd/ CmdFactory.cpp	??
Cmd/ CmdFactory.h	??
Cmd/ CmdMediator.cpp	??
Cmd/ CmdMediator.h	??
Cmd/ CmdMoveBy.cpp	??
Cmd/ CmdMoveBy.h	??
Cmd/ CmdPaste.cpp	??
Cmd/ CmdPaste.h	??
Cmd/ CmdPointChangeBase.cpp	??
Cmd/ CmdPointChangeBase.h	??
Cmd/ CmdRedoForTest.cpp	??
Cmd/ CmdRedoForTest.h	??

Cmd/CmdSelectCoordSystem.cpp	??
Cmd/CmdSelectCoordSystem.h	??
Cmd/CmdSettingsAxesChecker.cpp	??
Cmd/CmdSettingsAxesChecker.h	??
Cmd/CmdSettingsColorFilter.cpp	??
Cmd/CmdSettingsColorFilter.h	??
Cmd/CmdSettingsCoords.cpp	??
Cmd/CmdSettingsCoords.h	??
Cmd/CmdSettingsCurveAddRemove.cpp	??
Cmd/CmdSettingsCurveAddRemove.h	??
Cmd/CmdSettingsCurveProperties.cpp	??
Cmd/CmdSettingsCurveProperties.h	??
Cmd/CmdSettingsDigitizeCurve.cpp	??
Cmd/CmdSettingsDigitizeCurve.h	??
Cmd/CmdSettingsExportFormat.cpp	??
Cmd/CmdSettingsExportFormat.h	??
Cmd/CmdSettingsGeneral.cpp	??
Cmd/CmdSettingsGeneral.h	??
Cmd/CmdSettingsGridDisplay.cpp	??
Cmd/CmdSettingsGridDisplay.h	??
Cmd/CmdSettingsGridRemoval.cpp	??
Cmd/CmdSettingsGridRemoval.h	??
Cmd/CmdSettingsPointMatch.cpp	??
Cmd/CmdSettingsPointMatch.h	??
Cmd/CmdSettingsSegments.cpp	??
Cmd/CmdSettingsSegments.h	??
Cmd/CmdStackShadow.cpp	??
Cmd/CmdStackShadow.h	??
Cmd/CmdUndoForTest.cpp	??
Cmd/CmdUndoForTest.h	??
Color/ColorConstants.h	??
Color/ColorFilter.cpp	??
Color/ColorFilter.h	??
Color/ColorFilterEntry.h	??
Color/ColorFilterHistogram.cpp	??
Color/ColorFilterHistogram.h	??
Color/ColorFilterMode.cpp	??
Color/ColorFilterMode.h	??
Color/ColorFilterSettings.cpp	??
Color/ColorFilterSettings.h	??
Color/ColorFilterSettingsStrategyAbstractBase.cpp	??
Color/ColorFilterSettingsStrategyAbstractBase.h	??
Color/ColorFilterSettingsStrategyForeground.cpp	??
Color/ColorFilterSettingsStrategyForeground.h	??
Color/ColorFilterSettingsStrategyHue.cpp	??
Color/ColorFilterSettingsStrategyHue.h	??
Color/ColorFilterSettingsStrategyIntensity.cpp	??
Color/ColorFilterSettingsStrategyIntensity.h	??
Color/ColorFilterSettingsStrategySaturation.cpp	??
Color/ColorFilterSettingsStrategySaturation.h	??
Color/ColorFilterSettingsStrategyValue.cpp	??
Color/ColorFilterSettingsStrategyValue.h	??
Color/ColorFilterStrategyAbstractBase.cpp	??
Color/ColorFilterStrategyAbstractBase.h	??
Color/ColorFilterStrategyForeground.cpp	??
Color/ColorFilterStrategyForeground.h	??
Color/ColorFilterStrategyHue.cpp	??
Color/ColorFilterStrategyHue.h	??

Color/ColorFilterStrategyIntensity.cpp	??
Color/ColorFilterStrategyIntensity.h	??
Color/ColorFilterStrategySaturation.cpp	??
Color/ColorFilterStrategySaturation.h	??
Color/ColorFilterStrategyValue.cpp	??
Color/ColorFilterStrategyValue.h	??
Color/ColorPalette.cpp	??
Color/ColorPalette.h	??
Coord/CoordScale.cpp	??
Coord/CoordScale.h	??
Coord/CoordsType.cpp	??
Coord/CoordsType.h	??
Coord/CoordSymbol.cpp	??
Coord/CoordSymbol.h	??
Coord/CoordUnitsDate.cpp	??
Coord/CoordUnitsDate.h	??
Coord/CoordUnitsNonPolarTheta.cpp	??
Coord/CoordUnitsNonPolarTheta.h	??
Coord/CoordUnitsPolarTheta.cpp	??
Coord/CoordUnitsPolarTheta.h	??
Coord/CoordUnitsTime.cpp	??
Coord/CoordUnitsTime.h	??
CoordSystem/CoordSystem.cpp	??
CoordSystem/CoordSystem.h	??
CoordSystem/CoordSystemContext.cpp	??
CoordSystem/CoordSystemContext.h	??
CoordSystem/CoordSystemIndex.h	??
CoordSystem/CoordSystemInterface.cpp	??
CoordSystem/CoordSystemInterface.h	??
Correlation/Correlation.cpp	??
Correlation/Correlation.h	??
Cursor/CursorFactory.cpp	??
Cursor/CursorFactory.h	??
Cursor/CursorSize.cpp	??
Cursor/CursorSize.h	??
Curve/Curve.cpp	??
Curve/Curve.h	??
Curve/CurveConnectAs.cpp	??
Curve/CurveConnectAs.h	??
Curve/CurveNameList.cpp	??
Curve/CurveNameList.h	??
Curve/CurveNameListEntry.cpp	??
Curve/CurveNameListEntry.h	??
Curve/CurveSettingsInt.cpp	??
Curve/CurveSettingsInt.h	??
Curve/CurvesGraphs.cpp	??
Curve/CurvesGraphs.h	??
Curve/CurveStyle.cpp	??
Curve/CurveStyle.h	??
Curve/CurveStyles.cpp	??
Curve/CurveStyles.h	??
DigitizeState/DigitizeStateAbstractBase.cpp	??
DigitizeState/DigitizeStateAbstractBase.h	??
DigitizeState/DigitizeStateAxis.cpp	??
DigitizeState/DigitizeStateAxis.h	??
DigitizeState/DigitizeStateColorPicker.cpp	??
DigitizeState/DigitizeStateColorPicker.h	??
DigitizeState/DigitizeStateContext.cpp	??

DigitizeState/DigitizeStateContext.h	??
DigitizeState/DigitizeStateCurve.cpp	??
DigitizeState/DigitizeStateCurve.h	??
DigitizeState/DigitizeStateEmpty.cpp	??
DigitizeState/DigitizeStateEmpty.h	??
DigitizeState/DigitizeStatePointMatch.cpp	??
DigitizeState/DigitizeStatePointMatch.h	??
DigitizeState/DigitizeStateSegment.cpp	??
DigitizeState/DigitizeStateSegment.h	??
DigitizeState/DigitizeStateSelect.cpp	??
DigitizeState/DigitizeStateSelect.h	??
Dlg/DlgAbout.cpp	??
Dlg/DlgAbout.h	??
Dlg/DlgEditPointAxis.cpp	??
Dlg/DlgEditPointAxis.h	??
Dlg/DlgEditPointGraph.cpp	??
Dlg/DlgEditPointGraph.h	??
Dlg/DlgEditPointGraphLineEdit.cpp	??
Dlg/DlgEditPointGraphLineEdit.h	??
Dlg/DlgErrorReport.cpp	??
Dlg/DlgErrorReport.h	??
Dlg/DlgFilterCommand.cpp	??
Dlg/DlgFilterCommand.h	??
Dlg/DlgFilterThread.cpp	??
Dlg/DlgFilterThread.h	??
Dlg/DlgFilterWorker.cpp	??
Dlg/DlgFilterWorker.h	??
Dlg/DlgImportAdvanced.cpp	??
Dlg/DlgImportAdvanced.h	??
Dlg/DlgImportCroppingNonPdf.cpp	??
Dlg/DlgImportCroppingNonPdf.h	??
Dlg/DlgImportCroppingPdf.cpp	??
Dlg/DlgImportCroppingPdf.h	??
Dlg/DlgRequiresTransform.cpp	??
Dlg/DlgRequiresTransform.h	??
Dlg/DlgSettingsAbstractBase.cpp	??
Dlg/DlgSettingsAbstractBase.h	??
Dlg/DlgSettingsAxesChecker.cpp	??
Dlg/DlgSettingsAxesChecker.h	??
Dlg/DlgSettingsColorFilter.cpp	??
Dlg/DlgSettingsColorFilter.h	??
Dlg/DlgSettingsCoords.cpp	??
Dlg/DlgSettingsCoords.h	??
Dlg/DlgSettingsCurveAddRemove.cpp	??
Dlg/DlgSettingsCurveAddRemove.h	??
Dlg/DlgSettingsCurveProperties.cpp	??
Dlg/DlgSettingsCurveProperties.h	??
Dlg/DlgSettingsDigitizeCurve.cpp	??
Dlg/DlgSettingsDigitizeCurve.h	??
Dlg/DlgSettingsExportFormat.cpp	??
Dlg/DlgSettingsExportFormat.h	??
Dlg/DlgSettingsGeneral.cpp	??
Dlg/DlgSettingsGeneral.h	??
Dlg/DlgSettingsGridDisplay.cpp	??
Dlg/DlgSettingsGridDisplay.h	??
Dlg/DlgSettingsGridRemoval.cpp	??
Dlg/DlgSettingsGridRemoval.h	??
Dlg/DlgSettingsMainWindow.cpp	??

Dlg/DlgSettingsMainWindow.h	??
Dlg/DlgSettingsPointMatch.cpp	??
Dlg/DlgSettingsPointMatch.h	??
Dlg/DlgSettingsSegments.cpp	??
Dlg/DlgSettingsSegments.h	??
Dlg/DlgValidatorAbstract.cpp	??
Dlg/DlgValidatorAbstract.h	??
Dlg/DlgValidatorDateTime.cpp	??
Dlg/DlgValidatorDateTime.h	??
Dlg/DlgValidatorDegreesMinutesSeconds.cpp	??
Dlg/DlgValidatorDegreesMinutesSeconds.h	??
Dlg/DlgValidatorFactory.cpp	??
Dlg/DlgValidatorFactory.h	??
Dlg/DlgValidatorNumber.cpp	??
Dlg/DlgValidatorNumber.h	??
Document/Document.cpp	??
Document/Document.h	??
Document/DocumentAxesPointsRequired.h	??
Document/DocumentHash.h	??
Document/DocumentHashGenerator.cpp	??
Document/DocumentHashGenerator.h	??
Document/DocumentModelAbstractBase.cpp	??
Document/DocumentModelAbstractBase.h	??
Document/DocumentModelAxesChecker.cpp	??
Document/DocumentModelAxesChecker.h	??
Document/DocumentModelColorFilter.cpp	??
Document/DocumentModelColorFilter.h	??
Document/DocumentModelCoords.cpp	??
Document/DocumentModelCoords.h	??
Document/DocumentModelDigitizeCurve.cpp	??
Document/DocumentModelDigitizeCurve.h	??
Document/DocumentModelExportFormat.cpp	??
Document/DocumentModelExportFormat.h	??
Document/DocumentModelGeneral.cpp	??
Document/DocumentModelGeneral.h	??
Document/DocumentModelGridDisplay.cpp	??
Document/DocumentModelGridDisplay.h	??
Document/DocumentModelGridRemoval.cpp	??
Document/DocumentModelGridRemoval.h	??
Document/DocumentModelPointMatch.cpp	??
Document/DocumentModelPointMatch.h	??
Document/DocumentModelSegments.cpp	??
Document/DocumentModelSegments.h	??
Document/DocumentSerialize.cpp	??
Document/DocumentSerialize.h	??
Export/ExportAlignLinear.cpp	??
Export/ExportAlignLinear.h	??
Export/ExportAlignLog.cpp	??
Export/ExportAlignLog.h	??
Export/ExportDelimiter.cpp	??
Export/ExportDelimiter.h	??
Export/ExportFileAbstractBase.cpp	??
Export/ExportFileAbstractBase.h	??
Export/ExportFileFunctions.cpp	??
Export/ExportFileFunctions.h	??
Export/ExportFileRelations.cpp	??
Export/ExportFileRelations.h	??
Export/ExportHeader.cpp	??

Export/ExportHeader.h	??
Export/ExportImageForRegression.cpp	??
Export/ExportImageForRegression.h	??
Export/ExportLayoutFunctions.cpp	??
Export/ExportLayoutFunctions.h	??
Export/ExportOrdinalsSmooth.cpp	??
Export/ExportOrdinalsSmooth.h	??
Export/ExportOrdinalsStraight.cpp	??
Export/ExportOrdinalsStraight.h	??
Export/ExportPointsIntervalUnits.cpp	??
Export/ExportPointsIntervalUnits.h	??
Export/ExportPointsSelectionFunctions.cpp	??
Export/ExportPointsSelectionFunctions.h	??
Export/ExportPointsSelectionRelations.cpp	??
Export/ExportPointsSelectionRelations.h	??
Export/ExportToClipboard.cpp	??
Export/ExportToClipboard.h	??
Export/ExportToFile.cpp	??
Export/ExportToFile.h	??
Export/ExportValuesOrdinal.h	??
Export/ExportValuesXOrY.h	??
Export/ExportXThetaValuesMergedFunctions.cpp	??
Export/ExportXThetaValuesMergedFunctions.h	??
FileCmd/FileCmdAbstract.cpp	??
FileCmd/FileCmdAbstract.h	??
FileCmd/FileCmdClose.cpp	??
FileCmd/FileCmdClose.h	??
FileCmd/FileCmdExport.cpp	??
FileCmd/FileCmdExport.h	??
FileCmd/FileCmdFactory.cpp	??
FileCmd/FileCmdFactory.h	??
FileCmd/FileCmdImport.cpp	??
FileCmd/FileCmdImport.h	??
FileCmd/FileCmdOpen.cpp	??
FileCmd/FileCmdOpen.h	??
FileCmd/FileCmdScript.cpp	??
FileCmd/FileCmdScript.h	??
FileCmd/FileCmdSerialize.cpp	??
FileCmd/FileCmdSerialize.h	??
Filter/FilterImage.cpp	??
Filter/FilterImage.h	??
Format/FormatCoordsUnits.cpp	??
Format/FormatCoordsUnits.h	??
Format/FormatCoordsUnitsStrategyAbstractBase.cpp	??
Format/FormatCoordsUnitsStrategyAbstractBase.h	??
Format/FormatCoordsUnitsStrategyNonPolarTheta.cpp	??
Format/FormatCoordsUnitsStrategyNonPolarTheta.h	??
Format/FormatCoordsUnitsStrategyPolarTheta.cpp	??
Format/FormatCoordsUnitsStrategyPolarTheta.h	??
Format/FormatDateTime.cpp	??
Format/FormatDateTime.h	??
Format/FormatDegreesMinutesSecondsBase.cpp	??
Format/FormatDegreesMinutesSecondsBase.h	??
Format/FormatDegreesMinutesSecondsNonPolarTheta.cpp	??
Format/FormatDegreesMinutesSecondsNonPolarTheta.h	??
Format/FormatDegreesMinutesSecondsPolarTheta.cpp	??
Format/FormatDegreesMinutesSecondsPolarTheta.h	??
Geometry/GeometryModel.cpp	??

Geometry/GeometryModel.h	??
Geometry/GeometryStrategyAbstractBase.cpp	??
Geometry/GeometryStrategyAbstractBase.h	??
Geometry/GeometryStrategyContext.cpp	??
Geometry/GeometryStrategyContext.h	??
Geometry/GeometryStrategyFunctionSmooth.cpp	??
Geometry/GeometryStrategyFunctionSmooth.h	??
Geometry/GeometryStrategyFunctionStraight.cpp	??
Geometry/GeometryStrategyFunctionStraight.h	??
Geometry/GeometryStrategyRelationSmooth.cpp	??
Geometry/GeometryStrategyRelationSmooth.h	??
Geometry/GeometryStrategyRelationStraight.cpp	??
Geometry/GeometryStrategyRelationStraight.h	??
Geometry/GeometryWindow.cpp	??
Geometry/GeometryWindow.h	??
Ghosts/GhostEllipse.cpp	??
Ghosts/GhostEllipse.h	??
Ghosts/GhostPath.cpp	??
Ghosts/GhostPath.h	??
Ghosts/GhostPolygon.cpp	??
Ghosts/GhostPolygon.h	??
Ghosts/Ghosts.cpp	??
Ghosts/Ghosts.h	??
Graphics/GraphicsArcItem.cpp	??
Graphics/GraphicsArcItem.h	??
Graphics/GraphicsItemsExtractor.cpp	??
Graphics/GraphicsItemsExtractor.h	??
Graphics/GraphicsItemType.h	??
Graphics/GraphicsLinesForCurve.cpp	??
Graphics/GraphicsLinesForCurve.h	??
Graphics/GraphicsLinesForCurves.cpp	??
Graphics/GraphicsLinesForCurves.h	??
Graphics/GraphicsPoint.cpp	??
Graphics/GraphicsPoint.h	??
Graphics/GraphicsPointAbstractBase.cpp	??
Graphics/GraphicsPointAbstractBase.h	??
Graphics/GraphicsPointEllipse.cpp	??
Graphics/GraphicsPointEllipse.h	??
Graphics/GraphicsPointFactory.cpp	??
Graphics/GraphicsPointFactory.h	??
Graphics/GraphicsPointPolygon.cpp	??
Graphics/GraphicsPointPolygon.h	??
Graphics/GraphicsScene.cpp	??
Graphics/GraphicsScene.h	??
Graphics/GraphicsView.cpp	??
Graphics/GraphicsView.h	??
Grid/GridClassifier.cpp	??
Grid/GridClassifier.h	??
Grid/GridCoordDisable.cpp	??
Grid/GridCoordDisable.h	??
Grid/GridHealer.cpp	??
Grid/GridHealer.h	??
Grid/GridInitializer.cpp	??
Grid/GridInitializer.h	??
Grid/GridLine.cpp	??
Grid/GridLine.h	??
Grid/GridLineFactory.cpp	??
Grid/GridLineFactory.h	??

Grid/GridLineLimiter.cpp	??
Grid/GridLineLimiter.h	??
Grid/GridLines.cpp	??
Grid/GridLines.h	??
Grid/GridLineStyle.h	??
Grid/GridRemoval.cpp	??
Grid/GridRemoval.h	??
Help/HelpBrowser.cpp	??
Help/HelpBrowser.h	??
Help/HelpWindow.cpp	??
Help/HelpWindow.h	??
Import/ImportCropping.cpp	??
Import/ImportCropping.h	??
Import/ImportCroppingUtilBase.cpp	??
Import/ImportCroppingUtilBase.h	??
Import/ImportCroppingUtilNonPdf.cpp	??
Import/ImportCroppingUtilNonPdf.h	??
Import/ImportCroppingUtilPdf.cpp	??
Import/ImportCroppingUtilPdf.h	??
include/EngaugeAssert.h	??
include/ValuesVectorXOrY.h	??
Jpeg2000/Jpeg2000.cpp	??
Jpeg2000/Jpeg2000.h	??
Jpeg2000/Jpeg2000Callbacks.cpp	??
Jpeg2000/Jpeg2000Callbacks.h	??
Jpeg2000/Jpeg2000Color.cpp	??
Jpeg2000/Jpeg2000Color.h	??
Jpeg2000/Jpeg2000Convert.cpp	??
Jpeg2000/Jpeg2000Convert.h	??
Jpeg2000/Jpeg2000FormatDefs.h	??
Line/LineStyle.cpp	??
Line/LineStyle.h	??
Load/LoadFileInfo.cpp	??
Load/LoadFileInfo.h	??
Load/LoadImageFromUrl.cpp	??
Load/LoadImageFromUrl.h	??
Logger/Logger.cpp	??
Logger/Logger.h	??
Logger/LoggerUpload.cpp	??
Logger/LoggerUpload.h	??
main/main.cpp	??
main/MainTitleBarFormat.h	??
main/MainWindow.cpp	??
main/MainWindow.h	??
main/MainWindowModel.cpp	??
main/MainWindowModel.h	??
Mime/MimePoints.cpp	??
Mime/MimePoints.h	??
Network/NetworkClient.cpp	??
Network/NetworkClient.h	??
NonPdf/NonPdf.cpp	??
NonPdf/NonPdf.h	??
NonPdf/NonPdfCropping.cpp	??
NonPdf/NonPdfCropping.h	??
NonPdf/NonPdfFrameHandle.cpp	??
NonPdf/NonPdfFrameHandle.h	??
Ordinal/OrdinalGenerator.cpp	??
Ordinal/OrdinalGenerator.h	??

Ordinal/ OrdinalToGraphicsPoint.h	??
Pdf/ Pdf.cpp	??
Pdf/ Pdf.h	??
Pdf/ PdfCropping.cpp	??
Pdf/ PdfCropping.h	??
Pdf/ PdfFrameHandle.cpp	??
Pdf/ PdfFrameHandle.h	??
Pdf/ PdfResolution.cpp	??
Pdf/ PdfResolution.h	??
Point/ Point.cpp	??
Point/ Point.h	??
Point/ PointComparator.h	??
Point/ PointIdentifiers.cpp	??
Point/ PointIdentifiers.h	??
Point/ PointMatchAlgorithm.cpp	??
Point/ PointMatchAlgorithm.h	??
Point/ PointMatchPixel.cpp	??
Point/ PointMatchPixel.h	??
Point/ PointMatchTriplet.cpp	??
Point/ PointMatchTriplet.h	??
Point/ Points.h	??
Point/ PointShape.cpp	??
Point/ PointShape.h	??
Point/ PointStyle.cpp	??
Point/ PointStyle.h	??
Segment/ Segment.cpp	??
Segment/ Segment.h	??
Segment/ SegmentFactory.cpp	??
Segment/ SegmentFactory.h	??
Segment/ SegmentLine.cpp	??
Segment/ SegmentLine.h	??
Settings/ Settings.cpp	??
Settings/ Settings.h	??
Settings/ SettingsForGraph.cpp	??
Settings/ SettingsForGraph.h	??
Spline/ Spline.cpp	??
Spline/ Spline.h	??
Spline/ SplineCoeff.cpp	??
Spline/ SplineCoeff.h	??
Spline/ SplinePair.cpp	??
Spline/ SplinePair.h	??
StatusBar/ StatusBar.cpp	??
StatusBar/ StatusBar.h	??
StatusBar/ StatusBarMode.h	??
Test/ TestCorrelation.cpp	??
Test/ TestCorrelation.h	??
Test/ TestFormats.cpp	??
Test/ TestFormats.h	??
Test/ TestGraphCoords.cpp	??
Test/ TestGraphCoords.h	??
Test/ TestProjectedPoint.cpp	??
Test/ TestProjectedPoint.h	??
Test/ TestSegmentFill.cpp	??
Test/ TestSegmentFill.h	??
Test/ TestSpline.cpp	??
Test/ TestSpline.h	??
Test/ TestTransformation.cpp	??
Test/ TestTransformation.h	??

Test/TestValidators.cpp	??
Test/TestValidators.h	??
Transformation/Transformation.cpp	??
Transformation/Transformation.h	??
Transformation/TransformationStateAbstractBase.cpp	??
Transformation/TransformationStateAbstractBase.h	??
Transformation/TransformationStateContext.cpp	??
Transformation/TransformationStateContext.h	??
Transformation/TransformationStateDefined.cpp	??
Transformation/TransformationStateDefined.h	??
Transformation/TransformationStateUndefined.cpp	??
Transformation/TransformationStateUndefined.h	??
Translator/TranslatorContainer.cpp	??
Translator/TranslatorContainer.h	??
Tutorial/TutorialButton.cpp	??
Tutorial/TutorialButton.h	??
Tutorial/TutorialButtonRect.cpp	??
Tutorial/TutorialButtonRect.h	??
Tutorial/TutorialButtonText.cpp	??
Tutorial/TutorialButtonText.h	??
Tutorial/TutorialDlg.cpp	??
Tutorial/TutorialDlg.h	??
Tutorial/TutorialStateAbstractBase.cpp	??
Tutorial/TutorialStateAbstractBase.h	??
Tutorial/TutorialStateAxisPoints.cpp	??
Tutorial/TutorialStateAxisPoints.h	??
Tutorial/TutorialStateChecklistWizardAbstract.cpp	??
Tutorial/TutorialStateChecklistWizardAbstract.h	??
Tutorial/TutorialStateChecklistWizardLines.cpp	??
Tutorial/TutorialStateChecklistWizardLines.h	??
Tutorial/TutorialStateChecklistWizardPoints.cpp	??
Tutorial/TutorialStateChecklistWizardPoints.h	??
Tutorial/TutorialStateColorFilter.cpp	??
Tutorial/TutorialStateColorFilter.h	??
Tutorial/TutorialStateContext.cpp	??
Tutorial/TutorialStateContext.h	??
Tutorial/TutorialStateCurveSelection.cpp	??
Tutorial/TutorialStateCurveSelection.h	??
Tutorial/TutorialStateCurveType.cpp	??
Tutorial/TutorialStateCurveType.h	??
Tutorial/TutorialStateIntroduction.cpp	??
Tutorial/TutorialStateIntroduction.h	??
Tutorial/TutorialStatePointMatch.cpp	??
Tutorial/TutorialStatePointMatch.h	??
Tutorial/TutorialStateSegmentFill.cpp	??
Tutorial/TutorialStateSegmentFill.h	??
util/DataKey.cpp	??
util/DataKey.h	??
util/EnumsToQt.cpp	??
util/EnumsToQt.h	??
util/MigrateToVersion6.cpp	??
util/MigrateToVersion6.h	??
util/mmsubs.cpp	??
util/mmsubs.h	??
util/QtToString.cpp	??
util/QtToString.h	??
util/Version.cpp	??
util/Version.h	??

util/ Xml.cpp	??
util/ Xml.h	??
util/ ZValues.cpp	??
util/ ZValues.h	??
View/ ViewPointStyle.cpp	??
View/ ViewPointStyle.h	??
View/ ViewPreview.cpp	??
View/ ViewPreview.h	??
View/ ViewProfile.cpp	??
View/ ViewProfile.h	??
View/ ViewProfileDivider.cpp	??
View/ ViewProfileDivider.h	??
View/ ViewProfileParameters.cpp	??
View/ ViewProfileParameters.h	??
View/ ViewProfileScale.cpp	??
View/ ViewProfileScale.h	??
View/ ViewSegmentFilter.cpp	??
View/ ViewSegmentFilter.h	??
Zoom/ ZoomControl.h	??
Zoom/ ZoomFactor.h	??
Zoom/ ZoomFactorInitial.h	??
Zoom/ ZoomLabels.cpp	??
Zoom/ ZoomLabels.h	??

Chapter 5

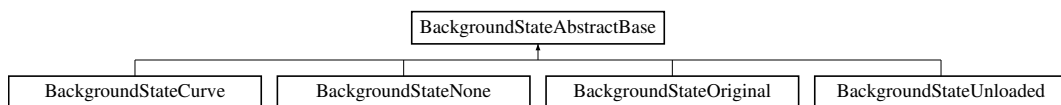
Class Documentation

5.1 BackgroundStateAbstractBase Class Reference

Background image state machine state base class.

```
#include <BackgroundStateAbstractBase.h>
```

Inheritance diagram for BackgroundStateAbstractBase:



Public Member Functions

- [BackgroundStateAbstractBase](#) ([BackgroundStateContext](#) &context, [GraphicsScene](#) &scene)
Single constructor.
- virtual void [begin](#) ()=0
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- [BackgroundStateContext](#) & [context](#) ()
Reference to the [BackgroundStateContext](#) that contains all the [BackgroundStateAbstractBase](#) subclasses, without const.
- const [BackgroundStateContext](#) & [context](#) () const
Reference to the [BackgroundStateContext](#) that contains all the [BackgroundStateAbstractBase](#) subclasses, without const.
- virtual void [end](#) ()=0
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [fitInView](#) ([GraphicsView](#) &view)=0
Zoom so background fills the window.
- QImage [image](#) () const
Image for the current state.
- QGraphicsPixmapItem & [imageItem](#) () const
Graphics image item for the current state.
- [GraphicsScene](#) & [scene](#) ()
Reference to the [GraphicsScene](#), without const.

- const [GraphicsScene](#) & [scene](#) () const
Reference to the [GraphicsScene](#), without const.
- virtual void [setCurveSelected](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)=0
Update the currently selected curve name.
- virtual void [setPixmap](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QPixmap &pixmap, const QString &curveSelected)=0
Update the image for this state, after the leaf class processes it appropriately.
- virtual QString [state](#) () const =0
State name for debugging.
- virtual void [updateColorFilter](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QString &curveSelected)=0
Apply color filter settings.

Protected Member Functions

- void [setImageVisible](#) (bool visible)
Show/hide background image.
- void [setProcessedPixmap](#) (const QPixmap &pixmap)
Save the image for this state after it has been processed by the leaf class.

5.1.1 Detailed Description

Background image state machine state base class.

Definition at line 30 of file BackgroundStateAbstractBase.h.

The documentation for this class was generated from the following files:

- Background/BackgroundStateAbstractBase.h
- Background/BackgroundStateAbstractBase.cpp

5.2 BackgroundStateContext Class Reference

Context class that manages the background image state machine.

```
#include <BackgroundStateContext.h>
```

Public Member Functions

- [BackgroundStateContext](#) ([MainWindow](#) &mainWindow)
Single constructor.
- void [close](#) ()
Open [Document](#) is being closed so remove the background.
- void [fitInView](#) ([GraphicsView](#) &view)
Zoom so background fills the window.
- QImage [imageForCurveState](#) () const
Image for the [Curve](#) state, even if the current state is different.
- void [requestStateTransition](#) (BackgroundState backgroundState)
Initiate state transition to be performed later, when BackgroundState is off the stack.
- void [setBackgroundImage](#) (BackgroundImage backgroundImage)
Transition to the specified state. This method is used by classes outside of the state machine to trigger transitions.
- void [setCurveSelected](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QString &curveSelected)
Update the selected curve.
- void [setPixmap](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QPixmap &pixmapOriginal, const QString &curveSelected)
Update the images of all states, rather than just the current state.
- void [updateColorFilter](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Apply color filter settings.

5.2.1 Detailed Description

Context class that manages the background image state machine.

Overall strategy is that changing the currently selected curve should not affect the background image if the original image is being shown, or no image is being shown. However, if the curve-specific color filter image is being shown, then it should be replaced by the filtered image specific to the new curve.

Other considerations are that the processing should be robust in terms of ordering of the following incoming events:

1. State transitions
2. Setting of the background image
3. Setting of the currently selected curve name

Definition at line 32 of file BackgroundStateContext.h.

5.2.2 Member Function Documentation

5.2.2.1 void BackgroundStateContext::setCurveSelected (const Transformation & transformation, const DocumentModelGridRemoval & modelGridRemoval, const DocumentModelColorFilter & modelColorFilter, const QString & curveSelected)

Update the selected curve.

Although this probably affects only the BACKGROUND_STATE_CURVE state, we will forward it to all states (consistent with setPixmap)

Definition at line 129 of file BackgroundStateContext.cpp.

The documentation for this class was generated from the following files:

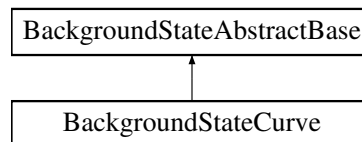
- Background/BackgroundStateContext.h
- Background/BackgroundStateContext.cpp

5.3 BackgroundStateCurve Class Reference

Background image state for showing filter image from current curve.

```
#include <BackgroundStateCurve.h>
```

Inheritance diagram for BackgroundStateCurve:



Public Member Functions

- [BackgroundStateCurve](#) ([BackgroundStateContext](#) &context, [GraphicsScene](#) &scene)
Single constructor.
- virtual void [begin](#) ()
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [fitInView](#) ([GraphicsView](#) &view)
Zoom so background fills the window.
- virtual void [setCurveSelected](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Update the currently selected curve name.
- virtual void [setPixmap](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QPixmap &pixmapOriginal, const QString &curveSelected)
Update the image for this state, after the leaf class processes it appropriately.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateColorFilter](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Apply color filter settings.

Additional Inherited Members

5.3.1 Detailed Description

Background image state for showing filter image from current curve.

Definition at line 13 of file BackgroundStateCurve.h.

The documentation for this class was generated from the following files:

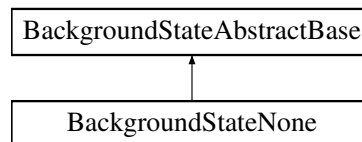
- Background/BackgroundStateCurve.h
- Background/BackgroundStateCurve.cpp

5.4 BackgroundStateNone Class Reference

Background image state for showing no image.

```
#include <BackgroundStateNone.h>
```

Inheritance diagram for BackgroundStateNone:



Public Member Functions

- [BackgroundStateNone](#) ([BackgroundStateContext](#) &context, [GraphicsScene](#) &scene)
Single constructor.
- virtual void [begin](#) ()
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [fitInView](#) ([GraphicsView](#) &view)
Zoom so background fills the window.
- virtual void [setCurveSelected](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Update the currently selected curve name.
- virtual void [setPixmap](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QPixmap &pixmap, const QString &curveSelected)
Update the image for this state, after the leaf class processes it appropriately.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateColorFilter](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Apply color filter settings.

Additional Inherited Members

5.4.1 Detailed Description

Background image state for showing no image.

Definition at line 13 of file BackgroundStateNone.h.

The documentation for this class was generated from the following files:

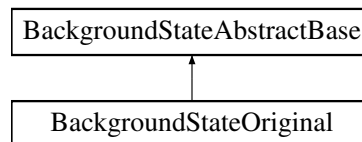
- Background/BackgroundStateNone.h
- Background/BackgroundStateNone.cpp

5.5 BackgroundStateOriginal Class Reference

Background image state for showing original (=unfiltered) image.

```
#include <BackgroundStateOriginal.h>
```

Inheritance diagram for BackgroundStateOriginal:



Public Member Functions

- [BackgroundStateOriginal](#) ([BackgroundStateContext](#) &context, [GraphicsScene](#) &scene)
Single constructor.
- virtual void [begin](#) ()
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [fitInView](#) ([GraphicsView](#) &view)
Zoom so background fills the window.
- virtual void [setCurveSelected](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Update the currently selected curve name.
- virtual void [setPixmap](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QPixmap &pixmap, const QString &curveSelected)
Update the image for this state, after the leaf class processes it appropriately.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateColorFilter](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Apply color filter settings.

Additional Inherited Members

5.5.1 Detailed Description

Background image state for showing original (=unfiltered) image.

Definition at line 13 of file BackgroundStateOriginal.h.

The documentation for this class was generated from the following files:

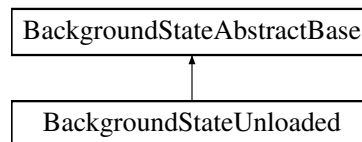
- Background/BackgroundStateOriginal.h
- Background/BackgroundStateOriginal.cpp

5.6 BackgroundStateUnloaded Class Reference

Background image state for interval between startup and loading of the image.

```
#include <BackgroundStateUnloaded.h>
```

Inheritance diagram for BackgroundStateUnloaded:



Public Member Functions

- [BackgroundStateUnloaded](#) ([BackgroundStateContext](#) &context, [GraphicsScene](#) &scene)
Single constructor.
- virtual void [begin](#) ()
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [fitInView](#) ([GraphicsView](#) &view)
Zoom so background fills the window.
- virtual void [setCurveSelected](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Update the currently selected curve name.
- virtual void [setPixmap](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &modelColorFilter, const QPixmap &pixmap, const QString &curveSelected)
Update the image for this state, after the leaf class processes it appropriately.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateColorFilter](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const [DocumentModelColorFilter](#) &colorFilter, const QString &curveSelected)
Apply color filter settings.

Additional Inherited Members

5.6.1 Detailed Description

Background image state for interval between startup and loading of the image.

Definition at line 13 of file BackgroundStateUnloaded.h.

The documentation for this class was generated from the following files:

- Background/BackgroundStateUnloaded.h
- Background/BackgroundStateUnloaded.cpp

5.7 CallbackAddPointsInCurvesGraphs Class Reference

Callback that is used when iterating through a read-only [CurvesGraphs](#) to add corresponding points in [Document](#).

```
#include <CallbackAddPointsInCurvesGraphs.h>
```

Public Member Functions

- [CallbackAddPointsInCurvesGraphs](#) ([CoordSystem](#) &coordSystem)
Single constructor.
- [CallbackSearchReturn](#) *callback* (const QString &curveName, const [Point](#) &point)
Callback method.

5.7.1 Detailed Description

Callback that is used when iterating through a read-only [CurvesGraphs](#) to add corresponding points in [Document](#).

Definition at line 17 of file `CallbackAddPointsInCurvesGraphs.h`.

The documentation for this class was generated from the following files:

- `Callback/CallbackAddPointsInCurvesGraphs.h`
- `Callback/CallbackAddPointsInCurvesGraphs.cpp`

5.8 CallbackAxesCheckerFromAxesPoints Class Reference

Callback for positioning the axes indicator according to the axes points.

```
#include <CallbackAxesCheckerFromAxesPoints.h>
```

Public Member Functions

- [CallbackAxesCheckerFromAxesPoints](#) ()
Single constructor.
- [CallbackSearchReturn](#) *callback* (const QString &curveName, const [Point](#) &point)
Callback method.
- Points [points](#) () const
Return the collected points as a polygon, in screen coordinates.

5.8.1 Detailed Description

Callback for positioning the axes indicator according to the axes points.

Definition at line 17 of file `CallbackAxesCheckerFromAxesPoints.h`.

The documentation for this class was generated from the following files:

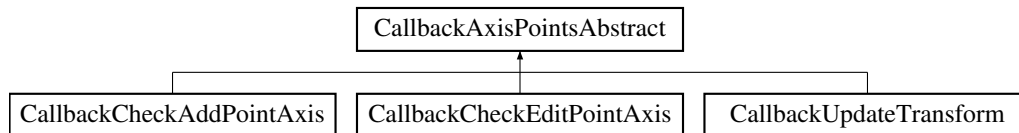
- `Callback/CallbackAxesCheckerFromAxesPoints.h`
- `Callback/CallbackAxesCheckerFromAxesPoints.cpp`

5.9 CallbackAxisPointsAbstract Class Reference

Callback for collecting axis points and then performing common calculations on those axis points.

```
#include <CallbackAxisPointsAbstract.h>
```

Inheritance diagram for CallbackAxisPointsAbstract:



Public Member Functions

- [CallbackAxisPointsAbstract](#) (const [DocumentModelCoords](#) &modelCoords, [DocumentAxesPointsRequired](#) documentAxesPointsRequired)
Constructor for when all of the existing axis points are to be processed as is.
- [CallbackAxisPointsAbstract](#) (const [DocumentModelCoords](#) &modelCoords, const QString pointIdentifier↵ Override, const QPointF &posGraphOverride, const QPointF &posScreenOverride, [DocumentAxesPointsRequired](#) documentAxesPointsRequired)↵
Constructor for when the data for one of the existing axis points is to be locally overwritten.
- [CallbackSearchReturn](#) callback (const QString &curveName, const [Point](#) &point)
Callback method.
- [QTransform](#) [matrixGraph](#) () const
Returns graph coordinates matrix after transformIsDefined has already indicated success.
- [QTransform](#) [matrixScreen](#) () const
Returns screen coordinates matrix after transformIsDefined has already indicated success.
- double [xGraphRange](#) () const
Return the range of the x graph coordinate from low to high, after the transform is defined.
- double [yGraphRange](#) () const
Return the range of the y graph coordinate from low to high, after the transform is defined.

Protected Member Functions

- [DocumentAxesPointsRequired](#) [documentAxesPointsRequired](#) () const
Number of axes points required for the transformation.
- QString [errorMessage](#) () const
This value is checked after iterating to see what was wrong if the axis data was incorrect.
- bool [isError](#) () const
This value is checked after iterating to see if the axis data is correct.
- unsigned int [numberAxisPoints](#) () const
Number of axis points which is less than 3 if the axes curve is incomplete.

Friends

- class [TestGraphCoords](#)
For unit testing.

5.9.1 Detailed Description

Callback for collecting axis points and then performing common calculations on those axis points.

This class collects 3x3 matrix G which contains columns of graph coordinates, and 3x3 matrix S which contains columns of screen coordinates. Although it goes almost as far as solving $(G) = (T) (S)$ for the transformation T, that is left for the [Transformation](#) class. This class does, however, do the sanity checking (like for collinear points) so the gui can provide immediate feedback to the user well before the [Transformation](#) class gets involved

This class is versatile. The cases are:

1. Use all existing axis points, and then the subclass can effectively append one more point to check if that additional point would violate any constraints (prior to adding the point)
2. Use all existing axis points, but override the details of one existing axis point to see if those details violate any constraint (prior to editing the point)
3. Use all existing axis points as is. This is for computing the transformation after axis points are added/edited

Definition at line 35 of file `CallbackAxisPointsAbstract.h`.

5.9.2 Member Function Documentation

5.9.2.1 `bool CallbackAxisPointsAbstract::isError () const` [inline], [protected]

This value is checked after iterating to see if the axis data is correct.

The error state does NOT include the case when there are not enough axis points

Definition at line 80 of file `CallbackAxisPointsAbstract.h`.

5.9.2.2 `QTransform CallbackAxisPointsAbstract::matrixGraph () const`

Returns graph coordinates matrix after `transformIsDefined` has already indicated success.

Since `QMatrix` is deprecated the results are returned as `QTransform`

Definition at line 366 of file `CallbackAxisPointsAbstract.cpp`.

5.9.2.3 `QTransform CallbackAxisPointsAbstract::matrixScreen () const`

Returns screen coordinates matrix after `transformIsDefined` has already indicated success.

Since `QMatrix` is deprecated the results are returned as `QTransform`

Definition at line 371 of file `CallbackAxisPointsAbstract.cpp`.

The documentation for this class was generated from the following files:

- `Callback/CallbackAxisPointsAbstract.h`
- `Callback/CallbackAxisPointsAbstract.cpp`

5.10 CallbackBoundingRects Class Reference

Callback for computing the bounding rectangles of the screen and graph coordinates of the points in the [Document](#).

```
#include <CallbackBoundingRects.h>
```

Public Member Functions

- [CallbackBoundingRects](#) (const [Transformation](#) &transformation)
Single constructor.
- [QRectF boundingRectGraph](#) (bool &isEmpty) const
Graph coordinate bounding rectangle.
- [QRectF boundingRectScreen](#) (bool &isEmpty) const
Screen coordinate bounding rectangle.
- [CallbackSearchReturn callback](#) (const [QString](#) &curveName, const [Point](#) &point)
Callback method.

5.10.1 Detailed Description

Callback for computing the bounding rectangles of the screen and graph coordinates of the points in the [Document](#).

Definition at line 19 of file [CallbackBoundingRects.h](#).

The documentation for this class was generated from the following files:

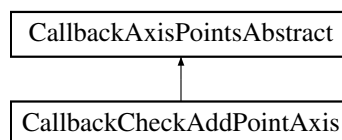
- [Callback/CallbackBoundingRects.h](#)
- [Callback/CallbackBoundingRects.cpp](#)

5.11 CallbackCheckAddPointAxis Class Reference

Callback for sanity checking the screen and graph coordinates of an axis point, before it is added to the axes curve.

```
#include <CallbackCheckAddPointAxis.h>
```

Inheritance diagram for [CallbackCheckAddPointAxis](#):



Public Member Functions

- [CallbackCheckAddPointAxis](#) (const [DocumentModelCoords](#) &modelCoords, const [QPointF](#) &posScreen, const [QPointF](#) &posGraph, [DocumentAxesPointsRequired](#) [documentAxesPointsRequired](#), bool isXOnly)
Single constructor.
- bool [isError](#) () const
True if an error occurred during iteration.
- [QString errorMessage](#) () const
Error message that explains the problem indicated by isError.

Additional Inherited Members

5.11.1 Detailed Description

Callback for sanity checking the screen and graph coordinates of an axis point, before it is added to the axes curve.

Definition at line 18 of file `CallbackCheckAddPointAxis.h`.

The documentation for this class was generated from the following files:

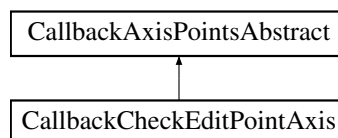
- `Callback/CallbackCheckAddPointAxis.h`
- `Callback/CallbackCheckAddPointAxis.cpp`

5.12 CallbackCheckEditPointAxis Class Reference

Callback for sanity checking the screen and graph coordinates of an axis point that is in the axes curve, before changing its graph coordinates.

```
#include <CallbackCheckEditPointAxis.h>
```

Inheritance diagram for `CallbackCheckEditPointAxis`:



Public Member Functions

- [CallbackCheckEditPointAxis](#) (const [DocumentModelCoords](#) &modelCoords, const QString &pointIdentifier, const QPointF &posScreen, const QPointF &posGraph, DocumentAxesPointsRequired [documentAxesPointsRequired](#))
Single constructor.
- bool [isError](#) () const
True if an error occurred during iteration.
- QString [errorMessage](#) () const
Error message that explains the problem indicated by isError.

Additional Inherited Members

5.12.1 Detailed Description

Callback for sanity checking the screen and graph coordinates of an axis point that is in the axes curve, before changing its graph coordinates.

Definition at line 18 of file `CallbackCheckEditPointAxis.h`.

The documentation for this class was generated from the following files:

- `Callback/CallbackCheckEditPointAxis.h`
- `Callback/CallbackCheckEditPointAxis.cpp`

5.13 CallbackDocumentHash Class Reference

Callback for DocumentHash value for a [Document](#).

```
#include <CallbackDocumentHash.h>
```

Public Member Functions

- [CallbackDocumentHash](#) (DocumentAxesPointsRequired documentAxesPointsRequired)
Single constructor.
- [CallbackSearchReturn callback](#) (const QString &curveName, const [Point](#) &point)
Callback method.
- DocumentHash [hash](#) () const
Computed hash value.

5.13.1 Detailed Description

Callback for DocumentHash value for a [Document](#).

Definition at line 19 of file CallbackDocumentHash.h.

The documentation for this class was generated from the following files:

- Callback/CallbackDocumentHash.h
- Callback/CallbackDocumentHash.cpp

5.14 CallbackGatherXThetaValuesFunctions Class Reference

Callback for collecting X/Theta independent variables, for functions, in preparation for exporting.

```
#include <CallbackGatherXThetaValuesFunctions.h>
```

Public Member Functions

- [CallbackGatherXThetaValuesFunctions](#) (const [DocumentModelExportFormat](#) &modelExport, const Q↔StringList &curveNamesIncluded, const [Transformation](#) &transformation)
Single constructor.
- [CallbackSearchReturn callback](#) (const QString &curveName, const [Point](#) &point)
Callback method.
- ValuesVectorXOrY [xThetaValuesRaw](#) () const
Resulting x/theta values for all included functions.

5.14.1 Detailed Description

Callback for collecting X/Theta independent variables, for functions, in preparation for exporting.

Definition at line 27 of file CallbackGatherXThetaValuesFunctions.h.

The documentation for this class was generated from the following files:

- Callback/CallbackGatherXThetaValuesFunctions.h
- Callback/CallbackGatherXThetaValuesFunctions.cpp

5.15 CallbackNextOrdinal Class Reference

Callback for computing the next ordinal for a new point.

```
#include <CallbackNextOrdinal.h>
```

Public Member Functions

- [CallbackNextOrdinal](#) (const QString &curveName)
Single constructor.
- [CallbackSearchReturn callback](#) (const QString &curveName, const [Point](#) &point)
Callback method.
- double [nextOrdinal](#) () const
Computed next ordinal.

5.15.1 Detailed Description

Callback for computing the next ordinal for a new point.

Definition at line 17 of file CallbackNextOrdinal.h.

The documentation for this class was generated from the following files:

- Callback/CallbackNextOrdinal.h
- Callback/CallbackNextOrdinal.cpp

5.16 CallbackPointOrdinal Class Reference

Callback for computing the ordinal for a specified point, as a function of the [LineStyle](#) and curve geometry.

```
#include <CallbackPointOrdinal.h>
```


Public Member Functions

- [CallbackPointOrdinal](#) (const [LineStyle](#) &lineStyle, const [Transformation](#) &transformation, const QPointF &posScreen)
Single constructor.
- [CallbackSearchReturn callback](#) (const [Point](#) &pointStart, const [Point](#) &pointStop)
Callback method.
- double [ordinal](#) () const
Computed ordinal.

5.16.1 Detailed Description

Callback for computing the ordinal for a specified point, as a function of the [LineStyle](#) and curve geometry.

Definition at line 19 of file [CallbackPointOrdinal.h](#).

The documentation for this class was generated from the following files:

- [Callback/CallbackPointOrdinal.h](#)
- [Callback/CallbackPointOrdinal.cpp](#)

5.17 CallbackRemovePointsInCurvesGraphs Class Reference

Callback that is used when iterating through a read-only [CurvesGraphs](#) to remove corresponding points in [Document](#).

```
#include <CallbackRemovePointsInCurvesGraphs.h>
```

Public Member Functions

- [CallbackRemovePointsInCurvesGraphs](#) ([CoordSystem](#) &coordSystem)
Single constructor.
- [CallbackSearchReturn callback](#) (const QString &curveName, const [Point](#) &point)
Callback method.

5.17.1 Detailed Description

Callback that is used when iterating through a read-only [CurvesGraphs](#) to remove corresponding points in [Document](#).

Definition at line 17 of file [CallbackRemovePointsInCurvesGraphs.h](#).

The documentation for this class was generated from the following files:

- [Callback/CallbackRemovePointsInCurvesGraphs.h](#)
- [Callback/CallbackRemovePointsInCurvesGraphs.cpp](#)

5.18 CallbackSceneUpdateAfterCommand Class Reference

Callback for updating the QGraphicsItems in the scene after a command may have modified Points in Curves.

```
#include <CallbackSceneUpdateAfterCommand.h>
```

Public Member Functions

- [CallbackSceneUpdateAfterCommand](#) ([GraphicsLinesForCurves](#) &graphicsLinesForCurves, [GraphicsScene](#) &scene, const [Document](#) &document, [GeometryWindow](#) *geometryWindow)
Single constructor.
- [CallbackSearchReturn callback](#) (const [QString](#) &, const [Point](#) &point)
Callback method.

5.18.1 Detailed Description

Callback for updating the QGraphicsItems in the scene after a command may have modified Points in Curves.

Definition at line 20 of file `CallbackSceneUpdateAfterCommand.h`.

The documentation for this class was generated from the following files:

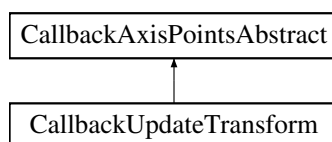
- `Callback/CallbackSceneUpdateAfterCommand.h`
- `Callback/CallbackSceneUpdateAfterCommand.cpp`

5.19 CallbackUpdateTransform Class Reference

Callback for collecting axis points and then calculating the current transform from those axis points.

```
#include <CallbackUpdateTransform.h>
```

Inheritance diagram for `CallbackUpdateTransform`:



Public Member Functions

- [CallbackUpdateTransform](#) (const [DocumentModelCoords](#) &modelCoords, [DocumentAxesPointsRequired](#) documentAxesPointsRequired)
Single constructor.
- [bool transformIsDefined](#) () const
True if enough Points were available to create a [Transformation](#).

Additional Inherited Members

5.19.1 Detailed Description

Callback for collecting axis points and then calculating the current transform from those axis points.

Sanity checking of the axis points was applied earlier when the axis points were added/edited.

Definition at line 19 of file CallbackUpdateTransform.h.

5.19.2 Member Function Documentation

5.19.2.1 bool CallbackUpdateTransform::transformIsDefined () const

True if enough Points were available to create a [Transformation](#).

Except for the node count, all other failure modes are caught externally so user gets immediate feedback as soon as bad axis point data appears

Definition at line 17 of file CallbackUpdateTransform.cpp.

The documentation for this class was generated from the following files:

- Callback/CallbackUpdateTransform.h
- Callback/CallbackUpdateTransform.cpp

5.20 Checker Class Reference

Box shape that is drawn through the three axis points, to temporarily (usually) or permanently (rarely) highlight the local up/down/left/right directions when all axis points have been defined.

```
#include <Checker.h>
```

Public Member Functions

- [Checker](#) (QGraphicsScene &scene)
Single constructor for [DlgSettingsAxesChecker](#), which does not have an explicit transformation. The identity transformation is assumed.
- void [prepareForDisplay](#) (const QPolygonF &polygon, int pointRadius, const [DocumentModelAxesChecker](#) &modelAxesChecker, const [DocumentModelCoords](#) &modelCoords, DocumentAxesPointsRequired documentAxesPointsRequired)
Create the polygon from current information, including pixel coordinates, just prior to display.
- void [prepareForDisplay](#) (const QList< [Point](#) > &Points, int pointRadius, const [DocumentModelAxesChecker](#) &modelAxesChecker, const [DocumentModelCoords](#) &modelCoords, const [Transformation](#) &transformation, DocumentAxesPointsRequired documentAxesPointsRequired)
Create the polygon from current information, including pixel and graph coordinates, just prior to display.
- void [setVisible](#) (bool visible)
Show/hide this axes checker.
- virtual void [updateModelAxesChecker](#) (const [DocumentModelAxesChecker](#) &modelAxesChecker)
Apply the new [DocumentModelAxesChecker](#), to the points already associated with this object.

5.20.1 Detailed Description

Box shape that is drawn through the three axis points, to temporarily (usually) or permanently (rarely) highlight the local up/down/left/right directions when all axis points have been defined.

The goal of the checker is to make it obvious when a mistake has happened so the screen-to-graph transformation is currently wrong - since the expected up/down/left/right directions will be awry which will distort the checker somehow. Unfortunately, errors in scale are not revealed by the checker.

Definition at line 33 of file Checker.h.

5.20.2 Member Function Documentation

5.20.2.1 void Checker::prepareForDisplay (const QPolygonF & *polygon*, int *pointRadius*, const DocumentModelAxesChecker & *modelAxesChecker*, const DocumentModelCoords & *modelCoords*, DocumentAxesPointsRequired *documentAxesPointsRequired*)

Create the polygon from current information, including pixel coordinates, just prior to display.

This is for [DlgSettingsAxesChecker](#). The identity matrix is used for the transformations between screen and graph coordinates. The point radius is used to exclude the lines from the axes points for clarity

Definition at line 129 of file Checker.cpp.

5.20.2.2 void Checker::prepareForDisplay (const QList< Point > & *Points*, int *pointRadius*, const DocumentModelAxesChecker & *modelAxesChecker*, const DocumentModelCoords & *modelCoords*, const Transformation & *transformation*, DocumentAxesPointsRequired *documentAxesPointsRequired*)

Create the polygon from current information, including pixel and graph coordinates, just prior to display.

This is for [TransformationStateDefined](#). The point radius is used to exclude the lines from the axes points for clarity

Definition at line 166 of file Checker.cpp.

5.20.2.3 void Checker::updateModelAxesChecker (const DocumentModelAxesChecker & *modelAxesChecker*)
[virtual]

Apply the new [DocumentModelAxesChecker](#), to the points already associated with this object.

This method starts the timer unless the mode is never or forever

Definition at line 244 of file Checker.cpp.

The documentation for this class was generated from the following files:

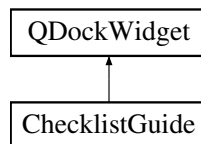
- Checker/Checker.h
- Checker/Checker.cpp

5.21 ChecklistGuide Class Reference

Dockable text window containing checklist guide.

```
#include <ChecklistGuide.h>
```

Inheritance diagram for ChecklistGuide:



Signals

- void [signalChecklistClosed](#) ()
Signal that this QDockWidget was just closed.

Public Member Functions

- [ChecklistGuide](#) (QWidget *parent)
Single constructor. Parent is needed or else this widget cannot be redocked after being undocked.
- bool [browserIsEmpty](#) () const
When browser is empty, it is pointless to show it.
- virtual void [closeEvent](#) (QCloseEvent *event)
Catch close event so corresponding menu item in [MainWindow](#) can be updated accordingly.
- void [setTemplateHtml](#) (const QString &html, const QStringList &curveNames)
Populate the browser with template html.
- void [update](#) (const [CmdMediator](#) &cmdMediator, bool documentIsExported)
Update using current CmdMediator/Document state.

5.21.1 Detailed Description

Dockable text window containing checklist guide.

Definition at line 16 of file ChecklistGuide.h.

The documentation for this class was generated from the following files:

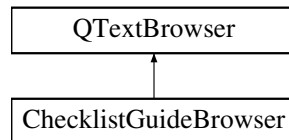
- Checklist/ChecklistGuide.h
- Checklist/ChecklistGuide.cpp

5.22 ChecklistGuideBrowser Class Reference

Class that adds rudimentary tree collapse/expand functionality to QTextBrowser.

```
#include <ChecklistGuideBrowser.h>
```

Inheritance diagram for ChecklistGuideBrowser:



Public Member Functions

- [ChecklistGuideBrowser](#) ()
Single constructor.
- virtual void [setTemplateHtml](#) (const QString &html, const QStringList &curveNames)
Populate the browser with template html. The template html will be converted to real html.
- void [update](#) (const [CmdMediator](#) &cmdMediator, bool documentIsExported)
Update using current CmdMediator/Document state.

5.22.1 Detailed Description

Class that adds rudimentary tree collapse/expand functionality to QTextBrowser.

Definition at line 15 of file ChecklistGuideBrowser.h.

The documentation for this class was generated from the following files:

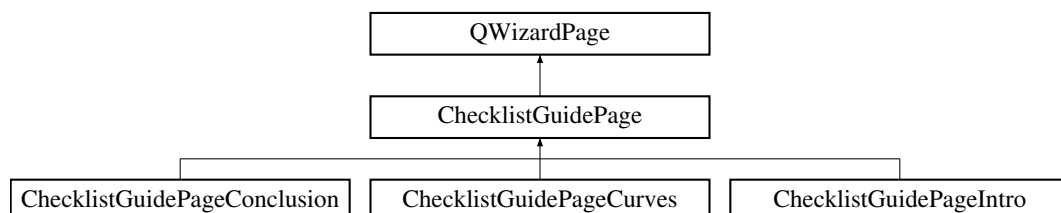
- Checklist/ChecklistGuideBrowser.h
- Checklist/ChecklistGuideBrowser.cpp

5.23 ChecklistGuidePage Class Reference

This class customizes QWizardPage for [ChecklistGuideWizard](#).

```
#include <ChecklistGuidePage.h>
```

Inheritance diagram for ChecklistGuidePage:



Public Member Functions

- [ChecklistGuidePage](#) (const QString &title)
Single constructor.
- void [addHtml](#) (const QString &html)
Insert html for display.
- QRadioButton * [addLabelAndRadioButton](#) (const QString &label, const QString &whatsThis)
Insert radio button and corresponding label.
- void [addLineEdit](#) ([ChecklistLineEdit](#) *edit, const QString &whatsThis)
Insert line edit.

5.23.1 Detailed Description

This class customizes QWizardPage for [ChecklistGuideWizard](#).

Definition at line 19 of file ChecklistGuidePage.h.

The documentation for this class was generated from the following files:

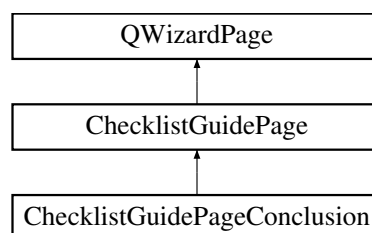
- Checklist/ChecklistGuidePage.h
- Checklist/ChecklistGuidePage.cpp

5.24 ChecklistGuidePageConclusion Class Reference

This class uses the validation method of the Conclusion page to perform final processing for [ChecklistGuideWizard](#).

```
#include <ChecklistGuidePageConclusion.h>
```

Inheritance diagram for ChecklistGuidePageConclusion:



Public Member Functions

- [ChecklistGuidePageConclusion](#) ()
Single constructor.
- virtual bool [validatePage](#) ()
Perform final processing.

5.24.1 Detailed Description

This class uses the validation method of the Conclusion page to perform final processing for [ChecklistGuideWizard](#).

Definition at line 13 of file ChecklistGuidePageConclusion.h.

The documentation for this class was generated from the following files:

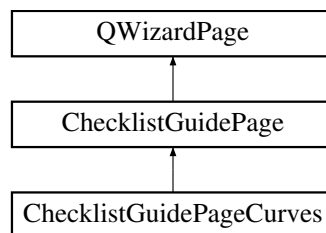
- Checklist/ChecklistGuidePageConclusion.h
- Checklist/ChecklistGuidePageConclusion.cpp

5.25 ChecklistGuidePageCurves Class Reference

This class adds validation to the Curves page.

```
#include <ChecklistGuidePageCurves.h>
```

Inheritance diagram for ChecklistGuidePageCurves:



Public Slots

- void [slotTableChanged](#) ()
Update after curve table update.
- bool [withLines](#) () const
Drawn with lines, else points.

Public Member Functions

- [ChecklistGuidePageCurves](#) (const QString &title)
Single constructor.
- QStringList [curveNames](#) () const
Wizard selection for curve names.
- virtual bool [isComplete](#) () const
Validate the contents of this page.

5.25.1 Detailed Description

This class adds validation to the Curves page.

Definition at line 17 of file ChecklistGuidePageCurves.h.

The documentation for this class was generated from the following files:

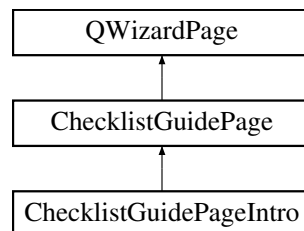
- Checklist/ChecklistGuidePageCurves.h
- Checklist/ChecklistGuidePageCurves.cpp

5.26 ChecklistGuidePageIntro Class Reference

This class sets up the introduction page.

```
#include <ChecklistGuidePageIntro.h>
```

Inheritance diagram for ChecklistGuidePageIntro:



Public Member Functions

- [ChecklistGuidePageIntro \(\)](#)

Single constructor.

5.26.1 Detailed Description

This class sets up the introduction page.

Definition at line 13 of file ChecklistGuidePageIntro.h.

The documentation for this class was generated from the following files:

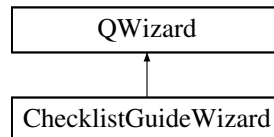
- Checklist/ChecklistGuidePageIntro.h
- Checklist/ChecklistGuidePageIntro.cpp

5.27 ChecklistGuideWizard Class Reference

Wizard for setting up the checklist guide.

```
#include <ChecklistGuideWizard.h>
```

Inheritance diagram for ChecklistGuideWizard:



Public Member Functions

- [ChecklistGuideWizard](#) ([MainWindow](#) &mainWindow, unsigned int numberCoordSystem)
Single constructor.
- QStringList [curveNames](#) (CoordSystemIndex coordSystemIndex) const
Curve names to be placed into [Document](#).
- void [populateCurvesGraphs](#) (CoordSystemIndex coordSystemIndex, [CurvesGraphs](#) &curvesGraphs)
Create entries in [CurvesGraphs](#) for each curve name that user provided.
- QString [templateHtml](#) (CoordSystemIndex coordSystemIndex) const
Template html comprising the checklist for display.

5.27.1 Detailed Description

Wizard for setting up the checklist guide.

Definition at line 23 of file ChecklistGuideWizard.h.

The documentation for this class was generated from the following files:

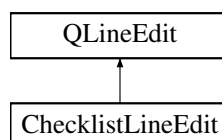
- Checklist/ChecklistGuideWizard.h
- Checklist/ChecklistGuideWizard.cpp

5.28 ChecklistLineEdit Class Reference

Adds key event handling to QLineEdit.

```
#include <ChecklistLineEdit.h>
```

Inheritance diagram for ChecklistLineEdit:



Signals

- void [signalKeyRelease](#) ()
Signal that user has just released a key.

Public Member Functions

- [ChecklistLineEdit](#) ()
Single constructor.
- virtual void [keyPressEvent](#) (QKeyEvent *event)
Intercept the key press event.

5.28.1 Detailed Description

Adds key event handling to QLineEdit.

Definition at line 13 of file ChecklistLineEdit.h.

The documentation for this class was generated from the following files:

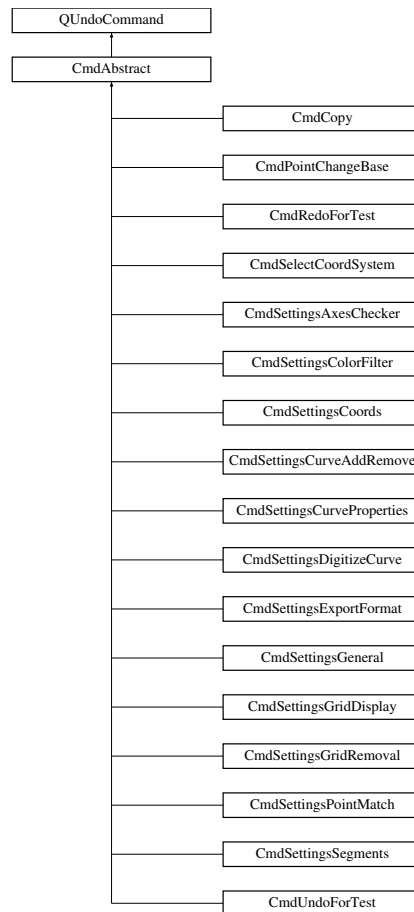
- Checklist/ChecklistLineEdit.h
- Checklist/ChecklistLineEdit.cpp

5.29 CmdAbstract Class Reference

Wrapper around QUndoCommand. This simplifies the more complicated feature set of QUndoCommand.

```
#include <CmdAbstract.h>
```

Inheritance diagram for CmdAbstract:



Public Member Functions

- [CmdAbstract](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &cmdDescription)
Single constructor.
- virtual void [cmdRedo](#) ()=0
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()=0
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const =0
Save commands as xml for later uploading.

Protected Member Functions

- [Document](#) & [document](#) ()
Return the [Document](#) that this command will modify during redo and undo.
- const [Document](#) & [document](#) () const
Return a const copy of the [Document](#) for non redo/undo interaction.
- [MainWindow](#) & [mainWindow](#) ()
Return the [MainWindow](#) so it can be updated by this command as a last step.
- void [resetSelection](#) (const [PointIdentifiers](#) &pointIdentifiersToSelect)
Since the set of selected points has probably changed, changed that set back to the specified set.
- void [saveOrCheckPostCommandDocumentStateHash](#) (const [Document](#) &[document](#))
Save, when called the first time, a hash value representing the state of the [Document](#).
- void [saveOrCheckPreCommandDocumentStateHash](#) (const [Document](#) &[document](#))
Save, when called the first time, a hash value representing the state of the [Document](#).

5.29.1 Detailed Description

Wrapper around QUndoCommand. This simplifies the more complicated feature set of QUndoCommand.

Definition at line 19 of file CmdAbstract.h.

5.29.2 Member Function Documentation

5.29.2.1 void CmdAbstract::resetSelection (const PointIdentifiers & *pointIdentifiersToSelect*) [protected]

Since the set of selected points has probably changed, changed that set back to the specified set.

This lets the user move selected point(s) repeatedly using arrow keys. Also provides expected behavior when pasting

Definition at line 81 of file CmdAbstract.cpp.

5.29.2.2 void CmdAbstract::saveOrCheckPostCommandDocumentStateHash (const Document & *document*) [protected]

Save, when called the first time, a hash value representing the state of the [Document](#).

Then on succeeding calls the hash is recomputed and compared to the original value to check for consistency. This "post" method is called immediately after the redo method of the subclass has done its processing. See also `saveOrCheckPreCommandDocumentState`

Definition at line 102 of file CmdAbstract.cpp.

5.29.2.3 void CmdAbstract::saveOrCheckPreCommandDocumentStateHash (const Document & *document*) [protected]

Save, when called the first time, a hash value representing the state of the [Document](#).

Then on succeeding calls the hash is recomputed and compared to the original value to check for consistency. This "pre" method is called immediately after the redo method of the subclass has done its processing. See also `saveOrCheckPostCommandDocumentState`

Definition at line 125 of file CmdAbstract.cpp.

The documentation for this class was generated from the following files:

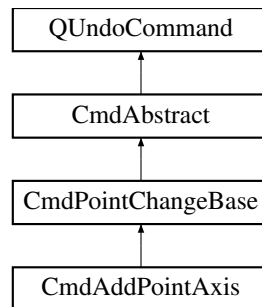
- Cmd/CmdAbstract.h
- Cmd/CmdAbstract.cpp

5.30 CmdAddPointAxis Class Reference

Command for adding one axis point.

```
#include <CmdAddPointAxis.h>
```

Inheritance diagram for CmdAddPointAxis:



Public Member Functions

- [CmdAddPointAxis](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QPointF &posScreen, const QPointF &posGraph, double ordinal, bool isXOnly)
Constructor for normal creation.
- [CmdAddPointAxis](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, Q↔XmlStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.30.1 Detailed Description

Command for adding one axis point.

Definition at line 16 of file CmdAddPointAxis.h.

The documentation for this class was generated from the following files:

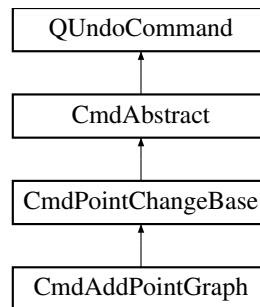
- Cmd/CmdAddPointAxis.h
- Cmd/CmdAddPointAxis.cpp

5.31 CmdAddPointGraph Class Reference

Command for adding one graph point.

```
#include <CmdAddPointGraph.h>
```

Inheritance diagram for CmdAddPointGraph:



Public Member Functions

- [CmdAddPointGraph](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &curveName, const QPointF &posScreen, double ordinal)
Constructor for normal creation.
- [CmdAddPointGraph](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.31.1 Detailed Description

Command for adding one graph point.

Definition at line 17 of file CmdAddPointGraph.h.

The documentation for this class was generated from the following files:

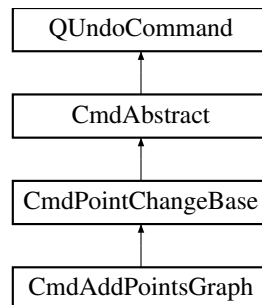
- Cmd/CmdAddPointGraph.h
- Cmd/CmdAddPointGraph.cpp

5.32 CmdAddPointsGraph Class Reference

Command for adding one or more graph points. This is for [Segment](#) Fill mode.

```
#include <CmdAddPointsGraph.h>
```

Inheritance diagram for CmdAddPointsGraph:



Public Member Functions

- [CmdAddPointsGraph](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &curveName, const QList< QPoint > &points, const QList< double > &ordinals)
Constructor for normal creation.
- [CmdAddPointsGraph](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QXmlStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.32.1 Detailed Description

Command for adding one or more graph points. This is for [Segment](#) Fill mode.

Definition at line 19 of file CmdAddPointsGraph.h.

The documentation for this class was generated from the following files:

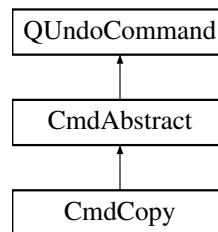
- Cmd/CmdAddPointsGraph.h
- Cmd/CmdAddPointsGraph.cpp

5.33 CmdCopy Class Reference

Command for moving all selected Points by a specified translation.

```
#include <CmdCopy.h>
```

Inheritance diagram for CmdCopy:



Public Member Functions

- [CmdCopy](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QStringList &selectedPointIdentifiers)
Constructor for normal creation.
- [CmdCopy](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &cmdDescription, QXmlStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.33.1 Detailed Description

Command for moving all selected Points by a specified translation.

Definition at line 18 of file CmdCopy.h.

The documentation for this class was generated from the following files:

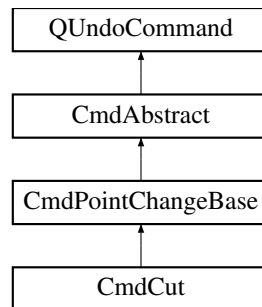
- Cmd/CmdCopy.h
- Cmd/CmdCopy.cpp

5.34 CmdCut Class Reference

Command for cutting all selected Points.

```
#include <CmdCut.h>
```

Inheritance diagram for CmdCut:



Public Member Functions

- [CmdCut](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QStringList &selectedPointIdentifiers)
Constructor for normal creation.
- [CmdCut](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.34.1 Detailed Description

Command for cutting all selected Points.

Definition at line 18 of file CmdCut.h.

The documentation for this class was generated from the following files:

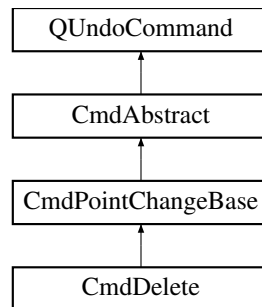
- Cmd/CmdCut.h
- Cmd/CmdCut.cpp

5.35 CmdDelete Class Reference

Command for deleting all selected Points.

```
#include <CmdDelete.h>
```

Inheritance diagram for CmdDelete:



Public Member Functions

- [CmdDelete](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QStringList &selectedPointIdentifiers)
Constructor for normal creation.
- [CmdDelete](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.35.1 Detailed Description

Command for deleting all selected Points.

Definition at line 18 of file CmdDelete.h.

The documentation for this class was generated from the following files:

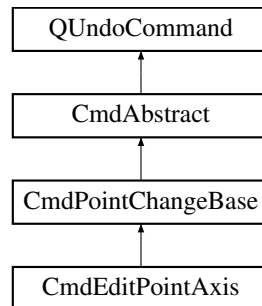
- Cmd/CmdDelete.h
- Cmd/CmdDelete.cpp

5.36 CmdEditPointAxis Class Reference

Command for editing the graph coordinates one axis point.

```
#include <CmdEditPointAxis.h>
```

Inheritance diagram for CmdEditPointAxis:



Public Member Functions

- [CmdEditPointAxis](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &pointIdentifier, const QPointF &posGraphBefore, const QPointF &posGraphAfter, bool isXOnly)
Constructor for normal creation.
- [CmdEditPointAxis](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, Q<↔ XMLStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.36.1 Detailed Description

Command for editing the graph coordinates one axis point.

The screen coordinates are handled by another command

Definition at line 18 of file CmdEditPointAxis.h.

The documentation for this class was generated from the following files:

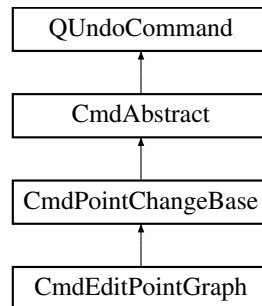
- Cmd/CmdEditPointAxis.h
- Cmd/CmdEditPointAxis.cpp

5.37 CmdEditPointGraph Class Reference

Command for editing the graph coordinates of one or more graph points.

```
#include <CmdEditPointGraph.h>
```

Inheritance diagram for CmdEditPointGraph:



Public Member Functions

- [CmdEditPointGraph](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QStringList &pointIdentifiers, bool isX, bool isY, double x, double y)
Constructor for normal creation.
- [CmdEditPointGraph](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.37.1 Detailed Description

Command for editing the graph coordinates of one or more graph points.

The screen coordinates are handled by another command

Definition at line 18 of file CmdEditPointGraph.h.

The documentation for this class was generated from the following files:

- Cmd/CmdEditPointGraph.h
- Cmd/CmdEditPointGraph.cpp

5.38 CmdFactory Class Reference

Factory for CmdAbstractBase objects.

```
#include <CmdFactory.h>
```

Public Member Functions

- [CmdFactory](#) ()
Single constructor.
- [CmdAbstract](#) * [createCmd](#) ([MainWindow](#) &mainWindow, [Document](#) &document, [QXmlStreamReader](#) &reader)
Factory method. Input is the xml node from an error report file.

5.38.1 Detailed Description

Factory for CmdAbstractBase objects.

Definition at line 16 of file CmdFactory.h.

The documentation for this class was generated from the following files:

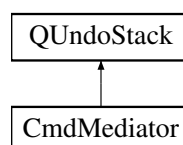
- Cmd/CmdFactory.h
- Cmd/CmdFactory.cpp

5.39 CmdMediator Class Reference

Command queue stack.

```
#include <CmdMediator.h>
```

Inheritance diagram for CmdMediator:



Public Member Functions

- [CmdMediator](#) ([MainWindow](#) &mainWindow, const QImage &image)
Constructor for imported images and dragged images. Only one coordinate system is created but others can be added later.
- [CmdMediator](#) ([MainWindow](#) &mainWindow, const QString &fileName)
Constructor for opened Documents and error report files. The specified xml file is opened and read.
- [~CmdMediator](#) ()
Destructor.
- const [CoordSystem](#) & [coordSystem](#) () const
Provide the current [CoordSystem](#) to commands with read-only access, primarily for undo/redo processing.
- const [Curve](#) & [curveAxes](#) () const
See [Document::curveAxes](#).
- QStringList [curvesGraphsNames](#) () const
See [CurvesGraphs::curvesGraphsNames](#).
- int [curvesGraphsNumPoints](#) (const QString &curveName) const
See [CurvesGraphs::curvesGraphsNumPoints](#).
- [Document](#) & [document](#) ()
Provide the [Document](#) to commands, primarily for undo/redo processing.
- const [Document](#) & [document](#) () const
Provide the [Document](#) to commands with read-only access, primarily for undo/redo processing.
- bool [isModified](#) () const
Dirty flag.
- void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [Callback](#)↔[SearchReturn](#) > &f) withCallback)
See [Curve::iterateThroughCurvePoints](#), for the single axes curve.
- void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [Callback](#)↔[SearchReturn](#) > &f) withCallback) const
See [Curve::iterateThroughCurvePoints](#), for the single axes curve.
- void [iterateThroughCurvesPointsGraphs](#) (const Functor2wRet< const QString &, const [Point](#) &, [Callback](#)↔[SearchReturn](#) > &f) withCallback)
See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
- QPixmap [pixmap](#) () const
See [Document::pixmap](#).
- QString [reasonForUnsuccessfulRead](#) () const
See [Document::reasonForUnsuccessfulRead](#).
- void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize to xml.
- QString [selectedCurveName](#) () const
Currently selected curve name. This is used to set the selected curve combobox in [MainWindow](#).
- void [setDocumentAxesPointsRequired](#) (DocumentAxesPointsRequired documentAxesPointsRequired)
Set the number of axes points required.
- void [setSelectedCurveName](#) (const QString &[selectedCurveName](#))
Save curve name that is selected for the current coordinate system, for the next time the coordinate system reappears.
- bool [successfulRead](#) () const
Wrapper for [Document::successfulRead](#).

5.39.1 Detailed Description

Command queue stack.

This class lies between the [Document](#) and the rest of the application. This approach is attractive because the command stack and [Document](#) are born together, work together, and deleted together. Also, wrapping this class around [Document](#) helps to encapsulate [Document](#) that much more.

Definition at line 23 of file CmdMediator.h.

5.39.2 Member Function Documentation

5.39.2.1 `bool CmdMediator::isModified () const`

Dirty flag.

[Document](#) is dirty if there are any unsaved changes. The dirty flag is pushed (rather than pulled from this method) through the `QUndoStack::cleanChanged` signal

Definition at line 82 of file CmdMediator.cpp.

5.39.2.2 `void CmdMediator::setDocumentAxesPointsRequired (DocumentAxesPointsRequired documentAxesPointsRequired)`

Set the number of axes points required.

This is called during the [Document](#) creation process, after imported images have been previewed or loaded files have had at least some xml parsing

Definition at line 132 of file CmdMediator.cpp.

The documentation for this class was generated from the following files:

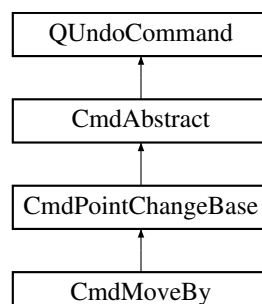
- Cmd/CmdMediator.h
- Cmd/CmdMediator.cpp

5.40 CmdMoveBy Class Reference

Command for moving all selected Points by a specified translation.

```
#include <CmdMoveBy.h>
```

Inheritance diagram for CmdMoveBy:



Public Member Functions

- [CmdMoveBy](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QPointF &[deltaScreen](#), const QString &[moveText](#), const QStringList &[selectedPointIdentifiers](#))
Constructor for normal creation.
- [CmdMoveBy](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &[cmdDescription](#), QXmlStreamReader &[reader](#))
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &[writer](#)) const
Save commands as xml for later uploading.

Additional Inherited Members

5.40.1 Detailed Description

Command for moving all selected Points by a specified translation.

Definition at line 18 of file CmdMoveBy.h.

The documentation for this class was generated from the following files:

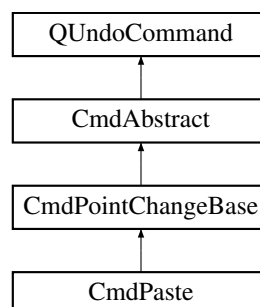
- Cmd/CmdMoveBy.h
- Cmd/CmdMoveBy.cpp

5.41 CmdPaste Class Reference

Command for moving all selected Points by a specified translation.

```
#include <CmdPaste.h>
```

Inheritance diagram for CmdPaste:



Public Member Functions

- [CmdPaste](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QStringList &selectedPointIdentifiers)
Constructor for normal creation.
- [CmdPaste](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &cmdDescription, QXmlStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.41.1 Detailed Description

Command for moving all selected Points by a specified translation.

Definition at line 18 of file CmdPaste.h.

The documentation for this class was generated from the following files:

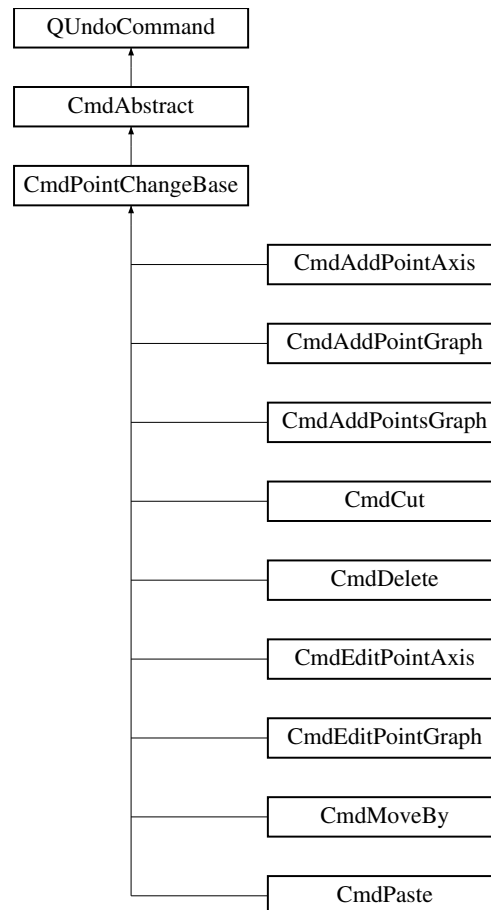
- Cmd/CmdPaste.h
- Cmd/CmdPaste.cpp

5.42 CmdPointChangeBase Class Reference

Base class for CmdBase leaf subclasses that involve point additions, deletions and/or modifications.

```
#include <CmdPointChangeBase.h>
```

Inheritance diagram for CmdPointChangeBase:



Public Member Functions

- [CmdPointChangeBase](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const [QString](#) &[cmdDescription](#))
Single constructor.

Protected Member Functions

- void [restoreDocumentState](#) ([Document](#) &[document](#)) const
Restore the document previously saved by [saveDocumentState](#).
- void [saveDocumentState](#) (const [Document](#) &[document](#))
Save the document state for restoration by [restoreDocumentState](#).

5.42.1 Detailed Description

Base class for CmdBase leaf subclasses that involve point additions, deletions and/or modifications.

This class uses a strategy of taking a snapshot of all points before the redo, and then applying that snapshot to the [Document](#) to (later) perform the undo. Before this strategy, the strategy was to just do the opposite steps of the redo, but that strategy was too fragile since it implicitly assumed no point changes occurred after the redo of this command and before the redo of the next command. However, point updates like "ordinal maintenance" do occur during that time period

Definition at line 22 of file [CmdPointChangeBase.h](#).

The documentation for this class was generated from the following files:

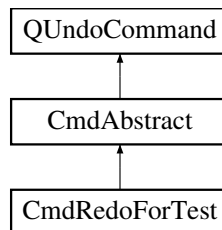
- [Cmd/CmdPointChangeBase.h](#)
- [Cmd/CmdPointChangeBase.cpp](#)

5.43 CmdRedoForTest Class Reference

Command for performing Redo during testing.

```
#include <CmdRedoForTest.h>
```

Inheritance diagram for CmdRedoForTest:



Public Member Functions

- [CmdRedoForTest](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#))
Constructor for normal creation.
- [CmdRedoForTest](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &[cmdDescription](#), Q←
XmlStreamReader &[reader](#))
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &[writer](#)) const
Save commands as xml for later uploading.

Additional Inherited Members

5.43.1 Detailed Description

Command for performing Redo during testing.

This command is never created automatically, since when the user triggers an Redo that just results in the command stack getting backed up by one command. This command is manually created by editing an xml test file

Definition at line 20 of file CmdRedoForTest.h.

The documentation for this class was generated from the following files:

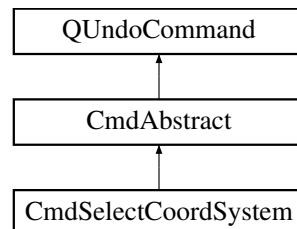
- Cmd/CmdRedoForTest.h
- Cmd/CmdRedoForTest.cpp

5.44 CmdSelectCoordSystem Class Reference

Command for changing the currently selected [CoordSystem](#).

```
#include <CmdSelectCoordSystem.h>
```

Inheritance diagram for CmdSelectCoordSystem:



Public Member Functions

- [CmdSelectCoordSystem](#) ([MainWindow](#) &mainWindow, [Document](#) &document, CoordSystemIndex coord↵System)
Constructor for normal creation.
- [CmdSelectCoordSystem](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmd↵Description, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.44.1 Detailed Description

Command for changing the currently selected [CoordSystem](#).

Definition at line 16 of file CmdSelectCoordSystem.h.

The documentation for this class was generated from the following files:

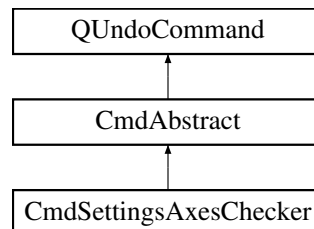
- Cmd/CmdSelectCoordSystem.h
- Cmd/CmdSelectCoordSystem.cpp

5.45 CmdSettingsAxesChecker Class Reference

Command for [DlgSettingsAxesChecker](#).

```
#include <CmdSettingsAxesChecker.h>
```

Inheritance diagram for CmdSettingsAxesChecker:



Public Member Functions

- [CmdSettingsAxesChecker](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModel](#) &modelAxesCheckerBefore, const [DocumentModelAxesChecker](#) &modelAxesCheckerAfter)
Constructor for normal creation.
- [CmdSettingsAxesChecker](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [QString](#) &cmdDescription, [QXmlStreamReader](#) &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) ([QXmlStreamWriter](#) &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.45.1 Detailed Description

Command for [DlgSettingsAxesChecker](#).

Definition at line 16 of file CmdSettingsAxesChecker.h.

The documentation for this class was generated from the following files:

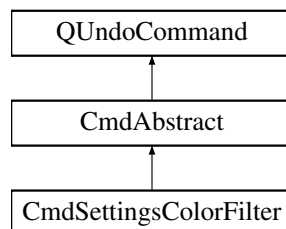
- Cmd/CmdSettingsAxesChecker.h
- Cmd/CmdSettingsAxesChecker.cpp

5.46 CmdSettingsColorFilter Class Reference

Command for [DlgSettingsColorFilter](#).

```
#include <CmdSettingsColorFilter.h>
```

Inheritance diagram for CmdSettingsColorFilter:



Public Member Functions

- [CmdSettingsColorFilter](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelColorFilter](#) &modelColorFilterBefore, const [DocumentModelColorFilter](#) &modelColorFilterAfter)
Constructor for normal creation.
- [CmdSettingsColorFilter](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [QString](#) &cmdDescription, [QXmlStreamReader](#) &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) ([QXmlStreamWriter](#) &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.46.1 Detailed Description

Command for [DlgSettingsColorFilter](#).

Definition at line 16 of file CmdSettingsColorFilter.h.

The documentation for this class was generated from the following files:

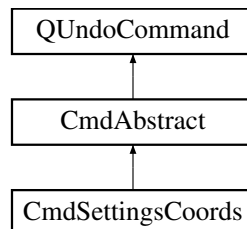
- Cmd/CmdSettingsColorFilter.h
- Cmd/CmdSettingsColorFilter.cpp

5.47 CmdSettingsCoords Class Reference

Command for [DlgSettingsCoords](#).

```
#include <CmdSettingsCoords.h>
```

Inheritance diagram for CmdSettingsCoords:



Public Member Functions

- [CmdSettingsCoords](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelCoords](#) &modelCoordsBefore, const [DocumentModelCoords](#) &modelCoordsAfter)
Constructor for normal creation.
- [CmdSettingsCoords](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.47.1 Detailed Description

Command for [DlgSettingsCoords](#).

Definition at line 16 of file CmdSettingsCoords.h.

The documentation for this class was generated from the following files:

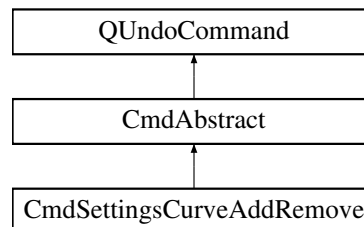
- Cmd/CmdSettingsCoords.h
- Cmd/CmdSettingsCoords.cpp

5.48 CmdSettingsCurveAddRemove Class Reference

Command for [DlgSettingsCurveAddRemove](#).

```
#include <CmdSettingsCurveAddRemove.h>
```

Inheritance diagram for CmdSettingsCurveAddRemove:



Public Member Functions

- [CmdSettingsCurveAddRemove](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [CurveNameList](#) &modelCurves)
Constructor for normal creation.
- [CmdSettingsCurveAddRemove](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, [QXmlStreamReader](#) &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) ([QXmlStreamWriter](#) &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.48.1 Detailed Description

Command for [DlgSettingsCurveAddRemove](#).

Definition at line 17 of file CmdSettingsCurveAddRemove.h.

The documentation for this class was generated from the following files:

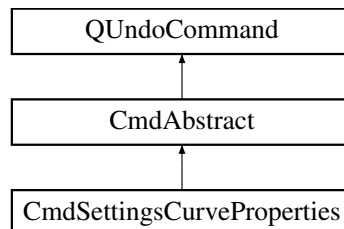
- Cmd/CmdSettingsCurveAddRemove.h
- Cmd/CmdSettingsCurveAddRemove.cpp

5.49 CmdSettingsCurveProperties Class Reference

Command for [DlgSettingsCurveProperties](#).

```
#include <CmdSettingsCurveProperties.h>
```

Inheritance diagram for CmdSettingsCurveProperties:



Public Member Functions

- [CmdSettingsCurveProperties](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [CurveStyles](#) &modelCurveStylesBefore, const [CurveStyles](#) &modelCurveStylesAfter)
Constructor for normal creation.
- [CmdSettingsCurveProperties](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.49.1 Detailed Description

Command for [DlgSettingsCurveProperties](#).

Definition at line 19 of file CmdSettingsCurveProperties.h.

The documentation for this class was generated from the following files:

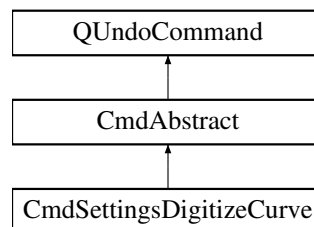
- Cmd/CmdSettingsCurveProperties.h
- Cmd/CmdSettingsCurveProperties.cpp

5.50 CmdSettingsDigitizeCurve Class Reference

Command for [DlgSettingsDigitizeCurve](#).

```
#include <CmdSettingsDigitizeCurve.h>
```

Inheritance diagram for CmdSettingsDigitizeCurve:



Public Member Functions

- [CmdSettingsDigitizeCurve](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurveBefore, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurveAfter)
Constructor for normal creation.
- [CmdSettingsDigitizeCurve](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.50.1 Detailed Description

Command for [DlgSettingsDigitizeCurve](#).

Definition at line 16 of file CmdSettingsDigitizeCurve.h.

The documentation for this class was generated from the following files:

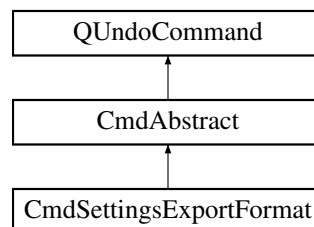
- Cmd/CmdSettingsDigitizeCurve.h
- Cmd/CmdSettingsDigitizeCurve.cpp

5.51 CmdSettingsExportFormat Class Reference

Command for [DlgSettingsExportFormat](#).

```
#include <CmdSettingsExportFormat.h>
```

Inheritance diagram for CmdSettingsExportFormat:



Public Member Functions

- [CmdSettingsExportFormat](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModel](#) &modelExportBefore, const [DocumentModelExportFormat](#) &modelExportAfter)
Constructor for normal creation.
- [CmdSettingsExportFormat](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.51.1 Detailed Description

Command for [DlgSettingsExportFormat](#).

Definition at line 16 of file CmdSettingsExportFormat.h.

The documentation for this class was generated from the following files:

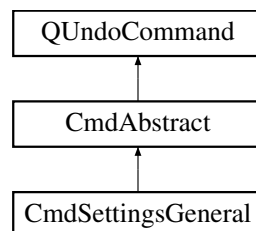
- Cmd/CmdSettingsExportFormat.h
- Cmd/CmdSettingsExportFormat.cpp

5.52 CmdSettingsGeneral Class Reference

Command for [DlgSettingsGeneral](#).

```
#include <CmdSettingsGeneral.h>
```

Inheritance diagram for CmdSettingsGeneral:



Public Member Functions

- [CmdSettingsGeneral](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelGeneral](#) &modelGeneralBefore, const [DocumentModelGeneral](#) &modelGeneralAfter)
Constructor for normal creation.
- [CmdSettingsGeneral](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.52.1 Detailed Description

Command for [DlgSettingsGeneral](#).

Definition at line 16 of file CmdSettingsGeneral.h.

The documentation for this class was generated from the following files:

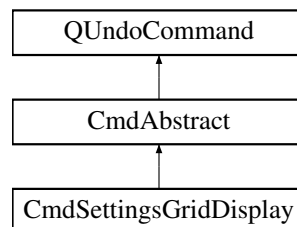
- Cmd/CmdSettingsGeneral.h
- Cmd/CmdSettingsGeneral.cpp

5.53 CmdSettingsGridDisplay Class Reference

Command for [DlgSettingsGridDisplay](#).

```
#include <CmdSettingsGridDisplay.h>
```

Inheritance diagram for CmdSettingsGridDisplay:



Public Member Functions

- [CmdSettingsGridDisplay](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelGridDisplay](#) &modelGridDisplayBefore, const [DocumentModelGridDisplay](#) &modelGridDisplayAfter)
Constructor for normal creation.
- [CmdSettingsGridDisplay](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [QString](#) &cmdDescription, [QXmlStreamReader](#) &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) ([QXmlStreamWriter](#) &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.53.1 Detailed Description

Command for [DlgSettingsGridDisplay](#).

Definition at line 16 of file [CmdSettingsGridDisplay.h](#).

The documentation for this class was generated from the following files:

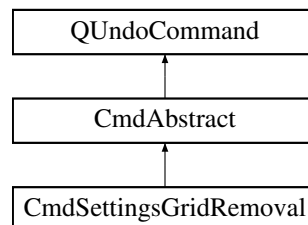
- [Cmd/CmdSettingsGridDisplay.h](#)
- [Cmd/CmdSettingsGridDisplay.cpp](#)

5.54 CmdSettingsGridRemoval Class Reference

Command for [DlgSettingsGridRemoval](#).

```
#include <CmdSettingsGridRemoval.h>
```

Inheritance diagram for CmdSettingsGridRemoval:



Public Member Functions

- [CmdSettingsGridRemoval](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelGridRemoval](#) &modelGridRemovalBefore, const [DocumentModelGridRemoval](#) &modelGridRemovalAfter)
Constructor for normal creation.
- [CmdSettingsGridRemoval](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [QString](#) &cmdDescription, [QXmlStreamReader](#) &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) ([QXmlStreamWriter](#) &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.54.1 Detailed Description

Command for [DlgSettingsGridRemoval](#).

Definition at line 16 of file CmdSettingsGridRemoval.h.

The documentation for this class was generated from the following files:

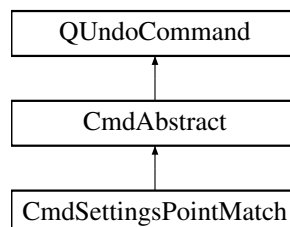
- Cmd/CmdSettingsGridRemoval.h
- Cmd/CmdSettingsGridRemoval.cpp

5.55 CmdSettingsPointMatch Class Reference

Command for [DlgSettingsPointMatch](#).

```
#include <CmdSettingsPointMatch.h>
```

Inheritance diagram for CmdSettingsPointMatch:



Public Member Functions

- [CmdSettingsPointMatch](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelPointMatch](#) &modelPointMatchBefore, const [DocumentModelPointMatch](#) &modelPointMatchAfter)
Constructor for normal creation.
- [CmdSettingsPointMatch](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const QString &cmdDescription, QDomStreamReader &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QDomStreamWriter &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.55.1 Detailed Description

Command for [DlgSettingsPointMatch](#).

Definition at line 16 of file CmdSettingsPointMatch.h.

The documentation for this class was generated from the following files:

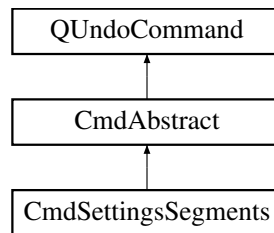
- Cmd/CmdSettingsPointMatch.h
- Cmd/CmdSettingsPointMatch.cpp

5.56 CmdSettingsSegments Class Reference

Command for [DlgSettingsSegments](#).

```
#include <CmdSettingsSegments.h>
```

Inheritance diagram for CmdSettingsSegments:



Public Member Functions

- [CmdSettingsSegments](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [DocumentModelSegments](#) &modelSegmentsBefore, const [DocumentModelSegments](#) &modelSegmentsAfter)
Constructor for normal creation.
- [CmdSettingsSegments](#) ([MainWindow](#) &mainWindow, [Document](#) &document, const [QString](#) &cmdDescription, [QXmlStreamReader](#) &reader)
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) ([QXmlStreamWriter](#) &writer) const
Save commands as xml for later uploading.

Additional Inherited Members

5.56.1 Detailed Description

Command for [DlgSettingsSegments](#).

Definition at line 16 of file CmdSettingsSegments.h.

The documentation for this class was generated from the following files:

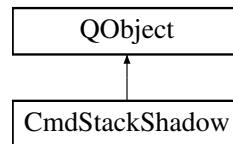
- Cmd/CmdSettingsSegments.h
- Cmd/CmdSettingsSegments.cpp

5.57 CmdStackShadow Class Reference

Command stack that shadows the [CmdMediator](#) command stack at startup when reading commands from an error report file.

```
#include <CmdStackShadow.h>
```

Inheritance diagram for CmdStackShadow:



Public Slots

- void [slotRedo](#) ()
Move next command from list to [CmdMediator](#). Noop if there are no more commands.
- void [slotUndo](#) ()
Throw away every command since trying to reconcile two different command stacks after an undo is too dangerous.

Signals

- void [signalRedo](#) ()
Signal used to emulate a shift-control-z redo command from user during testing.
- void [signalUndo](#) ()
Signal used to emulate a shift-z undo command from user during testing.

Public Member Functions

- [CmdStackShadow](#) ()
Single constructor.
- bool [canRedo](#) () const
Return true if there is a command available.
- void [loadCommands](#) ([MainWindow](#) &mainWindow, [Document](#) &document, [QXmlStreamReader](#) &reader)
Load commands from serialized xml.

5.57.1 Detailed Description

Command stack that shadows the [CmdMediator](#) command stack at startup when reading commands from an error report file.

The commands are loaded into this container rather than [CmdMediator](#), since [CmdMediator](#) would try to execute all the commands immediately. For the best debugging, we want to be able to execute each command one by one. This container nicely stores commands until we want to copy them to [CmdMediator](#) so they can be executed.

This class is not subclassed from [QUndoStack](#) since that class is designed to prevent access to individual commands, to preserve their integrity

This class is not named [CmdMediatorShadow](#) since does not maintain a [Document](#) like [CmdMediator](#), although in some ways that name might be a useful alias

Definition at line 30 of file [CmdStackShadow.h](#).

The documentation for this class was generated from the following files:

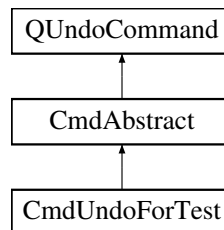
- [Cmd/CmdStackShadow.h](#)
- [Cmd/CmdStackShadow.cpp](#)

5.58 CmdUndoForTest Class Reference

Command for performing Undo during testing.

```
#include <CmdUndoForTest.h>
```

Inheritance diagram for CmdUndoForTest:



Public Member Functions

- [CmdUndoForTest](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#))
Constructor for normal creation.
- [CmdUndoForTest](#) ([MainWindow](#) &[mainWindow](#), [Document](#) &[document](#), const QString &[cmdDescription](#), QXmlStreamReader &[reader](#))
Constructor for parsing error report file xml.
- virtual void [cmdRedo](#) ()
Redo method that is called when QUndoStack is moved one command forward.
- virtual void [cmdUndo](#) ()
Undo method that is called when QUndoStack is moved one command backward.
- virtual void [saveXml](#) (QXmlStreamWriter &[writer](#)) const
Save commands as xml for later uploading.

Additional Inherited Members

5.58.1 Detailed Description

Command for performing Undo during testing.

This command is never created automatically, since when the user triggers an Undo that just results in the command stack getting backed up by one command. This command is manually created by editing an xml test file

Definition at line 20 of file CmdUndoForTest.h.

The documentation for this class was generated from the following files:

- Cmd/CmdUndoForTest.h
- Cmd/CmdUndoForTest.cpp

5.59 ColorFilter Class Reference

Class for filtering image to remove unimportant information.

```
#include <ColorFilter.h>
```

Public Member Functions

- [ColorFilter](#) ()
Single constructor.
- bool [colorCompare](#) (QRgb rgb1, QRgb rgb2) const
See if the two color values are close enough to be considered to be the same.
- void [filterImage](#) (const QImage &imageOriginal, QImage &imageFiltered, ColorFilterMode colorFilterMode, double low, double high, QRgb rgbBackground)
Filter the original image according to the specified filtering parameters.
- QRgb [marginColor](#) (const QImage *image) const
Identify the margin color of the image, which is defined as the most common color in the four margins.
- bool [pixelFilteredIsOn](#) (const QImage &image, int x, int y) const
Return true if specified filtered pixel is on.
- double [pixelToZeroToOneOrMinusOne](#) (ColorFilterMode colorFilterMode, const QColor &pixel, QRgb rgbBackground) const
Return pixel converted according to the current filter parameter, normalized to zero to one.
- bool [pixelUnfilteredIsOn](#) (ColorFilterMode colorFilterMode, const QColor &pixel, QRgb rgbBackground, double low0To1, double high0To1) const
Return true if specified unfiltered pixel is on.
- int [zeroToOneToValue](#) (ColorFilterMode colorFilterMode, double s) const
Inverse of pixelToZeroToOneOrMinusOne.

5.59.1 Detailed Description

Class for filtering image to remove unimportant information.

Definition at line 20 of file ColorFilter.h.

5.59.2 Member Function Documentation

5.59.2.1 QRgb ColorFilter::marginColor (const QImage * image) const

Identify the margin color of the image, which is defined as the most common color in the four margins.

For speed, only pixels in the four borders are examined, with the results from those borders safely representing the most common color of the entire margin areas.

Definition at line 73 of file ColorFilter.cpp.

5.59.2.2 `double ColorFilter::pixelToZeroToOneOrMinusOne (ColorFilterMode colorFilterMode, const QColor & pixel, QRgb rgbBackground) const`

Return pixel converted according to the current filter parameter, normalized to zero to one.

Special case is -1 for a pixel that cannot be converted, like finding hue value for gray scale pixel

Definition at line 171 of file ColorFilter.cpp.

The documentation for this class was generated from the following files:

- Color/ColorFilter.h
- Color/ColorFilter.cpp

5.60 ColorFilterEntry Struct Reference

Helper class so [ColorFilter](#) class can compute the background color.

```
#include <ColorFilterEntry.h>
```

Public Attributes

- QColor [color](#)
Unique color entry.
- unsigned int [count](#)
Number of times this color has appeared.

5.60.1 Detailed Description

Helper class so [ColorFilter](#) class can compute the background color.

Definition at line 13 of file ColorFilterEntry.h.

The documentation for this struct was generated from the following file:

- Color/ColorFilterEntry.h

5.61 ColorFilterHistogram Class Reference

Class that generates a histogram according to the current filter.

```
#include <ColorFilterHistogram.h>
```

Public Member Functions

- [ColorFilterHistogram](#) ()
Single constructor.
- int [binFromPixel](#) (const [ColorFilter](#) &filter, ColorFilterMode colorFilterMode, const QColor &pixel, const QRgb &rgbBackground) const
Compute histogram bin number from pixel according to filter.
- void [generate](#) (const [ColorFilter](#) &filter, double histogramBins[], ColorFilterMode colorFilterMode, const QImage &image, int &maxBinCount) const
Generate the histogram.
- int [valueFromBin](#) (const [ColorFilter](#) &filter, ColorFilterMode colorFilterMode, int bin)
Inverse of binFromPixel.

Static Public Member Functions

- static int [HISTOGRAM_BINS](#) ()
Number of histogram bins.

5.61.1 Detailed Description

Class that generates a histogram according to the current filter.

Definition at line 17 of file ColorFilterHistogram.h.

5.61.2 Member Function Documentation

5.61.2.1 void [ColorFilterHistogram::generate](#) (const [ColorFilter](#) & filter, double histogramBins[], ColorFilterMode colorFilterMode, const QImage & image, int & maxBinCount) const

Generate the histogram.

The resolution is coarse since

1. finer resolution is not needed
2. this smooths out the curve

Definition at line 40 of file ColorFilterHistogram.cpp.

The documentation for this class was generated from the following files:

- Color/ColorFilterHistogram.h
- Color/ColorFilterHistogram.cpp

5.62 ColorFilterSettings Class Reference

Color filter parameters for one curve. For a class, this is handled the same as [LineStyle](#) and [PointStyle](#).

```
#include <ColorFilterSettings.h>
```

Public Member Functions

- [ColorFilterSettings](#) ()
Default constructor only for use when this class is being stored by a container that requires the default constructor.
- [ColorFilterSettings](#) (ColorFilterMode [colorFilterMode](#), int [intensityLow](#), int [intensityHigh](#), int [foregroundLow](#), int [foregroundHigh](#), int [hueLow](#), int [hueHigh](#), int [saturationLow](#), int [saturationHigh](#), int [valueLow](#), int [valueHigh](#))
Normal constructor. The style type and radius are determined by the currently selected [Curve](#).
- [ColorFilterSettings](#) (const [ColorFilterSettings](#) &other)
Copy constructor.
- [ColorFilterSettings](#) (QXmlStreamReader &reader)
Constructor when loading from serialized xml.
- [ColorFilterSettings](#) & operator= (const [ColorFilterSettings](#) &other)
Assignment operator.
- ColorFilterMode [colorFilterMode](#) () const
Get method for filter mode.
- int [foregroundHigh](#) () const
Get method for foreground higher bound.
- int [foregroundLow](#) () const
Get method for foreground lower bound.
- double [high](#) () const
High value of foreground, hue, intensity, saturation or value according to current filter mode, normalized to 0 to 1.
- int [hueHigh](#) () const
Get method for hue higher bound.
- int [hueLow](#) () const
Get method for hue lower bound.
- int [intensityHigh](#) () const
Get method for intensity higher bound.
- int [intensityLow](#) () const
Get method for intensity lower bound.
- void [loadXml](#) (QXmlStreamReader &reader)
Load curve filter to stream.
- double [low](#) () const
Low value of foreground, hue, intensity, saturation or value according to current filter mode, normalized to 0 to 1.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- int [saturationHigh](#) () const
Get method for saturation higher bound.
- int [saturationLow](#) () const
Get method for saturation lower bound.
- void [saveXml](#) (QXmlStreamWriter &writer, const QString &curveName) const
Save curve filter to stream.
- void [setColorFilterMode](#) (ColorFilterMode [colorFilterMode](#))
Set method for filter mode.
- void [setForegroundHigh](#) (int [foregroundHigh](#))
Set method for foreground higher bound.
- void [setForegroundLow](#) (int [foregroundLow](#))
Set method for foreground lower bound.
- void [setHigh](#) (double s0To1)
Set the high value for the current filter mode.
- void [setHueHigh](#) (int [hueHigh](#))
Set method for hue higher bound.

- void [setHueLow](#) (int [hueLow](#))
Set method for hue lower bound.
- void [setIntensityHigh](#) (int [intensityHigh](#))
Set method for intensity higher bound.
- void [setIntensityLow](#) (int [intensityLow](#))
Set method for intensity lower bound.
- void [setLow](#) (double s0To1)
Set the low value for the current filter mode.
- void [setSaturationHigh](#) (int [saturationHigh](#))
Set method for saturation high.
- void [setSaturationLow](#) (int [saturationLow](#))
Set method for saturation low.
- void [setValueHigh](#) (int [valueHigh](#))
Set method for value high.
- void [setValueLow](#) (int [valueLow](#))
Set method for value low.
- int [valueHigh](#) () const
Get method for value high.
- int [valueLow](#) () const
Get method for value low.

Static Public Member Functions

- static [ColorFilterSettings defaultFilter](#) ()
Initial default for any [Curve](#).

5.62.1 Detailed Description

Color filter parameters for one curve. For a class, this is handled the same as [LineStyle](#) and [PointStyle](#).

Definition at line 19 of file [ColorFilterSettings.h](#).

5.62.2 Member Function Documentation

5.62.2.1 double [ColorFilterSettings::high](#) () const

High value of foreground, hue, intensity, saturation or value according to current filter mode, normalized to 0 to 1.

Definition at line 136 of file [ColorFilterSettings.cpp](#).

5.62.2.2 double [ColorFilterSettings::low](#) () const

Low value of foreground, hue, intensity, saturation or value according to current filter mode, normalized to 0 to 1.

Definition at line 218 of file [ColorFilterSettings.cpp](#).

The documentation for this class was generated from the following files:

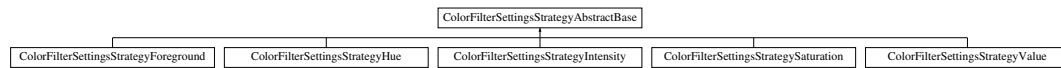
- [Color/ColorFilterSettings.h](#)
- [Color/ColorFilterSettings.cpp](#)

5.63 ColorFilterSettingsStrategyAbstractBase Class Reference

Base class for strategy pattern whose subclasses process the different color filter settings modes (one strategy per mode).

```
#include <ColorFilterSettingsStrategyAbstractBase.h>
```

Inheritance diagram for ColorFilterSettingsStrategyAbstractBase:



Public Member Functions

- [ColorFilterSettingsStrategyAbstractBase](#) ()
Single constructor.
- virtual double [high](#) (const [ColorFilterSettings](#) &colorFilterSettings) const =0
Return the high value normalized to 0 to 1.
- virtual double [low](#) (const [ColorFilterSettings](#) &colorFilterSettings) const =0
Return the low value normalized to 0 to 1.
- virtual void [printStream](#) (const [ColorFilterSettings](#) &colorFilterSettings, QString indentation, QTextStream &str) const =0
Print the low and high values.
- virtual void [setHigh](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)=0
Set the high value given the normalized value.
- virtual void [setLow](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)=0
Set the low value given the normalized value.

5.63.1 Detailed Description

Base class for strategy pattern whose subclasses process the different color filter settings modes (one strategy per mode).

The strategy pattern nicely removes cyclomatic complexity from [ColorFilterSettings](#)

Definition at line 17 of file ColorFilterSettingsStrategyAbstractBase.h.

The documentation for this class was generated from the following files:

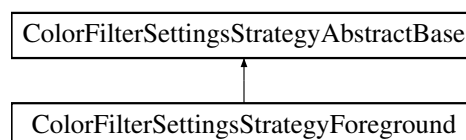
- Color/ColorFilterSettingsStrategyAbstractBase.h
- Color/ColorFilterSettingsStrategyAbstractBase.cpp

5.64 ColorFilterSettingsStrategyForeground Class Reference

Leaf class for foreground strategy for [ColorFilterSettings](#).

```
#include <ColorFilterSettingsStrategyForeground.h>
```

Inheritance diagram for ColorFilterSettingsStrategyForeground:



Public Member Functions

- [ColorFilterSettingsStrategyForeground](#) ()
Single constructor.
- virtual double [high](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the high value normalized to 0 to 1.
- virtual double [low](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the low value normalized to 0 to 1.
- virtual void [printStream](#) (const [ColorFilterSettings](#) &colorFilterSettings, QString indentation, QTextStream &str) const
Print the low and high values.
- virtual void [setHigh](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the high value given the normalized value.
- virtual void [setLow](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the low value given the normalized value.

5.64.1 Detailed Description

Leaf class for foreground strategy for [ColorFilterSettings](#).

Definition at line 13 of file [ColorFilterSettingsStrategyForeground.h](#).

The documentation for this class was generated from the following files:

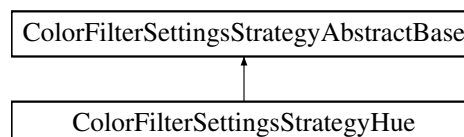
- [Color/ColorFilterSettingsStrategyForeground.h](#)
- [Color/ColorFilterSettingsStrategyForeground.cpp](#)

5.65 ColorFilterSettingsStrategyHue Class Reference

Leaf class for hue strategy for [ColorFilterSettings](#).

```
#include <ColorFilterSettingsStrategyHue.h>
```

Inheritance diagram for [ColorFilterSettingsStrategyHue](#):



Public Member Functions

- [ColorFilterSettingsStrategyHue](#) ()
Single constructor.
- virtual double [high](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the high value normalized to 0 to 1.
- virtual double [low](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the low value normalized to 0 to 1.
- virtual void [printStream](#) (const [ColorFilterSettings](#) &colorFilterSettings, QString indentation, QTextStream &str) const
Print the low and high values.
- virtual void [setHigh](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the high value given the normalized value.
- virtual void [setLow](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the low value given the normalized value.

5.65.1 Detailed Description

Leaf class for hue strategy for [ColorFilterSettings](#).

Definition at line 13 of file ColorFilterSettingsStrategyHue.h.

The documentation for this class was generated from the following files:

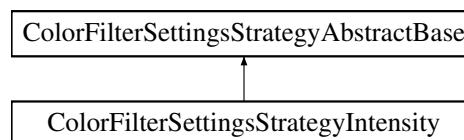
- Color/ColorFilterSettingsStrategyHue.h
- Color/ColorFilterSettingsStrategyHue.cpp

5.66 ColorFilterSettingsStrategyIntensity Class Reference

Leaf class for intensity strategy for [ColorFilterSettings](#).

```
#include <ColorFilterSettingsStrategyIntensity.h>
```

Inheritance diagram for ColorFilterSettingsStrategyIntensity:



Public Member Functions

- [ColorFilterSettingsStrategyIntensity](#) ()
Single constructor.
- virtual double [high](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the high value normalized to 0 to 1.
- virtual double [low](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the low value normalized to 0 to 1.
- virtual void [printStream](#) (const [ColorFilterSettings](#) &colorFilterSettings, QString indentation, QTextStream &str) const
Print the low and high values.
- virtual void [setHigh](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the high value given the normalized value.
- virtual void [setLow](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the low value given the normalized value.

5.66.1 Detailed Description

Leaf class for intensity strategy for [ColorFilterSettings](#).

Definition at line 13 of file ColorFilterSettingsStrategyIntensity.h.

The documentation for this class was generated from the following files:

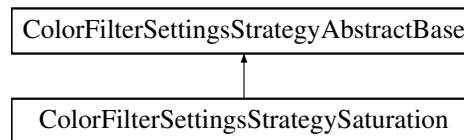
- Color/ColorFilterSettingsStrategyIntensity.h
- Color/ColorFilterSettingsStrategyIntensity.cpp

5.67 ColorFilterSettingsStrategySaturation Class Reference

Leaf class for saturation strategy for [ColorFilterSettings](#).

```
#include <ColorFilterSettingsStrategySaturation.h>
```

Inheritance diagram for ColorFilterSettingsStrategySaturation:



Public Member Functions

- [ColorFilterSettingsStrategySaturation](#) ()
Single constructor.
- virtual double [high](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the high value normalized to 0 to 1.
- virtual double [low](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the low value normalized to 0 to 1.
- virtual void [printStream](#) (const [ColorFilterSettings](#) &colorFilterSettings, QString indentation, QTextStream &str) const
Print the low and high values.
- virtual void [setHigh](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the high value given the normalized value.
- virtual void [setLow](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the low value given the normalized value.

5.67.1 Detailed Description

Leaf class for saturation strategy for [ColorFilterSettings](#).

Definition at line 13 of file ColorFilterSettingsStrategySaturation.h.

The documentation for this class was generated from the following files:

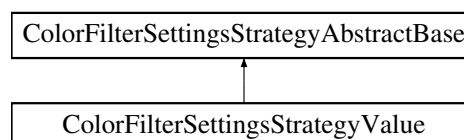
- Color/ColorFilterSettingsStrategySaturation.h
- Color/ColorFilterSettingsStrategySaturation.cpp

5.68 ColorFilterSettingsStrategyValue Class Reference

Leaf class for value strategy for [ColorFilterSettings](#).

```
#include <ColorFilterSettingsStrategyValue.h>
```

Inheritance diagram for ColorFilterSettingsStrategyValue:



Public Member Functions

- [ColorFilterSettingsStrategyValue](#) ()
Single constructor.
- virtual double [high](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the high value normalized to 0 to 1.
- virtual double [low](#) (const [ColorFilterSettings](#) &colorFilterSettings) const
Return the low value normalized to 0 to 1.
- virtual void [printStream](#) (const [ColorFilterSettings](#) &colorFilterSettings, QString indentation, QTextStream &str) const
Print the low and high values.
- virtual void [setHigh](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the high value given the normalized value.
- virtual void [setLow](#) ([ColorFilterSettings](#) &colorFilterSettings, double s0To1)
Set the low value given the normalized value.

5.68.1 Detailed Description

Leaf class for value strategy for [ColorFilterSettings](#).

Definition at line 13 of file ColorFilterSettingsStrategyValue.h.

The documentation for this class was generated from the following files:

- Color/ColorFilterSettingsStrategyValue.h
- Color/ColorFilterSettingsStrategyValue.cpp

5.69 ColorFilterStrategyAbstractBase Class Reference

Base class for strategy pattern whose subclasses process the different color filter settings modes (one strategy per mode).

```
#include <ColorFilterStrategyAbstractBase.h>
```

Inheritance diagram for ColorFilterStrategyAbstractBase:



Public Member Functions

- [ColorFilterStrategyAbstractBase](#) ()
Single constructor.
- virtual double [pixelToZeroToOne](#) (const QColor &pixel, QRgb rgbBackground) const =0
Return a normalized value of 0 to 1 given input pixel.
- virtual int [zeroToOneToValue](#) (double s) const =0
Return the low value normalized to 0 to 1.

5.69.1 Detailed Description

Base class for strategy pattern whose subclasses process the different color filter settings modes (one strategy per mode).

The strategy pattern nicely removes cyclomatic complexity from [ColorFilter](#)

Definition at line 19 of file ColorFilterStrategyAbstractBase.h.

The documentation for this class was generated from the following files:

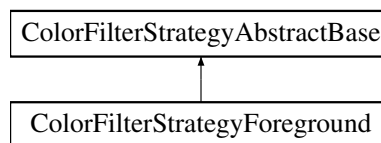
- Color/ColorFilterStrategyAbstractBase.h
- Color/ColorFilterStrategyAbstractBase.cpp

5.70 ColorFilterStrategyForeground Class Reference

Leaf class for foreground strategy for [ColorFilter](#).

```
#include <ColorFilterStrategyForeground.h>
```

Inheritance diagram for ColorFilterStrategyForeground:



Public Member Functions

- [ColorFilterStrategyForeground](#) ()
Single constructor.
- virtual double [pixelToZeroToOne](#) (const QColor &pixel, QRgb rgbBackground) const
Return a normalized value of 0 to 1 given input pixel.
- virtual int [zeroToOneToValue](#) (double s) const
Return the low value normalized to 0 to 1.

5.70.1 Detailed Description

Leaf class for foreground strategy for [ColorFilter](#).

Definition at line 13 of file ColorFilterStrategyForeground.h.

The documentation for this class was generated from the following files:

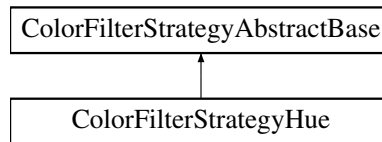
- Color/ColorFilterStrategyForeground.h
- Color/ColorFilterStrategyForeground.cpp

5.71 ColorFilterStrategyHue Class Reference

Leaf class for hue strategy for [ColorFilter](#).

```
#include <ColorFilterStrategyHue.h>
```

Inheritance diagram for ColorFilterStrategyHue:



Public Member Functions

- [ColorFilterStrategyHue](#) ()
Single constructor.
- virtual double [pixelToZeroToOne](#) (const QColor &pixel, QRgb rgbBackground) const
Return a normalized value of 0 to 1 given input pixel.
- virtual int [zeroToOneToValue](#) (double s) const
Return the low value normalized to 0 to 1.

5.71.1 Detailed Description

Leaf class for hue strategy for [ColorFilter](#).

Definition at line 13 of file `ColorFilterStrategyHue.h`.

The documentation for this class was generated from the following files:

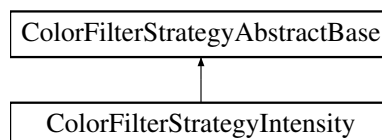
- `Color/ColorFilterStrategyHue.h`
- `Color/ColorFilterStrategyHue.cpp`

5.72 ColorFilterStrategyIntensity Class Reference

Leaf class for intensity strategy for [ColorFilter](#).

```
#include <ColorFilterStrategyIntensity.h>
```

Inheritance diagram for ColorFilterStrategyIntensity:



Public Member Functions

- [ColorFilterStrategyIntensity](#) ()
Single constructor.
- virtual double [pixelToZeroToOne](#) (const QColor &pixel, QRgb rgbBackground) const
Return a normalized value of 0 to 1 given input pixel.
- virtual int [zeroToOneToValue](#) (double s) const
Return the low value normalized to 0 to 1.

5.72.1 Detailed Description

Leaf class for intensity strategy for [ColorFilter](#).

Definition at line 13 of file ColorFilterStrategyIntensity.h.

The documentation for this class was generated from the following files:

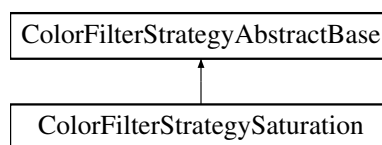
- Color/ColorFilterStrategyIntensity.h
- Color/ColorFilterStrategyIntensity.cpp

5.73 ColorFilterStrategySaturation Class Reference

Leaf class for saturation strategy for [ColorFilter](#).

```
#include <ColorFilterStrategySaturation.h>
```

Inheritance diagram for ColorFilterStrategySaturation:



Public Member Functions

- [ColorFilterStrategySaturation](#) ()
Single constructor.
- virtual double [pixelToZeroToOne](#) (const QColor &pixel, QRgb rgbBackground) const
Return a normalized value of 0 to 1 given input pixel.
- virtual int [zeroToOneToValue](#) (double s) const
Return the low value normalized to 0 to 1.

5.73.1 Detailed Description

Leaf class for saturation strategy for [ColorFilter](#).

Definition at line 13 of file ColorFilterStrategySaturation.h.

The documentation for this class was generated from the following files:

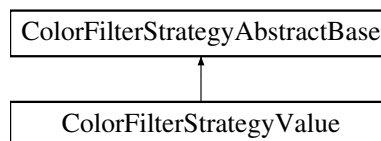
- Color/ColorFilterStrategySaturation.h
- Color/ColorFilterStrategySaturation.cpp

5.74 ColorFilterStrategyValue Class Reference

Leaf class for value strategy for [ColorFilter](#).

```
#include <ColorFilterStrategyValue.h>
```

Inheritance diagram for ColorFilterStrategyValue:



Public Member Functions

- [ColorFilterStrategyValue](#) ()
Single constructor.
- virtual double [pixelToZeroToOne](#) (const QColor &pixel, QRgb rgbBackground) const
Return a normalized value of 0 to 1 given input pixel.
- virtual int [zeroToOneToValue](#) (double s) const
Return the low value normalized to 0 to 1.

5.74.1 Detailed Description

Leaf class for value strategy for [ColorFilter](#).

Definition at line 13 of file ColorFilterStrategyValue.h.

The documentation for this class was generated from the following files:

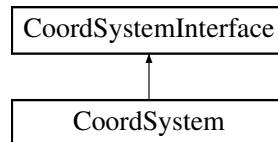
- Color/ColorFilterStrategyValue.h
- Color/ColorFilterStrategyValue.cpp

5.75 CoordSystem Class Reference

Storage of data belonging to one coordinate system.

```
#include <CoordSystem.h>
```

Inheritance diagram for CoordSystem:



Public Member Functions

- [CoordSystem](#) (DocumentAxesPointsRequired documentAxesPointsRequired)
Single constructor.
- [CoordSystem](#) (const QString &fileName)
Constructor for opened Graphs, and error report files. The specified file is opened and read.
- virtual void [addGraphCurveAtEnd](#) (const QString &curveName)
Add new graph curve to the list of existing graph curves.
- virtual void [addPointAxisWithGeneratedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)
Add a single axis point with a generated point identifier.
- virtual void [addPointAxisWithSpecifiedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, const QString &identifier, double ordinal, bool isXOnly)
Add a single axis point with the specified point identifier.
- virtual void [addPointGraphWithGeneratedIdentifier](#) (const QString &curveName, const QPointF &posScreen, QString &generatedIdentifier, double ordinal)
Add a single graph point with a generated point identifier.
- virtual void [addPointGraphWithSpecifiedIdentifier](#) (const QString &curveName, const QPointF &posScreen, const QString &identifier, double ordinal)
Add a single graph point with the specified point identifier. Note that [PointStyle](#) is not applied to the point within the Graph.
- virtual void [addPointsInCurvesGraphs](#) (CurvesGraphs &curvesGraphs)
Add all points identified in the specified [CurvesGraphs](#). See also [removePointsInCurvesGraphs](#).
- virtual void [checkAddPointAxis](#) (const QPointF &posScreen, const QPointF &posGraph, bool &isError, Q<String &errorMessage, bool isXOnly)
Check before calling [addPointAxis](#). Also returns the next available ordinal number (to prevent clashes)
- virtual void [checkEditPointAxis](#) (const QString &pointIdentifier, const QPointF &posScreen, const QPointF &posGraph, bool &isError, QString &errorMessage)
Check before calling [editPointAxis](#).
- virtual const [Curve](#) & [curveAxes](#) () const
Get method for axis curve.
- virtual [Curve](#) * [curveForCurveName](#) (const QString &curveName)
See [CurvesGraphs::curveForCurveName](#), although this also works for [AXIS_CURVE_NAME](#).
- virtual const [Curve](#) * [curveForCurveName](#) (const QString &curveName) const
See [CurvesGraphs::curveForCurveNames](#), although this also works for [AXIS_CURVE_NAME](#).
- virtual const [CurvesGraphs](#) & [curvesGraphs](#) () const
Make all Curves available, read only, for [CmdAbstract](#) classes only.

- virtual QStringList [curvesGraphsNames](#) () const
See [CurvesGraphs::curvesGraphsNames](#).
- virtual int [curvesGraphsNumPoints](#) (const QString &curveName) const
See [CurvesGraphs::curvesGraphsNumPoints](#).
- virtual void [editPointAxis](#) (const QPointF &posGraph, const QString &identifier)
Edit the graph coordinates of a single axis point. Call this after [checkAddPointAxis](#) to guarantee success in this call.
- virtual void [editPointGraph](#) (bool isX, bool isY, double x, double y, const QStringList &identifiers, const [Transformation](#) &transformation)
Edit the graph coordinates of one or more graph points.
- bool [isXOnly](#) (const QString &pointIdentifier) const
Return true if y coordinate is undefined, otherwise x coordinate is undefined in DOCUMENT_AXES_POINT_REQUIRED_4 mode.
- virtual void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)
See [Curve::iterateThroughCurvePoints](#), for the axes curve.
- virtual void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
See [Curve::iterateThroughCurvePoints](#), for the axes curve.
- virtual void [iterateThroughCurveSegments](#) (const QString &curveName, const Functor2wRet< const [Point](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
See [Curve::iterateThroughCurveSegments](#), for any axes or graph curve.
- virtual void [iterateThroughCurvesPointsGraphs](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)
See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
- virtual void [iterateThroughCurvesPointsGraphs](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
- virtual bool [loadCurvesFile](#) (const QString &curvesFile)
Load the curve names in the specified Engauge file into the current graph. This is called near the end of the import process only.
- void [loadPreVersion6](#) (QDataStream &str, double version)
Load from file in pre-version 6 format.
- void [loadVersion6](#) (QXmlStreamReader &reader)
Load from file in version 6 format.
- void [loadVersions7AndUp](#) (QXmlStreamReader &reader, DocumentAxesPointsRequired documentAxesPointsRequired)
Load from file in versions 7 and 8 formats.
- virtual [DocumentModelAxesChecker](#) [modelAxesChecker](#) () const
Get method for [DocumentModelAxesChecker](#).
- virtual [DocumentModelColorFilter](#) [modelColorFilter](#) () const
Get method for [DocumentModelColorFilter](#).
- virtual [DocumentModelCoords](#) [modelCoords](#) () const
Get method for [DocumentModelCoords](#).
- virtual [CurveStyles](#) [modelCurveStyles](#) () const
Get method for [CurveStyles](#).
- virtual [DocumentModelDigitizeCurve](#) [modelDigitizeCurve](#) () const
Get method for [DocumentModelDigitizeCurve](#).
- virtual [DocumentModelExportFormat](#) [modelExport](#) () const
Get method for [DocumentModelExportFormat](#).
- virtual [DocumentModelGeneral](#) [modelGeneral](#) () const
Get method for [DocumentModelGeneral](#).
- virtual [DocumentModelGridDisplay](#) [modelGridDisplay](#) () const

- Get method for [DocumentModelGridDisplay](#).

 - virtual [DocumentModelGridRemoval](#) `modelGridRemoval () const`

Get method for [DocumentModelGridRemoval](#).
- virtual [DocumentModelPointMatch](#) `modelPointMatch () const`

Get method for [DocumentModelPointMatch](#).
- virtual [DocumentModelSegments](#) `modelSegments () const`

Get method for [DocumentModelSegments](#).
- virtual void [movePoint](#) (const QString &pointIdentifier, const QPointF &deltaScreen)

See [Curve::movePoint](#).
- virtual int [nextOrdinalForCurve](#) (const QString &curveName) const

Default next ordinal value for specified curve.
- virtual QPointF [positionGraph](#) (const QString &pointIdentifier) const

See [Curve::positionGraph](#).
- virtual QPointF [positionScreen](#) (const QString &pointIdentifier) const

See [Curve::positionScreen](#).
- virtual void [print](#) () const

Debugging method for printing directly from symbolic debugger.
- virtual void [printStream](#) (QString indentation, QTextStream &str) const

Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual QString [reasonForUnsuccessfulRead](#) () const

Return an informative text message explaining why startup loading failed. Applies if `successfulRead` returns false.
- virtual void [removePointAxis](#) (const QString &identifier)

Perform the opposite of `addPointAxis`.
- virtual void [removePointGraph](#) (const QString &identifier)

Perform the opposite of `addPointGraph`.
- virtual void [removePointsInCurvesGraphs](#) ([CurvesGraphs](#) &[curvesGraphs](#))

Remove all points identified in the specified [CurvesGraphs](#). See also `addPointsInCurvesGraphs`.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const

Save graph to xml.
- virtual QString [selectedCurveName](#) () const

Currently selected curve name. This is used to set the selected curve combobox in [MainWindow](#).
- virtual void [setCurveAxes](#) (const [Curve](#) &[curveAxes](#))

Let [CmdAbstract](#) classes overwrite axes [Curve](#). Applies to current coordinate system.
- virtual void [setCurvesGraphs](#) (const [CurvesGraphs](#) &[curvesGraphs](#))

Let [CmdAbstract](#) classes overwrite [CurvesGraphs](#). Applies to current coordinate system.
- virtual void [setModelAxesChecker](#) (const [DocumentModelAxesChecker](#) &[modelAxesChecker](#))

Set method for [DocumentModelAxesChecker](#).
- virtual void [setModelColorFilter](#) (const [DocumentModelColorFilter](#) &[modelColorFilter](#))

Set method for [DocumentModelColorFilter](#).
- virtual void [setModelCoords](#) (const [DocumentModelCoords](#) &[modelCoords](#))

Set method for [DocumentModelCoords](#).
- virtual void [setModelCurveStyles](#) (const [CurveStyles](#) &[modelCurveStyles](#))

Set method for [CurveStyles](#).
- virtual void [setModelDigitizeCurve](#) (const [DocumentModelDigitizeCurve](#) &[modelDigitizeCurve](#))

Set method for [DocumentModelDigitizeCurve](#).
- virtual void [setModelExport](#) (const [DocumentModelExportFormat](#) &[modelExport](#))

Set method for [DocumentModelExportFormat](#).
- virtual void [setModelGeneral](#) (const [DocumentModelGeneral](#) &[modelGeneral](#))

Set method for [DocumentModelGeneral](#).
- virtual void [setModelGridDisplay](#) (const [DocumentModelGridDisplay](#) &[modelGridDisplay](#))

Set method for [DocumentModelGridDisplay](#).

- virtual void [setModelGridRemoval](#) (const [DocumentModelGridRemoval](#) &[modelGridRemoval](#))
Set method for [DocumentModelGridRemoval](#).
- void [setModelPointMatch](#) (const [DocumentModelPointMatch](#) &[modelPointMatch](#))
Set method for [DocumentModelPointMatch](#).
- virtual void [setModelSegments](#) (const [DocumentModelSegments](#) &[modelSegments](#))
Set method for [DocumentModelSegments](#).
- virtual void [setSelectedCurveName](#) (const QString &[selectedCurveName](#))
Save curve name that is selected for the current coordinate system, for the next time the coordinate system reappears.
- virtual bool [successfulRead](#) () const
Return true if startup loading succeeded. If the loading failed then [reasonForUnsuccessfulRed](#) will explain why.
- virtual void [updatePointOrdinals](#) (const [Transformation](#) &[transformation](#))
Update point ordinals after point addition/removal or dragging.

5.75.1 Detailed Description

Storage of data belonging to one coordinate system.

There can be one or more coordinate systems per graph, and one or more graphs in the image belonging to a [Document](#)

Definition at line 42 of file [CoordSystem.h](#).

5.75.2 Member Function Documentation

5.75.2.1 void [CoordSystem::addPointAxisWithGeneratedIdentifier](#) (const QPointF & *posScreen*, const QPointF & *posGraph*, QString & *identifier*, double *ordinal*, bool *isXOnly*) [\[virtual\]](#)

Add a single axis point with a generated point identifier.

Call this after [checkAddPointAxis](#) to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if graph coordinates have only x coordinate

Implements [CoordSystemInterface](#).

Definition at line 68 of file [CoordSystem.cpp](#).

5.75.2.2 void [CoordSystem::addPointAxisWithSpecifiedIdentifier](#) (const QPointF & *posScreen*, const QPointF & *posGraph*, const QString & *identifier*, double *ordinal*, bool *isXOnly*) [\[virtual\]](#)

Add a single axis point with the specified point identifier.

Call this after [checkAddPointAxis](#) to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if graph coordinates have only x coordinate

Implements [CoordSystemInterface](#).

Definition at line 90 of file CoordSystem.cpp.

5.75.2.3 bool CoordSystem::isXOnly (const QString & *pointIdentifier*) const

Return true if y coordinate is undefined, otherwise x coordinae is undefined in DOCUMENT_AXES_POINT_REQUIRE_4 mode.

Applies to axes points only

Definition at line 296 of file CoordSystem.cpp.

5.75.2.4 void CoordSystem::updatePointOrdinals (const Transformation & *transformation*) [virtual]

Update point ordinals after point addition/removal or dragging.

See GraphicsScene::updatePointOrdinalsAfterDrag. Graph coordinates of point must be up to date

Implements [CoordSystemInterface](#).

Definition at line 943 of file CoordSystem.cpp.

The documentation for this class was generated from the following files:

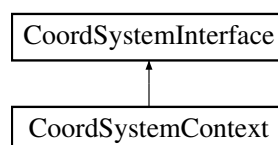
- CoordSystem/CoordSystem.h
- CoordSystem/CoordSystem.cpp

5.76 CoordSystemContext Class Reference

This class plays the role of context class in a state machine, although the 'states' are actually different instantiations of the [CoordSystem](#) class.

```
#include <CoordSystemContext.h>
```

Inheritance diagram for CoordSystemContext:



Public Member Functions

- [CoordSystemContext](#) ()
Default constructor for constructing from opened file.
- void [addCoordSystems](#) (DocumentAxesPointsRequired documentAxesPointsRequired, unsigned int numberCoordSystemToAdd)
Add specified number of coordinate systems to the original one created by the constructor.
- virtual void [addGraphCurveAtEnd](#) (const QString &curveName)
Add new graph curve to the list of existing graph curves.
- virtual void [addPointAxisWithGeneratedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)
Add a single axis point with a generated point identifier.
- virtual void [addPointAxisWithSpecifiedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, const QString &identifier, double ordinal, bool isXOnly)
Add a single axis point with the specified point identifier.
- virtual void [addPointGraphWithGeneratedIdentifier](#) (const QString &curveName, const QPointF &posScreen, QString &generatedIdentifier, double ordinal)
Add a single graph point with a generated point identifier.
- virtual void [addPointGraphWithSpecifiedIdentifier](#) (const QString &curveName, const QPointF &posScreen, const QString &identifier, double ordinal)
Add a single graph point with the specified point identifier. Note that [PointStyle](#) is not applied to the point within the Graph.
- virtual void [addPointsInCurvesGraphs](#) ([CurvesGraphs](#) &[curvesGraphs](#))
Add all points identified in the specified [CurvesGraphs](#). See also [removePointsInCurvesGraphs](#).
- virtual void [checkAddPointAxis](#) (const QPointF &posScreen, const QPointF &posGraph, bool &isError, QString &errorMessage, bool isXOnly)
Check before calling [addPointAxis](#). Also returns the next available ordinal number (to prevent clashes)
- virtual void [checkEditPointAxis](#) (const QString &pointIdentifier, const QPointF &posScreen, const QPointF &posGraph, bool &isError, QString &errorMessage)
Check before calling [editPointAxis](#).
- const [CoordSystem](#) & [coordSystem](#) () const
Current [CoordSystem](#).
- unsigned int [coordSystemCount](#) () const
Number of [CoordSystem](#).
- CoordSystemIndex [coordSystemIndex](#) () const
Index of current [CoordSystem](#).
- virtual const [Curve](#) & [curveAxes](#) () const
Get method for axis curve.
- virtual [Curve](#) * [curveForCurveName](#) (const QString &curveName)
See [CurvesGraphs::curveForCurveName](#), although this also works for [AXIS_CURVE_NAME](#).
- virtual const [Curve](#) * [curveForCurveName](#) (const QString &curveName) const
See [CurvesGraphs::curveForCurveNames](#), although this also works for [AXIS_CURVE_NAME](#).
- virtual const [CurvesGraphs](#) & [curvesGraphs](#) () const
Make all Curves available, read only, for [CmdAbstract](#) classes only.
- virtual QStringList [curvesGraphsNames](#) () const
See [CurvesGraphs::curvesGraphsNames](#).
- virtual int [curvesGraphsNumPoints](#) (const QString &curveName) const
See [CurvesGraphs::curvesGraphsNumPoints](#).
- virtual void [editPointAxis](#) (const QPointF &posGraph, const QString &identifier)
Edit the graph coordinates of a single axis point. Call this after [checkAddPointAxis](#) to guarantee success in this call.
- virtual void [editPointGraph](#) (bool isX, bool isY, double x, double y, const QStringList &identifiers, const [Transformation](#) &transformation)

- Edit the graph coordinates of one or more graph points.*

 - bool [isXOnly](#) (const QString &pointIdentifier) const
True/false if y/x value is empty.
 - virtual void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)
See [Curve::iterateThroughCurvePoints](#), for the axes curve.
 - virtual void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
See [Curve::iterateThroughCurvePoints](#), for the axes curve.
 - virtual void [iterateThroughCurveSegments](#) (const QString &curveName, const Functor2wRet< const [Point](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
See [Curve::iterateThroughCurveSegments](#), for any axes or graph curve.
 - virtual void [iterateThroughCurvesPointsGraphs](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)
See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
 - virtual void [iterateThroughCurvesPointsGraphs](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
 - virtual bool [loadCurvesFile](#) (const QString &curvesFile)
Load the curve names in the specified Engauge file into the current graph. This is called near the end of the import process only.
 - void [loadPreVersion6](#) (QDataStream &str, double version)
Load from file in pre-version 6 format.
 - void [loadVersion6](#) (QXmlStreamReader &reader)
Load from file in version 6 format, into the single [CoordSystem](#).
 - void [loadVersions7AndUp](#) (QXmlStreamReader &reader, DocumentAxesPointsRequired documentAxes↵ PointsRequired)
Load one [CoordSystem](#) from file in version 7 format or newer, into the most recent [CoordSystem](#) which was just created before the call to this method.
 - virtual [DocumentModelAxesChecker](#) [modelAxesChecker](#) () const
Get method for [DocumentModelAxesChecker](#).
 - virtual [DocumentModelColorFilter](#) [modelColorFilter](#) () const
Get method for [DocumentModelColorFilter](#).
 - virtual [DocumentModelCoords](#) [modelCoords](#) () const
Get method for [DocumentModelCoords](#).
 - virtual [CurveStyles](#) [modelCurveStyles](#) () const
Get method for [CurveStyles](#).
 - virtual [DocumentModelDigitizeCurve](#) [modelDigitizeCurve](#) () const
Get method for [DocumentModelDigitizeCurve](#).
 - virtual [DocumentModelExportFormat](#) [modelExport](#) () const
Get method for [DocumentModelExportFormat](#).
 - virtual [DocumentModelGeneral](#) [modelGeneral](#) () const
Get method for [DocumentModelGeneral](#).
 - virtual [DocumentModelGridDisplay](#) [modelGridDisplay](#) () const
Get method for [DocumentModelGridDisplay](#).
 - virtual [DocumentModelGridRemoval](#) [modelGridRemoval](#) () const
Get method for [DocumentModelGridRemoval](#).
 - virtual [DocumentModelPointMatch](#) [modelPointMatch](#) () const
Get method for [DocumentModelPointMatch](#).
 - virtual [DocumentModelSegments](#) [modelSegments](#) () const
Get method for [DocumentModelSegments](#).
 - virtual void [movePoint](#) (const QString &pointIdentifier, const QPointF &deltaScreen)

- See [Curve::movePoint](#).
- virtual int [nextOrdinalForCurve](#) (const QString &curveName) const
Default next ordinal value for specified curve.
- virtual QPointF [positionGraph](#) (const QString &pointIdentifier) const
See [Curve::positionGraph](#).
- virtual QPointF [positionScreen](#) (const QString &pointIdentifier) const
See [Curve::positionScreen](#).
- virtual void [print](#) () const
Debugging method for printing directly from symbolic debugger.
- virtual void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual QString [reasonForUnsuccessfulRead](#) () const
Return an informative text message explaining why startup loading failed. Applies if successfulRead returns false.
- virtual void [removePointAxis](#) (const QString &identifier)
Perform the opposite of addPointAxis.
- virtual void [removePointGraph](#) (const QString &identifier)
Perform the opposite of addPointGraph.
- virtual void [removePointsInCurvesGraphs](#) (CurvesGraphs &curvesGraphs)
Remove all points identified in the specified [CurvesGraphs](#). See also [addPointsInCurvesGraphs](#).
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save graph to xml.
- virtual QString [selectedCurveName](#) () const
Currently selected curve name. This is used to set the selected curve combobox in [MainWindow](#).
- void [setCoordSystemIndex](#) (CoordSystemIndex coordSystemIndex)
Index of current [CoordSystem](#).
- virtual void [setCurveAxes](#) (const Curve &curveAxes)
Let [CmdAbstract](#) classes overwrite axes [Curve](#). Applies to current coordinate system.
- virtual void [setCurvesGraphs](#) (const CurvesGraphs &curvesGraphs)
Let [CmdAbstract](#) classes overwrite [CurvesGraphs](#). Applies to current coordinate system.
- virtual void [setModelAxesChecker](#) (const DocumentModelAxesChecker &modelAxesChecker)
Set method for [DocumentModelAxesChecker](#).
- virtual void [setModelColorFilter](#) (const DocumentModelColorFilter &modelColorFilter)
Set method for [DocumentModelColorFilter](#).
- virtual void [setModelCoords](#) (const DocumentModelCoords &modelCoords)
Set method for [DocumentModelCoords](#).
- virtual void [setModelCurveStyles](#) (const CurveStyles &modelCurveStyles)
Set method for [CurveStyles](#).
- virtual void [setModelDigitizeCurve](#) (const DocumentModelDigitizeCurve &modelDigitizeCurve)
Set method for [DocumentModelDigitizeCurve](#).
- virtual void [setModelExport](#) (const DocumentModelExportFormat &modelExport)
Set method for [DocumentModelExportFormat](#).
- virtual void [setModelGeneral](#) (const DocumentModelGeneral &modelGeneral)
Set method for [DocumentModelGeneral](#).
- virtual void [setModelGridDisplay](#) (const DocumentModelGridDisplay &modelGridDisplay)
Set method for [DocumentModelGridDisplay](#).
- virtual void [setModelGridRemoval](#) (const DocumentModelGridRemoval &modelGridRemoval)
Set method for [DocumentModelGridRemoval](#).
- void [setModelPointMatch](#) (const DocumentModelPointMatch &modelPointMatch)
Set method for [DocumentModelPointMatch](#).
- virtual void [setModelSegments](#) (const DocumentModelSegments &modelSegments)
Set method for [DocumentModelSegments](#).

- virtual void [setSelectedCurveName](#) (const QString &[selectedCurveName](#))
Save curve name that is selected for the current coordinate system, for the next time the coordinate system reappears.
- virtual bool [successfulRead](#) () const
Return true if startup loading succeeded. If the loading failed then [reasonForUnsuccessfulRed](#) will explain why.
- virtual void [updatePointOrdinals](#) (const [Transformation](#) &transformation)
Update point ordinals after point addition/removal or dragging.

5.76.1 Detailed Description

This class plays the role of context class in a state machine, although the 'states' are actually different instantiations of the [CoordSystem](#) class.

At any point in time, one [CoordSystem](#) is active (as selected by the user)

Definition at line 24 of file [CoordSystemContext.h](#).

5.76.2 Member Function Documentation

5.76.2.1 void [CoordSystemContext::addPointAxisWithGeneratedIdentifier](#) (const QPointF & *posScreen*, const QPointF & *posGraph*, QString & *identifier*, double *ordinal*, bool *isXOnly*) [virtual]

Add a single axis point with a generated point identifier.

Call this after [checkAddPointAxis](#) to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if graph coordinates have only x coordinate

Implements [CoordSystemInterface](#).

Definition at line 51 of file [CoordSystemContext.cpp](#).

5.76.2.2 void [CoordSystemContext::addPointAxisWithSpecifiedIdentifier](#) (const QPointF & *posScreen*, const QPointF & *posGraph*, const QString & *identifier*, double *ordinal*, bool *isXOnly*) [virtual]

Add a single axis point with the specified point identifier.

Call this after [checkAddPointAxis](#) to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if graph coordinates have only x coordinate

Implements [CoordSystemInterface](#).

Definition at line 66 of file CoordSystemContext.cpp.

5.76.2.3 void CoordSystemContext::updatePointOrdinals (const Transformation & transformation) [virtual]

Update point ordinals after point addition/removal or dragging.

See GraphicsScene::updatePointOrdinalsAfterDrag. Graph coordinates of point must be up to date

Implements [CoordSystemInterface](#).

Definition at line 584 of file CoordSystemContext.cpp.

The documentation for this class was generated from the following files:

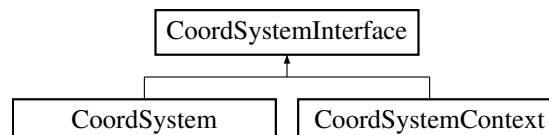
- CoordSystem/CoordSystemContext.h
- CoordSystem/CoordSystemContext.cpp

5.77 CoordSystemInterface Class Reference

Interface common to [CoordSystemContext](#) and [CoordSystem](#) classes.

```
#include <CoordSystemInterface.h>
```

Inheritance diagram for CoordSystemInterface:



Public Member Functions

- [CoordSystemInterface](#) ()
Single constructor.
- virtual void [addGraphCurveAtEnd](#) (const QString &curveName)=0
Add new graph curve to the list of existing graph curves.
- virtual void [addPointAxisWithGeneratedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)=0
Add a single axis point with a generated point identifier.
- virtual void [addPointAxisWithSpecifiedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, const QString &identifier, double ordinal, bool isXOnly)=0
Add a single axis point with the specified point identifier.
- virtual void [addPointGraphWithGeneratedIdentifier](#) (const QString &curveName, const QPointF &posScreen, QString &generatedIdentifier, double ordinal)=0
Add a single graph point with a generated point identifier.
- virtual void [addPointGraphWithSpecifiedIdentifier](#) (const QString &curveName, const QPointF &posScreen, const QString &identifier, double ordinal)=0

Add a single graph point with the specified point identifier. Note that [PointStyle](#) is not applied to the point within the Graph.

- virtual void [addPointsInCurvesGraphs](#) ([CurvesGraphs](#) &[curvesGraphs](#))=0
Add all points identified in the specified [CurvesGraphs](#). See also [removePointsInCurvesGraphs](#).
- virtual void [checkAddPointAxis](#) (const [QPointF](#) &posScreen, const [QPointF](#) &posGraph, bool &isError, [Q](#)←[String](#) &errorMessage, bool isXOnly)=0
Check before calling [addPointAxis](#). Also returns the next available ordinal number (to prevent clashes)
- virtual void [checkEditPointAxis](#) (const [QString](#) &pointIdentifier, const [QPointF](#) &posScreen, const [QPointF](#) &posGraph, bool &isError, [QString](#) &errorMessage)=0
Check before calling [editPointAxis](#).
- virtual const [Curve](#) & [curveAxes](#) () const =0
Get method for axis curve.
- virtual [Curve](#) * [curveForCurveName](#) (const [QString](#) &curveName)=0
See [CurvesGraphs::curveForCurveName](#), although this also works for [AXIS_CURVE_NAME](#).
- virtual const [Curve](#) * [curveForCurveName](#) (const [QString](#) &curveName) const =0
See [CurvesGraphs::curveForCurveNames](#), although this also works for [AXIS_CURVE_NAME](#).
- virtual const [CurvesGraphs](#) & [curvesGraphs](#) () const =0
Make all Curves available, read only, for [CmdAbstract](#) classes only.
- virtual [QStringList](#) [curvesGraphsNames](#) () const =0
See [CurvesGraphs::curvesGraphsNames](#).
- virtual int [curvesGraphsNumPoints](#) (const [QString](#) &curveName) const =0
See [CurvesGraphs::curvesGraphsNumPoints](#).
- virtual void [editPointAxis](#) (const [QPointF](#) &posGraph, const [QString](#) &identifier)=0
Edit the graph coordinates of a single axis point. Call this after [checkAddPointAxis](#) to guarantee success in this call.
- virtual void [editPointGraph](#) (bool isX, bool isY, double x, double y, const [QStringList](#) &identifiers, const [Transformation](#) &transformation)=0
Edit the graph coordinates of one or more graph points.
- virtual void [iterateThroughCurvePointsAxes](#) (const [Functor2wRet](#)< const [QString](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)=0
See [Curve::iterateThroughCurvePoints](#), for the axes curve.
- virtual void [iterateThroughCurvePointsAxes](#) (const [Functor2wRet](#)< const [QString](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const =0
See [Curve::iterateThroughCurvePoints](#), for the axes curve.
- virtual void [iterateThroughCurveSegments](#) (const [QString](#) &curveName, const [Functor2wRet](#)< const [Point](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const =0
See [Curve::iterateThroughCurveSegments](#), for any axes or graph curve.
- virtual void [iterateThroughCurvesPointsGraphs](#) (const [Functor2wRet](#)< const [QString](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)=0
See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
- virtual void [iterateThroughCurvesPointsGraphs](#) (const [Functor2wRet](#)< const [QString](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const =0
See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
- virtual bool [loadCurvesFile](#) (const [QString](#) &curvesFile)=0
Load the curve names in the specified Engauge file into the current graph. This is called near the end of the import process only.
- virtual [DocumentModelAxesChecker](#) [modelAxesChecker](#) () const =0
Get method for [DocumentModelAxesChecker](#).
- virtual [DocumentModelColorFilter](#) [modelColorFilter](#) () const =0
Get method for [DocumentModelColorFilter](#).
- virtual [DocumentModelCoords](#) [modelCoords](#) () const =0
Get method for [DocumentModelCoords](#).
- virtual [CurveStyles](#) [modelCurveStyles](#) () const =0

- Get method for [CurveStyles](#).

 - virtual [DocumentModelDigitizeCurve](#) `modelDigitizeCurve ()` const =0

Get method for [DocumentModelDigitizeCurve](#).
- virtual [DocumentModelExportFormat](#) `modelExport ()` const =0

Get method for [DocumentModelExportFormat](#).
- virtual [DocumentModelGeneral](#) `modelGeneral ()` const =0

Get method for [DocumentModelGeneral](#).
- virtual [DocumentModelGridDisplay](#) `modelGridDisplay ()` const =0

Get method for [DocumentModelGridDisplay](#).
- virtual [DocumentModelGridRemoval](#) `modelGridRemoval ()` const =0

Get method for [DocumentModelGridRemoval](#).
- virtual [DocumentModelPointMatch](#) `modelPointMatch ()` const =0

Get method for [DocumentModelPointMatch](#).
- virtual [DocumentModelSegments](#) `modelSegments ()` const =0

Get method for [DocumentModelSegments](#).
- virtual void [movePoint](#) (const QString &pointIdentifier, const QPointF &deltaScreen)=0

See [Curve::movePoint](#).
- virtual int [nextOrdinalForCurve](#) (const QString &curveName) const =0

Default next ordinal value for specified curve.
- virtual QPointF [positionGraph](#) (const QString &pointIdentifier) const =0

See [Curve::positionGraph](#).
- virtual QPointF [positionScreen](#) (const QString &pointIdentifier) const =0

See [Curve::positionScreen](#).
- virtual void [print](#) () const =0

Debugging method for printing directly from symbolic debugger.
- virtual void [printStream](#) (QString indentation, QTextStream &str) const =0

Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual QString [reasonForUnsuccessfulRead](#) () const =0

Return an informative text message explaining why startup loading failed. Applies if successfulRead returns false.
- virtual void [removePointAxis](#) (const QString &identifier)=0

Perform the opposite of [addPointAxis](#).
- virtual void [removePointGraph](#) (const QString &identifier)=0

Perform the opposite of [addPointGraph](#).
- virtual void [removePointsInCurvesGraphs](#) ([CurvesGraphs](#) &[curvesGraphs](#))=0

Remove all points identified in the specified [CurvesGraphs](#). See also [addPointsInCurvesGraphs](#).
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const =0

Save graph to xml.
- virtual QString [selectedCurveName](#) () const =0

Currently selected curve name. This is used to set the selected curve combobox in [MainWindow](#).
- virtual void [setCurveAxes](#) (const [Curve](#) &[curveAxes](#))=0

Let [CmdAbstract](#) classes overwrite axes [Curve](#). Applies to current coordinate system.
- virtual void [setCurvesGraphs](#) (const [CurvesGraphs](#) &[curvesGraphs](#))=0

Let [CmdAbstract](#) classes overwrite [CurvesGraphs](#). Applies to current coordinate system.
- virtual void [setModelAxesChecker](#) (const [DocumentModelAxesChecker](#) &[modelAxesChecker](#))=0

Set method for [DocumentModelAxesChecker](#).
- virtual void [setModelColorFilter](#) (const [DocumentModelColorFilter](#) &[modelColorFilter](#))=0

Set method for [DocumentModelColorFilter](#).
- virtual void [setModelCoords](#) (const [DocumentModelCoords](#) &[modelCoords](#))=0

Set method for [DocumentModelCoords](#).
- virtual void [setModelCurveStyles](#) (const [CurveStyles](#) &[modelCurveStyles](#))=0

Set method for [CurveStyles](#).

- virtual void [setModelDigitizeCurve](#) (const [DocumentModelDigitizeCurve](#) &[modelDigitizeCurve](#))=0
Set method for [DocumentModelDigitizeCurve](#).
- virtual void [setModelExport](#) (const [DocumentModelExportFormat](#) &[modelExport](#))=0
Set method for [DocumentModelExportFormat](#).
- virtual void [setModelGeneral](#) (const [DocumentModelGeneral](#) &[modelGeneral](#))=0
Set method for [DocumentModelGeneral](#).
- virtual void [setModelGridDisplay](#) (const [DocumentModelGridDisplay](#) &[modelGridDisplay](#))=0
Set method for [DocumentModelGridDisplay](#).
- virtual void [setModelGridRemoval](#) (const [DocumentModelGridRemoval](#) &[modelGridRemoval](#))=0
Set method for [DocumentModelGridRemoval](#).
- virtual void [setModelPointMatch](#) (const [DocumentModelPointMatch](#) &[modelPointMatch](#))=0
Set method for [DocumentModelPointMatch](#).
- virtual void [setModelSegments](#) (const [DocumentModelSegments](#) &[modelSegments](#))=0
Set method for [DocumentModelSegments](#).
- virtual void [setSelectedCurveName](#) (const QString &[selectedCurveName](#))=0
Save curve name that is selected for the current coordinate system, for the next time the coordinate system reappears.
- virtual bool [successfulRead](#) () const =0
Return true if startup loading succeeded. If the loading failed then [reasonForUnsuccessfulRed](#) will explain why.
- virtual void [updatePointOrdinals](#) (const [Transformation](#) &[transformation](#))=0
Update point ordinals after point addition/removal or dragging.

5.77.1 Detailed Description

Interface common to [CoordSystemContext](#) and [CoordSystem](#) classes.

Definition at line 33 of file [CoordSystemInterface.h](#).

5.77.2 Member Function Documentation

- 5.77.2.1 virtual void [CoordSystemInterface::addPointAxisWithGeneratedIdentifier](#) (const QPointF & [posScreen](#), const QPointF & [posGraph](#), QString & [identifier](#), double [ordinal](#), bool [isXOnly](#)) [pure virtual]

Add a single axis point with a generated point identifier.

Call this after [checkAddPointAxis](#) to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if graph coordinates have only x coordinate

Implemented in [CoordSystem](#), and [CoordSystemContext](#).

5.77.2.2 `virtual void CoordSystemInterface::addPointAxisWithSpecifiedIdentifier (const QPointF & posScreen, const QPointF & posGraph, const QString & identifier, double ordinal, bool isXOnly)` `[pure virtual]`

Add a single axis point with the specified point identifier.

Call this after `checkAddPointAxis` to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if graph coordinates have only x coordinate

Implemented in [CoordSystem](#), and [CoordSystemContext](#).

5.77.2.3 `virtual void CoordSystemInterface::updatePointOrdinals (const Transformation & transformation)` `[pure virtual]`

Update point ordinals after point addition/removal or dragging.

See `GraphicsScene::updatePointOrdinalsAfterDrag`. Graph coordinates of point must be up to date

Implemented in [CoordSystem](#), and [CoordSystemContext](#).

The documentation for this class was generated from the following files:

- `CoordSystem/CoordSystemInterface.h`
- `CoordSystem/CoordSystemInterface.cpp`

5.78 Correlation Class Reference

Fast cross correlation between two functions.

```
#include <Correlation.h>
```

Public Member Functions

- [Correlation](#) (int N)
Single constructor. Slow memory allocations are done once and then reused repeatedly.
- void [correlateWithShift](#) (int N, const double function1[], const double function2[], int &binStartMax, double &corrMax, double correlations[]) const
Return the shift in function1 that best aligns that function with function2.
- void [correlateWithoutShift](#) (int N, const double function1[], const double function2[], double &corrMax) const
Return the correlation of the two functions, without any shift.

5.78.1 Detailed Description

Fast cross correlation between two functions.

We do not use `complex.h` along with `fftw3.h` since then the complex numbers will be native, which would then require platform-dependent code

Definition at line 14 of file `Correlation.h`.

5.78.2 Member Function Documentation

5.78.2.1 `void Correlation::correlateWithoutShift (int N, const double function1[], const double function2[], double & corrMax) const`

Return the correlation of the two functions, without any shift.

The functions are normalized internally.

Definition at line 131 of file `Correlation.cpp`.

5.78.2.2 `void Correlation::correlateWithShift (int N, const double function1[], const double function2[], int & binStartMax, double & corrMax, double correlations[]) const`

Return the shift in `function1` that best aligns that function with `function2`.

The functions are normalized internally. The correlations vector, as a function of shift, is returned for logging

Definition at line 44 of file `Correlation.cpp`.

The documentation for this class was generated from the following files:

- `Correlation/Correlation.h`
- `Correlation/Correlation.cpp`

5.79 CursorFactory Class Reference

Create standard cross cursor, or custom cursor, according to settings.

```
#include <CursorFactory.h>
```

Public Member Functions

- [CursorFactory](#) ()
Single constructor.
- `QCursor generate` (const [DocumentModelDigitizeCurve](#) &`modelDigitizeCurve`) const
Factory method to generate standard or custom cursor.

5.79.1 Detailed Description

Create standard cross cursor, or custom cursor, according to settings.

Definition at line 15 of file CursorFactory.h.

The documentation for this class was generated from the following files:

- Cursor/CursorFactory.h
- Cursor/CursorFactory.cpp

5.80 Curve Class Reference

Container for one set of digitized Points.

```
#include <Curve.h>
```

Public Member Functions

- [Curve](#) (const QString &curveName, const [ColorFilterSettings](#) &colorFilterSettings, const [CurveStyle](#) &curveStyle)
Constructor from scratch.
- [Curve](#) (QDataStream &str)
Constructor from serialized binary pre-version 6 file.
- [Curve](#) (QXmlStreamReader &reader)
Constructor for use when loading from serialized xml.
- [Curve](#) (const [Curve](#) &curve)
Copy constructor. Copying a [Curve](#) only helps for making a copy, since access to any Points inside must be via functor.
- [Curve](#) & operator= (const [Curve](#) &curve)
Assignment constructor.
- void [addPoint](#) ([Point](#) point)
Add [Point](#) to this [Curve](#).
- [ColorFilterSettings](#) colorFilterSettings () const
Return the color filter.
- QString [curveName](#) () const
Name of this [Curve](#).
- [CurveStyle](#) curveStyle () const
Return the curve style.
- void [editPointAxis](#) (const QPointF &posGraph, const QString &identifier)
Edit the graph coordinates of an axis point. This method does not apply to a graph point.
- void [editPointGraph](#) (bool isX, bool isY, double x, double y, const QStringList &identifiers, const [Transformation](#) &transformation)
Edit the graph coordinates of one or more graph points. This method does not apply to an axis point.
- void [exportToClipboard](#) (const QHash< QString, bool > &selectedHash, const [Transformation](#) &transformation, QTextStream &strCsv, QTextStream &strHtml, [CurvesGraphs](#) &curvesGraphs) const
Export points in this [Curve](#) found in the specified point list.
- bool [isXOnly](#) (const QString &pointIdentifier) const
Determine if specified point has just x coordinate. Otherwise has just y coordinate, or both x and y coordinates.

- void [iterateThroughCurvePoints](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
Apply functor to Points on [Curve](#).
- void [iterateThroughCurveSegments](#) (const Functor2wRet< const [Point](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
Apply functor to successive Points, as line segments, on [Curve](#). This could be a bit slow.
- void [movePoint](#) (const QString &pointIdentifier, const QPointF &deltaScreen)
Translate the position of a point by the specified distance vector.
- int [numPoints](#) () const
Number of points.
- const Points [points](#) () const
Return a shallow copy of the Points.
- QPointF [positionGraph](#) (const QString &pointIdentifier) const
Return the position, in graph coordinates, of the specified [Point](#).
- QPointF [positionScreen](#) (const QString &pointIdentifier) const
Return the position, in screen coordinates, of the specified [Point](#).
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [removePoint](#) (const QString &identifier)
Perform the opposite of addPointAtEnd.
- void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize curve.
- void [setColorFilterSettings](#) (const [ColorFilterSettings](#) &colorFilterSettings)
Set color filter.
- void [setCurveName](#) (const QString &curveName)
Change the curve name.
- void [setCurveStyle](#) (const [CurveStyle](#) &curveStyle)
Set curve style.
- void [updatePointOrdinals](#) (const [Transformation](#) &transformation)
See [CurveGraphs::updatePointOrdinals](#).

5.80.1 Detailed Description

Container for one set of digitized Points.

Definition at line 32 of file [Curve.h](#).

5.80.2 Member Function Documentation

5.80.2.1 void [Curve::updatePointOrdinals](#) (const [Transformation](#) & *transformation*)

See [CurveGraphs::updatePointOrdinals](#).

Same algorithm as [GraphicsLineForCurve::updatePointOrdinalsAfterDrag](#), although graph coordinates of points have been updated before this is called so the graph coordinates are not updated by this method

Definition at line 559 of file [Curve.cpp](#).

The documentation for this class was generated from the following files:

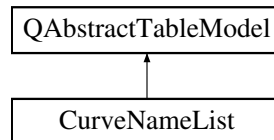
- [Curve/Curve.h](#)
- [Curve/Curve.cpp](#)

5.81 CurveNameList Class Reference

Model for [DlgSettingsCurveAddRemove](#) and [CmdSettingsCurveAddRemove](#).

```
#include <CurveNameList.h>
```

Inheritance diagram for CurveNameList:



Public Member Functions

- [CurveNameList](#) ()
Default constructor.
- virtual int [columnCount](#) (const QModelIndex &parent=QModelIndex()) const
Columns are current curve name in first column, and original curve name in second column.
- bool [containsCurveNameCurrent](#) (const QString &curveName) const
Return true if specified curve name is already in the list.
- virtual QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const
Retrieve data from model.
- virtual Qt::ItemFlags [flags](#) (const QModelIndex &index) const
Override normal flags with additional editing flags.
- virtual bool [insertRows](#) (int row, int count, const QModelIndex &parent=QModelIndex())
Insert one row.
- virtual bool [removeRows](#) (int row, int count, const QModelIndex &parent)
Remove one row.
- virtual int [rowCount](#) (const QModelIndex &parent=QModelIndex()) const
One row per curve name.
- virtual bool [setData](#) (const QModelIndex &index, const QVariant &value, int role=Qt::EditRole)
Store one curve name data.
- virtual Qt::DropActions [supportedDropActions](#) () const
Allow dragging for reordering.

5.81.1 Detailed Description

Model for [DlgSettingsCurveAddRemove](#) and [CmdSettingsCurveAddRemove](#).

This is displayed as a QListView, with visible first column showing current curve name. Second column is hidden with curve name at the start of editing, or empty if none.

Definition at line 16 of file CurveNameList.h.

The documentation for this class was generated from the following files:

- Curve/CurveNameList.h
- Curve/CurveNameList.cpp

5.82 CurveNameListEntry Class Reference

Utility class for converting the QVariant in [CurveNameList](#) to/from the curve names as QStrings, for use by the [CurveNameList](#) model class.

```
#include <CurveNameListEntry.h>
```

Public Member Functions

- [CurveNameListEntry](#) ()
Constructor for empty entry.
- [CurveNameListEntry](#) (const QString &[curveNameCurrent](#), const QString &[curveNameOriginal](#), int [numPoints](#))
Constructor for converting to QVariant.
- [CurveNameListEntry](#) (const QString &fromText)
Constructor for converting from QVariant.
- QString [curveNameCurrent](#) () const
Curve name displayed in [DlgSettingsCurveAddRemove](#).
- QString [curveNameOriginal](#) () const
Original curve name in document. Empty if there was no original curve.
- bool [entryHasNotBeenPopulated](#) () const
Return true if entry is unpopulated. This is true between insertRows (where added to model) and setData (where fields are set)
- int [numPoints](#) () const
Number of points in curve.
- void [setCurveNameCurrent](#) (const QString &[curveNameCurrent](#))
Set method for current curve name.
- void [setCurveNameOriginal](#) (const QString &[curveNameOriginal](#))
Set method for original curve name.
- void [setNumPoints](#) (int [numPoints](#))
Set method for point count.
- QString [toString](#) () const
QString for creating QVariant.

Static Public Member Functions

- static int [COL_CURVE_NAME_CURRENT](#) ()
Get method for current curve name constant.
- static int [COL_CURVE_NAME_ORIGINAL](#) ()
Get method for original curve name constant.
- static int [COL_NUM_POINTS](#) ()
Get method for number of points constant.

5.82.1 Detailed Description

Utility class for converting the QVariant in [CurveNameList](#) to/from the curve names as QStrings, for use by the [CurveNameList](#) model class.

Subclassing QVariant brings up difficult challenges, so the QString userType was chosen.

Each entry has the current curve name, original curve name and point count, separated by a delimiter.

Definition at line 19 of file CurveNameListEntry.h.

The documentation for this class was generated from the following files:

- Curve/CurveNameListEntry.h
- Curve/CurveNameListEntry.cpp

5.83 CurveSettingsInt Class Reference

Internal settings for one curve, such as [LineStyle](#), [PointStyle](#) and CurveFilter.

```
#include <CurveSettingsInt.h>
```

Public Member Functions

- [CurveSettingsInt](#) (const [ColorFilterSettings](#) &colorFilterSettings, const [PointStyle](#) &pointStyle, const [LineStyle](#) &lineStyle, CurveConnectAs curveConnectAs)
Single constructor.
- CurveConnectAs [curveConnectAs](#) () const
Get method for connection method.
- [ColorFilterSettings](#) [colorFilterSettings](#) () const
Get method for color filter.
- [LineStyle](#) [lineStyle](#) () const
Get method for line style.
- [PointStyle](#) [pointStyle](#) () const
Get method for point style.

5.83.1 Detailed Description

Internal settings for one curve, such as [LineStyle](#), [PointStyle](#) and CurveFilter.

These settings are used only internally by [Curve](#), and are not related to the DlgSettings classes at all

Definition at line 17 of file CurveSettingsInt.h.

The documentation for this class was generated from the following files:

- Curve/CurveSettingsInt.h
- Curve/CurveSettingsInt.cpp

5.84 CurvesGraphs Class Reference

Container for all graph curves. The axes point curve is external to this class.

```
#include <CurvesGraphs.h>
```

Public Member Functions

- void [addGraphCurveAtEnd](#) ([Curve](#) curve)
Append new graph [Curve](#) to end of [Curve](#) list.
- void [addPoint](#) (const [Point](#) &point)
Append new [Point](#) to the specified [Curve](#).
- [Curve](#) * [curveForCurveName](#) (const QString &curveName)
Return the axis or graph curve for the specified curve name.
- const [Curve](#) * [curveForCurveName](#) (const QString &curveName) const
Return the axis or graph curve for the specified curve name.
- QStringList [curvesGraphsNames](#) () const
List of graph curve names.
- int [curvesGraphsNumPoints](#) (const QString &curveName) const
[Point](#) count.
- void [editPointGraph](#) (bool isX, bool isY, double x, double y, const QStringList &identifiers, const [Transformation](#) &transformation)
Set the x and/or y coordinate values of the specified points.
- void [iterateThroughCurvePoints](#) (const QString &curveNameWanted, const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)
Apply functor to [Points](#) in the specified axis or graph [Curve](#).
- void [iterateThroughCurveSegments](#) (const QString &curveNameWanted, const Functor2wRet< const [Point](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
Apply functor to segments on the specified axis or graph [Curve](#).
- void [iterateThroughCurvesPoints](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback)
Apply functor to [Points](#) on all of the [Curves](#).
- void [iterateThroughCurvesPoints](#) (const Functor2wRet< const QString &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
Apply functor to [Points](#) on all of the [Curves](#).
- void [loadPreVersion6](#) (QDataStream &str)
Load from serialized binary pre-version 6 file.
- void [loadXml](#) (QXmlStreamReader &reader)
Load from serialized xml post-version 5 file.
- int [numCurves](#) () const
Current number of graphs curves.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [removePoint](#) (const QString &pointIdentifier)
Remove the [Point](#) from its [Curve](#).
- void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize curves.
- void [updatePointOrdinals](#) (const [Transformation](#) &transformation)
Update point ordinals to be consistent with their [CurveStyle](#) and x/theta coordinate.

5.84.1 Detailed Description

Container for all graph curves. The axes point curve is external to this class.

Definition at line 24 of file CurvesGraphs.h.

The documentation for this class was generated from the following files:

- Curve/CurvesGraphs.h
- Curve/CurvesGraphs.cpp

5.85 CurveStyle Class Reference

Container for [LineStyle](#) and [PointStyle](#) for one [Curve](#).

```
#include <CurveStyle.h>
```

Public Member Functions

- [CurveStyle](#) ()
Default constructor.
- [CurveStyle](#) (const [LineStyle](#) &lineStyle, const [PointStyle](#) &pointStyle)
Constructor with styles.
- [LineStyle](#) lineStyle () const
Get method for [LineStyle](#).
- QString loadXml (QXmlStreamReader &reader)
Load from serialized xml. Returns the curve name.
- [PointStyle](#) pointStyle () const
Get method for [PointStyle](#).
- void printStream (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void saveXml (QXmlStreamWriter &writer, const QString &curveName) const
Serialize to xml.
- void setLineColor (ColorPalette lineColor)
Set method for line color in specified curve.
- void setLineConnectAs (CurveConnectAs curveConnectAs)
Set method for connect as method for lines in specified curve.
- void setLineStyle (const [LineStyle](#) &lineStyle)
Set method for [LineStyle](#).
- void setLineWidth (int width)
Set method for line width in specified curve.
- void setPointColor (ColorPalette curveColor)
Set method curve point color in specified curve.
- void setPointLineWidth (int width)
Set method for curve point perimeter line width.
- void setPointRadius (int radius)
Set method for curve point radius.
- void setPointShape (PointShape shape)
Set method for curve point shape in specified curve.
- void setPointStyle (const [PointStyle](#) &pointStyle)
Set method for [PointStyle](#).

5.85.1 Detailed Description

Container for [LineStyle](#) and [PointStyle](#) for one [Curve](#).

Definition at line 18 of file [CurveStyle.h](#).

The documentation for this class was generated from the following files:

- [Curve/CurveStyle.h](#)
- [Curve/CurveStyle.cpp](#)

5.86 CurveStyles Class Reference

Model for [DlgSettingsCurveProperties](#) and [CmdSettingsCurveProperties](#).

```
#include <CurveStyles.h>
```

Public Member Functions

- [CurveStyles](#) ()
Default constructor.
- [CurveStyles](#) (const [CoordSystem](#) &coordSystem)
Initial constructor from [Document](#).
- [CurveStyles](#) (const [CurveStyles](#) &other)
Copy constructor.
- [CurveStyles](#) & operator= (const [CurveStyles](#) &other)
Assignment constructor.
- QList [curveNames](#) () const
List of all curve names.
- [CurveStyle](#) [curveStyle](#) (const QString &curveName) const
CurveStyle in specified curve.
- ColorPalette [lineColor](#) (const QString &curveName) const
Get method for line color in specified curve.
- CurveConnectAs [lineConnectAs](#) (const QString &curveName) const
Get method for connect as method for lines in specified curve.
- const [LineStyle](#) [lineStyle](#) (const QString &curveName) const
Get method for copying one line style in one step.
- int [lineWidth](#) (const QString &curveName) const
Get method for line width in specified curve.
- void [loadXml](#) (QXmlStreamReader &reader)
Load from serialized xml.
- ColorPalette [pointColor](#) (const QString &curveName) const
Get method for curve point color in specified curve.
- bool [pointIsCircle](#) (const QString &curveName) const
Get method for curve point is circle in specified curve.
- int [pointLineWidth](#) (const QString &curveName) const
Get method for curve point line width.
- QPolygonF [pointPolygon](#) (const QString &curveName) const
Get method for curve point polygon in specified curve.

- int [pointRadius](#) (const QString &curveName) const
Get method for curve point radius.
- PointShape [pointShape](#) (const QString &curveName) const
Get method for curve point shape.
- const [PointStyle pointStyle](#) (const QString &curveName) const
Get method for copying one point style. Cannot return just a reference or else there is a warning about returning reference to temporary.
- void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize to xml.
- void [setCurveStyle](#) (const QString &curveName, const [CurveStyle](#) &curveStyle)
Set method for curve style.
- void [setLineColor](#) (const QString &curveName, ColorPalette [lineColor](#))
Set method for line color in specified curve.
- void [setLineConnectAs](#) (const QString &curveName, CurveConnectAs curveConnectAs)
Set method for connect as method for lines in specified curve.
- void [setLineWidth](#) (const QString &curveName, int width)
Set method for line width in specified curve.
- void [setPointColor](#) (const QString &curveName, ColorPalette curveColor)
Set method curve point color in specified curve.
- void [setPointIsCircle](#) (const QString &curveName, bool [pointIsCircle](#))
Set method for curve point is circle in specified curve.
- void [setPointLineWidth](#) (const QString &curveName, int width)
Set method for curve point perimeter line width.
- void [setPointRadius](#) (const QString &curveName, int radius)
Set method for curve point radius.
- void [setPointShape](#) (const QString &curveName, PointShape shape)
Set method for curve point shape in specified curve.

5.86.1 Detailed Description

Model for [DlgSettingsCurveProperties](#) and [CmdSettingsCurveProperties](#).

Definition at line 22 of file CurveStyles.h.

The documentation for this class was generated from the following files:

- Curve/CurveStyles.h
- Curve/CurveStyles.cpp

5.87 DigitizeStateAbstractBase Class Reference

Base class for all digitizing states. This serves as an interface to [DigitizeStateContext](#).

```
#include <DigitizeStateAbstractBase.h>
```

Inheritance diagram for DigitizeStateAbstractBase:



Public Member Functions

- [DigitizeStateAbstractBase](#) ([DigitizeStateContext](#) &context)
Single constructor.
- virtual QString [activeCurve](#) () const =0
Name of the active [Curve](#). This can include `AXIS_CURVE_NAME`.
- virtual void [begin](#) ([CmdMediator](#) *cmdMediator, [DigitizeState](#) previousState)=0
Method that is called at the exact moment a state is entered.
- [DigitizeStateContext](#) & context ()
Reference to the [DigitizeStateContext](#) that contains all the [DigitizeStateAbstractBase](#) subclasses, without const.
- const [DigitizeStateContext](#) & context () const
Reference to the [DigitizeStateContext](#) that contains all the [DigitizeStateAbstractBase](#) subclasses, without const.
- virtual void [end](#) ()=0
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const QString &pointIdentifier)=0
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const QStringList &pointIdentifiers)=0
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)=0
Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.
- virtual void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)=0
Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)=0
Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point Match](#) mode is worthwhile.
- virtual void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, QPointF pos)=0
Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, QPointF pos)=0
Handle a mouse release that was intercepted earlier.
- void [setCursor](#) ([CmdMediator](#) *cmdMediator)
Update the cursor according to the current state.
- virtual QString [state](#) () const =0
State name for debugging.
- virtual void [updateAfterPointAddition](#) ()=0
Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurve)=0
Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)=0
Update the segments given the new settings.

Protected Member Functions

- virtual QCursor [cursor](#) ([CmdMediator](#) *cmdMediator) const =0
Returns the state-specific cursor shape.

5.87.1 Detailed Description

Base class for all digitizing states. This serves as an interface to [DigitizeStateContext](#).

Definition at line 37 of file [DigitizeStateAbstractBase.h](#).

5.87.2 Member Function Documentation

5.87.2.1 `virtual void DigitizeStateAbstractBase::begin (CmdMediator * cmdMediator, DigitizeState previousState)`
`[pure virtual]`

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The previousState value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implemented in [DigitizeStateColorPicker](#), [DigitizeStatePointMatch](#), [DigitizeStateSegment](#), [DigitizeStateSelect](#), [DigitizeStateAxis](#), [DigitizeStateCurve](#), and [DigitizeStateEmpty](#).

The documentation for this class was generated from the following files:

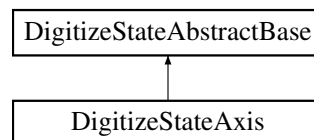
- DigitizeState/DigitizeStateAbstractBase.h
- DigitizeState/DigitizeStateAbstractBase.cpp

5.88 DigitizeStateAxis Class Reference

Digitizing state for digitizing one axis point at a time.

```
#include <DigitizeStateAxis.h>
```

Inheritance diagram for DigitizeStateAxis:



Public Member Functions

- [DigitizeStateAxis](#) ([DigitizeStateContext](#) &context)
Single constructor.
- virtual QString [activeCurve](#) () const
Name of the active [Curve](#). This can include `AXIS_CURVE_NAME`.
- virtual void [begin](#) ([CmdMediator](#) *cmdMediator, DigitizeState previousState)
Method that is called at the exact moment a state is entered.
- virtual QCursor [cursor](#) ([CmdMediator](#) *cmdMediator) const
Returns the state-specific cursor shape.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const QString &pointIdentifier)
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const QStringList &pointIdentifiers)
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)

- Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.*

 - virtual void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)

Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)

Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point Match](#) mode is worthwhile.
- virtual void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)

Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)

Handle a mouse release that was intercepted earlier.
- virtual QString [state](#) () const

State name for debugging.
- virtual void [updateAfterPointAddition](#) ()

Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurve)

Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)

Update the segments given the new settings.

Additional Inherited Members

5.88.1 Detailed Description

Digitizing state for digitizing one axis point at a time.

Once three axis points are defined, those points define an affine transformation from pixel screen coordinates to graph coordinates.

Definition at line 16 of file DigitizeStateAxis.h.

5.88.2 Member Function Documentation

5.88.2.1 void [DigitizeStateAxis::begin](#) ([CmdMediator](#) * *cmdMediator*, [DigitizeState](#) *previousState*) [virtual]

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The previousState value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implements [DigitizeStateAbstractBase](#).

Definition at line 38 of file DigitizeStateAxis.cpp.

The documentation for this class was generated from the following files:

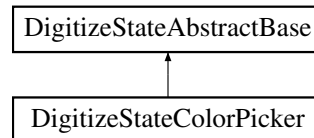
- DigitizeState/DigitizeStateAxis.h
- DigitizeState/DigitizeStateAxis.cpp

5.89 DigitizeStateColorPicker Class Reference

Digitizing state for selecting a color for [DigitizeStateSegment](#).

```
#include <DigitizeStateColorPicker.h>
```

Inheritance diagram for DigitizeStateColorPicker:



Public Member Functions

- [DigitizeStateColorPicker](#) ([DigitizeStateContext](#) &context)
Single constructor.
- virtual QString [activeCurve](#) () const
Name of the active [Curve](#). This can include `AXIS_CURVE_NAME`.
- virtual void [begin](#) ([CmdMediator](#) *cmdMediator, [DigitizeState](#) previousState)
Method that is called at the exact moment a state is entered.
- virtual [QCursor](#) [cursor](#) ([CmdMediator](#) *cmdMediator) const
Returns the state-specific cursor shape.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const QString &pointIdentifier)
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const QStringList &pointIdentifiers)
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)
Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.
- virtual void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)
Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point](#) Match mode is worthwhile.
- virtual void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse release that was intercepted earlier.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateAfterPointAddition](#) ()
Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurve)
Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)
Update the segments given the new settings.

Additional Inherited Members

5.89.1 Detailed Description

Digitizing state for selecting a color for [DigitizeStateSegment](#).

The basic strategy is that this class acts like a special case of `DlgSettingsFilter`. Specifically, the pixel just selected by a mouse click is used to change the segment filter for the currently specified curve

Definition at line 21 of file `DigitizeStateColorPicker.h`.

5.89.2 Member Function Documentation

5.89.2.1 `void DigitizeStateColorPicker::begin (CmdMediator * cmdMediator, DigitizeState previousState)` `[virtual]`

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The `previousState` value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implements [DigitizeStateAbstractBase](#).

Definition at line 37 of file `DigitizeStateColorPicker.cpp`.

The documentation for this class was generated from the following files:

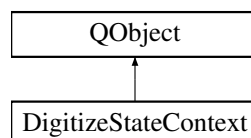
- `DigitizeState/DigitizeStateColorPicker.h`
- `DigitizeState/DigitizeStateColorPicker.cpp`

5.90 DigitizeStateContext Class Reference

Container for all [DigitizeStateAbstractBase](#) subclasses. This functions as the context class in a standard state machine implementation.

```
#include <DigitizeStateContext.h>
```

Inheritance diagram for `DigitizeStateContext`:



Public Member Functions

- [DigitizeStateContext](#) ([MainWindow](#) &mainWindow, [QGraphicsView](#) &view, bool isGnuplot)
Single constructor.
- [QString](#) [activeCurve](#) () const
Curve name for active Curve. This can include AXIS_CURVE_NAME, and empty string.
- void [appendNewCmd](#) ([CmdMediator](#) *cmdMediator, [QUndoCommand](#) *cmd)
Append just-created QUndoCommand to command stack. This is called from [DigitizeStateAbstractBase](#) subclasses.
- void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const [QString](#) &pointIdentifier)
See [DigitizeStateAbstractBase::handleContextMenuEventAxis](#).
- void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const [QStringList](#) &pointIdentifiers)
See [DigitizeStateAbstractBase::handleContextMenuEventGraph](#).
- void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)
See [DigitizeStateAbstractBase::handleCurveChange](#).
- void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, [Qt::Key](#) key, bool atLeastOneSelectedItem)
See [DigitizeStateAbstractBase::handleKeyPress](#).
- void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) pos)
See [DigitizeStateAbstractBase::handleMouseMove](#).
- void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) pos)
See [DigitizeStateAbstractBase::handleMousePress](#).
- void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) pos)
See [DigitizeStateAbstractBase::handleMouseRelease](#).
- bool [isGnuplot](#) () const
Get method for gnuplot flag.
- [MainWindow](#) & [mainWindow](#) ()
Reference to the [MainWindow](#), without const.
- const [MainWindow](#) & [mainWindow](#) () const
Reference to the [MainWindow](#), with const.
- void [requestDelayedStateTransition](#) ([DigitizeState](#) digitizeState)
Initiate state transition to be performed later, when DigitizeState is off the stack.
- void [requestImmediateStateTransition](#) ([CmdMediator](#) *cmdMediator, [DigitizeState](#) digitizeState)
Perform immediate state transition. Called from outside state machine.
- void [resetOnLoad](#) ([CmdMediator](#) *cmdMediator)
Resetting makes re-initializes for documents after the first.
- void [setCursor](#) ([CmdMediator](#) *cmdMediator)
Set cursor after asking state for the new cursor shape.
- void [setDragMode](#) ([QGraphicsView::DragMode](#) dragMode)
Set QGraphicsView drag mode (in m_view). Called from [DigitizeStateAbstractBase](#) subclasses.
- void [setImagesLoaded](#) ([CmdMediator](#) *cmdMediator, bool imagesLoaded)
Set the image so QGraphicsView cursor and drag mode are accessible.
- [QString](#) [state](#) () const
State name for debugging.
- void [updateAfterPointAddition](#) ()
Update the graphics attributes.
- void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &model←
[DigitizeCurve](#))
Update the digitize curve settings.
- void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)
Update the segments given the new settings.
- [QGraphicsView](#) & [view](#) ()
QGraphicsView for use by [DigitizeStateAbstractBase](#) subclasses.

5.90.1 Detailed Description

Container for all [DigitizeStateAbstractBase](#) subclasses. This functions as the context class in a standard state machine implementation.

Definition at line 25 of file DigitizeStateContext.h.

The documentation for this class was generated from the following files:

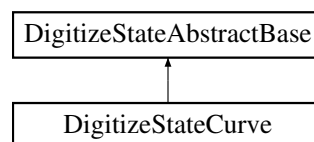
- DigitizeState/DigitizeStateContext.h
- DigitizeState/DigitizeStateContext.cpp

5.91 DigitizeStateCurve Class Reference

Digitizing state for creating [Curve](#) Points, one at a time.

```
#include <DigitizeStateCurve.h>
```

Inheritance diagram for DigitizeStateCurve:



Public Member Functions

- [DigitizeStateCurve](#) ([DigitizeStateContext](#) &context)
Single constructor.
- virtual QString [activeCurve](#) () const
Name of the active [Curve](#). This can include `AXIS_CURVE_NAME`.
- virtual void [begin](#) ([CmdMediator](#) *cmdMediator, DigitizeState previousState)
Method that is called at the exact moment a state is entered.
- virtual QCursor [cursor](#) ([CmdMediator](#) *cmdMediator) const
Returns the state-specific cursor shape.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const QString &pointIdentifier)
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const QStringList &pointIdentifiers)
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)
Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.
- virtual void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)
Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)
Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point](#) Match mode is worthwhile.

- virtual void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse release that was intercepted earlier.
- virtual [QString](#) [state](#) () const
State name for debugging.
- virtual void [updateAfterPointAddition](#) ()
Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurve)
Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)
Update the segments given the new settings.

Additional Inherited Members

5.91.1 Detailed Description

Digitizing state for creating [Curve](#) Points, one at a time.

Definition at line 13 of file [DigitizeStateCurve.h](#).

5.91.2 Member Function Documentation

5.91.2.1 void [DigitizeStateCurve::begin](#) ([CmdMediator](#) * cmdMediator, [DigitizeState](#) previousState) [virtual]

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The previousState value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implements [DigitizeStateAbstractBase](#).

Definition at line 33 of file [DigitizeStateCurve.cpp](#).

The documentation for this class was generated from the following files:

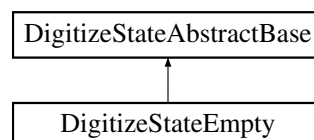
- [DigitizeState/DigitizeStateCurve.h](#)
- [DigitizeState/DigitizeStateCurve.cpp](#)

5.92 DigitizeStateEmpty Class Reference

Digitizing state before a [Document](#) has been created. In this state, the cursor is [Qt::ArrowCursor](#).

```
#include <DigitizeStateEmpty.h>
```

Inheritance diagram for [DigitizeStateEmpty](#):



Public Member Functions

- [DigitizeStateEmpty](#) ([DigitizeStateContext](#) &context)
Single constructor.
- virtual QString [activeCurve](#) () const
Name of the active [Curve](#). This can include `AXIS_CURVE_NAME`.
- virtual void [begin](#) ([CmdMediator](#) *cmdMediator, DigitizeState previousState)
Method that is called at the exact moment a state is entered.
- virtual QCursor [cursor](#) ([CmdMediator](#) *cmdMediator) const
Returns the state-specific cursor shape.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)
Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.
- virtual void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const QString &pointIdentifier)
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const QStringList &pointIdentifiers)
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)
Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)
Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point Match](#) mode is worthwhile.
- virtual void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)
Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)
Handle a mouse release that was intercepted earlier.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateAfterPointAddition](#) ()
Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurve)
Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)
Update the segments given the new settings.

Additional Inherited Members

5.92.1 Detailed Description

Digitizing state before a [Document](#) has been created. In this state, the cursor is Qt::ArrowCursor.

Definition at line 13 of file [DigitizeStateEmpty.h](#).

5.92.2 Member Function Documentation

5.92.2.1 void DigitizeStateEmpty::begin (CmdMediator * cmdMediator, DigitizeState previousState) [virtual]

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The previousState value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implements [DigitizeStateAbstractBase](#).

Definition at line 27 of file DigitizeStateEmpty.cpp.

The documentation for this class was generated from the following files:

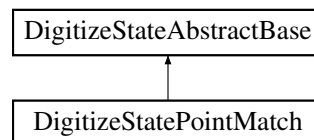
- DigitizeState/DigitizeStateEmpty.h
- DigitizeState/DigitizeStateEmpty.cpp

5.93 DigitizeStatePointMatch Class Reference

Digitizing state for matching [Curve](#) Points, one at a time.

```
#include <DigitizeStatePointMatch.h>
```

Inheritance diagram for DigitizeStatePointMatch:



Public Member Functions

- [DigitizeStatePointMatch](#) ([DigitizeStateContext](#) &context)
Single constructor.
- virtual QString [activeCurve](#) () const
Name of the active [Curve](#). This can include AXIS_CURVE_NAME.
- virtual void [begin](#) ([CmdMediator](#) *cmdMediator, DigitizeState previousState)
Method that is called at the exact moment a state is entered.
- virtual QCursor [cursor](#) ([CmdMediator](#) *cmdMediator) const
Returns the state-specific cursor shape.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const QString &pointIdentifier)
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const QStringList &pointIdentifiers)
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)

- Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.*

 - virtual void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)

Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)

Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point](#) Match mode is worthwhile.
- virtual void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)

Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, QPointF posScreen)

Handle a mouse release that was intercepted earlier.
- virtual QString [state](#) () const

State name for debugging.
- virtual void [updateAfterPointAddition](#) ()

Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurve)

Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)

Update the segments given the new settings.

Additional Inherited Members

5.93.1 Detailed Description

Digitizing state for matching [Curve](#) Points, one at a time.

Definition at line 21 of file DigitizeStatePointMatch.h.

5.93.2 Member Function Documentation

5.93.2.1 void [DigitizeStatePointMatch::begin](#) ([CmdMediator](#) * cmdMediator, [DigitizeState](#) previousState) [virtual]

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The previousState value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implements [DigitizeStateAbstractBase](#).

Definition at line 50 of file DigitizeStatePointMatch.cpp.

The documentation for this class was generated from the following files:

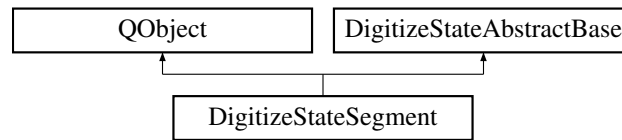
- DigitizeState/DigitizeStatePointMatch.h
- DigitizeState/DigitizeStatePointMatch.cpp

5.94 DigitizeStateSegment Class Reference

Digitizing state for creating multiple Points along a highlighted segment.

```
#include <DigitizeStateSegment.h>
```

Inheritance diagram for DigitizeStateSegment:



Public Slots

- void [slotMouseClickedOnSegment](#) (QPointF)
Receive signal from [Segment](#) that has been clicked on. The [CmdMediator](#) from the begin method will be used.

Public Member Functions

- [DigitizeStateSegment](#) (DigitizeStateContext &context)
Single constructor.
- virtual QString [activeCurve](#) () const
Name of the active [Curve](#). This can include `AXIS_CURVE_NAME`.
- virtual void [begin](#) (CmdMediator *cmdMediator, DigitizeState previousState)
Method that is called at the exact moment a state is entered.
- virtual QCursor [cursor](#) (CmdMediator *cmdMediator) const
Returns the state-specific cursor shape.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [handleContextMenuEventAxis](#) (CmdMediator *cmdMediator, const QString &pointIdentifier)
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) (CmdMediator *cmdMediator, const QStringList &pointIdentifiers)
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleCurveChange](#) (CmdMediator *cmdMediator)
Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.
- virtual void [handleKeyPress](#) (CmdMediator *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)
Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) (CmdMediator *cmdMediator, QPointF posScreen)
Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point Match](#) mode is worthwhile.
- virtual void [handleMousePress](#) (CmdMediator *cmdMediator, QPointF posScreen)
Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) (CmdMediator *cmdMediator, QPointF posScreen)
Handle a mouse release that was intercepted earlier.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateAfterPointAddition](#) ()
Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) (CmdMediator *cmdMediator, const DocumentModelDigitizeCurve &modelDigitizeCurve)
Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const DocumentModelSegments &modelSegments)
Update the segments given the new settings.

Additional Inherited Members

5.94.1 Detailed Description

Digitizing state for creating multiple Points along a highlighted segment.

Definition at line 17 of file DigitizeStateSegment.h.

5.94.2 Member Function Documentation

5.94.2.1 `void DigitizeStateSegment::begin (CmdMediator * cmdMediator, DigitizeState previousState)` [virtual]

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The previousState value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implements [DigitizeStateAbstractBase](#).

Definition at line 34 of file DigitizeStateSegment.cpp.

The documentation for this class was generated from the following files:

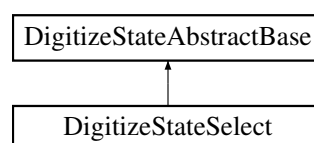
- DigitizeState/DigitizeStateSegment.h
- DigitizeState/DigitizeStateSegment.cpp

5.95 DigitizeStateSelect Class Reference

Digitizing state for selecting one or more Points in the [Document](#).

```
#include <DigitizeStateSelect.h>
```

Inheritance diagram for DigitizeStateSelect:



Public Member Functions

- [DigitizeStateSelect](#) ([DigitizeStateContext](#) &context)
Single constructor.
- virtual QString [activeCurve](#) () const
Name of the active [Curve](#). This can include `AXIS_CURVE_NAME`.
- virtual void [begin](#) ([CmdMediator](#) *cmdMediator, [DigitizeState](#) previousState)
Method that is called at the exact moment a state is entered.
- virtual [QCursor](#) [cursor](#) ([CmdMediator](#) *cmdMediator) const
Returns the state-specific cursor shape.
- virtual void [end](#) ()
Method that is called at the exact moment a state is exited. Typically called just before [begin](#) for the next state.
- virtual void [handleContextMenuEventAxis](#) ([CmdMediator](#) *cmdMediator, const QString &pointIdentifier)
Handle a right click, on an axis point, that was intercepted earlier.
- virtual void [handleContextMenuEventGraph](#) ([CmdMediator](#) *cmdMediator, const QStringList &pointIdentifiers)
Handle a right click, on a graph point, that was intercepted earlier.
- virtual void [handleCurveChange](#) ([CmdMediator](#) *cmdMediator)
Handle the selection of a new curve. At a minimum, [DigitizeStateSegment](#) will generate a new set of Segments.
- virtual void [handleKeyPress](#) ([CmdMediator](#) *cmdMediator, Qt::Key key, bool atLeastOneSelectedItem)
Handle a key press that was intercepted earlier.
- virtual void [handleMouseMove](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse move. This is part of an experiment to see if augmenting the cursor in [Point Match](#) mode is worthwhile.
- virtual void [handleMousePress](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse press that was intercepted earlier.
- virtual void [handleMouseRelease](#) ([CmdMediator](#) *cmdMediator, [QPointF](#) posScreen)
Handle a mouse release that was intercepted earlier.
- virtual QString [state](#) () const
State name for debugging.
- virtual void [updateAfterPointAddition](#) ()
Update graphics attributes after possible new points. This is useful for highlight opacity.
- virtual void [updateModelDigitizeCurve](#) ([CmdMediator](#) *cmdMediator, const [DocumentModelDigitizeCurve](#) &modelDigitizeCurve)
Update the digitize curve settings.
- virtual void [updateModelSegments](#) (const [DocumentModelSegments](#) &modelSegments)
Update the segments given the new settings.

Additional Inherited Members

5.95.1 Detailed Description

Digitizing state for selecting one or more Points in the [Document](#).

Originally this class set the cursor for each [QGraphicsItem](#) at the beginning of the state, but that triggered Qt bug 4190 which has the description 'If you have set the cursor for some [QGraphicsItems](#) you can no longer change the cursor for the view in for example a [mouseReleaseEvent](#)'. In turn, that lead to Engauge issue #155. Unfortunately, this means the user no longer has need feedback that suggests the user can do something with the [QGraphicsItems](#).

Definition at line 19 of file [DigitizeStateSelect.h](#).

5.95.2 Member Function Documentation

5.95.2.1 void DigitizeStateSelect::begin (CmdMediator * cmdMediator, DigitizeState previousState) [virtual]

Method that is called at the exact moment a state is entered.

Typically called just after end for the previous state. The previousState value is used by [DigitizeStateColorPicker](#) to return to the previous state

Implements [DigitizeStateAbstractBase](#).

Definition at line 65 of file DigitizeStateSelect.cpp.

The documentation for this class was generated from the following files:

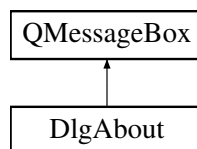
- DigitizeState/DigitizeStateSelect.h
- DigitizeState/DigitizeStateSelect.cpp

5.96 DlgAbout Class Reference

About Engauge dialog. This provides a hidden shortcut for triggering ENGAGE_ASSERT.

```
#include <DlgAbout.h>
```

Inheritance diagram for DlgAbout:



Public Member Functions

- [DlgAbout](#) (MainWindow &mainWindow)
Single constructor.

5.96.1 Detailed Description

About Engauge dialog. This provides a hidden shortcut for triggering ENGAGE_ASSERT.

Definition at line 15 of file DlgAbout.h.

The documentation for this class was generated from the following files:

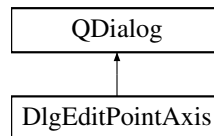
- Dlg/DlgAbout.h
- Dlg/DlgAbout.cpp

5.97 DlgEditPointAxis Class Reference

Dialog box for editing the information of one axis point.

```
#include <DlgEditPointAxis.h>
```

Inheritance diagram for DlgEditPointAxis:



Public Member Functions

- [DlgEditPointAxis](#) ([MainWindow](#) &mainWindow, const [DocumentModelCoords](#) &modelCoords, const [Main<WindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, [DocumentAxesPointsRequired](#) documentAxesPointsRequired, bool isXOnly=false, const double *xInitialValue=0, const double *yInitial<Value=0)
Constructor for existing point which already has graph coordinates (which may be changed using this dialog).
- [QPointF](#) [posGraph](#) (bool &isXOnly) const
Return the graph coordinates position specified by the user. Only applies if dialog was accepted.

5.97.1 Detailed Description

Dialog box for editing the information of one axis point.

Definition at line 24 of file DlgEditPointAxis.h.

5.97.2 Constructor & Destructor Documentation

5.97.2.1 [DlgEditPointAxis::DlgEditPointAxis](#) ([MainWindow](#) & *mainWindow*, const [DocumentModelCoords](#) & *modelCoords*, const [MainWindowModel](#) & *modelMainWindow*, const [Transformation](#) & *transformation*, [DocumentAxesPointsRequired](#) *documentAxesPointsRequired*, bool *isXOnly* = false, const double * *xInitialValue* = 0, const double * *yInitialValue* = 0)

Constructor for existing point which already has graph coordinates (which may be changed using this dialog).

If initial values are unspecified then the value fields will be initially empty

Definition at line 37 of file DlgEditPointAxis.cpp.

The documentation for this class was generated from the following files:

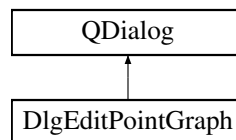
- Dlg/DlgEditPointAxis.h
- Dlg/DlgEditPointAxis.cpp

5.98 DlgEditPointGraph Class Reference

Dialog box for editing the information of one or more points.

```
#include <DlgEditPointGraph.h>
```

Inheritance diagram for DlgEditPointGraph:



Public Member Functions

- [DlgEditPointGraph](#) ([MainWindow](#) &mainWindow, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, const double *xInitialValue=0, const double *yInitialValue=0)
Constructor for existing point which already has graph coordinates (which may be changed using this dialog).
- void [posGraph](#) (bool &isX, double &x, bool &isY, double &y) const
Return one or both coordinates. Only applies if dialog was accepted.

5.98.1 Detailed Description

Dialog box for editing the information of one or more points.

Definition at line 24 of file DlgEditPointGraph.h.

5.98.2 Constructor & Destructor Documentation

5.98.2.1 `DlgEditPointGraph::DlgEditPointGraph (MainWindow & mainWindow, const DocumentModelCoords & modelCoords, const MainWindowModel & modelMainWindow, const Transformation & transformation, const double * xInitialValue = 0, const double * yInitialValue = 0)`

Constructor for existing point which already has graph coordinates (which may be changed using this dialog).

If initial values are unspecified then the value fields will be initially empty

Definition at line 27 of file DlgEditPointGraph.cpp.

The documentation for this class was generated from the following files:

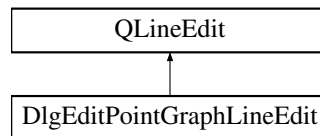
- Dlg/DlgEditPointGraph.h
- Dlg/DlgEditPointGraph.cpp

5.99 DlgEditPointGraphLineEdit Class Reference

Adds hover highlighting to QLineEdit.

```
#include <DlgEditPointGraphLineEdit.h>
```

Inheritance diagram for DlgEditPointGraphLineEdit:



Public Member Functions

- [DlgEditPointGraphLineEdit](#) (QWidget *widget=0)
Single constructor.
- virtual void [enterEvent](#) (QEvent *)
Hover entry triggers clearing of the background color so user does not think of widget as disabled and is encouraged to enter text.
- virtual void [leaveEvent](#) (QEvent *)
Hover exit triggers restoration of the background color.
- void [updateBackground](#) ()
Update background given the current state.

5.99.1 Detailed Description

Adds hover highlighting to QLineEdit.

Definition at line 15 of file DlgEditPointGraphLineEdit.h.

The documentation for this class was generated from the following files:

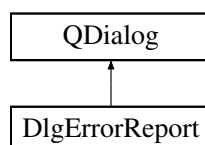
- Dlg/DlgEditPointGraphLineEdit.h
- Dlg/DlgEditPointGraphLineEdit.cpp

5.100 DlgErrorReport Class Reference

Dialog for sending error report.

```
#include <DlgErrorReport.h>
```

Inheritance diagram for DlgErrorReport:



Public Member Functions

- [DlgErrorReport](#) (const QString &xmlWithImage, QWidget *parent=0)
Single constructor. With the original data, the extra context improves debugging. With anonymization, user privacy is maintained.
- QString [xmlToUpload](#) () const
Xml to be uploaded. Includes document if user has approved.

5.100.1 Detailed Description

Dialog for sending error report.

Even if it is not sent, the information is available while this dialog is open, as a file in the executable directory

Definition at line 17 of file DlgErrorReport.h.

The documentation for this class was generated from the following files:

- Dlg/DlgErrorReport.h
- Dlg/DlgErrorReport.cpp

5.101 DlgFilterCommand Class Reference

Command pattern object for receiving new parameters in [DlgFilterWorker](#) from GUI thread.

```
#include <DlgFilterCommand.h>
```

Public Member Functions

- [DlgFilterCommand](#) (ColorFilterMode [colorFilterMode](#), double [low0To1](#), double [high0To1](#))
Initial constructor.
- [DlgFilterCommand](#) (const [DlgFilterCommand](#) &other)
Copy constructor.
- [DlgFilterCommand](#) & [operator=](#) (const [DlgFilterCommand](#) &other)
Assignment operator.
- ColorFilterMode [colorFilterMode](#) () const
Get method for filter mode.
- double [high0To1](#) () const
Get method for high value.
- double [low0To1](#) () const
Get method for low value.

5.101.1 Detailed Description

Command pattern object for receiving new parameters in [DlgFilterWorker](#) from GUI thread.

Definition at line 13 of file DlgFilterCommand.h.

The documentation for this class was generated from the following files:

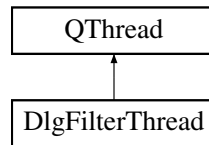
- Dlg/DlgFilterCommand.h
- Dlg/DlgFilterCommand.cpp

5.102 DlgFilterThread Class Reference

Class for processing new filter settings. This is based on <http://blog.debao.me/2013/08/how-to-use-qthread-in->

```
#include <DlgFilterThread.h>
```

Inheritance diagram for DlgFilterThread:



Signals

- void [signalTransferPiece](#) (int xLeft, QImage image)
Send a processed vertical piece of the original pixmap. The destination is between xLeft and xLeft+pixmap.width()

Public Member Functions

- [DlgFilterThread](#) (const QPixmap &pixmapOriginal, QRgb rgbBackground, [DlgSettingsColorFilter](#) &dlg↔
SettingsColorFilter)
Single constructor.
- virtual void [run](#) ()
Run this thread.

5.102.1 Detailed Description

Class for processing new filter settings. This is based on <http://blog.debao.me/2013/08/how-to-use-qthread-in->

Definition at line 18 of file DlgFilterThread.h.

The documentation for this class was generated from the following files:

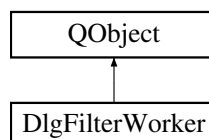
- Dlg/DlgFilterThread.h
- Dlg/DlgFilterThread.cpp

5.103 DlgFilterWorker Class Reference

Class for processing new filter settings. This is based on <http://blog.debao.me/2013/08/how-to-use-qworker-in->

```
#include <DlgFilterWorker.h>
```

Inheritance diagram for DlgFilterWorker:



Public Slots

- void [slotNewParameters](#) (ColorFilterMode colorFilterMode, double low, double high)
Start processing with a new set of parameters. Any ongoing processing is interrupted when m_filterMode changes.

Signals

- void [signalTransferPiece](#) (int xLeft, QImage image)
Send a processed vertical piece of the original pixmap. The destination is between xLeft and xLeft+pixmap.width()

Public Member Functions

- [DlgFilterWorker](#) (const QPixmap &pixmapOriginal, QRgb m_rgbBackground)
Single constructor.

5.103.1 Detailed Description

Class for processing new filter settings. This is based on <http://blog.debao.me/2013/08/how-to-use-qworker-in->

Definition at line 22 of file DlgFilterWorker.h.

The documentation for this class was generated from the following files:

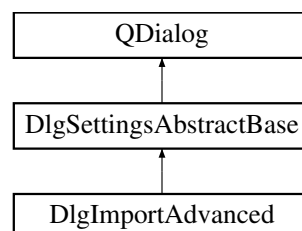
- Dlg/DlgFilterWorker.h
- Dlg/DlgFilterWorker.cpp

5.104 DlgImportAdvanced Class Reference

Dialog for setting the advanced parameters in a newly imported [Document](#).

```
#include <DlgImportAdvanced.h>
```

Inheritance diagram for DlgImportAdvanced:



Public Member Functions

- [DlgImportAdvanced](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) ([QHBoxLayout](#) *[layout](#))
Let subclass define an optional Save As Default button.
- virtual [QWidget](#) * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- [DocumentAxesPointsRequired](#) [documentAxesPointsRequired](#) () const
Number of axes points selected by user.
- virtual void [handleOk](#) ()
Process slotOk.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).
- unsigned int [numberCoordSystem](#) () const
Number of coordinate systems selected by user.

Additional Inherited Members

5.104.1 Detailed Description

Dialog for setting the advanced parameters in a newly imported [Document](#).

Definition at line 19 of file [DlgImportAdvanced.h](#).

The documentation for this class was generated from the following files:

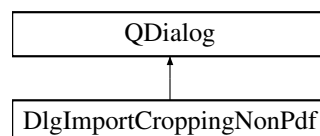
- [Dlg/DlgImportAdvanced.h](#)
- [Dlg/DlgImportAdvanced.cpp](#)

5.105 DlgImportCroppingNonPdf Class Reference

Dialog for selecting a page and frame on that page when importing an image from a non-pdf file.

```
#include <DlgImportCroppingNonPdf.h>
```

Inheritance diagram for [DlgImportCroppingNonPdf](#):



Public Member Functions

- [DlgImportCroppingNonPdf](#) (const QString &fileName)
Single constructor.
- QImage [image](#) () const
Image that was selected. Value is null if loading failed.
- virtual void [showEvent](#) (QShowEvent *event)
Do preparation before dialog is displayed.

5.105.1 Detailed Description

Dialog for selecting a page and frame on that page when importing an image from a non-pdf file.

Definition at line 24 of file DlgImportCroppingNonPdf.h.

The documentation for this class was generated from the following files:

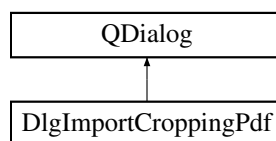
- Dlg/DlgImportCroppingNonPdf.h
- Dlg/DlgImportCroppingNonPdf.cpp

5.106 DlgImportCroppingPdf Class Reference

Dialog for selecting a page and frame on that page when importing an image from a pdf file.

```
#include <DlgImportCroppingPdf.h>
```

Inheritance diagram for DlgImportCroppingPdf:



Public Member Functions

- [DlgImportCroppingPdf](#) (const Poppler::Document &document, int resolution)
Single constructor.
- QImage [image](#) () const
Image that was selected. Value is null if loading failed.
- virtual void [showEvent](#) (QShowEvent *event)
Do preparation before dialog is displayed.

5.106.1 Detailed Description

Dialog for selecting a page and frame on that page when importing an image from a pdf file.

Definition at line 28 of file DlgImportCroppingPdf.h.

The documentation for this class was generated from the following files:

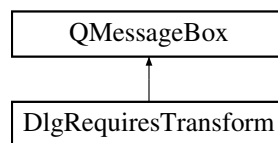
- Dlg/DlgImportCroppingPdf.h
- Dlg/DlgImportCroppingPdf.cpp

5.107 DlgRequiresTransform Class Reference

Dialog to be displayed whenever some operation or processing cannot be performed since the axis points are not defined.

```
#include <DlgRequiresTransform.h>
```

Inheritance diagram for DlgRequiresTransform:



Public Member Functions

- [DlgRequiresTransform](#) (const QString &context)
Single constructor.

5.107.1 Detailed Description

Dialog to be displayed whenever some operation or processing cannot be performed since the axis points are not defined.

Definition at line 13 of file DlgRequiresTransform.h.

The documentation for this class was generated from the following files:

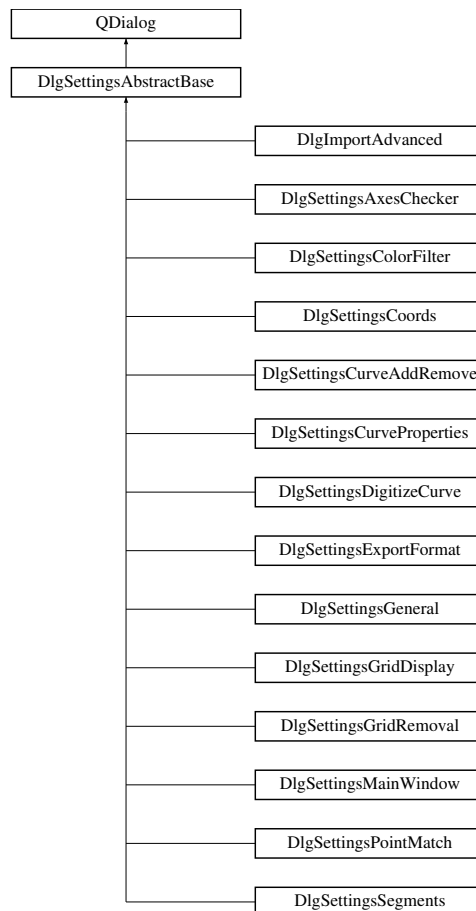
- Dlg/DlgRequiresTransform.h
- Dlg/DlgRequiresTransform.cpp

5.108 DlgSettingsAbstractBase Class Reference

Abstract base class for all Settings dialogs.

```
#include <DlgSettingsAbstractBase.h>
```

Inheritance diagram for DlgSettingsAbstractBase:



Public Member Functions

- [DlgSettingsAbstractBase](#) (const QString &title, const QString &dialogName, [MainWindow](#) &mainWindow)
Single constructor.

Protected Member Functions

- [CmdMediator](#) & [cmdMediator](#) ()
Provide access to [Document](#) information wrapped inside [CmdMediator](#).
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)=0
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()=0
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- void [enableOk](#) (bool enable)
Let leaf subclass control the Ok button.

- void `finishPanel` (QWidget *subPanel)
Add Ok and Cancel buttons to subpanel to get the whole dialog.
- virtual void `handleOk` ()=0
Process slotOk.
- virtual void `load` (CmdMediator &cmdMediator)=0
Load settings from Document.
- `MainWindow` & `mainWindow` ()
Get method for MainWindow.
- const `MainWindow` & `mainWindow` () const
Const get method for MainWindow.
- void `populateColorComboWithoutTransparent` (QComboBox &combo)
Add colors in color palette to combobox, without transparent entry at end.
- void `populateColorComboWithTransparent` (QComboBox &combo)
Add colors in color palette to combobox, with transparent entry at end.
- void `setCmdMediator` (CmdMediator &cmdMediator)
Store CmdMediator for easy access by the leaf class.
- void `setDisableOkAtStartup` (bool disableOkAtStartup)
Override the default Ok button behavior applied in showEvent.

Static Protected Attributes

- static int `MINIMUM_DIALOG_WIDTH` = 350
Dialog layout constant that guarantees every widget has sufficient room.
- static int `MINIMUM_PREVIEW_HEIGHT` = 200
Dialog layout constant that guarantees preview has sufficient room.

5.108.1 Detailed Description

Abstract base class for all Settings dialogs.

Definition at line 19 of file DlgSettingsAbstractBase.h.

5.108.2 Member Function Documentation

5.108.2.1 void DlgSettingsAbstractBase::enableOk (bool enable) [protected]

Let leaf subclass control the Ok button.

This method is separate from the subclasses' `updateControls`, rather than part of that method since `updateControls` is not aware of when it is called at startup - at which point the ok button should ALWAYS be disabled since there are not yet any changes. In other words, we call this method at startup to override the ok button state that was just set by `updateControls`

Note - if this method is called with a constant value of true from `updateControls`, one of two cases applies: 1) There are no constraints to worry about (like a required text field cannot be empty) 2) There are constraints, but they are already handled by validators and/or other constraint logic

Definition at line 51 of file DlgSettingsAbstractBase.cpp.

The documentation for this class was generated from the following files:

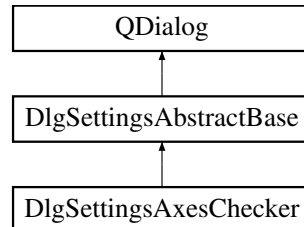
- Dlg/DlgSettingsAbstractBase.h
- Dlg/DlgSettingsAbstractBase.cpp

5.109 DlgSettingsAxesChecker Class Reference

Dialog for editing axes checker settings.

```
#include <DlgSettingsAxesChecker.h>
```

Inheritance diagram for DlgSettingsAxesChecker:



Public Member Functions

- [DlgSettingsAxesChecker](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) ([QHBoxLayout](#) *[layout](#))
Let subclass define an optional Save As Default button.
- virtual [QWidget](#) * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.109.1 Detailed Description

Dialog for editing axes checker settings.

Definition at line 24 of file `DlgSettingsAxesChecker.h`.

The documentation for this class was generated from the following files:

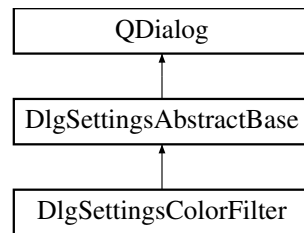
- `Dlg/DlgSettingsAxesChecker.h`
- `Dlg/DlgSettingsAxesChecker.cpp`

5.110 DlgSettingsColorFilter Class Reference

Dialog for editing filtering settings.

```
#include <DlgSettingsColorFilter.h>
```

Inheritance diagram for DlgSettingsColorFilter:



Public Slots

- void [slotTransferPiece](#) (int xLeft, QImage image)
Receive processed piece of preview image, to be inserted at xLeft to xLeft+ pixmap.width().

Signals

- void [signalApplyFilter](#) (ColorFilterMode colorFilterMode, double low, double high)
Send filter parameters to [DlgFilterThread](#) and [DlgFilterWorker](#) for processing.

Public Member Functions

- [DlgSettingsColorFilter](#) (MainWindow &mainWindow)
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) (CmdMediator &cmdMediator)
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.110.1 Detailed Description

Dialog for editing filtering settings.

Definition at line 29 of file DlgSettingsColorFilter.h.

The documentation for this class was generated from the following files:

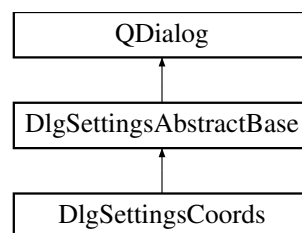
- Dlg/DlgSettingsColorFilter.h
- Dlg/DlgSettingsColorFilter.cpp

5.111 DlgSettingsCoords Class Reference

Dialog for editing coordinates settings.

```
#include <DlgSettingsCoords.h>
```

Inheritance diagram for DlgSettingsCoords:



Public Member Functions

- [DlgSettingsCoords](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.111.1 Detailed Description

Dialog for editing coordinates settings.

Definition at line 27 of file DlgSettingsCoords.h.

The documentation for this class was generated from the following files:

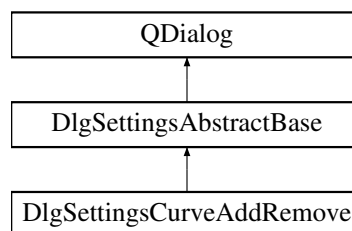
- Dlg/DlgSettingsCoords.h
- Dlg/DlgSettingsCoords.cpp

5.112 DlgSettingsCurveAddRemove Class Reference

Dialog for editing curve names settings.

```
#include <DlgSettingsCurveAddRemove.h>
```

Inheritance diagram for DlgSettingsCurveAddRemove:



Public Member Functions

- [DlgSettingsCurveAddRemove](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.112.1 Detailed Description

Dialog for editing curve names settings.

The debug macro `DLG_SETTINGS_DEBUG` can be temporarily set to see the hidden columns

Definition at line 23 of file `DlgSettingsCurveAddRemove.h`.

The documentation for this class was generated from the following files:

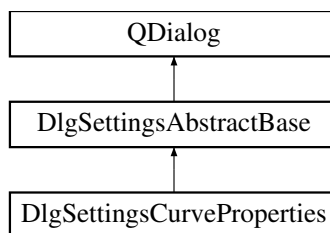
- `Dlg/DlgSettingsCurveAddRemove.h`
- `Dlg/DlgSettingsCurveAddRemove.cpp`

5.113 DlgSettingsCurveProperties Class Reference

Dialog for editing curve properties settings.

```
#include <DlgSettingsCurveProperties.h>
```

Inheritance diagram for `DlgSettingsCurveProperties`:



Public Member Functions

- [DlgSettingsCurveProperties \(MainWindow &mainWindow\)](#)
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) (CmdMediator &cmdMediator)
Load settings from [Document](#).
- void [setCurveName](#) (const QString &curveName)
Load information for the specified curve name. When called externally, the load method must have been called first.

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.113.1 Detailed Description

Dialog for editing curve properties settings.

Definition at line 23 of file DlgSettingsCurveProperties.h.

The documentation for this class was generated from the following files:

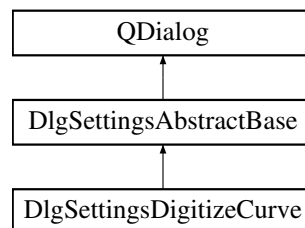
- Dlg/DlgSettingsCurveProperties.h
- Dlg/DlgSettingsCurveProperties.cpp

5.114 DlgSettingsDigitizeCurve Class Reference

Dialog for editing [DigitizeStateCurve](#) settings.

```
#include <DlgSettingsDigitizeCurve.h>
```

Inheritance diagram for DlgSettingsDigitizeCurve:



Public Member Functions

- [DlgSettingsDigitizeCurve](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.114.1 Detailed Description

Dialog for editing [DigitizeStateCurve](#) settings.

The preview window would should the selected cursor in the center, but there is no way to access the image of Qt::CrossCursor (QCursor::pixmap only works for custom cursors that were defined by a QPixmap)

Definition at line 26 of file DlgSettingsDigitizeCurve.h.

The documentation for this class was generated from the following files:

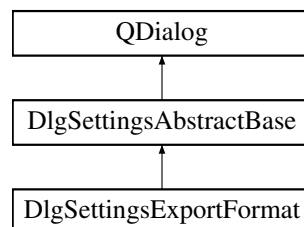
- Dlg/DlgSettingsDigitizeCurve.h
- Dlg/DlgSettingsDigitizeCurve.cpp

5.115 DlgSettingsExportFormat Class Reference

Dialog for editing exporting settings.

```
#include <DlgSettingsExportFormat.h>
```

Inheritance diagram for DlgSettingsExportFormat:



Public Member Functions

- [DlgSettingsExportFormat](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *[layout](#))
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.115.1 Detailed Description

Dialog for editing exporting settings.

Definition at line 28 of file DlgSettingsExportFormat.h.

The documentation for this class was generated from the following files:

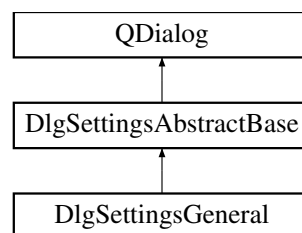
- Dlg/DlgSettingsExportFormat.h
- Dlg/DlgSettingsExportFormat.cpp

5.116 DlgSettingsGeneral Class Reference

Dialog for editing general settings.

```
#include <DlgSettingsGeneral.h>
```

Inheritance diagram for DlgSettingsGeneral:



Public Member Functions

- [DlgSettingsGeneral](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.116.1 Detailed Description

Dialog for editing general settings.

Definition at line 18 of file DlgSettingsGeneral.h.

The documentation for this class was generated from the following files:

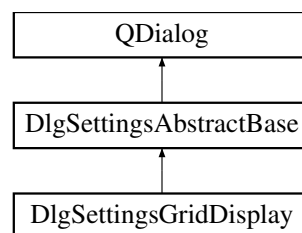
- Dlg/DlgSettingsGeneral.h
- Dlg/DlgSettingsGeneral.cpp

5.117 DlgSettingsGridDisplay Class Reference

Dialog for editing grid display settings.

```
#include <DlgSettingsGridDisplay.h>
```

Inheritance diagram for DlgSettingsGridDisplay:



Public Member Functions

- [DlgSettingsGridDisplay](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.117.1 Detailed Description

Dialog for editing grid display settings.

Definition at line 25 of file DlgSettingsGridDisplay.h.

The documentation for this class was generated from the following files:

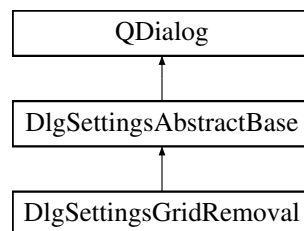
- Dlg/DlgSettingsGridDisplay.h
- Dlg/DlgSettingsGridDisplay.cpp

5.118 DlgSettingsGridRemoval Class Reference

Dialog for editing grid removal settings.

```
#include <DlgSettingsGridRemoval.h>
```

Inheritance diagram for DlgSettingsGridRemoval:



Public Member Functions

- [DlgSettingsGridRemoval](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *[layout](#))
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.118.1 Detailed Description

Dialog for editing grid removal settings.

Definition at line 23 of file DlgSettingsGridRemoval.h.

The documentation for this class was generated from the following files:

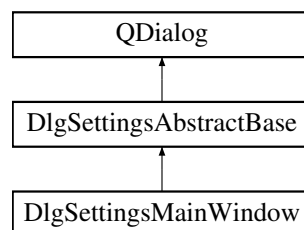
- Dlg/DlgSettingsGridRemoval.h
- Dlg/DlgSettingsGridRemoval.cpp

5.119 DlgSettingsMainWindow Class Reference

Dialog for editing main window settings, which are entirely independent of all documents.

```
#include <DlgSettingsMainWindow.h>
```

Inheritance diagram for DlgSettingsMainWindow:



Public Member Functions

- [DlgSettingsMainWindow](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *[layout](#))
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).
- void [loadMainWindowModel](#) ([CmdMediator](#) &[cmdMediator](#), const [MainWindowModel](#) &[modelMainWindow](#))
Replaced load method since the main window settings are independent of document, unlike other DlgSettings classes.*

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.119.1 Detailed Description

Dialog for editing main window settings, which are entirely independent of all documents.

Definition at line 22 of file DlgSettingsMainWindow.h.

The documentation for this class was generated from the following files:

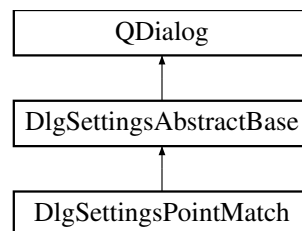
- Dlg/DlgSettingsMainWindow.h
- Dlg/DlgSettingsMainWindow.cpp

5.120 DlgSettingsPointMatch Class Reference

Dialog for editing point match settings, for [DigitizeStatePointMatch](#).

```
#include <DlgSettingsPointMatch.h>
```

Inheritance diagram for DlgSettingsPointMatch:



Public Member Functions

- [DlgSettingsPointMatch](#) ([MainWindow](#) &mainWindow)
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &cmdMediator)
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.120.1 Detailed Description

Dialog for editing point match settings, for [DigitizeStatePointMatch](#).

Definition at line 24 of file DlgSettingsPointMatch.h.

The documentation for this class was generated from the following files:

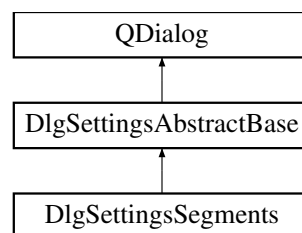
- Dlg/DlgSettingsPointMatch.h
- Dlg/DlgSettingsPointMatch.cpp

5.121 DlgSettingsSegments Class Reference

Dialog for editing Segments settings, for [DigitizeStateSegment](#).

```
#include <DlgSettingsSegments.h>
```

Inheritance diagram for DlgSettingsSegments:



Public Member Functions

- [DlgSettingsSegments](#) ([MainWindow](#) &[mainWindow](#))
Single constructor.
- virtual void [createOptionalSaveDefault](#) (QHBoxLayout *layout)
Let subclass define an optional Save As Default button.
- virtual QWidget * [createSubPanel](#) ()
Create dialog-specific panel to which base class will add Ok and Cancel buttons.
- virtual void [load](#) ([CmdMediator](#) &[cmdMediator](#))
Load settings from [Document](#).

Protected Member Functions

- virtual void [handleOk](#) ()
Process slotOk.

Additional Inherited Members

5.121.1 Detailed Description

Dialog for editing Segments settings, for [DigitizeStateSegment](#).

Definition at line 27 of file DlgSettingsSegments.h.

The documentation for this class was generated from the following files:

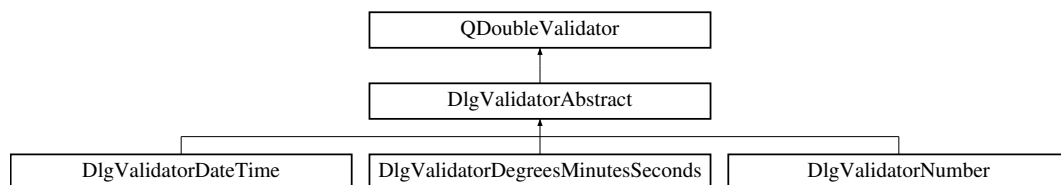
- Dlg/DlgSettingsSegments.h
- Dlg/DlgSettingsSegments.cpp

5.122 DlgValidatorAbstract Class Reference

Abstract validator for all numeric formats.

```
#include <DlgValidatorAbstract.h>
```

Inheritance diagram for DlgValidatorAbstract:



Public Member Functions

- [DlgValidatorAbstract](#) (QObject *parent=0)
Single constructor.
- virtual QValidator::State [validate](#) (QString &input, int &pos) const =0
Validate according to the numeric format specific to the leaf class.

5.122.1 Detailed Description

Abstract validator for all numeric formats.

Definition at line 14 of file DlgValidatorAbstract.h.

The documentation for this class was generated from the following files:

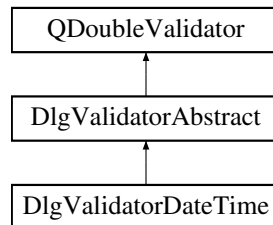
- Dlg/DlgValidatorAbstract.h
- Dlg/DlgValidatorAbstract.cpp

5.123 DlgValidatorDateTime Class Reference

Validator for numeric value expressed as date and/or time.

```
#include <DlgValidatorDateTime.h>
```

Inheritance diagram for DlgValidatorDateTime:



Public Member Functions

- [DlgValidatorDateTime](#) (CoordScale coordScale, CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, QObject *parent=0)
Single constructor.
- virtual QValidator::State [validate](#) (QString &input, int &pos) const
Validate according to the numeric format specific to the leaf class.

5.123.1 Detailed Description

Validator for numeric value expressed as date and/or time.

Definition at line 16 of file DlgValidatorDateTime.h.

The documentation for this class was generated from the following files:

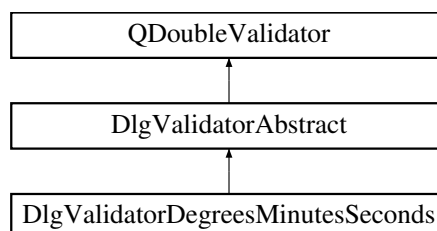
- Dlg/DlgValidatorDateTime.h
- Dlg/DlgValidatorDateTime.cpp

5.124 DlgValidatorDegreesMinutesSeconds Class Reference

Validator for angles in real degrees, integer degrees and real minutes, or integer degrees with integer minutes with real seconds.

```
#include <DlgValidatorDegreesMinutesSeconds.h>
```

Inheritance diagram for DlgValidatorDegreesMinutesSeconds:



Public Member Functions

- [DlgValidatorDegreesMinutesSeconds](#) (CoordScale coordScale, QObject *parent=0)
Single constructor.
- virtual QValidator::State [validate](#) (QString &input, int &pos) const
Validate according to the numeric format specific to the leaf class.

5.124.1 Detailed Description

Validator for angles in real degrees, integer degrees and real minutes, or integer degrees with integer minutes with real seconds.

Definition at line 17 of file DlgValidatorDegreesMinutesSeconds.h.

The documentation for this class was generated from the following files:

- Dlg/DlgValidatorDegreesMinutesSeconds.h
- Dlg/DlgValidatorDegreesMinutesSeconds.cpp

5.125 DlgValidatorFactory Class Reference

Validator factory.

```
#include <DlgValidatorFactory.h>
```

Public Member Functions

- [DlgValidatorFactory](#) ()
Single constructor.
- [DlgValidatorAbstract](#) * [createCartesianOrPolarWithNonPolarPolar](#) (CoordScale coordScale, bool isCartesian, CoordUnitsNonPolarTheta coordUnitsCartesian, CoordUnitsNonPolarTheta coordUnitsPolar, CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, const QLocale &locale) const
Factory method for generating validators for either cartesian or polar case, when polar format is specified by CoordUnitsNonPolarTheta.
- [DlgValidatorAbstract](#) * [createCartesianOrPolarWithPolarPolar](#) (CoordScale coordScale, bool isCartesian, CoordUnitsNonPolarTheta coordUnitsCartesian, CoordUnitsPolarTheta coordUnitsPolar, CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, const QLocale &locale) const
Factory method for generating validators for either cartesian or polar case, when polar format is specified by CoordUnitsPolarTheta.
- [DlgValidatorAbstract](#) * [createWithNonPolar](#) (CoordScale coordScale, CoordUnitsNonPolarTheta coordUnits, CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, const QLocale &locale) const
Factory method for generating validators when cartesian/polar case handling is handled externally, and format is specified by CoordUnitsNonPolarTheta.
- [DlgValidatorAbstract](#) * [createWithPolar](#) (CoordScale coordScale, CoordUnitsPolarTheta coordUnits, const QLocale &locale) const
Factory method for generating validators when cartesian/polar case handling is handled externally, and format is specified by CoordUnitsNonPolarTheta.

5.125.1 Detailed Description

Validator factory.

Definition at line 18 of file DlgValidatorFactory.h.

The documentation for this class was generated from the following files:

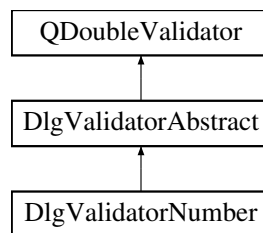
- Dlg/DlgValidatorFactory.h
- Dlg/DlgValidatorFactory.cpp

5.126 DlgValidatorNumber Class Reference

Validator for generic (=simple) numbers.

```
#include <DlgValidatorNumber.h>
```

Inheritance diagram for DlgValidatorNumber:



Public Member Functions

- [DlgValidatorNumber](#) (CoordScale coordScale, const QLocale &locale, QObject *parent=0)
Single constructor.
- virtual QValidator::State [validate](#) (QString &input, int &pos) const
Apply the standard validation with 0 as the exclusive minimum. Call setCoordScale just before calling this method.

5.126.1 Detailed Description

Validator for generic (=simple) numbers.

Definition at line 17 of file DlgValidatorNumber.h.

The documentation for this class was generated from the following files:

- Dlg/DlgValidatorNumber.h
- Dlg/DlgValidatorNumber.cpp

5.127 Document Class Reference

Storage of one imported image and the data attached to that image.

```
#include <Document.h>
```

Public Member Functions

- [Document](#) (const QImage &image)
Constructor for imported images and dragged images. Only one coordinate system is create - others are added later externally.
- [Document](#) (const QString &fileName)
Constructor for opened Documents, and error report files. The specified file is opened and read.
- void [addCoordSystems](#) (unsigned int numberCoordSystemToAdd)
Add some number (0 or more) of additional coordinate systems.
- void [addGraphCurveAtEnd](#) (const QString &curveName)
Add new graph curve to the list of existing graph curves.
- void [addPointAxisWithGeneratedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, QString &identifier, double ordinal, bool isXOnly)
Add a single axis point with a generated point identifier.
- void [addPointAxisWithSpecifiedIdentifier](#) (const QPointF &posScreen, const QPointF &posGraph, const QString &identifier, double ordinal, bool isXOnly)
Add a single axis point with the specified point identifier.
- void [addPointGraphWithGeneratedIdentifier](#) (const QString &curveName, const QPointF &posScreen, QString &generatedIdentifier, double ordinal)
Add a single graph point with a generated point identifier.
- void [addPointGraphWithSpecifiedIdentifier](#) (const QString &curveName, const QPointF &posScreen, const QString &identifier, double ordinal)
Add a single graph point with the specified point identifier. Note that [PointStyle](#) is not applied to the point within the [Document](#).
- void [addPointsInCurvesGraphs](#) ([CurvesGraphs](#) &curvesGraphs)
Add all points identified in the specified [CurvesGraphs](#). See also [removePointsInCurvesGraphs](#).
- void [checkAddPointAxis](#) (const QPointF &posScreen, const QPointF &posGraph, bool &isError, QString &errorMessage, bool isXOnly)
Check before calling [addPointAxis](#). Also returns the next available ordinal number (to prevent clashes)
- void [checkEditPointAxis](#) (const QString &pointIdentifier, const QPointF &posScreen, const QPointF &posGraph, bool &isError, QString &errorMessage)
Check before calling [editPointAxis](#).
- const [CoordSystem](#) & [coordSystem](#) () const
Currently active [CoordSystem](#).
- unsigned int [coordSystemCount](#) () const
Number of [CoordSystem](#).
- CoordSystemIndex [coordSystemIndex](#) () const
Index of current active [CoordSystem](#).
- const [Curve](#) & [curveAxes](#) () const
Get method for axis curve.
- const [Curve](#) * [curveForCurveName](#) (const QString &curveName) const
See [CurvesGraphs::curveForCurveNames](#), although this also works for `AXIS_CURVE_NAME`.
- const [CurvesGraphs](#) & [curvesGraphs](#) () const
Make all Curves available, read only, for [CmdAbstract](#) classes only.
- QStringList [curvesGraphsNames](#) () const

- See [CurvesGraphs::curvesGraphsNames](#).
- int [curvesGraphsNumPoints](#) (const QString &curveName) const
 See [CurvesGraphs::curvesGraphsNumPoints](#).
- DocumentAxesPointsRequired [documentAxesPointsRequired](#) () const
 Get method for [DocumentAxesPointsRequired](#).
- void [editPointAxis](#) (const QPointF &posGraph, const QString &identifier)
 Edit the graph coordinates of a single axis point. Call this after [checkAddPointAxis](#) to guarantee success in this call.
- void [editPointGraph](#) (bool isX, bool isY, double x, double y, const QStringList &identifiers, const [Transformation](#) &transformation)
 Edit the graph coordinates of one or more graph points.
- void [initializeGridDisplay](#) (const [Transformation](#) &transformation)
 Initialize grid display. This is called immediately after the transformation has been defined for the first time.
- bool [isXOnly](#) (const QString &pointIdentifier) const
 See [Curve::isXOnly](#).
- void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [Callback↔SearchReturn](#) > &ftorWithCallback)
 See [Curve::iterateThroughCurvePoints](#), for the axes curve.
- void [iterateThroughCurvePointsAxes](#) (const Functor2wRet< const QString &, const [Point](#) &, [Callback↔SearchReturn](#) > &ftorWithCallback) const
 See [Curve::iterateThroughCurvePoints](#), for the axes curve.
- void [iterateThroughCurveSegments](#) (const QString &curveName, const Functor2wRet< const [Point](#) &, const [Point](#) &, [CallbackSearchReturn](#) > &ftorWithCallback) const
 See [Curve::iterateThroughCurveSegments](#), for any axes or graph curve.
- void [iterateThroughCurvesPointsGraphs](#) (const Functor2wRet< const QString &, const [Point](#) &, [Callback↔SearchReturn](#) > &ftorWithCallback)
 See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
- void [iterateThroughCurvesPointsGraphs](#) (const Functor2wRet< const QString &, const [Point](#) &, [Callback↔SearchReturn](#) > &ftorWithCallback) const
 See [Curve::iterateThroughCurvePoints](#), for all the graphs curves.
- bool [loadCurvesFile](#) (const QString &curvesFile)
 Load the curve names in the specified Engauge file into the current document. This is called near the end of the import process only.
- [DocumentModelAxesChecker](#) [modelAxesChecker](#) () const
 Get method for [DocumentModelAxesChecker](#).
- [DocumentModelColorFilter](#) [modelColorFilter](#) () const
 Get method for [DocumentModelColorFilter](#).
- [DocumentModelCoords](#) [modelCoords](#) () const
 Get method for [DocumentModelCoords](#).
- [CurveStyles](#) [modelCurveStyles](#) () const
 Get method for [CurveStyles](#).
- [DocumentModelDigitizeCurve](#) [modelDigitizeCurve](#) () const
 Get method for [DocumentModelDigitizeCurve](#).
- [DocumentModelExportFormat](#) [modelExport](#) () const
 Get method for [DocumentModelExportFormat](#).
- [DocumentModelGeneral](#) [modelGeneral](#) () const
 Get method for [DocumentModelGeneral](#).
- [DocumentModelGridDisplay](#) [modelGridDisplay](#) () const
 Get method for [DocumentModelGridDisplay](#).
- [DocumentModelGridRemoval](#) [modelGridRemoval](#) () const
 Get method for [DocumentModelGridRemoval](#).
- [DocumentModelPointMatch](#) [modelPointMatch](#) () const
 Get method for [DocumentModelPointMatch](#).

- [DocumentModelSegments modelSegments \(\)](#) const
Get method for [DocumentModelSegments](#).
- void [movePoint](#) (const QString &pointIdentifier, const QPointF &deltaScreen)
See [Curve::movePoint](#).
- int [nextOrdinalForCurve](#) (const QString &curveName) const
Default next ordinal value for specified curve.
- QPixmap [pixmap \(\)](#) const
Return the image that is being digitized.
- QPointF [positionGraph](#) (const QString &pointIdentifier) const
See [Curve::positionGraph](#).
- QPointF [positionScreen](#) (const QString &pointIdentifier) const
See [Curve::positionScreen](#).
- void [print \(\)](#) const
Debugging method for printing directly from symbolic debugger.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- QString [reasonForUnsuccessfulRead \(\)](#) const
Return an informative text message explaining why startup loading failed. Applies if successfulRead returns false.
- void [removePointAxis](#) (const QString &identifier)
Perform the opposite of addPointAxis.
- void [removePointGraph](#) (const QString &identifier)
Perform the opposite of addPointGraph.
- void [removePointsInCurvesGraphs](#) ([CurvesGraphs](#) &[curvesGraphs](#))
Remove all points identified in the specified [CurvesGraphs](#). See also [addPointsInCurvesGraphs](#).
- void [saveXml](#) (QXmlStreamWriter &writer) const
Save document to xml.
- QString [selectedCurveName \(\)](#) const
Currently selected curve name. This is used to set the selected curve combobox in [MainWindow](#).
- void [setCoordSystemIndex](#) (CoordSystemIndex [coordSystemIndex](#))
Set the index of current active [CoordSystem](#).
- void [setCurveAxes](#) (const [Curve](#) &[curveAxes](#))
Let [CmdAbstract](#) classes overwrite axes [Curve](#).
- void [setCurvesGraphs](#) (const [CurvesGraphs](#) &[curvesGraphs](#))
Let [CmdAbstract](#) classes overwrite [CurvesGraphs](#).
- void [setDocumentAxesPointsRequired](#) (DocumentAxesPointsRequired [documentAxesPointsRequired](#))
Set the number of axes points required.
- void [setModelAxesChecker](#) (const [DocumentModelAxesChecker](#) &[modelAxesChecker](#))
Set method for [DocumentModelAxesChecker](#).
- void [setModelColorFilter](#) (const [DocumentModelColorFilter](#) &[modelColorFilter](#))
Set method for [DocumentModelColorFilter](#).
- void [setModelCoords](#) (const [DocumentModelCoords](#) &[modelCoords](#))
Set method for [DocumentModelCoords](#).
- void [setModelCurveStyles](#) (const [CurveStyles](#) &[modelCurveStyles](#))
Set method for [CurveStyles](#).
- void [setModelDigitizeCurve](#) (const [DocumentModelDigitizeCurve](#) &[modelDigitizeCurve](#))
Set method for [DocumentModelDigitizeCurve](#).
- void [setModelExport](#) (const [DocumentModelExportFormat](#) &[modelExport](#))
Set method for [DocumentModelExportFormat](#).
- void [setModelGeneral](#) (const [DocumentModelGeneral](#) &[modelGeneral](#))
Set method for [DocumentModelGeneral](#).
- void [setModelGridDisplay](#) (const [DocumentModelGridDisplay](#) &[modelGridDisplay](#))

- Set method for [DocumentModelGridDisplay](#).
- void [setModelGridRemoval](#) (const [DocumentModelGridRemoval](#) &[modelGridRemoval](#))
 - Set method for [DocumentModelGridRemoval](#).
- void [setModelPointMatch](#) (const [DocumentModelPointMatch](#) &[modelPointMatch](#))
 - Set method for [DocumentModelPointMatch](#).
- void [setModelSegments](#) (const [DocumentModelSegments](#) &[modelSegments](#))
 - Set method for [DocumentModelSegments](#).
- void [setPixmap](#) (const QImage &[image](#))
 - Set method for the background pixmap.
- void [setSelectedCurveName](#) (const QString &[selectedCurveName](#))
 - Save curve name that is selected for the current coordinate system, for the next time the coordinate system reappears.
- bool [successfulRead](#) () const
 - Return true if startup loading succeeded. If the loading failed then [reasonForUnsuccessfulRed](#) will explain why.
- void [updatePointOrdinals](#) (const [Transformation](#) &[transformation](#))
 - Update point ordinals after point addition/removal or dragging.

5.127.1 Detailed Description

Storage of one imported image and the data attached to that image.

Definition at line 41 of file Document.h.

5.127.2 Member Function Documentation

5.127.2.1 void Document::addCoordSystems (unsigned int *numberCoordSystemToAdd*)

Add some number (0 or more) of additional coordinate systems.

This is only safe to call during import and before any changes have been made to the [Document](#)

Definition at line 145 of file Document.cpp.

5.127.2.2 void Document::addPointAxisWithGeneratedIdentifier (const QPointF & *posScreen*, const QPointF & *posGraph*, QString & *identifier*, double *ordinal*, bool *isXOnly*)

Add a single axis point with a generated point identifier.

Call this after [checkAddPointAxis](#) to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if point has only an x coordinate

Definition at line 161 of file Document.cpp.

5.127.2.3 void Document::addPointAxisWithSpecifiedIdentifier (const QPointF & *posScreen*, const QPointF & *posGraph*, const QString & *identifier*, double *ordinal*, bool *isXOnly*)

Add a single axis point with the specified point identifier.

Call this after checkAddPointAxis to guarantee success in this call.

Parameters

<i>posScreen</i>	Screen coordinates from QGraphicsView
<i>posGraph</i>	Graph coordiantes from user
<i>identifier</i>	Identifier for new axis point
<i>ordinal</i>	Unique, for curve, ordinal number
<i>isXOnly</i>	True if point has only an x coordinate

Definition at line 176 of file Document.cpp.

5.127.2.4 void Document::setDocumentAxesPointsRequired (DocumentAxesPointsRequired *documentAxesPointsRequired*)

Set the number of axes points required.

This is called during the [Document](#) creation process, after imported images have been previewed or loaded files have had at least some xml parsing

Definition at line 864 of file Document.cpp.

5.127.2.5 void Document::updatePointOrdinals (const Transformation & *transformation*)

Update point ordinals after point addition/removal or dragging.

See GraphicsScene::updatePointOrdinalsAfterDrag. Graph coordinates of point must be up to date

Definition at line 986 of file Document.cpp.

The documentation for this class was generated from the following files:

- Document/Document.h
- Document/Document.cpp

5.128 DocumentHashGenerator Class Reference

Generates a DocumentHash value representing the state of the entire [Document](#).

```
#include <DocumentHashGenerator.h>
```

Public Member Functions

- [DocumentHashGenerator](#) ()
Single constructor.
- DocumentHash [generate](#) (const [Document](#) &document) const
Generate the hash for external storage.

5.128.1 Detailed Description

Generates a DocumentHash value representing the state of the entire [Document](#).

Definition at line 15 of file DocumentHashGenerator.h.

The documentation for this class was generated from the following files:

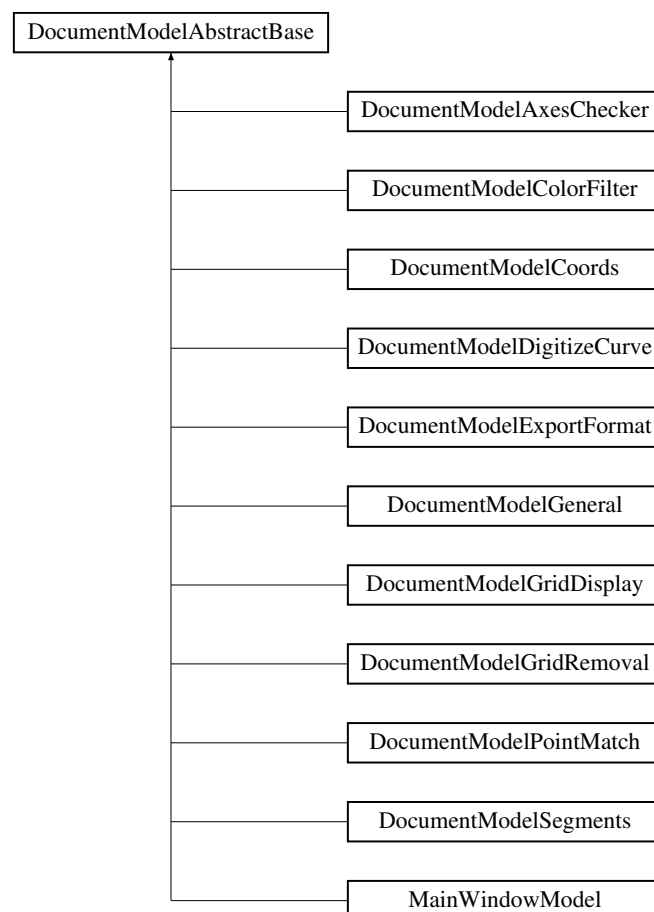
- Document/DocumentHashGenerator.h
- Document/DocumentHashGenerator.cpp

5.129 DocumentModelAbstractBase Class Reference

Abstract base class for document models. This class enforces a common interface for the leaf subclasses.

```
#include <DocumentModelAbstractBase.h>
```

Inheritance diagram for DocumentModelAbstractBase:



Public Member Functions

- [DocumentModelAbstractBase](#) ()
Single constructor.
- virtual [~DocumentModelAbstractBase](#) ()
Single destructor.

Protected Member Functions

- virtual void [loadXml](#) (QXmlStreamReader &reader)=0
Load model from serialized xml.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const =0
Save entire model as xml into stream.

5.129.1 Detailed Description

Abstract base class for document models. This class enforces a common interface for the leaf subclasses.

Definition at line 16 of file DocumentModelAbstractBase.h.

The documentation for this class was generated from the following files:

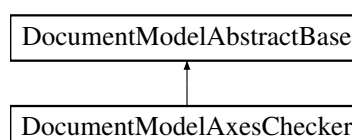
- Document/DocumentModelAbstractBase.h
- Document/DocumentModelAbstractBase.cpp

5.130 DocumentModelAxesChecker Class Reference

Model for [DlgSettingsAxesChecker](#) and [CmdSettingsAxesChecker](#).

```
#include <DocumentModelAxesChecker.h>
```

Inheritance diagram for DocumentModelAxesChecker:



Public Member Functions

- [DocumentModelAxesChecker](#) ()
Default constructor.
- [DocumentModelAxesChecker](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelAxesChecker](#) (const [DocumentModelAxesChecker](#) &other)
Copy constructor.
- [DocumentModelAxesChecker](#) & [operator=](#) (const [DocumentModelAxesChecker](#) &other)
Assignment constructor.
- CheckerMode [checkerMode](#) () const
Get method for checker lifetime mode.
- int [checkerSeconds](#) () const
Get method for checker lifetime in seconds.
- ColorPalette [lineColor](#) () const
Get method for line color.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setCheckerMode](#) (CheckerMode [checkerMode](#))
Set method for checker mode.
- void [setCheckerSeconds](#) (int seconds)
Set method for checker lifetime in seconds.
- void [setLineColor](#) (ColorPalette [lineColor](#))
Set method for line color.

Additional Inherited Members

5.130.1 Detailed Description

Model for [DlgSettingsAxesChecker](#) and [CmdSettingsAxesChecker](#).

Definition at line 18 of file [DocumentModelAxesChecker.h](#).

The documentation for this class was generated from the following files:

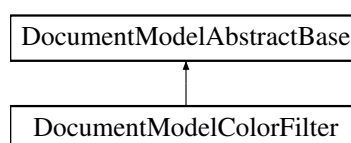
- [Document/DocumentModelAxesChecker.h](#)
- [Document/DocumentModelAxesChecker.cpp](#)

5.131 DocumentModelColorFilter Class Reference

Model for [DlgSettingsColorFilter](#) and [CmdSettingsColorFilter](#).

```
#include <DocumentModelColorFilter.h>
```

Inheritance diagram for [DocumentModelColorFilter](#):



Public Member Functions

- [DocumentModelColorFilter](#) ()
Default constructor.
- [DocumentModelColorFilter](#) (const [DocumentModelColorFilter](#) &other)
Copy constructor.
- [DocumentModelColorFilter](#) (const [CoordSystem](#) &coordSystem)
Initial constructor from [CoordSystem](#).
- [DocumentModelColorFilter](#) & operator= (const [DocumentModelColorFilter](#) &other)
Assignment constructor.
- ColorFilterMode [colorFilterMode](#) (const QString &curveName) const
Get method for filter mode.
- const [ColorFilterSettings](#) [colorFilterSettings](#) (const QString &curveName) const
Get method for copying one color filter. Cannot return just a reference or else there is a warning about returning reference to temporary.
- const [ColorFilterSettingsList](#) & [colorFilterSettingsList](#) () const
Get method for copying all color filters in one step.
- int [foregroundHigh](#) (const QString &curveName) const
Get method for foreground higher bound.
- int [foregroundLow](#) (const QString &curveName) const
Get method for foreground lower bound.
- double [high](#) (const QString &curveName) const
High value of foreground, hue, intensity, saturation or value according to current filter mode.
- int [hueHigh](#) (const QString &curveName) const
Get method for hue higher bound.
- int [hueLow](#) (const QString &curveName) const
Get method for hue lower bound.
- int [intensityHigh](#) (const QString &curveName) const
Get method for intensity higher bound.
- int [intensityLow](#) (const QString &curveName) const
Get method for intensity lower bound.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- double [low](#) (const QString &curveName) const
Low value of foreground, hue, intensity, saturation or value according to current filter mode normalized to 0 to 1.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- int [saturationHigh](#) (const QString &curveName) const
Get method for saturation higher bound.
- int [saturationLow](#) (const QString &curveName) const
Get method for saturation lower bound.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setColorFilterMode](#) (const QString &curveName, ColorFilterMode [colorFilterMode](#))
Set method for filter mode.
- void [setForegroundHigh](#) (const QString &curveName, int [foregroundHigh](#))
Set method for foreground higher bound.
- void [setForegroundLow](#) (const QString &curveName, int [foregroundLow](#))
Set method for foreground lower bound.
- void [setHigh](#) (const QString &curveName, double s0To1)
Set the high value for the current filter mode.

- void [setHueHigh](#) (const QString &curveName, int [hueHigh](#))
Set method for hue higher bound.
- void [setHueLow](#) (const QString &curveName, int [hueLow](#))
Set method for hue lower bound.
- void [setIntensityHigh](#) (const QString &curveName, int [intensityHigh](#))
Set method for intensity higher bound.
- void [setIntensityLow](#) (const QString &curveName, int [intensityLow](#))
Set method for intensity lower bound.
- void [setLow](#) (const QString &curveName, double s0To1)
Set the low value for the current filter mode.
- void [setSaturationHigh](#) (const QString &curveName, int [saturationHigh](#))
Set method for saturation high.
- void [setSaturationLow](#) (const QString &curveName, int [saturationLow](#))
Set method for saturation low.
- void [setValueHigh](#) (const QString &curveName, int [valueHigh](#))
Set method for value high.
- void [setValueLow](#) (const QString &curveName, int [valueLow](#))
Set method for value low.
- int [valueHigh](#) (const QString &curveName) const
Get method for value high.
- int [valueLow](#) (const QString &curveName) const
Get method for value low.

Additional Inherited Members

5.131.1 Detailed Description

Model for [DlgSettingsColorFilter](#) and [CmdSettingsColorFilter](#).

Definition at line 21 of file DocumentModelColorFilter.h.

5.131.2 Member Function Documentation

5.131.2.1 double DocumentModelColorFilter::high (const QString & curveName) const

High value of foreground, hue, intensity, saturation or value according to current filter mode.

normalized to 0 to 1.

Definition at line 105 of file DocumentModelColorFilter.cpp.

5.131.2.2 double DocumentModelColorFilter::low (const QString & curveName) const

Low value of foreground, hue, intensity, saturation or value according to current filter mode normalized to 0 to 1.

Definition at line 212 of file DocumentModelColorFilter.cpp.

The documentation for this class was generated from the following files:

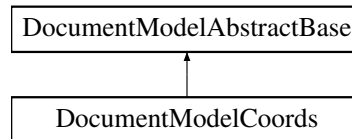
- Document/DocumentModelColorFilter.h
- Document/DocumentModelColorFilter.cpp

5.132 DocumentModelCoords Class Reference

Model for [DlgSettingsCoords](#) and [CmdSettingsCoords](#).

```
#include <DocumentModelCoords.h>
```

Inheritance diagram for DocumentModelCoords:



Public Member Functions

- [DocumentModelCoords](#) ()
Default constructor.
- [DocumentModelCoords](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelCoords](#) (const [DocumentModelCoords](#) &other)
Copy constructor.
- [DocumentModelCoords](#) & operator= (const [DocumentModelCoords](#) &other)
Assignment constructor.
- CoordScale [coordScaleXTheta](#) () const
Get method for linear/log scale on x/theta.
- CoordScale [coordScaleYRadius](#) () const
Get method for linear/log scale on y/radius.
- CoordsType [coordsType](#) () const
Get method for coordinates type.
- CoordUnitsDate [coordUnitsDate](#) () const
Get method for date format when used.
- CoordUnitsNonPolarTheta [coordUnitsRadius](#) () const
Get method for radius units.
- CoordUnitsPolarTheta [coordUnitsTheta](#) () const
Get method for theta unit.
- CoordUnitsTime [coordUnitsTime](#) () const
Get method for time format when used.
- CoordUnitsNonPolarTheta [coordUnitsX](#) () const
Get method for x units.
- CoordUnitsNonPolarTheta [coordUnitsY](#) () const
Get method for y units.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- double [originRadius](#) () const
Get method for origin radius in polar mode.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.

- void [setCoordScaleXTheta](#) (CoordScale coordScale)
Set method for linear/log scale on x/theta.
- void [setCoordScaleYRadius](#) (CoordScale coordScale)
Set method for linear/log scale on y/radius.
- void [setCoordsType](#) (CoordsType coordsType)
Set method for coordinates type.
- void [setCoordUnitsDate](#) (CoordUnitsDate coordUnits)
Set method for date units.
- void [setCoordUnitsRadius](#) (CoordUnitsNonPolarTheta coordUnits)
Set method for radius units.
- void [setCoordUnitsTheta](#) (CoordUnitsPolarTheta coordUnits)
Set method for theta units.
- void [setCoordUnitsTime](#) (CoordUnitsTime coordUnits)
Set method for time units.
- void [setCoordUnitsX](#) (CoordUnitsNonPolarTheta coordUnits)
Set method for x units.
- void [setCoordUnitsY](#) (CoordUnitsNonPolarTheta coordUnits)
Set method for y units.
- void [setOriginRadius](#) (double originRadius)
Set method for origin radius in polar mode.
- double [thetaPeriod](#) () const
Return the period of the theta value for polar coordinates, consistent with CoordThetaUnits.

Additional Inherited Members

5.132.1 Detailed Description

Model for [DlgSettingsCoords](#) and [CmdSettingsCoords](#).

Definition at line 20 of file DocumentModelCoords.h.

The documentation for this class was generated from the following files:

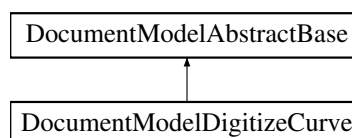
- Document/DocumentModelCoords.h
- Document/DocumentModelCoords.cpp

5.133 DocumentModelDigitizeCurve Class Reference

Model for [DlgSettingsDigitizeCurve](#) and [CmdSettingsDigitizeCurve](#).

```
#include <DocumentModelDigitizeCurve.h>
```

Inheritance diagram for DocumentModelDigitizeCurve:



Public Member Functions

- [DocumentModelDigitizeCurve](#) ()
Default constructor.
- [DocumentModelDigitizeCurve](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelDigitizeCurve](#) (const [DocumentModelDigitizeCurve](#) &other)
Copy constructor.
- [DocumentModelDigitizeCurve](#) & [operator=](#) (const [DocumentModelDigitizeCurve](#) &other)
Assignment constructor.
- int [cursorInnerRadius](#) () const
Get method for cursor inner radius.
- int [cursorLineWidth](#) () const
Get method for cursor line width.
- CursorSize [cursorSize](#) () const
Get method for cursor size.
- bool [cursorStandardCross](#) () const
Get method for cursor type.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setCursorInnerRadius](#) (int innerRadius)
Set method for cursor inner radius.
- void [setCursorLineWidth](#) (int lineWidth)
Set method for cursor line width.
- void [setCursorSize](#) (CursorSize [cursorSize](#))
Set method for cursor size.
- void [setCursorStandardCross](#) (bool [cursorStandardCross](#))
Set method for cursor type.

Additional Inherited Members

5.133.1 Detailed Description

Model for [DlgSettingsDigitizeCurve](#) and [CmdSettingsDigitizeCurve](#).

No color is involved because the documentation in QCursor suggests that not all platforms support colored cursors

Definition at line 18 of file DocumentModelDigitizeCurve.h.

The documentation for this class was generated from the following files:

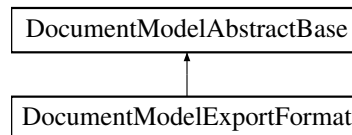
- Document/DocumentModelDigitizeCurve.h
- Document/DocumentModelDigitizeCurve.cpp

5.134 DocumentModelExportFormat Class Reference

Model for [DlgSettingsExportFormat](#) and [CmdSettingsExportFormat](#).

```
#include <DocumentModelExportFormat.h>
```

Inheritance diagram for DocumentModelExportFormat:



Public Member Functions

- [DocumentModelExportFormat](#) ()
Default constructor.
- [DocumentModelExportFormat](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelExportFormat](#) (const [DocumentModelExportFormat](#) &other)
Copy constructor.
- [DocumentModelExportFormat](#) & operator= (const [DocumentModelExportFormat](#) &other)
Assignment constructor.
- QStringList [curveNamesNotExported](#) () const
Get method for curve names not exported.
- ExportDelimiter [delimiter](#) () const
Get method for delimiter.
- ExportHeader [header](#) () const
Get method for header.
- ExportLayoutFunctions [layoutFunctions](#) () const
Get method for functions layout.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- bool [overrideCsvTsv](#) () const
Get method for csv/tsv format override.
- double [pointsIntervalFunctions](#) () const
Get method for points interval for functions.
- double [pointsIntervalRelations](#) () const
Get method for relations interval for relations.
- ExportPointsIntervalUnits [pointsIntervalUnitsFunctions](#) () const
Get method for points interval units for functions.
- ExportPointsIntervalUnits [pointsIntervalUnitsRelations](#) () const
Get method for points interval units for relations.
- ExportPointsSelectionFunctions [pointsSelectionFunctions](#) () const
Get method for point selection for functions.
- ExportPointsSelectionRelations [pointsSelectionRelations](#) () const
Get method for point selection for relations.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)

- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setCurveNamesNotExported](#) (const QStringList &[curveNamesNotExported](#))
Set method for curve names not exported.
- void [setDelimiter](#) (ExportDelimiter exportDelimiter)
Set method for delimiter.
- void [setHeader](#) (ExportHeader exportHeader)
Set method for header.
- void [setLayoutFunctions](#) (ExportLayoutFunctions exportLayoutFunctions)
Set method for functions layout.
- void [setOverrideCsvTsv](#) (bool [overrideCsvTsv](#))
Set method for csv/tsv format override.
- void [setPointsIntervalFunctions](#) (double [pointsIntervalFunctions](#))
Set method for points interval for functions.
- void [setPointsIntervalRelations](#) (double [pointsIntervalRelations](#))
Set method for relations interval for relations.
- void [setPointsIntervalUnitsFunctions](#) (ExportPointsIntervalUnits [pointsIntervalUnitsFunctions](#))
Set method for points interval units for functions.
- void [setPointsIntervalUnitsRelations](#) (ExportPointsIntervalUnits [pointsIntervalUnitsRelations](#))
Set method for points interval units for relations.
- void [setPointsSelectionFunctions](#) (ExportPointsSelectionFunctions exportPointsSelectionFunctions)
Set method for point selection for functions.
- void [setPointsSelectionRelations](#) (ExportPointsSelectionRelations exportPointsSelectionRelations)
Set method for point selection for relations.
- void [setXLabel](#) (const QString &[xLabel](#))
Set method for x label.
- QString [xLabel](#) () const
Get method for x label.

Additional Inherited Members

5.134.1 Detailed Description

Model for [DlgSettingsExportFormat](#) and [CmdSettingsExportFormat](#).

Definition at line 23 of file DocumentModelExportFormat.h.

The documentation for this class was generated from the following files:

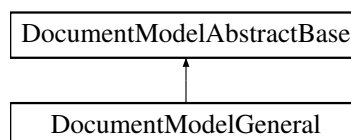
- Document/DocumentModelExportFormat.h
- Document/DocumentModelExportFormat.cpp

5.135 DocumentModelGeneral Class Reference

Model for [DlgSettingsGeneral](#) and [CmdSettingsGeneral](#).

```
#include <DocumentModelGeneral.h>
```

Inheritance diagram for DocumentModelGeneral:



Public Member Functions

- [DocumentModelGeneral](#) ()
Default constructor.
- [DocumentModelGeneral](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelGeneral](#) (const [DocumentModelGeneral](#) &other)
Copy constructor.
- [DocumentModelGeneral](#) & operator= (const [DocumentModelGeneral](#) &other)
Assignment constructor.
- int [cursorSize](#) () const
Get method for effective cursor size.
- int [extraPrecision](#) () const
Get method for extra digits of precision.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setCursorSize](#) (int [cursorSize](#))
Set method for effective cursor size.
- void [setExtraPrecision](#) (int [extraPrecision](#))
Set method for extra digits of precision.

Additional Inherited Members

5.135.1 Detailed Description

Model for [DlgSettingsGeneral](#) and [CmdSettingsGeneral](#).

Definition at line 16 of file [DocumentModelGeneral.h](#).

The documentation for this class was generated from the following files:

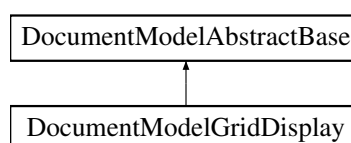
- [Document/DocumentModelGeneral.h](#)
- [Document/DocumentModelGeneral.cpp](#)

5.136 DocumentModelGridDisplay Class Reference

Model for [DlgSettingsGridDisplay](#) and [CmdSettingsGridDisplay](#).

```
#include <DocumentModelGridDisplay.h>
```

Inheritance diagram for [DocumentModelGridDisplay](#):



Public Member Functions

- [DocumentModelGridDisplay](#) ()
Default constructor.
- [DocumentModelGridDisplay](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelGridDisplay](#) (const [DocumentModelGridDisplay](#) &other)
Copy constructor.
- [DocumentModelGridDisplay](#) & operator= (const [DocumentModelGridDisplay](#) &other)
Assignment constructor.
- unsigned int [countX](#) () const
Get method for x grid line count.
- unsigned int [countY](#) () const
Get method for y grid line count.
- GridCoordDisable [disableX](#) () const
Get method for x grid line disabled variable.
- GridCoordDisable [disableY](#) () const
Get method for y grid line disabled variable.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- ColorPalette [paletteColor](#) () const
Get method for color.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setCountX](#) (unsigned int [countX](#))
Set method for x grid line count.
- void [setCountY](#) (unsigned int [countY](#))
Set method for y grid line count.
- void [setDisableX](#) (GridCoordDisable [disableX](#))
Set method for x grid line disabled variable.
- void [setDisableY](#) (GridCoordDisable [disableY](#))
Set method for y grid line disabled variable.
- void [setPaletteColor](#) (ColorPalette [paletteColor](#))
Set method for color.
- void [setStable](#) (bool [stable](#))
Set method for stable flag.
- void [setStartX](#) (double [startX](#))
Set method for x grid line lower bound (inclusive).
- void [setStartY](#) (double [yStart](#))
Set method for y grid line lower bound (inclusive).
- void [setStepX](#) (double [stepX](#))
Set method for x grid line increment.
- void [setStepY](#) (double [yStep](#))
Set method for y grid line increment.
- void [setStopX](#) (double [stopX](#))
Set method for x grid line upper bound (inclusive).
- void [setStopY](#) (double [yStop](#))
Set method for y grid line upper bound (inclusive).
- bool [stable](#) () const

- *Get method for stable flag.*
- double `startX ()` const
Get method for x grid line lower bound (inclusive).
- double `startY ()` const
Get method for y grid line lower bound (inclusive).
- double `stepX ()` const
Get method for x grid line increment.
- double `stepY ()` const
Get method for y grid line increment.
- double `stopX ()` const
Get method for x grid line upper bound (inclusive).
- double `stopY ()` const
Get method for y grid line upper bound (inclusive).

Additional Inherited Members

5.136.1 Detailed Description

Model for [DlgSettingsGridDisplay](#) and [CmdSettingsGridDisplay](#).

Definition at line 18 of file `DocumentModelGridDisplay.h`.

5.136.2 Member Function Documentation

5.136.2.1 `bool DocumentModelGridDisplay::stable ()` const

Get method for stable flag.

The flag is false to let the settings get automatically updated, until the user selects settings - at which point the stable flag is set to true

Definition at line 268 of file `DocumentModelGridDisplay.cpp`.

The documentation for this class was generated from the following files:

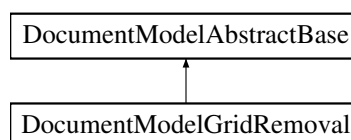
- `Document/DocumentModelGridDisplay.h`
- `Document/DocumentModelGridDisplay.cpp`

5.137 DocumentModelGridRemoval Class Reference

Model for [DlgSettingsGridRemoval](#) and [CmdSettingsGridRemoval](#). The settings are unstable until the user approves.

```
#include <DocumentModelGridRemoval.h>
```

Inheritance diagram for `DocumentModelGridRemoval`:



Public Member Functions

- [DocumentModelGridRemoval](#) ()
Default constructor.
- [DocumentModelGridRemoval](#) (double [startX](#), double [startY](#), double [stepX](#), double [stepY](#), int [countX](#), int [countY](#))
Constructor fed by [GridClassifier](#).
- [DocumentModelGridRemoval](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelGridRemoval](#) (const [DocumentModelGridRemoval](#) &other)
Copy constructor.
- [DocumentModelGridRemoval](#) & [operator=](#) (const [DocumentModelGridRemoval](#) &other)
Assignment constructor.
- double [closeDistance](#) () const
Get method for close distance.
- int [countX](#) () const
Get method for x count.
- int [countY](#) () const
Get method for y count.
- GridCoordDisable [gridCoordDisableX](#) () const
Get method for x coord parameter to disable.
- GridCoordDisable [gridCoordDisableY](#) () const
Get method for y coord parameter to disable.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- bool [removeDefinedGridLines](#) () const
Get method for removing defined grid lines.
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setCloseDistance](#) (double [closeDistance](#))
Set method for close distance.
- void [setCountX](#) (int [countX](#))
Set method for x count.
- void [setCountY](#) (int [countY](#))
Set method for y count.
- void [setGridCoordDisableX](#) (GridCoordDisable gridCoordDisable)
Set method for x coord parameter to disable.
- void [setGridCoordDisableY](#) (GridCoordDisable gridCoordDisable)
Set method for y coord parameter to disable.
- void [setRemoveDefinedGridLines](#) (bool [removeDefinedGridLines](#))
Set method for removing defined grid lines.
- void [setStable](#) ()
Set the stable flag to true. This public version has no argument since it cannot be undone.
- void [setStartX](#) (double [startX](#))
Set method for x start.
- void [setStartY](#) (double [startY](#))
Set method for y start.
- void [setStepX](#) (double [stepX](#))
Set method for x step.

- void [setStepY](#) (double [stepY](#))
Set method for y step.
- void [setStopX](#) (double [stopX](#))
Set method for x stop.
- void [setStopY](#) (double [stopY](#))
Set method for y stop.
- bool [stable](#) () const
Get method for stable flag.
- double [startX](#) () const
Get method for x start.
- double [startY](#) () const
Get method for y start.
- double [stepX](#) () const
Get method for x step.
- double [stepY](#) () const
Get method for y step.
- double [stopX](#) () const
Get method for x stop.
- double [stopY](#) () const
Get method for y stop.

Additional Inherited Members

5.137.1 Detailed Description

Model for [DlgSettingsGridRemoval](#) and [CmdSettingsGridRemoval](#). The settings are unstable until the user approves.

Definition at line 17 of file `DocumentModelGridRemoval.h`.

5.137.2 Member Function Documentation

5.137.2.1 bool DocumentModelGridRemoval::stable () const

Get method for stable flag.

The flag is false to let the settings get automatically updated, until the user selects settings - at which point the stable flag is set to true

Definition at line 320 of file `DocumentModelGridRemoval.cpp`.

The documentation for this class was generated from the following files:

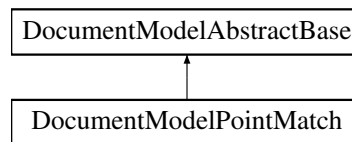
- `Document/DocumentModelGridRemoval.h`
- `Document/DocumentModelGridRemoval.cpp`

5.138 DocumentModelPointMatch Class Reference

Model for [DlgSettingsPointMatch](#) and [CmdSettingsPointMatch](#).

```
#include <DocumentModelPointMatch.h>
```

Inheritance diagram for DocumentModelPointMatch:



Public Member Functions

- [DocumentModelPointMatch](#) ()
Default constructor.
- [DocumentModelPointMatch](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelPointMatch](#) (const [DocumentModelPointMatch](#) &other)
Copy constructor.
- [DocumentModelPointMatch](#) & operator= (const [DocumentModelPointMatch](#) &other)
Assignment constructor.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- double [maxPointSize](#) () const
Get method for max point size.
- ColorPalette [paletteColorAccepted](#) () const
Get method for accepted color.
- ColorPalette [paletteColorCandidate](#) () const
Get method for candidate color.
- ColorPalette [paletteColorRejected](#) () const
Get method for rejected color.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setMaxPointSize](#) (double [maxPointSize](#))
Set method for max point size.
- void [setPaletteColorAccepted](#) (ColorPalette [paletteColorAccepted](#))
Set method for accepted color.
- void [setPaletteColorCandidate](#) (ColorPalette [paletteColorCandidate](#))
Set method for candidate color.
- void [setPaletteColorRejected](#) (ColorPalette [paletteColorRejected](#))
Set method for rejected color.

Additional Inherited Members

5.138.1 Detailed Description

Model for [DlgSettingsPointMatch](#) and [CmdSettingsPointMatch](#).

Definition at line 17 of file DocumentModelPointMatch.h.

The documentation for this class was generated from the following files:

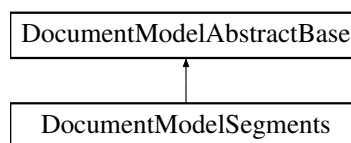
- Document/DocumentModelPointMatch.h
- Document/DocumentModelPointMatch.cpp

5.139 DocumentModelSegments Class Reference

Model for [DlgSettingsSegments](#) and [CmdSettingsSegments](#).

```
#include <DocumentModelSegments.h>
```

Inheritance diagram for DocumentModelSegments:



Public Member Functions

- [DocumentModelSegments](#) ()
Default constructor.
- [DocumentModelSegments](#) (const [Document](#) &document)
Initial constructor from [Document](#).
- [DocumentModelSegments](#) (const [DocumentModelSegments](#) &other)
Copy constructor.
- [DocumentModelSegments](#) & operator= (const [DocumentModelSegments](#) &other)
Assignment constructor.
- bool [fillCorners](#) () const
Get method for fill corners.
- ColorPalette [lineColor](#) () const
Get method for line color.
- double [lineWidth](#) () const
Get method for line width.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- double [minLength](#) () const
Get method for min length.
- double [pointSeparation](#) () const
Get method for point separation.

- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setFillCorners](#) (bool [fillCorners](#))
Set method for fill corners.
- void [setLineColor](#) (ColorPalette [lineColor](#))
Set method for line color.
- void [setLineWidth](#) (double [lineWidth](#))
Set method for line width.
- void [setMinLength](#) (double [minLength](#))
Set method for min length.
- void [setPointSeparation](#) (double [pointSeparation](#))
Set method for point separation.

Additional Inherited Members

5.139.1 Detailed Description

Model for [DlgSettingsSegments](#) and [CmdSettingsSegments](#).

Definition at line 17 of file DocumentModelSegments.h.

The documentation for this class was generated from the following files:

- Document/DocumentModelSegments.h
- Document/DocumentModelSegments.cpp

5.140 ExportAlignLinear Class Reference

Pick first simplest x value between specified min and max, for linear scaling.

```
#include <ExportAlignLinear.h>
```

Public Member Functions

- [ExportAlignLinear](#) (double xMin, double xMax)
Single constructor.
- double [firstSimplestNumber](#) () const
Result.

5.140.1 Detailed Description

Pick first simplest x value between specified min and max, for linear scaling.

A simplest value is defined here as one having the smallest number of significant digits, and is used for aligning periodic values on simple numbers. Examples:

1. 0.4 to 3.4, result is 1
2. 110 to 1100, result is 200 (not 1000 although both have same number of significant digits)
3. 112.123 to 122.456, result is 120

Definition at line 17 of file ExportAlignLinear.h.

The documentation for this class was generated from the following files:

- Export/ExportAlignLinear.h
- Export/ExportAlignLinear.cpp

5.141 ExportAlignLog Class Reference

Pick first simplest x value between specified min and max, for log scaling.

```
#include <ExportAlignLog.h>
```

Public Member Functions

- [ExportAlignLog](#) (double xMin, double xMax)
Single constructor.
- double [firstSimplestNumber](#) () const
Result.

5.141.1 Detailed Description

Pick first simplest x value between specified min and max, for log scaling.

A simplest value is defined here as one having the smallest number of significant digits when log value is taken, and is used for aligning periodic values on simple numbers. Examples:

1. 0.9 to 2, result is 1 which is 10^0
2. 1.1 to 9, result is $\sqrt{10}$ which is midway between 1 and 10 in log scale, and equal to $10^{0.5}$
3. 9.81 to 9.93, result is $10^{0.992}$ since $9.81=10^{0.99166}$ and $9.93=10^{0.9969}$

Definition at line 17 of file ExportAlignLog.h.

The documentation for this class was generated from the following files:

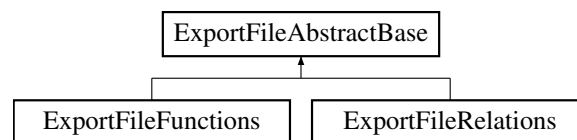
- Export/ExportAlignLog.h
- Export/ExportAlignLog.cpp

5.142 ExportFileAbstractBase Class Reference

Strategy base class for exporting to a file. This class provides common methods.

```
#include <ExportFileAbstractBase.h>
```

Inheritance diagram for ExportFileAbstractBase:



Public Member Functions

- [ExportFileAbstractBase](#) ()
Single constructor.

Protected Member Functions

- [QString](#) [curveSeparator](#) (const [QString](#) *string) const
Gnuplot requires, and other graphing tools probably prefer, blank lines between successive curves.
- [QStringList](#) [curvesToInclude](#) (const [DocumentModelExportFormat](#) &modelExportOverride, const [Document](#) &document, const [QStringList](#) &curvesGraphsNames, [CurveConnectAs](#) curveConnectAs1, [CurveConnectAs](#) curveConnectAs2) const
Identify curves to include in export. The specified [DocumentModelExportFormat](#) overrides same data in [Document](#) for previewing window.
- void [destroy2DArray](#) ([QVector](#)< [QVector](#)< [QString](#) * > > &array) const
Deallocate memory for array.
- [QString](#) [gnuplotComment](#) () const
Gnuplot comment delimiter.
- void [insertLineSeparator](#) (bool &isFirst, [ExportHeader](#) exportHeader, [QTextStream](#) &str) const
Insert line(s) between successive sets of curves.

5.142.1 Detailed Description

Strategy base class for exporting to a file. This class provides common methods.

Definition at line 24 of file [ExportFileAbstractBase.h](#).

The documentation for this class was generated from the following files:

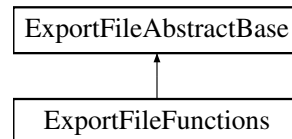
- [Export/ExportFileAbstractBase.h](#)
- [Export/ExportFileAbstractBase.cpp](#)

5.143 ExportFileFunctions Class Reference

Strategy class for exporting to a file. This strategy is external to the [Document](#) class so that class is simpler.

```
#include <ExportFileFunctions.h>
```

Inheritance diagram for ExportFileFunctions:



Public Member Functions

- [ExportFileFunctions](#) ()
Single constructor.
- void [exportToFile](#) (const [DocumentModelExportFormat](#) &modelExportOverride, const [Document](#) &document, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QTextStream &str) const
Export [Document](#) points according to the settings.

Additional Inherited Members

5.143.1 Detailed Description

Strategy class for exporting to a file. This strategy is external to the [Document](#) class so that class is simpler.

Definition at line 23 of file ExportFileFunctions.h.

5.143.2 Member Function Documentation

5.143.2.1 void ExportFileFunctions::exportToFile (const DocumentModelExportFormat & modelExportOverride, const Document & document, const MainWindowModel & modelMainWindow, const Transformation & transformation, QTextStream & str) const

Export [Document](#) points according to the settings.

The [DocumentModelExportFormat](#) inside the [Document](#) is ignored so DlgSettingsExport can supply its own [DocumentModelExportFormat](#) when previewing what would be exported.

Definition at line 117 of file ExportFileFunctions.cpp.

The documentation for this class was generated from the following files:

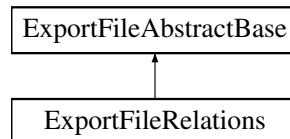
- Export/ExportFileFunctions.h
- Export/ExportFileFunctions.cpp

5.144 ExportFileRelations Class Reference

Strategy class for exporting to a file. This strategy is external to the [Document](#) class so that class is simpler.

```
#include <ExportFileRelations.h>
```

Inheritance diagram for ExportFileRelations:



Public Member Functions

- [ExportFileRelations](#) ()
Single constructor.
- void [exportToFile](#) (const [DocumentModelExportFormat](#) &modelExportOverride, const [Document](#) &document, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QTextStream &str) const
Export [Document](#) points according to the settings.

Additional Inherited Members

5.144.1 Detailed Description

Strategy class for exporting to a file. This strategy is external to the [Document](#) class so that class is simpler.

Definition at line 24 of file ExportFileRelations.h.

5.144.2 Member Function Documentation

5.144.2.1 void ExportFileRelations::exportToFile (const [DocumentModelExportFormat](#) & *modelExportOverride*, const [Document](#) & *document*, const [MainWindowModel](#) & *modelMainWindow*, const [Transformation](#) & *transformation*, QTextStream & *str*) const

Export [Document](#) points according to the settings.

The [DocumentModelExportFormat](#) inside the [Document](#) is ignored so DlgSettingsExport can supply its own [DocumentModelExportFormat](#) when previewing what would be exported.

Definition at line 95 of file ExportFileRelations.cpp.

The documentation for this class was generated from the following files:

- Export/ExportFileRelations.h
- Export/ExportFileRelations.cpp

5.145 ExportImageForRegression Class Reference

Class for exporting during regression, when the [Transformation](#) has not yet been defined.

```
#include <ExportImageForRegression.h>
```

Public Member Functions

- [ExportImageForRegression](#) (const QPixmap &pixmap)
Single constructor.
- void [fileExport](#) (const QString &filename) const
Export to the specified file. This is called when the [Transformation](#) has not been defined.

5.145.1 Detailed Description

Class for exporting during regression, when the [Transformation](#) has not yet been defined.

This class just exports the image size

Definition at line 15 of file ExportImageForRegression.h.

The documentation for this class was generated from the following files:

- Export/ExportImageForRegression.h
- Export/ExportImageForRegression.cpp

5.146 ExportOrdinalsSmooth Class Reference

Utility class to interpolate points spaced evenly along a piecewise defined curve with fitted spline.

```
#include <ExportOrdinalsSmooth.h>
```

Public Member Functions

- [ExportOrdinalsSmooth](#) ()
Single constructor.
- void [loadSplinePairsWithoutTransformation](#) (const Points &points, std::vector< double > &t, std::vector< [SplinePair](#) > &xy) const
Load t (=ordinal) and xy (=screen position) spline pairs, without any conversion to graph coordinates.
- void [loadSplinePairsWithTransformation](#) (const Points &points, const [Transformation](#) &transformation, std::vector< double > &t, std::vector< [SplinePair](#) > &xy) const
Load t (=ordinal) and xy (=screen position) spline pairs, converting screen coordinates to graph coordinates.
- ExportValuesOrdinal [ordinalsAtIntervalsGraph](#) (const std::vector< double > &t, const std::vector< [SplinePair](#) > &xy, double pointsInterval) const
Perform the interpolation on the arrays loaded by the other methods.

5.146.1 Detailed Description

Utility class to interpolate points spaced evenly along a piecewise defined curve with fitted spline.

Definition at line 20 of file ExportOrdinalsSmooth.h.

The documentation for this class was generated from the following files:

- Export/ExportOrdinalsSmooth.h
- Export/ExportOrdinalsSmooth.cpp

5.147 ExportOrdinalsStraight Class Reference

Utility class to interpolate points spaced evenly along a piecewise defined curve with line segments between points.

```
#include <ExportOrdinalsStraight.h>
```

Public Member Functions

- [ExportOrdinalsStraight](#) ()
Single constructor.
- ExportValuesOrdinal [ordinalsAtIntervalsGraphWithoutTransformation](#) (const Points &points, double pointsInterval) const
Compute ordinals, without any conversion to graph coordinates.
- ExportValuesOrdinal [ordinalsAtIntervalsGraphWithTransformation](#) (const Points &points, const [Transformation](#) &transformation, double pointsInterval) const
Compute ordinals, converting screen coordinates to graph coordinates.

5.147.1 Detailed Description

Utility class to interpolate points spaced evenly along a piecewise defined curve with line segments between points.

Definition at line 19 of file ExportOrdinalsStraight.h.

The documentation for this class was generated from the following files:

- Export/ExportOrdinalsStraight.h
- Export/ExportOrdinalsStraight.cpp

5.148 ExportToClipboard Class Reference

Strategy class for exporting to the clipboard. This strategy is external to the [Document](#) class so that class is simpler.

```
#include <ExportToClipboard.h>
```

Public Member Functions

- [ExportToClipboard \(\)](#)
Single constructor.
- void [exportToClipboard](#) (const QStringList &selected, const [Transformation](#) &transformation, QTextStream &strCsv, QTextStream &strHtml, const [Curve](#) &curveAxis, const [CurvesGraphs](#) &curvesGraphsAll, [CurvesGraphs](#) &curvesGraphsSelected) const
Export, curve-by-curve, raw data points to a string that will be copied to the clipboard.

5.148.1 Detailed Description

Strategy class for exporting to the clipboard. This strategy is external to the [Document](#) class so that class is simpler.

Definition at line 18 of file [ExportToClipboard.h](#).

5.148.2 Member Function Documentation

5.148.2.1 void [ExportToClipboard::exportToClipboard](#) (const QStringList & *selected*, const [Transformation](#) & *transformation*, QTextStream & *strCsv*, QTextStream & *strHtml*, const [Curve](#) & *curveAxis*, const [CurvesGraphs](#) & *curvesGraphsAll*, [CurvesGraphs](#) & *curvesGraphsSelected*) const

Export, curve-by-curve, raw data points to a string that will be copied to the clipboard.

Parameters

in	<i>selected</i>	Simple list of selected points that will be exported
in	<i>transformation</i>	Transformation which may or may not be defined
out	<i>strCsv</i>	Selected points as comma separated value list
out	<i>strHtml</i>	Selected points as html
in	<i>curveAxis</i>	Axis curve in the Document and its points
in	<i>curvesGraphsAll</i>	All graph curves in the Document and their points
out	<i>curvesGraphsSelected</i>	Selected points as a subset of document.curvesGraphs()

Definition at line 18 of file [ExportToClipboard.cpp](#).

The documentation for this class was generated from the following files:

- [Export/ExportToClipboard.h](#)
- [Export/ExportToClipboard.cpp](#)

5.149 ExportToFile Class Reference

Strategy class for exporting to a file. This strategy is external to the [Document](#) class so that class is simpler.

```
#include <ExportToFile.h>
```

Public Member Functions

- [ExportToFile](#) ()
Single constructor.
- void [exportToFile](#) (const [DocumentModelExportFormat](#) &modelExport, const [Document](#) &document, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QTextStream &str) const
Export [Document](#) points according to the settings.
- QString [fileExtensionCsv](#) () const
File extension for csv export files.
- QString [fileExtensionTsv](#) () const
File extension for tsv export files.
- QString [filterCsv](#) () const
QFileDialog filter for CSV files.
- QString [filterTsv](#) () const
QFileDialog filter for TSV files.

5.149.1 Detailed Description

Strategy class for exporting to a file. This strategy is external to the [Document](#) class so that class is simpler.

Definition at line 25 of file [ExportToFile.h](#).

5.149.2 Member Function Documentation

- 5.149.2.1 void [ExportToFile::exportToFile](#) (const [DocumentModelExportFormat](#) & *modelExport*, const [Document](#) & *document*, const [MainWindowModel](#) & *modelMainWindow*, const [Transformation](#) & *transformation*, QTextStream & *str*) const

Export [Document](#) points according to the settings.

The [DocumentModelExportFormat](#) inside the [Document](#) is ignored so [DlgSettingsExport](#) can supply its own [DocumentModelExportFormat](#) when previewing what would be exported.

Definition at line 23 of file [ExportToFile.cpp](#).

The documentation for this class was generated from the following files:

- [Export/ExportToFile.h](#)
- [Export/ExportToFile.cpp](#)

5.150 ExportXThetaValuesMergedFunctions Class Reference

Creates the set of merged x/theta values for exporting functions, using interpolation.

```
#include <ExportXThetaValuesMergedFunctions.h>
```

Public Member Functions

- [ExportXThetaValuesMergedFunctions](#) (const [DocumentModelExportFormat](#) &modelExport, const [Values](#)↔
VectorXOrY &xThetaValuesRaw, const [Transformation](#) &transformation)
Single constructor.
- [ExportValuesXOrY](#) [xThetaValues](#) () const
Resulting x/theta values for all included functions.

5.150.1 Detailed Description

Creates the set of merged x/theta values for exporting functions, using interpolation.

Definition at line 19 of file [ExportXThetaValuesMergedFunctions.h](#).

The documentation for this class was generated from the following files:

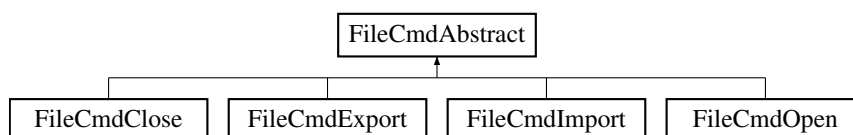
- [Export/ExportXThetaValuesMergedFunctions.h](#)
- [Export/ExportXThetaValuesMergedFunctions.cpp](#)

5.151 FileCmdAbstract Class Reference

Base class for 'file commands' that are used specifically for regression testing of file import/open/export features.

```
#include <FileCmdAbstract.h>
```

Inheritance diagram for FileCmdAbstract:



Public Member Functions

- [FileCmdAbstract](#) (const [QString](#) &cmdDescription)
Single constructor.
- virtual void [redo](#) ([MainWindow](#) &mainWindow)=0
Apply this command, through [MainWindow](#).

Protected Member Functions

- [QString](#) [cmdDescription](#) () const
Command description for logging.

5.151.1 Detailed Description

Base class for 'file commands' that are used specifically for regression testing of file import/open/export features.

These commands operate outside of the normal undo/redo command framework, since that framework uses commands that are attached to an open [Document](#). The file commands follow special rules:

1. Never generated by the code
2. Created by manually editing a 'file command' xml file, which does NOT have a [Document](#) (so error report files cannot be used unless pretty much everything is removed)
3. Are only read during regression testing normally. Although they can be loaded otherwise, there is no point in doing so
4. These commands operate in the forward direction only, since undoing a File Close could be quite messy

Definition at line 22 of file FileCmdAbstract.h.

The documentation for this class was generated from the following files:

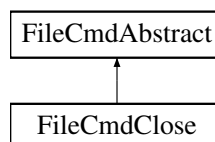
- FileCmd/FileCmdAbstract.h
- FileCmd/FileCmdAbstract.cpp

5.152 FileCmdClose Class Reference

Command for closing a file.

```
#include <FileCmdClose.h>
```

Inheritance diagram for FileCmdClose:



Public Member Functions

- [FileCmdClose](#) (QXmlStreamReader &reader)
Constructor for parsing file script xml.
- virtual void [redo](#) ([MainWindow](#) &mainWindow)
Apply this command, through [MainWindow](#).

Additional Inherited Members

5.152.1 Detailed Description

Command for closing a file.

Definition at line 15 of file FileCmdClose.h.

The documentation for this class was generated from the following files:

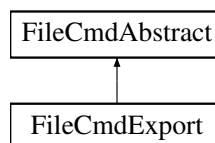
- FileCmd/FileCmdClose.h
- FileCmd/FileCmdClose.cpp

5.153 FileCmdExport Class Reference

Command for exporting a file.

```
#include <FileCmdExport.h>
```

Inheritance diagram for FileCmdExport:



Public Member Functions

- [FileCmdExport](#) (QXmlStreamReader &reader)
Constructor for parsing file script xml.
- virtual void [redo](#) ([MainWindow](#) &mainWindow)
Apply this command, through [MainWindow](#).

Additional Inherited Members

5.153.1 Detailed Description

Command for exporting a file.

Definition at line 15 of file FileCmdExport.h.

The documentation for this class was generated from the following files:

- FileCmd/FileCmdExport.h
- FileCmd/FileCmdExport.cpp

5.154 FileCmdFactory Class Reference

Factory that creates FileCmds from a file cmd script file, in xml format.

```
#include <FileCmdFactory.h>
```

Public Member Functions

- [FileCmdFactory](#) ()
Single constructor.
- [FileCmdAbstract](#) * [createFileCmd](#) (QXmlStreamReader &reader) const
Create one [FileCmdAbstract](#) from the specified xml subtree.

5.154.1 Detailed Description

Factory that creates FileCmds from a file cmd script file, in xml format.

Definition at line 15 of file FileCmdFactory.h.

The documentation for this class was generated from the following files:

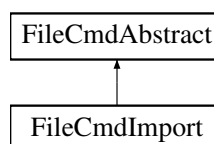
- FileCmd/FileCmdFactory.h
- FileCmd/FileCmdFactory.cpp

5.155 FileCmdImport Class Reference

Command for importing a file.

```
#include <FileCmdImport.h>
```

Inheritance diagram for FileCmdImport:



Public Member Functions

- [FileCmdImport](#) (QXmlStreamReader &reader)
Constructor for parsing file script xml.
- virtual void [redo](#) ([MainWindow](#) &mainWindow)
Apply this command, through [MainWindow](#).

Additional Inherited Members

5.155.1 Detailed Description

Command for importing a file.

Definition at line 15 of file FileCmdImport.h.

The documentation for this class was generated from the following files:

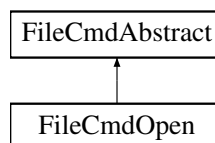
- FileCmd/FileCmdImport.h
- FileCmd/FileCmdImport.cpp

5.156 FileCmdOpen Class Reference

Command for opening a file.

```
#include <FileCmdOpen.h>
```

Inheritance diagram for FileCmdOpen:



Public Member Functions

- [FileCmdOpen](#) (QXmlStreamReader &reader)
Constructor for parsing file script xml.
- virtual void [redo](#) ([MainWindow](#) &mainWindow)
Apply this command, through [MainWindow](#).

Additional Inherited Members

5.156.1 Detailed Description

Command for opening a file.

Definition at line 15 of file FileCmdOpen.h.

The documentation for this class was generated from the following files:

- FileCmd/FileCmdOpen.h
- FileCmd/FileCmdOpen.cpp

5.157 FileCmdScript Class Reference

File that manages a command stack for regression testing of file import/open/export/close.

```
#include <FileCmdScript.h>
```

Public Member Functions

- [FileCmdScript](#) (const QString &fileCmdScriptFile)
Single constructor.
- bool [canRedo](#) () const
Returns true if there is at least one command on the stack.
- void [redo](#) ([MainWindow](#) &mainWindow)
Apply the next command. Requires non-empty stack.

5.157.1 Detailed Description

File that manages a command stack for regression testing of file import/open/export/close.

This command stack (with a lifetime the same as the application's) is independent of the command stack in [CmdMediator](#) (which is Document-specific)

Definition at line 20 of file FileCmdScript.h.

The documentation for this class was generated from the following files:

- FileCmd/FileCmdScript.h
- FileCmd/FileCmdScript.cpp

5.158 FilterImage Class Reference

Filters an image using a combination of color filtering and grid removal.

```
#include <FilterImage.h>
```

Public Member Functions

- [FilterImage](#) ()
Single constructor.
- QPixmap [filter](#) (const QImage &imageUnfiltered, const [Transformation](#) &transformation, const QString &curveSelected, const [DocumentModelColorFilter](#) &modelColorFilter, const [DocumentModelGridRemoval](#) &modelGridRemoval) const
Filter original unfiltered image into filtered pixmap.

5.158.1 Detailed Description

Filters an image using a combination of color filtering and grid removal.

Definition at line 18 of file FilterImage.h.

The documentation for this class was generated from the following files:

- Filter/FilterImage.h
- Filter/FilterImage.cpp

5.159 FormatCoordsUnits Class Reference

Highest-level wrapper around other Formats classes.

```
#include <FormatCoordsUnits.h>
```

Public Member Functions

- [FormatCoordsUnits](#) ()
Single constructor.
- void [formattedToUnformatted](#) (const QString &xThetaFormatted, const QString &yRadiusFormatted, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &mainWindowModel, double &xThetaUnformatted, double &yRadiusUnformatted) const
Convert formatted string to unformatted numeric value.
- void [unformattedToFormatted](#) (double xThetaUnformatted, double yRadiusUnformatted, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &mainWindowModel, QString &xThetaFormatted, QString &yRadiusFormatted, const [Transformation](#) &transformation) const
Convert unformatted numeric value to formatted string. [Transformation](#) is used to determine best resolution.

5.159.1 Detailed Description

Highest-level wrapper around other Formats classes.

Definition at line 16 of file FormatCoordsUnits.h.

The documentation for this class was generated from the following files:

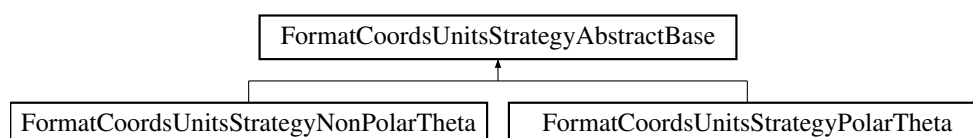
- Format/FormatCoordsUnits.h
- Format/FormatCoordsUnits.cpp

5.160 FormatCoordsUnitsStrategyAbstractBase Class Reference

Common methods for formatting strategies.

```
#include <FormatCoordsUnitsStrategyAbstractBase.h>
```

Inheritance diagram for FormatCoordsUnitsStrategyAbstractBase:



Public Member Functions

- [FormatCoordsUnitsStrategyAbstractBase](#) ()

Single constructor.

Protected Member Functions

- int [precisionDigitsForRawNumber](#) (double valueUnformatted, double valueUnformattedOther, bool isXTheta, const [Transformation](#) &transformation) const

Compute precision for outputting an unformatted value, consistent with the resolution at the point where that point lies.

5.160.1 Detailed Description

Common methods for formatting strategies.

Definition at line 13 of file FormatCoordsUnitsStrategyAbstractBase.h.

5.160.2 Member Function Documentation

5.160.2.1 int [FormatCoordsUnitsStrategyAbstractBase::precisionDigitsForRawNumber](#) (double *valueUnformatted*, double *valueUnformattedOther*, bool *isXTheta*, const [Transformation](#) & *transformation*) const [protected]

Compute precision for outputting an unformatted value, consistent with the resolution at the point where that point lies.

This algorithm causes many digits to appear when a graph's dynamic range is relatively small (like -118.4 to -118.2 degrees in longitude), and fewer digits to appear when a graph's dynamic range is relatively large (like 0 to 100)

Definition at line 16 of file FormatCoordsUnitsStrategyAbstractBase.cpp.

The documentation for this class was generated from the following files:

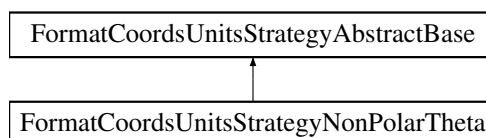
- Format/FormatCoordsUnitsStrategyAbstractBase.h
- Format/FormatCoordsUnitsStrategyAbstractBase.cpp

5.161 FormatCoordsUnitsStrategyNonPolarTheta Class Reference

Format conversions between unformatted and formatted for CoordUnitsNonPolarTheta.

```
#include <FormatCoordsUnitsStrategyNonPolarTheta.h>
```

Inheritance diagram for FormatCoordsUnitsStrategyNonPolarTheta:



Public Member Functions

- [FormatCoordsUnitsStrategyNonPolarTheta](#) ()
Single constructor.
- double [formattedToUnformatted](#) (const QString &string, const QLocale &locale, CoordUnitsNonPolarTheta coordUnits, CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime) const
Convert formatted string to simple unformatted number.
- QString [unformattedToFormatted](#) (double valueUnformatted, const QLocale &locale, CoordUnitsNonPolarTheta coordUnits, CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, bool isXTheta, const [Transformation](#) &transformation, double valueUnformattedOther) const
Convert simple unformatted number to formatted string.

Additional Inherited Members

5.161.1 Detailed Description

Format conversions between unformatted and formatted for CoordUnitsNonPolarTheta.

Definition at line 20 of file FormatCoordsUnitsStrategyNonPolarTheta.h.

The documentation for this class was generated from the following files:

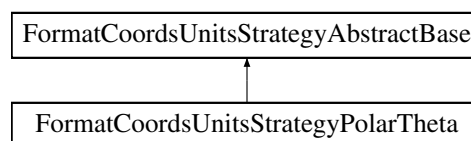
- Format/FormatCoordsUnitsStrategyNonPolarTheta.h
- Format/FormatCoordsUnitsStrategyNonPolarTheta.cpp

5.162 FormatCoordsUnitsStrategyPolarTheta Class Reference

Format conversions between unformatted and formatted for CoordUnitsStrategyPolarTheta.

```
#include <FormatCoordsUnitsStrategyPolarTheta.h>
```

Inheritance diagram for FormatCoordsUnitsStrategyPolarTheta:



Public Member Functions

- [FormatCoordsUnitsStrategyPolarTheta](#) ()
Single constructor.
- double [formattedToUnformatted](#) (const QString &string, const QLocale &locale, CoordUnitsPolarTheta coordUnits) const
Convert formatted string to simple unformatted number.
- QString [unformattedToFormatted](#) (double valueUnformatted, const QLocale &locale, CoordUnitsPolarTheta coordUnits, const [Transformation](#) &transformation, double valueUnformattedOther) const
Convert simple unformatted number to formatted string.

Additional Inherited Members

5.162.1 Detailed Description

Format conversions between unformatted and formatted for CoordUnitsStrategyPolarTheta.

Definition at line 18 of file FormatCoordsUnitsStrategyPolarTheta.h.

The documentation for this class was generated from the following files:

- Format/FormatCoordsUnitsStrategyPolarTheta.h
- Format/FormatCoordsUnitsStrategyPolarTheta.cpp

5.163 FormatDateTime Class Reference

Input parsing and output formatting for date/time values.

```
#include <FormatDateTime.h>
```

Public Member Functions

- [FormatDateTime](#) ()
Single constructor.
- QString [formatOutput](#) (CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, double value) const
Format the date/time value according to date/time format settings.
- QValidator::State [parseInput](#) (CoordUnitsDate coordUnitsDate, CoordUnitsTime coordUnitsTime, const QString &stringUntrimmed, double &value) const
Parse the input string into a time value.

5.163.1 Detailed Description

Input parsing and output formatting for date/time values.

Definition at line 25 of file FormatDateTime.h.

5.163.2 Member Function Documentation

- 5.163.2.1 QValidator::State FormatDateTime::parseInput (CoordUnitsDate *coordUnitsDate*, CoordUnitsTime *coordUnitsTime*, const QString & *stringUntrimmed*, double & *value*) const

Parse the input string into a time value.

Success flag is false if parsing failed. Leading/trailing spaces are trimmed (=ignored)

Definition at line 417 of file FormatDateTime.cpp.

The documentation for this class was generated from the following files:

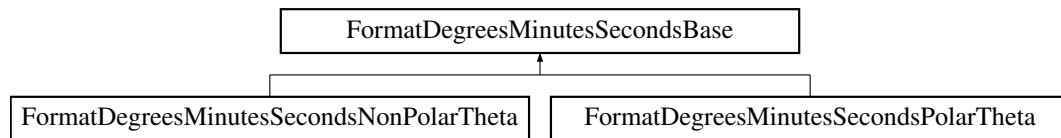
- Format/FormatDateTime.h
- Format/FormatDateTime.cpp

5.164 FormatDegreesMinutesSecondsBase Class Reference

Common input parsing and output formatting for degrees/minutes/seconds values.

```
#include <FormatDegreesMinutesSecondsBase.h>
```

Inheritance diagram for FormatDegreesMinutesSecondsBase:



Public Member Functions

- [FormatDegreesMinutesSecondsBase](#) ()
Single constructor.
- `QValidator::State` [parseInput](#) (const QString &stringUntrimmed, double &value) const
Parse the input string into a number value.

Protected Member Functions

- `QString` [formatOutputDegreesMinutesSeconds](#) (double value) const
Format as degrees, minutes and seconds without hemisphere.
- `QString` [formatOutputDegreesMinutesSecondsNsew](#) (double value, bool isNsHemisphere) const
Format as degrees, minutes and seconds with hemisphere.

5.164.1 Detailed Description

Common input parsing and output formatting for degrees/minutes/seconds values.

Definition at line 14 of file FormatDegreesMinutesSecondsBase.h.

5.164.2 Member Function Documentation

5.164.2.1 `QValidator::State` [FormatDegreesMinutesSecondsBase::parseInput](#) (const QString & *stringUntrimmed*, double & *value*) const

Parse the input string into a number value.

Success flag is false if the parsing failed. Either signed values or hemisphere (North, South, East, West) values can be accepted irregardless of the output format selected by the user. Leading/trailing spaces are trimmed. Leading/trailing spaces are trimmed (=ignored)

Definition at line 87 of file FormatDegreesMinutesSecondsBase.cpp.

The documentation for this class was generated from the following files:

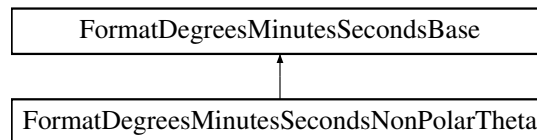
- Format/FormatDegreesMinutesSecondsBase.h
- Format/FormatDegreesMinutesSecondsBase.cpp

5.165 FormatDegreesMinutesSecondsNonPolarTheta Class Reference

Angular units according to CoordUnitsNonPolarTheta.

```
#include <FormatDegreesMinutesSecondsNonPolarTheta.h>
```

Inheritance diagram for FormatDegreesMinutesSecondsNonPolarTheta:



Public Member Functions

- [FormatDegreesMinutesSecondsNonPolarTheta \(\)](#)
Single constructor.
- QString [formatOutput](#) (CoordUnitsNonPolarTheta coordUnits, double value, bool isXTheta) const
Format the degrees/minutes/seconds value. Distinguishing x/theta versus y/radius is required for N/S/E/W hemispheres.

Additional Inherited Members

5.165.1 Detailed Description

Angular units according to CoordUnitsNonPolarTheta.

Definition at line 15 of file FormatDegreesMinutesSecondsNonPolarTheta.h.

The documentation for this class was generated from the following files:

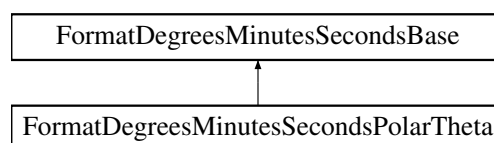
- Format/FormatDegreesMinutesSecondsNonPolarTheta.h
- Format/FormatDegreesMinutesSecondsNonPolarTheta.cpp

5.166 FormatDegreesMinutesSecondsPolarTheta Class Reference

Angular units according to CoordUnitsPolarTheta.

```
#include <FormatDegreesMinutesSecondsPolarTheta.h>
```

Inheritance diagram for FormatDegreesMinutesSecondsPolarTheta:



Public Member Functions

- [FormatDegreesMinutesSecondsPolarTheta](#) ()
Single constructor.
- QString [formatOutput](#) (CoordUnitsPolarTheta coordUnits, double value, bool isXTheta) const
Format the degrees/minutes/seconds value. Distinguishing x/theta versus y/radius is required for N/S/E/W hemispheres.

Additional Inherited Members

5.166.1 Detailed Description

Angular units according to CoordUnitsPolarTheta.

Definition at line 15 of file FormatDegreesMinutesSecondsPolarTheta.h.

The documentation for this class was generated from the following files:

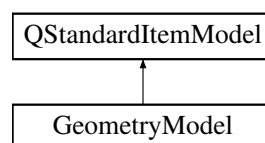
- Format/FormatDegreesMinutesSecondsPolarTheta.h
- Format/FormatDegreesMinutesSecondsPolarTheta.cpp

5.167 GeometryModel Class Reference

Model that adds row highlighting according to the currently select point identifier.

```
#include <GeometryModel.h>
```

Inheritance diagram for GeometryModel:



Public Member Functions

- [GeometryModel](#) ()
Single constructor.
- virtual QVariant [data](#) (const QModelIndex &index, int role=Qt::DisplayRole) const
Override for special processing.
- void [setCurrentPointIdentifier](#) (const QString &pointIdentifier)
Set the point identifier to be highlighted. Value is empty for no highlighting.

5.167.1 Detailed Description

Model that adds row highlighting according to the currently select point identifier.

Definition at line 14 of file GeometryModel.h.

The documentation for this class was generated from the following files:

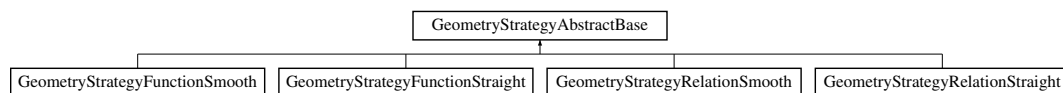
- Geometry/GeometryModel.h
- Geometry/GeometryModel.cpp

5.168 GeometryStrategyAbstractBase Class Reference

Base class for all geometry strategies.

```
#include <GeometryStrategyAbstractBase.h>
```

Inheritance diagram for GeometryStrategyAbstractBase:



Public Member Functions

- [GeometryStrategyAbstractBase](#) ()
Single constructor.
- virtual void [calculateGeometry](#) (const Points &points, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QString &funcArea, QString &polyArea, QVector< QString > &x, QVector< QString > &y, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const =0
Calculate geometry parameters.

Protected Member Functions

- void [calculatePositionsGraph](#) (const Points &points, const [Transformation](#) &transformation, QVector< QPointF > &positionsGraph) const
Convert screen positions to graph positions.
- double [functionArea](#) (const QVector< QPointF > &positionsGraph) const
Use trapezoidal approximation to compute area under the function. Does not apply to relation.
- void [insertSubintervalsAndLoadDistances](#) (int subintervalsPerInterval, const QVector< QPointF > &positionsGraph, QVector< QPointF > &positionsGraphWithSubintervals, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const
Insert the specified number of subintervals into each interval.
- void [loadXY](#) (const QVector< QPointF > &positionsGraph, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QVector< QString > &x, QVector< QString > &y) const
Load x and y coordinate vectors.
- double [polygonAreaForSimplyConnected](#) (const QVector< QPointF > &points) const
Area in polygon using Shoelace formula, which only works if polygon is simply connected.

5.168.1 Detailed Description

Base class for all geometry strategies.

Each strategy computes geometry parameters according to the curve's settings.

The numbering for the strategies is specified as the CurveConnectAs enumeration

Definition at line 22 of file GeometryStrategyAbstractBase.h.

5.168.2 Member Function Documentation

5.168.2.1 `void GeometryStrategyAbstractBase::insertSubintervalsAndLoadDistances (int subintervalsPerInterval, const QVector< QPointF > & positionsGraph, QVector< QPointF > & positionsGraphWithSubintervals, QVector< QString > & distanceGraphForward, QVector< QString > & distancePercentForward, QVector< QString > & distanceGraphBackward, QVector< QString > & distancePercentBackward) const` `[protected]`

Insert the specified number of subintervals into each interval.

For straight curves `subintervalsPerInterval=1` so the linearity is maintained, and for smooth curves `subintervalsPerInterval>1` so the geometry calculations take into account the curvature(s) of the line

Definition at line 61 of file GeometryStrategyAbstractBase.cpp.

5.168.2.2 `double GeometryStrategyAbstractBase::polygonAreaForSimplyConnected (const QVector< QPointF > & points) const` `[protected]`

Area in polygon using Shoelace formula, which only works if polygon is simply connected.

We do not check to see if the polygon is simply connected since that would be (1) slow and (2) much work

Definition at line 164 of file GeometryStrategyAbstractBase.cpp.

The documentation for this class was generated from the following files:

- Geometry/GeometryStrategyAbstractBase.h
- Geometry/GeometryStrategyAbstractBase.cpp

5.169 GeometryStrategyContext Class Reference

Class for that manages geometry strategies.

```
#include <GeometryStrategyContext.h>
```

Public Member Functions

- [GeometryStrategyContext](#) ()
Single constructor.
- void [calculateGeometry](#) (const Points &points, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, CurveConnectAs connectAs, QString &funcArea, QString &polyArea, QVector< QString > &x, QVector< QString > &y, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const
Calculate geometry parameters.

5.169.1 Detailed Description

Class for that manages geometry strategies.

Definition at line 20 of file GeometryStrategyContext.h.

The documentation for this class was generated from the following files:

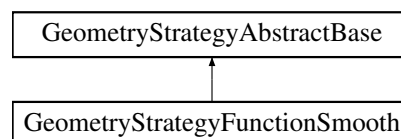
- Geometry/GeometryStrategyContext.h
- Geometry/GeometryStrategyContext.cpp

5.170 GeometryStrategyFunctionSmooth Class Reference

Calculate for line through the points that is smoothly connected as a function.

```
#include <GeometryStrategyFunctionSmooth.h>
```

Inheritance diagram for GeometryStrategyFunctionSmooth:



Public Member Functions

- [GeometryStrategyFunctionSmooth](#) ()
Single constructor.
- virtual void [calculateGeometry](#) (const Points &points, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QString &funcArea, QString &polyArea, QVector< QString > &x, QVector< QString > &y, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const
Calculate geometry parameters.

Additional Inherited Members

5.170.1 Detailed Description

Calculate for line through the points that is smoothly connected as a function.

Definition at line 16 of file GeometryStrategyFunctionSmooth.h.

The documentation for this class was generated from the following files:

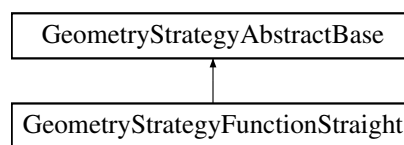
- Geometry/GeometryStrategyFunctionSmooth.h
- Geometry/GeometryStrategyFunctionSmooth.cpp

5.171 GeometryStrategyFunctionStraight Class Reference

Calculate for line through the points that is straightly connected as a function.

```
#include <GeometryStrategyFunctionStraight.h>
```

Inheritance diagram for GeometryStrategyFunctionStraight:



Public Member Functions

- [GeometryStrategyFunctionStraight\(\)](#)
Single constructor.
- virtual void [calculateGeometry](#) (const Points &points, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QString &funcArea, QString &polyArea, QVector< QString > &x, QVector< QString > &y, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const
Calculate geometry parameters.

Additional Inherited Members

5.171.1 Detailed Description

Calculate for line through the points that is straightly connected as a function.

Definition at line 16 of file GeometryStrategyFunctionStraight.h.

The documentation for this class was generated from the following files:

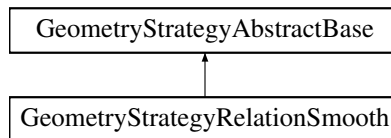
- Geometry/GeometryStrategyFunctionStraight.h
- Geometry/GeometryStrategyFunctionStraight.cpp

5.172 GeometryStrategyRelationSmooth Class Reference

Calculate for line through the points that is smoothly connected as a relation.

```
#include <GeometryStrategyRelationSmooth.h>
```

Inheritance diagram for GeometryStrategyRelationSmooth:



Public Member Functions

- [GeometryStrategyRelationSmooth \(\)](#)
Single constructor.
- virtual void [calculateGeometry](#) (const Points &points, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QString &funcArea, QString &polyArea, QVector< QString > &x, QVector< QString > &y, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const
Calculate geometry parameters.

Additional Inherited Members

5.172.1 Detailed Description

Calculate for line through the points that is smoothly connected as a relation.

Definition at line 16 of file GeometryStrategyRelationSmooth.h.

The documentation for this class was generated from the following files:

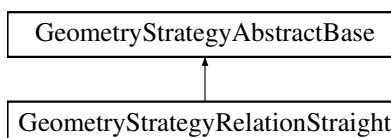
- Geometry/GeometryStrategyRelationSmooth.h
- Geometry/GeometryStrategyRelationSmooth.cpp

5.173 GeometryStrategyRelationStraight Class Reference

Calculate for line through the points that is straightly connected as a relation.

```
#include <GeometryStrategyRelationStraight.h>
```

Inheritance diagram for GeometryStrategyRelationStraight:



Public Member Functions

- [GeometryStrategyRelationStraight](#) ()

Single constructor.

- virtual void [calculateGeometry](#) (const Points &points, const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, QString &funcArea, QString &polyArea, QVector< QString > &x, QVector< QString > &y, QVector< QString > &distanceGraphForward, QVector< QString > &distancePercentForward, QVector< QString > &distanceGraphBackward, QVector< QString > &distancePercentBackward) const

Calculate geometry parameters.

Additional Inherited Members

5.173.1 Detailed Description

Calculate for line through the points that is straightly connected as a relation.

Definition at line 16 of file GeometryStrategyRelationStraight.h.

The documentation for this class was generated from the following files:

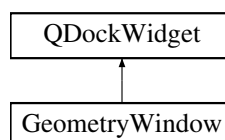
- Geometry/GeometryStrategyRelationStraight.h
- Geometry/GeometryStrategyRelationStraight.cpp

5.174 GeometryWindow Class Reference

Window that displays the geometry information, as a table, for the current curve.

```
#include <GeometryWindow.h>
```

Inheritance diagram for GeometryWindow:



Public Slots

- void [slotPointHoverEnter](#) (QString)
Highlight the row for the specified point.
- void [slotPointHoverLeave](#) (QString)
Unhighlight the row for the specified point.
- void [slotSelectionChanged](#) (const QTableWidgetItem &, const QTableWidgetItem &)
Prepare for copy after selection has changed.

Signals

- void [signalGeometryWindowClosed](#) ()
Signal that this QDockWidget was just closed.

Public Member Functions

- [GeometryWindow](#) (QWidget *parent)
Single constructor. Parent is needed or else this widget cannot be redocked after being undocked.
- void [clear](#) ()
Clear stale information.
- virtual void [closeEvent](#) (QCloseEvent *event)
Catch close event so corresponding menu item in [MainWindow](#) can be updated accordingly.
- void [update](#) (const [CmdMediator](#) &cmdMediator, const [MainWindowModel](#) &modelMainWindow, const Q↔String &curveSelected, const [Transformation](#) &transformation)
Populate the table with the specified [Curve](#).

Static Public Member Functions

- static int [columnBodyPointIdentifiers](#) ()
Hidden column that has the point identifiers.

5.174.1 Detailed Description

Window that displays the geometry information, as a table, for the current curve.

Column COLUMN_BODY_POINT_IDENTIFIERS is hidden. It contains the point identifiers so we can find the line associated with a point, and then highlight that line

Definition at line 27 of file GeometryWindow.h.

The documentation for this class was generated from the following files:

- Geometry/GeometryWindow.h
- Geometry/GeometryWindow.cpp

5.175 GhostEllipse Class Reference

Ghost for a QGraphicsEllipseItem.

```
#include <GhostEllipse.h>
```

Public Member Functions

- [GhostEllipse](#) (const QRectF &[rect](#), const QPen &[pen](#), const QBrush &[brush](#))
Initial constructor.
- [GhostEllipse](#) (const [GhostEllipse](#) &other)
Copy constructor.
- [GhostEllipse](#) & [operator=](#) (const [GhostEllipse](#) &other)
Assignment operator.
- QBrush [brush](#) () const
Get method for brush.
- QPen [pen](#) () const
Get method for pen.
- QRectF [rect](#) () const
Get method for bounding rectangle.

5.175.1 Detailed Description

Ghost for a QGraphicsEllipseItem.

Definition at line 15 of file GhostEllipse.h.

The documentation for this class was generated from the following files:

- Ghosts/GhostEllipse.h
- Ghosts/GhostEllipse.cpp

5.176 GhostPath Class Reference

Ghost for a QGraphicsPathItem.

```
#include <GhostPath.h>
```

Public Member Functions

- [GhostPath](#) (const QPainterPath &[path](#), const QPen &[pen](#), const QBrush &[brush](#))
Initial constructor.
- [GhostPath](#) (const [GhostPath](#) &other)
Copy constructor.
- [GhostPath](#) & [operator=](#) (const [GhostPath](#) &other)
Assignment operator.
- QBrush [brush](#) () const
Get method for brush.
- QPainterPath [path](#) () const
Get method for path.
- QPen [pen](#) () const
Get method for pen.

5.176.1 Detailed Description

Ghost for a QGraphicsPathItem.

Definition at line 15 of file GhostPath.h.

The documentation for this class was generated from the following files:

- Ghosts/GhostPath.h
- Ghosts/GhostPath.cpp

5.177 GhostPolygon Class Reference

Ghost for a QGraphicsPolygonItem.

```
#include <GhostPolygon.h>
```

Public Member Functions

- [GhostPolygon](#) (const QPolygonF &[polygon](#), const QPen &[pen](#), const QBrush &[brush](#))
Initial constructor.
- [GhostPolygon](#) (const [GhostPolygon](#) &other)
Copy constructor.
- [GhostPolygon](#) & [operator=](#) (const [GhostPolygon](#) &other)
Assignment operator.
- QBrush [brush](#) () const
Get method for brush.
- QPen [pen](#) () const
Get method for pen.
- QPolygonF [polygon](#) () const
Get method for polygon.

5.177.1 Detailed Description

Ghost for a QGraphicsPolygonItem.

Definition at line 15 of file GhostPolygon.h.

The documentation for this class was generated from the following files:

- Ghosts/GhostPolygon.h
- Ghosts/GhostPolygon.cpp

5.178 Ghosts Class Reference

Class for showing points and lines for all coordinate systems simultaneously, even though the code normally only allows graphical items for once coordinate system to be visible at a time.

```
#include <Ghosts.h>
```

Public Member Functions

- [Ghosts](#) (unsigned int [coordSystemIndexToBeRestored](#))
Single constructor.
- unsigned int [coordSystemIndexToBeRestored](#) () const
Coordinate system index that was active before the ghosts.
- void [captureGraphicsItems](#) (QGraphicsScene &scene)
Take a snapshot of the graphics items.
- void [createGhosts](#) (QGraphicsScene &scene)
Create ghosts from the path/rect/polygon lists.
- void [destroyGhosts](#) (QGraphicsScene &scene)
Destory ghosts. Called at end of algorithm.

5.178.1 Detailed Description

Class for showing points and lines for all coordinate systems simultaneously, even though the code normally only allows graphical items for once coordinate system to be visible at a time.

QGraphicsLineItems are ignored since those are just used for the AxesChecker, and QGraphicsPixmapItems are ignored since those are just used for the background. The other QGraphicsItem subclasses are captured and converted into ghosts.

Definition at line 26 of file Ghosts.h.

The documentation for this class was generated from the following files:

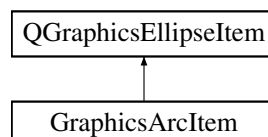
- Ghosts/Ghosts.h
- Ghosts/Ghosts.cpp

5.179 GraphicsArcItem Class Reference

Draw an arc as an ellipse but without lines from the center to the start and end points.

```
#include <GraphicsArcItem.h>
```

Inheritance diagram for GraphicsArcItem:



Public Member Functions

- [GraphicsArcItem](#) (double x, double y, double width, double height, QGraphicsItem *parent=0)
Constructor with individual coordinates.
- [GraphicsArcItem](#) (const QRectF &rect, QGraphicsItem *parent=0)
Constructor with coordinates specified as rectangle.
- virtual void [paint](#) (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
Paint without interior fill.

5.179.1 Detailed Description

Draw an arc as an ellipse but without lines from the center to the start and end points.

Originally this class overrode `QGraphicsEllipseItem::boundingRect` and called `QGraphicsScene::boundingRect`. However, that led to an infinite loop since `QGraphicsScene::boundingRect` looped back around to `QGraphicsEllipseItem::boundingRect`

Definition at line 17 of file `GraphicsArcItem.h`.

The documentation for this class was generated from the following files:

- `Graphics/GraphicsArcItem.h`
- `Graphics/GraphicsArcItem.cpp`

5.180 GraphicsItemsExtractor Class Reference

This class consolidates utility routines that deal with graphics items that are getting extracted from the scene.

```
#include <GraphicsItemsExtractor.h>
```

Public Member Functions

- [GraphicsItemsExtractor](#) ()
Single constructor.
- bool [allSelectedItemsAreEitherAxisOrGraph](#) (const QList< QGraphicsItem * > &items, AxisOrGraph axisOrGraph) const
Return true if all selected points are of the specified axis or graph type.
- QStringList [selectedPointIdentifiers](#) (const QList< QGraphicsItem * > &items) const
Return list of selected point identifiers.

5.180.1 Detailed Description

This class consolidates utility routines that deal with graphics items that are getting extracted from the scene.

Definition at line 20 of file `GraphicsItemsExtractor.h`.

The documentation for this class was generated from the following files:

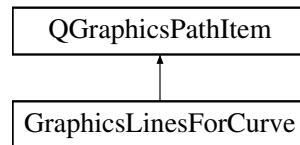
- `Graphics/GraphicsItemsExtractor.h`
- `Graphics/GraphicsItemsExtractor.cpp`

5.181 GraphicsLinesForCurve Class Reference

This class stores the GraphicsLine objects for one [Curve](#).

```
#include <GraphicsLinesForCurve.h>
```

Inheritance diagram for GraphicsLinesForCurve:



Public Member Functions

- [GraphicsLinesForCurve](#) (const QString &curveName)
Single constructor.
- void [addPoint](#) (const QString &pointIdentifier, double ordinal, [GraphicsPoint](#) &point)
Add new line.
- double [identifierToOrdinal](#) (const QString &identifier) const
Get ordinal for specified identifier.
- void [lineMembershipPurge](#) (const [LineStyle](#) &lineStyle)
Mark the end of addPoint calls. Remove stale lines, insert missing lines, and draw the graphics lines.
- void [lineMembershipReset](#) ()
Mark points as unwanted. Afterwards, lineMembershipPurge gets called.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [removePoint](#) (double ordinal)
Remove the specified point. The act of deleting it will automatically remove it from the [GraphicsScene](#).
- void [removeTemporaryPointIfExists](#) ()
Remove temporary point if it exists.
- void [updateAfterCommand](#) ([GraphicsScene](#) &scene, const [PointStyle](#) &pointStyle, const [Point](#) &point, [GeometryWindow](#) *geometryWindow)
Update the [GraphicsScene](#) with the specified [Point](#) from the [Document](#). If it does not exist yet in the scene, we add it.
- void [updateCurveStyle](#) (const [CurveStyle](#) &curveStyle)
Update the curve style for this curve.
- void [updateGraphicsLinesToMatchGraphicsPoints](#) (const [LineStyle](#) &lineStyle)
Calls to moveLinesWithDraggedPoint have finished so update the lines correspondingly.
- void [updateHighlightOpacity](#) (double highlightOpacity)
Update the highlight opacity value. This may or may not affect the current display immediately depending on the state.
- void [updatePointOrdinalsAfterDrag](#) (const [LineStyle](#) &lineStyle, const [Transformation](#) &transformation)
See GraphicsScene::updateOrdinalsAfterDrag. Pretty much the same steps as [Curve::updatePointOrdinals](#).

5.181.1 Detailed Description

This class stores the GraphicsLine objects for one [Curve](#).

The container is a QMap since that container maintains order by key

Definition at line 25 of file GraphicsLinesForCurve.h.

5.181.2 Member Function Documentation

5.181.2.1 void GraphicsLinesForCurve::addPoint (const QString &pointIdentifier, double ordinal, GraphicsPoint &point)

Add new line.

The [GraphicsPoint](#) arguments are not const since this line binds to the points, so dragging points also drags the lines. The ordinal is already in the [GraphicsPoint](#) as DATA_KEY_ORDINAL

Definition at line 53 of file GraphicsLinesForCurve.cpp.

5.181.2.2 void GraphicsLinesForCurve::removeTemporaryPointIfExists ()

Remove temporary point if it exists.

Temporary point handling is so complicated that this method quietly allows redundant calls to this method, without complaining that the point has already been removed when called again

Definition at line 281 of file GraphicsLinesForCurve.cpp.

The documentation for this class was generated from the following files:

- Graphics/GraphicsLinesForCurve.h
- Graphics/GraphicsLinesForCurve.cpp

5.182 GraphicsLinesForCurves Class Reference

This class stores the [GraphicsLinesForCurves](#) objects, one per [Curve](#).

```
#include <GraphicsLinesForCurves.h>
```

Public Member Functions

- [GraphicsLinesForCurves](#) ()
Single constructor.
- void [addPoint](#) (const QString &curveName, const QString &pointIdentifier, double ordinal, [GraphicsPoint](#) &point)
Add new point.
- void [addRemoveCurves](#) ([GraphicsScene](#) &scene, const QStringList &curveNames)
Add new curves and remove expired curves to match the specified list.
- void [lineMembershipPurge](#) (const [CurveStyles](#) &curveStyles)
Mark the end of addPoint calls. Remove stale lines, insert missing lines, and draw the graphics lines.
- void [lineMembershipReset](#) ()
Mark points as unwanted. Afterwards, lineMembershipPurge gets called.
- void [print](#) () const
Debugging method for printing directly from symbolic debugger.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [removePoint](#) (const QString &identifier)

- Remove the specified point. The act of deleting it will automatically remove it from the [GraphicsScene](#).*
- void [removeTemporaryPointIfExists](#) ()
Remove temporary point if it exists.
- void [resetOnLoad](#) ()
Reset, when loading a document after the first, to same state that first document was at when loaded.
- void [updateAfterCommand](#) ([GraphicsScene](#) &scene, const [CurveStyles](#) &curveStyles, const QString &curveName, const [Point](#) &point, [GeometryWindow](#) *geometryWindow)
Update the [GraphicsScene](#) with the specified [Point](#) from the [Document](#). If it does not exist yet in the scene, we add it.
- void [updateCurveStyles](#) (const [CurveStyles](#) &modelCurveStyles)
Update the curve style for every curve.
- void [updateGraphicsLinesToMatchGraphicsPoints](#) (const [CurveStyles](#) &curveStyles)
Calls to [moveLinesWithDraggedPoint](#) have finished so update the lines correspondingly.
- void [updateHighlightOpacity](#) (double highlightOpacity)
Update the highlight opacity value. This may or may not affect the current display immediately depending on the state.
- void [updatePointOrdinalsAfterDrag](#) (const [CurveStyles](#) &curveStyles, const [Transformation](#) &transformation)
See [GraphicsScene::updateOrdinalsAfterDrag](#).

5.182.1 Detailed Description

This class stores the [GraphicsLinesForCurves](#) objects, one per [Curve](#).

Definition at line 26 of file [GraphicsLinesForCurves.h](#).

5.182.2 Member Function Documentation

5.182.2.1 void [GraphicsLinesForCurves::addPoint](#) (const QString & curveName, const QString & pointIdentifier, double ordinal, [GraphicsPoint](#) & point)

Add new point.

The ordinal is already in the [GraphicsPoint](#) as DATA_KEY_ORDINAL

Definition at line 28 of file [GraphicsLinesForCurves.cpp](#).

5.182.2.2 void [GraphicsLinesForCurves::removeTemporaryPointIfExists](#) ()

Remove temporary point if it exists.

Temporary point handling is so complicated that this method quietly allows redundant calls to this method, without complaining that the point has already been removed when called again

Definition at line 149 of file [GraphicsLinesForCurves.cpp](#).

The documentation for this class was generated from the following files:

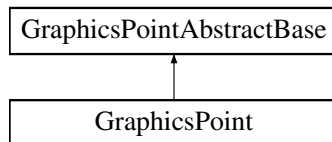
- [Graphics/GraphicsLinesForCurves.h](#)
- [Graphics/GraphicsLinesForCurves.cpp](#)

5.183 GraphicsPoint Class Reference

Graphics item for drawing a circular or polygonal [Point](#).

```
#include <GraphicsPoint.h>
```

Inheritance diagram for GraphicsPoint:



Public Member Functions

- [GraphicsPoint](#) (QGraphicsScene &scene, const QString &identifier, const QPointF &posScreen, const QColor &color, unsigned int radius, double lineWidth, [GeometryWindow](#) *geometryWindow)
Constructor of circular point.
- [GraphicsPoint](#) (QGraphicsScene &scene, const QString &identifier, const QPointF &posScreen, const QColor &color, const QPolygonF &polygon, double lineWidth, [GeometryWindow](#) *geometryWindow)
Constructor of polygon point.
- [~GraphicsPoint](#) ()
Destructor. This remove the graphics item from the scene.
- QVariant [data](#) (int key) const
Proxy method for QGraphicsItem::data.
- double [highlightOpacity](#) () const
Get method for highlight opacity.
- QPointF [pos](#) () const
Proxy method for QGraphicsItem::pos.
- void [printStream](#) (QString indentation, QTextStream &str, double ordinalKey) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [reset](#) ()
Mark point as unwanted, and unbind any bound lines.
- void [setData](#) (int key, const QVariant &data)
Proxy method for QGraphicsItem::setData.
- void [setHighlightOpacity](#) (double [highlightOpacity](#))
Set method for highlight opacity.
- void [setPointStyle](#) (const [PointStyle](#) &pointStyle)
Update the point style.
- void [setPos](#) (const QPointF [pos](#))
Update the position.
- void [setWanted](#) ()
Mark point as wanted. Marking as unwanted is done by the reset function.
- void [updateCurveStyle](#) (const [CurveStyle](#) &curveStyle)
Update point and line styles that comprise the curve style.
- bool [wanted](#) () const
Identify point as wanted/unwanted.

5.183.1 Detailed Description

Graphics item for drawing a circular or polygonal [Point](#).

In this class, lines are drawn twice: 1) As nonzero-width lines so user can have thick, and highly visible, points 2) As a 'shadow' with zero-width lines since these always appear even when zooming results in some pixel rows/columns disappearing. This dual-line approach is better than using `QGraphicsItem::ItemIgnoresTransformations` to prevent horrible aliasing problems, since that approach involves complicated transformation matrix manipulations.

Layering is used for the single graphics item contained by this class. External code only has to deal with this single class, and there is no multiple inheritance involved. If inheritance was used, we would have one class based on `QGraphicsEllipseItem` and another on `QGraphicsPolygonItem`, so having a single class (for the convenience of the external code) would involve multiple inheritance (of those two classes). With the inheritance approach, using just the methods supplied by `QGraphicsItem` would be inadequate.

Definition at line 42 of file `GraphicsPoint.h`.

The documentation for this class was generated from the following files:

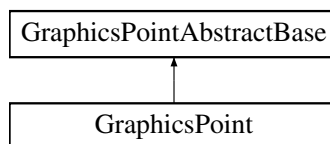
- `Graphics/GraphicsPoint.h`
- `Graphics/GraphicsPoint.cpp`

5.184 GraphicsPointAbstractBase Class Reference

Base class for adding identifiers to graphics items that represent Points.

```
#include <GraphicsPointAbstractBase.h>
```

Inheritance diagram for `GraphicsPointAbstractBase`:



Public Member Functions

- [GraphicsPointAbstractBase](#) ()
Single constructor.

5.184.1 Detailed Description

Base class for adding identifiers to graphics items that represent Points.

Identifiers are needed to distinguish which nodes are selected from those that are not selected. Each identifier is stored as a data item in `QGraphicsItem`.

This abstract base class no longer does anything.

Definition at line 18 of file `GraphicsPointAbstractBase.h`.

The documentation for this class was generated from the following files:

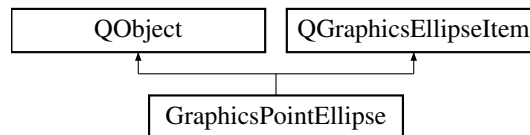
- `Graphics/GraphicsPointAbstractBase.h`
- `Graphics/GraphicsPointAbstractBase.cpp`

5.185 GraphicsPointEllipse Class Reference

This class add event handling to QGraphicsEllipseItem.

```
#include <GraphicsPointEllipse.h>
```

Inheritance diagram for GraphicsPointEllipse:



Signals

- void [signalPointHoverEnter](#) (QString)
Signal for geometry window to highlight the current point upon hover enter.
- void [signalPointHoverLeave](#) (QString)
Signal for geometry window to unhighlight the current point upon hover leave.

Public Member Functions

- [GraphicsPointEllipse](#) ([GraphicsPoint](#) &graphicsPoint, const QRect &rect)
Single constructor.
- QVariant [itemChange](#) (GraphicsItemChange change, const QVariant &value)
Intercept moves by dragging so moved items can be identified. This replaces unreliable hit tests.
- virtual void [hoverEnterEvent](#) (QGraphicsSceneHoverEvent *event)
Accept hover so point can be highlighted when cursor is over it as a guide to user.
- virtual void [hoverLeaveEvent](#) (QGraphicsSceneHoverEvent *event)
Unhighlight this point.
- void [setRadius](#) (int radius)
Update the radius.
- void [setShadow](#) ([GraphicsPointEllipse](#) *shadow)
Bind this graphics item to its shadow.

5.185.1 Detailed Description

This class add event handling to QGraphicsEllipseItem.

Definition at line 17 of file GraphicsPointEllipse.h.

The documentation for this class was generated from the following files:

- Graphics/GraphicsPointEllipse.h
- Graphics/GraphicsPointEllipse.cpp

5.186 GraphicsPointFactory Class Reference

Factor for generating [GraphicsPointAbstractBase](#) class objects.

```
#include <GraphicsPointFactory.h>
```

Public Member Functions

- [GraphicsPointFactory](#) ()
Single constructor.
- [GraphicsPoint](#) * [createPoint](#) (QGraphicsScene &scene, const QString &identifier, const QPointF &posScreen, const [PointStyle](#) &pointStyle, [GeometryWindow](#) *geometryWindow)
Create circle or polygon point according to the [PointStyle](#).

5.186.1 Detailed Description

Factor for generating [GraphicsPointAbstractBase](#) class objects.

Definition at line 19 of file GraphicsPointFactory.h.

The documentation for this class was generated from the following files:

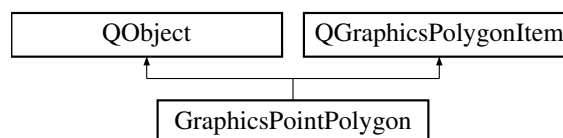
- Graphics/GraphicsPointFactory.h
- Graphics/GraphicsPointFactory.cpp

5.187 GraphicsPointPolygon Class Reference

This class add event handling to QGraphicsPolygonItem.

```
#include <GraphicsPointPolygon.h>
```

Inheritance diagram for GraphicsPointPolygon:



Signals

- void [signalPointHoverEnter](#) (QString)
Signal for geometry window to highlight the current point upon hover enter.
- void [signalPointHoverLeave](#) (QString)
Signal for geometry window to unhighlight the current point upon hover leave.

Public Member Functions

- [GraphicsPointPolygon](#) ([GraphicsPoint](#) &graphicsPoint, const [QPolygonF](#) &polygon)
Single constructor.
- [QVariant itemChange](#) ([GraphicsItemChange](#) change, const [QVariant](#) &value)
Intercept moves by dragging so moved items can be identified. This replaces unreliable hit tests.
- virtual void [hoverEnterEvent](#) ([QGraphicsSceneHoverEvent](#) *event)
Accept hover so point can be highlighted when cursor is over it as a guide to user.
- virtual void [hoverLeaveEvent](#) ([QGraphicsSceneHoverEvent](#) *event)
Unhighlight this point.
- void [setRadius](#) (int radius)
Update the radius.
- void [setShadow](#) ([GraphicsPointPolygon](#) *shadow)
Bind this graphics item to its shadow.

5.187.1 Detailed Description

This class add event handling to [QGraphicsPolygonItem](#).

Definition at line 17 of file [GraphicsPointPolygon.h](#).

The documentation for this class was generated from the following files:

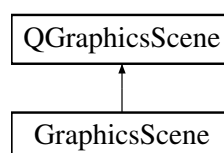
- [Graphics/GraphicsPointPolygon.h](#)
- [Graphics/GraphicsPointPolygon.cpp](#)

5.188 GraphicsScene Class Reference

Add point and line handling to generic [QGraphicsScene](#).

```
#include <GraphicsScene.h>
```

Inheritance diagram for [GraphicsScene](#):



Public Member Functions

- [GraphicsScene](#) ([MainWindow](#) *mainWindow)
Single constructor.
- void [addTemporaryPoint](#) (const QString &identifier, [GraphicsPoint](#) *point)
Add one temporary point to m_graphicsLinesForCurves. Non-temporary points are handled by the updateLine↔Membership functions.
- [GraphicsPoint](#) * [createPoint](#) (const QString &identifier, const [PointStyle](#) &pointStyle, const QPointF &pos↔Screen, [GeometryWindow](#) *geometryWindow)
Create one QGraphicsItem-based object that represents one [Point](#). It is NOT added to m_graphicsLinesForCurves (see [addPoint](#))
- void [hideAllItemsExceptImage](#) ()
Hide all graphics items, except background image, in preparation for preview during IMPORT_TYPE_ADVANCED.
- QStringList [positionHasChangedPointIdentifiers](#) () const
Return a list of identifiers for the points that have moved since the last call to [resetPositionHasChanged](#).
- void [printStream](#) (QString indentation, QTextStream &str)
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [removePoint](#) (const QString &identifier)
Remove specified point. This aborts if the point does not exist.
- void [removeTemporaryPointIfExists](#) ()
Remove temporary point if it exists.
- void [resetOnLoad](#) ()
Reset, when loading a document after the first, to same state that first document was at when loaded.
- void [resetPositionHasChangedFlags](#) ()
Reset positionHasChanged flag for all items. Typically this is done as part of mousePressEvent.
- void [showCurves](#) (bool show, bool showAll=false, const QString &curveName="")
Show or hide all Curves (if showAll is true) or just the selected [Curve](#) (if showAll is false);.
- void [updateAfterCommand](#) ([CmdMediator](#) &cmdMediator, double highlightOpacity, [GeometryWindow](#) *geometryWindow)
Update the Points and their Curves after executing a command.
- void [updateCurveStyles](#) (const [CurveStyles](#) &modelCurveStyles)
Update curve styles after settings changed.
- void [updateGraphicsLinesToMatchGraphicsPoints](#) (const [CurveStyles](#) &modelCurveStyles, const [Transformation](#) &transformation)
A mouse move has just occurred so move the selected points, since they were dragged.

5.188.1 Detailed Description

Add point and line handling to generic QGraphicsScene.

The primary tasks are:

1. update the graphics items to stay in sync with the explicit Points in the [Document](#)
2. update the graphics items to stay in sync with the implicit lines between the Points, according to [Document](#) settings

This class stores points and lines as QGraphicsItems, but also maintains identifier-to-QGraphicsItems mappings to the points and lines are accessible for updates (like when dragging points around and we need to update the attached lines).

Definition at line 32 of file GraphicsScene.h.

5.188.2 Member Function Documentation

5.188.2.1 void GraphicsScene::removeTemporaryPointIfExists ()

Remove temporary point if it exists.

Temporary point handling is so complicated that this method quietly allows redundant calls to this method, without complaining that the point has already been removed when called again

Definition at line 168 of file GraphicsScene.cpp.

5.188.2.2 void GraphicsScene::updateAfterCommand (CmdMediator & cmdMediator, double highlightOpacity, GeometryWindow * geometryWindow)

Update the Points and their Curves after executing a command.

After a mouse drag, the lines are already updated and updating would be done on out of date information (since that would be brought up to date by the NEXT command)

Definition at line 246 of file GraphicsScene.cpp.

5.188.2.3 void GraphicsScene::updateGraphicsLinesToMatchGraphicsPoints (const CurveStyles & modelCurveStyles, const Transformation & transformation)

A mouse move has just occurred so move the selected points, since they were dragged.

The transformation is needed so the screen coordinates can be converted to graph coordinates when updating point ordinals

Definition at line 281 of file GraphicsScene.cpp.

The documentation for this class was generated from the following files:

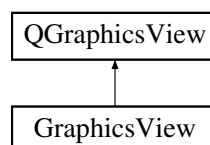
- Graphics/GraphicsScene.h
- Graphics/GraphicsScene.cpp

5.189 GraphicsView Class Reference

QGraphicsView class with event handling added. Typically the events are sent to the active digitizing state.

```
#include <GraphicsView.h>
```

Inheritance diagram for GraphicsView:



Signals

- void [signalContextMenuEventAxis](#) (QString pointIdentifier)
Send right click on axis point to [MainWindow](#) for editing.
- void [signalContextMenuEventGraph](#) (QStringList pointIdentifiers)
Send right click on graph point(s) to [MainWindow](#) for editing.
- void [signalDraggedDigFile](#) (QString)
Send dragged dig file to [MainWindow](#) for import. This comes from dragging an engage dig file.
- void [signalDraggedImage](#) (QImage)
Send dragged image to [MainWindow](#) for import. This typically comes from dragging a file.
- void [signalDraggedImageUrl](#) (QUrl)
Send dragged url to [MainWindow](#) for import. This typically comes from dragging an image from a browser.
- void [signalKeyPress](#) (Qt::Key, bool atLeastOneSelectedItem)
Send keypress to [MainWindow](#) for eventual processing by [DigitizeStateAbstractBase](#) subclasses.
- void [signalMouseMove](#) (QPointF)
Send mouse move to [MainWindow](#) for eventual display of cursor coordinates in [StatusBar](#).
- void [signalMousePress](#) (QPointF)
Send mouse press to [MainWindow](#) for creating one or more Points.
- void [signalMouseRelease](#) (QPointF)
Send mouse release to [MainWindow](#) for moving Points.
- void [signalViewZoomIn](#) ()
Send wheel event to [MainWindow](#) for zooming in.
- void [signalViewZoomOut](#) ()
Send wheel event to [MainWindow](#) for zooming out.

Public Member Functions

- [GraphicsView](#) (QGraphicsScene *scene, [MainWindow](#) &mainWindow)
Single constructor.
- virtual void [contextMenuEvent](#) (QContextMenuEvent *event)
Intercept context event to support point editing.
- virtual void [dragEnterEvent](#) (QDragEnterEvent *event)
Intercept mouse drag event to support drag-and-drop.
- virtual void [dragMoveEvent](#) (QDragMoveEvent *event)
Intercept mouse move event to support drag-and-drop.
- virtual void [dropEvent](#) (QDropEvent *event)
Intercept mouse drop event to support drag-and-drop. This initiates asynchronous loading of the dragged image.
- virtual void [keyPressEvent](#) (QKeyEvent *event)
Intercept key press events to handle left/right/up/down moving.
- virtual void [mouseMoveEvent](#) (QMouseEvent *event)
Intercept mouse move events to populate the current cursor position in [StatusBar](#).
- virtual void [mousePressEvent](#) (QMouseEvent *event)
Intercept mouse press events to create one or more Points.
- virtual void [mouseReleaseEvent](#) (QMouseEvent *event)
Intercept mouse release events to move one or more Points.
- virtual void [wheelEvent](#) (QWheelEvent *event)
Convert wheel events into zoom in/out.

5.189.1 Detailed Description

QGraphicsView class with event handling added. Typically the events are sent to the active digitizing state.

Definition at line 20 of file GraphicsView.h.

The documentation for this class was generated from the following files:

- Graphics/GraphicsView.h
- Graphics/GraphicsView.cpp

5.190 GridClassifier Class Reference

Classify the grid pattern in an original image.

```
#include <GridClassifier.h>
```

Public Member Functions

- [GridClassifier](#) ()
Single constructor.
- void [classify](#) (bool isGnuplot, const QPixmap &originalPixmap, const [Transformation](#) &transformation, int &countX, double &startX, double &stepX, int &countY, double &startY, double &stepY)
Classify the specified image, and return the most probably x and y grid settings.

5.190.1 Detailed Description

Classify the grid pattern in an original image.

This class uses the following tricks for faster performance:

1. FFT is used for "fast correlations" in frequency space rather than graph space
2. FFT initialization/shutdown housekeeping is done once
3. Rather than a combinatorial search of grid line start, step and count, we exploit the periodicity of the FFT to search start and step as the first step, and then as a separate second step we search count. In the first step, the periodicity means the repeating grid lines wrap around the end of the image back around to the start of the image - so the grid line count is not even relevant. In other words, the searches are START X STEP + COUNT rather than START X STEP X COUNT

Definition at line 26 of file GridClassifier.h.

The documentation for this class was generated from the following files:

- Grid/GridClassifier.h
- Grid/GridClassifier.cpp

5.191 GridHealer Class Reference

Class that 'heals' the curves after grid lines have been removed.

```
#include <GridHealer.h>
```

Public Member Functions

- [GridHealer](#) (const QImage &imageBefore, const [DocumentModelGridRemoval](#) &modelGridRemoval)
Single constructor.
- void [erasePixel](#) (int xCol, int yRow)
Remember that pixel was erased since it belongs to an grid line.
- void [heal](#) (QImage &imageToHeal)
Heal the broken curve lines by spanning the gaps across the newly-removed grid lines.

5.191.1 Detailed Description

Class that 'heals' the curves after grid lines have been removed.

Specifically, gaps that span the pixels in the removed grid lines are filled in, if they are less than some epsilon value

Definition at line 37 of file GridHealer.h.

5.191.2 Member Function Documentation

5.191.2.1 void GridHealer::erasePixel (int xCol, int yRow)

Remember that pixel was erased since it belongs to an grid line.

In the image, erasure correponds to a foreground pixel being changed to the background color

Definition at line 96 of file GridHealer.cpp.

The documentation for this class was generated from the following files:

- Grid/GridHealer.h
- Grid/GridHealer.cpp

5.192 GridInitializer Class Reference

This class initializes the count, start, step and stop parameters for one coordinate (either x/theta or y/range)

```
#include <GridInitializer.h>
```


Public Member Functions

- [GridInitializer](#) ()
Single constructor.
- int [computeCount](#) (bool linearAxis, double start, double stop, double step) const
Compute axis scale count from the other axis parameters.
- double [computeStart](#) (bool linearAxis, double stop, double step, int count) const
Compute axis scale start from the other axis parameters.
- double [computeStep](#) (bool linearAxis, double start, double stop, int count) const
Compute axis scale step from the other axis parameters.
- double [computeStop](#) (bool linearAxis, double start, double step, int count) const
Compute axis scale stop from the other axis parameters.
- [DocumentModelGridDisplay](#) [initializeWithNarrowCoverage](#) (const QRectF &boundingRectGraph, const [DocumentModelCoords](#) &modelCoords) const
Initialize given the boundaries of the graph coordinates. The output is useful for the [Checker](#) class.
- [DocumentModelGridDisplay](#) [initializeWithWidePolarCoverage](#) (const QRectF &boundingRectGraph, const [DocumentModelCoords](#) &modelCoords, const [Transformation](#) &transformation, const QSize &imageSize) const
Initialize given the boundaries of the graph coordinates, and then extra processing for polar coordinates:
 1. radial range expanded to cover the center (to remove hole at center) to the image corners (to guarantee coverage at corners of graph)
 2. angular range is expanded to cover the entire circle (so coverage is total for all directions)
- int [valuePower](#) (double value) const
*Compute power of 10 for input value, rounding down to nearest integer solution of $value \geq 10^{**}solution$.*

5.192.1 Detailed Description

This class initializes the count, start, step and stop parameters for one coordinate (either x/theta or y/range)

Definition at line 13 of file GridInitializer.h.

The documentation for this class was generated from the following files:

- Grid/GridInitializer.h
- Grid/GridInitializer.cpp

5.193 GridLine Class Reference

Single grid line drawn a straight or curved line.

```
#include <GridLine.h>
```

Public Member Functions

- [GridLine](#) ()
Default constructor for storage in containers.
- [GridLine](#) (const [GridLine](#) &other)
Copy constructor. This will assert if called since copying of pointer containers is problematic.
- [GridLine](#) & [operator=](#) ([GridLine](#) &other)
Assignment constructor. This will assert if called since copying of pointer containers is problematic.
- void [add](#) (QGraphicsItem *item)
Add graphics item which represents one segment of the line.
- void [setPen](#) (const QPen &pen)
Set the pen style.
- void [setVisible](#) (bool visible)
Set each grid line as visible or hidden.

5.193.1 Detailed Description

Single grid line drawn a straight or curved line.

This is expected to be composed of QGraphicsEllipseItem and QGraphicsLineItem objects

Definition at line 20 of file GridLine.h.

The documentation for this class was generated from the following files:

- Grid/GridLine.h
- Grid/GridLine.cpp

5.194 GridLineFactory Class Reference

Factory class for generating the points, composed of QGraphicsItem objects, along a [GridLine](#).

```
#include <GridLineFactory.h>
```

Public Member Functions

- [GridLineFactory](#) (QGraphicsScene &scene, const [DocumentModelCoords](#) &modelCoords)
Simple constructor for general use (i.e. not by [Checker](#))
- [GridLineFactory](#) (QGraphicsScene &scene, int pointRadius, const QList< [Point](#) > &pointsToIsolate, const [DocumentModelCoords](#) &modelCoords)
Constructor for use by [Checker](#), which has points that are isolated.
- [GridLine](#) * [createGridLine](#) (double xFrom, double yFrom, double xTo, double yTo, const [Transformation](#) &transformation)
Create grid line, either along constant X/theta or constant Y/radius side.
- void [createGridLinesForEvenlySpacedGrid](#) (const [DocumentModelGridDisplay](#) &modelGridDisplay, const [MainWindowModel](#) &modelMainWindow, const [Transformation](#) &transformation, [GridLines](#) &gridLines)
Create a rectangular (cartesian) or annular (polar) grid of evenly spaced grid lines.

5.194.1 Detailed Description

Factory class for generating the points, composed of QGraphicsItem objects, along a [GridLine](#).

For polar coordinates, the grid lines will appear as an annular segments.

For the [Checker](#) class, a set of Points can be specified which will be isolated by having grid lines stop at a specified distance (or radius) from each point

Definition at line 28 of file GridLineFactory.h.

5.194.2 Member Function Documentation

5.194.2.1 `GridLine * GridLineFactory::createGridLine (double xFrom, double yFrom, double xTo, double yTo, const Transformation & transformation)`

Create grid line, either along constant X/theta or constant Y/radius side.

Line goes from pointFromGraph to pointToGraph. If the coordinates are polar, we go clockwise from pointFrom↔Graph to pointToGraph (as set up by adjustPolarAngleRange).

Definition at line 72 of file GridLineFactory.cpp.

The documentation for this class was generated from the following files:

- Grid/GridLineFactory.h
- Grid/GridLineFactory.cpp

5.195 GridLineLimiter Class Reference

Limit the number of grid lines so a bad combination of start/step/stop value will not lead to extremely long delays when the step size is much too small for the start/stop values.

```
#include <GridLineLimiter.h>
```

Public Member Functions

- [GridLineLimiter](#) ()
Single constructor.
- double [limitedStepXTheta](#) (const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [DocumentModelGridDisplay](#) &modelGrid) const
Limit step value for x/theta coordinate. This is a noop if the maximum grid line limit in [MainWindowModel](#) is not exceeded.
- double [limitedStepYRange](#) (const [DocumentModelCoords](#) &modelCoords, const [MainWindowModel](#) &modelMainWindow, const [DocumentModelGridDisplay](#) &modelGrid) const
Limit step value for y/range coordinate. This is a noop if the maximum grid line limit in [MainWindowModel](#) is not exceeded.

5.195.1 Detailed Description

Limit the number of grid lines so a bad combination of start/step/stop value will not lead to extremely long delays when the step size is much too small for the start/stop values.

Definition at line 19 of file GridLineLimiter.h.

The documentation for this class was generated from the following files:

- Grid/GridLineLimiter.h
- Grid/GridLineLimiter.cpp

5.196 GridLines Class Reference

Container class for [GridLine](#) objects.

```
#include <GridLines.h>
```

Public Member Functions

- [GridLines](#) ()
Single constructor.
- void [add](#) ([GridLine](#) *gridLine)
Add specified grid line. Ownership of all allocated QGraphicsItems is passed to new [GridLine](#).
- void [clear](#) ()
Deallocate and remove all grid lines.
- void [setPen](#) (const QPen &pen)
Set the pen style of each grid line.
- void [setVisible](#) (bool visible)
Make all grid lines visible or hidden.

5.196.1 Detailed Description

Container class for [GridLine](#) objects.

Definition at line 18 of file GridLines.h.

The documentation for this class was generated from the following files:

- Grid/GridLines.h
- Grid/GridLines.cpp

5.197 GridRemoval Class Reference

Strategy class for grid removal.

```
#include <GridRemoval.h>
```

Public Member Functions

- [GridRemoval](#) ()
Single constructor.
- QPixmap [remove](#) (const [Transformation](#) &transformation, const [DocumentModelGridRemoval](#) &modelGridRemoval, const QImage &imageBefore)
Process QImage into QPixmap, removing the grid lines.

5.197.1 Detailed Description

Strategy class for grid removal.

Definition at line 19 of file GridRemoval.h.

The documentation for this class was generated from the following files:

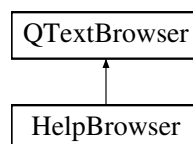
- Grid/GridRemoval.h
- Grid/GridRemoval.cpp

5.198 HelpBrowser Class Reference

Text browser with resource loading enhanced for use as help text browser.

```
#include <HelpBrowser.h>
```

Inheritance diagram for HelpBrowser:



Public Member Functions

- [HelpBrowser](#) (QHelpEngine *engine, QWidget *parent=0)
Single constructor.
- QVariant [loadResource](#) (int type, const QUrl &url)
Load resources.

5.198.1 Detailed Description

Text browser with resource loading enhanced for use as help text browser.

Definition at line 15 of file HelpBrowser.h.

The documentation for this class was generated from the following files:

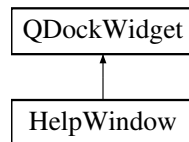
- Help/HelpBrowser.h
- Help/HelpBrowser.cpp

5.199 HelpWindow Class Reference

Dockable help window.

```
#include <HelpWindow.h>
```

Inheritance diagram for HelpWindow:



Public Member Functions

- [HelpWindow](#) (QWidget *parent)
Single constructor.

5.199.1 Detailed Description

Dockable help window.

Despite a lot of work trying to work with the OSX sandbox, support for the sandbox was never completed since QHelpEngine requires WRITE-access to the collection file. Even trying to create a temporary directory does not work since copying would involve QHelpEngine::copyCollectionFile which copies from the CURRENT collection file (versus just some arbitrary file name)

Definition at line 16 of file HelpWindow.h.

The documentation for this class was generated from the following files:

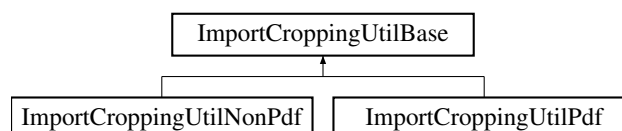
- Help/HelpWindow.h
- Help/HelpWindow.cpp

5.200 ImportCroppingUtilBase Class Reference

Utility class for import cropping options.

```
#include <ImportCroppingUtilBase.h>
```

Inheritance diagram for ImportCroppingUtilBase:



Public Member Functions

- [ImportCroppingUtilBase](#) ()

Single constructor.

Static Public Member Functions

- static QString [importCroppingToString](#) (ImportCropping importCropping)

Option as string for display to user.

5.200.1 Detailed Description

Utility class for import cropping options.

Default option is oldest, and simplest, behavior, which is no cropping.

A complication is that a dialog for cropping is not wanted during batch-mode regression testing, so this class and its subclasses offer methods for overriding the current setting during regression testing

Definition at line 17 of file ImportCroppingUtilBase.h.

The documentation for this class was generated from the following files:

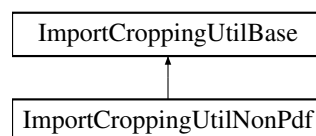
- Import/ImportCroppingUtilBase.h
- Import/ImportCroppingUtilBase.cpp

5.201 ImportCroppingUtilNonPdf Class Reference

Import of non-pdf files.

```
#include <ImportCroppingUtilNonPdf.h>
```

Inheritance diagram for ImportCroppingUtilNonPdf:



Public Member Functions

- [ImportCroppingUtilNonPdf](#) ()

Single constructor.

- bool [applyImportCropping](#) (bool isRegression, ImportCropping importCropping) const

Skip cropping dialog during regression testing, otherwise crop if it is always turned on.

Additional Inherited Members

5.201.1 Detailed Description

Import of non-pdf files.

Definition at line 14 of file ImportCroppingUtilNonPdf.h.

The documentation for this class was generated from the following files:

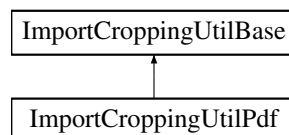
- Import/ImportCroppingUtilNonPdf.h
- Import/ImportCroppingUtilNonPdf.cpp

5.202 ImportCroppingUtilPdf Class Reference

Import of pdf files.

```
#include <ImportCroppingUtilPdf.h>
```

Inheritance diagram for ImportCroppingUtilPdf:



Public Member Functions

- [ImportCroppingUtilPdf \(\)](#)
Single constructor.
- [bool applyImportCropping](#) (bool isRegression, const QString &fileName, ImportCropping importCropping, Poppler::Document *&document) const
For pdf files, skip cropping dialog during regression testing, otherwise crop if it is always turned on or if there are multiple pages.

Additional Inherited Members

5.202.1 Detailed Description

Import of pdf files.

Definition at line 19 of file ImportCroppingUtilPdf.h.

5.202.2 Member Function Documentation

5.202.2.1 `bool ImportCroppingUtilPdf::applyImportCropping (bool isRegression, const QString & fileName, ImportCropping importCropping, Poppler::Document *& document) const`

For pdf files, skip cropping dialog during regression testing, otherwise crop if it is always turned on or if there are multiple pages.

For speed, the [Document](#) is returned if cropping is to be performed so the file needs to be read only once

Definition at line 17 of file `ImportCroppingUtilPdf.cpp`.

The documentation for this class was generated from the following files:

- `Import/ImportCroppingUtilPdf.h`
- `Import/ImportCroppingUtilPdf.cpp`

5.203 Jpeg2000 Class Reference

Wrapper around OpenJPEG library, in C, for opening jpeg2000 files.

```
#include <Jpeg2000.h>
```

Public Member Functions

- [Jpeg2000](#) ()
Single constructor.
- `bool load (const QString &filename, QImage &image) const`
Load image from jpeg2000 file.
- `QStringList supportedImageWildcards () const`
List the supported jpeg2000 file extensions, for filtering import files.

5.203.1 Detailed Description

Wrapper around OpenJPEG library, in C, for opening jpeg2000 files.

Definition at line 26 of file `Jpeg2000.h`.

The documentation for this class was generated from the following files:

- `Jpeg2000/Jpeg2000.h`
- `Jpeg2000/Jpeg2000.cpp`

5.204 LineStyle Class Reference

Details for a specific Line.

```
#include <LineStyle.h>
```

Public Member Functions

- [LineStyle](#) ()
Default constructor only for use when this class is being stored by a container that requires the default constructor.
- [LineStyle](#) (unsigned int [width](#), ColorPalette [paletteColor](#), CurveConnectAs [curveConnectAs](#))
Normal constructor.
- [LineStyle](#) (const [LineStyle](#) &other)
Copy constructor.
- [LineStyle](#) & [operator=](#) (const [LineStyle](#) &other)
Assignment operator.
- CurveConnectAs [curveConnectAs](#) () const
Get method for connect type.
- void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml. Returns the curve name.
- ColorPalette [paletteColor](#) () const
Line color.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize to stream.
- void [setCurveConnectAs](#) (CurveConnectAs [curveConnectAs](#))
Set connect as.
- void [setPaletteColor](#) (ColorPalette [paletteColor](#))
Set method for line color.
- void [setWidth](#) (int [width](#))
Set width of line.
- unsigned int [width](#) () const
Width of line.

Static Public Member Functions

- static [LineStyle](#) [defaultAxesCurve](#) ()
Initial default for axes curve.
- static [LineStyle](#) [defaultGraphCurve](#) (int index)
Initial default for index'th graph curve.

5.204.1 Detailed Description

Details for a specific Line.

Definition at line 19 of file LineStyle.h.

The documentation for this class was generated from the following files:

- Line/LineStyle.h
- Line/LineStyle.cpp

5.205 LoadFileInfo Class Reference

Returns information about files.

```
#include <LoadFileInfo.h>
```

Public Member Functions

- [LoadFileInfo](#) ()
Single constructor.
- bool [loadsAsDigFile](#) (const QString &urlString) const
Returns true if specified file name can be loaded as a DIG file.

5.205.1 Detailed Description

Returns information about files.

Definition at line 13 of file LoadFileInfo.h.

The documentation for this class was generated from the following files:

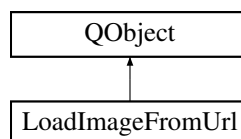
- Load/LoadFileInfo.h
- Load/LoadFileInfo.cpp

5.206 LoadImageFromUrl Class Reference

Load QImage from url. This is trivial for a file, but requires an asynchronous download step for http urls.

```
#include <LoadImageFromUrl.h>
```

Inheritance diagram for LoadImageFromUrl:



Signals

- void [signalImportImage](#) (QString, QImage)
Send the imported image to [MainWindow](#). This completes the asynchronous loading of the image.

Public Member Functions

- [LoadImageFromUrl](#) ([MainWindow](#) &mainWindow)
Single constructor.
- void [startLoadImage](#) (const QUrl &url)
Start the asynchronous loading of an image from the specified url.

5.206.1 Detailed Description

Load QImage from url. This is trivial for a file, but requires an asynchronous download step for http urls.

Definition at line 20 of file LoadImageFromUrl.h.

The documentation for this class was generated from the following files:

- Load/LoadImageFromUrl.h
- Load/LoadImageFromUrl.cpp

5.207 LoggerUpload Class Reference

Upload logging information to website for developer support.

```
#include <LoggerUpload.h>
```

Public Member Functions

- [LoggerUpload \(\)](#)
Single constructor.

Static Public Member Functions

- static void [bindToMainWindow](#) (const [MainWindow](#) *mainWindow)
Bind to [MainWindow](#) so this class can access the command stack.
- static void [loggerAssert](#) (const char *condition, const char *file, int line) NO_RETURN_VALUE
Smart equivalent to standard assert method and Q_ASSERT (in qglobal.h).
- static void [loggerCheckPtr](#) (const char *pointer, const char *file, int line) NO_RETURN_VALUE
Smart equivalent to Q_CHECK_PTR (in qglobal.h). Similar to loggerAssert but for checking newly-allocated pointers.

5.207.1 Detailed Description

Upload logging information to website for developer support.

Definition at line 21 of file LoggerUpload.h.

5.207.2 Member Function Documentation

5.207.2.1 void [LoggerUpload::loggerAssert](#) (const char * *condition*, const char * *file*, int *line*) [static]

Smart equivalent to standard assert method and Q_ASSERT (in qglobal.h).

Upon error, an upload is proposed. This is static for easy access from anywhere else in the application

Definition at line 22 of file LoggerUpload.cpp.

The documentation for this class was generated from the following files:

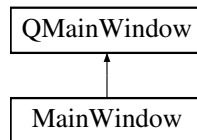
- Logger/LoggerUpload.h
- Logger/LoggerUpload.cpp

5.208 MainWindow Class Reference

Main window consisting of menu, graphics scene, status bar and optional toolbars as a Single [Document](#) Interface.

```
#include <MainWindow.h>
```

Inheritance diagram for MainWindow:



Signals

- void [signalZoom](#) (int)
Send zoom selection, picked from menu or keystroke, to [StatusBar](#).

Public Member Functions

- [MainWindow](#) (const QString &errorReportFile, const QString &fileCmdScriptFile, bool isRegressionTest, bool [isGnuplot](#), bool isReset, QStringList loadStartupFiles, QWidget *parent=0)
Single constructor.
- void [cmdFileClose](#) ()
Close file. This is called from a file script command.
- void [cmdFileExport](#) (const QString &fileName)
Export file. This is called from a file script command.
- void [cmdFileImport](#) (const QString &fileName)
Import file. This is called from a file script command.
- void [cmdFileOpen](#) (const QString &fileName)
Open file. This is called from a file script command.
- [CmdMediator](#) * [cmdMediator](#) ()
Accessor for commands to process the [Document](#).
- virtual bool [eventFilter](#) (QObject *, QEvent *)
Catch secret keypresses.
- QImage [imageFiltered](#) () const
Background image that has been filtered for the current curve. This asserts if a curve-specific image is not being shown.
- bool [isGnuplot](#) () const
Get method for gnuplot flag.
- [MainWindowModel](#) [modelMainWindow](#) () const
Get method for main window model.
- void [resizeEvent](#) (QResizeEvent *event)
Intercept resize event so graphics scene can be appropriately resized when in Fill mode.
- void [saveErrorReportFileAndExit](#) (const char *comment, const char *file, int line, const char *context) const
Save error report and exit.
- [GraphicsScene](#) & [scene](#) ()
Scene container for the QImage and QGraphicsItems.
- BackgroundImage [selectOriginal](#) (BackgroundImage backgroundImage)

- Make original background visible, for [DigitizeStateColorPicker](#).
- `QString selectedGraphCurve () const`
Curve name that is currently selected in m_cmbCurve.
- `virtual void showEvent (QShowEvent *)`
Processing performed after gui becomes available.
- `void showTemporaryMessage (const QString &temporaryMessage)`
Show temporary message in status bar.
- `Transformation transformation () const`
Return read-only copy of transformation.
- `bool transformIsDefined () const`
Return true if all three axis points have been defined.
- `void updateAfterCommand ()`
See [GraphicsScene::updateAfterCommand](#).
- `void updateAfterMouseRelease ()`
Call MainWindow::updateControls (which is private) after the very specific case - a mouse press/release.
- `void updateCoordSystem (CoordSystemIndex coordSystemIndex)`
Select a different [CoordSystem](#).
- `void updateDigitizeStateIfSoftwareTriggered (DigitizeState digitizeState)`
After software-triggered state transition, this method manually triggers the action as if user had clicked on digitize button.
- `void updateGraphicsLinesToMatchGraphicsPoints ()`
Update the graphics lines so they follow the graphics points, after a drag, addition, removal, and such.
- `void updateSettingsAxesChecker (const DocumentModelAxesChecker &modelAxesChecker)`
Update with new axes indicator properties.
- `void updateSettingsColorFilter (const DocumentModelColorFilter &modelColorFilter)`
Update with new color filter properties.
- `void updateSettingsCoords (const DocumentModelCoords &modelCoords)`
Update with new coordinate properties.
- `void updateSettingsCurveAddRemove (const CurvesGraphs &curvesGraphs)`
Update with new curves.
- `void updateSettingsCurveStyles (const CurveStyles &modelCurveStyles)`
Update with new curve styles.
- `void updateSettingsDigitizeCurve (const DocumentModelDigitizeCurve &modelDigitizeCurve)`
Update with new curve digitization styles.
- `void updateSettingsExportFormat (const DocumentModelExportFormat &modelExport)`
Update with new export properties.
- `void updateSettingsGeneral (const DocumentModelGeneral &modelGeneral)`
Update with new general properties.
- `void updateSettingsGridDisplay (const DocumentModelGridDisplay &modelGridDisplay)`
Update with new grid display properties.
- `void updateSettingsGridRemoval (const DocumentModelGridRemoval &modelGridRemoval)`
Update with new grid removal properties.
- `void updateSettingsMainWindow (const MainWindowModel &modelMainWindow)`
Update with new main window properties.
- `void updateSettingsPointMatch (const DocumentModelPointMatch &modelPointMatch)`
Update with new point match properties.
- `void updateSettingsSegments (const DocumentModelSegments &modelSegments)`
Update with new segments properties.
- `void updateViewsOfSettings (const QString &activeCurve)`
Update curve-specific view of settings. Private version gets active curve name from [DigitizeStateContext](#).
- `GraphicsView & view ()`
View for the QImage and QGraphicsItems, without const.
- `const GraphicsView & view () const`
View for the QImage and QGraphicsItems, without const.

5.208.1 Detailed Description

Main window consisting of menu, graphics scene, status bar and optional toolbars as a Single [Document](#) Interface.

Definition at line 83 of file MainWindow.h.

5.208.2 Constructor & Destructor Documentation

5.208.2.1 `MainWindow::MainWindow (const QString & errorReportFile, const QString & fileCmdScriptFile, bool isRegressionTest, bool isGnuplot, bool isReset, QStringList loadStartupFiles, QWidget * parent = 0)`

Single constructor.

Parameters

<i>errorReportFile</i>	Optional error report file to be read at startup. Empty if unused. Incompatible with <i>fileCmdScript</i>
<i>fileCmdScriptFile</i>	Optional file command script file to be read at startup. Empty if unused. Incompatible with <i>errorReportFile</i>
<i>isRegressionTest</i>	True if <i>errorReportFile</i> or <i>fileCmdScript</i> is for regression testing, in which case it is executed and the program exits
<i>isGnuplot</i>	True if diagnostic gnuplot files are generated for math-intense sections of the code. Used for development and debugging
<i>isReset</i>	True to reset all settings that would otherwise be restored from the previous execution of Engauge
<i>loadStartupFiles</i>	Zero or more Engauge document files to load at startup. A separate instance of Engauge is created for each file
<i>parent</i>	Optional parent widget for this widget

Definition at line 127 of file MainWindow.cpp.

5.208.3 Member Function Documentation

5.208.3.1 `BackgroundImage MainWindow::selectOriginal (BackgroundImage backgroundImage)`

Make original background visible, for [DigitizeStateColorPicker](#).

This returns the previous background state for restoring when state finishes

Definition at line 2296 of file MainWindow.cpp.

5.208.3.2 `void MainWindow::updateGraphicsLinesToMatchGraphicsPoints ()`

Update the graphics lines so they follow the graphics points, after a drag, addition, removal, and such.

The points in the [Document](#) may (and probably are) out of date with respect to the graphics points

Definition at line 4262 of file MainWindow.cpp.

The documentation for this class was generated from the following files:

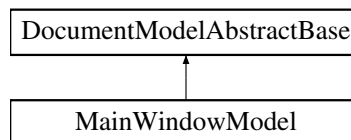
- main/MainWindow.h
- main/MainWindow.cpp

5.209 MainWindowModel Class Reference

Model for [DlgSettingsMainWindow](#).

```
#include <MainWindowModel.h>
```

Inheritance diagram for MainWindowModel:



Public Member Functions

- [MainWindowModel](#) ()
Default constructor.
- [MainWindowModel](#) (const [MainWindowModel](#) &other)
Copy constructor.
- [MainWindowModel](#) & operator= (const [MainWindowModel](#) &other)
Assignment constructor.
- virtual void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml.
- double [highlightOpacity](#) () const
Get method for highlight opacity.
- ImportCropping [importCropping](#) () const
Get method for import cropping.
- QLocale [locale](#) () const
Get method for locale.
- MainTitleBarFormat [mainTitleBarFormat](#) () const
Get method for [MainWindow](#) titlebar filename format.
- int [maximumGridLines](#) () const
Maximum number of grid lines.
- int [pdfResolution](#) () const
Get method for resolution of imported PDF files, in dots per inch.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- virtual void [saveXml](#) (QXmlStreamWriter &writer) const
Save entire model as xml into stream.
- void [setHighlightOpacity](#) (double [highlightOpacity](#))
Set method for highlight opacity.
- void [setLocale](#) (QLocale::Language language, QLocale::Country country)
Set method for locale given attributes.
- void [setLocale](#) (const QLocale &locale)
Set method for locale given locale object.
- void [setImportCropping](#) (ImportCropping [importCropping](#))
Set method for import cropping.
- void [setMainTitleBarFormat](#) (MainTitleBarFormat [mainTitleBarFormat](#))
Set method for [MainWindow](#) titlebar filename format.

- void [setMaximumGridLines](#) (int [maximumGridLines](#))
Set method for maximum number of grid lines.
- void [setPdfResolution](#) (int resolution)
Set method for resolution of imported PDF files, in dots per inch.
- void [setZoomControl](#) (ZoomControl [zoomControl](#))
Set method for zoom control.
- void [setZoomFactorInitial](#) (ZoomFactorInitial [zoomFactorInitial](#))
Set method for initial zoom factor.
- ZoomControl [zoomControl](#) () const
Get method for zoom control.
- ZoomFactorInitial [zoomFactorInitial](#) () const
Get method for initial zoom factor.

Additional Inherited Members

5.209.1 Detailed Description

Model for [DlgSettingsMainWindow](#).

Unlike the other models (DocumentModel*) this data is not saved and loaded within the document, so no xml or working with the [Document](#) class is involved. Also, there is no associated Cmd. Instead, the settings are saved using QSettings. Method involving xml/Document (from [DocumentModelAbstractBase](#)) are stubbed out

Definition at line 24 of file MainWindowModel.h.

The documentation for this class was generated from the following files:

- main/MainWindowModel.h
- main/MainWindowModel.cpp

5.210 MigrateToVersion6 Class Reference

Converts old (=pre version 6) enums to new (=version 6) enums, for reading of old document files.

```
#include <MigrateToVersion6.h>
```

Public Member Functions

- [MigrateToVersion6](#) ()
Single constructor.
- ColorPalette [colorPalette](#) (int preVersion6) const
Color from color palette.
- CurveConnectAs [curveConnectAs](#) (int preVersion6) const
Line drawn between points.
- PointShape [pointShape](#) (int preVersion6) const
Point shape.
- int [pointRadius](#) (int preVersion6) const
Point radius.

5.210.1 Detailed Description

Converts old (=pre version 6) enums to new (=version 6) enums, for reading of old document files.

Definition at line 15 of file MigrateToVersion6.h.

The documentation for this class was generated from the following files:

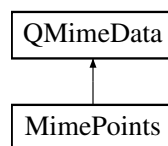
- util/MigrateToVersion6.h
- util/MigrateToVersion6.cpp

5.211 MimePoints Class Reference

Custom mime type for separate treatment of graph coordinates and, when there is no transform, points coordinates.

```
#include <MimePoints.h>
```

Inheritance diagram for MimePoints:



Public Member Functions

- [MimePoints](#) ()
Default constructor. Initial contents are overwritten by other constructors.
- [MimePoints](#) (const QString &[csvGraph](#), const QString &[htmlGraph](#))
Constructor when graph coordinates are available because the transformation is defined.
- [MimePoints](#) (const QString &[csvPoints](#))
Constructor when transformation is not defined. This data is not meant to leave this application.
- [MimePoints](#) & [operator=](#) (const [MimePoints](#) &other)
Assignment operator.
- virtual [~MimePoints](#) ()
Destructor.
- QString [csvGraph](#) () const
Get method for csvGraph.
- QString [csvPoints](#) () const
Get method for csvPoints.
- virtual QStringList [formats](#) () const
Available formats, which depend on whether or not the transform is defined.
- QString [htmlGraph](#) () const
Get method for htmlGraph.

Protected Member Functions

- virtual QVariant [retrieveData](#) (const QString &format, QVariant::Type preferredType) const
Returns a variant with the data for the specified format.

5.211.1 Detailed Description

Custom mime type for separate treatment of graph coordinates and, when there is no transform, points coordinates.

Definition at line 15 of file MimePoints.h.

The documentation for this class was generated from the following files:

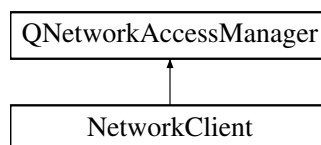
- Mime/MimePoints.h
- Mime/MimePoints.cpp

5.212 NetworkClient Class Reference

Client for interacting with Engauge server.

```
#include <NetworkClient.h>
```

Inheritance diagram for NetworkClient:



Public Slots

- void [slotFinished](#) (QNetworkReply *)
Cleanup after response is returned.

Public Member Functions

- [NetworkClient](#) (QObject *parent)
Single constructor.
- void [uploadErrorReport](#) (const QString &report)
Upload the error report asynchronously.

5.212.1 Detailed Description

Client for interacting with Engauge server.

Definition at line 16 of file NetworkClient.h.

The documentation for this class was generated from the following files:

- Network/NetworkClient.h
- Network/NetworkClient.cpp

5.213 NonPdf Class Reference

Wrapper around the QImage class for read and importing non-PDF files.

```
#include <NonPdf.h>
```

Public Member Functions

- [NonPdf](#) ()
Single constructor.
- [NonPdfReturn](#) [load](#) (const QString &fileName, QImage &image, ImportCropping importCropping, bool is↵
ErrorReportRegressionTest) const
Try to load the specified file. Success is indicated in the function return value.

5.213.1 Detailed Description

Wrapper around the QImage class for read and importing non-PDF files.

Definition at line 26 of file NonPdf.h.

The documentation for this class was generated from the following files:

- NonPdf/NonPdf.h
- NonPdf/NonPdf.cpp

5.214 NonPdfCropping Class Reference

This class shows a frame around the selected portion of the import preview window.

```
#include <NonPdfCropping.h>
```

Public Member Functions

- [NonPdfCropping](#) (QGraphicsScene &scene, [ViewPreview](#) &view)
Single constructor.
- [QRectF](#) [frameRect](#) () const
Frame rectangle selected by user.
- void [moveBL](#) (const QPointF &newPos, const QPointF &oldPos)
Bottom left corner handle was moved.
- void [moveBR](#) (const QPointF &newPos, const QPointF &oldPos)
Bottom right corner handle was moved.
- void [moveTL](#) (const QPointF &newPos, const QPointF &oldPos)
Top left corner handle was moved.
- void [moveTR](#) (const QPointF &newPos, const QPointF &oldPos)
Top right corner handle was moved.
- [QSize](#) [windowSize](#) () const
Size of window in scene coordinates.

Static Public Attributes

- static const int [NON_PDF_CROPPING_BOTTOM](#) = 1
Bit flag when handle is aligned with bottom edge at reference point.
- static const int [NON_PDF_CROPPING_LEFT](#) = 2
Bit flag when handle is aligned with left edge at reference point.
- static const int [NON_PDF_CROPPING_RIGHT](#) = 4
Bit flag when handle is aligned with right edge at reference point.
- static const int [NON_PDF_CROPPING_TOP](#) = 8
Bit flag when handle is aligned with top edge at reference point.

5.214.1 Detailed Description

This class shows a frame around the selected portion of the import preview window.

This class was developed as a non-pdf equivalent to the [PdfCropping](#) class. See that class for more details

Definition at line 22 of file NonPdfCropping.h.

The documentation for this class was generated from the following files:

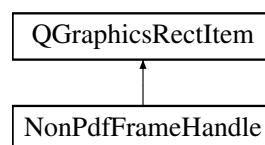
- NonPdf/NonPdfCropping.h
- NonPdf/NonPdfCropping.cpp

5.215 NonPdfFrameHandle Class Reference

This class acts as a single handle for the [NonPdfCropping](#) class.

```
#include <NonPdfFrameHandle.h>
```

Inheritance diagram for NonPdfFrameHandle:



Public Member Functions

- [NonPdfFrameHandle](#) (QGraphicsScene &scene, QGraphicsView &view, const QPointF &pointReference, int orientationFlags, [NonPdfCropping](#) &nonPdfCropping, int zValue)
Single constructor.
- virtual QVariant [itemChange](#) (GraphicsItemChange change, const QVariant &value)
Intercept the drags and process them, which is the whole point of handles.
- virtual void [paint](#) (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
Override the paint method so the dashed-border-when-selected can be removed.
- void [setDisableEventsWhileMovingAutomatically](#) (bool disable)
Temporarily disable event handling so code can move this object without triggering a cascade of events.

5.215.1 Detailed Description

This class acts as a single handle for the [NonPdfCropping](#) class.

Definition at line 19 of file NonPdfFrameHandle.h.

The documentation for this class was generated from the following files:

- NonPdf/NonPdfFrameHandle.h
- NonPdf/NonPdfFrameHandle.cpp

5.216 OrdinalGenerator Class Reference

Utility class for generating ordinal numbers.

```
#include <OrdinalGenerator.h>
```

Public Member Functions

- double [generateAxisPointOrdinal](#) (const [Document](#) &document)
Select ordinal just for uniqueness, since there is never a curve drawn through the axis points.
- double [generateCurvePointOrdinal](#) (const [Document](#) &document, const [Transformation](#) &transformation, const QPointF &posScreen, const QString &curveName)
Select ordinal so new point curve passes smoothly through existing points.

5.216.1 Detailed Description

Utility class for generating ordinal numbers.

For axis points, the ordinals are arbitrary but must be unique. For curve points, point is inserted according to its position and the [CurveStyle](#) settings

Definition at line 18 of file OrdinalGenerator.h.

The documentation for this class was generated from the following files:

- Ordinal/OrdinalGenerator.h
- Ordinal/OrdinalGenerator.cpp

5.217 Pdf Class Reference

Wrapper around the Poppler library.

```
#include <Pdf.h>
```

Public Member Functions

- Pdf ()
Single constructor.
- PdfReturn load (const QString &fileName, QImage &image, int resolution, ImportCropping importCropping, bool isErrorReportRegressionTest) const
Try to load the specified file. Success is indicated in the function return value.

5.217.1 Detailed Description

Wrapper around the Poppler library.

Engauge uses that library to read and import PDF files.

This class is only compiled and linked in when ENGAUGE_PDF is defined, since it links to the optional poppler library.

Definition at line 28 of file Pdf.h.

The documentation for this class was generated from the following files:

- Pdf/Pdf.h
- Pdf/Pdf.cpp

5.218 PdfCropping Class Reference

This class shows a frame around the selected portion of the pdf import preview window.

```
#include <PdfCropping.h>
```

Public Member Functions

- PdfCropping (QGraphicsScene &scene, ViewPreview &view)
Single constructor.
- QRectF frameRect () const
Frame rectangle selected by user.
- void moveBL (const QPointF &newPos, const QPointF &oldPos)
Bottom left corner handle was moved.
- void moveBR (const QPointF &newPos, const QPointF &oldPos)
Bottom right corner handle was moved.
- void moveTL (const QPointF &newPos, const QPointF &oldPos)
Top left corner handle was moved.
- void moveTR (const QPointF &newPos, const QPointF &oldPos)
Top right corner handle was moved.
- QSize windowSize () const
Size of window in scene coordinates.

Static Public Attributes

- static const int [PDF_CROPPING_BOTTOM](#) = 1
Bit flag when handle is aligned with bottom edge at reference point.
- static const int [PDF_CROPPING_LEFT](#) = 2
Bit flag when handle is aligned with left edge at reference point.
- static const int [PDF_CROPPING_RIGHT](#) = 4
Bit flag when handle is aligned with right edge at reference point.
- static const int [PDF_CROPPING_TOP](#) = 8
Bit flag when handle is aligned with top edge at reference point.

5.218.1 Detailed Description

This class shows a frame around the selected portion of the pdf import preview window.

Originally there were 4 handles at the corners and 4 handles at the middles of the sides, but dragging the corner handles did not result in 1/2 the movement at the middle handles. The middle handles were deemed not worth the effort

Definition at line 24 of file PdfCropping.h.

The documentation for this class was generated from the following files:

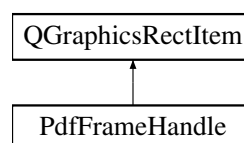
- Pdf/PdfCropping.h
- Pdf/PdfCropping.cpp

5.219 PdfFrameHandle Class Reference

This class acts as a single handle for the [PdfCropping](#) class.

```
#include <PdfFrameHandle.h>
```

Inheritance diagram for PdfFrameHandle:



Public Member Functions

- [PdfFrameHandle](#) (QGraphicsScene &scene, QGraphicsView &view, const QPointF &pointReference, int orientationFlags, [PdfCropping](#) &pdfCropping, int zValue)
Single constructor.
- virtual QVariant [itemChange](#) (GraphicsItemChange change, const QVariant &value)
Intercept the drags and process them, which is the whole point of handles.
- virtual void [paint](#) (QPainter *painter, const QStyleOptionGraphicsItem *option, QWidget *widget)
Override the paint method so the dashed-border-when-selected can be removed.
- void [setDisableEventsWhileMovingAutomatically](#) (bool disable)
Temporarily disable event handling so code can move this object without triggering a cascade of events.

5.219.1 Detailed Description

This class acts as a single handle for the [PdfCropping](#) class.

Definition at line 19 of file PdfFrameHandle.h.

The documentation for this class was generated from the following files:

- Pdf/PdfFrameHandle.h
- Pdf/PdfFrameHandle.cpp

5.220 Point Class Reference

Class that represents one digitized point. The screen-to-graph coordinate transformation is always external to this class.

```
#include <Point.h>
```

Public Member Functions

- [Point](#) ()
Default constructor so this class can be used inside a container.
- [Point](#) (const QString &curveName, const QPointF &posScreen)
Constructor for [Checker](#) temporary points, before real point gets added.
- [Point](#) (const QString &curveName, const QPointF &posScreen, const QPointF &posGraph, bool isXOnly)
Constructor for temporary point used to pre-check transformation points, before real point gets added.
- [Point](#) (const QString &curveName, const QString &identifier, const QPointF &posScreen, const QPointF &posGraph, double ordinal, bool isXOnly)
Constructor for axis points with identifier (after redo). The position, in screen coordinates, applies to the center of the [Point](#).
- [Point](#) (const QString &curveName, const QPointF &posScreen, const QPointF &posGraph, double ordinal, bool isXOnly)
Constructor for axis points without identifier (after redo). The position, in screen coordinates, applies to the center of the [Point](#).
- [Point](#) (const QString &curveName, const QString &identifier, const QPointF &posScreen, double ordinal)
Constructor for graph points with identifier (after redo)
- [Point](#) (const QString &curveName, const QPointF &posScreen, double ordinal)
Constructor for graph points without identifier (after redo)
- [Point](#) (QXmlStreamReader &reader)
Constructor when loading from serialized xml.
- [Point](#) & operator= (const [Point](#) &point)
Assignment constructor.
- [Point](#) (const [Point](#) &point)
Copy constructor.
- bool hasOrdinal () const
True if ordinal is defined.
- bool hasPosGraph () const
True if graph position is defined.
- QString identifier () const

- Unique identifier for a specific [Point](#).*

 - bool [isXOnly](#) () const
In DOCUMENT_AXES_POINTS_REQUIRED_4 modes, this is true/false if y/x coordinate is undefined.
 - bool [isAxisPoint](#) () const
True if point is an axis point. This is used only for sanity checks.
 - double [ordinal](#) (ApplyHasCheck applyHasCheck=KEEP_HAS_CHECK) const
Get method for ordinal. Skip check if copying one instance to another.
 - QPointF [posGraph](#) (ApplyHasCheck applyHasCheck=KEEP_HAS_CHECK) const
Accessor for graph position. Skip check if copying one instance to another.
 - QPointF [posScreen](#) () const
Accessor for screen position.
 - void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
 - void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize to stream.
 - void [setCurveName](#) (const QString &curveName)
Update the point identifier to match the specified curve name.
 - void [setOrdinal](#) (double [ordinal](#))
Set the ordinal used for ordering Points.
 - void [setPosGraph](#) (const QPointF &[posGraph](#))
Set method for position in graph coordinates.
 - void [setPosScreen](#) (const QPointF &[posScreen](#))
Set method for position in screen coordinates.

Static Public Member Functions

- static QString [curveNameFromPointIdentifier](#) (const QString &pointIdentifier)
Parse the curve name from the specified point identifier. This does the opposite of uniqueIdentifierGenerator.
- static unsigned int [identifierIndex](#) ()
Return the current index for storage in case we need to reset it later while performing a Redo.
- static void [setIdentifierIndex](#) (unsigned int [identifierIndex](#))
Reset the current index while performing a Redo.
- static QString [temporaryPointIdentifier](#) ()
[Point](#) identifier for temporary point that is used by DigitizeStateAxis.
- static double [UNDEFINED_ORDINAL](#) ()
Get method for undefined ordinal constant.

5.220.1 Detailed Description

Class that represents one digitized point. The screen-to-graph coordinate transformation is always external to this class.

Definition at line 23 of file Point.h.

5.220.2 Constructor & Destructor Documentation

5.220.2.1 Point::Point (const QString & curveName, const QPointF & posScreen)

Constructor for [Checker](#) temporary points, before real point gets added.

The position, in screen coordinates, applies to the center of the [Point](#)

Definition at line 33 of file Point.cpp.

5.220.2.2 `Point::Point (const QString & curveName, const QPointF & posScreen, const QPointF & posGraph, bool isXOnly)`

Constructor for temporary point used to pre-check transformation points, before real point gets added.

The position, in screen coordinates, applies to the center of the [Point](#)

Definition at line 52 of file `Point.cpp`.

The documentation for this class was generated from the following files:

- `Point/Point.h`
- `Point/Point.cpp`

5.221 PointComparator Struct Reference

Comparator for sorting [Point](#) class.

```
#include <PointComparator.h>
```

Public Member Functions

- `bool operator() (const Point &a, const Point &b) const`
Comparison function used by qSort.

5.221.1 Detailed Description

Comparator for sorting [Point](#) class.

Definition at line 13 of file `PointComparator.h`.

The documentation for this struct was generated from the following file:

- `Point/PointComparator.h`

5.222 PointIdentifiers Class Reference

Hash table class that tracks point identifiers as the key, with a corresponding boolean value.

```
#include <PointIdentifiers.h>
```

Public Member Functions

- [PointIdentifiers](#) ()
Single constructor.
- bool [contains](#) (const QString &pointIdentifier) const
True if specified entry exists in the table.
- int [count](#) () const
Number of entries.
- QString [getKey](#) (int i) const
Get key for index.
- bool [getValue](#) (const QString &pointIdentifier) const
Get value for key.
- void [loadXml](#) (QXmlStreamReader &reader)
Load from serialized xml.
- void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize table to xml.
- void [setKeyValue](#) (const QString &pointIdentifier, bool value)
Set key/value pair.

5.222.1 Detailed Description

Hash table class that tracks point identifiers as the key, with a corresponding boolean value.

Definition at line 19 of file PointIdentifiers.h.

5.222.2 Member Function Documentation

5.222.2.1 QString PointIdentifiers::getKey (int i) const

Get key for index.

This involves copying of all the keys and is therefore slower than using key lookup, so should not be used for extremely numerous point sets

Definition at line 33 of file PointIdentifiers.cpp.

The documentation for this class was generated from the following files:

- Point/PointIdentifiers.h
- Point/PointIdentifiers.cpp

5.223 PointMatchAlgorithm Class Reference

Algorithm returning a list of points that match the specified point.

```
#include <PointMatchAlgorithm.h>
```

Public Member Functions

- [PointMatchAlgorithm](#) (bool isGnuplot)
Single constructor.
- [QList< QPoint > findPoints](#) (const [QList< PointMatchPixel >](#) &samplePointPixels, const [QImage](#) &image, const [DocumentModelPointMatch](#) &modelPointMatch, const [Points](#) &pointsExisting)
Find points that match the specified sample point pixels. They are sorted by best-to-worst match.

5.223.1 Detailed Description

Algorithm returning a list of points that match the specified point.

This returns a list of matches, from best to worst. This is executed in a separate QThread so the gui thread is not blocked

Definition at line 26 of file [PointMatchAlgorithm.h](#).

The documentation for this class was generated from the following files:

- [Point/PointMatchAlgorithm.h](#)
- [Point/PointMatchAlgorithm.cpp](#)

5.224 PointMatchPixel Class Reference

Single on or off pixel out of the pixels that define the point match mode's candidate point.

```
#include <PointMatchPixel.h>
```

Public Member Functions

- [PointMatchPixel](#) (int xOffset, int yOffset, bool pixellsOn)
Single basic constructor.
- [PointMatchPixel](#) (const [PointMatchPixel](#) &other)
Copy constructor.
- [PointMatchPixel & operator=](#) (const [PointMatchPixel](#) &other)
Assignment operator.
- bool [pixellsOn](#) () const
True/false if pixel is on/off.
- int [xOffset](#) () const
X position relative to the center of the point.
- int [yOffset](#) () const
Y position relative to the center of the point.

5.224.1 Detailed Description

Single on or off pixel out of the pixels that define the point match mode's candidate point.

Definition at line 13 of file PointMatchPixel.h.

The documentation for this class was generated from the following files:

- Point/PointMatchPixel.h
- Point/PointMatchPixel.cpp

5.225 PointMatchTriplet Class Reference

Representation of one matched point as produced from the point match algorithm.

```
#include <PointMatchTriplet.h>
```

Public Member Functions

- [PointMatchTriplet](#) (int [x](#), int [y](#), double [correlation](#))
Single constructor.
- bool [operator<](#) (const [PointMatchTriplet](#) &other) const
Comparison operator for sorting lists of this class using qSort.
- double [correlation](#) () const
Get method for correlation.
- QPoint [point](#) () const
Return (x,y) coordinates as a point.
- int [x](#) () const
Get method for x coordinate.
- int [y](#) () const
Get method for y coordinate.

5.225.1 Detailed Description

Representation of one matched point as produced from the point match algorithm.

Definition at line 13 of file PointMatchTriplet.h.

The documentation for this class was generated from the following files:

- Point/PointMatchTriplet.h
- Point/PointMatchTriplet.cpp

5.226 PointStyle Class Reference

Details for a specific [Point](#).

```
#include <PointStyle.h>
```

Public Member Functions

- [PointStyle](#) ()
Default constructor only for use when this class is being stored by a container that requires the default constructor.
- [PointStyle](#) (PointShape pointShape, unsigned int [radius](#), int [lineWidth](#), ColorPalette [paletteColor](#))
Normal constructor. The style type and radius are determined by the currently selected [Curve](#).
- [PointStyle](#) (const [PointStyle](#) &other)
Copy constructor.
- [PointStyle](#) & [operator=](#) (const [PointStyle](#) &other)
Assignment constructor.
- bool [isCircle](#) () const
Return true if point is a circle, otherwise it is a polygon. For a circle, the radius is important and no polygon is needed from this class.
- int [lineWidth](#) () const
Get method for line width.
- void [loadXml](#) (QXmlStreamReader &reader)
Load model from serialized xml. Returns the curve name.
- ColorPalette [paletteColor](#) () const
Get method for point color.
- QPolygonF [polygon](#) () const
Return the polygon for creating a QGraphicsPolygonItem. The size is determined by the radius.
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- int [radius](#) () const
Radius of point. For a circle this is all that is needed to draw a circle. For a polygon, the radius determines the size of the polygon.
- void [saveXml](#) (QXmlStreamWriter &writer) const
Serialize to stream.
- void [setLineWidth](#) (int width)
Set method for line width.
- void [setPaletteColor](#) (ColorPalette [paletteColor](#))
Set method for point color.
- void [setRadius](#) (int [radius](#))
Set method for point radius.
- void [setShape](#) (PointShape [shape](#))
Set method for point shape.
- PointShape [shape](#) () const
Get method for point shape.

Static Public Member Functions

- static [PointStyle](#) [defaultAxesCurve](#) ()
Initial default for axes curve.
- static [PointStyle](#) [defaultGraphCurve](#) (int index)
Initial default for index'th graph curve.

5.226.1 Detailed Description

Details for a specific [Point](#).

Definition at line 20 of file `PointStyle.h`.

The documentation for this class was generated from the following files:

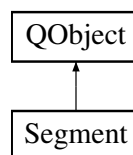
- `Point/PointStyle.h`
- `Point/PointStyle.cpp`

5.227 Segment Class Reference

Selectable piecewise-defined line that follows a filtered line in the image.

```
#include <Segment.h>
```

Inheritance diagram for `Segment`:



Public Slots

- void [slotHover](#) (bool hover)
Slot for hover enter/leave events in the associated `SegmentLines`.

Signals

- void [signalMouseClickedOnSegment](#) (QPointF posSegmentStart)
Pass mouse press event, with coordinates of first point in the [Segment](#) since that info uniquely identifies the owning [Segment](#).

Public Member Functions

- [Segment](#) (QGraphicsScene &scene, int yLast, bool isGnuplot)
Single constructor.
- void [appendColumn](#) (int x, int y, const [DocumentModelSegments](#) &modelSegments)
Add some more pixels in a new column to an active segment.
- QList< QPointF > [fillPoints](#) (const [DocumentModelSegments](#) &modelSegments)
Create evenly spaced points along the segment.
- QPointF [firstPoint](#) () const
Coordinates of first point in [Segment](#).
- void [forwardMousePress](#) ()
Forward mouse press event from a component [SegmentLine](#) that was just clicked on.
- double [length](#) () const
Get method for length in pixels.
- int [lineCount](#) () const
Get method for number of lines.
- void [removeUnneededLines](#) (int *foldedLines)
Try to compress a segment that was just completed, by folding together line from point i to point $i+1$, with the line from $i+1$ to $i+2$, then the line from $i+2$ to $i+3$, until one of the points is more than a half pixel from the folded line.
- void [updateModelSegment](#) (const [DocumentModelSegments](#) &modelSegments)
Update this segment given the new settings.

5.227.1 Detailed Description

Selectable piecewise-defined line that follows a filtered line in the image.

Clicking on a [Segment](#) results in the immediate creation of multiple Points along that [Segment](#).

Definition at line 21 of file Segment.h.

5.227.2 Member Function Documentation

5.227.2.1 QPointF Segment::firstPoint () const

Coordinates of first point in [Segment](#).

This info can be used to uniquely identify a [Segment](#). This method relies on SegmentFactory::removeEmpty↵ Segments to guarantee every [Segment](#) has at least one line

Definition at line 284 of file Segment.cpp.

5.227.2.2 void Segment::removeUnneededLines (int * foldedLines)

Try to compress a segment that was just completed, by folding together line from point i to point i+1, with the line from i+1 to i+2, then the line from i+2 to i+3, until one of the points is more than a half pixel from the folded line.

this should save memory and improve user interface responsiveness

Definition at line 419 of file Segment.cpp.

The documentation for this class was generated from the following files:

- Segment/Segment.h
- Segment/Segment.cpp

5.228 SegmentFactory Class Reference

Factory class for [Segment](#) objects.

```
#include <SegmentFactory.h>
```

Public Member Functions

- [SegmentFactory](#) (QGraphicsScene &scene, bool isGnuplot)
Single constructor.
- void [clearSegments](#) (QList< [Segment](#) * > &segments)
Remove the segments created by makeSegments.
- QList< QPoint > [fillPoints](#) (const [DocumentModelSegments](#) &modelSegments, QList< [Segment](#) * > segments)
Return segment fill points for all segments, for previewing.
- void [makeSegments](#) (const QImage &imageFiltered, const [DocumentModelSegments](#) &modelSegments, QList< [Segment](#) * > &segments, bool useDlg=true)
Main entry point for creating all Segments for the filtered image.

5.228.1 Detailed Description

Factory class for [Segment](#) objects.

The input is the filtered image.

The strategy is to fill out the segments output array as each segment finishes. This makes it easy to keep too-short Segments out of the output array, versus adding every new [Segment](#) to the output array as soon as it is created

Definition at line 27 of file SegmentFactory.h.

The documentation for this class was generated from the following files:

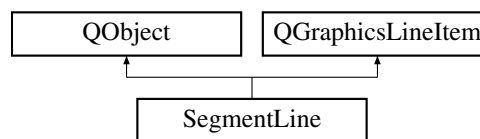
- Segment/SegmentFactory.h
- Segment/SegmentFactory.cpp

5.229 SegmentLine Class Reference

This class is a special case of the standard QGraphicsLineItem for segments.

```
#include <SegmentLine.h>
```

Inheritance diagram for SegmentLine:



Signals

- void [signalHover](#) (bool)
Pass hover enter/leave events to [Segment](#) that owns this.

Public Member Functions

- [SegmentLine](#) (QGraphicsScene &scene, const [DocumentModelSegments](#) &modelSegments, [Segment](#) *segment)
Single constructor.
- virtual void [hoverEnterEvent](#) (QGraphicsSceneHoverEvent *event)
Highlight this and all other SegmentLines belonging to the same [Segment](#) upon hover enter.
- virtual void [hoverLeaveEvent](#) (QGraphicsSceneHoverEvent *event)
Unset highlighting triggered by hover enter.
- virtual void [mousePressEvent](#) (QGraphicsSceneMouseEvent *event)
Create points along this curve.
- [Segment](#) * [segment](#) () const
[Segment](#) that owns this line.
- void [setHover](#) (bool hover)
Apply/remove highlighting triggered by hover enter/leave.
- void [updateModelSegment](#) (const [DocumentModelSegments](#) &modelSegments)
Update this segment line with new settings.

5.229.1 Detailed Description

This class is a special case of the standard QGraphicsLineItem for segments.

Definition at line 17 of file SegmentLine.h.

The documentation for this class was generated from the following files:

- Segment/SegmentLine.h
- Segment/SegmentLine.cpp

5.230 SettingsForGraph Class Reference

Manage storage and retrieval of the settings for the curves.

```
#include <SettingsForGraph.h>
```

Public Member Functions

- [SettingsForGraph](#) ()
Single constructor.
- QString [defaultCurveName](#) (int indexOneBased, const QString &defaultName) const
Default graph name for the specified curve index.
- QString [groupNameForNthCurve](#) (int indexOneBased) const
Return the group name, that appears in the settings file/registry, for the specified curve index.

5.230.1 Detailed Description

Manage storage and retrieval of the settings for the curves.

Definition at line 13 of file SettingsForGraph.h.

The documentation for this class was generated from the following files:

- Settings/SettingsForGraph.h
- Settings/SettingsForGraph.cpp

5.231 Spline Class Reference

Cubic interpolation given independent and dependent value vectors.

```
#include <Spline.h>
```

Public Member Functions

- [Spline](#) (const std::vector< double > &t, const std::vector< [SplinePair](#) > &xy)
Initialize spline with independent (t) and dependent (x and y) value vectors.
- [SplinePair findSplinePairForFunctionX](#) (double x, int numIterations) const
Use bisection algorithm to iteratively find the [SplinePair](#) interpolated to best match the specified x value.
- [SplinePair interpolateCoeff](#) (double t) const
Return interpolated y for specified x.
- [SplinePair interpolateControlPoints](#) (double t) const
Return interpolated y for specified x, for testing.
- [SplinePair p1](#) (unsigned int i) const
Bezier p1 control point for specified interval. P0 is m_xy[i] and P3 is m_xy[i+1].
- [SplinePair p2](#) (unsigned int i) const
Bezier p2 control point for specified interval. P0 is m_xy[i] and P3 is m_xy[i+1].

5.231.1 Detailed Description

Cubic interpolation given independent and dependent value vectors.

X is handled as a dependent variable based on the unitless independent parameter t so curves are not restricted to $x(i) \neq x(i+1)$.

This class has two modes that can be run side by side:

1. Coefficient mode, where each interval has coefficients a,b,c,d in $xy = ai + bi*(t-ti) + ci*(t-ti)^2 + di*(t-ti)^3$
2. Bezier mode, where each interval has 2 endpoints and 2 control points in $P = (1-s)^3*P0 + 3*(1-s)^2*s*P1 + 3*(1-s)*s^2*P2 + s^3*P3$. Although interpolation can be performed in bezier mode, this interpolation was just to verify consistent operation between the two modes. The real purpose of this mode is to produce reliable control points, p1 and p2, for each interval. The control points can be used by external code that relies on control points to perform its own interpolation

Definition at line 21 of file Spline.h.

5.231.2 Constructor & Destructor Documentation

5.231.2.1 [Spline::Spline](#) (const std::vector< double > &t, const std::vector< [SplinePair](#) > &xy)

Initialize spline with independent (t) and dependent (x and y) value vectors.

Besides initializing the a,b,c,d coefficients for each interval, this constructor initializes bezier points (P1 and P2) for each interval, where P0 and P3 are the start and end points for each interval.

Definition at line 11 of file Spline.cpp.

5.231.3 Member Function Documentation

5.231.3.1 [SplinePair Spline::findSplinePairForFunctionX](#) (double x, int numIterations) const

Use bisection algorithm to iteratively find the [SplinePair](#) interpolated to best match the specified x value.

This assumes the curve is a function since otherwise there is the potential for multiple solutions

Definition at line 116 of file Spline.cpp.

5.231.3.2 SplinePair Spline::interpolateCoeff (double t) const

Return interpolated y for specified x.

The appropriate interval is selected from the entire set of piecewise-defined intervals, then the corresponding a,b,c,d coefficients are applied

Definition at line 166 of file Spline.cpp.

5.231.3.3 SplinePair Spline::interpolateControlPoints (double t) const

Return interpolated y for specified x, for testing.

This uses the bezier points. If the t values are not separated by +1 consistently then this algorithm will probably need additional effort to work right

Definition at line 179 of file Spline.cpp.

The documentation for this class was generated from the following files:

- Spline/Spline.h
- Spline/Spline.cpp

5.232 SplineCoeff Class Reference

Four element vector of a,b,c,d coefficients and the associated x value, for one interval of a set of piecewise-defined intervals.

```
#include <SplineCoeff.h>
```

Public Member Functions

- [SplineCoeff](#) (double t)
Partial constructor for use mostly by container classes.
- [SplineCoeff](#) (double t, const [SplinePair](#) &a, const [SplinePair](#) &b, const [SplinePair](#) &c, const [SplinePair](#) &d)
Full constructor.
- bool [operator<](#) (const [SplineCoeff](#) &e) const
Comparison operator for collection.
- bool [operator<](#) (double t) const
Comparison operator for collection.
- [SplinePair](#) a () const
Get method for a.
- [SplinePair](#) b () const
Get method for b.
- [SplinePair](#) c () const
Get method for c.
- [SplinePair](#) d () const
Get method for d.
- [SplinePair](#) eval (double t) const
Evaluate the value using the a,b,c,d coefficients, over this interval.
- double t () const
T value associated with these a,b,c,d coefficients.

5.232.1 Detailed Description

Four element vector of a,b,c,d coefficients and the associated x value, for one interval of a set of piecewise-defined intervals.

Definition at line 14 of file SplineCoeff.h.

The documentation for this class was generated from the following files:

- Spline/SplineCoeff.h
- Spline/SplineCoeff.cpp

5.233 SplinePair Class Reference

Single X/Y pair for cubic spline interpolation initialization and calculations.

```
#include <SplinePair.h>
```

Public Member Functions

- [SplinePair](#) ()
Default constructor. Normally used only by generic container classes.
- [SplinePair](#) (double scalar)
Constructor for filling vector with a single scalar. Provided for convenience over preferred constructor.
- [SplinePair](#) (double x, double y)
Preferred constructor. Used when default constructor is not being used by generic container classes.
- [SplinePair](#) (const [SplinePair](#) &other)
Assignment constructor.
- [SplinePair operator+](#) (const [SplinePair](#) &other) const
Addition operator.
- [SplinePair operator-](#) (const [SplinePair](#) &other) const
Subtraction operator.
- [SplinePair operator*](#) (const [SplinePair](#) &other) const
Multiplication operator.
- [SplinePair operator/](#) (const [SplinePair](#) &other) const
Division operator.
- double [x](#) () const
Get method for x.
- double [y](#) () const
Get method for y.

5.233.1 Detailed Description

Single X/Y pair for cubic spline interpolation initialization and calculations.

Definition at line 11 of file SplinePair.h.

The documentation for this class was generated from the following files:

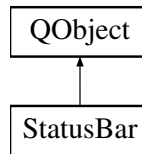
- Spline/SplinePair.h
- Spline/SplinePair.cpp

5.234 StatusBar Class Reference

Wrapper around QStatusBar to manage permanent widgets.

```
#include <StatusBar.h>
```

Inheritance diagram for StatusBar:



Public Slots

- void [slotZoom](#) (int)
Receive zoom selection from [MainWindow](#).

Signals

- void [signalZoom](#) (int)
Send zoom factor, that was just selected in the status bar, to [MainWindow](#).

Public Member Functions

- [StatusBar](#) (QStatusBar &statusBar)
Single constructor that accepts the previously-constructed standard QStatusBar.
- void [setCoordinates](#) (const QString &coordsScreen, const QString &coordsGraph, const QString &resolutionGraph)
Populate the coordinates fields. Unavailable values are empty. Html-encoding to highlight with colors is supported.
- void [setStatusBarMode](#) (StatusBarMode [statusBarMode](#))
Set the status bar visibility mode.
- void [showTemporaryMessage](#) (const QString &message)
Show temporary message in status bar. After a short interval the message will disappear.
- StatusBarMode [statusBarMode](#) () const
Current mode for status bar visibility. This is tracked locally so this class knows when to hide/show the status bar.
- void [wakeUp](#) ()
Enable all widgets in the status bar. This is called just after a [Document](#) becomes active.

5.234.1 Detailed Description

Wrapper around QStatusBar to manage permanent widgets.

This class does not inherit from QStatusBar since QApplication automatically sets up its own QStatusBar

Definition at line 21 of file StatusBar.h.

The documentation for this class was generated from the following files:

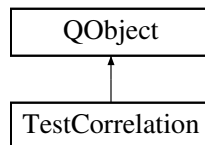
- StatusBar/StatusBar.h
- StatusBar/StatusBar.cpp

5.235 TestCorrelation Class Reference

Unit tests of fast correlation algorithm.

```
#include <TestCorrelation.h>
```

Inheritance diagram for TestCorrelation:



Public Member Functions

- [TestCorrelation](#) (QObject *parent=0)
Single constructor.

5.235.1 Detailed Description

Unit tests of fast correlation algorithm.

Definition at line 7 of file TestCorrelation.h.

The documentation for this class was generated from the following files:

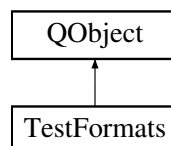
- Test/TestCorrelation.h
- Test/TestCorrelation.cpp

5.236 TestFormats Class Reference

Unit tests of formats.

```
#include <TestFormats.h>
```

Inheritance diagram for TestFormats:



Public Member Functions

- [TestFormats](#) (QObject *parent=0)
Single constructor.

5.236.1 Detailed Description

Unit tests of formats.

Definition at line 8 of file TestFormats.h.

The documentation for this class was generated from the following files:

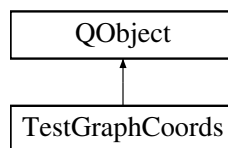
- Test/TestFormats.h
- Test/TestFormats.cpp

5.237 TestGraphCoords Class Reference

Unit tests of graph coordinate sanity checking.

```
#include <TestGraphCoords.h>
```

Inheritance diagram for TestGraphCoords:



Public Member Functions

- [TestGraphCoords](#) (QObject *parent=0)
Single constructor.

5.237.1 Detailed Description

Unit tests of graph coordinate sanity checking.

Definition at line 10 of file TestGraphCoords.h.

The documentation for this class was generated from the following files:

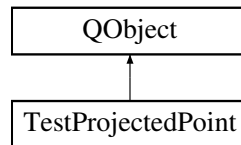
- Test/TestGraphCoords.h
- Test/TestGraphCoords.cpp

5.238 TestProjectedPoint Class Reference

Unit test of spline library.

```
#include <TestProjectedPoint.h>
```

Inheritance diagram for TestProjectedPoint:



Public Member Functions

- [TestProjectedPoint](#) (QObject *parent=0)
Single constructor.

5.238.1 Detailed Description

Unit test of spline library.

Definition at line 7 of file TestProjectedPoint.h.

The documentation for this class was generated from the following files:

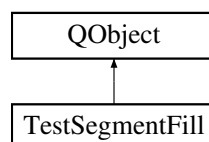
- Test/TestProjectedPoint.h
- Test/TestProjectedPoint.cpp

5.239 TestSegmentFill Class Reference

Unit test of segment fill feature.

```
#include <TestSegmentFill.h>
```

Inheritance diagram for TestSegmentFill:



Public Member Functions

- [TestSegmentFill](#) (QObject *parent=0)
Single constructor.

5.239.1 Detailed Description

Unit test of segment fill feature.

Definition at line 7 of file TestSegmentFill.h.

The documentation for this class was generated from the following files:

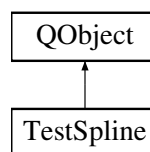
- Test/TestSegmentFill.h
- Test/TestSegmentFill.cpp

5.240 TestSpline Class Reference

Unit test of spline library.

```
#include <TestSpline.h>
```

Inheritance diagram for TestSpline:



Public Member Functions

- [TestSpline](#) (QObject *parent=0)
Single constructor.

5.240.1 Detailed Description

Unit test of spline library.

Definition at line 7 of file TestSpline.h.

The documentation for this class was generated from the following files:

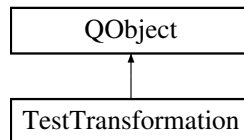
- Test/TestSpline.h
- Test/TestSpline.cpp

5.241 TestTransformation Class Reference

Unit test of transformation class. Checking mostly involves verifying forward/reverse are inverses of each other.

```
#include <TestTransformation.h>
```

Inheritance diagram for TestTransformation:



Public Member Functions

- [TestTransformation](#) (QObject *parent=0)
Single constructor.

5.241.1 Detailed Description

Unit test of transformation class. Checking mostly involves verifying forward/reverse are inverses of each other.

Definition at line 9 of file TestTransformation.h.

The documentation for this class was generated from the following files:

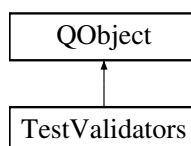
- Test/TestTransformation.h
- Test/TestTransformation.cpp

5.242 TestValidators Class Reference

Unit tests of validators.

```
#include <TestValidators.h>
```

Inheritance diagram for TestValidators:



Public Member Functions

- [TestValidators](#) (QObject *parent=0)
Single constructor.

5.242.1 Detailed Description

Unit tests of validators.

Definition at line 10 of file TestValidators.h.

The documentation for this class was generated from the following files:

- Test/TestValidators.h
- Test/TestValidators.cpp

5.243 Transformation Class Reference

Affine transformation between screen and graph coordinates, based on digitized axis points.

```
#include <Transformation.h>
```

Public Member Functions

- [Transformation](#) ()
Default constructor. This is marked as undefined until the proper number of axis points are added.
- [Transformation](#) (const [Transformation](#) &other)
Copy constructor.
- [Transformation](#) & [operator=](#) (const [Transformation](#) &other)
Assignment operator.
- void [identity](#) ()
Identity transformation.
- bool [operator!=](#) (const [Transformation](#) &other)
Inequality operator. This is marked as defined.
- void [coordTextForStatusBar](#) (QPointF cursorScreen, QString &coordsScreen, QString &coordsGraph, Q←String &resolutionGraph)
Return string descriptions of cursor coordinates for status bar.
- [DocumentModelCoords](#) [modelCoords](#) () const
Get method for [DocumentModelCoords](#).
- [MainWindowModel](#) [modelMainWindow](#) () const
Get method for [MainWindowModel](#).
- void [printStream](#) (QString indentation, QTextStream &str) const
Debugging method that supports print method of this class and printStream method of some other class(es)
- void [resetOnLoad](#) ()
Reset, when loading a document after the first, to same state that first document was at when loaded.
- bool [transformsDefined](#) () const
Transform is defined when at least three axis points have been digitized.
- void [transformLinearCartesianGraphToRawGraph](#) (const QPointF &coordGraph, QPointF &coordScreen) const
Transform from linear cartesian graph coordinates to cartesian, polar, linear, log coordinates.
- void [transformLinearCartesianGraphToScreen](#) (const QPointF &coordGraph, QPointF &coordScreen) const
Transform from linear cartesian graph coordinates to cartesian pixel screen coordinates.
- QTransform [transformMatrix](#) () const
Get method for copying only, for the transform matrix.

- void [transformRawGraphToLinearCartesianGraph](#) (const QPointF &pointRaw, QPointF &pointLinear↔ Cartesian) const
Convert graph coordinates (linear or log, cartesian or polar) to linear cartesian coordinates.
- void [transformRawGraphToScreen](#) (const QPointF &pointRaw, QPointF &pointScreen) const
Transform from raw graph coordinates to linear cartesian graph coordinates, then to screen coordinates.
- void [transformScreenToLinearCartesianGraph](#) (const QPointF &pointScreen, QPointF &pointLinear↔ Cartesian) const
Transform screen coordinates to linear cartesian coordinates.
- void [transformScreenToRawGraph](#) (const QPointF &coordScreen, QPointF &coordGraph) const
Transform from cartesian pixel screen coordinates to cartesian/polar graph coordinates.
- void [update](#) (bool fileIsLoaded, const [CmdMediator](#) &cmdMediator, const [MainWindowModel](#) &modelMain↔ Window)
Update transform by iterating through the axis points.

Static Public Member Functions

- static QTransform [calculateTransformFromLinearCartesianPoints](#) (const QPointF &posFrom0, const QPointF &posFrom1, const QPointF &posFrom2, const QPointF &posTo0, const QPointF &posTo1, const QPointF &posTo2)
Calculate QTransform using from/to points that have already been adjusted for, when applicable, log scaling and polar coordinates.
- static QPointF [cartesianFromCartesianOrPolar](#) (const [DocumentModelCoords](#) &modelCoords, const QPointF &posGraphIn)
Output cartesian coordinates from input cartesian or polar coordinates. This is static for easier use externally.
- static QPointF [cartesianOrPolarFromCartesian](#) (const [DocumentModelCoords](#) &modelCoords, const QPointF &posGraphIn)
Output cartesian or polar coordinates from input cartesian coordinates. This is static for easier use externally.
- static double [logToLinearCartesian](#) (double xy)
Convert cartesian scaling from log to linear. Calling code is responsible for determining if this is necessary.
- static double [logToLinearRadius](#) (double r, double rCenter)
Convert radius scaling from log to linear. Calling code is responsible for determining if this is necessary.

Friends

- class **TestTransformation**

5.243.1 Detailed Description

Affine transformation between screen and graph coordinates, based on digitized axis points.

[Transformation](#) from screen pixels to graph coordinates involves two steps:

1. Transform from screen pixels (which are linear and cartesian) to linear cartesian graph coordinates
2. Transform from linear cartesian graph coordinates to the final graph coordinates, which are linear or log scaled, and cartesian or polar

[Transformation](#) from graph coordinates to screen pixels reverses the steps involved in the transformation from screen pixels to graph coordinates

Log scaling is calculated as $(x_{\text{Linear}} - x_{\text{LogMin}}) / (x_{\text{LogMax}} - x_{\text{LogMin}}) = (\ln(x_{\text{Log}}) - \ln(x_{\text{LogMin}})) / (\ln(x_{\text{LogMax}}) - \ln(x_{\text{LogMin}}))$, which leaves the points $(x_{\text{LogMin}}, y_{\text{LogMin}})$ and $(x_{\text{LogMax}}, y_{\text{LogMax}})$ unaffected but gives log growth on all other points

Definition at line 30 of file Transformation.h.

5.243.2 Member Function Documentation

5.243.2.1 `QTransform Transformation::calculateTransformFromLinearCartesianPoints (const QPointF & posFrom0, const QPointF & posFrom1, const QPointF & posFrom2, const QPointF & posTo0, const QPointF & posTo1, const QPointF & posTo2) [static]`

Calculate QTransform using from/to points that have already been adjusted for, when applicable, log scaling and polar coordinates.

The points are linear and cartesian.

This method is kept very generic since it used for different types of transformations:

1. In many place to calculate screen-to/from-graph
2. By [Checker](#) to calculate unaligned-to/from-aligned

Definition at line 56 of file Transformation.cpp.

The documentation for this class was generated from the following files:

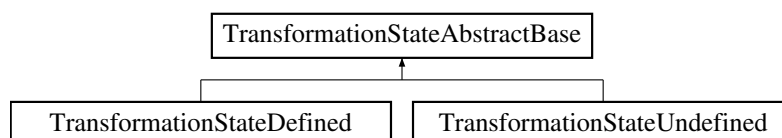
- Transformation/Transformation.h
- Transformation/Transformation.cpp

5.244 TransformationStateAbstractBase Class Reference

Base class for all transformation states. This serves as an interface to [TransformationStateContext](#).

```
#include <TransformationStateAbstractBase.h>
```

Inheritance diagram for TransformationStateAbstractBase:



Public Member Functions

- [TransformationStateAbstractBase](#) ([TransformationStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation, const QString &selectedGraphCurve)=0
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- virtual void [end](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation)=0
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [updateAxesChecker](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation)=0
Apply the new [DocumentModelAxesChecker](#).

Protected Member Functions

- [TransformationStateContext](#) & [context](#) ()

Reference to the [TransformationStateContext](#) that contains all the [TransformationStateAbstractBase](#) subclasses, without const.

5.244.1 Detailed Description

Base class for all transformation states. This serves as an interface to [TransformationStateContext](#).

Definition at line 25 of file [TransformationStateAbstractBase.h](#).

The documentation for this class was generated from the following files:

- Transformation/TransformationStateAbstractBase.h
- Transformation/TransformationStateAbstractBase.cpp

5.245 TransformationStateContext Class Reference

Context class for transformation state machine.

```
#include <TransformationStateContext.h>
```

Public Member Functions

- [TransformationStateContext](#) (QGraphicsScene &scene, bool [isGnuplot](#))
Single constructor.
- bool [isGnuplot](#) () const
Flag for gnuplot debug files.
- void [resetOnLoad](#) ()
Reset, when loading a document after the first, to same state that first document was at when loaded.
- void [triggerStateTransition](#) (TransformationState transformationState, [CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation, const QString &selectedGraphCurve)
Trigger a state transition to be performed immediately.
- void [updateAxesChecker](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation)
Apply the new [DocumentModelAxesChecker](#).

5.245.1 Detailed Description

Context class for transformation state machine.

This removes some tricky state processing from [MainWindow](#). Unlike typical state machines, the transitions are driven directly from the outside rather than indirectly by events that are processed by the states (this has trigger↔StateTransition rather than requestStateTransition)

Definition at line 21 of file [TransformationStateContext.h](#).

The documentation for this class was generated from the following files:

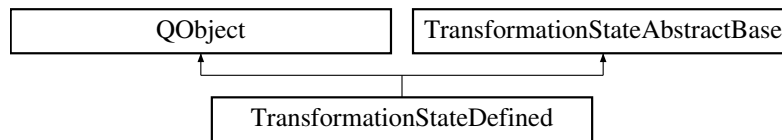
- Transformation/TransformationStateContext.h
- Transformation/TransformationStateContext.cpp

5.246 TransformationStateDefined Class Reference

Class to show transformation since transformation is defined.

```
#include <TransformationStateDefined.h>
```

Inheritance diagram for TransformationStateDefined:



Public Member Functions

- [TransformationStateDefined](#) ([TransformationStateContext](#) &context, QGraphicsScene &scene)
Single constructor.
- virtual void [begin](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation, const QString &selectedGraphCurve)
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- virtual void [end](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation)
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [updateAxesChecker](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation)
Apply the new [DocumentModelAxesChecker](#).

Additional Inherited Members

5.246.1 Detailed Description

Class to show transformation since transformation is defined.

Definition at line 18 of file TransformationStateDefined.h.

The documentation for this class was generated from the following files:

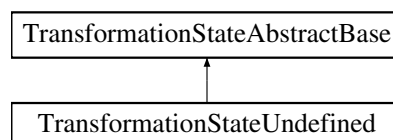
- Transformation/TransformationStateDefined.h
- Transformation/TransformationStateDefined.cpp

5.247 TransformationStateUndefined Class Reference

Class to not show transformation since transformation is undefined.

```
#include <TransformationStateUndefined.h>
```

Inheritance diagram for TransformationStateUndefined:



Public Member Functions

- [TransformationStateUndefined](#) ([TransformationStateContext](#) &context, QGraphicsScene &scene)
Single constructor.
- virtual void [begin](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation, const QString &selectedGraphCurve)
Method that is called at the exact moment a state is entered. Typically called just after end for the previous state.
- virtual void [end](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation)
Method that is called at the exact moment a state is exited. Typically called just before begin for the next state.
- virtual void [updateAxesChecker](#) ([CmdMediator](#) &cmdMediator, const [Transformation](#) &transformation)
Apply the new [DocumentModelAxesChecker](#).

Additional Inherited Members

5.247.1 Detailed Description

Class to not show transformation since transformation is undefined.

Definition at line 13 of file TransformationStateUndefined.h.

The documentation for this class was generated from the following files:

- Transformation/TransformationStateUndefined.h
- Transformation/TransformationStateUndefined.cpp

5.248 TranslatorContainer Class Reference

Class that stores QTranslator objects for the duration of application execution.

```
#include <TranslatorContainer.h>
```

Public Member Functions

- [TranslatorContainer](#) (QApplication &app)
Single constructor.

5.248.1 Detailed Description

Class that stores QTranslator objects for the duration of application execution.

Definition at line 8 of file TranslatorContainer.h.

The documentation for this class was generated from the following files:

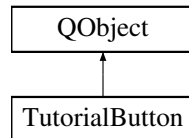
- Translator/TranslatorContainer.h
- Translator/TranslatorContainer.cpp

5.249 TutorialButton Class Reference

Show a button with text for clicking ion. The button is implemented using layering of two graphics items (text and rectangle)

```
#include <TutorialButton.h>
```

Inheritance diagram for TutorialButton:



Signals

- void [signalTriggered](#) ()
Signal that button was triggered.

Public Member Functions

- [TutorialButton](#) (const QString &text, QGraphicsScene &scene)
Single constructor. Position is set after creation using setGeometry.
- void [handleTriggered](#) ()
Callback to be called when button was triggered by mouse event.
- void [setGeometry](#) (const QPoint &pos)
Set the position. This is called after creation so screen extent is available for positioning calculations.
- QSize [size](#) () const
Size of this button.

5.249.1 Detailed Description

Show a button with text for clicking ion. The button is implemented using layering of two graphics items (text and rectangle)

Definition at line 20 of file TutorialButton.h.

The documentation for this class was generated from the following files:

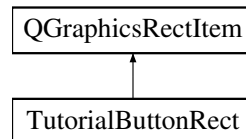
- Tutorial/TutorialButton.h
- Tutorial/TutorialButton.cpp

5.250 TutorialButtonRect Class Reference

This class customizes QGraphicsRectItem so it performs a callback after a mouse event.

```
#include <TutorialButtonRect.h>
```

Inheritance diagram for TutorialButtonRect:



Public Member Functions

- [TutorialButtonRect](#) ([TutorialButton](#) &tutorialButton)
Single constructor.
- virtual void [mouseReleaseEvent](#) (QGraphicsSceneMouseEvent *event)
Forward mouse event to [TutorialButton](#).

5.250.1 Detailed Description

This class customizes QGraphicsRectItem so it performs a callback after a mouse event.

Definition at line 15 of file TutorialButtonRect.h.

The documentation for this class was generated from the following files:

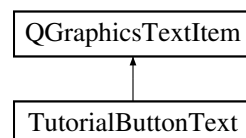
- Tutorial/TutorialButtonRect.h
- Tutorial/TutorialButtonRect.cpp

5.251 TutorialButtonText Class Reference

This class customizes QGraphicsTextItem so it performs a callback after a mouse event.

```
#include <TutorialButtonText.h>
```

Inheritance diagram for TutorialButtonText:



Public Member Functions

- [TutorialButtonText](#) ([TutorialButton](#) &tutorialButton, const QString &text, [TutorialButtonRect](#) *rect)
Single constructor.
- virtual void [mouseReleaseEvent](#) (QGraphicsSceneMouseEvent *event)
Forward mouse event to [TutorialButton](#).

5.251.1 Detailed Description

This class customizes QGraphicsTextItem so it performs a callback after a mouse event.

Definition at line 15 of file TutorialButtonText.h.

The documentation for this class was generated from the following files:

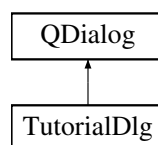
- Tutorial/TutorialButtonText.h
- Tutorial/TutorialButtonText.cpp

5.252 TutorialDlg Class Reference

Tutorial using a strategy like a comic strip with decision points deciding which panels appear.

```
#include <TutorialDlg.h>
```

Inheritance diagram for TutorialDlg:



Public Member Functions

- [TutorialDlg](#) ([MainWindow](#) *mainWindow)
Single constructor.
- QSize [backgroundSize](#) () const
Make geometry available for layout.
- QGraphicsScene & [scene](#) ()
Single scene the covers the entire tutorial dialog.
- QGraphicsView & [view](#) ()
Single view that displays the single scene.

5.252.1 Detailed Description

Tutorial using a strategy like a comic strip with decision points deciding which panels appear.

This is implemented as a QGraphicsScene with states in charge of managing the contents of the scene

Definition at line 19 of file TutorialDlg.h.

The documentation for this class was generated from the following files:

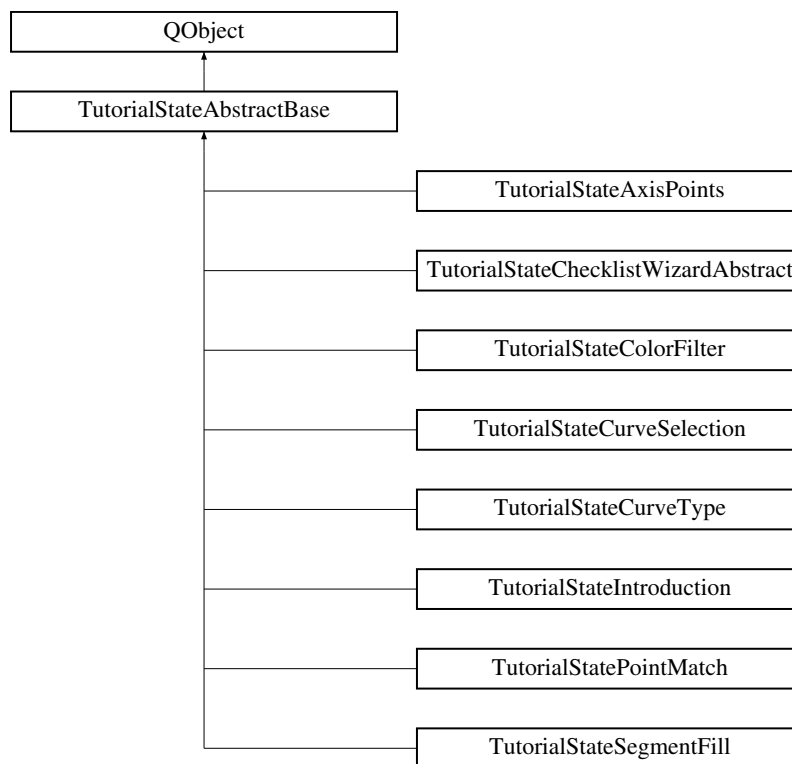
- Tutorial/TutorialDlg.h
- Tutorial/TutorialDlg.cpp

5.253 TutorialStateAbstractBase Class Reference

One state manages one panel of the tutorial.

```
#include <TutorialStateAbstractBase.h>
```

Inheritance diagram for TutorialStateAbstractBase:



Public Member Functions

- [TutorialStateAbstractBase](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()=0
Transition into this state.
- virtual void [end](#) ()=0
Transition out of this state.

Protected Member Functions

- int [buttonMargin](#) () const
Buttons are placed up against bottom side, and left or right side, separated by this margin.
- [TutorialStateContext](#) & [context](#) ()
Context class for the tutorial state machine.
- QGraphicsPixmapItem * [createPixmapItem](#) (const QString &resource, const QPoint &pos)
Factory method for pixmap items.
- QGraphicsTextItem * [createTextItem](#) (const QString &text, const QPoint &pos)
Factory method for text items.
- QGraphicsTextItem * [createTitle](#) (const QString &text)
Factory method for title items.

5.253.1 Detailed Description

One state manages one panel of the tutorial.

Definition at line 30 of file TutorialStateAbstractBase.h.

The documentation for this class was generated from the following files:

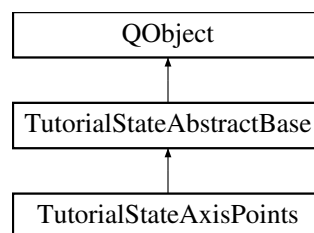
- Tutorial/TutorialStateAbstractBase.h
- Tutorial/TutorialStateAbstractBase.cpp

5.254 TutorialStateAxisPoints Class Reference

Axis points panel discusses axis point digitization.

```
#include <TutorialStateAxisPoints.h>
```

Inheritance diagram for TutorialStateAxisPoints:



Public Slots

- void [slotNext](#) ()
Slot called when next button is triggered.
- void [slotPrevious](#) ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStateAxisPoints](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Transition into this state.
- virtual void [end](#) ()
Transition out of this state.

Additional Inherited Members

5.254.1 Detailed Description

Axis points panel discusses axis point digitization.

Definition at line 18 of file TutorialStateAxisPoints.h.

The documentation for this class was generated from the following files:

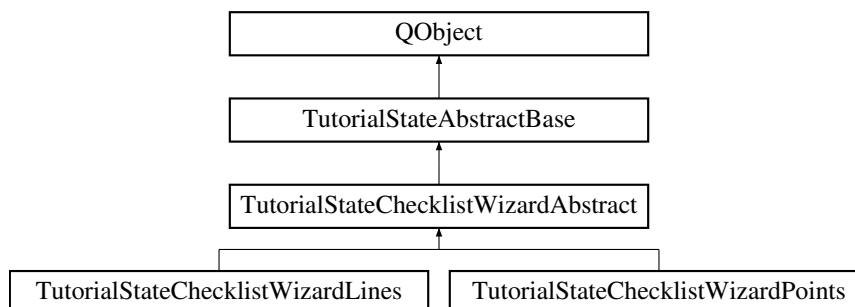
- Tutorial/TutorialStateAxisPoints.h
- Tutorial/TutorialStateAxisPoints.cpp

5.255 TutorialStateChecklistWizardAbstract Class Reference

Abstract class that supports checklist wizard panels.

```
#include <TutorialStateChecklistWizardAbstract.h>
```

Inheritance diagram for TutorialStateChecklistWizardAbstract:



Public Member Functions

- [TutorialStateChecklistWizardAbstract](#) ([TutorialStateContext](#) &context)
Single constructor.

Protected Member Functions

- void `begin` ()
Common begin processing.
- void `end` ()
Common end processing.
- `TutorialButton` * `previous` ()
Previous button for hooking up button to slot.

5.255.1 Detailed Description

Abstract class that supports checklist wizard panels.

Definition at line 18 of file TutorialStateChecklistWizardAbstract.h.

The documentation for this class was generated from the following files:

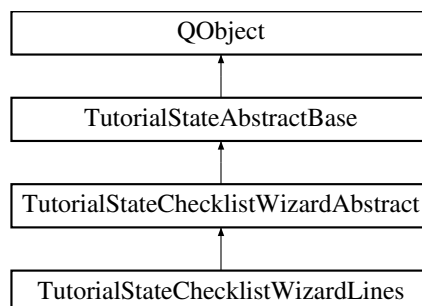
- Tutorial/TutorialStateChecklistWizardAbstract.h
- Tutorial/TutorialStateChecklistWizardAbstract.cpp

5.256 TutorialStateChecklistWizardLines Class Reference

Checklist wizard panel for lines discusses the checklist wizard, and returns to `TRANSITION_STATE_SEGMENT`↵
`_FILL`.

```
#include <TutorialStateChecklistWizardLines.h>
```

Inheritance diagram for TutorialStateChecklistWizardLines:



Public Slots

- void `slotPrevious` ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStateChecklistWizardLines](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Common begin processing.
- virtual void [end](#) ()
Common end processing.

Additional Inherited Members

5.256.1 Detailed Description

Checklist wizard panel for lines discusses the checklist wizard, and returns to `TRANSITION_STATE_SEGMENT_FILL`.

Definition at line 18 of file `TutorialStateChecklistWizardLines.h`.

The documentation for this class was generated from the following files:

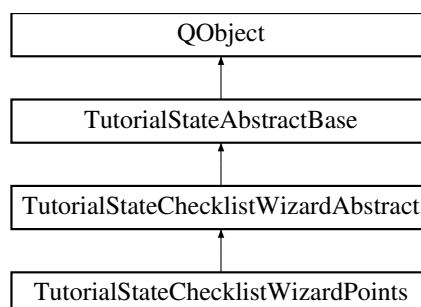
- `Tutorial/TutorialStateChecklistWizardLines.h`
- `Tutorial/TutorialStateChecklistWizardLines.cpp`

5.257 TutorialStateChecklistWizardPoints Class Reference

Checklist wizard panel for points discusses the checklist wizard, and returns to `TRANSITION_STATE_POINT_MATCH`.

```
#include <TutorialStateChecklistWizardPoints.h>
```

Inheritance diagram for `TutorialStateChecklistWizardPoints`:



Public Slots

- void [slotPrevious](#) ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStateChecklistWizardPoints](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Common begin processing.
- virtual void [end](#) ()
Common end processing.

Additional Inherited Members

5.257.1 Detailed Description

Checklist wizard panel for points discusses the checklist wizard, and returns to `TRANSITION_STATE_POINT_MATCH`.

Definition at line 18 of file `TutorialStateChecklistWizardPoints.h`.

The documentation for this class was generated from the following files:

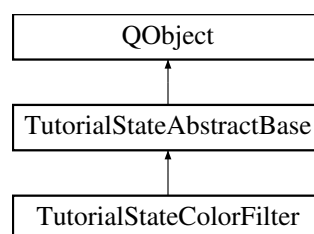
- `Tutorial/TutorialStateChecklistWizardPoints.h`
- `Tutorial/TutorialStateChecklistWizardPoints.cpp`

5.258 TutorialStateColorFilter Class Reference

Color filter panel discusses the curve-specific color filtering.

```
#include <TutorialStateColorFilter.h>
```

Inheritance diagram for `TutorialStateColorFilter`:



Public Slots

- void [slotBack](#) ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStateColorFilter](#) ([TutorialStateContext](#) &[context](#))
Single constructor.
- virtual void [begin](#) ()
Transition into this state.
- virtual void [end](#) ()
Transition out of this state.

Additional Inherited Members

5.258.1 Detailed Description

Color filter panel discusses the curve-specific color filtering.

Definition at line 18 of file `TutorialStateColorFilter.h`.

The documentation for this class was generated from the following files:

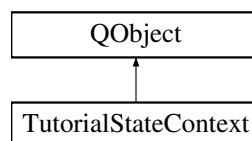
- `Tutorial/TutorialStateColorFilter.h`
- `Tutorial/TutorialStateColorFilter.cpp`

5.259 TutorialStateContext Class Reference

Context class for tutorial state machine.

```
#include <TutorialStateContext.h>
```

Inheritance diagram for `TutorialStateContext`:



Public Member Functions

- [TutorialStateContext](#) ([TutorialDlg](#) &[tutorialDlg](#))
Single constructor.
- void [requestDelayedStateTransition](#) ([TutorialState](#) [tutorialState](#))
Request a transition to the specified state from the current state.
- void [requestImmediateStateTransition](#) ([TutorialState](#) [tutorialState](#))
Request a transition to the specified state from the current state.
- [TutorialDlg](#) & [tutorialDlg](#) ()
Access to tutorial dialogs and its scene.

5.259.1 Detailed Description

Context class for tutorial state machine.

Each state represents one panel in the tutorial Tutorial assumptions:

1. Dealing with multiple curves is postponed until the end of the tutorial.

Definition at line 20 of file TutorialStateContext.h.

5.259.2 Member Function Documentation

5.259.2.1 void TutorialStateContext::requestDelayedStateTransition (TutorialState *tutorialState*)

Request a transition to the specified state from the current state.

A timer is used. This assumes [TutorialStateContext](#) is NOT on the stack - probably since an external event (mouse click, ...) resulted in a callback to the current state

Definition at line 81 of file TutorialStateContext.cpp.

5.259.2.2 void TutorialStateContext::requestImmediateStateTransition (TutorialState *tutorialState*)

Request a transition to the specified state from the current state.

The transition is delayed until the current state is off the stack to prevent stack corruption errors. This assumes [TutorialStateContext](#) is on the stack to finish the transition after execution returns from the state

Definition at line 90 of file TutorialStateContext.cpp.

The documentation for this class was generated from the following files:

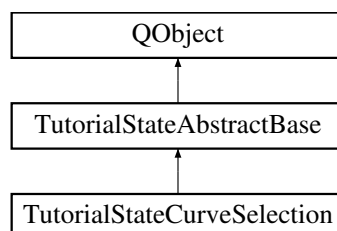
- Tutorial/TutorialStateContext.h
- Tutorial/TutorialStateContext.cpp

5.260 TutorialStateCurveSelection Class Reference

[Curve](#) selection panel discusses how to select a curve, and perform setup on the selected curve.

```
#include <TutorialStateCurveSelection.h>
```

Inheritance diagram for TutorialStateCurveSelection:



Public Slots

- void [slotColorFilter](#) ()
Slot called when settings button is triggered.
- void [slotNext](#) ()
Slot called when next button is triggered.
- void [slotPrevious](#) ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStateCurveSelection](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Transition into this state.
- virtual void [end](#) ()
Transition out of this state.

Additional Inherited Members

5.260.1 Detailed Description

[Curve](#) selection panel discusses how to select a curve, and perform setup on the selected curve.

Definition at line 18 of file `TutorialStateCurveSelection.h`.

The documentation for this class was generated from the following files:

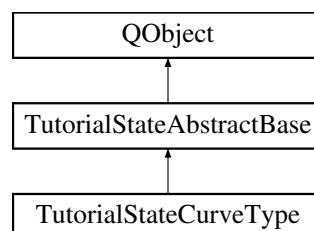
- `Tutorial/TutorialStateCurveSelection.h`
- `Tutorial/TutorialStateCurveSelection.cpp`

5.261 TutorialStateCurveType Class Reference

[Curve](#) type state/panel lets user select the curve type (lines or points)

```
#include <TutorialStateCurveType.h>
```

Inheritance diagram for `TutorialStateCurveType`:



Public Slots

- void [slotNextCurves](#) ()
Slot called when next button for curves is triggered.
- void [slotNextLines](#) ()
Slot called when next button for lines is triggered.
- void [slotPrevious](#) ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStateCurveType](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Transition into this state.
- virtual void [end](#) ()
Transition out of this state.

Additional Inherited Members

5.261.1 Detailed Description

[Curve](#) type state/panel lets user select the curve type (lines or points)

Definition at line 18 of file `TutorialStateCurveType.h`.

The documentation for this class was generated from the following files:

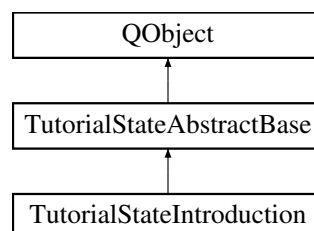
- `Tutorial/TutorialStateCurveType.h`
- `Tutorial/TutorialStateCurveType.cpp`

5.262 TutorialStateIntroduction Class Reference

Introduction state/panel is the first panel the user sees.

```
#include <TutorialStateIntroduction.h>
```

Inheritance diagram for `TutorialStateIntroduction`:



Public Slots

- void [slotNext](#) ()
Slot called when next button is triggered.

Public Member Functions

- [TutorialStateIntroduction](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Transition into this state.
- virtual void [end](#) ()
Transition out of this state.

Additional Inherited Members

5.262.1 Detailed Description

Introduction state/panel is the first panel the user sees.

Definition at line 18 of file TutorialStateIntroduction.h.

The documentation for this class was generated from the following files:

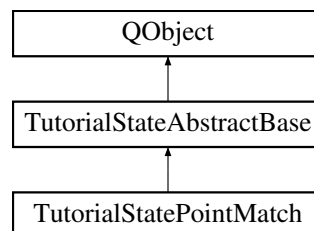
- Tutorial/TutorialStateIntroduction.h
- Tutorial/TutorialStateIntroduction.cpp

5.263 TutorialStatePointMatch Class Reference

[Point](#) match panel discusses the matching of points in curves without lines.

```
#include <TutorialStatePointMatch.h>
```

Inheritance diagram for TutorialStatePointMatch:



Public Slots

- void [slotNext](#) ()
Slot called when next button is triggered.
- void [slotPrevious](#) ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStatePointMatch](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Transition into this state.
- virtual void [end](#) ()
Transition out of this state.

Additional Inherited Members

5.263.1 Detailed Description

[Point](#) match panel discusses the matching of points in curves without lines.

Definition at line 18 of file TutorialStatePointMatch.h.

The documentation for this class was generated from the following files:

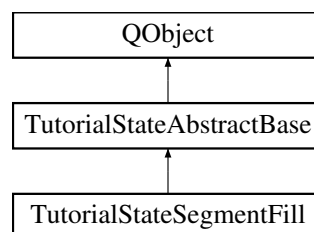
- Tutorial/TutorialStatePointMatch.h
- Tutorial/TutorialStatePointMatch.cpp

5.264 TutorialStateSegmentFill Class Reference

[Segment](#) fill panel discusses the digitization of points along curve lines.

```
#include <TutorialStateSegmentFill.h>
```

Inheritance diagram for TutorialStateSegmentFill:



Public Slots

- void [slotNext](#) ()
Slot called when next button is triggered.
- void [slotPrevious](#) ()
Slot called to return to previous panel.

Public Member Functions

- [TutorialStateSegmentFill](#) ([TutorialStateContext](#) &context)
Single constructor.
- virtual void [begin](#) ()
Transition into this state.
- virtual void [end](#) ()
Transition out of this state.

Additional Inherited Members

5.264.1 Detailed Description

[Segment](#) fill panel discusses the digitization of points along curve lines.

Definition at line 18 of file `TutorialStateSegmentFill.h`.

The documentation for this class was generated from the following files:

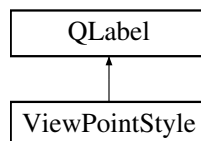
- `Tutorial/TutorialStateSegmentFill.h`
- `Tutorial/TutorialStateSegmentFill.cpp`

5.265 ViewPointStyle Class Reference

Class that displays a view of the current [Curve](#)'s point style.

```
#include <ViewPointStyle.h>
```

Inheritance diagram for ViewPointStyle:



Public Member Functions

- [ViewPointStyle](#) (QWidget *parent=0)
Single constructor.
- void [setEnabled](#) (bool enabled)
Show the style with semi-transparency or full-transparency to indicate if associated [Curve](#) is active or not.
- void [setPointStyle](#) (const [PointStyle](#) &pointStyle)
Apply the [PointStyle](#) of the currently selected curve.
- void [unsetPointStyle](#) ()
Apply no [PointStyle](#).

5.265.1 Detailed Description

Class that displays a view of the current [Curve](#)'s point style.

Do NOT apply a visible border since that would hide a square drawn just inside the four sides.

Definition at line 16 of file ViewPointStyle.h.

The documentation for this class was generated from the following files:

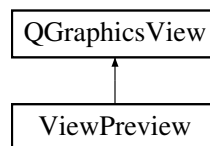
- View/ViewPointStyle.h
- View/ViewPointStyle.cpp

5.266 ViewPreview Class Reference

Class that modifies QGraphicsView to automatically expand/shrink the view to fit the window, after resize events.

```
#include <ViewPreview.h>
```

Inheritance diagram for ViewPreview:



Public Types

- enum [ViewAspectRatio](#) { **VIEW_ASPECT_RATIO_VARIABLE**, **VIEW_ASPECT_RATIO_ONE_TO_ONE** }
Prevent aspect ratio distortion in certain previews by providing fixed 1:1 aspect ratio option.

Signals

- void [signalMouseMove](#) (QPointF pos)
Forward the mouse move events.

Public Member Functions

- [ViewPreview](#) (QGraphicsScene *scene, [ViewAspectRatio](#) viewAspectRatio, QWidget *parent=0)
Single constructor.
- virtual void [mouseMoveEvent](#) (QMouseEvent *event)
Intercept cursor move events and forward them.
- virtual void [resizeEvent](#) (QResizeEvent *event)
Intercept resize events so we can rescale to the graphics items just fit into the resized window.
- virtual void [wheelEvent](#) (QWheelEvent *event)
Intercept wheel event and discard it so accidentally moving the wheel does not move drawn items out of view.

5.266.1 Detailed Description

Class that modifies QGraphicsView to automatically expand/shrink the view to fit the window, after resize events.

Definition at line 14 of file ViewPreview.h.

The documentation for this class was generated from the following files:

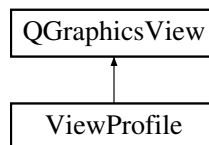
- View/ViewPreview.h
- View/ViewPreview.cpp

5.267 ViewProfile Class Reference

Class that modifies QGraphicsView to present a two-dimensional profile, with movable dividers for selecting a range.

```
#include <ViewProfile.h>
```

Inheritance diagram for ViewProfile:



Public Member Functions

- [ViewProfile](#) (QGraphicsScene *scene, int minimumWidth, QWidget *parent=0)
Single constructor.
- virtual void [resizeEvent](#) (QResizeEvent *event)
Intercept resize events so the geometry can be scaled to perfectly fit into the window.

5.267.1 Detailed Description

Class that modifies QGraphicsView to present a two-dimensional profile, with movable dividers for selecting a range.

Definition at line 15 of file ViewProfile.h.

The documentation for this class was generated from the following files:

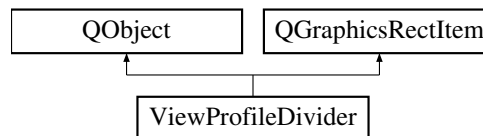
- View/ViewProfile.h
- View/ViewProfile.cpp

5.268 ViewProfileDivider Class Reference

Divider that can be dragged, in a dialog QGraphicsView.

```
#include <ViewProfileDivider.h>
```

Inheritance diagram for ViewProfileDivider:



Signals

- void [signalMovedLow](#) (double xSceneOther)
Signal used when divider is dragged and m_isLowerBoundary is true.
- void [signalMovedHigh](#) (double xSceneOther)
Signal used when divider is dragged and m_isLowerBoundary is false.

Public Member Functions

- [ViewProfileDivider](#) (QGraphicsScene &scene, QGraphicsView &view, int sceneWidth, int sceneHeight, int yCenter, bool isLowerBoundary)
Single constructor.
- virtual QVariant [itemChange](#) (GraphicsItemChange change, const QVariant &value)
Intercept changes so divider movement can be restricted to horizontal direction only.
- virtual void [mousePressEvent](#) (QGraphicsSceneMouseEvent *event)
Save paddle position at start of click-and-drag.
- void [setX](#) (double x, double xLow, double xHigh)
Set the position by specifying the new x coordinate.

5.268.1 Detailed Description

Divider that can be dragged, in a dialog QGraphicsView.

Click on the paddle to drag. There are three parts:

1. Paddle which is the superclass of this class, since we catch its events so dragging works
2. Divider which is a vertical line
3. Shaded area that extends from xAnchor to the divider

Definition at line 23 of file ViewProfileDivider.h.

The documentation for this class was generated from the following files:

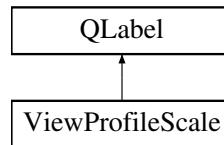
- View/ViewProfileDivider.h
- View/ViewProfileDivider.cpp

5.269 ViewProfileScale Class Reference

Linear horizontal scale, with the spectrum reflecting the active filter parameter.

```
#include <ViewProfileScale.h>
```

Inheritance diagram for ViewProfileScale:



Public Member Functions

- [ViewProfileScale](#) (int minimumWidth, QWidget *parent=0)
Single constructor.
- virtual void [paintEvent](#) (QPaintEvent *)
Draw the gradient.
- void [setBackgroundColor](#) (QRgb rgbBackground)
Save the background color for foreground calculations.
- void [setColorFilterMode](#) (ColorFilterMode colorFilterMode)
Change the gradient type.

5.269.1 Detailed Description

Linear horizontal scale, with the spectrum reflecting the active filter parameter.

Definition at line 16 of file ViewProfileScale.h.

The documentation for this class was generated from the following files:

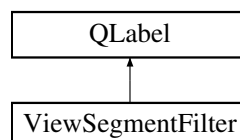
- View/ViewProfileScale.h
- View/ViewProfileScale.cpp

5.270 ViewSegmentFilter Class Reference

Class that displays the current [Segment](#) Filter in a [MainWindow](#) toolbar.

```
#include <ViewSegmentFilter.h>
```

Inheritance diagram for ViewSegmentFilter:



Public Member Functions

- [ViewSegmentFilter](#) (QWidget *parent=0)
Single constructor.
- virtual void [paintEvent](#) (QPaintEvent *event)
Paint with a horizontal linear gradient.
- void [setColorFilterSettings](#) (const [ColorFilterSettings](#) &colorFilterSettings, const QPixmap &pixmap)
Apply the color filter of the currently selected curve. The pixmap is included so the background color can be computed.
- void [setEnabled](#) (bool enabled)
Show the style with semi-transparency or full-transparency to indicate if associated [Curve](#) is active or not.
- void [unsetColorFilterSettings](#) ()
Apply no color filter.

5.270.1 Detailed Description

Class that displays the current [Segment](#) Filter in a [MainWindow](#) toolbar.

A gradient is displayed. No border is drawn so the appearance is consistent with [ViewPointStyle](#) which would not work with a border.

Definition at line 18 of file ViewSegmentFilter.h.

The documentation for this class was generated from the following files:

- View/ViewSegmentFilter.h
- View/ViewSegmentFilter.cpp

Chapter 6

File Documentation

6.1 Callback/CallbackSearchReturn.h File Reference

Enumerations

- enum `CallbackSearchReturn` { `CALLBACK_SEARCH_RETURN_CONTINUE`, `CALLBACK_SEARCH_RETURN_INTERRUPT`, `CALLBACK_SEARCH_RETURN_NO_RECURSE` }

Return values for search callback methods.

6.1.1 Detailed Description

Copyright

Copyright (c) 2012 Mark Mitchell. All rights reserved. Distribution requires prior written permission.

6.1.2 Enumeration Type Documentation

6.1.2.1 enum `CallbackSearchReturn`

Return values for search callback methods.

This lets each task-specific callback method control the generic search algorithm.

Interrupting a search is useful when searching for the first node that meets some criteria that the callback method applies.

Not recursing into a node's children is useful when trying to collect a list of subtrees, and the root node for each. This is useful for `TransformRecognizerManualBaseExponentExponent`.

Enumerator

`CALLBACK_SEARCH_RETURN_CONTINUE` Continue normal execution of the search.

`CALLBACK_SEARCH_RETURN_INTERRUPT` Immediately terminate the current search.

`CALLBACK_SEARCH_RETURN_NO_RECURSE` Do not recurse into the current node's children, but continue current search.

Definition at line 24 of file `CallbackSearchReturn.h`.

Index

- addCoordSystems
 - Document, [174](#)
- addPoint
 - GraphicsLinesForCurve, [227](#)
 - GraphicsLinesForCurves, [228](#)
- addPointAxisWithGeneratedIdentifier
 - CoordSystem, [107](#)
 - CoordSystemContext, [112](#)
 - CoordSystemInterface, [116](#)
 - Document, [174](#)
- addPointAxisWithSpecifiedIdentifier
 - CoordSystem, [107](#)
 - CoordSystemContext, [112](#)
 - CoordSystemInterface, [116](#)
 - Document, [175](#)
- applyImportCropping
 - ImportCroppingUtilPdf, [247](#)
- BackgroundStateAbstractBase, [33](#)
- BackgroundStateContext, [34](#)
 - setCurveSelected, [35](#)
- BackgroundStateCurve, [36](#)
- BackgroundStateNone, [37](#)
- BackgroundStateOriginal, [38](#)
- BackgroundStateUnloaded, [39](#)
- begin
 - DigitizeStateAbstractBase, [129](#)
 - DigitizeStateAxis, [130](#)
 - DigitizeStateColorPicker, [132](#)
 - DigitizeStateCurve, [135](#)
 - DigitizeStateEmpty, [137](#)
 - DigitizeStatePointMatch, [138](#)
 - DigitizeStateSegment, [140](#)
 - DigitizeStateSelect, [142](#)
- CALLBACK_SEARCH_RETURN_CONTINUE
 - CallbackSearchReturn.h, [311](#)
- CALLBACK_SEARCH_RETURN_INTERRUPT
 - CallbackSearchReturn.h, [311](#)
- CALLBACK_SEARCH_RETURN_NO_RECURSE
 - CallbackSearchReturn.h, [311](#)
- calculateTransformFromLinearCartesianPoints
 - Transformation, [285](#)
- Callback/CallbackSearchReturn.h, [311](#)
- CallbackAddPointsInCurvesGraphs, [40](#)
- CallbackAxesCheckerFromAxesPoints, [40](#)
- CallbackAxisPointsAbstract, [41](#)
 - isError, [42](#)
 - matrixGraph, [42](#)
 - matrixScreen, [42](#)
- CallbackBoundingRects, [43](#)
- CallbackCheckAddPointAxis, [43](#)
- CallbackCheckEditPointAxis, [44](#)
- CallbackDocumentHash, [45](#)
- CallbackGatherXThetaValuesFunctions, [45](#)
- CallbackNextOrdinal, [46](#)
- CallbackPointOrdinal, [46](#)
- CallbackRemovePointsInCurvesGraphs, [47](#)
- CallbackSceneUpdateAfterCommand, [48](#)
- CallbackSearchReturn
 - CallbackSearchReturn.h, [311](#)
- CallbackSearchReturn.h
 - CALLBACK_SEARCH_RETURN_CONTINUE, [311](#)
 - CALLBACK_SEARCH_RETURN_INTERRUPT, [311](#)
 - CALLBACK_SEARCH_RETURN_NO_RECURSE, [311](#)
 - CallbackSearchReturn, [311](#)
- CallbackUpdateTransform, [48](#)
 - transformIsDefined, [49](#)
- Checker, [49](#)
 - prepareForDisplay, [50](#)
 - updateModelAxesChecker, [50](#)
- ChecklistGuide, [51](#)
- ChecklistGuideBrowser, [52](#)
- ChecklistGuidePage, [52](#)
- ChecklistGuidePageConclusion, [53](#)
- ChecklistGuidePageCurves, [54](#)
- ChecklistGuidePageIntro, [55](#)
- ChecklistGuideWizard, [56](#)
- ChecklistLineEdit, [56](#)
- CmdAbstract, [57](#)
 - resetSelection, [59](#)
 - saveOrCheckPostCommandDocumentStateHash, [59](#)
 - saveOrCheckPreCommandDocumentStateHash, [59](#)
- CmdAddPointAxis, [60](#)
- CmdAddPointGraph, [61](#)
- CmdAddPointsGraph, [62](#)
- CmdCopy, [63](#)
- CmdCut, [64](#)
- CmdDelete, [65](#)
- CmdEditPointAxis, [66](#)
- CmdEditPointGraph, [67](#)
- CmdFactory, [68](#)
- CmdMediator, [68](#)
 - isModified, [70](#)

- setDocumentAxesPointsRequired, 70
- CmdMoveBy, 70
- CmdPaste, 71
- CmdPointChangeBase, 72
- CmdRedoForTest, 74
- CmdSelectCoordSystem, 75
- CmdSettingsAxesChecker, 76
- CmdSettingsColorFilter, 77
- CmdSettingsCoords, 78
- CmdSettingsCurveAddRemove, 79
- CmdSettingsCurveProperties, 80
- CmdSettingsDigitizeCurve, 81
- CmdSettingsExportFormat, 82
- CmdSettingsGeneral, 83
- CmdSettingsGridDisplay, 84
- CmdSettingsGridRemoval, 85
- CmdSettingsPointMatch, 86
- CmdSettingsSegments, 87
- CmdStackShadow, 88
- CmdUndoForTest, 89
- ColorFilter, 90
 - marginColor, 90
 - pixelToZeroToOneOrMinusOne, 90
- ColorFilterEntry, 91
- ColorFilterHistogram, 91
 - generate, 92
- ColorFilterSettings, 92
 - high, 94
 - low, 94
- ColorFilterSettingsStrategyAbstractBase, 95
- ColorFilterSettingsStrategyForeground, 95
- ColorFilterSettingsStrategyHue, 96
- ColorFilterSettingsStrategyIntensity, 97
- ColorFilterSettingsStrategySaturation, 98
- ColorFilterSettingsStrategyValue, 98
- ColorFilterStrategyAbstractBase, 99
- ColorFilterStrategyForeground, 100
- ColorFilterStrategyHue, 101
- ColorFilterStrategyIntensity, 101
- ColorFilterStrategySaturation, 102
- ColorFilterStrategyValue, 103
- CoordSystem, 104
 - addPointAxisWithGeneratedIdentifier, 107
 - addPointAxisWithSpecifiedIdentifier, 107
 - isXOnly, 108
 - updatePointOrdinals, 108
- CoordSystemContext, 108
 - addPointAxisWithGeneratedIdentifier, 112
 - addPointAxisWithSpecifiedIdentifier, 112
 - updatePointOrdinals, 113
- CoordSystemInterface, 113
 - addPointAxisWithGeneratedIdentifier, 116
 - addPointAxisWithSpecifiedIdentifier, 116
 - updatePointOrdinals, 117
- correlateWithShift
 - Correlation, 118
- correlateWithoutShift
 - Correlation, 118
- Correlation, 117
 - correlateWithShift, 118
 - correlateWithoutShift, 118
- createGridLine
 - GridLineFactory, 241
- CursorFactory, 118
- Curve, 119
 - updatePointOrdinals, 120
- CurveNameList, 121
- CurveNameListEntry, 122
- CurveSettingsInt, 123
- CurveStyle, 125
- CurveStyles, 126
- CurvesGraphs, 124
- DigitizeStateAbstractBase, 127
 - begin, 129
- DigitizeStateAxis, 129
 - begin, 130
- DigitizeStateColorPicker, 131
 - begin, 132
- DigitizeStateContext, 132
- DigitizeStateCurve, 134
 - begin, 135
- DigitizeStateEmpty, 135
 - begin, 137
- DigitizeStatePointMatch, 137
 - begin, 138
- DigitizeStateSegment, 139
 - begin, 140
- DigitizeStateSelect, 140
 - begin, 142
- DlgAbout, 142
- DlgEditPointAxis, 143
 - DlgEditPointAxis, 143
- DlgEditPointGraph, 144
 - DlgEditPointGraph, 144
- DlgEditPointGraphLineEdit, 145
- DlgErrorReport, 145
- DlgFilterCommand, 146
- DlgFilterThread, 147
- DlgFilterWorker, 147
- DlgImportAdvanced, 148
- DlgImportCroppingNonPdf, 149
- DlgImportCroppingPdf, 150
- DlgRequiresTransform, 151
- DlgSettingsAbstractBase, 152
 - enableOk, 153
- DlgSettingsAxesChecker, 154
- DlgSettingsColorFilter, 155
- DlgSettingsCoords, 156
- DlgSettingsCurveAddRemove, 157
- DlgSettingsCurveProperties, 158
- DlgSettingsDigitizeCurve, 159
- DlgSettingsExportFormat, 160
- DlgSettingsGeneral, 161
- DlgSettingsGridDisplay, 162
- DlgSettingsGridRemoval, 163
- DlgSettingsMainWindow, 164

- DlgSettingsPointMatch, 165
- DlgSettingsSegments, 166
- DlgValidatorAbstract, 167
- DlgValidatorDateTime, 168
- DlgValidatorDegreesMinutesSeconds, 168
- DlgValidatorFactory, 169
- DlgValidatorNumber, 170
- Document, 171
 - addCoordSystems, 174
 - addPointAxisWithGeneratedIdentifier, 174
 - addPointAxisWithSpecifiedIdentifier, 175
 - setDocumentAxesPointsRequired, 175
 - updatePointOrdinals, 175
- DocumentHashGenerator, 175
- DocumentModelAbstractBase, 176
- DocumentModelAxesChecker, 177
- DocumentModelColorFilter, 178
 - high, 180
 - low, 180
- DocumentModelCoords, 181
- DocumentModelDigitizeCurve, 182
- DocumentModelExportFormat, 184
- DocumentModelGeneral, 185
- DocumentModelGridDisplay, 186
 - stable, 188
- DocumentModelGridRemoval, 188
 - stable, 190
- DocumentModelPointMatch, 191
- DocumentModelSegments, 192
- enableOk
 - DlgSettingsAbstractBase, 153
- erasePixel
 - GridHealer, 238
- ExportAlignLinear, 193
- ExportAlignLog, 194
- ExportFileAbstractBase, 195
- ExportFileFunctions, 196
 - exportToFile, 196
- ExportFileRelations, 197
 - exportToFile, 197
- ExportImageForRegression, 198
- ExportOrdinalsSmooth, 198
- ExportOrdinalsStraight, 199
- ExportToClipboard, 199
 - exportToClipboard, 200
- exportToClipboard
 - ExportToClipboard, 200
- ExportToFile, 200
 - exportToFile, 201
- exportToFile
 - ExportFileFunctions, 196
 - ExportFileRelations, 197
 - ExportToFile, 201
- ExportXThetaValuesMergedFunctions, 201
- FileCmdAbstract, 202
- FileCmdClose, 203
- FileCmdExport, 204
- FileCmdFactory, 205
- FileCmdImport, 205
- FileCmdOpen, 206
- FileCmdScript, 207
- FilterImage, 207
- findSplinePairForFunctionX
 - Spline, 274
- firstPoint
 - Segment, 271
- FormatCoordsUnits, 208
- FormatCoordsUnitsStrategyAbstractBase, 208
 - precisionDigitsForRawNumber, 209
- FormatCoordsUnitsStrategyNonPolarTheta, 209
- FormatCoordsUnitsStrategyPolarTheta, 210
- FormatDateTime, 211
 - parseInput, 211
- FormatDegreesMinutesSecondsBase, 212
 - parseInput, 212
- FormatDegreesMinutesSecondsNonPolarTheta, 213
- FormatDegreesMinutesSecondsPolarTheta, 213
- generate
 - ColorFilterHistogram, 92
- GeometryModel, 214
- GeometryStrategyAbstractBase, 215
 - insertSubintervalsAndLoadDistances, 216
 - polygonAreaForSimplyConnected, 216
- GeometryStrategyContext, 216
- GeometryStrategyFunctionSmooth, 217
- GeometryStrategyFunctionStraight, 218
- GeometryStrategyRelationSmooth, 219
- GeometryStrategyRelationStraight, 219
- GeometryWindow, 220
- getKey
 - PointIdentifiers, 266
- GhostEllipse, 221
- GhostPath, 222
- GhostPolygon, 223
- Ghosts, 223
- GraphicsArcItem, 224
- GraphicsItemsExtractor, 225
- GraphicsLinesForCurve, 226
 - addPoint, 227
 - removeTemporaryPointIfExists, 227
- GraphicsLinesForCurves, 227
 - addPoint, 228
 - removeTemporaryPointIfExists, 228
- GraphicsPoint, 229
- GraphicsPointAbstractBase, 230
- GraphicsPointEllipse, 231
- GraphicsPointFactory, 232
- GraphicsPointPolygon, 232
- GraphicsScene, 233
 - removeTemporaryPointIfExists, 235
 - updateAfterCommand, 235
 - updateGraphicsLinesToMatchGraphicsPoints, 235
- GraphicsView, 235
- GridClassifier, 237
- GridHealer, 238

- erasePixel, [238](#)
- GridInitializer, [238](#)
- GridLine, [239](#)
- GridLineFactory, [240](#)
 - createGridLine, [241](#)
- GridLineLimiter, [241](#)
- GridLines, [242](#)
- GridRemoval, [242](#)
- HelpBrowser, [243](#)
- HelpWindow, [244](#)
- high
 - ColorFilterSettings, [94](#)
 - DocumentModelColorFilter, [180](#)
- ImportCroppingUtilBase, [244](#)
- ImportCroppingUtilNonPdf, [245](#)
- ImportCroppingUtilPdf, [246](#)
 - applyImportCropping, [247](#)
- insertSubintervalsAndLoadDistances
 - GeometryStrategyAbstractBase, [216](#)
- interpolateCoeff
 - Spline, [274](#)
- interpolateControlPoints
 - Spline, [275](#)
- isError
 - CallbackAxisPointsAbstract, [42](#)
- isModified
 - CmdMediator, [70](#)
- isXOnly
 - CoordSystem, [108](#)
- Jpeg2000, [247](#)
- LineStyle, [247](#)
- LoadFileInfo, [249](#)
- LoadImageFromUrl, [249](#)
- loggerAssert
 - LoggerUpload, [250](#)
- LoggerUpload, [250](#)
 - loggerAssert, [250](#)
- low
 - ColorFilterSettings, [94](#)
 - DocumentModelColorFilter, [180](#)
- MainWindow, [251](#)
 - MainWindow, [253](#)
 - selectOriginal, [253](#)
 - updateGraphicsLinesToMatchGraphicsPoints, [253](#)
- MainWindowModel, [254](#)
- marginColor
 - ColorFilter, [90](#)
- matrixGraph
 - CallbackAxisPointsAbstract, [42](#)
- matrixScreen
 - CallbackAxisPointsAbstract, [42](#)
- MigrateToVersion6, [255](#)
- MimePoints, [256](#)
- NetworkClient, [257](#)
- NonPdf, [258](#)
- NonPdfCropping, [258](#)
- NonPdfFrameHandle, [259](#)
- OrdinalGenerator, [260](#)
- parseInput
 - FormatDateTime, [211](#)
 - FormatDegreesMinutesSecondsBase, [212](#)
- Pdf, [260](#)
- PdfCropping, [261](#)
- PdfFrameHandle, [262](#)
- pixelToZeroToOneOrMinusOne
 - ColorFilter, [90](#)
- Point, [263](#)
 - Point, [264](#)
- PointComparator, [265](#)
- PointIdentifiers, [265](#)
 - getKey, [266](#)
- PointMatchAlgorithm, [266](#)
- PointMatchPixel, [267](#)
- PointMatchTriplet, [268](#)
- PointStyle, [268](#)
- polygonAreaForSimplyConnected
 - GeometryStrategyAbstractBase, [216](#)
- precisionDigitsForRawNumber
 - FormatCoordsUnitsStrategyAbstractBase, [209](#)
- prepareForDisplay
 - Checker, [50](#)
- removeTemporaryPointIfExists
 - GraphicsLinesForCurve, [227](#)
 - GraphicsLinesForCurves, [228](#)
 - GraphicsScene, [235](#)
- removeUnneededLines
 - Segment, [271](#)
- requestDelayedStateTransition
 - TutorialStateContext, [299](#)
- requestImmediateStateTransition
 - TutorialStateContext, [299](#)
- resetSelection
 - CmdAbstract, [59](#)
- saveOrCheckPostCommandDocumentStateHash
 - CmdAbstract, [59](#)
- saveOrCheckPreCommandDocumentStateHash
 - CmdAbstract, [59](#)
- Segment, [270](#)
 - firstPoint, [271](#)
 - removeUnneededLines, [271](#)
- SegmentFactory, [271](#)
- SegmentLine, [272](#)
- selectOriginal
 - MainWindow, [253](#)
- setCurveSelected
 - BackgroundStateContext, [35](#)
- setDocumentAxesPointsRequired
 - CmdMediator, [70](#)
 - Document, [175](#)

- SettingsForGraph, [273](#)
- Spline, [273](#)
 - findSplinePairForFunctionX, [274](#)
 - interpolateCoeff, [274](#)
 - interpolateControlPoints, [275](#)
 - Spline, [274](#)
- SplineCoeff, [275](#)
- SplinePair, [276](#)
- stable
 - DocumentModelGridDisplay, [188](#)
 - DocumentModelGridRemoval, [190](#)
- StatusBar, [277](#)
- TestCorrelation, [278](#)
- TestFormats, [278](#)
- TestGraphCoords, [279](#)
- TestProjectedPoint, [280](#)
- TestSegmentFill, [280](#)
- TestSpline, [281](#)
- TestTransformation, [282](#)
- TestValidators, [282](#)
- transformsIsDefined
 - CallbackUpdateTransform, [49](#)
- Transformation, [283](#)
 - calculateTransformFromLinearCartesianPoints, [285](#)
- TransformationStateAbstractBase, [285](#)
- TransformationStateContext, [286](#)
- TransformationStateDefined, [287](#)
- TransformationStateUndefined, [287](#)
- TranslatorContainer, [288](#)
- TutorialButton, [289](#)
- TutorialButtonRect, [290](#)
- TutorialButtonText, [290](#)
- TutorialDlg, [291](#)
- TutorialStateAbstractBase, [292](#)
- TutorialStateAxisPoints, [293](#)
- TutorialStateChecklistWizardAbstract, [294](#)
- TutorialStateChecklistWizardLines, [295](#)
- TutorialStateChecklistWizardPoints, [296](#)
- TutorialStateColorFilter, [297](#)
- TutorialStateContext, [298](#)
 - requestDelayedStateTransition, [299](#)
 - requestImmediateStateTransition, [299](#)
- TutorialStateCurveSelection, [299](#)
- TutorialStateCurveType, [300](#)
- TutorialStateIntroduction, [301](#)
- TutorialStatePointMatch, [302](#)
- TutorialStateSegmentFill, [303](#)
- updateAfterCommand
 - GraphicsScene, [235](#)
- updateGraphicsLinesToMatchGraphicsPoints
 - GraphicsScene, [235](#)
 - MainWindow, [253](#)
- updateModelAxesChecker
 - Checker, [50](#)
- updatePointOrdinals
 - CoordSystem, [108](#)
- CoordSystemContext, [113](#)
- CoordSystemInterface, [117](#)
- Curve, [120](#)
- Document, [175](#)
- ViewPointStyle, [304](#)
- ViewPreview, [305](#)
- ViewProfile, [306](#)
- ViewProfileDivider, [307](#)
- ViewProfileScale, [308](#)
- ViewSegmentFilter, [308](#)