QCAD



QCAD User Reference Manual

QCAD 27.0.5 · 2026-11-11

Copyright © 2026 by RibbonSoft GmbH. All rights reserved.
qcad.org

Contents

1 File 1

1.1 New 1
1.2 New from Template... 1
1.3 Open... 2
1.4 Import... 2
1.5 Revert 3
1.6 Show in Explorer / Finder / File Manager (FI) 3
1.7 Save 4
1.8 Save As... 4
1.9 Close 4
1.10 Bitmap Export... (XB) 5
1.11 Quick SVG Export... (XS) 5
1.12 Advanced SVG Export... 5
1.13 SVG Import... 6
1.14 PDF Export... (XD) 6
1.15 Print Preview 7
1.16 Print... 7
1.17 Print Current View... 8
1.18 Print Window... 8

2 Edit 9

2.1 Undo [-] (OO) 9
2.2 Redo [-] (UU) 9
2.3 Delete (ER) 10
2.4 Cut 10
2.5 Cut with Reference (RT) 10
2.6 Copy (CP) 11
2.7 Copy with Reference (RC) 11
2.8 Paste (PS) 12
2.9 Paste along Entity (PE) 12
2.10 Duplicate (DP) 13
2.11 Quick Modify 13
2.11.1 Move Left (←) 13
2.11.2 Move Right (→) 14
2.11.3 Move Up (↑) 14
2.11.4 Move Down (↓) 14
2.11.5 Rotate Counter-Clockwise (F5) 15
2.11.6 Rotate Clockwise (F6) 15
2.12 Find/Replace... (RP) 16
2.13 Substitute Fonts... (FF) 16
2.14 Scale Text Heights... (MG) 17
2.15 Convert Drawing Unit (CU) 17
2.16 Left Click 17
2.17 Escape 18
2.18 Reset / Idle (QQ) 18
2.19 Drawing Preferences... 18
2.20 Application Preferences... 19

3 View ... 20
3.1 Draft Mode (DF) ... 20
3.2 Screen-based Linetypes (NL) ... 20
3.3 Anti-aliasing (NT) ... 21
3.4 Grid (GR) ... 21
3.5 Zoom ... 21
3.5.1 Zoom In ... 21
3.5.2 Zoom Out ... 22
3.5.3 Auto Zoom (ZA) ... 22
3.5.4 Zoom to Selection (ZS) ... 22
3.5.5 Previous View (ZV) ... 23
3.5.6 Window Zoom (ZW) ... 23
3.5.7 Pan Zoom (ZP) ... 23
3.6 Overlays ... 24
3.6.1 Direction (VD) ... 24
3.6.2 Order (VO) ... 24
3.6.3 Startpoint (VS) ... 25
3.7 Stored Views ... 25
3.7.1 Add View... (VA) ... 25
3.7.2 Remove View (VR) ... 26
3.7.3 Edit View... (VE) ... 26
3.7.4 Pick View (VP) ... 26
3.8 Isometric Grid Off (JO) ... 27
3.9 Top Projection (JT) ... 27
3.10 Right Projection (JR) ... 27
3.11 Left Projection (JL) ... 28
3.12 Layer List (GY) ... 28
3.13 Block List (GB) ... 28
3.14 View List (GV) ... 29
3.15 Property Editor (GP) ... 29
3.16 Selection Filter (GF) ... 30
3.17 Library Browser (GL) ... 31
3.18 Command Line (GM) ... 31
3.19 Clipboard Display (GC) ... 33
3.20 Status Bar (GS) ... 33
3.21 Layout Tabs (GT) ... 33
3.22 Focus on Command Line ... 33
3.23 Focus on Options Toolbar ... 34
3.24 Focus on Tool Matrix ... 34
3.25 Display Distance/Angle ... 34

4 Select ... 35
4.1 Deselect All (TN) ... 36
4.2 Select All (TA) ... 36
4.3 Select View (TV) ... 36
4.4 Invert Selection (TI) ... 37
4.5 (De-)Select Rectangular Area (TR) ... 37
4.6 (De-)Select Polygon (TP) ... 37
4.7 (De-)Select Contour (TC) ... 38
4.8 (De-)Select Intersected Entities (TX) ... 38
4.9 (De-)Select Layer (TL) ... 39

5 Draw 40
5.1 Point 40
5.1.1 Single Point (PO) 40
5.1.2 N Points on Line (PN) 40
5.1.3 MxN Points (PM) 41
5.2 Line 41
5.2.1 Line from 2 Points (LI) 42
5.2.2 Line from Angle (LA) 43
5.2.3 Horizontal Line (LH) 43
5.2.4 Vertical Line (LV) 43
5.2.5 Angle Bisector (LB) 44
5.2.6 Parallel (with Distance) (LP) 44
5.2.7 Parallel (through Point) (LG) 45
5.2.8 Tangent (Point, Circle) (LT1) 45
5.2.9 Tangent (Two Circles) (LT2) 45
5.2.10 Orthogonal / Tangent (LN) 46
5.2.11 Relative Angle (LR) 46
5.2.12 Orthogonal (LO) 47
5.2.13 Freehand Line (LF) 47
5.3 Arc 48
5.3.1 Center, Point, Angles (AR) 48
5.3.2 2 Points and Radius (AD) 49
5.3.3 2 Points and Angle (A2) 49
5.3.4 2 Points and Length (AL) 49
5.3.5 2 Points and Height (AH) 50
5.3.6 3 Points (A3) 50
5.3.7 Concentric (with Distance) (AC) 51
5.3.8 Concentric (through Point) (AG) 51
5.3.9 Tangentially Connected (AN) 51
5.3.10 Tangent, Point, Radius (AT) 52
5.4 Circle 52
5.4.1 Center, Point (CI) 53
5.4.2 Center, Radius (CR) 53
5.4.3 Center, Diameter (CA) 53
5.4.4 2 Points and Radius (CD) 54
5.4.5 2 Points (C2) 54
5.4.6 3 Points (C3) 54
5.4.7 Concentric (with Distance) (CC) 55
5.4.8 Concentric (through Point) (CG) 55
5.4.9 Tangent and 2 Points (CT1) 56
5.4.10 Tangent, Point, Radius (CTP) 56
5.4.11 2 Tangents and Point (CT2) 56
5.4.12 2 Tangents and Radius (CTR) 57
5.4.13 3 Tangents (CT3) 57
5.5 Ellipse 58
5.5.1 Ellipse (Center, Point, Ratio) (EP) 58
5.5.2 Ellipse Arc (EA) 58
5.5.3 Ellipse with Radii (EI) 59
5.5.4 Ellipse with Diameters (ED) 59
5.5.5 Parallel Curve (with Distance) (EC) 60
5.5.6 Parallel Curve (through Point) (EG) 60
5.5.7 Inscribed in Quadrilateral (EQ) 60
5.6 Spline 61
5.6.1 Spline (Control Points) (SP) 61

5.6.2 Spline (Fit Points) (SL) ... 62
5.6.3 Insert Fit Point (NI) ... 63
5.6.4 Append Fit Point(s) (NA) ... 63
5.6.5 Remove Fit Point (NR) ... 63
5.6.6 Simplify (NP) ... 64
5.7 Polyline ... 64
5.7.1 Draw Polyline (PL) ... 64
5.7.2 Polyline from Segments (OG) ... 65
5.7.3 Polyline from Selection (OC) ... 66
5.7.4 Partial Polyline Along Selection (OL) ... 66
5.7.5 Insert Node (OA) ... 67
5.7.6 Append Node(s) (OP) ... 67
5.7.7 Delete Node(s) (OD) ... 67
5.7.8 Remove Nodes between Two Nodes (OB) ... 68
5.7.9 Trim Segments (OT) ... 68
5.7.10 Relocate Start Point (OR) ... 69
5.7.11 Change Segment Type (OX) ... 69
5.7.12 Arc to Line Segments ... 69
5.7.13 Offset (OQ) ... 70
5.7.14 Morph (OM) ... 70
5.7.15 Boolean Operations (OW) ... 71
5.7.16 Simplify (OS) ... 72
5.7.17 Normalize Polylines (ON) ... 72
5.7.18 Insert Nodes at Self-Intersections (OV) ... 72
5.7.19 Logically close Polylines (OE) ... 73
5.7.20 Logically open Polylines (OJ) ... 73
5.8 Shape ... 73
5.8.1 Rectangle (RE) ... 74
5.8.2 Rectangle with Size (RS) ... 74
5.8.3 Rectangle (3 Points) (R3) ... 75
5.8.4 Polygon (Center, Corner) (PG1) ... 75
5.8.5 Polygon (2 Points of Side) (PG2) ... 76
5.8.6 Polygon (Center, Side) (PG3) ... 76
5.8.7 Polygon (Side, Side) (PG4) ... 76
5.8.8 Star (PT) ... 77
5.9 Hatch ... 77
5.9.1 Auto Hatch (HT) ... 78
5.9.2 Hatch from Selection... (HA) ... 79
5.9.3 Hatch from Segments... (HS) ... 80
5.9.4 Merge Hatches (HM) ... 80
5.10 Viewport ... 81
5.10.1 Add Viewport (VT) ... 81
5.10.2 Manage Viewport Layers... (VY) ... 82
5.11 Ring (RI) ... 83
5.12 Wipeout (WP) ... 83
5.13 Text... (TE) ... 84
5.14 Insert Bitmap... (IM) ... 85
5.15 Trace Bitmap... ... 85

6 Dimension ... 87

6.1 Aligned (DA) ... 87
6.2 Rotated (Linear) (DL) ... 88
6.3 Horizontal (DH) ... 88
6.4 Vertical (DV) ... 88

6.5 Baseline (DB) ... 89
6.6 Continue (DC) ... 90
6.7 Ordinate (X/Y) (DO) ... 90
6.8 Leader (DE) ... 91
6.9 Datum (DT) ... 92
6.10 Tolerance... (TO) ... 92
6.11 Radial (DR) ... 93
6.12 Diametric (DD) ... 93
6.13 Angular (DN) ... 94
6.14 Arc Length (DG) ... 95
6.15 Reset Label Position (DS) ... 95
6.16 Remove Style Overrides (DX) ... 96

7 Modify ... 97

7.1 Transform ... 97
7.1.1 Move / Copy (MV) ... 97
7.1.2 Rotate (RO) ... 98
7.1.3 Scale (SZ) ... 98
7.1.4 Mirror (MI) ... 99
7.1.5 Flip Horizontal (FH) ... 99
7.1.6 Flip Vertical (FV) ... 100
7.1.7 Move and Rotate (MR) ... 100
7.1.8 Rotate Two (R2) ... 101
7.1.9 Align Reference Points (AE) ... 102
7.1.10 Align (MA) ... 103
7.2 Offset ... 103
7.2.1 Offset (with Distance) (OF) ... 103
7.2.2 Offset (through Point) (OH) ... 105
7.3 Trim and Divide ... 105
7.3.1 Trim (RM) ... 105
7.3.2 Trim Both (TM) ... 106
7.3.3 Lengthen / Shorten (LE) ... 106
7.3.4 Stretch (SS) ... 107
7.3.5 Clip to Rectangle (CLR) ... 108
7.3.6 Divide (DI) ... 108
7.3.7 Split Entities (MS) ... 108
7.3.8 Break out Segment (D2) ... 109
7.3.9 Break out Manual (B2) ... 109
7.3.10 Auto Trim (AX) ... 110
7.3.11 Break out Gap (D3) ... 110
7.4 Corners ... 111
7.4.1 Chamfer / Bevel (CH) ... 111
7.4.2 Round (RN) ... 112
7.5 Reverse (RV) ... 112
7.6 Entities ... 113
7.6.1 Edit Text (MT) ... 113
7.6.2 Edit Tolerance (ML) ... 113
7.7 Explode (XP) ... 114
7.8 Draw Order ... 114
7.8.1 Bring to Front (MF) ... 114
7.8.2 Send to Back (MB) ... 115
7.8.3 Reorder (MO) ... 115
7.8.4 Order Connected Entities (MC) ... 116

7.9 Detection ... 116
7.9.1 Detect Duplicates (MD) ... 116
7.9.2 Detect Zero-Length Entities (MZ) ... 117
7.10 Projection ... 118
7.10.1 Orthographic Projection (Cylindrical) (PR) ... 118
7.10.2 Isometric Projection (PJ) ... 119
7.10.3 Isometric Projection (Cylindrical) (PC) ... 119
7.10.4 Matrix Transformation (MX) ... 120
7.11 Property Painter (PI) ... 121

8 Snap ... 122

8.1 Free (SF) ... 122
8.2 Grid (SG) ... 122
8.3 End (SE) ... 123
8.4 On Entity (ST) ... 123
8.5 Perpendicular (SU) ... 123
8.6 Tangential (SB) ... 124
8.7 Center (SC) ... 124
8.8 Reference (SR) ... 124
8.9 Middle (SM) ... 125
8.10 Middle Manual (SN) ... 125
8.11 Distance (SD) ... 126
8.12 Distance Manual (SH) ... 126
8.13 Intersection (SI) ... 127
8.14 Intersection Manual (SY) ... 127
8.15 Auto (SA) ... 128
8.16 Coordinate (SX) ... 128
8.17 Polar Coordinate (SO) ... 129
8.18 X/Y from Points (PX) ... 129
8.19 Y/X from Points (PY) ... 130
8.20 Selection Reference Points (SK) ... 130
8.21 Restrictions Off (EN) ... 130
8.22 Restrict Orthogonally (EO) ... 131
8.23 Restrict Horizontally (EH) ... 131
8.24 Restrict Vertically (EV) ... 131
8.25 Restrict Angle or Length (EL) ... 132
8.26 Set Relative Zero (RZ) ... 132
8.27 Lock Relative Zero (RL) ... 132
8.28 Lock Snap (SQ) ... 133

9 Info ... 134

9.1 Position (IO) ... 134
9.2 Relative Position (IV) ... 134
9.3 Polar Position (IL) ... 135
9.4 Relative Polar Position (IT) ... 135
9.5 Distance Point to Point (IP) ... 135
9.6 Distance Entity to Point (IE) ... 136
9.7 Distance Entity to Entity (IN) ... 136
9.8 Angle (IA) ... 136
9.9 Total Length of Selection (IS) ... 137
9.10 Polygonal Area (IR) ... 137
9.11 Arc/Circle/Ellipse Area (IC) ... 138

9.12 Polyline Area (II) 140

10 Layer 141

10.1 Toggle Visibility (YV) 141
10.2 Toggle Frozen Status (YF) 141
10.3 Toggle Lock Status (YL) 142
10.4 Show Only Active (YO) 142
10.5 Show All Layers (YS) 142
10.6 Hide All Layers (YH) 143
10.7 Thaw All Layers (YW) 143
10.8 Freeze All Layers (YZ) 143
10.9 Lock All Layers (YK) 144
10.10 Unlock All Layers (YN) 144
10.11 Add Layer... (YA) 144
10.12 Add Sublayer... (YU) 145
10.13 Set Parent Layer... (YX) 145
10.14 Create Layer from Selection... (YT) 145
10.15 Move Selection to Current Layer (YJ) 146
10.16 Duplicate Layer... (YY) 146
10.17 Delete Layer(s) (YR) 146
10.18 Purge Unused Layers (YP) 147
10.19 Edit Layer... (YE) 147
10.20 Activate Layer of Entity (YB) 147
10.21 Select Layer Entities (YC) 148
10.22 Deselect Layer Entities (YD) 148
10.23 Layer States 148
10.23.1 Add... (YM) 148

11 Block 149

11.1 Explode (XP) 149
11.2 Toggle Visibility (BV) 149
11.3 Show All Blocks (BS) 149
11.4 Hide All Blocks (BH) 150
11.5 Add Empty Block... (BA) 150
11.6 Create Block from Selection (BC) 150
11.7 Add Layout Block... (BL) 151
11.8 Remove Block (BR) 151
11.9 Purge Unused Blocks (BP) 152
11.10 Rename Block... (BN) 152
11.11 Edit Block (BE) 152
11.12 Edit Block from Reference (BD) 153
11.13 Edit Block In-place (BB) 153
11.14 Duplicate Block... (BY) 154
11.15 Return to Main Drawing (EM) 154
11.16 Select Block References (BK) 155
11.17 Deselect Block References (BX) 155
11.18 Insert Block (BI) 155
11.19 Insert External Reference... (XR) 156
11.20 Open External Reference File 157
11.21 Reload External References 157
11.22 Bind External Reference 158
11.23 Create Library Item (BT) 158

11.24 Attributes ... 158
11.24.1 Define Block Attribute (UA) ... 159
11.24.2 Synchronize Attributes (UY) ... 159

12 Window ... 161

12.1 Close All ... 161
12.2 Next ... 161
12.3 Previous ... 161
12.4 Full Screen ... 162

13 Misc ... 163

13.1 Import/Export ... 163
13.1.1 Import Points... ... 163
13.1.2 Import Points with Labels... ... 163
13.1.3 Draw from CSV... ... 164
13.1.4 Batch2Pdf visible layout blocks (2PF) ... 164
13.1.5 KML Export... ... 164
13.2 Select ... 164
13.2.1 By Handle... (TH) ... 165
13.2.2 By Color (TF) ... 165
13.2.3 Select Next Entity ... 165
13.2.4 Select Previous Entity ... 165
13.2.5 Store Selection (TS) ... 166
13.2.6 Use Selection (TU) ... 166
13.3 Draw ... 166
13.3.1 Boolean Operations (Legacy) ... 166
13.3.2 Horizontal Construction Line (XH) ... 167
13.3.3 Vertical Construction Line (XV) ... 168
13.3.4 Centerline Cross (CS) ... 168
13.3.5 Counter (C12) ... 168
13.3.6 Text Along Entity (TG) ... 168
13.3.7 Text Aligned Leader (2T) ... 169
13.3.8 Leader from Selected Text (T2L) ... 169
13.3.9 Box Joint from 2 Points (LJ) ... 169
13.3.10 Dovetail Joint from 2 Points ... 170
13.3.11 Polyline break symbol from 2 points (S3) ... 170
13.3.12 Draw Fms Lanes (PF) ... 170
13.3.13 Pole of Line and Circle ... 170
13.3.14 Power Center of Three Circles ... 171
13.3.15 Radical Axis of 2 Circles ... 171
13.3.16 Similarity Axes of 3 Circles ... 171
13.3.17 Draw Bounding Box ... 171
13.3.18 Polygon Hull ... 171
13.4 Modify ... 172
13.4.1 Scale in X ... 172
13.4.2 Scale in Y ... 172
13.4.3 Scale One Way ... 172
13.4.4 Offset (Legacy) ... 173
13.4.5 Polyline Offset (Alternative) ... 174
13.4.6 Offset (through Point, Legacy) ... 174
13.4.7 Set Attributes by Layer ... 174
13.4.8 Set Attributes by Block ... 175
13.4.9 Purge Unused Layers and Blocks ... 175
13.4.10 Flatten Drawing to 2D ... 175

13.4.11 Reload Linetype Definitions ... 175
13.4.12 Flush Undo History ... 175
13.4.13 Set Font to Arial ... 175
13.4.14 Regenerate Hatch Patterns ... 176
13.4.15 Convert Splines to Bézier Segments ... 176
13.4.16 Convert Straight Splines to Lines ... 176
13.4.17 Reset Splines Tangents ... 176
13.4.18 Split polyline at arc / line segment change ... 176
13.5 Projection ... 177
13.5.1 Map Projection ... 177
13.6 Layer ... 177
13.6.1 Make all White Layers Black ... 177
13.6.2 Make all Black Layers White ... 177
13.6.3 Create New Layer from Hatches ... 178
13.7 Block ... 178
13.7.1 Fix Block Names ... 178
13.7.2 Export Block List... ... 178
13.7.3 Export Block References List... ... 178
13.7.4 List Block Attributes ... 178
13.7.5 Prefix Block Names... ... 179
13.8 Information ... 179
13.8.1 Store Positions (TT) ... 179
13.8.2 Use Positions (TY) ... 179
13.8.3 Bearing / Azimuth ... 179
13.9 Development ... 180
13.9.1 Script Shell (GE) ... 180
13.9.2 Run Script (XC) ... 180
13.10 Reset Configuration ... 180
13.11 Manage RCC Plugins ... 181

14 CAM ... 182

14.1 CAM Configuration... (KK) ... 182
14.2 CAM Export (KX) ... 182
14.3 G-Code Import... ... 183
14.4 Display Toolpaths (KV) ... 183
14.5 Add Tool (KA) ... 183
14.6 Edit Tool (KE) ... 184
14.7 Remove Tool (KR) ... 184
14.8 Export Tools (KO) ... 184
14.9 Import Tools (KI) ... 185
14.10 Add Profile Toolpath (KP) ... 185
14.11 Add Pocket Toolpath (KC) ... 186
14.12 Add Drill Toolpath (KL) ... 187
14.13 Edit Toolpath (KD) ... 188
14.14 Remove Toolpath (KM) ... 189
14.15 Move Toolpath Up ... 189
14.16 Move Toolpath Down ... 189
14.17 Regenerate Toolpath ... 190
14.18 Regenerate Toolpath from Tool ... 190
14.19 Regenerate All Toolpaths ... 190
14.20 Update Toolpath from Selection ... 191
14.21 Add Tab (KT) ... 191
14.22 Add Workpiece (KW) ... 192

14.23 Show CAM Tool List (GO) ... 192
14.24 Show CAM Toolpath List (GA) ... 193
14.25 Show Simulation Panel (GI) ... 193
14.26 Nesting (NE) ... 194
14.27 Set Nesting Quantity ... 195

15 Command Line Tools ... 196

15.1 merge ... 196
15.2 bmp2dwg ... 196
15.3 bbox ... 197
15.4 dwg2bmp ... 198
15.5 dwg2csv ... 199
15.6 dwg2dwg ... 200
15.7 dwg2maptiles ... 201
15.8 dwg2pdf ... 202
15.9 dwg2svg ... 204
15.10 dwgexplode ... 205
15.11 dwghatch ... 206
15.12 dwghatcharea ... 206
15.13 dwginfo ... 207
15.14 dwgmapconvert ... 208
15.15 extracttextinfo ... 209
15.16 svg2dwg ... 209

16 Tutorials ... 211

16.1 Command Line Tools ... 211
16.1.1 Generating Map Tiles ... 211
16.2 Development ... 214
16.2.1 Command Line Tool Scripts ... 214
16.2.2 Creating a QCAD Plugin ... 216
16.2.3 Interactive Script Actions ... 219
16.2.4 Running Scripts from a File ... 224
16.2.5 The Simple API ... 225
16.2.6 The Script Shell ... 245
16.3 Drawing ... 246
16.3.1 Custom Lineweights ... 247
16.3.2 Importing Points with Labels from CSV ... 248
16.3.3 Morphing ... 254
16.3.4 Reusing Drawing Positions ... 259
16.3.5 Using a List of Coordinates ... 261
16.4 Fonts ... 262
16.4.1 Editing CXF Fonts ... 262
16.5 Installation ... 264
16.5.1 Installing QCAD under Linux ... 264
16.5.1.1 Installing QCAD from a Flatpak ... 265
16.5.1.2 Installing QCAD from the Installer ... 271
16.5.1.3 Installing QCAD on a Chromebook ... 272
16.5.2 Installing QCAD on macOS ... 280
16.5.3 QCAD Portable ... 281
16.5.4 Installing QCAD under Windows ... 282
16.6 Layers ... 285
16.6.1 Layers, Blocks and Attributes ... 285
16.6.2 Working with Layer States ... 286

16.6.3 Working with Layers ... 289
16.7 MCP ... 292
16.7.1 Using QCAD with Claude Desktop ... 292
16.8 Part Library and Blocks ... 296
16.8.1 Working with the Part Library ... 296
16.9 Preferences ... 299
16.9.1 Adjusting Colors ... 300
16.9.2 User Interface Language ... 300
16.9.3 Resetting the QCAD Configuration ... 303
16.10 Printing ... 305
16.10.1 Printing ... 305
16.10.2 Working with Layout Blocks and Viewports ... 309
16.11 Properties ... 312
16.11.1 Property Editor ... 313
16.12 QCAD/CAM ... 315
16.12.1 Nesting ... 315
16.12.2 QCAD/CAM Postprocessor ... 320
16.12.3 QCAD/CAM ... 325
16.13 Video Tutorials ... 338

1 File

Toolbar / Icon:



Commands: `filemenu`

Description

The file menu contains all tools to open, save, import, export and print files.

1.1 New

Toolbar / Icon:




Menu: `File > New`
Shortcuts: `Ctrl+N (Mac: ⌘N) | Ctrl+T (Mac: ⌘T)`
Commands: `new`

Description

Create new, empty drawings using this command. New drawings are completely empty (no entities, only one default layer, no blocks). As a frequent user of QCAD, you might instead want to create drawing templates with the layers and blocks you use the most and probably a drawing border. Once you have such a template you can load that instead of starting a new drawing from scratch. Please refer also to the command New from Template to learn how to use drawing templates.

1.2 New from Template...

Toolbar / Icon:




Add-on: **QCAD Professional**
Menu: `File > New from Template...`
Shortcut: `Ctrl+Shift+N (Mac: ⇧⌘N)`
Commands: `newfromtemplate`

Description

This tool opens a drawing template as a starting point for a new drawing. Any regular drawing can be used as drawing template if it is stored in the library/templates directory or another template directory configured in the application preferences dialog.

1.3 Open...

Toolbar / Icon:



Menu: `File > Open...`
Shortcut: `Ctrl+O (Mac: ⌘O)`
Commands: `open`

Description

Use this command to open a drawing file. The file is loaded and shown in a new window within the QCAD application window. You can switch between different loaded drawings using the menu "Windows" or the tabs at the top of the drawing area.

QCAD Professional

PDF Files

If you open a PDF file, QCAD show a dialog. In this dialog, you can specify the page to import, if you want to import images and if clipping should be applied. Imported images are stored in a new folder called "pdf_images" located where the PDF file is located. Without clipping, all vector data is imported from the PDF file. This might include data outside the paper.

Editing CXF Fonts

QCAD comes with a collection of line fonts in its own CXF format. Line fonts are sometimes used in CAD instead of TrueType fonts. Line fonts are also often used for engraving or other further processing.
QCAD can load and edit fonts in CXF format. To load an existing font, simply use the menu File > Open and choose the format filter "CXF Font Files (*.cxf)". Then choose the font file you want to open. The fonts that come with QCAD are stored in directory "fonts" of your QCAD installation.
The font glyphs (representation of the various letters and symbols) are loaded into blocks. Make sure the block list is visible (View > Block List), so you can navigate through the font glyphs.
Each block has a name in the format "[Hexadecimal Unicode] [Symbol]", for example "0x0041 A" for a capital "A" with Unicode 65 (41 in hexadecimal).
By convention, a capital "A" should be constructed with a height of 9 drawing units. When the font is used for a text, it is scaled by QCAD according to the text height chosen.
Glyphs should have their reference point 0,0 at the lower left corner of the base line. In the CXF format, glyphs can contain lines, arcs and polylines. Splines may be used, but are converted into polylines during export.

1.4 Import...

Toolbar / Icon:



Menu: `File > Import...`
Shortcut: `Ctrl+Shift+I (Mac: ⇧⌘I)`
Commands: `import`

Description

The import command inserts a drawing from a file on disk into the current drawing.
All layers that are used by the imported drawing are added to the layer list of the current drawing. Existing layers with the same name can be overwritten if desired ("overwrite layers" option in the options tool bar).
Block references of the imported drawing will be inserted together with the block definitions they refer to. Blocks in the current drawing can be overwritten if desired ("overwrite blocks" option in the options tool bar).
The options tool bar also offers some tools to scale, rotate or flip the imported drawing while positioning it.

Usage

Set the target point for the imported drawing with the mouse or enter a coordinate in the command line. The target point corresponds to the absolute zero point of the imported drawing.

1.5 Revert

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `File > Revert`
Shortcut: `Ctrl+R (Mac: ⌘R)`
Commands: `revert`

Description

Reloads the current drawing from disk. All unsaved changes are lost. Note that this operation cannot be undone. This function can for example be used to refresh the view if QCAD is used as file viewer for files which are modified by another application or process.

1.6 Show in Explorer / Finder / File Manager (FI)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `File > Show in Explorer / Finder / File Manager`
Shortcut: `F, I`
Commands: `showfile`

Description

This tool can be used to open the folder that contains the current drawing in a file manager window. This can be Explorer under Windows, Finder under macOS or a file manager such as Nautilus under Linux. Explorer and Finder will also highlight the drawing file in the folder.

1.7 Save

Toolbar / Icon:



Menu: `File > Save`
Shortcut: `Ctrl+S (Mac: ⌘S)`
Commands: `save`

Description

This command saves the current drawing to the same file it was loaded from. If you want to save a newly created drawing or save the current drawing to a new file, use the menu File - Save As instead. You will then be asked for a file name before the drawing is saved.

1.8 Save As...

Toolbar / Icon:



Menu: `File > Save As...`
Shortcut: `Ctrl+Shift+S (Mac: ⇧⌘S)`
Commands: `saveas`

Description

Saves the current drawing as a new file. The dialog that is shown also allows you to choose the format and format version that you want to use to save the file.

1.9 Close

Toolbar / Icon:



Menu: `File > Close`
Shortcut: `Ctrl+W (Mac: ⌘W)`
Commands: `close`

Description

This command closes the current drawing. If the current drawing contains unsaved changes, you will be given the option to save the drawing before closing it or to discard those changes.

1.10 Bitmap Export... (XB)

Toolbar / Icon:



Menu: `File > Bitmap Export...`
Shortcut: `X, B`
Commands: `bitmapexport`

Description

This tool exports the current drawing as a bitmap file.

Usage

A dialog for selecting the output file is presented. After the file name has been specified, a second dialog asks you for the bitmap size and the preferred background color.
Note that exporting to bitmaps with a very large size can take a long time, depending on your hardware. The maximum size for bitmaps is width x height ≤ 2.147.483.647.

1.11 Quick SVG Export... (XS)

Toolbar / Icon:



Add-on: **QCAD Professional**
Menu: `File > Quick SVG Export…`
Shortcut: `X, S`
Commands: `svgexport`

Description

Exports the current drawing to an SVG file. While the SVG format is not ideal for storing CAD data, it is very popular for storing various types of vector graphics mainly for presentation and sometimes for further processing.
This tool uses the parameters that can be configured in the application preferences under Load/Save - Quick SVG Export.

1.12 Advanced SVG Export...

Toolbar / Icon:



Add-on: **QCAD Professional**
Menu: `File > Advanced SVG Export...`
Commands: `svgexportas`

Description

Exports the current drawing in the SVG format. An advanced export dialog is shown that can be used to configure various options for the export.
Most notably, the option "Preserve Geometry" can be used to export the drawing for further processing. The visual appearance of the output might not exactly match that of the CAD drawing in this case, but as much of the geometry as possible is carried across to the SVG file.

1.13 SVG Import...

Toolbar / Icon:



Menu: `File > SVG Import...`
Commands: `svgimport`

Description

This tool imports an SVG file into the current drawing. Note that only a very limited set of SVG tags and arguments is supported. The idea is to import as much of the raw geometry in the SVG file as possible.

1.14 PDF Export... (XD)

Toolbar / Icon:



Menu: `File > PDF Export…`
Shortcut: `X, D`
Commands: `pdf`

Description

This tool exports the current drawing to a PDF file. Exporting to PDF is very similar to printing a drawing. You will most likely have to adjust your drawing scale and drawing position as well as page settings before exporting a drawing to PDF.
For this reason, QCAD automatically switches to the print preview when this tool is used. If the print preview does not match your expectations, cancel the export, adjust the page settings and start the PDF export again.

1.15 Print Preview

Toolbar / Icon:



Menu: `File > Print Preview`
Shortcut: `Ctrl+Shift+P (Mac: ⇧⌘P)`
Commands: `printpreview`

Description

The print preview shows how your drawing will look like in the printout. Use the combo box in the options tool bar to adjust the scale factor of your drawing. This option does not actually scale or otherwise modify your drawing entities, it only sets up the scale factor that is applied for fitting your drawing on paper.
The buttons right of the scale combo box can be used to automatically fit your drawing on the chosen paper or to automatically add pages, so your entire drawing can be printed on multiple pages.
After choosing the "Move Paper Position" tool, you can move the paper around freely by dragging it with the left mouse button. Click the right mouse button or hit the escape key to terminate the tool.
The options tool bar offers some other tools to quickly change some common drawing preferences that are also accessible in the drawing preferences dialog.

1.16 Print...

Toolbar / Icon:



Menu: `File > Print...`
Shortcut: `Ctrl+P (Mac: ⌘P)`
Commands: `print`

Description

Use this command to print a drawing. Make sure to always switch to the print preview before printing to check if the drawing scale and position is set up as desired.
A setup dialog is shown before anything is printed. Use this dialog to choose the printer and adjust printer specific settings.

1.17 Print Current View...

Toolbar / Icon:



Menu: `File > Print Current View...`
Shortcut: `Ctrl+Alt+P (Mac: ⌘Alt+P)`
Commands: `printview`

Description

Use this command to print the drawing portion that is currently shown in the graphics view.

1.18 Print Window...

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `File > Print Window...`
Shortcut: `Ctrl+Alt+Shift+P (Mac: ⌘Alt+⇧P)`
Commands: `printwindow`

Description

This tool prints a rectangular window of the drawing. The print preview is setup in such a way that the chosen rectangular area fits exactly on the printable area of the paper.

2 Edit

Toolbar / Icon:



Commands: `editmenu`

Description

The edit menu contains tools for basic editing such as copy / paste and deleting.

2.1 Undo [-] (OO)

Toolbar / Icon:



Menu: `Edit > Undo [-]`
Shortcuts: `Ctrl+Z (Mac: ⌘Z) | O, O`
Commands: `undo | oops`

Description

This tool takes back the last change of the drawing. QCAD can take back more than one command. For example if you have just created a line and a circle and you want to undo both, use the undo command twice.

2.2 Redo [-] (UU)

Toolbar / Icon:



Menu: `Edit > Redo [-]`
Shortcuts: `Ctrl+Shift+Z (Mac: ⇧⌘Z) | U, U`
Commands: `redo | uu`

Description

Redo reverts the effects of the previous Undo Action.

2.3 Delete (ER)

Toolbar / Icon:



Menu: `Edit > Delete`
Shortcuts: `Del | Backspace | E, R`
Commands: `delete | er`

Description

This tool deletes the entities that are currently selected.

2.4 Cut

Toolbar / Icon:



Menu: `Edit > Cut`
Shortcut: `Ctrl+X (Mac: ⌘X)`
Commands: `cut`

Description

The Cut tool works like the Copy tool with the only difference that the selected entities are removed from the current drawing after being copied to the clipboard.

2.5 Cut with Reference (RT)

Toolbar / Icon:



Menu: `Edit > Cut with Reference`
Shortcuts: `Ctrl+Shift+X (Mac: ⇧⌘X) | R, T`
Commands: `cutwithreference | rt`

Description

The Cut with Reference tool works like the Copy with Reference tool with the only difference that the selected entities are removed from the current drawing after being copied to the clipboard.

2.6 Copy (CP)

Toolbar / Icon:



Menu: `Edit > Copy`
Shortcuts: `Ctrl+C (Mac: ⌘C) | C, P`
Commands: `copy | cp`

Description

QCAD offers a set of tools to copy entities from one drawing to another, similar like other applications.

Usage

1. Use the selection tools to prepare a selection of entities you want to copy to the clipboard.
2. Choose the copy tool.

The standard copy tool of QCAD acts like the copy tool of other applications. It does not allow you to choose a reference point for your selection but instead computes the reference point automatically at the center of the selection. This is not always desirable as you most likely want to place the pasted entities at a precisely defined position. For this, QCAD offers and alternative, more CAD specific copy tool: Copy with Reference.

The selection is now on the QCAD internal clipboard and can be pasted into the same or a different drawing that is opened with the same running instance of QCAD. Note that if you quit the QCAD application, the contents of the clipboard is lost. The QCAD clipboard can only be accessed by QCAD itself. You cannot exchange drawing entities with other applications through copy and paste. Please use the export and import functionality of QCAD for this.

2.7 Copy with Reference (RC)

Toolbar / Icon:



Menu: `Edit > Copy with Reference`
Shortcuts: `Ctrl+Shift+C (Mac: ⇧⌘C) | R, C`
Commands: `copywithreference | rc`

Description

Copies the current selection to the QCAD clipboard. This tool lets you specify a reference point that is used to position the selection when pasting it into a drawing.

Usage

1. Use the selection tools to prepare a selection of entities you want to copy to the clipboard.
2. Start the Copy with Reference tool.
3. Specify the reference point that you want to use when pasting the selection.

2.8 Paste (PS)

Toolbar / Icon:



Menu: `Edit > Paste`
Shortcuts: `Ctrl+V (Mac: ⌘V) | P, S`
Commands: `paste | ps`

Description

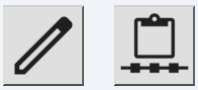
The paste command inserts the entities that were previously copied to the clipboard. Paste is especially useful to transfer entities from one drawing to another.
All layers that are used by the clipboard contents are pasted into the layer list of the current drawing. Existing layers with the same name can be overwritten if desired ("overwrite layers" option in the options tool bar).
Block references that are on the clipboard will be pasted together with the block definitions they refer to. Blocks in the current drawing can be overwritten if desired ("overwrite blocks" option in the options tool bar).
The options tool bar also offers some tools to scale, rotate or flip the clipboard contents while pasting.

Usage

Set the target point for the pasted entities with the mouse or enter a coordinate in the command line. The target point corresponds to the reference point that was chosen when copying or cutting the entities if Copy / Cut with Reference was used. If Cut or Copy without reference point was used, the center of the selection is used as reference point.

2.9 Paste along Entity (PE)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Paste along Entity`
Shortcuts: `Ctrl+Shift+V (Mac: ⇧⌘V) | P, E`
Commands: `pastealongentity`

Description

This tool pastes the clipboard contents along an existing line, arc, polyline or spline entity. The clipboard is pasted multiple times in such a way that the reference point of the clipboard contents is placed on the entity.

The clipboard can be inserted multiple times with a given distance between items or a given number of times, equally spread along the entity.

The "Align" option can be used to rotate the clipboard to align its angle with the angle along the entity.

The "Include end points" option causes the first and last items to be placed on the start and end point of the entity.

In addition, the clipboard can be scaled or rotated while pasting.

2.10 Duplicate (DP)

Toolbar / Icon:

Menu: `Edit > Duplicate`
Shortcut: `D, P`
Commands: `duplicate | dp`

Description

This tool duplicates the selected entities and selects the created copy.

Usage

1. Use the selection tools to prepare a selection of entities you want to duplicate.
2. Choose the duplication tool.

2.11 Quick Modify

Toolbar / Icon:



Shortcut: `W, Q`
Commands: `quickmodifymenu`

Description

Tools for quickly modifying entities, such as trimming, extending and stretching, often with a single click.

2.11.1 Move Left (←)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Quick Modify > Move Left`
Shortcut: ←

Description

This function can be used to quickly move the current selection by a previously configured distance. The distance can be configured under Edit > Application Preferences

> Edit > Move.

2.11.2 Move Right (→)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Quick Modify > Move Right`
Shortcut: →

Description

This function can be used to quickly move the current selection by a previously configured distance. The distance can be configured under Edit > Application Preferences

> Edit > Move.

2.11.3 Move Up (↑)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Quick Modify > Move Up`
Shortcut: ↑

Description

This function can be used to quickly move the current selection by a previously configured distance. The distance can be configured under Edit > Application Preferences

> Edit > Move.

2.11.4 Move Down (↓)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Quick Modify > Move Down`
Shortcut: ↓

Description

This function can be used to quickly move the current selection by a previously configured distance. The distance can be configured under Edit > Application Preferences

> Edit > Move.

2.11.5 Rotate Counter-Clockwise (F5)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Quick Modify > Rotate Counter-Clockwise`
Shortcut: `F5`
Commands: `rotateccw`

Description

This function can be used to quickly rotate the current selection by a previously configured angle or to rotate entities on the fly while pasting or inserting a block or library item.

Preferences

Edit > Application Preferences > Edit > Rotate

- *Angle step*
  The rotation angle.
- *Shift multiplier*
  Factor that is applied to the angle when the Shift key is pressed while rotating.
- *Alt multiplier*
  Factor that is applied to the angle when the Alt key is pressed while rotating.
- *Center of Rotation*
  This preference only has an effect when rotating a selection.
  - *Centroid of convex bounding polygon (default):*
    The center of rotation is always in the same location at about the center of a selection. With this preference, rotations are reversible. This means that a selection that is rotated clockwise and then rotated back counter-clockwise is back at the same position again.
  - *Relative zero point:*
    Rotate around the current location of the relative zero point.
  - *Center of bounding box:*
    Rotate around the center of the bounding box of the current selection. This type of rotation is not generally reversible.

2.11.6 Rotate Clockwise (F6)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Quick Modify > Rotate Clockwise`
Shortcut: `F6`
Commands: `rotatecw`

Description

This function can be used to quickly rotate the current selection by a previously configured angle or to rotate entities on the fly while pasting or inserting a block or library item.

Preferences

Edit > Application Preferences > Edit > Rotate

- *Angle step*
  The rotation angle.
- *Shift multiplier*
  Factor that is applied to the angle when the Shift key is pressed while rotating.
- *Alt multiplier*
  Factor that is applied to the angle when the Alt key is pressed while rotating.
- *Center of Rotation*
  This preference only has an effect when rotating a selection.
  - *Centroid of convex bounding polygon (default):*
    The center of rotation is always in the same location at about the center of a selection. With this preference, rotations are reversible. This means that a selection that is rotated clockwise and then rotated back counter-clockwise is back at the same position again.
  - *Relative zero point:*
    Rotate around the current location of the relative zero point.
  - *Center of bounding box:*
    Rotate around the center of the bounding box of the current selection. This type of rotation is not generally reversible.

2.12 Find/Replace... (RP)

Toolbar / Icon:



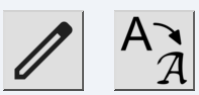
Add-on: QCAD Professional
Menu: `Edit > Find/Replace…`
Shortcuts: `Ctrl+F (Mac: ⌘F) | R, P`

Description

This tool replaces all occurrences of a text string in text entities with another text string. If one or more entities are selected, the search is limited to only those entities. If nothing is selected, all text entities are searched. Entities on locked or invisible layers are not searched.

2.13 Substitute Fonts... (FF)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Substitute Fonts…`
Shortcut: `F, F`

Description

With this tool, you can easily replace all uses of a particular font with another one. This is particularly useful if an imported drawing uses fonts which are not available on your system or which do not support all glyphs that are required.

2.14 Scale Text Heights... (MG)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Edit > Scale Text Heights…`
Shortcut: `M, G`
Commands: `scaletextheight | mg`

Description

This tool scales the text height of all selected texts, without changing the position of the texts.

2.15 Convert Drawing Unit (CU)

Toolbar / Icon:




Menu: `Edit > Convert Drawing Unit`
Shortcut: `C, U`
Commands: `convertunit | cu`

Description

Converts the current drawing to a different drawing unit, scaling all geometry accordingly.

2.16 Left Click

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Edit > Left Click`
Shortcut: `Ctrl+Return (Mac: ⌘Return)`
Commands: `leftclick`

Description

Triggers the default left-click action, so the current tool's primary click can be invoked from a command or keyboard shortcut.

2.17 Escape

Toolbar / Icon:



Menu: `Edit > Escape`
Shortcut: `Esc`
Commands: `escape`

Description

The escape key can be used to terminate the current tool and return to the idle state of QCAD. If the current tool requires multiple steps, escape returns to the previous step of the tool. To terminate the tool, escape might have to be pressed multiple times in this case.

2.18 Reset / Idle (QQ)

Toolbar / Icon:



Menu: `Edit > Reset / Idle`
Shortcut: `Q, Q`
Commands: `reset | qq`

Description

This tool terminates all active tools and returns QCAD to its idle state in which you can use the mouse cursor to select entities, drag and drop entities, etc.

2.19 Drawing Preferences...

Toolbar / Icon:



Menu: `Edit > Drawing Preferences…`
Shortcut: `Ctrl+I (Mac: ⌘I)`
Commands: `drawingpreferences`

Description

The drawing preferences dialog allows you to change preferences that affect various aspects of the current drawing. These preferences are stored in your drawing.

2.20 Application Preferences...

Toolbar / Icon:



Menu: `Edit > Application Preferences…`
Shortcut: `Ctrl+, (Mac: ⌘,)`
Commands: `preferences`

Description

In the application preferences dialog, you can adjust the look and behavior of QCAD and its tools and graphics views.
QCAD also stores some of these settings inside the drawings. Those settings can be changed in the application preferences as default settings for new drawings. In other words, they don't affect existing drawings but only drawings that are created after the preferences have been changed.

3 View

Toolbar / Icon:



Shortcut: `W, V`
Commands: `viewmenu`

Description

The view menu contains all tools for zooming, panning and adjusting the display.

3.1 Draft Mode (DF)

Toolbar / Icon:



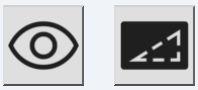
Menu: `View > Draft Mode`
Shortcut: `D, F`
Commands: `draftmode | df`

Description

Toggles the draft mode of the current drawing. In draft mode, all lines are shown with a width of 1 pixel. Large texts are simplified. Use the draft mode if your drawing becomes very large and it takes long to redraw it.

3.2 Screen-based Linetypes (NL)

Toolbar / Icon:



Menu: `View > Screen-based Linetypes`
Shortcut: `N, L`
Commands: `screenlinetype | nl`

Description

Toggles the screen-based linetype mode of the current drawing.
If this mode is activated, linetypes are optimized for a computer screen. All lines are displayed with a screen based line width (lines don't get wider when zooming in) and all patterns are shown as fixed, pixel based patterns on screen (dashes don't get longer when zooming in).
If this mode is off, line widths and linetypes are shown in drawing units (default).

3.3 Anti-aliasing (NT)

Toolbar / Icon:



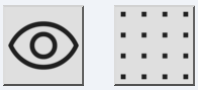
Menu: `View > Anti-aliasing`
Shortcut: `N, T`
Commands: `antialiasing | nt`

Description

Toggles anti-aliasing for the current drawing. With anti-aliasing, skewed lines, arcs and texts are displayed smoother.

3.4 Grid (GR)

Toolbar / Icon:



Menu: `View > Grid`
Shortcut: `G, R`

Description

Toggles the grid visibility of the current drawing.

3.5 Zoom

Toolbar / Icon:



Shortcut: `W, Z`
Commands: `zoommenu`

Description

Tools for zooming and panning the drawing view.

3.5.1 Zoom In

Toolbar / Icon:



Menu: `View > Zoom > Zoom In`
Shortcuts: `+ | = | Ctrl++ (Mac: ⌘+)`
Commands: `zoomin`

Description

This tool increases the current viewing factor. The same effect can also be achieved by turning the mouse wheel away from you.

3.5.2 Zoom Out

Toolbar / Icon:



Menu: `View > Zoom > Zoom Out`
Shortcuts: `-` | `Ctrl+-` (Mac: ⌘-)
Commands: `zoomout`

Description

This tool decreases the current viewing factor. The same effect can also be achieved by turning the mouse wheel toward you.

3.5.3 Auto Zoom (ZA)

Toolbar / Icon:



Menu: `View > Zoom > Auto Zoom`
Shortcut: `Z, A`
Commands: `zoomauto | za`

Description

Scales the drawing view so that all entities that are on visible layers fit on the screen.

3.5.4 Zoom to Selection (ZS)

Toolbar / Icon:



Menu: `View > Zoom > Zoom to Selection`
Shortcut: `Z, S`
Commands: `zoomselection | zs`

Description

This tool zooms to the current selection, ensuring that all selected entities are visible.

3.5.5 Previous View (ZV)

Toolbar / Icon:



Menu: `View > Zoom > Previous View`
Shortcut: `Z, V`
Commands: `zoomprevious | zv`

Description

Shows the last used view. Use this to switch back to the previous view for example after performing an auto zoom or after zooming in with the window zoom.

3.5.6 Window Zoom (ZW)

Toolbar / Icon:



Menu: `View > Zoom > Window Zoom`
Shortcut: `Z, W`
Commands: `zoomwindow | zw`

Description

This tool offers a quick way to zoom in on a certain area of the drawing.

Usage

1. Specify the first corner of the area you want to view. Press the left mouse button at that corner and keep it down.
2. Drag the mouse to the second corner of the area.
3. Let go of the mouse button at the second corner.

Note: Alternatively you can specify the first and the second corner each with a single mouse click instead of dragging the mouse around with the mouse button pressed.

3.5.7 Pan Zoom (ZP)

Toolbar / Icon:



Menu: `View > Zoom > Pan Zoom`
Shortcut: `Z, P`
Commands: `zoompan | zp`

Description

Panning means moving (scrolling) around in a drawing. The quickest way to do so is using the middle mouse button and move the view similar like you would move a paper around: press the middle mouse button and hold it while moving the drawing around. If your mouse does not have

a middle mouse button you can press the Control key (Mac OS X: Command key) and use the left mouse button instead.
Alternatively, you can activate this tool and then move the view with the left mouse button without pressing any keys.
Click the right mouse button when you are done.

3.6 Overlays

Toolbar / Icon:



Commands: `overlaymenu`

Description

Tools for managing overlays — reference drawings shown on top of the current drawing.

3.6.1 Direction (VD)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Overlays > Direction`
Shortcut: `V, D`
Commands: `overlaydirection | vd`

Description

This tool shows a direction overlay over the current drawing. An arrow is shown over each entity with a direction. Entities with a direction are: lines, arcs, ellipse arcs, splines and polylines.

3.6.2 Order (VO)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Overlays > Order`
Shortcut: `V, O`
Commands: `overlayorder | vo`

Description

This tool shows an order overlay over the current drawing. An order number is shown over each entity.

3.6.3 Startpoint (VS)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Overlays > Startpoint`
Shortcut: `V, S`
Commands: `overlaystartpoint | vs`

Description

This tool shows a startpoint overlay over the current drawing. An arrow is shown over each startpoint of an entity with a start / end point. Entities with a startpoint are: lines, arcs, ellipse arcs, splines and polylines.

3.7 Stored Views

Toolbar / Icon:



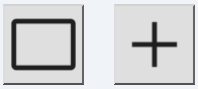
Commands: `storedviewsmenu`

Description

Tools for storing and recalling named views (specific zoom and pan positions) of the drawing.

3.7.1 Add View... (VA)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Stored Views > Add View…`
Shortcut: `V, A`
Commands: `viewadd`

Description

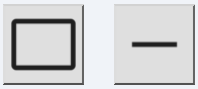
Saves the current view (zoom and offset) as an entry in the view list. Saved views can quickly be recovered by clicking the view in the view list instead of zooming and scrolling.

Usage

1. Zoom in on a part of your drawing you frequently want to view.
2. Launch this tool.
3. Enter a name for the view and click OK.

3.7.2 Remove View (VR)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Stored Views > Remove View`
Shortcut: `V, R`
Commands: `viewremove`

Description

Removes the active view from the view list.

3.7.3 Edit View... (VE)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Stored Views > Edit View…`
Shortcut: `V, E`
Commands: `viewedit`

Description

This tool edits the name of a previously saved view.

Usage

1. Activate that view you want to edit in the view list.
2. Launch this tool.
3. Edit the name of the view and click "OK".

3.7.4 Pick View (VP)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Stored Views > Pick View`
Shortcut: `V, P`
Commands: `viewpick`

Description

This tool updates an existing saved view to match the current view.

Usage

1. Activate the view you want to update in the view list.
2. Zoom and pan to display the new view you want to use.

3. Launch this tool to update the active view in the view list.

3.8 Isometric Grid Off (JO)

Toolbar / Icon:



Menu: `View > Isometric Grid Off`
Shortcut: `J, O`
Commands: `isometricgridoff | jo`

Description

Switches the isometric grid off and returns to the normal orthogonal grid.

3.9 Top Projection (JT)

Toolbar / Icon:



Menu: `View > Top Projection`
Shortcut: `J, T`
Commands: `isometricgridtop | jt`

Description

Sets the isometric grid and crosshair to the top projection plane.

3.10 Right Projection (JR)

Toolbar / Icon:



Menu: `View > Right Projection`
Shortcut: `J, R`
Commands: `isometricgridright | jr`

Description

Sets the isometric grid and crosshair to the right projection plane.

3.11 Left Projection (JL)

Toolbar / Icon:



Menu: `View > Left Projection`
Shortcut: `J, L`
Commands: `isometricgridleft | jl`

Description

Sets the isometric grid and crosshair to the left projection plane.

3.12 Layer List (GY)

Toolbar / Icon:



Menu: `View > Layer List`
Shortcut: `G, Y`
Commands: `gy`

Description

This tool shows / hides the layer list widget.

QCAD Professional

By default, the layer list shows columns for the layer visibility (on/off) and the layer lock (locked/unlocked). More columns for other layer states (frozen, snappable, plottable) can be shown under Edit > Application Preferences > Layer List.

3.13 Block List (GB)

Add-on: **QCAD Professional**
Menu: `View > Block List`
Shortcut: `G, B`
Commands: `gb`

Description

Provides the block list dock widget in the Professional edition.

3.14 View List (GV)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > View List`
Shortcut: `G, V`
Commands: `gv`

Description

Shows or hides the view list dock widget, which lists the stored views of the drawing.

3.15 Property Editor (GP)

Toolbar / Icon:



Menu: `View > Property Editor`
Shortcut: `G, P`
Commands: `gp`

Description

Shows or hides the property editor dock widget for viewing and editing the properties of the selected entities.

Usage

1. Use the selection tools to select one or multiple entities whose properties you want to view or change.
2. The properties of the selection are displayed in the property editor.

If a single entity is selected, all of its properties are shown, including its geometry (for example the start and end point of a line or the center and radius of a circle). If multiple entities are selected, the property editor shows the properties that are common to all selected entities. A property with different values across the selection is displayed as a mixed field.

To change a property, enter a new value into the corresponding field in the property editor and confirm your input (Enter or Tab).

The change is immediately applied to all selected entities. Geometric properties, attributes such as layer, color, line weight and line type, as well as custom properties can be edited this way.

If nothing is selected, the property editor is empty.

3.16 Selection Filter (GF)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Selection Filter`
Shortcut: `G, F`
Commands: `gf`

Description

The selection filter widget can be used to select entities based on their type or their properties. With the selection filter, you can for example easily select all circles with a radius between 3 and 4 or all text entities with a height of 5 units.
You can also remove entities from a selection or select only those entities that match the filter and are already selected (intersection).

Usage

1. Display the selection filter widget using View > Selection Filter.
2. Choose the entity type to filter by or select "Any" to match any entity type.
3. Choose the property name of the property to filter by.
4. Choose the comparison operator. Depending on the property type chosen, you can use logical comparisons, mathematical comparisons (=, ≠, >, ≥, < ≤) or string comparisons (is, is not, contains, contains not, starts with, ends with, regular expression match).
5. Choose the value to compare with.
6. Click one of the buttons at the bottom to apply the filter.
   The first button at the left selects all entities that match the filter. The existing selection is ignored.
   The second button adds the entities that match to the selection.
   The third button deselects the matching entities.
   The last button selects only those entities that match the filter and are already selected (boolean intersection).

Examples

- **To select all texts with text height 5**
  Entity type: Text
  Property: Text Height
  Comparison: = (equal to)
  Value: 5
- **To select all texts with text height larger than 5**
  Entity type: Text
  Property: Text Height
  Comparison: > (greater than)
  Value: 5
- **To select all circles with a radius between 5 and 6 (inclusive)**
  Entity type: Circle
  Property: Radius

Comparison: ≥ (greater than or equal to)
Value: 5
Click first button at the left
Entity type: Circle
Property: Radius
Comparison: ≤ (less than or equal to)
Value: 6
Click last button at the right

Special properties

The property color matches entities by their color value. This can be "By Layer" in which case the displayed color is irrelevant. To match by the displayed color (regardless if the value is "By Layer" or fixed), choose the property "Displayed Color" instead.

The boundary and size properties can be used to select entities based on their location or size.

3.17 Library Browser (GL)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Library Browser`
Shortcut: `G, L`
Commands: `gl`

Description

The part libraries of QCAD are collections of symbols and other drawing files that can be used in your drawings. To insert an item from a part library, you need to show the library browser first using this tool.
Parts can be inserted into your drawing using drag and drop. Once the mouse cursor is inside the drawing area, you can use the regular snap tools to position the inserted part and the options tool bar to scale, flip or rotate the part.

3.18 Command Line (GM)

Toolbar / Icon:



Menu: `View > Command Line`
Shortcut: `G, M`
Commands: `gm`

Description

The command line of QCAD allows you to start commands, enter coordinates or enter values such as distances or radii.

Entering Coordinates

Absolute coordinates are entered in the format "x,y":

40,5

Relative coordinates are entered in the format "@x,y":

@10,6

Absolute polar coordinates are entered in the format "distance<angle":

10<30

Relative polar coordinates are entered in the format "@distance<angle":

@10<45

When entering coordinates or values, mathematical expressions may be used to calculate a coordinate based on known values. For example the coordinate 10,5 may also be entered as:

5+5,30/6

Calculator

The command line can also be used as a calculator. To do this, enter a mathematical expression proceeded by an equal sign:

=3+4
7

Variables may be used to store values:

=a=5+6
11
=a/2
5.5

Mathematical Expressions

Mathematical constants available are:

PI, LN2, LN10, LOG2E, LOG10E, SQRT1_2, SQRT2

Mathematical functions available are:

abs, ceil, floor, exp, log, max, min, pow, sqrt, random, round, rad2deg, deg2rad, sin, cos, tan, asin, acos, atan, atan2, log10, log1p, log2, sign, cosh, sinh, tanh, acosh, asinh, atanh, expm1, hypot, cbrt, trunc

Most of these constants and functions are standard ECMAScript (JavaScript) functions and documented online. In standard ECMAScript, these functions are part of the Math class so the function *abs* would have to written as *Math.abs*. In the QCAD command line, you can leave away the *Math.* part for convenience. Trigonometric functions (sin, cos, tan, asin, acos, atan, atan2) accept or return angles in degrees. If you prefer the radian versions of these functions, use the original *Math.* functions instead.

The functions *rad2deg* and *deg2rad* can be used to convert angles between radian and degrees.

3.19 Clipboard Display (GC)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Clipboard Display`
Shortcut: `G, C`
Commands: `gc`

Description

Shows or hides a dock widget that previews the current contents of the clipboard.

3.20 Status Bar (GS)

Menu: `View > Status Bar`
Shortcut: `G, S`
Commands: `gs`

Description

Shows or hides the application status bar at the bottom of the window.

3.21 Layout Tabs (GT)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Layout Tabs`
Shortcut: `G, T`

Description

Provides the layout tabs used to switch between model space and the print layouts of the drawing.

3.22 Focus on Command Line

Toolbar / Icon:



Menu: `View > Focus on Command Line`
Shortcuts: `Space | Ctrl+M (Mac: ⌘M)`

Description

This command activates the command line for input, for example to enter a coordinate when specifying a point. This is usually done by pressing the Space key.

3.23 Focus on Options Toolbar

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `View > Focus on Options Toolbar`
Shortcut: `Meta+.`

Description

Moves the keyboard focus to the options toolbar so its fields can be edited without using the mouse.

3.24 Focus on Tool Matrix

Toolbar / Icon:



Menu: `View > Focus on Tool Matrix`
Shortcut: `Meta+Shift+. (Mac: Meta+⇧.)`

Description

This command activates the filter (search) field of the tool matrix for input.

3.25 Display Distance/Angle

Toolbar / Icon:



Menu: `View > Display Distance/Angle`
Shortcut: `F8`

Description

Displays the distance and angle between two picked points.

4 Select

Toolbar / Icon:



Shortcut: `W, S`
Commands: `selectionmenu`

Description

Cross Selection

Some selection tools can be used in cross selection mode. In this mode, not only entities that are completely inside a given area are selected but also entities that are only partly inside the area. This selection is also known as "cross selection".

Selection Mode

Some selection tools allow you to choose a selection mode in the options toolbar. The selection modes that are available are:

- Replace selection:

  Replaces the current selection with the new selection made by the active selection tool (default).
- Add to selection:

  Adds the selection to the current selection.
- Remove from selection:

  Removes (subtracts) the selection from the current selection.
- Intersect:

  Only entities that were already selected and that match the criteria of the selection tool are selected.

4.1 Deselect All (TN)

Toolbar / Icon:



Menu: `Select > Deselect All`
Shortcuts: `T, N | Ctrl+K (Mac: ⌘K) | Ctrl+D (Mac: ⌘D) | Ctrl+Shift+A (Mac: ⇧⌘A)`
Commands: `deselectall | tn`

Description

Deselects all entities. Use this tool to make sure that no entities are selected. Alternatively, you can also click into an empty area of your drawing.

4.2 Select All (TA)

Toolbar / Icon:



Menu: `Select > Select All`
Shortcuts: `T, A | Ctrl+A (Mac: ⌘A)`
Commands: `selectall | ta`

Description

Selects all entities on all visible, unlocked layers.

4.3 Select View (TV)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Select > Select View`
Shortcut: `T, V`
Commands: `selectview | tv`

Description

Selects all entities that are completely inside the currently visible area.

4.4 Invert Selection (TI)

Toolbar / Icon:

Menu: `Select > Invert Selection`
Shortcut: `T, I`
Commands: `selectinvert | invertselection | ti`

Description

Selects all entities that are not currently selected and deselects all selected entities.

4.5 (De-)Select Rectangular Area (TR)

Toolbar / Icon:

Menu: `Select > (De-)Select Rectangular Area`
Shortcuts: `T, R | T, W`
Commands: `selectrectangle | selectwindow | tr | tw`

Description

Selects all entities that are within a given rectangular area.

Usage

1. Choose the selection mode in the options tool bar.
2. Tick the check box for cross selection to also select entities that intersect the given rectangle.
3. Use the mouse to draw the selection rectangle. All matching entities are selected or deselected depending on the selection mode.

4.6 (De-)Select Polygon (TP)

Toolbar / Icon:

Add-on: **QCAD Professional**
Menu: `Select > (De-)Select Polygon`
Shortcut: `T, P`
Commands: `selectpolygon | tp`

Description

Selects or deselects all entities that are within a given polygon.

Usage

1. Choose the selection mode in the options tool bar.
2. Tick the check box for "cross selection" to also select entities that intersect the given polygon.
3. Use the mouse to draw the selection polygon. All matching entities are selected or deselected depending on the selection mode.

4.7 (De-)Select Contour (TC)

Toolbar / Icon:



Menu: `Select > (De-)Select Contour`
Shortcut: `T, C`
Commands: `selectcontour | tc`

Description

Selects or deselects entities that are connected to each other and form a contour (closed or open).

Usage

1. Choose the selection mode in the options tool bar.
2. Use the mouse to pick one entity of the contour you want to select. The tool then searches from the endpoints of the selected entity in both directions for entities that are directly or indirectly connected to this entity. All connected entities are selected or deselected depending on the chosen selection mode.

Instead of using this tool, you can simply double-click an entity to select it and all connected entities.

4.8 (De-)Select Intersected Entities (TX)

Toolbar / Icon:



Menu: `Select > (De-)Select Intersected Entities`
Shortcut: `T, X`
Commands: `selectintersected | tx | ux`

Description

Selects or deselects all entities that are intersected by a line.

Usage

1. Choose the selection mode in the options tool bar.
2. Click the start point of the intersection line.
3. Click the end point of the intersection line.
4. All intersected entities are selected or deselected depending on the selection mode.

4.9 (De-)Select Layer (TL)

Toolbar / Icon:

Menu: `Select > (De-)Select Layer`
Shortcut: `T, L`
Commands: `selectlayerbyentity | tl`

Description

Selects or deselects all entities on the same layer as a chosen entity.

Usage

1. Choose the selection mode in the options tool bar.
2. Use the mouse to pick one entity on the layer you want to (de-)select. All entities that are on the same layer as the entity you have picked are selected or deselected depending on the selection mode.

5 Draw

Description

The draw menu contains construction tools for the various entity types.

5.1 Point

Toolbar / Icon:



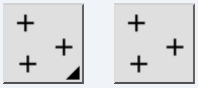
Shortcut: `W, G`
Commands: `pointmenu`

Description

Points mark a single position in the drawing. Their display size and style can be configured in the drawing preferences. Points are often used as reference or construction markers.

5.1.1 Single Point (PO)

Toolbar / Icon:



Menu: `Draw > Point > Single Point`
Shortcut: `P, O`
Commands: `point | po`

Description

This command is used to draw single points. Points are visually represented by a small cross.

Usage

- Use the mouse to specify the location of the point or enter a coordinate in the command line.

5.1.2 N Points on Line (PN)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Point > N Points on Line`
Shortcut: `P, N`
Commands: `npoints | pn`

Description

With this tool you can draw a given number of points that are evenly distributed on a line.

Usage

1. Enter the number of points you want to create.
2. Specify the start and end point of the line with the mouse or by entering the coordinates in the command line.

5.1.3 MxN Points (PM)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Point > MxN Points`
Shortcut: `P, M`
Commands: `mnpoints | pm`

Description

With this tool you can draw a grid of N times M points. The points are evenly distributed in an irregular quadrilateral.

Usage

1. Enter the number of points along the first side of the quadrilateral as "M" in the options tool bar.
2. Enter the number of points along the second side of the quadrilateral as "N" in the options tool bar.
3. Specify the four corner points of the quadrilateral with the mouse or by entering their coordinates in the command line.

5.2 Line

Toolbar / Icon:



Shortcut: `W, L`
Commands: `linemenu`

Description

Lines are straight entities defined by two points. The line tools offer various ways to construct a line, for example from a start point to an end point, parallel to an existing entity, tangentially to circles or arcs or orthogonally to another line.

Line Types

Most line tools allow you to choose the type of line to create in the options toolbar. The line types available are:

- Auto:



Automatically create a line of the same type as a chosen other line. This applies for the parallel line tool.

- Line Segment:



Creates line segments from a start point to an end point.

- Infinite Line:



Creates infinite lines that go through two given points. These are sometimes called X-lines or construction lines.

- Rays:



Creates rays from a given start point, through another point with an infinite length.

5.2.1 Line from 2 Points (LI)

Toolbar / Icon:




Menu: `Draw > Line > Line from 2 Points`
Shortcut: `L, I`
Commands: `line | ln | li | l`

Description

This tool lets you draw a sequence of one or more straight lines.

Usage

1. Choose the desired line type in the options toolbar.
2. Specify the start point of the first line segment. You can use the mouse or enter a coordinate in the console.
3. Specify the endpoint of the first line segment.
4. Specify the endpoints of additional line segments. Click the "Close" button in the options tool bar to close the sequence:



If you need to undo a single line segment, you can do so by clicking the "Undo" button:



5. You can set the length or angle of the line to fixed values using the appropriate inputs in the options toolbar.

5.2.2 Line from Angle (LA)

Toolbar / Icon:

Menu: `Draw > Line > Line from Angle`
Shortcut: `L, A`
Commands: `lineangle | la`

Description

Use this tool to create lines with a given angle. The line is usually trimmed to the desired length after creating it.

Usage

1. Choose the desired line type in the options toolbar.
2. Enter the angle and length of the line in the options tool bar.
3. In the options tool bar, choose the reference point on the line which you want to use for positioning the line. 'Start' means that the line will have its start point at the point where you position it.
4. Place the line with the mouse or by entering a coordinate in the command line.

5.2.3 Horizontal Line (LH)

Toolbar / Icon:

Menu: `Draw > Line > Horizontal Line`
Shortcut: `L, H`
Commands: `linehorizontal | lh`

Description

Use this tool to create horizontal lines. This tool is used like the tool for lines at any given angle, except that the angle is fixed to be horizontal.

5.2.4 Vertical Line (LV)

Toolbar / Icon:

Menu: `Draw > Line > Vertical Line`
Shortcut: `L, V`
Commands: `linevertical | lv`

Description

Use this tool to create vertical lines. This tool is used like the tool for lines at a given angle, except that the angle is fixed to be vertical.

5.2.5 Angle Bisector (LB)

Toolbar / Icon:

Menu: `Draw > Line > Angle Bisector`
Shortcut: `L, B`
Commands: `linebisector | bisector | lb`

Description

Use this tool for creating angle bisectors between two line entities.

Usage

1. Choose the desired line type in the options toolbar.
2. In the options tool bar enter the length of the bisector(s), starting from the intersection point of the two lines. In the second text box, choose the number of angle bisectors you want to create. The default is '1' but you can also create multiple bisectors as shown here:



3. Click the first line entity which limits the angle.
4. Click the second line entity with the mouse cursor on that side of the line on which you want to create the angle bisector(s). The preview shows the angle bisector(s) before they are created.

5.2.6 Parallel (with Distance) (LP)

Toolbar / Icon:

Menu: `Draw > Line > Parallel (with Distance)`
Shortcuts: `L, P | P, A`
Commands: `lineparallel | lineoffset | parallel | par | lp | pa`

Description

With this tool you can create parallels to existing lines (or concentric arcs and circles).

Usage

1. Choose the desired line type in the options toolbar.
2. Enter the distance of the concentric or parallel entity from the original entity in the options tool bar shown at the top.
3. Enter the number of parallel or concentric entities to create in the options tool bar.
4. Click the base entity. The parallel or concentric entities are created on that side on which the mouse cursor is located while selecting the entity.

5.2.7 Parallel (through Point) (LG)

Toolbar / Icon:



Menu: `Draw > Line > Parallel (through Point)`
Shortcut: `L, G`
Commands: `lineparallelthrough | lineoffsetthrough | parallelthrough | lg`

Description

With this tool you can create parallels to existing lines or concentric arcs and circles. The parallel or concentric arc or circle goes through a given point.

Usage

1. Choose the desired line type in the options toolbar.
2. Enter the number of entities to create in the options tool bar.
3. Click the base entity.
4. Click the position through which the first parallel or concentric arc or circle runs.

5.2.8 Tangent (Point, Circle) (LT1)

Toolbar / Icon:



Menu: `Draw > Line > Tangent (Point, Circle)`
Shortcut: `L, T, 1`
Commands: `linetangent | tangent | lt1`

Description

Create tangents from a coordinate to an existing arc, circle or ellipse entity with this tool.

Usage

1. Choose the desired line type in the options toolbar.
2. Use the mouse to specify the location of the start point of the line or enter a coordinate in the command line.
3. Click the entity to which you want to create the tangent. Usually, two tangents are possible. When moving the mouse around, you can see a preview of the tangent that will be created.

5.2.9 Tangent (Two Circles) (LT2)

Toolbar / Icon:



Menu: `Draw > Line > Tangent (Two Circles)`
Shortcut: `L, T, 2`
Commands: `linetangent2 | tangent2 | lt2`

Description

Create tangents from one existing arc or circle entity to another one with this tool.

Usage

1. Choose the desired line type in the options toolbar.
2. Click the first arc or circle entity to which you want to create the tangent.
3. Click the second arc or circle entity to which you want to create the tangent. Usually, four tangents are possible. When moving the mouse around, you can see a preview of the tangent that will be created.

5.2.10 Orthogonal / Tangent (LN)

Toolbar / Icon:



Menu: `Draw > Line > Orthogonal / Tangent`
Shortcut: `L, N`
Commands: `lineorthogonaltangent | orthotangent`

Description

Use this tool to create a line that is orthogonal to another line and tangent to an existing arc, circle or ellipse entity.

Usage

1. Choose the desired line type in the options toolbar.
2. Click the line entity to which the new line should be orthogonal.
3. Click the arc, circle or ellipse entity to which the line should be a tangent. There are two possible solutions. When moving the mouse around, you can see a preview of the tangent that will be created.

5.2.11 Relative Angle (LR)

Toolbar / Icon:



Menu: `Draw > Line > Relative Angle`
Shortcut: `L, R`
Commands: `linerelativeangle | lr`

Description

Create lines with a relative angle to existing entities with this tool. The existing entity can be a line or an arc / circle. Lines with a relative angle of 0 degrees to an arc are tangents or parallels to tangents. Lines with a relative angle of 90 degrees to an arc or line are orthogonal lines.

Usage

1. Choose the desired line type in the options toolbar.
2. Enter the relative angle in the options tool bar.

3. Click the existing base entity.
4. Place the line with the mouse or enter a coordinate in the command line.

5.2.12 Orthogonal (LO)

Toolbar / Icon:



Menu: `Draw > Line > Orthogonal`
Shortcut: `L, O`
Commands: `lineorthogonal | lo`

Description

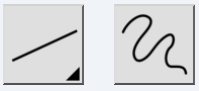
This tool lets you create lines orthogonal to an existing base entity. The base entity can be a line, arc or circle.

Usage

1. Choose the desired line type in the options toolbar.
2. Click the existing base entity.
3. Place the line with the mouse or enter a coordinate in the command line.

5.2.13 Freehand Line (LF)

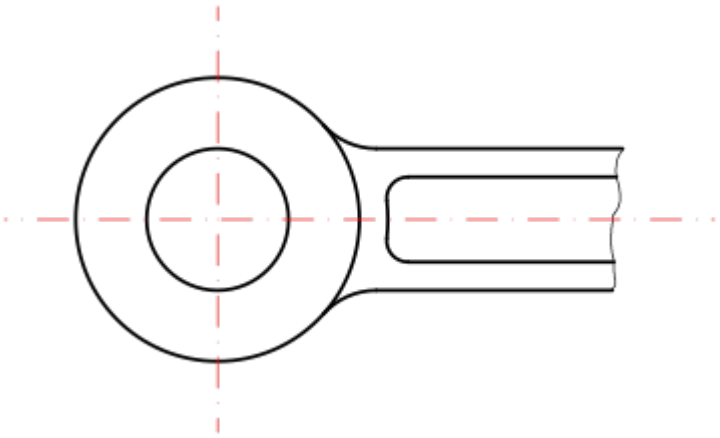
Toolbar / Icon:



Menu: `Draw > Line > Freehand Line`
Shortcut: `L, F`
Commands: `linefree | freehand | lf`

Description

With this tool you can draw freehand lines. This tool is usually used only sparingly since technical drawings require absolute accuracy. However, on some occasions this tool might come in handy, for example for broken edges like shown here:



Usage

1. Place the mouse at the start point of the freehand line, press the left mouse button and keep it down.
2. Now move the mouse around to draw the line. At the endpoint of the freehand line, let go of the mouse button.

5.3 Arc

Toolbar / Icon:



Shortcut: `W, A`
Commands: `arcmenu`

Description

Arcs are circular segments defined by a center, a radius and a start and end angle. The arc tools offer various ways to construct an arc, for example from a center point and a radius, through three points or tangentially to existing entities.

5.3.1 Center, Point, Angles (AR)

Toolbar / Icon:



Menu: `Draw > Arc > Center, Point, Angles`
Shortcut: `A, R`
Commands: `arcc | ar`

Description

Draws an arc from its center point, a point on the arc line, start angle and end angle.

Usage

1. Choose the direction of the arc in the options toolbar.
2. Set the center of the arc using the mouse or enter a coordinate in the command line.
3. Define the radius by clicking a point on the arc or by entering a coordinate into the command line. You may also enter the radius in the command line.
4. Set the start angle with the mouse or by entering a coordinate or the angle in the command line.
5. Set the end angle the same way as the start angle.

QCAD Professional

- Instead of clicking a point for the radius, start angle and end angle, you can also define these values in the appropriate fields in the options toolbar or enter them directly on your keyboard. If you enter the values directly, there is no need to click into the options toolbar fields or the command line.

5.3.2 2 Points and Radius (AD)

Toolbar / Icon:



Menu: `Draw > Arc > 2 Points and Radius`
Shortcut: `A, D`
Commands: `arcradius | ad`

Description

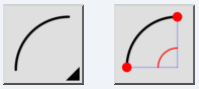
Draws an arc using the start point, end point and the radius.

Usage

1. Type the arc radius into the options tool bar and choose the arc direction (clockwise or counter-clockwise) and the solution (larger arc or smaller arc).
2. Specify the start point of the arc.
3. Specify the end point of the arc. If the end point is too far away from the start point for a solution, the closest solution is drawn (a half circle with the given radius and direction).

5.3.3 2 Points and Angle (A2)

Toolbar / Icon:



Menu: `Draw > Arc > 2 Points and Angle`
Shortcut: `A, 2`
Commands: `arc2 | a2`

Description

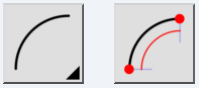
Draws an arc using the start point, end point and the included angle.

Usage

1. Type the included angle into the options tool bar and choose the arc direction (clockwise or counter-clockwise).
2. Specify the start point of the arc.
3. Specify the end point of the arc.

5.3.4 2 Points and Length (AL)

Toolbar / Icon:



Menu: `Draw > Arc > 2 Points and Length`
Shortcut: `A, L`
Commands: `arclength | al`

Description

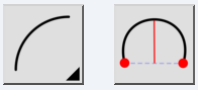
Draws an arc using the start point, end point and the arc length.

Usage

1. Type the arc length into the options tool bar and choose the arc direction (clockwise or counter-clockwise).
2. Specify the start point of the arc.
3. Specify the end point of the arc. If the end point is too far away from the start point for a solution, nothing is drawn.

5.3.5 2 Points and Height (AH)

Toolbar / Icon:



Menu: `Draw > Arc > 2 Points and Height`
Shortcut: `A, H`
Commands: `archeight | ah`

Description

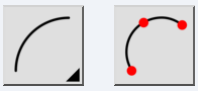
Draws an arc using the start point, end point and the arc height.

Usage

1. Type the arc height into the options tool bar and choose the arc direction (clockwise or counter-clockwise).
2. Specify the start point of the arc.
3. Specify the end point of the arc.

5.3.6 3 Points (A3)

Toolbar / Icon:



Menu: `Draw > Arc > 3 Points`
Shortcut: `A, 3`
Commands: `arc3 | a3`

Description

If you know the start point, the end point and a point in between on the arc line, you can use this tool to draw an arc.

Usage

1. Set the start point using the mouse or enter a coordinate in the command line.
2. Set the second point to a known point on the arc line.
3. Set the endpoint of the arc.

5.3.7 Concentric (with Distance) (AC)

Toolbar / Icon:

Menu: `Draw > Arc > Concentric (with Distance)`
Shortcut: `A, C`
Commands: `arcconcentric | ac`

Description

With this tool you can create one or multiple concentric arcs with a given distance to an existing arc.

Usage

1. Enter the distance of the concentric arc from the original base arc in the options tool bar.
2. Enter the number of concentric arcs to create in the options tool bar.
3. Click the base arc. The concentric arc(s) are created on that side on which the mouse cursor is located while clicking the base arc.

5.3.8 Concentric (through Point) (AG)

Toolbar / Icon:

Menu: `Draw > Arc > Concentric (through Point)`
Shortcut: `A, G`
Commands: `arcconcentricthrough | ag`

Description

With this tool you can create concentric arcs that go through a specified point.

Usage

1. Enter the number of concentric arcs to create in the options tool bar.
2. Click the base arc.
3. Click the location through which the first concentric arc should go.

5.3.9 Tangentially Connected (AN)

Toolbar / Icon:

Menu: `Draw > Arc > Tangentially Connected`
Shortcut: `A, N`
Commands: `arctangential | an`

Description

Draws an arc that connects tangentially to an existing arc or line.

Usage

1. Enter the radius of the arc in the options tool bar.
2. Click the base entity close to the end where the arc should be connected.
3. Specify the end point of the arc.

QCAD Professional

- Instead of an arc or a line, you can also choose an ellipse arc or spline as base entity.
- Entities inside blocks can be chosen as base entities as well.
- Tick the "Radius" check box in the options toolbar to fix the arc radius to the chosen value. Otherwise, the arc is constructed from the start point to the position of the mouse cursor.

5.3.10 Tangent, Point, Radius (AT)

Toolbar / Icon:

Menu: `Draw > Arc > Tangent, Point, Radius`
Shortcut: `A, T`
Commands: `arctangentpointradius | at`

Description

Draws an arc with a given radius, tangential to an entity and through a point.

Usage

1. Enter the radius in the options tool bar.
2. Specify the tangential entity.
3. Specify the point on the arc.
4. Choose the desired solution by clicking close to it with the left mouse button.

5.4 Circle

Toolbar / Icon:



Shortcut: `W, C`
Commands: `circlemenu`

Description

Circles are defined by a center point and a radius. The circle tools offer various ways to construct a circle, for example from a center point and a radius, through two or three points or tangentially to existing entities.

5.4.1 Center, Point (CI)

Toolbar / Icon:



Menu: `Draw > Circle > Center, Point`
Shortcut: `C, I`
Commands: `circle | ci`

Description

Draws a circle with a given center and a point on the circle line.

Usage

1. Set the center of the circle using the mouse or enter a coordinate in the command line.
2. Define the radius by clicking a point on the circle line or by entering a coordinate into the command line. You may also enter the radius in the command line.

5.4.2 Center, Radius (CR)

Toolbar / Icon:



Menu: `Draw > Circle > Center, Radius`
Shortcut: `C, R`
Commands: `circlecr | cr`

Description

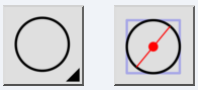
This tool lets you create circles with a given center and radius.

Usage

1. Enter the radius in the option tool bar.
2. Set the center of the circle using the mouse or enter a coordinate in the command line.

5.4.3 Center, Diameter (CA)

Toolbar / Icon:



Menu: `Draw > Circle > Center, Diameter`
Shortcut: `C, A`
Commands: `circlediameter | ca`

Description

This tool lets you create circles with a given center and diameter.

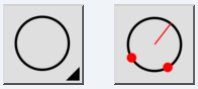
Usage

1. Enter the diameter in the option tool bar.

2. Set the center of the circle using the mouse or enter a coordinate in the command line.

5.4.4 2 Points and Radius (CD)

Toolbar / Icon:



Menu: `Draw > Circle > 2 Points and Radius`
Shortcut: `C, D`
Commands: `circleradius | cd`

Description

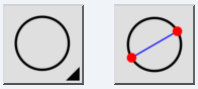
Draws a circle from two points on the circle line and a radius.

Usage

1. Type the radius into the options tool bar.
2. Specify the first point on the circle line.
3. Specify the second point on the circle line.

5.4.5 2 Points (C2)

Toolbar / Icon:



Menu: `Draw > Circle > 2 Points`
Shortcut: `C, 2`
Commands: `circle2p | c2`

Description

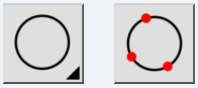
Draws a circle with two diametrically opposed points.

Usage

1. Set the first point using the mouse or enter a coordinate in the command line.
2. Set the second point.

5.4.6 3 Points (C3)

Toolbar / Icon:



Menu: `Draw > Circle > 3 Points`
Shortcut: `C, 3`
Commands: `Circle3P | c3`

Description

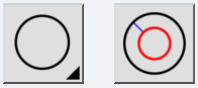
Draws a circle from three known points on the circle line.

Usage

1. Set the first point on the circle line using the mouse or enter a coordinate in the command line.
2. Set the second point on the circle line.
3. Set the third point.

5.4.7 Concentric (with Distance) (CC)

Toolbar / Icon:



Menu: `Draw > Circle > Concentric (with Distance)`
Shortcut: `C, C`
Commands: `circleconcentric | cc`

Description

With this tool you can create one or multiple concentric circles with a given distance to an existing circle.

Usage

1. Enter the distance of the concentric circle from the original base circle in the options toolbar.
2. Enter the number of concentric circles to create in the options toolbar.
3. Click the base circle. The concentric circle(s) are created on that side on which the mouse cursor is located while clicking the base circle.

5.4.8 Concentric (through Point) (CG)

Toolbar / Icon:



Menu: `Draw > Circle > Concentric (through Point)`
Shortcut: `C, G`
Commands: `circleconcentricthrough | cg`

Description

With this tool you can create concentric circles that go through a specified point.

Usage

1. Enter the number of concentric circles to create in the options toolbar.
2. Click the base circle.
3. Click the location through which the first concentric circle should go.

5.4.9 Tangent and 2 Points (CT1)

Toolbar / Icon:

Menu: `Draw > Circle > Tangent and 2 Points`
Shortcut: `C, T, 1`
Commands: `circletangent2p | ct1`

Description

Draws a circle that is tangential to an entity and goes through two points.

Usage

1. Specify the tangential entity.
2. Specify the first point on the circle line.
3. Specify the second point on the circle line.

5.4.10 Tangent, Point, Radius (CTP)

Toolbar / Icon:

Menu: `Draw > Circle > Tangent, Point, Radius`
Shortcut: `C, T, P`
Commands: `circletangentpointradius | ctp`

Description

Draws an arc with a given radius, tangential to an entity and through a point.

Usage

1. Enter the radius in the options tool bar.
2. Specify the tangential entity.
3. Specify the point on the arc.
4. Choose the desired solution by clicking close to it with the left mouse button.

5.4.11 2 Tangents and Point (CT2)

Toolbar / Icon:

Menu: `Draw > Circle > 2 Tangents and Point`
Shortcut: `C, T, 2`
Commands: `circletangent2 | ct2`

Description

Draws a circle that is tangential to two entities and goes through a point.

Usage

1. Specify the first tangential entity.
2. Specify the second tangential entity.
3. Specify the point on the circle line.

5.4.12 2 Tangents and Radius (CTR)

Toolbar / Icon:



Menu: `Draw > Circle > 2 Tangents and Radius`
Shortcut: `C, T, R`
Commands: `circletangent2radius | ctr`

Description

Draws a circle with a given radius that is tangential to two entities.

Usage

1. Type the radius into the options tool bar.
2. Specify the first tangential entity.
3. Specify the second tangential entity.

5.4.13 3 Tangents (CT3)

Toolbar / Icon:



Menu: `Draw > Circle > 3 Tangents`
Shortcut: `C, T, 3`
Commands: `circletangent3 | ct3`

Description

Draws a circle that is tangential to three entities.

Usage

1. Specify the first tangential entity.
2. Specify the second tangential entity.
3. Specify the third tangential entity.

5.5 Ellipse

Toolbar / Icon:



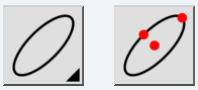
Shortcut: `W, E`
Commands: `ellipsemenu`

Description

Ellipses are defined by a center point, a major and a minor axis. The ellipse tools allow you to construct full ellipses and elliptic arcs, for example from a center point and two axis points or as an offset of an existing ellipse.

5.5.1 Ellipse (Center, Point, Ratio) (EP)

Toolbar / Icon:



Menu: `Draw > Ellipse > Ellipse (Center, Point, Ratio)`
Shortcut: `E, P`
Commands: `ellipse | ep`

Description

Draws ellipses with a given center, major axis and minor axis.

Usage

1. Set the center of the ellipse using the mouse or enter a coordinate in the command line.
2. Define the major axis by clicking the endpoint of the axis, which is a point on the ellipse. You can also enter a coordinate into the command line or enter an angle and major radius in the format @50<30 where 50 is the major radius and 30 is the ellipse angle.
3. Define the endpoint of the minor axis which is also a point on the ellipse.

5.5.2 Ellipse Arc (EA)

Toolbar / Icon:



Menu: `Draw > Ellipse > Ellipse Arc`
Shortcut: `E, A`
Commands: `ellipsearc | ea`

Description

Draws ellipse arcs with a given center, major and minor axis and start and end angles.

Usage

1. Set the center of the ellipse using the mouse or enter a coordinate in the command line.

2. Define the major axis by clicking the endpoint of the axis, which is a point on the ellipse. You can also enter a coordinate into the command line or enter an angle and major radius in the format @50<30 where 50 is the major radius and 30 is the ellipse angle.
3. Define the endpoint of the minor axis which is also a point on the ellipse or enter the length of the minor axis.
4. Set the start angle with the mouse or by entering a coordinate or the angle in the command line.
5. Set the end angle the same way as the start angle.

5.5.3 Ellipse with Radii (EI)

Toolbar / Icon:



Menu: `Draw > Ellipse > Ellipse with Radii`
Shortcut: `E, I`
Commands: `ellipseradii | ei`

Description

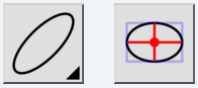
Draws ellipses with given major and minor radius.

Usage

1. Enter the radius in X (major radius), the radius in Y (minor radius) and the ellipse angle in the options toolbar.
2. Set the center of the ellipse using the mouse or enter a coordinate in the command line.

5.5.4 Ellipse with Diameters (ED)

Toolbar / Icon:



Menu: `Draw > Ellipse > Ellipse with Diameters`
Shortcut: `E, D`
Commands: `ellipsediameters | ed`

Description

Draws ellipses with given major and minor diameter (width / height).

Usage

1. Enter the diameter in X (major diameter), the diameter in Y (minor diameter) and the ellipse angle in the options toolbar.
2. Set the center of the ellipse using the mouse or enter a coordinate in the command line.

5.5.5 Parallel Curve (with Distance) (EC)

Toolbar / Icon:



Menu: `Draw > Ellipse > Parallel Curve (with Distance)`
Shortcut: `E, C`
Commands: `ellipseoffset | ec`

Description

With this tool you can create one or multiple parallel curves with a given distance to an existing ellipse.

Usage

1. Enter the distance of the parallel curves from the original base ellipse in the options toolbar.
2. Enter the number of parallel curves to create in the options toolbar.
3. Click the base ellipse. The concentric curve(s) are created on that side on which the mouse cursor is located while clicking the base ellipse.

5.5.6 Parallel Curve (through Point) (EG)

Toolbar / Icon:



Menu: `Draw > Ellipse > Parallel Curve (through Point)`
Shortcut: `E, G`
Commands: `ellipseoffsetthrough | eg`

Description

With this tool you can create curves that are parallel to an ellipse and go through a specified point.

Usage

1. Enter the number of parallel curves to create in the options toolbar.
2. Click the base ellipse.
3. Click the location through which the first parallel curve should go.

5.5.7 Inscribed in Quadrilateral (EQ)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Ellipse > Inscribed in Quadrilateral`
Shortcut: `E, Q`
Commands: `ellipseinscribedquad | eq`

Description

Draws an ellipse that is inscribed in a quadrilateral.

Usage

1. Specify the first line of the quadrilateral.
2. Specify the second line of the quadrilateral.
3. Specify the third line of the quadrilateral.
4. Specify the fourth line of the quadrilateral.

5.6 Spline

Toolbar / Icon:



Shortcut: `W, N`
Commands: `splinemenu`

Description

Splines are smooth curves defined by a set of control points or fit points.

5.6.1 Spline (Control Points) (SP)

Toolbar / Icon:




Menu: `Draw > Spline > Spline (Control Points)`
Shortcut: `S, P`
Commands: `spline | sp`

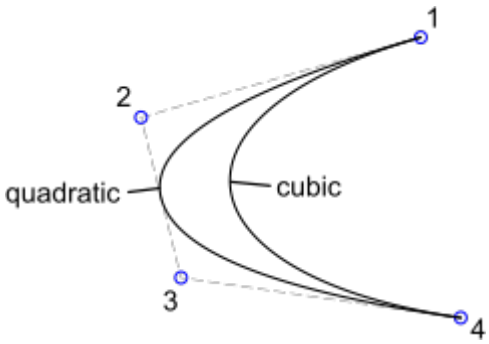
Description

Draws spline curves from control points. Non-uniform rational B-Splines (NURBS) with homogeneous weighting factors are the only splines that are supported.

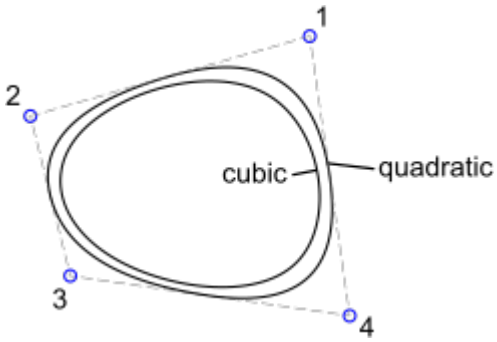
Usage

1. Choose the degree of the spline in the options tool bar. Supported degrees are 2 (quadratic b-spline) and 3 (cubic b-spline). The higher the degree, the 'smoother' the curve becomes.
   Note that a quadratic b-spline requires by definition at least 3 control points while a cubic b-spline has at least 4 control points.
2. Check the 'Closed' check box in the options tool bar if you want to create a closed spline. Closed splines are continuous closed loops.
3. Specify the control points. After defining the third (for quadratic b-splines) or fourth (for cubic b-splines) control point, a preview will be shown. You can remove the last control point again by clicking the 'Undo' button in the options tool bar.
4. Hit escape or click the right mouse button after setting the last point to confirm the spline shape.

5. You can now create another spline or right-click again to terminate the tool. Example for two open splines (quadratic and cubic):

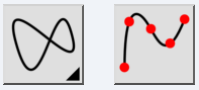


Example for two closed, periodic splines (quadratic and cubic):



5.6.2 Spline (Fit Points) (SL)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Spline > Spline (Fit Points)`
Shortcut: `S, L`
Commands: `splinefit | sl`

Description

Draws cubic spline curves from fit points (points on the spline curve).

Usage

1. Check the "Closed" check box in the options tool bar if you want to create a closed spline. Closed splines are continuous closed loops.
2. Specify the fit points. You can remove the last fit point by clicking the "Undo" button in the options tool bar.
3. Hit escape or click the right mouse button after setting the last point to confirm the spline shape.
4. You can not create another spline or right-click again to terminate the tool.

5.6.3 Insert Fit Point (NI)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Spline > Insert Fit Point`
Shortcut: `N, I`
Commands: `splineinsertfit | ni`

Description

With this tool you can insert fit points into existing splines.

Usage

- Click the position on the spline at which a new fit point should be inserted. The fit point is inserted at the point on the spline which is closest to the point that is clicked. If the spline is selected, the new fit points are visualized.

5.6.4 Append Fit Point(s) (NA)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Spline > Append Fit Point(s)`
Shortcut: `N, A`
Commands: `splineappend | na`

Description

With this tool you can append fit points to existing splines.

Usage

1. Click a position near the start of a spline to append fit points at the beginning. Click a position near the end of a spline to append fit points at the end.
2. Click the positions of the new fit points.

5.6.5 Remove Fit Point (NR)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Spline > Remove Fit Point`
Shortcut: `N, R`
Commands: `splineremovefit | nr`

Description

With this tool you can remove fit points from splines.

Usage

- Click the fit point of a spline to remove it.

5.6.6 Simplify (NP)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Spline > Simplify`
Shortcut: `N, P`
Commands: `splinesimplify | np`

Description

Simplifies selected splines by reducing the number of control points while keeping the shape close to the original.

5.7 Polyline

Toolbar / Icon:



Shortcut: `W, O`
Commands: `polylinemenu`

Description

Polylines are connected sequences of line and arc segments that form a single entity. Polylines can be open or closed and they can have a custom line weight.

5.7.1 Draw Polyline (PL)

Toolbar / Icon:

Menu: `Draw > Polyline > Draw Polyline`
Shortcut: `P, L`
Commands: `polyline | pl`

Description

This tool lets you draw polylines. A polyline is a continuous line formed from one or more connected line or arc segments.

Please note that it is almost always more convenient to use the regular line and arc tools to draw a new polyline. The loosely connected line and arc segments can then be converted into a polyline entity using the "Create Polyline from Segments" tool.

Usage

1. Specify the start point of the first polyline segment. You can use the mouse or enter a coordinate in the console.
2. Specify the endpoint of the first polyline segment.
3. Specify the endpoints of additional polyline segments.

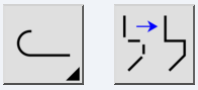
If the polyline contains arc segments, you can tick the "Arc" check box and enter the arc radius in the options tool bar when setting the endpoint of an arc segment. The arc will connect smoothly (tangential) to the last segment.

QCAD Professional

- Tick "Tangential arc" in the options toolbar to create arc segments that are tangentially connected to the last segment.
- Tick the "Radius" checkbox to limit the arc segment radius to the given value. Otherwise the arc segment is constructed from the last segment to the mouse cursor.
- Tick the "Arc angle" checkbox to limit the total arc angle to the given value.

5.7.2 Polyline from Segments (OG)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Polyline from Segments`
Shortcut: `O, G`
Commands: `polylinefromsegments | og`

Description

This tool lets you create polylines from existing segments (lines and arcs that are connected). Connected splines are automatically converted to polylines with arc segments.

Usage

Click one of the segments. A polyline is created from the selected entity and all line or arc entities that are connected to its end- or start point.
The polyline is created on the same layer with the same attributes as the entity that is clicked. If that layer is locked, the polyline is created on the current layer instead.

5.7.3 Polyline from Selection (OC)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Polyline from Selection`
Shortcut: `O, C`
Commands: `polylinefromselection | oc`

Description

This tool lets you create polylines from selected segments (connected lines, arcs and polylines).

Usage

1. Select the entities you want to join into polylines.
2. Launch this tool.
3. All connected entities are joined into a polyline. If not all selected entities are connected, multiple polylines are created.
4. The number of polylines created is printed in the command line history.

5.7.4 Partial Polyline Along Selection (OL)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Partial Polyline Along Selection`
Shortcut: `O, L`
Commands: `polylinealongselection | ol`

Description

This tool lets you create polylines along a partial length along a selected path (connected lines, arcs and polylines).

Usage

1. Select the entities along which you want to create a polyline.
2. Launch this tool.
3. Click the start point of the polyline. To start point can be at any position along the selected entities.
4. Click the end point of the polyline.
5. The polyline is created between the chosen points along the selected entities. Disconnected entities which are not part of the new polyline are deselected.

5.7.5 Insert Node (OA)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Insert Node`
Shortcuts: `O, A | O, I`
Commands: `polylineinsert | polylineadd | oa | oi`

Description

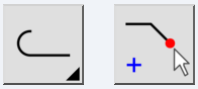
With this tool you can add nodes to existing polylines. Adding a new node to a polyline means to split up one segment into two segments. The two new segments are both straight, even if one of original segments was an arc segment.

Usage

1. Specify the segment of the polyline in which you want to add the new node.
2. Specify the position of the new node with your mouse or by entering a coordinate.

5.7.6 Append Node(s) (OP)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Append Node(s)`
Shortcut: `O, P`
Commands: `polylineappend | op`

Description

With this tool you can append nodes to the beginning or end of existing polylines. Appending a node to a polyline means to extend it by one segment at the beginning or end.

Usage

1. Click the polyline somewhere near the end to which you want to append a node.
2. Specify the position of the new node(s) with your mouse or by entering a coordinate.

5.7.7 Delete Node(s) (OD)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Delete Node(s)`
Shortcut: `O, D`
Commands: `polylinedel | od`

Description

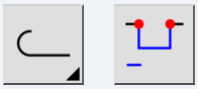
Use this tool to remove nodes from existing polylines. This way two segments of a polyline can be joined into one.

Usage

1. Specify the polyline from which you want to delete nodes.
2. Specify the position of the node that you want to remove.

5.7.8 Remove Nodes between Two Nodes (OB)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Remove Nodes between Two Nodes`
Shortcut: `O, B`
Commands: `polylinedelbetween | ob`

Description

Use this function to delete all nodes between two nodes of an existing polyline.

Usage

1. Specify the polyline from which you want to delete segments.
2. Specify the position of the first limiting node. That is the node from where you want to start deleting segments. The node itself will not be deleted. In most cases it is appropriate to use the endpoint snap function to specify the node.
3. Specify the position of the node where you want to stop deleting. All segments that are between the first node and the second node are removed from the polyline. After the segments are deleted, the two limiting nodes are connected by a straight line segment so the polyline is still one continuous sequence of line segments.

5.7.9 Trim Segments (OT)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Trim Segments`
Shortcut: `O, T`
Commands: `polylinetrim | ot`

Description

Use this function to trim two polyline segments. This means to extend or shorten the two segments, in a way that they meet at one common point.

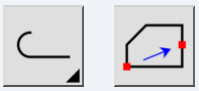
Usage

1. Specify the first polyline segment to trim by clicking it with the mouse.

2. Specify the second segment to be trimmed from the same polyline. All segments that are between the first segment and the second segment are removed to keep the polyline a continuous line string.

5.7.10 Relocate Start Point (OR)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Relocate Start Point`
Shortcut: `O, R`
Commands: `polylinestart | or`

Description

Use this tool to choose a new start point for a closed polyline.

Usage

1. Click the closed polyline of which the start point should be relocated.
2. Click the new position of the start point anywhere along the polyline.

5.7.11 Change Segment Type (OX)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Change Segment Type`
Shortcut: `O, X`
Commands: `polylinechange | ox`

Description

With this tool you can change the type of a polyline segment from an arc to a line and vice versa.

Usage

- Click an existing polyline segment. An arc segment is immediately converted into a line segment. For a line segment, the arc can be defined by clicking a point on the arc.

5.7.12 Arc to Line Segments

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Arc to Line Segments`
Commands: `polylinearctoline`

Description

Use this tool to convert all arc segments in polylines into straight line segments.

Usage

1. Select the polyline(s) from which you want to convert all segments into line segments.
2. Start this tool.

5.7.13 Offset (OQ)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Polyline > Offset`
Shortcut: `O, Q`
Commands: `polylineequidistant | polylineoffset | oq`

Description

This tool lets you create equidistant polylines to existing polylines.

Usage

1. Enter the distance of the equidistant and the number of equidistant polylines you want to create at once in the options tool bar.
2. Click the original polyline. The equidistant will be created on the side on which you click next to the polyline.

5.7.14 Morph (OM)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Polyline > Morph`
Shortcut: `O, M`
Commands: `polylinemorph | om`

Description

With this tool, you can morph between two polylines. A number of new intermediate polylines are created. Each intermediate polyline is a mix of the two chosen polylines. With each step of the morphing operation, the position and shape of the first polyline is transformed to be more like the second polyline.

Usage

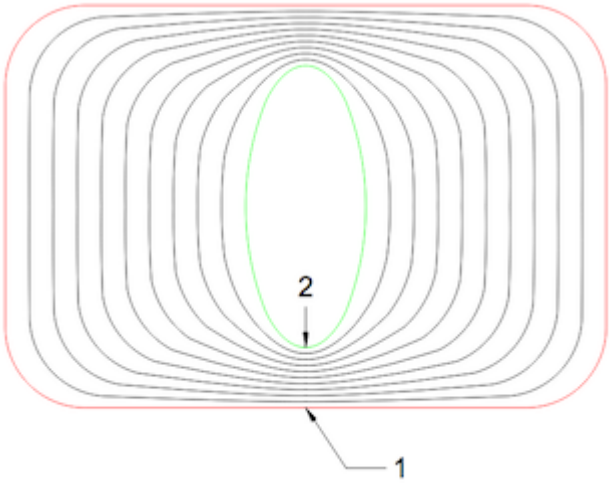
1. Prepare the two polylines.
   Usually the two polylines should have the same direction or orientation. For closed polylines, you would typically want the start point to be aligned, so that no rotation occurs while

morphing. More complex polylines can be split at sharp corners to maintain the position of those corners throughout the morphing process.

2. Launch this tool.
3. Enter the number of desired steps in the options toolbar.
4. Choose the desired easing function in the options toolbar. The normal easing function is "linear" which interpolates linearly between the two polylines. Easing functions can be used to create various uneven types of interpolations.
   For example to start morphing at the first polyline, first advancing slowly toward the second polyline and then accelerate toward the second polyline in a quadratic function, choose "in quadratic".
5. Click the first polyline. You can also select a line, an arc, a circle, an ellipse or a spline.
6. Click the second polyline.
7. The entered number of intermediate polylines is created.

Example:

In this example, the rounded rectangle is morphed to the ellipse in nine steps. Both shapes are first converted into polylines (the ellipse is exploded into a polyline with arc segments). The startpoint of the rounded rectangle is set to be at the bottom center (1). The start point of the ellipse is set at its bottom center (2). Both polylines are made counter-clockwise.



5.7.15 Boolean Operations (OW)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Draw > Polyline > Boolean Operations`
Shortcut: `O, W`
Commands: `polylinebooleancavalier | ow`

Description

Performs boolean operations (union, intersection and difference) on closed polylines, adapted for cavalier (isometric) projection.

5.7.16 Simplify (OS)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Simplify`
Shortcut: `O, S`
Commands: `polylinesimplify | os`

Description

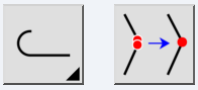
Simplifies all selected polylines using the Ramer–Douglas–Peucker algorithm.

Usage

1. Select the polyline(s) you want to simplify.
2. Start this tool.
3. Enter the desired tolerance in the Options toolbar. The tolerance is the maximum distance between the original polyline and the simplified polyline.

5.7.17 Normalize Polylines (ON)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Normalize Polylines`
Shortcut: `O, N`
Commands: `polylinenormalize | on`

Description

Use this tool to automatically remove duplicate nodes from existing polylines.

Usage

1. Select the polyline(s) from which you want to delete duplicate nodes.
2. Start this tool.

5.7.18 Insert Nodes at Self-Intersections (OV)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Polyline > Insert Nodes at Self-Intersections`
Shortcut: `O, V`
Commands: `polylineinsertatselfx | ov`

Description

Use this tool to insert nodes at points where the polyline intersects itself.

5.7.19 Logically close Polylines (OE)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Draw > Polyline > Logically close Polylines`
Shortcut: `O, E`
Commands: `polylinetoclosed | oe`

Description

Polylines can be geometrically closed or logically closed. A geometrically closed polyline is essentially an open polyline for which the last node just happens to be at the same position as the first node. In a logically closed polyline with the same shape, the last node is omitted and the information that the polyline is closed is stored in the polyline instead. Logically closed polylines require one node less than geometrically closed ones. Logically closed polylines are often better suitable for further processing (CAM) than geometrically closed ones.

This tool converts all geometrically closed polylines into logically closed polylines.

5.7.20 Logically open Polylines (OJ)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Draw > Polyline > Logically open Polylines`
Shortcut: `O, J`
Commands: `polylinetoopen | oj`

Description

This tool converts all logically closed polylines into logically open (but geometrically closed) polylines.

See also Draw > Polyline > Logically close Polylines

5.8 Shape

Toolbar / Icon:



Shortcut: `W, H`
Commands: `shapemenu`

Description

Shapes are closed geometric figures such as rectangles or regular polygons.

Common Shape Options

Polyline

Shape tools can either create loose lines and arcs or a closed polyline. Tick "Create Polyline" to create a single polyline entity per constructed shape.

Fill

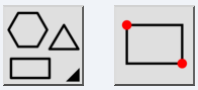
Tick the "Fill" check box to create a solid fill in addition to the boundary of the shape.

Radius

Tick the "Radius" check box to round all corners of the shape with the given radius.

5.8.1 Rectangle (RE)

Toolbar / Icon:



Menu: `Draw > Shape > Rectangle`
Shortcut: `R, E`
Commands: `rectangle | linerectangle | rect | re`

Description

Use this tool to create rectangular shapes from two diagonally opposite corners.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Specify the first corner of the rectangle.
3. Move the mouse to the second corner and click to specify the second corner of the rectangle. Alternatively you can enter the coordinate of the second corner in the command prompt. E.g. to create a rectangle with width 50 and height 25, enter the relative coordinate of the second corner as follows:
   @50,25

5.8.2 Rectangle with Size (RS)

Toolbar / Icon:



Menu: `Draw > Shape > Rectangle with Size`
Shortcut: `R, S`
Commands: `rectanglesize | linerectanglesize | rs`

Description

Use this tool to create rectangular shapes with a given size.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Enter the size of the rectangle in the options tool bar.
3. Choose the reference point you want to use to position the rectangle.
4. Specify the position of the rectangle with the mouse or by entering a coordinate in the command line.

5.8.3 Rectangle (3 Points) (R3)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Shape > Rectangle (3 Points)`
Shortcut: `R, 3`
Commands: `rectangle3p | linerectangle3p | rect3p | r3`

Description

Use this tool to create rectangular shapes from two corners of the first edge and a point on the opposite side.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Specify the first corner of the rectangle.
3. Move the mouse to the second corner and click to specify the second corner of the first edge of the rectangle. This defines the length and rotation angle of the rectangle.
4. Click a third point that is on the opposite edge. This defines the width of the rectangle.

5.8.4 Polygon (Center, Corner) (PG1)

Toolbar / Icon:

Menu: `Draw > Shape > Polygon (Center, Corner)`
Shortcuts: `P, G, 1 | H, C`
Commands: `linepolygon | polygon | pg1`

Description

Creates polygons with the center and one corner given.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Enter the degree of the polygon (number of edges / corners) in the options tool bar. Valid numbers range from 3 to 99.
3. Set the center of the polygon using the mouse or enter a coordinate in the command line.
4. Set the first corner of the polygon.

5.8.5 Polygon (2 Points of Side) (PG2)

Toolbar / Icon:




Menu: `Draw > Shape > Polygon (2 Points of Side)`
Shortcuts: `P, G, 2 | H, P`
Commands: `linepolygon2 | polygon2 | pg2`

Description

Creates polygons with two corners given.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Enter the degree of the polygon (number of edges / corners) in the options tool bar. Valid numbers range from 3 to 99.
3. Set the first corner of the polygon using the mouse or enter a coordinate in the command line.
4. Set the second corner of the polygon.

5.8.6 Polygon (Center, Side) (PG3)

Toolbar / Icon:




Menu: `Draw > Shape > Polygon (Center, Side)`
Shortcut: `P, G, 3`
Commands: `polygoncs | pg3`

Description

Creates polygons with the center and middle point of one side.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Enter the degree of the polygon (number of edges / corners) in the options tool bar. Valid numbers range from 3 to 99.
3. Set the center of the polygon using the mouse or enter a coordinate in the command line.
4. Set the middle point of one side of the polygon.

5.8.7 Polygon (Side, Side) (PG4)

Toolbar / Icon:




Menu: `Draw > Shape > Polygon (Side, Side)`
Shortcut: `P, G, 4`
Commands: `polygonss | pg4`

Description

Creates polygons from two opposite corners or sides.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Enter the degree of the polygon (number of edges / corners) in the options tool bar. Valid numbers range from 3 to 99.
3. Choose the option Corner to Corner to define the polygon with two opposite corners.
4. Set the first corner or middle of a side of the polygon using the mouse or enter a coordinate in the command line.
5. Set the second, opposite corner or middle of a side of the polygon.

5.8.8 Star (PT)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Shape > Star`
Shortcut: `P, T`
Commands: `star | pt`

Description

Creates stars from a given outer and inner corner.

Usage

1. Choose the desired common shape options in the options toolbar.
2. Enter the number of outermost corners of the star shape in the options tool bar. Valid numbers range from 3 to 99.
3. Tick the option "symmetrical" to create a regular, symmetrical star. Without this option, the inner corner can be placed freely.
4. Set the first outer corner of the star using the mouse or enter a coordinate in the command line.
5. Set the inner corner of the star.

5.9 Hatch

Toolbar / Icon:



Shortcut: `W, X`
Commands: `hatchmenu`

Description

Hatches and solid fills are used to fill a closed area with a pattern or a solid color.

5.9.1 Auto Hatch (HT)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Draw > Hatch > Auto Hatch`
Shortcut: `H, T`
Commands: `hatchauto | ht`

Description

Creates a hatch or solid fill by automatically detecting the closed contour around a picked point.

Usage

1. Make sure that the area to be filled is enclosed on all sides with entities (e.g. lines, arcs, circles, polylines, splines, etc.). The entities do not have to be trimmed to corners for this tool.
2. Start this tool.
3. A dialog for the hatch options is displayed. Choose the fill type with the tabs at the top of the dialog:
   - Hatch: fill the area with a hatch pattern. Choose a pattern, a scale factor and a rotation angle for the pattern.
   - Solid: fill the area with a solid color. The color of the hatch entity is used (by layer or as chosen in the property editor).
   - Gradient: fill the area with a smooth transition between two colors.
4. To create a gradient fill, switch to the tab "Gradient" and choose the gradient parameters:
   - Type: the shape of the color transition. "Linear" changes the color along a straight line from one side of the area to the other. "Cylinder" changes the color from both sides towards the middle. "Spherical" and "Hemispherical" change the color in circles around a center point. "Curved" is like "Linear" but with a non-linear, curved transition. The inverted variants of each type swap the roles of the two colors.
   - Color 1 / Color 2: the two colors of the gradient. The color changes gradually from color 1 to color 2. To fade a single color towards black or white instead, check "One color". The second color is then derived from color 1 with the slider "Shade / Tint": positions left of the center shade towards black, positions right of the center tint towards white, the center position keeps the color unchanged (no visible gradient).
   - Angle: rotates the gradient within the filled area. With an angle of 0, the color changes from left (color 1) to right (color 2). For example, an angle of 90 degrees changes the color from bottom to top instead.
   - Centered: if checked, the gradient is centered in the filled area, i.e. the color transition is placed symmetrically within the boundary of the hatch.
   - Shift: if "Centered" is unchecked, the shift moves the gradient out of the center, up and to the left relative to the chosen gradient angle. A shift of 0.0 means the gradient is centered, a shift of 1.0 moves it as far as possible. This can be used to create the illusion of a light source to the upper left of the filled object. The preview at the bottom of the dialog shows the effect of the chosen parameters.
5. Click OK.

6. Choose how closed contours inside the area to be filled (islands) are handled in the options toolbar:
   - Ignore: no island detection. The whole contour is filled, including any islands inside it.
   - Normal: islands and islands within islands are detected. Islands are cut out from the hatch, islands within islands are filled again.
   - Outer: only the first order of islands is detected and cut out from the hatch. Contours inside those islands are ignored.
7. Click a point inside the closed area you want to fill. The smallest closed contour around the clicked point is detected automatically and used as the hatch boundary.
8. Repeat the previous step to fill more areas with the same pattern or click the right mouse button to terminate the tool.

5.9.2 Hatch from Selection... (HA)

Toolbar / Icon:



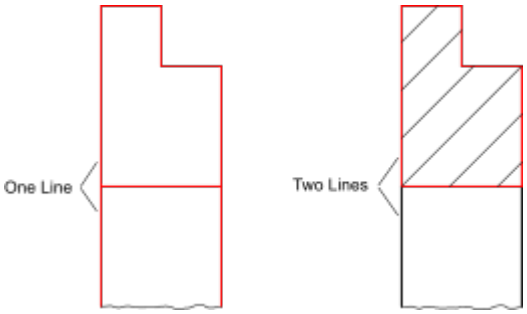
Menu: `Draw > Hatch > Hatch from Selection...`
Shortcut: `H, A`
Commands: `hatch | ha`

Description

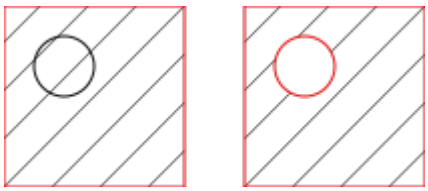
This tool fills an area surrounded by existing entities with a hatch pattern or a solid color.

Usage

1. Prepare the entities that surround the hatching area so they form a closed contour. The contour must be closed in a way that one entity is connected to the next one as shown at the right in this example:



2. Select the contour(s) you want to fill. Note that islands inside contours will get hatched if they are not selected:

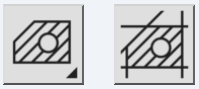


3. Launch the hatch tool.

4. A dialog for the hatch options is displayed. Choose a hatch pattern, scale factor and a rotation angle for the hatch pattern. If you want to fill the object with a solid color instead of a pattern, check the check box "Solid Fill".
5. Click 'OK' to proceed with the hatching. Depending on the complexity of the contour and the scale factor of the chosen pattern, it might take a while to create the hatch.

5.9.3 Hatch from Segments... (HS)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Hatch > Hatch from Segments…`
Shortcut: `H, S`
Commands: `hatchsegments | hs`

Description

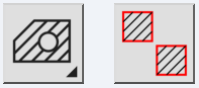
This tool fills an area with a hatch pattern or a solid color. The area is defined by clicking existing entities or untrimmed entity segments which are part of the hatch boundary.

Usage

1. Make sure that the area to be filled is enclosed on all sides with entities (e.g. lines, arcs, splines, etc.). The entities do not have to be trimmed to corners for this tool. You can use partial entities or segments of entities as part of the boundary.
2. Start this tool.
3. Click OK.
4. A dialog for the hatch options is displayed. Choose a hatch pattern, scale factor and a rotation angle for the hatch pattern. If you want to fill the object with a solid color instead of a pattern, check the check box "Solid Fill".
5. Click all entity segments that are part of the hatch or fill boundary. The selected segments are highlighted. You can also remove a segment from the boundary by clicking it again. The boundary segments do not have to be clicked in any particular order and you can also define islands inside an area.

5.9.4 Merge Hatches (HM)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Hatch > Merge Hatches`
Shortcut: `H, M`
Commands: `hatchmerge | hm`

Description

This tool merges two or more selected hatches into a single hatch entity. The boundaries of all selected hatches become boundary loops of the merged hatch. This is useful to turn a number of

separate hatches, for example hatches created one by one with "Auto Hatch", into one entity that can be selected, moved, edited or deleted as a whole.

Usage

1. Select at least two hatches. Other entities in the selection are ignored.
2. Start this tool.
3. The selected hatches are merged into one hatch entity. The merged hatch keeps the pattern, scale, angle, layer, color and other attributes of the oldest hatch in the selection (the hatch that was created first). The other hatches are removed and their boundaries are added to the merged hatch as additional loops.
4. If the selected hatches use different patterns or mix solid fills with pattern hatches, a message is shown and the pattern of the oldest hatch is used for the whole merged hatch.

Note that the boundaries are combined as they are. Overlapping or nested boundaries are not united or subtracted from each other, but behave like islands within a hatch: by default, areas enclosed by an odd number of loops are filled, areas enclosed by an even number of loops are left empty.

5.10 Viewport

Toolbar / Icon:



Shortcut: `W, T`
Commands: `viewportmenu`

Description

Viewports are special entities intended to be used in layout blocks. A viewport shows a part of the model space in the layout.

5.10.1 Add Viewport (VT)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Viewport > Add Viewport`
Shortcut: `V, T`
Commands: `viewport | vt`

Description

Creates a new viewport in a layout block. A viewport is a rectangular area that displays a part of the drawing at a given scale and angle. Viewports can only exist in layout blocks, special blocks dedicated to printing the drawing or parts of it, often along with a border and header.

Usage

1. After starting this tool, the model space (main drawing) is shown to define the rectangular area of the drawing the viewport should display.
2. Adjust the rotation angle of the viewport in the options toolbar to create a viewport that shows a part of the drawing rotated by a given angle. Note that rotated viewports are not supported in other CAD systems.
3. Click the first corner of the viewport area.
4. Click the second corner of the viewport area.
5. QCAD now switches automatically to a layout block where the viewport can be created. If there are multiple layout blocks and this tool was not started from a layout block, a dialog is shown to let you choose in which layout block you want to place the viewport.
6. Adjust the viewport scale in the options toolbar if necessary. If the drawing is larger or much smaller than the available paper size, you will likely want to scale it up or down to fit on the paper.
7. Click the center position of the viewport.

The viewport is now added to the layout. To change properties such as scale or rotation angle of the viewport, you can click it to select it and adjust properties using the property editor.

The rectangular border of the viewport is shown if the layer the viewport is on is shown. If the layer the viewport is on is switched off, the border is hidden but the contents of the viewport is still shown. Viewports are typically placed on a dedicated layer (e.g. "viewports") which can be switched off for printing.

5.10.2 Manage Viewport Layers... (VY)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Draw > Viewport > Manage Viewport Layers…`
Shortcut: `V, Y`
Commands: `viewportlayers | vy`

Description

This tool can be used to hide layers in selected viewports.

Usage

1. This tool can be started from the context menu of a viewport. Alternatively, select the viewport(s) you want to manage and start this tool from the menu.
2. A dialog is shown with a layer list. Click the eye icon next to a layer to toggle the layer on or off for the viewport. This does not affect layers in the main drawing or in other viewports.
3. Note that a viewport can only turn additional layers off. It cannot turn a layer on that is off in the main drawing.

5.11 Ring (RI)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Ring`
Shortcut: `R, I`
Commands: `ring | donut | ri`

Description

This tool lets you create rings (donuts) with a given center and an inner and outer diameter.

Usage

1. Enter the inner and outer diameters in the option tool bar.
2. Set the center of the ring using the mouse or enter a coordinate in the command line.

5.12 Wipeout (WP)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Draw > Wipeout`
Shortcut: `W, P`
Commands: `wipeout | wp`

Description

This tool creates a wipeout entity. A wipeout is a blank area in a drawing that conceals a part of the drawing with the background color. This is typically used to improve the readability of text labels or to hide a part of a drawing that is not relevant.

Usage

1. Check the frame option in the options toolbar to create a wipeout with a frame (a line around the boundary).
2. Choose the desired tool mode in the options toolbar. The wipeout tool offers three different modi:
   - Rectangle: creates a rectangular wipeout from two diagonally opposite points. Click the first and then the second corner of the rectangle.

     
   - Polygon: creates a polygonal wipeout. Click the corners of the polygon and click the right mouse button after the last corner has been set. The undo / redo buttons in the options toolbar may be used to correct mistakes.



- From polyline: creates a wipeout from an existing polyline. Click a polyline in the drawing to create a wipeout from it. If the option "Delete Polyline" is checked, the original polyline will be deleted. For convenience, the tool also accepts circles, ellipses or splines.



5.13 Text... (TE)

Toolbar / Icon:



Menu: `Draw > Text…`
Shortcut: `T, E`
Commands: `text | te`

Description

Use this tool to create single line or multi line text entities.

Usage

1. After launching the tool, the text dialog is presented in which you can choose the font and enter the text you want to create.
2. Choose the font name, text height and initial text attributes in the upper left area of the dialog. CAD fonts that are made up of lines and arcs are listed at the top of the font list, TrueType fonts are listed below.
   In the "Height" field, enter the height of the text in the unit of your drawing. To choose normal line spacing, enter a line spacing factor or "1". You can increase or decrease the line spacing by entering an alternative value. The value entered is a factor relative to the default (1.0).
3. Enter the rotation angle for the text. Normal horizontal texts have an angle of 0 degrees and texts that are readable from the right 90 degrees.
4. Choose the alignment of the text.
5. Check the "Simple Text" check box to create a single line text without any advanced formatting.
6. Enter the text you want to create in the large text box.
   You can use the various tools above the box to further format the text that you have entered or to paste texts from the clipboard. Use the button at the right to quickly insert symbols or to choose a Unicode character ("Others...").
   Note that the font you have chosen might not contain the characters you insert. In that case the characters will not be displayed in the drawing.
7. Click "OK" to exit the text dialog.
8. Use the mouse to specify the location of the text or enter a coordinate in the command line.
9. Often, users need to create a number of similar texts. For that reason the text tool does not terminate after creating the first text entity. You can change the text and the text angle in the options tool bar and create other texts with similar settings.
10. Right click or press Escape to stop the text tool.

5.14 Insert Bitmap... (IM)

Toolbar / Icon:



Menu: `Draw > Insert Bitmap...`
Shortcut: `I, M`
Commands: `insertbitmap | bitmap | im`

Description

Inserts a raster image (bitmap) into the drawing.
Note that the drawing file which is created when saving the drawing, contains only references to inserted images. It is recommended to keep the image file and the drawing file in the same folder, so QCAD can find the image again when the drawing file is loaded later.
Transparent backgrounds are supported for PNG images.
Note that large bitmaps can cause the drawing display to become very slow.
Usually you will want images to be in the background of other entities. Please refer to the 'Modify' - 'Send to back' tool to learn how to do this.

Usage

1. Choose the image file which you want to insert.
2. Enter the desired height or width for the image in drawing units in the options tool bar.
3. Type a rotation angle in the options tool bar.
4. Specify the insertion point of the image. That is the left bottom corner of the image.

5.15 Trace Bitmap...

Toolbar / Icon:



Add-on: **QCAD Professional**
Menu: `Draw > Trace Bitmap...`
Commands: `tracebitmap | trace`

Description

This tool traces an inserted image with polylines. This can be used as a means to vectorize in image, for example to quickly trace the outlines of an inserted image from a silhouette.

Usage

1. Insert the image you want to trace into your drawing if you have not done so already.
2. Select the image and start this tool.
3. Choose the desired options in the dialog:
4. **Monochrome:** Enabling this option converts the image to monochrome on the fly before tracing. This can yield better results for silhouette type images that are not completely

monochrome already. You can also use this option if you are only interested in the outline of an image with a transparent or white background.

5. **Alpha Threshold:** The maximum alpha (transparency) value of a pixel to be considered white when converting to monochrome. Valid values range from 0 (everything not completely transparent is black) to 255 (everything is white).
6. **Brightness Threshold:** The maximum brightness of a pixel to be considered black. Valid values range from 0 (everything is black) to 255 (everything is white).
7. **Number of colors:** Images are reduced in color before tracing. This number indicates the number of colors to be used for the palette. Lower numbers yield less details or less noise and higher numbers more details or noise.
8. You can click the Update button anytime to update a small preview of the image tracing.
9. Click OK to trace the image. The polylines are added on the current layer.

6 Dimension

Toolbar / Icon:



Shortcut: `W, D`
Commands: `dimensionmenu`

Description

Dimensions are used to add measurements to a drawing.
Dimensions typically consist of a text label and some lines and arrows to indicate the precise location of the measurement. The text label may be accompanied by tolerance indications or symbols. The options tool bar that is shown while a dimension is being created supports you to define the dimension label contents and its tolerances.

6.1 Aligned (DA)

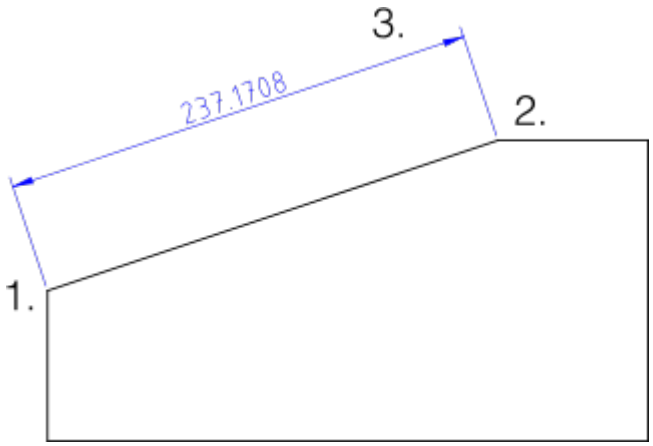
Toolbar / Icon:



Menu: `Dimension > Aligned`
Shortcut: `D, A`
Commands: `dimaligned | da`

Description

Creates aligned dimensions. Aligned dimensions usually measure the length of an existing line. The dimension line is always parallel to the line between the two chosen points 1. and 2.:

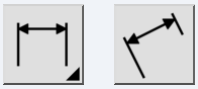


Usage

1. Set the first extension line end point with the mouse or enter a coordinate in the command line.
2. Set the second extension line end point.
3. Set the position of the dimension line.

6.2 Rotated (Linear) (DL)

Toolbar / Icon:



Menu: `Dimension > Rotated (Linear)`
Shortcut: `D, L`
Commands: `dimlinear | dimrotated | dl`

Description

Creates rotated (linear) dimensions. Linear dimensions are usually used to measure vertical or horizontal distances but can also measure distances with any other angle.

Usage

1. Enter the angle of the dimension in the options toolbar.
2. Set the first extension line endpoint with the mouse or enter a coordinate in the command line.
3. Set the second extension line endpoint.
4. Set the position of the dimension line.

6.3 Horizontal (DH)

Toolbar / Icon:



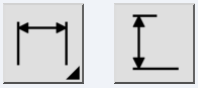
Menu: `Dimension > Horizontal`
Shortcut: `D, H`
Commands: `dimhor | dimhorizontal | dh`

Description

This tool is provided for convenience and behaves essentially like the tool for rotated dimensions. The only difference is that the angle is fixed to 0 degrees (horizontal).

6.4 Vertical (DV)

Toolbar / Icon:



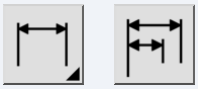
Menu: `Dimension > Vertical`
Shortcut: `D, V`
Commands: `dimver | dimvertical | dv`

Description

This tool is provided for convenience and behaves essentially like the tool for rotated dimensions. The only difference is that the angle is fixed to 90 degrees (vertical).

6.5 Baseline (DB)

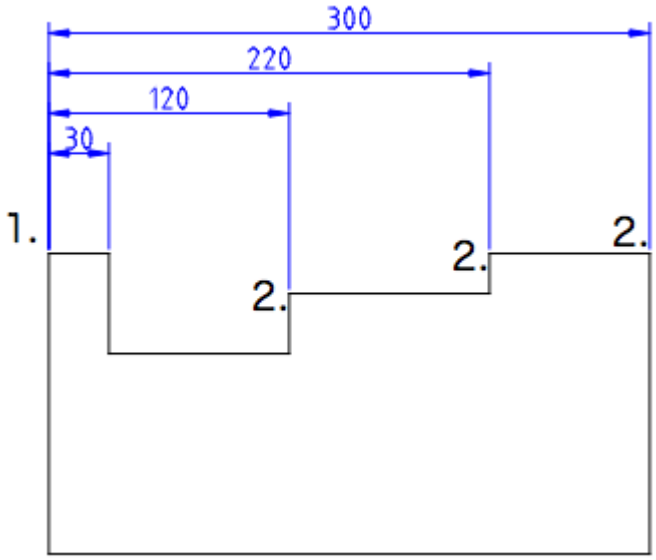
Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Dimension > Baseline`
Shortcut: `D, B`
Commands: `dimbase | baselinedimension | db`

Description

Creates a rotated (linear), angular, or ordinate dimension, starting at the baseline of an existing dimension.

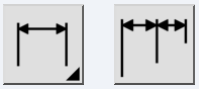


Usage

1. Click the base line of an existing linear, angular or ordinate dimension.
2. Set the second extension line end point with the mouse or enter a coordinate in the command line. The dimension lines are automatically stacked with a fixed distance between them from each other. You can adjust this distance (the dimension line increment) under *Edit > Drawing Preferences > Dimension > Dimension Settings > Dimension line increment*. You can also override this setting for individual dimensions in the options toolbar by unchecking *Default spacing* and entering a value for *Spacing*.

6.6 Continue (DC)

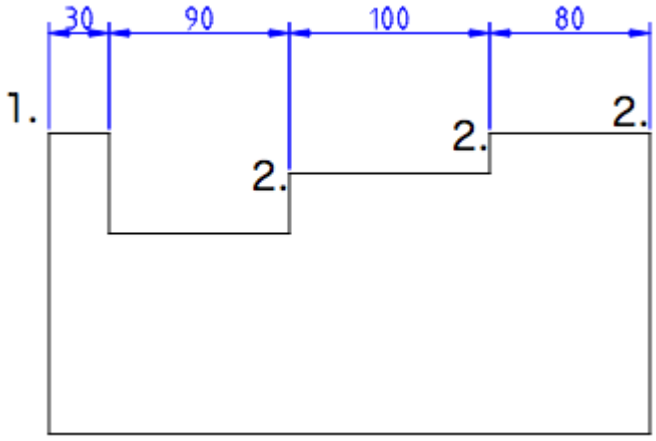
Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Dimension > Continue`
Shortcut: `D, C`
Commands: `dimcontinue | continuedimension | dc`

Description

Creates a dimension that starts from an extension line of a selected linear, angular or ordinate dimension.

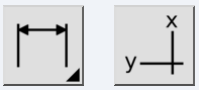


Usage

1. Click the base line of an existing linear, angular or ordinate dimension.
2. Set the second extension line end point with the mouse or enter a coordinate in the command line.
3. Continue to click more extension line end points if desired to create a chain of connected dimensions.

6.7 Ordinate (X/Y) (DO)

Toolbar / Icon:



Menu: `Dimension > Ordinate (X/Y)`
Shortcut: `D, O`
Commands: `dimordinate | do`

Description

This tool draws an ordinate dimension. Ordinate dimensions indicate the X-ordinate or Y-ordinate of a feature in the drawing, relative to an origin point.

Usage

1. Specify the location of the feature with the mouse or by entering a coordinate.
2. Specify the location of the text label and end point of the leader. If the leader is closer to being vertical, the X datum is indicated, if it is closer to being horizontal, the Y datum is indicated.
3. The origin of the ordinate dimension is at the absolute zero point. To move the origin by mouse, select the ordinate dimension entity and move the origin with the mouse (drag and drop). To specify a coordinate for the origin, use the property editor.

QCAD Professional

- Click the mouse arrow symbol in the options tool bar to set the origin for new ordinate dimensions:

  You can then set the new origin location by clicking in the drawing or entering a coordinate in the command line. All new ordinate dimensions created after this measure the distance from that origin.

6.8 Leader (DE)

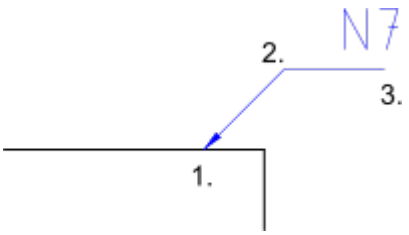
Toolbar / Icon:



Menu: `Dimension > Leader`
Shortcuts: `D, E | L, D`
Commands: `leader | dimlea | de | ld`

Description

Leaders are arrows that usually point from a text entity to another entity as shown below. In the example, the text entity "N7" is describing a surface property by pointing to it with a leader.



Usage

1. Set the location where the leader points to or enter a coordinate in the command line.
2. Set the location of the next edge of the leader line.
3. Set the other edges of the leader line and click the right mouse button or press the Escape key to finish.

6.9 Datum (DT)

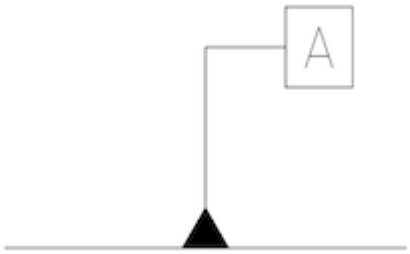
Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Dimension > Datum`
Shortcut: `D, T`
Commands: `datum | dt`

Description

Creates a special type of leader typically used to attach a datum identifier symbol to an edge or surface in a mechanical drawing.



Usage

1. Click the start point of the datum leader. That's the point that indicates which feature the datum refers to.
2. Set the location of the next edge of the leader line.
3. Set the other edges of the leader line and click the right mouse button or press the Escape key to finish.

6.10 Tolerance... (TO)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Dimension > Tolerance…`
Shortcut: `T, O`
Commands: `tolerance | to`

Description

Creates a feature control frame typically used for geometric tolerances or datum identifier symbols.



Usage

1. After starting this tool, the Geometric Tolerance dialog is shown.
2. Choose the desired tolerance symbol and fill in the tolerance values and datum reference letters as desired. To create a datum identifier symbol (for example [A]), only fill in the field "Datum Identifier".
3. Click OK.
4. Click the location for the feature control frame.
5. Click the right mouse button or press the Escape key to terminate the tool.

6.11 Radial (DR)

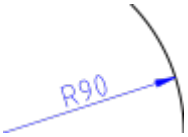
Toolbar / Icon:



Menu: `Dimension > Radial`
Shortcut: `D, R`
Commands: `dimrad | dimradial | dr`

Description

Creates radial dimensions for circle or arc entities.



Usage

1. Choose a circle or arc entity.
2. Set the position of the radial dimension line using the mouse or by entering a coordinate or an angle in the command line.

6.12 Diametric (DD)

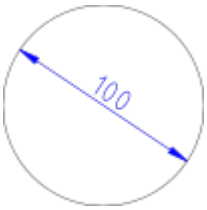
Toolbar / Icon:



Menu: `Dimension > Diametric`
Shortcut: `D, D`
Commands: `dimdia | dimdiametric | dd`

Description

Creates diametric dimensions for circle or arc entities.

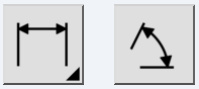


Usage

1. Choose a circle or arc entity.
2. Set the position of the diametric dimension line using the mouse or by entering a coordinate or an angle in the command line.

6.13 Angular (DN)

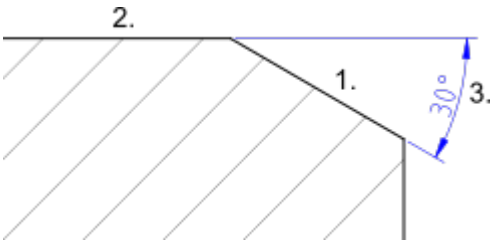
Toolbar / Icon:



Menu: `Dimension > Angular`
Shortcut: `D, N`
Commands: `dimangular | dn`

Description

Creates angular dimensions between two reference lines as shown here:

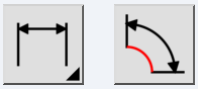


Usage

1. Choose the first line entity.
2. Choose the second line entity.
3. Set the position of the angular dimension arc using the mouse or by entering a coordinate in the command line.

6.14 Arc Length (DG)

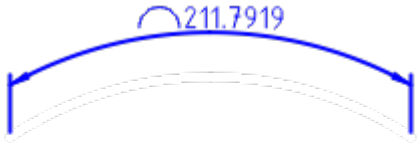
Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Dimension > Arc Length`
Shortcut: `D, G`
Commands: `dimarclength | dg`

Description

Creates arc length dimensions as shown here:





Usage

1. Choose the arc entity.
2. Set the position of the dimension arc using the mouse or by entering a coordinate in the command line.

The position of the arc symbol can be adjusted in the property editor (property "Arc Symbol Type").

6.15 Reset Label Position (DS)

Toolbar / Icon:



Menu: `Dimension > Reset Label Position`
Shortcut: `D, S`
Commands: `dimregen | ds`

Description

This tool resets the label position of all selected dimension entities and places their label back to their automatically computed position.

6.16 Remove Style Overrides (DX)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Dimension > Remove Style Overrides`
Shortcut: `D, X`
Commands: `dimremovestyle | dx`

Description

Removes all dimension style overrides from selected entities. These are dimension style properties that differ from the drawing default.

7 Modify

Toolbar / Icon:



Shortcut: `W, M`
Commands: `modifymenu`

Description

The modify menu contains tools for transforming and editing existing entities, such as moving, rotating, scaling, mirroring, trimming and stretching.

7.1 Transform

7.1.1 Move / Copy (MV)

Toolbar / Icon:




Menu: `Modify > Transform > Move / Copy`
Shortcut: `M, V`
Commands: `move | mv`

Description

Moves or copies entities.

Usage

1. Select the entities you want to move or copy.
2. Launch this tool.
3. Set the reference point with the mouse or enter a coordinate in the command line.
4. Set the target point. To move the selected entities by a given amount, enter a relative coordinate. For example to move it by 50 drawing units to the right, enter @50,0 in the command line.
5. The move dialog is displayed.
   To move the entities, choose "Delete Original", to copy them choose "Keep Original". You can also create any given number of copies at once by choosing "Multiple Copies" and entering the number of copies in the text line below. Note that "9" will create 9 copies and keep the original - so there will be 10 instances of the selected entities in the end.
   The new entities are placed on the same layer as the originals and have the same attributes. To use the current layer and current attributes instead, tick "Use current layer and attributes".
6. Click "OK" to move or copy the entities.

7.1.2 Rotate (RO)

Toolbar / Icon:

Menu: `Modify > Transform > Rotate`
Shortcut: `R, O`
Commands: `rotate | ro`

Description

Rotates entities by a given angle around a given center.

Usage

1. Select the entities you want to rotate.
2. Launch this tool.
3. Set the center of the rotation with the mouse or enter a coordinate in the command line.
4. The rotate dialog is displayed where you can enter the number of copies and the rotation angle. If you want to specify the rotation angle with the mouse cursor, check the mouse cursor button that is displayed beside the angle input:

   To delete the original entities, check "Delete Original", to copy them choose "Keep Original". You can also create any given number of copies by choosing "Multiple Copies".
   The new entities are placed on the same layer as the originals and have the same attributes. To use the current layer and current attributes instead, tick "Use current layer and attributes".
5. Click "OK".
6. If you have previously chosen to specify the rotation angle by mouse, you now have to first specify a reference point for the rotation and then a target point. The rotation angle is the angle enclosed by the reference point, center of rotation and target point.

7.1.3 Scale (SZ)

Toolbar / Icon:

Menu: `Modify > Transform > Scale`
Shortcut: `S, Z`
Commands: `scale | sz`

Description

Scales entities by a given factor towards a given center.

Usage

1. Select the entities you want to scale.
2. Launch this tool.
3. Set the center of the scaling with the mouse or enter a coordinate in the command line.

4. The scale dialog is displayed where you can enter the scale factor.
   If you want to scale with two different factors in X and Y direction, uncheck the button for proportional scaling:
   
   You can then enter two different scaling factors for X and Y.
   If you want to scale the selection by mouse, check the mouse cursor button:
   
5. Click "OK".
6. If you have previously chosen to scale the selection by mouse, you now have to specify a reference point and a target point for the scaling operation.

7.1.4 Mirror (MI)

Toolbar / Icon:


Menu: `Modify > Transform > Mirror`
Shortcut: `M, I`
Commands: `mirror | mi`

Description

Mirrors entities along a given axis.

Usage

1. Select the entities you want to mirror.
2. Launch this tool.
3. Specify a first point on the mirror axis with the mouse or enter a coordinate in the command line.
4. Specify a second point on the mirror axis.
5. The mirror dialog is displayed.
   To mirror the entities without keeping the original entities, choose "Delete Original", to copy them choose "Keep Original".
   The new entities are placed on the same layer as the originals and have the same attributes. To use the current layer and current attributes instead, tick "Use current layer and attributes".
6. Click "OK" to mirror the entities.

7.1.5 Flip Horizontal (FH)

Toolbar / Icon:


Menu: `Modify > Transform > Flip Horizontal`
Shortcut: `F, H`
Commands: `fliphorizontally | fh`

Description

This function flips (mirrors) the current selection horizontally.

7.1.6 Flip Vertical (FV)

Toolbar / Icon:

Menu: `Modify > Transform > Flip Vertical`
Shortcut: `F, V`
Commands: `flipvertically | fv`

Description

This function flips (mirrors) the current selection vertically.

7.1.7 Move and Rotate (MR)

Toolbar / Icon:

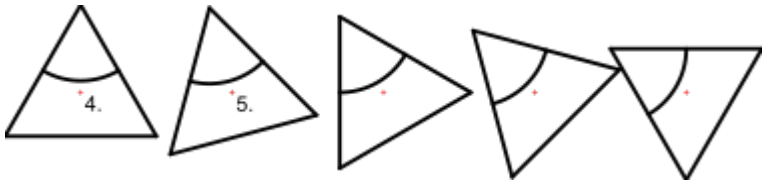
Menu: `Modify > Transform > Move and Rotate`
Shortcut: `M, R`
Commands: `moverotate | mr`

Description

Moves or copies and simultaneously rotates entities.

Usage

1. Select the entities you want to move or copy.
2. Launch this tool.
3. Enter the rotation angle in the options toolbar
4. Set the reference point with the mouse or enter a coordinate in the command line.
5. Set the target point.
   In the figure below, the two reference points are labeled. The rotation angle in the example is 15 degrees and the number of copies four. This results in a total rotation angle of 60 degrees.
6. The move and rotate dialog is displayed.
   To move the entities, choose "Delete Original", to copy them choose "Keep Original". You can also create any given number of copies at once by choosing "Multiple Copies" and entering the number of copies in the text line below.
   The new entities are placed on the same layer as the originals and have the same attributes. To use the current layer and current attributes instead, tick "Use current layer and attributes".
7. Click "OK" to move and rotate the entities.



7.1.8 Rotate Two (R2)

Toolbar / Icon:



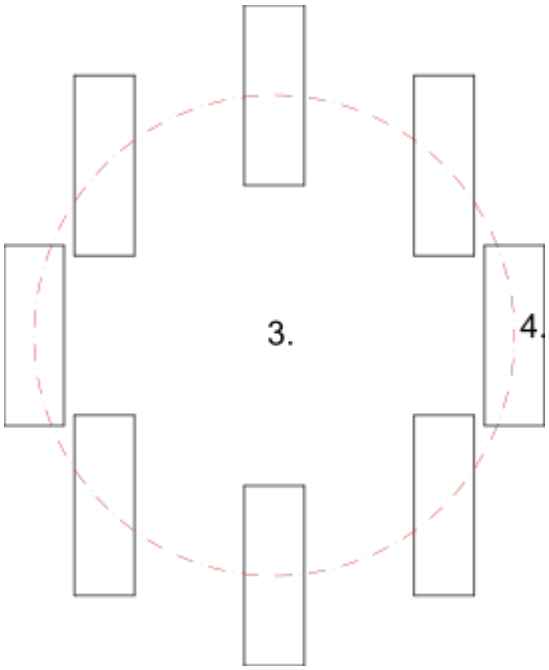

Menu: `Modify > Transform > Rotate Two`
Shortcut: `R, 2`
Commands: `rotate2 | r2`

Description

Rotates and counter-rotates entities. This tool can be useful to rotate entities around a center while maintaining the original orientation of the entities themselves.

Usage

1. Select the entities you want to rotate and counter-rotate.
2. Launch this tool.
3. Set the center for the main rotation with the mouse or enter a coordinate in the command line.
4. Set the center of the rotation for the individual objects. This second center of rotation is rotated together with the entities around the first center.
5. The rotate two dialog is displayed where you can enter the rotation angle and counter-rotation angle.
   To delete the original entities, check "Delete Original", to copy them choose "Keep Original". You can also create any given number of copies by choosing "Multiple Copies".
   The new entities are placed on the same layer as the originals and have the same attributes. To use the current layer and current attributes instead, tick "Use current layer and attributes".
6. Click "OK" to rotate the entities.



7.1.9 Align Reference Points (AE)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Modify > Transform > Align Reference Points`
Shortcut: `A, E`
Commands: `align | ae`

Description

Aligns the selected entities with other, existing entities.

Usage

1. Select the entities you want to align.
2. Launch this tool.
3. Check "Scale" to also scale the selection to fit between the two target points.
4. Check "Copy" to create an aligned copy of the selected entities.
5. Set the first reference point with the mouse or enter a coordinate in the command line. This point will be aligned with the first target point.
6. Set the first target point.
7. Set the second reference point to be aligned with the second target point.
8. Set the second target point.
9. The selection is moved, rotated and optionally scaled in such a way that the two chosen reference points align with the two chosen target points.

7.1.10 Align (MA)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Modify > Transform > Align`
Shortcut: `M, A`
Commands: `alignhv`

Description

With this tool, the current selection can be aligned relative to an entity in the drawing, relative to the drawing bounding box or relative to the biggest or smallest entity in the selection.

Usage

1. Select the entities you want to align.
2. Start the tool.
3. In the options toolbar, choose to what you want to align the selection (a picked entity, the whole drawing or a position).
4. Choose how you want to align the selection (left, center, right, top, middle, bottom).
5. Tick the check box "Treat selection as group" to treat the geometry of the selected entities as one.
6. If you have chosen to align to a picked entity, click the entity now. When aligning to a position, choose the position. When aligning to the whole drawing, click the tick button in the options toolbar:



7.2 Offset

7.2.1 Offset (with Distance) (OF)

Toolbar / Icon:

Menu: `Modify > Offset > Offset (with Distance)`
Shortcut: `O, F`
Commands: `offset | of`

Description

With this tool you can create parallels to existing lines (or concentric arcs and circles).

Usage

1. Enter the distance of the concentric or parallel entity from the original entity in the options tool bar shown at the top.

2. Enter the number of parallel or concentric entities to create in the options tool bar.
3. Click the base entity. The parallel or concentric entities are created on that side on which the mouse cursor is located while selecting the entity.

QCAD Professional

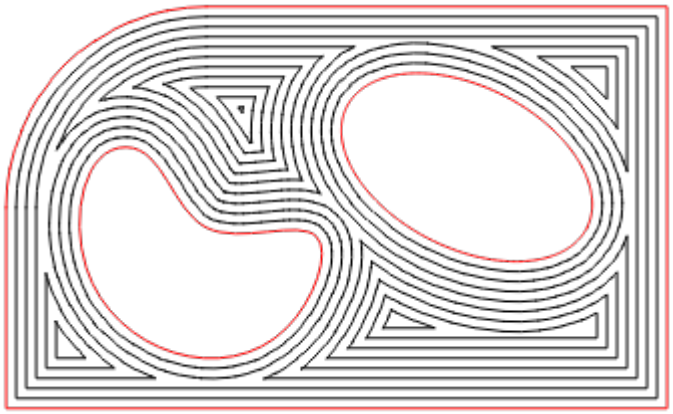
Additional options for the handling of polylines are shown in the options tool bar:

Mode

- Polyline Mode: In this mode, the offset tool offsets entire polylines and other contours at once.

  If multiple polylines or contours are selected, they are all offset together as shown here:

- Segment Mode: In this mode, individual segments of polylines can be offset.

Corners

Offset segments that do not intersect each other (segments at convex corners of the original contour) can be connected in two ways:

- Round Corners: The segments are connected with an arc (round join).

- Sharp Corners: The segments are extended until they intersect each other, forming a sharp corner (miter join). If the extended segments cannot intersect (for example at a 180 degree turn), an arc is used instead.

7.2.2 Offset (through Point) (OH)

Toolbar / Icon:

Menu: `Modify > Offset > Offset (through Point)`
Shortcut: `O, H`
Commands: `offsetthrough | oh`

Description

With this tool you can create parallels to existing lines or concentric arcs and circles. The parallel or concentric arc or circle goes through a given point.

7.3 Trim and Divide

7.3.1 Trim (RM)

Toolbar / Icon:

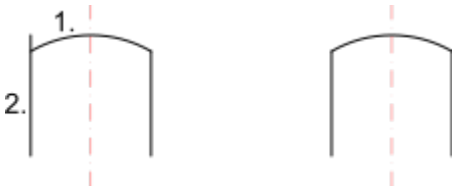
Menu: `Modify > Trim and Divide > Trim`
Shortcuts: `R, M | X, T`
Commands: `trim | extend | rm | xt`

Description

Trims or extends a line, arc or ellipse to another entity.

Usage

1. Choose the limiting entity to which you want to trim or extend one or more other entities.
2. Choose the entities that you want to trim to the limiting entity. Often there are two possibilities for the entity to be trimmed. In the example below, you might want the top part of the line to remain in the drawing and the lower part to be trimmed away. In that case you would have to click the trim entity at the upper part. Always click that part of the entity which you want to retain.
3. Right click twice or press Escape twice to terminate the tool.



QCAD Professional

- In the options toolbar you can activate the line mode to trim all entities intersected by a line.
- In this mode, you choose the limiting entity first and then click a start and an end point to trim all entities that are intersected by the line between the two points.

- As when clicking entities one by one, the part of an entity which is crossed by the line is retained.

7.3.2 Trim Both (TM)

Toolbar / Icon:

Menu: `Modify > Trim and Divide > Trim Both`
Shortcut: `T, M`
Commands: `trim2 | extend2 | tm`

Description

Trims or extends two lines, arcs or ellipses to their common intersection point.

Usage

1. Choose the first entity to be trimmed.
2. Choose the second entity to be trimmed.
3. Right click twice or press Escape twice to terminate the tool.



7.3.3 Lengthen / Shorten (LE)

Toolbar / Icon:

Menu: `Modify > Trim and Divide > Lengthen / Shorten`
Shortcut: `L, E`
Commands: `lengthen | shorten | trimamount | le`

Description

This tool can be used to either lengthen or shorten lines or arcs by a given amount.

Usage

1. Enter the distance you want to lengthen an entity by in the options tool bar. A positive value lengthens the entity, while a negative value shortens it. E.g. a value of '5' will lengthen the chosen entity by 5 units.
2. Choose the entity to be lengthened or shortened somewhere close to the end you want to modify.
3. Right click or press the Escape key to terminate the tool.



7.3.4 Stretch (SS)

Toolbar / Icon:

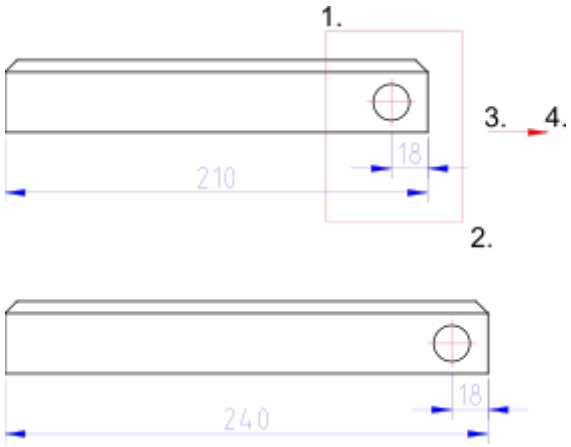


Menu: `Modify > Trim and Divide > Stretch`
Shortcut: `S, S`
Commands: `stretch | ss`

Description

This tool stretches contours and dimension entities. This can also be described as moving all endpoints within a given rectangular or polygonal area.

If there is a selection of entities, this tool only affects the selected entities. Otherwise, this tool operates on all entities in the given area.



Usage

1. Choose in the options toolbar if you want to stretch a rectangular area (rectangle symbol) or a polygonal area (polygon symbol).
2. Set the first corner of the stretch area.
3. Set the second corner of the stretch area.
4. For polygonal areas, set the other corners of the stretch area and right-click the last corner when finished.
5. Set the reference point for moving using the mouse or enter a coordinate in the command line.
6. Set the target point.

QCAD Professional

- In the options toolbar you can choose to stretch a rectangular or a polygonal area.
- For polygonal areas, all corner points of the polygon must be clicked. The last corner point is clicked with the right mouse button.

7.3.5 Clip to Rectangle (CLR)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Modify > Trim and Divide > Clip to Rectangle`
Shortcut: `C, L, R`
Commands: `cliprectangle | cliprect | clr`

Description

Clips the current selection to the boundaries of a rectangular area.

Usage

1. Select the entities you want to clip.
2. Specify the first corner of the rectangular area.
3. Move the mouse to the second corner and click to specify the second corner of the rectangular area.

7.3.6 Divide (DI)

Toolbar / Icon:

Menu: `Modify > Trim and Divide > Divide`
Shortcut: `D, I`
Commands: `divide | di`

Description

Divides entities into two separate entities.

Usage

1. Choose the entity you want to divide.
2. Set the dividing point using the mouse. The dividing point is usually the intersection point with another entity. Choose the intersection snap mode to automatically snap to intersections.

7.3.7 Split Entities (MS)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Modify > Trim and Divide > Split Entities`
Shortcut: `M, S`
Commands: `split | ms`

Description

This tool splits the selected lines, arcs and circles up into N equally long segments.

Usage

1. Select the entities you want to split up.
2. Launch this tool.
3. Use the options tool bar to specify into how many segments you want to split up the entities. If any circles are selected, you may also enter an angle offset for splitting up these circles.
4. Click the button with the tick to split up the selected entities:



7.3.8 Break out Segment (D2)

Toolbar / Icon:




Menu: `Modify > Trim and Divide > Break out Segment`
Shortcut: `D, 2`
Commands: `break | breakout | d2`

Description

Divides entities by cutting out a segment that is limited by intersecting entities.

Usage

Choose the entity you want to divide by clicking the segment you want to cut out. The segment is removed automatically if the check box "Remove Segment" is checked. If not, the entity is only cut at the intersections without removing the segment.

QCAD Professional

- In the options toolbar you can activate the line mode to break out all segments intersected by a line.
- In this mode, you can click a start and an end point to break out all segments that are intersected by the line between the two points.

7.3.9 Break out Manual (B2)

Toolbar / Icon:




Menu: `Modify > Trim and Divide > Break out Manual`
Shortcut: `B, 2`
Commands: `brk | b2`

Description

Divides objects by cutting out a segment in between two user defined points.

Usage

1. Select the entity you want to break up.
2. Click the first point at which the entity is to be divided.
3. If the point does not lie on the entity, the separating point is assumed to be the closest point on the entity.
4. Click the second point at which the entity is to be divided.
5. The segment between the two points is removed if the "Remove Segment" option in the options toolbar is active. Otherwise, the entity is only separated at the given points.

7.3.10 Auto Trim (AX)

Toolbar / Icon:

Menu: `Modify > Trim and Divide > Auto Trim`
Shortcut: `A, X`
Commands: `autotrim | ax`

Description

Trims or extends an entity on both sides to the next limiting entities.

Usage

- Choose the entity you want to trim or extend by clicking the segment you want to keep. The entity is automatically trimmed or extended, so that the two end points align with the next two entities.

QCAD Professional

- In the options toolbar you can activate the line mode to trim all entities intersected by a line.
- In this mode, you can click a start and an end point to trim all entities that are intersected by the line between the two points.
- As when clicking entities one by one, the segment which is crossed by the line is retained.

7.3.11 Break out Gap (D3)

Toolbar / Icon:

Menu: `Modify > Trim and Divide > Break out Gap`
Shortcut: `D, 3`
Commands: `breakoutgap | d3`

Description

Breaks out a gap from a line, arc, circle, ellipse, ellipse arc, spline or polyline.

Usage

- Pick the entity from which you wish to break out a gap.
- Click the center of the gap on the entity.

QCAD Professional

- In QCAD Professional, gaps can also be broken out from ellipses, ellipse arcs, splines and polylines.
- The gap is centered at the chosen point and its length is measured along the entity.
- Breaking out a gap from a closed entity (circle, ellipse, closed spline or closed polyline) results in a single open entity.
- If the option "One click" is checked in the options toolbar, the gap is broken out at the position where you click the entity. There is no need to click the center of the gap separately.

7.4 Corners

7.4.1 Chamfer / Bevel (CH)

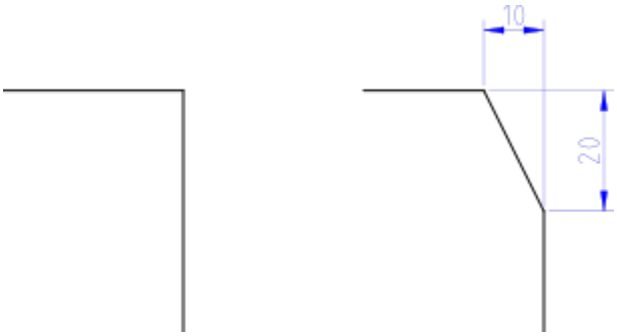
Toolbar / Icon:




Menu: `Modify > Corners > Chamfer / Bevel`
Shortcut: `C, H`
Commands: `bevel | chamfer | ch`

Description

Bevels a corner shaped by two entities. Optionally, the edge entities of the corner can be trimmed automatically to fit the new shape.



Usage

1. Enter the geometry of the bevel in the options tool bar. "Distance 1" is the distance the bevel line will have from the (imaginary) intersection of the two edges (10 in the example shown). "Distance 2" is the same distance for the second edge (20 in the example shown).
2. Check "Trim" if you want to trim the entities automatically. If the option is disabled, the two entities will remain untouched.
3. Choose the first edge entity (a line or an arc).
4. Choose the second entity.

7.4.2 Round (RN)

Toolbar / Icon:



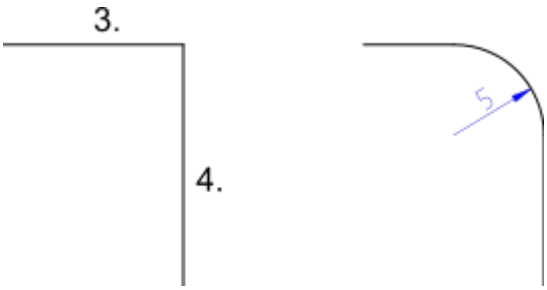

Menu: `Modify > Corners > Round`
Shortcut: `R, N`
Commands: `round | rn`

Description

Rounds the corner of a contour. Optionally, the edge entities of the corner can be trimmed automatically to fit the new shape.

Usage

1. Enter the radius of the rounding in the options tool bar (e.g. 5).
2. Check "Trim" if you want to trim the entities automatically. If the option is disabled, the two corner entities will remain untouched.
3. Choose the first edge entity (a line or arc).
4. Choose the second entity.



7.5 Reverse (RV)

Toolbar / Icon:




Menu: `Modify > Reverse`
Shortcut: `R, V`
Commands: `reverse | rv`

Description

This tool reverses the direction of all selected line, arc and ellipse arc entities. This is mostly useful for drawings that are being prepared for further processing (e.g. CAM).

7.6 Entities

7.6.1 Edit Text (MT)

Toolbar / Icon:

Menu: `Modify > Entities > Edit Text`
Shortcut: `M, T`
Commands: `edittext | modifytext | mt`

Description

This tool can be used to edit existing text entities.

Usage

1. Click the text entity you want to edit.
2. The text dialog is shown where you can adjust the text and its attributes.

Note that instead of using this tool, you can also double-click the text entity that you want to edit.

7.6.2 Edit Tolerance (ML)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Modify > Entities > Edit Tolerance`
Shortcut: `M, L`
Commands: `edittolerance | modifytolerance | ml`

Description

This tool can be used to edit existing text entities.

Usage

1. Click the text entity you want to edit.
2. The text dialog is shown where you can adjust the text and its attributes.

Note that instead of using this tool, you can also double-click the text entity that you want to edit.

7.7 Explode (XP)

Toolbar / Icon:

Menu: `Block > Explode`
Shortcut: `X, P`
Commands: `explode | xp`

Description

Converts block references, polylines, splines, ellipses, texts, dimensions and hatches into more basic entities. Dimensions are split up into lines, triangles and text entities. Text entities are split into polylines (or lines, arcs and splines). Splines are converted into polylines with tangentially connected arc segments. Ellipses are converted into polylines with arc segments. Polylines are split up into loose segments. Hatches are split up into lines. Solid fills are converted to the outline of the solid fill.

Usage

1. Select the entities you want to explode.
2. Launch this tool to explode the selected entities.
3. Repeat step 2 to further break up the selected entities if desired. If an entity cannot be further broken up, it remains unchanged.

7.8 Draw Order

Toolbar / Icon:



Description

Tools for changing the draw order (stacking order) of entities, which determines which entities are drawn on top of others.

7.8.1 Bring to Front (MF)

Toolbar / Icon:



Menu: `Modify > Draw Order > Bring to Front`
Shortcut: `M, F`
Commands: `tofront | mf`

Description

Changes the order in which entities are displayed. Use this tool to bring entities into the foreground if they are hidden behind images, solid fills or other entities.

Usage

1. Select the entities to bring into the foreground of the drawing.
2. Launch this tool.

7.8.2 Send to Back (MB)

Toolbar / Icon:



Menu: `Modify > Draw Order > Send to Back`
Shortcut: `M, B`
Commands: `toback | mb`

Description

Changes the order in which entities are displayed. Use this tool to send entities that are hiding other entities into the background.

Usage

1. Select the entities to move into the background of the drawing.
2. Launch this tool.

7.8.3 Reorder (MO)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Modify > Draw Order > Reorder`
Shortcut: `M, O`
Commands: `reorder | mo`

Description

With this tool you can reorder entities manually. The draw order of entities is usually not important unless an entity is hiding another one. However, for certain types of further processing (such as CAM), the order of entities can be crucial.

Usage

1. To reorder all entities in your drawing, make sure that nothing is selected. Otherwise, select the entities you wish to reorder. Note that if you reorder only selected entities, the order of all other entities remains unchanged. Other entities might be drawn before, after or in between selected entities.
2. Click the entity that should be drawn or processed first and subsequently click all entities in the desired order. When the last entity is clicked, the tool automatically terminates. The numerical order of the entities is shown as an overlay while this tool is running.

7.8.4 Order Connected Entities (MC)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Modify > Draw Order > Order Connected Entities`
Shortcut: `M, C`
Commands: `orderconnected | mc`

Description

Reorders connected entities so that they follow one another in sequence, for example to prepare a continuous cutting path.

7.9 Detection

Toolbar / Icon:



Description

Tools for detecting zero length entities or duplicate entities.

7.9.1 Detect Duplicates (MD)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Modify > Detection > Detect Duplicates`
Shortcut: `M, D`
Commands: `duplicates`

Description

This tool can be used to find, select and optionally delete identical entities in the current drawing.

Usage

1. After launching the tool, the options tool bar displays controls to enter tolerance values for the comparison of entities. Adjust these if appropriate.
2. Tick the option "Ignore Layer" if you want two identical entities on different layers to be considered equal.
3. Click the refresh button to run the search again if you have made any adjustments to the tolerances and the "Auto" check box is not checked:



If the "Auto" check box is checked, the search for duplicates is automatically repeated if you change any of the options.

4. Click the delete button if you wish to delete the selected duplicates:



5. Click the close button to terminate this tool:



7.9.2 Detect Zero-Length Entities (MZ)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Modify > Detection > Detect Zero-Length Entities`
Shortcut: `M, Z`
Commands: `zerolength | zero`

Description

This tool can be used to find, select and optionally delete very small entities (length almost 0) in the current drawing.

Usage

1. After launching the tool, the options tool bar displays controls to enter tolerance values that are used to detect zero length entities. Adjust these if appropriate.
2. Click the refresh button to run the search again if you have made any adjustments to the tolerances and the "Auto" check box is not checked:



If the "Auto" check box is checked, the search for duplicates is automatically repeated if you change any of the options.

3. Click the delete button if you wish to delete the selected zero length entities:



4. Click the close button to terminate this tool:



7.10 Projection

Toolbar / Icon:



Shortcut: `W, J`
Commands: `projectmenu`

Description

Tools for projecting geometry, for example into isometric views.

7.10.1 Orthographic Projection (Cylindrical) (PR)

Toolbar / Icon:



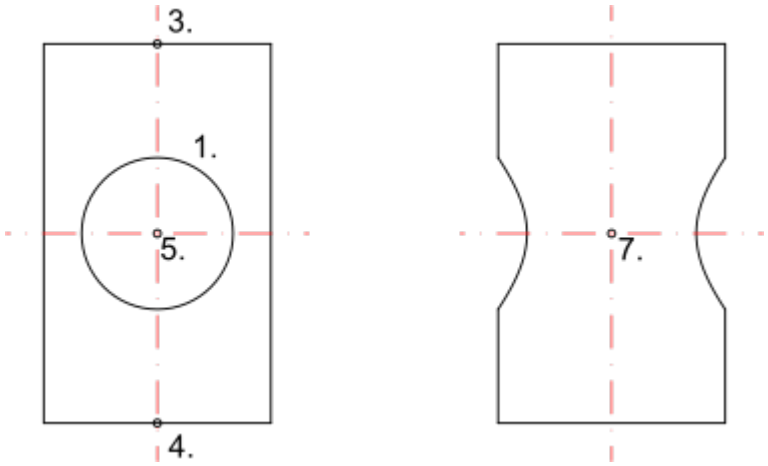
Add-on: QCAD Professional
Menu: `Modify > Projection > Orthographic Projection (Cylindrical)`
Shortcut: `P, R`
Commands: `orthocylindrical | orthocyl | pr`

Description

Creates an orthographic projection of the selected entities onto a cylindric surface. This is typically used to create the edge of a drilling in a side view of a cylinder as shown in the figure below on the right.

Usage

1. Select the entities you want to project. In the example below, that is the circle at the left.
2. Start this tool.
3. Click the start point of the unprojected cylinder axis. In the example below that is at the top center of the cylinder at the left.
4. Click the end point of the cylinder axis.
5. Click the reference point. The reference point in the selection will be used to place the projected entities into the side view.
6. Adjust the parameters in the options tool bar as desired.
   The cylinder diameter is the diameter of the cylindric shape we are projecting onto.
   The cylinder rotation is the rotation of the projected entities around the cylinder axis. Typically 90 or -90 degrees.
   Since the resulting shapes are likely neither lines or arcs, projected entities are split up into polylines with small segments. You can adjust the segment length used in the options tool bar.
   The clipping option can be enabled to automatically hide the part of the projected shape that would lay behind the X/Y plane.
7. Click the target point to position the projection.



7.10.2 Isometric Projection (PJ)

Toolbar / Icon:



Menu: `Modify > Projection > Isometric Projection`
Shortcut: `P, J`
Commands: `isometric | pj`

Description

This tool creates isometric projections (and other types of projections) of the current selection in the drawing.

Usage

1. Select the entities you want to project.
2. Launch this tool.
3. Choose a projection type and a viewing direction for the projection in the options tool bar.
4. Set the reference point for the projection. This is the point you will use to position the projection in the next step.
5. Move the mouse cursor to the position where you want to create the projection and click to set the position.

7.10.3 Isometric Projection (Cylindrical) (PC)

Toolbar / Icon:



Add-on: **QCAD Professional**
Menu: `Modify > Projection > Isometric Projection (Cylindrical)`
Shortcut: `P, C`
Commands: `isometriccylindrical | isocyl | pc`

Description

This tool creates an isometric projection of the current selection onto a cylindric surface.

Usage

1. Select the entities you want to project.
2. Launch this tool.
3. Choose a viewing direction for the projection in the options tool bar.
4. Enter the diameter of the cylinder in the options tool bar.
5. Enter the rotation of the projected entities around the cylinder axis in degrees in the options tool bar.
6. Set the start point of the unprojected cylinder axis.
7. Set the end point of the cylinder axis.
8. Set the reference point for the projection. This is the point you will use to position the projection in the next step.
9. Move the mouse cursor to the position where you want to create the projection and click to set the position. Before setting the definitive position, you might still want to adjust the cylinder diameter, rotation angle and viewing direction in the options tool bar, until the preview matches your expectations.

7.10.4 Matrix Transformation (MX)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Modify > Projection > Matrix Transformation`
Shortcut: `M, X`
Commands: `matrixtransform | mx`

Description

This tool transforms the selection using an arbitrary user defined transformation matrix.

Usage

1. Select the entities you want to transform.
2. Launch this tool.
3. Enter the transformation matrix in the options tool bar.
4. For convenience, you may additionally enter a factor that is applied to each element of the matrix.
5. Set the reference point for the transformation. This is the point you will use to position the transformed entities in the next step.
6. Move the mouse cursor to the position where you want to create the transformation and click to set the position. Before setting the definitive position, you might still want to adjust the transformation matrix in the options tool bar, until the preview matches your expectations.

7.11 Property Painter (PI)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Modify > Property Painter`
Shortcut: `P, I`
Commands: `propertypainter | pi`

Description

This tool applies properties of a chosen entity to the current selection. Properties that can be copied for all entities are:

- Layer
- Color
- Lineweight
- Linetype
- Custom properties

If the source and destination entities are both text entities, the following text specific properties can be copied:

- Text font
- Text height
- Bold property
- Italic property

For hatch entities, the pattern can be copied:

- Solid property
- Pattern
- Scale
- Angle

Usage

1. Select the target entities to which you want to apply the properties.
2. Start this tool.
3. Choose in the options toolbar which properties you want to copy.
4. Click the entity from which the properties are copied.

8 Snap

Toolbar / Icon:



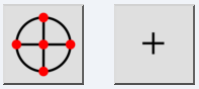
Commands: `snapmenu`

Description

The snap menu contains tools for controlling how the cursor snaps to coordinates, entities and reference points while drawing.

8.1 Free (SF)

Toolbar / Icon:



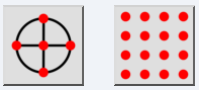
Menu: `Snap > Free`
Shortcut: `S, F`
Commands: `snapfree | sf`

Description

Allows free positioning using the mouse. Note that this is almost never the recommended way of setting coordinates in a CAD system except for drawing freehand lines.

8.2 Grid (SG)

Toolbar / Icon:



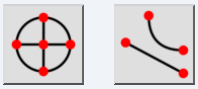
Menu: `Snap > Grid`
Shortcut: `S, G`
Commands: `snapgrid | sg`

Description

Snaps to grid points.

8.3 End (SE)

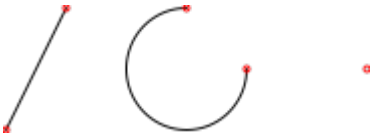
Toolbar / Icon:



Menu: `Snap > End`
Shortcut: `S, E`
Commands: `snapend | se`

Description

Snaps to end points of lines, arcs polyline segments, splines, ellipse arcs and to points.



8.4 On Entity (ST)

Toolbar / Icon:



Menu: `Snap > On Entity`
Shortcut: `S, T`
Commands: `snaponentity | st`

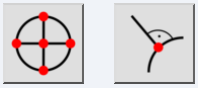
Description

Snaps to the closest point on an entity.



8.5 Perpendicular (SU)

Toolbar / Icon:



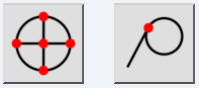
Menu: `Snap > Perpendicular`
Shortcut: `S, U`
Commands: `snapperpendicular | su`

Description

Snaps to the perpendicular point on a line, arc, circle or ellipse. This only applies when drawing lines.

8.6 Tangential (SB)

Toolbar / Icon:



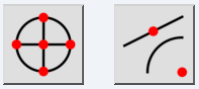
Menu: `Snap > Tangential`
Shortcut: `S, B`
Commands: `snaptangential | sb`

Description

Snaps to the point of tangency on an arc, circle or ellipse. This only applies when drawing lines. The snap point is the tangent point of an imaginary line from the relative zero point, tangential to the clicked arc or circle.

8.7 Center (SC)

Toolbar / Icon:



Menu: `Snap > Center`
Shortcut: `S, C`
Commands: `snapcenter | sc`

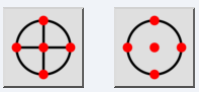
Description

Snaps to the center of arcs, circles and ellipses as well as to the middle of lines and to points.



8.8 Reference (SR)

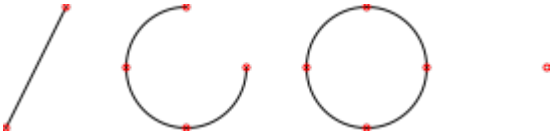
Toolbar / Icon:



Menu: `Snap > Reference`
Shortcut: `S, R`
Commands: `snapreference | sr`

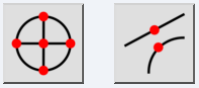
Description

Snaps to reference points. Reference points are the blue points that are shown when an entity is selected. This is particularly useful to snap to reference points of circles and arcs, for example when drawing center lines.



8.9 Middle (SM)

Toolbar / Icon:



Menu: `Snap > Middle`
Shortcut: `S, M`
Commands: `snapmiddle | sm`

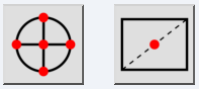
Description

Snaps to middle points of lines and arcs and to point entities. Note that the middle of an arc is at the middle of the arc line on the arc, not at its center.



8.10 Middle Manual (SN)

Toolbar / Icon:



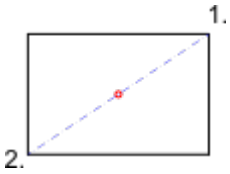
Menu: `Snap > Middle Manual`
Shortcut: `S, N`
Commands: `snapmiddlemanual | sn`

Description

This tool allows you to snap to a point that is in the middle between two points. This is most commonly used to snap to the center of a rectangle or polygon by choosing two diagonally opposite corners.

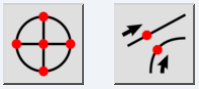
Usage

1. After activating this snap tool, specify the first of the two points that define the middle point. For example one corner of a rectangle.
2. Click the second of the two points, for example the diagonally opposite corner of the rectangle.



8.11 Distance (SD)

Toolbar / Icon:



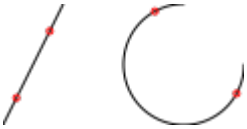
Menu: `Snap > Distance`
Shortcut: `S, D`
Commands: `snapdistance | sd`

Description

Snaps to points with a given distance to the end point of lines or arcs.

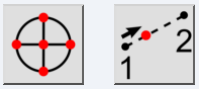
Usage

1. Enter the distance into the options toolbar.
2. Move the cursor to the entity along which the distance should be measured.
3. When the desired position / preview is shown, click the left mouse button to set the coordinate.



8.12 Distance Manual (SH)

Toolbar / Icon:



Menu: `Snap > Distance Manual`
Shortcut: `S, H`
Commands: `snapdistancemanual | sh`

Description

This tool allows you to snap to a point that lays on the imaginary line between two points at a given distance from the first point.
This snap tool does not restrict the distance, percentage or fraction entered. Entering a distance greater than the distance between the two points, a percentage greater than 100 or a fraction greater than 1.0 will allow you to snap to a point beyond the second point.
Negative values can be entered to snap to points beyond the first point.
Fractions may be entered as decimal numbers (0.5, 0.7) or as fractions (1/7, 3/11).

Usage

1. After activating this snap tool, choose "Distance", "Percentage", or "Fraction" and enter the desired value in the options tool bar.

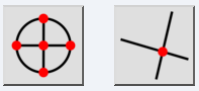


2. Specify the first point. That is the point to measure the distance from.
2. Click the second point.



8.13 Intersection (SI)

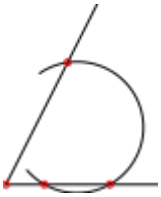
Toolbar / Icon:



Menu: `Snap > Intersection`
Shortcut: `S, I`
Commands: `snapintersection | si`

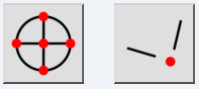
Description

Snaps to intersection points between entities.



8.14 Intersection Manual (SY)

Toolbar / Icon:



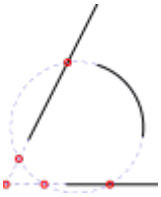
Menu: `Snap > Intersection Manual`
Shortcut: `S, Y`
Commands: `snapintersectionmanual | sy`

Description

Sometimes the intersection snap mode cannot be used to snap to an intersection point because the intersection point lays outside one or both of the entities. This snap tool lets you explicitly specify two intersecting entities and snap to their intersection point.

Usage

1. After activating this snap tool, click the first of the two intersecting entities.
2. Click the second of the two intersecting entities. If two intersection points are possible, make sure to click the second entity somewhere closer to the intersection point you want to snap to.



8.15 Auto (SA)

Toolbar / Icon:



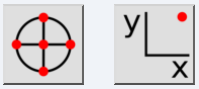
Menu: `Snap > Auto`
Shortcut: `S, A`
Commands: `snapauto | sa`

Description

The auto snap tool snaps to the closest intersection point, end point, middle point, perpendicular point, reference point, grid point or point on an entity in that order of priority.
The types of reference points the auto snap tool snaps to can be configured under *Edit* > *Application Preferences* > *Snap* > *Auto snap*.

8.16 Coordinate (SX)

Toolbar / Icon:



Menu: `Snap > Coordinate`
Shortcut: `S, X`
Commands: `snapcoordinate | sx`

Description

Defines a point by entering an absolute or relative Cartesian coordinate.

Usage

1. Start this tool when you have to specify a point and want to do so by entering a pair of coordinates (x/y).
2. Enter the coordinates in the options tool bar and choose if they are absolute coordinates or a relative ones (relative to the relative zero point).
3. Click the OK button or press Enter to confirm the input and set the coordinates:



8.17 Polar Coordinate (SO)

Toolbar / Icon:




Menu: `Snap > Polar Coordinate`
Shortcut: `S, O`
Commands: `snappolar | so`

Description

Defines a point by entering an absolute or relative polar coordinate.

Usage

1. Start this tool when you have to specify a point and want to do so by entering a polar coordinate.
2. Enter the polar coordinate (radius and angle) in the options tool bar and choose if it is an absolute coordinate or a relative one (relative to the relative zero point).
3. Click the OK button or press Enter to confirm the input and set the coordinate:



8.18 X/Y from Points (PX)

Toolbar / Icon:




Menu: `Snap > X/Y from Points`
Shortcut: `P, X`
Commands: `px`

Description

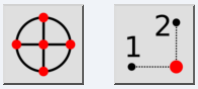
This snap tool can be used to extract one coordinate (e.g. X) from the location of one existing entity and the other coordinate (e.g. Y) from the location of another object.
This can for example be used to locate the center of a rectangle.
This concept is also referred to as 'combining coordinates', 'point filtering' or 'coordinate filtering'.

Usage

1. When specifying a position, start this tool.
2. Click a position in the drawing to lock the X coordinate at that position.
3. Click a position in the drawing to lock the Y coordinate at that position.

8.19 Y/X from Points (PY)

Toolbar / Icon:



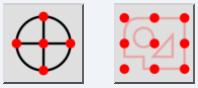
Menu: `Snap > Y/X from Points`
Shortcut: `P, Y`
Commands: `py`

Description

This tool works the same way as the tool X/Y from Points, but with opposite order of coordinate input.

8.20 Selection Reference Points (SK)

Toolbar / Icon:



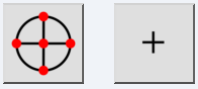
Menu: `Snap > Selection Reference Points`
Shortcut: `S, K`
Commands: `snapselection | sk`

Description

Snaps to one of nine reference points of the bounding box around the current selection: the four corners, the middle points of the four edges or the center. The point closest to the mouse cursor is used.
This snap tool is useful for moving, copying or otherwise aligning a selection relative to the grid or to other entities, for example to place the center or a corner of the selection at a specific position.

8.21 Restrictions Off (EN)

Toolbar / Icon:



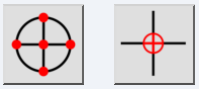
Menu: `Snap > Restrictions Off`
Shortcut: `E, N`
Commands: `restrictoff | en`

Description

Switches off all snap restrictions.

8.22 Restrict Orthogonally (EO)

Toolbar / Icon:



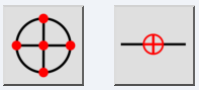
Menu: `Snap > Restrict Orthogonally`
Shortcut: `E, O`
Commands: `restrictorthogonal | eo`

Description

Restricts the cursor vertically or horizontally to the same X or Y position as the relative zero point.

8.23 Restrict Horizontally (EH)

Toolbar / Icon:



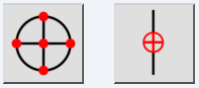
Menu: `Snap > Restrict Horizontally`
Shortcut: `E, H`
Commands: `restricthorizontal | eh`

Description

Restricts the cursor horizontally to the same Y position as the relative zero point.

8.24 Restrict Vertically (EV)

Toolbar / Icon:



Menu: `Snap > Restrict Vertically`
Shortcut: `E, V`
Commands: `restrictvertical | ev`

Description

Restricts the cursor vertically to the same X position as the relative zero point.

8.25 Restrict Angle or Length (EL)

Toolbar / Icon:



Menu: `Snap > Restrict Angle or Length`
Shortcut: `E, L`
Commands: `restrictanglelength | el`

Description

Restricts the cursor movement by angle and distance in relation to the relative zero point.

8.26 Set Relative Zero (RZ)

Toolbar / Icon:



Menu: `Snap > Set Relative Zero`
Shortcut: `R, Z`
Commands: `setrelativezero | rz`

Description

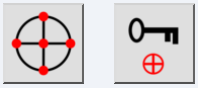
Lets you set a new location for the relative zero point.

Usage

- Click the new location of the relative zero point.

8.27 Lock Relative Zero (RL)

Toolbar / Icon:



Menu: `Snap > Lock Relative Zero`
Shortcut: `R, L`
Commands: `lockrelativezero | rl`

Description

When enabled, this switch locks the position of the relative zero point. This means it does not move around automatically but you can still move it manually using the tool to set the position of the relative zero point.

8.28 Lock Snap (SQ)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Snap > Lock Snap`
Shortcut: `S, Q`
Commands: `locksnap | sq`

Description

When enabled, this switch keeps the current snap tool active even when a new tool is started.

9 Info

Toolbar / Icon:



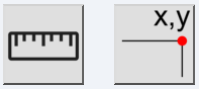
Shortcut: `W, I`
Commands: `infomenu | informationmenu | measuringmenu`

Description

The tools in the Info menu can be used to measure distances, areas and query other information from the drawing.
Some of the tools offer an option to add the measured result to the drawing as text if desired.

9.1 Position (IO)

Toolbar / Icon:



Menu: `Info > Position`
Shortcut: `I, O`
Commands: `infopos | io`

Description

This tool outputs the absolute, Cartesian coordinates of chosen points in the drawing.

9.2 Relative Position (IV)

Toolbar / Icon:



Menu: `Info > Relative Position`
Shortcut: `I, V`
Commands: `infoposrel | iv`

Description

This tool outputs the relative, Cartesian coordinates of chosen points in the drawing.

9.3 Polar Position (IL)

Toolbar / Icon:



Menu: `Info > Polar Position`
Shortcut: `I, L`
Commands: `infopospol | il`

Description

This tool outputs the absolute, polar coordinates of chosen points in the drawing.

9.4 Relative Polar Position (IT)

Toolbar / Icon:



Menu: `Info > Relative Polar Position`
Shortcut: `I, T`
Commands: `infopospolrel | it`

Description

This tool outputs the relative, polar coordinates of chosen points in the drawing.

9.5 Distance Point to Point (IP)

Toolbar / Icon:



Menu: `Info > Distance Point to Point`
Shortcut: `I, P`
Commands: `infodist | ip`

Description

This tool measures the exact distance between two points given by the user.

Usage

1. Use the mouse to specify the location of the first point or enter a coordinate in the command line.
2. Specify the second point.
3. The measured distance is printed on screen and in the command line.

9.6 Distance Entity to Point (IE)

Toolbar / Icon:



Menu: `Info > Distance Entity to Point`
Shortcut: `I, E`
Commands: `infodistep | ie`

Description

This tool measures the exact distance between an entity and a point given by the user.

Usage

1. Specify the entity.
2. Use the mouse to specify the location of the point or enter a coordinate in the command line.
3. The measured distance is printed on screen and in the command line.

9.7 Distance Entity to Entity (IN)

Toolbar / Icon:



Menu: `Info > Distance Entity to Entity`
Shortcut: `I, N`
Commands: `infodistee | in`

Description

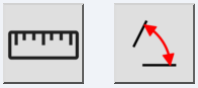
This tool measures the exact distance between an entity and a point given by the user.

Usage

1. Specify the entity.
2. Use the mouse to specify the location of the point or enter a coordinate in the command line.
3. The measured distance is printed on screen and in the command line.

9.8 Angle (IA)

Toolbar / Icon:



Menu: `Info > Angle`
Shortcut: `I, A`
Commands: `infoangle | ia`

Description

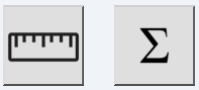
This tool measures the angle between two given lines.

Usage

1. Specify the first line.
2. Specify the second line.
3. The measured angle (in degrees) is printed on screen and in the command line.

9.9 Total Length of Selection (IS)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Info > Total Length of Selection`
Shortcut: `I, S`
Commands: `infosum | is`

Description

This tool calculates the total length of a selection of entities.

Usage

1. Select the entities you want to measure the total length from.
2. Start the tool.
3. The total length of the selected entities is printed in the command line.

9.10 Polygonal Area (IR)

Toolbar / Icon:



Menu: `Info > Polygonal Area`
Shortcut: `I, R`
Commands: `infoarea | ir`

Description

This tool measures the area covered by a polygon and its circumference.

Usage

1. Specify the first point of the area.
2. Specify the other points that define the area and right click when you are finished.
3. The measured area is printed on screen and in the command line. The circumference is also printed in the command line.

9.11 Arc/Circle/Ellipse Area (IC)

Toolbar / Icon:



Menu: `Info > Arc/Circle/Ellipse Area`
Shortcut: `I, C`
Commands: `acearea | ic`

Description

Author: Robert S.

This command calculates the area of arcs, circles, ellipses and elliptical arcs, and optionally adds the information to the current layer of a drawing. If you select any other type of entity a warning is shown in the command line.

For arcs, there are two options for calculating areas, namely Segment or Sector. The area of a segment (or slice) is the area bound by the arc and two lines drawn from the arc's startpoint and endpoint to the arc's centre. The area of a sector is the area bound by the arc and a chord drawn from the arc's startpoint to the endpoint.

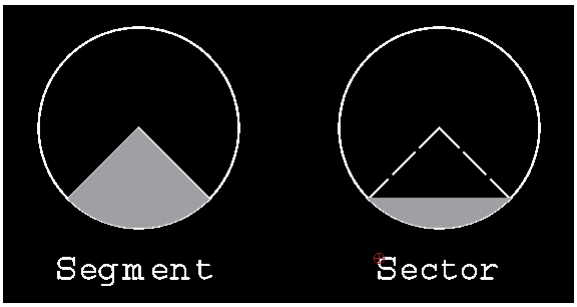


Figure 1: a segment (slice) and sector.

Options

This command has the standard "*Add measurement to drawing*" and "*Text Height*" options.

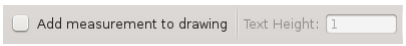


Figure 2: Standard Info options

Whether this is off or not, when you click on an arc, circle, ellipse or elliptical arc, the information is displayed in the command line widget.

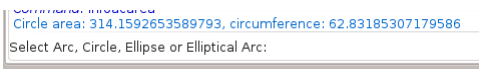


Figure 3: Info displayed in command line

(Note: Every time you start an info command the "*Add measurement to drawing*" checkbox is always unchecked. You must remember to switch it on if you want the info added to the drawing).

It also adds the following options.

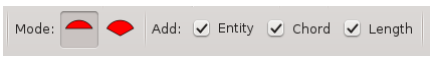


Figure 4: Added options

Mode:

The mode only applies to arcs.



Sector: This mode returns the area of the sector. This is useful when used in conjunction with the "*Polygonal Area*" command.



Segment: This mode returns the area of the segment.

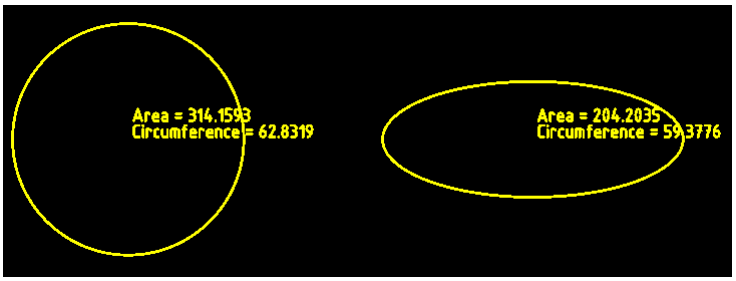
Add:

Entity: This selects whether you want to add a copy of the measured entity to the current layer of the drawing.

Chord: This selects if you want the chord (sector), or radius lines (segment), added to the drawing.

Length: By default the area is always printed on the drawing. This option selects whether you want extra information added to the drawing.

For circles and full ellipses the extra information is the circumference. This uses the centre point as the lower left corner for the added text.



For circular arcs the extra information is the length of the arc and the length of the chord (in sector mode), or the radius (in segment mode).

For ellipses the extra information is the length of the arc and the length of the chord (in sector mode), or the distance from the centre point to the startpoint of the arc, followed by the distance from the centre point to the endpoint of the arc (in segment mode).

The lower left corner for the added text is the endpoint of the arc.









9.12 Polyline Area (II)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Info > Polyline Area`
Shortcut: `I, I`
Commands: `infopolylinearea | ii`

Description

This tool measures the area of a polyline.

Usage

1. Specify the polyline entity.
2. The measured area is printed on screen and in the command line.

10 Layer

Toolbar / Icon:



Shortcut: `W, Y`
Commands: `layermenu`

Description

The tools in the Layer menu can be used to add, modify and delete layers. Some of the tools are also available as tool buttons at the top of the layer list.

10.1 Toggle Visibility (YV)

Toolbar / Icon:



Menu: `Layer > Toggle Visibility`
Shortcut: `Y, V`
Commands: `layertogglevisibility | yv`

Description

This tool toggles the visibility of the current layer. You may also toggle layers by clicking on the eye icon beside the layer name in the layer list. A black eye icon means that the layer is visible, a gray icon that the layer is hidden.

10.2 Toggle Frozen Status (YF)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Toggle Frozen Status`
Shortcut: `Y, F`
Commands: `layertogglefrozen | yf`

Description

This tool toggles the frozen status of the current layer. You may also toggle the frozen status of layers by clicking on the ice or sun icon beside the layer name in the layer list.

Note that block references on frozen layers are not shown while block references on layers that are off are still shown (provided that they contain any entities on layers which are on and not frozen).

10.3 Toggle Lock Status (YL)

Toolbar / Icon:



Menu: `Layer > Toggle Lock Status`
Shortcut: `Y, L`
Commands: `layertogglelock | yl`

Description

This tool toggles the locked status of the current layer. You may also toggle the locked status of layers by clicking on the lock icon beside the layer name in the layer list.

Entities on locked layers cannot be selected, deleted or modified.

10.4 Show Only Active (YO)

Toolbar / Icon:



Menu: `Layer > Show Only Active`
Shortcut: `Y, O`
Commands: `layershowactive | yo`

Description

This tool hides all layers except for the selected (active) layer in the layer list.

10.5 Show All Layers (YS)

Toolbar / Icon:



Menu: `Layer > Show All Layers`
Shortcut: `Y, S`
Commands: `layershowall | ys`

Description

This tool can be used to conveniently show all layers at once. This can be useful while working with a drawing with many layers.

10.6 Hide All Layers (YH)

Toolbar / Icon:



Menu: `Layer > Hide All Layers`
Shortcut: `Y, H`
Commands: `layerhideall | yh`

Description

This tool can be used to conveniently hide all layers at once. This can be useful while working with a drawing with many layers.

10.7 Thaw All Layers (YW)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Thaw All Layers`
Shortcut: `Y, W`
Commands: `layerthawall | yw`

Description

This tool can be used to conveniently thaw all layers at once. This can be useful while working with a drawing with many layers.

10.8 Freeze All Layers (YZ)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Freeze All Layers`
Shortcut: `Y, Z`
Commands: `layerfreezeall | yz`

Description

This tool can be used to conveniently hide (freeze) all layers at once. This can be useful while working with a drawing with many layers.

10.9 Lock All Layers (YK)

Toolbar / Icon:



Menu: `Layer > Lock All Layers`
Shortcut: `Y, K`
Commands: `layerlockall | yk`

Description

This tool can be used to conveniently lock all layers at once.

10.10 Unlock All Layers (YN)

Toolbar / Icon:



Menu: `Layer > Unlock All Layers`
Shortcut: `Y, N`
Commands: `layerunlockall | yn`

Description

This tool can be used to conveniently unlock all layers at once.

10.11 Add Layer... (YA)

Toolbar / Icon:



Menu: `Layer > Add Layer…`
Shortcut: `Y, A`
Commands: `layeradd | addlayer | ya`

Description

This command creates a new layer. The layer dialog is shown where you can specify the name for the new layer and adjust its attributes.

QCAD Professional

In addition, you can choose if objects on this layer can be snapped to and if this layer should be printed.
A layer can be made a sub-layer of a parent layer. The layer list will then show a tree structure of hierarchical layers.

10.12 Add Sublayer... (YU)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `Layer > Add Sublayer…`
Shortcut: `Y, U`
Commands: `layeraddsub | yu`

Description

Adds a sublayer beneath the current CAM layer.

10.13 Set Parent Layer... (YX)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Set Parent Layer…`
Shortcut: `Y, X`
Commands: `layersetparent | setparentlayer | yx`

Description

Assigns a parent layer to the selected layer, creating a layer hierarchy.

10.14 Create Layer from Selection... (YT)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Create Layer from Selection…`
Shortcut: `Y, T`
Commands: `layerfromselection | yt`

Description

Creates a new layer from the current selection and moves the selected entities onto it.

10.15 Move Selection to Current Layer (YJ)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Move Selection to Current Layer`
Shortcut: `Y, J`
Commands: `layertocurrent | yj`

Description

Moves the selected entities to the current layer.

10.16 Duplicate Layer... (YY)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Duplicate Layer…`
Shortcut: `Y, Y`
Commands: `layercopy | layerduplicate | yy`

Description

This tool creates a copy of the layer currently selected in the layer list. A dialog is shown that allows you to enter a name for the new layer.

10.17 Delete Layer(s) (YR)

Toolbar / Icon:



Menu: `Layer > Delete Layer(s)`
Shortcut: `Y, R`
Commands: `layerdelete | yr`

Description

You can delete the currently selected layer from the layer list with this command. Please note that all entities on that layer will be deleted as well.
Layer "0" can never be deleted.

10.18 Purge Unused Layers (YP)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Purge Unused Layers`
Shortcut: `Y, P`
Commands: `purgelayers`

Description

Removes all layers that are empty and unused from the drawing.

10.19 Edit Layer... (YE)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `Layer > Edit Layer…`
Shortcut: `Y, E`
Commands: `layeredit | ye`

Description

Edits the properties of the selected CAM layer, such as its association with a CAM configuration.

10.20 Activate Layer of Entity (YB)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Layer > Activate Layer of Entity`
Shortcut: `Y, B`
Commands: `layeractivatefromentity | yb`

Description

This tool activates the layer of the entity under the cursor. The shortcut [Y,B] can be used to trigger this tool both while idle as well as while constructing entities.

Usage

1. Move the mouse cursor over an entity on the desired layer.
2. Type the shortcut to trigger this tool.
3. The layer of the entity under the cursor is activated and becomes the current layer.

10.21 Select Layer Entities (YC)

Toolbar / Icon:



Menu: `Layer > Select Layer Entities`
Shortcut: `Y, C`
Commands: `layerselect | selectlayer | yc`

Description

This tool adds all entities on the current layer to the selection. Only entities on the current block can be selected, entities on other blocks are not selected.

10.22 Deselect Layer Entities (YD)

Toolbar / Icon:



Menu: `Layer > Deselect Layer Entities`
Shortcut: `Y, D`
Commands: `layerdeselect | deselectlayer | y- | yd`

Description

This tool deselects all entities on the current layer.

10.23 Layer States

Toolbar / Icon:



Description

Tools for saving and restoring layer states — named snapshots of the visibility, lock and other properties of all layers.

10.23.1 Add... (YM)

Add-on: QCAD Professional
Menu: `Layer > Layer States > Add...`
Shortcut: `Y, M`

Description

Saves the current layer configuration as a new, named layer state.

11 Block

Toolbar / Icon:



Shortcut: `W, B`
Commands: `blockmenu`

Description

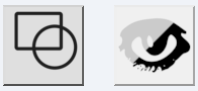
The block menu contains tools for creating and managing blocks, which are reusable groups of entities that can be inserted multiple times.

11.1 Explode (XP)

→ 7.7 Explode (XP) (114)

11.2 Toggle Visibility (BV)

Toolbar / Icon:



Menu: `Block > Toggle Visibility`
Shortcut: `B, V`
Commands: `blocktogglevisibility | bv`

Description

Click the eye icon in the block list to toggle the visibility of a block. Block references of invisible blocks are not displayed.

11.3 Show All Blocks (BS)

Toolbar / Icon:



Menu: `Block > Show All Blocks`
Shortcut: `B, S`
Commands: `blockshowAll | bs`

Description

This tool makes sure that all block references in the drawing are shown.

11.4 Hide All Blocks (BH)

Toolbar / Icon:



Menu: `Block > Hide All Blocks`
Shortcut: `B, H`
Commands: `blockhideAll`

Description

This tool hides all block references in the drawing.

11.5 Add Empty Block... (BA)

Toolbar / Icon:



Menu: `Block > Add Empty Block…`
Shortcut: `B, A`
Commands: `blockadd | ba`

Description

This command creates a new empty block. The block dialog is shown where you can specify a name for the new block. If you want to create a block from existing entities, please use the Create Block command instead.

11.6 Create Block from Selection (BC)

Toolbar / Icon:



Menu: `Block > Create Block from Selection`
Shortcut: `B, C`
Commands: `blockcreate | bc`

Description

Creates a new block from existing entities.

Usage

1. Select the entities you want to use for the new block.
2. Specify the reference point of the block.
3. Enter a name for the new block in the dialog that is shown and click OK.

The block is now added to the block list and instances of it can be inserted into the drawing. The entities you have selected in the first step are removed and replaced by a block reference of the newly created block.

11.7 Add Layout Block... (BL)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Add Layout Block…`
Shortcut: `B, L`
Commands: `layoutadd | bl`

Description

This tool creates a new layout block. Layout blocks are typically used to arrange different parts of a drawing on a sheet of paper for printing. Layout blocks typically contain a drawing border, header and one or multiple viewports.

Usage

- Layout blocks have a name (shown in the block list) and a tab order (order in the block list). Both can be entered when creating the layout block.

11.8 Remove Block (BR)

Toolbar / Icon:



Menu: `Block > Remove Block`
Shortcut: `B, R`
Commands: `blockremove | br`

Description

Removes the active block. The block, all entities on it and all block references of the block will be deleted.

QCAD Professional

Multiple blocks can be selected to delete all selected blocks at once.

11.9 Purge Unused Blocks (BP)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Purge Unused Blocks`
Shortcut: `B, P`
Commands: `purgeblocks`

Description

Removes all block definitions that are not used anywhere in the drawing.

11.10 Rename Block... (BN)

Toolbar / Icon:



Menu: `Block > Rename Block…`
Shortcut: `B, N`
Commands: `blockrename | bn`

Description

Renames the active block. The same dialog is presented as for creating new blocks. Block references are automatically updated to reference the renamed block.

QCAD Professional

If the currently selected block is a layout block, the layout can be renamed with this tool. In this case, the dialog also shows the order number of the layout. Layouts in the block list are ordered by this number.

11.11 Edit Block (BE)

Toolbar / Icon:



Menu: `Block > Edit Block`
Shortcut: `B, E`
Commands: `blockedit | be`

Description

This tool opens the active block for editing. It can be modified just like the main drawing. To return to the main drawing (called "*Model_Space"), activate block "*Model_Space" and click the same button again to edit that block or start the tool "Edit Main Drawing" in menu "Block".

11.12 Edit Block from Reference (BD)

Toolbar / Icon:



Menu: `Block > Edit Block from Reference`
Shortcut: `B, D`
Commands: `blockeditfromreference | bd`

Description

With this tool, a block can be edited by choosing an existing block reference entity. This is useful and usually faster if you don't know the name of the block you want to edit.

Usage

- Click any block reference entity that references the block you want to edit.

The block is opened for editing. To return to the main drawing (called "*Model_Space"), edit block "*Model_Space" or start the tool "Edit Main Drawing" in menu "Block".

11.13 Edit Block In-place (BB)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Edit Block In-place`
Shortcut: `B, B`
Commands: `blockeditinplace | bb`

Description

With this tool a block can be edited at the location where it was inserted into the drawing. The block can be inserted uniformly scaled or rotated.

This may appear as if the contents of a block reference was being edited. This is not the case. It's important to keep in mind that it is the block definition that is being edited. Any changes made to the block will affect not only the block reference that is being edited but also all other block references that are using that same block.

Usage

1. Click the block reference you want to edit in-place.
2. QCAD switches into block editing mode with the block entities as working set. The working set is the set of entities that can currently be modified. Various tool buttons are shown in the options toolbar. These tools are only shown in block editing mode when no other tool is active (neutral mode of QCAD). In this mode, the block entities (working set) are shown in their regular colors while the rest of the drawing is faded. This indicates that the faded entities can still be used as references for example for trimming or to create offsets but they cannot be modified as they are not in the current working set.

3. You can now change the block as desired using the normal CAD tools to draw, modify or delete entities.
4. When you are done, click "Save and Close" to exit the block editor and return to the main drawing.
5. Other tools in block editing mode are:
   - **Save and Update** to save the block and update all block references accordingly. This can be used to update the drawing and preview what effect the current modifications have. The block editing mode is not closed.
   - **Auto Update**to automatically update all other block references whenever the contents of the block is edited.
   - **Add to Working Set** to add selected entities from outside the current working set to the working set and add them to the block.
   - **Remove from Working Set** to remove the selected entities from the working set and from the block.
   - **Cancel** to cancel all changes since the last save and return to the main drawing.

Preferences

Edit > Application Preferences > Graphics View > Behavior

- *Edit block reference in-place:* Check this preference to open the in-place block editor when double-clicking a block. To exit the in-place block editor, you can double-click into an empty space of your drawing.

11.14 Duplicate Block... (BY)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Duplicate Block…`
Shortcut: `B, Y`
Commands: `blockcopy | blockduplicate | by`

Description

This tool creates a copy of the block currently selected in the block list. A dialog is shown that allows you to enter a name for the new block.

11.15 Return to Main Drawing (EM)

Toolbar / Icon:



Menu: `Block > Return to Main Drawing`
Shortcut: `E, M`
Commands: `editmain | em`

Description

Choose this tool when you are editing a block and want to return to the main drawing (block "*Model_Space").

11.16 Select Block References (BK)

Toolbar / Icon:

Menu: `Block > Select Block References`
Shortcut: `B, K`
Commands: `blockselect | selectblock | bk`

Description

This tool selects all block references of the block that is currently selected in the block list. Only block references which are part of the block that is currently being edited are selected.

11.17 Deselect Block References (BX)

Toolbar / Icon:

Menu: `Block > Deselect Block References`
Shortcut: `B, X`
Commands: `blockdeselect | deselectblock | bx`

Description

This tool deselects all block references of the block that is currently selected in the block list.

11.18 Insert Block (BI)

Toolbar / Icon:

Menu: `Block > Insert Block`
Shortcut: `B, I`
Commands: `blockinsert | minsert | insert | bi`

Description

This tool inserts the active block into the drawing. One or multiple block references are created in the drawing to represent the active block.

Usage

1. Choose the block you want to insert from the block list.
2. Click the insert button or choose 'Insert Block' from the menu.
3. Enter the rotation angle and scale factor for the block reference in the options tool bar.
4. Specify the position of the inserted block by clicking a coordinate or entering a coordinate in the console.
5. To create a whole array of blocks, click the array button in the options toolbar:

   Enter the columns, rows, column spacing and row spacing in the dialog that is shown. These properties can also be edited later using the property editor.

11.19 Insert External Reference... (XR)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Block > Insert External Reference...`
Shortcut: `X, R`
Commands: `externalreference | xref`

Description

This tool inserts an external reference into the active drawing. In QCAD, an external reference is a special type of block that is linked to an external file and automatically inherits its contents from that file. When the referenced file is modified, the external reference in the drawing can be updated to reflect those changes.

External reference blocks appear in the block list alongside regular blocks but are marked with a distinct icon indicating that the block links to an external file:



Because external references are controlled by their source files, they cannot be edited directly within the current drawing. To make changes, you must edit the external reference's source file.

If you share a drawing that contains external references, you must also include all referenced files. Please refer also to the notes on relative paths below.

Usage

1. Choose the file you want to insert as an external reference.
2. By default, the external reference block is named after the selected file. If needed, you can change this name in the options toolbar.
3. External references can be linked using either absolute or relative paths. Absolute paths often fail when the drawing and its referenced files are moved to another computer. Relative paths are typically more reliable, as references remain valid as long as the folder structure between the drawing and the external file is preserved.

For example, suppose your drawing is stored in "Data/Drawings" and your external references are located in "Data/External References". The relative path to a reference might be: "../External References/Item.dxf"

4. If the entire "Data" folder is moved to a different disk or computer, the reference will still be resolved correctly.
5. The options toolbar also provides controls to rotate, scale or flip the external reference during insertion. These transformations affect only the inserted reference in the drawing and do not modify the source file.
6. If the referenced file contains block attribute definitions, the options toolbar shows a list of these attributes. Choose an attribute from the list and enter its value, just like when inserting a regular block with attributes. The attributes are added to the inserted external reference and can be edited later using the property editor.
7. Specify the insertion point by clicking a location in the drawing or by entering coordinates in the command line.

11.20 Open External Reference File

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Open External Reference File`
Commands: `openexternalreference | openxref`

Description

This tool opens the file referenced by the currently selected external reference block in a new drawing tab for editing. After making your changes, save the file and return to the main drawing. QCAD will then prompt you to update the external references in your drawing to reflect the modifications.

11.21 Reload External References

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Reload External References`
Commands: `reloadxref`

Description

This tool forces a reload of all external references in the current drawing. Use it if QCAD previously detected changes in external references and prompted you to update them, but you chose No and now wish to apply those updates.

11.22 Bind External Reference

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Bind External Reference`
Commands: `bindexternalreference | bindxref`

Description

This tool converts the currently selected external reference block into a regular block stored in your drawing. This severs the link between your drawing and the file that was referenced by the external reference. You can use this if you want to make sure that your drawing no longer depends on an external file.

11.23 Create Library Item (BT)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Block > Create Library Item`
Shortcut: `B, T`
Commands: `librarycreate | bt`

Description

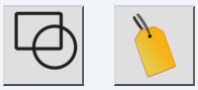
Creates a new part library item from a selection in the drawing.

Usage

1. Select the entities you want to use for the new library item.
2. Specify the reference point of the item.
3. Enter a name for the new item and choose a library and location where to store the item. Click OK.

11.24 Attributes

Toolbar / Icon:



Shortcut: `W, U`
Commands: `attributemenu`

Description

Tools for defining and editing block attributes (text fields attached to blocks).

11.24.1 Define Block Attribute (UA)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Block > Attributes > Define Block Attribute`
Shortcut: `U, A`
Commands: `attributedefinition | attdef | ua`

Description

Use this tool to add an attribute definition to a block definition.

Usage

1. The attribute definition dialog is shown.
2. Choose the font name, text height and text properties as described in the documentation of the text tool.
3. Enter a tag for the attribute definition.
   This identifier is used to link attributes to their attribute definition. You can also think of the tag as a variable name.
4. Enter a prompt for the attribute definition. This text is shown whenever the user is requested to enter a value for this attribute.
5. Enter a default value for attributes that are created from this attribute definition. This can be a commonly used value for the attribute or a placeholder such as "?" or "#".
6. Click "OK" to exit the dialog.
7. Use the mouse to specify the location of the attribute definition or enter a coordinate in the command line.
8. Right click or press Escape to stop the tool.

11.24.2 Synchronize Attributes (UY)

Toolbar / Icon:

Add-on: QCAD Professional
Menu: `Block > Attributes > Synchronize Attributes`
Shortcut: `U, Y`
Commands: `attributesynchronization | attsync | uy`

Description

This tool synchronizes attributes in block references with the attribute definitions in the appropriate block definition.
Attributes for which no definition is available are deleted.
For new attribute definitions, a new attribute is created with the attribute definition default value as value.
The value of existing attributes is not changed. The text style, position, alignment and angle are reset to those of the attribute definition.

If nothing is selected, this tool synchronizes all attributes of all block references.
If there is a selection, only those block references that are selected are synchronized with their block definitions.

12 Window

Description

The window menu contains tools for arranging and switching between open drawing windows.

12.1 Close All

Toolbar / Icon:



Menu: `Window > Close All`
Shortcut: `Ctrl+Shift+W (Mac: ⇧⌘W)`

Description

This tool closes all open drawings.

12.2 Next

Toolbar / Icon:



Menu: `Window > Next`
Shortcuts: `Ctrl+Right (Mac: ⌘Right) | Ctrl+PgUp (Mac: ⌘PgUp)`

Description

This tool activates the next open drawing window / tab.

12.3 Previous

Toolbar / Icon:



Menu: `Window > Previous`
Shortcuts: `Ctrl+Left (Mac: ⌘Left) | Ctrl+PgDown (Mac: ⌘PgDown)`

Description

This tool activates the previous open drawing window / tab.

12.4 Full Screen

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Window > Full Screen`
Shortcut: `Ctrl+Shift+F (Mac: ⇧⌘F)`
Commands: `fullscreen`

Description

Switches the application window to full screen mode or back to normal mode. The application menu can be accessed by clicking at the very top of the screen.

13 Misc

Toolbar / Icon:



Commands: `miscmenu`

Description

The Misc menu groups together additional, more specialized tools that are not part of the standard menus, organized into drawing, modification, information, selection, block and import / export sub-menus.

13.1 Import/Export

Toolbar / Icon:



Commands: `misciomenu`

Description

Contains additional import and export tools.

13.1.1 Import Points...

Menu: `Misc > Import/Export > Import Points…`

Description

Imports a list of point coordinates from a text or CSV file and creates point entities from them.

13.1.2 Import Points with Labels...

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Import/Export > Import Points with Labels...`

Description

Imports points together with text labels from a file.

13.1.3 Draw from CSV...

Toolbar / Icon:



Menu: `Misc > Import/Export > Draw from CSV…`
Commands: `drawcsv | dcsv`

Description

Reads geometry data from a CSV file and draws the corresponding entities in the current drawing.

13.1.4 Batch2Pdf visible layout blocks (2PF)

Toolbar / Icon:



Menu: `Misc > Import/Export > Batch2Pdf visible layout blocks`
Shortcut: `2, P, F`
Commands: `batch2pdf | b2pdf`

Description

Exports the visible layout blocks of the current drawing to PDF in a single batch operation.

13.1.5 KML Export...

Add-on: **QCAD Professional**
Menu: `Misc > Import/Export > KML Export…`
Commands: `kmlexport`

Description

Exports the drawing to KML for use in geographic applications such as Google Earth.

13.2 Select

Toolbar / Icon:



Commands: `miscselectmenu`

Description

Contains additional selection tools.

13.2.1 By Handle... (TH)

Toolbar / Icon:



Menu: `Misc > Select > By Handle…`
Shortcut: `T, H`

Description

Selects the entity with a given internal handle (ID).

13.2.2 By Color (TF)

Toolbar / Icon:



Menu: `Misc > Select > By Color`
Shortcut: `T, F`

Description

Selects all entities that use a given color.

13.2.3 Select Next Entity

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Select > Select Next Entity`
Shortcut: >

Description

Selects the next entity, based on entity order.

13.2.4 Select Previous Entity

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Select > Select Previous Entity`
Shortcut: <

Description

Selects the previous entity, based on entity order.

13.2.5 Store Selection (TS)

Add-on: QCAD Professional
Menu: `Misc > Select > Store Selection`
Shortcut: `T, S`
Commands: `storeselection | ts`

Description

Stores the current selection so it can be restored later.

13.2.6 Use Selection (TU)

Add-on: QCAD Professional
Menu: `Misc > Select > Use Selection`
Shortcut: `T, U`
Commands: `useselection`

Description

Restores a selection previously stored with the Store Selection tool.

13.3 Draw

Toolbar / Icon:



Commands: `miscdrawmenu`

Description

Contains additional, specialized drawing tools.

13.3.1 Boolean Operations (Legacy)

Toolbar / Icon:




Add-on: QCAD Professional
Menu: `Misc > Draw > Boolean Operations (Legacy)`
Commands: `polylinebooleanlegacy`

Description

Performs a boolean operation on selected closed polylines. This can be used to combine two polylines (union), to intersect them or to subtract a polyline from another (difference).

Usage

1. Select one or multiple closed polylines to unify or intersect. Alternatively, select the polylines from which you want to subtract another polyline

2. Start this tool.
3. Choose the desired boolean operation in the options toolbar:
   - Intersection (AND): The result is the intersection of all selected polylines. This is a polyline around the area that all polylines have in common.

     
   - Union (OR): The result is the union of all selected polylines. This is a polyline that contains all the selected polylines.

     
   - Difference (NOT): The result is the difference between the union of all selected polylines and the polylines chosen in the next step.

     
   - Exclusive or (XOR): The result is the set of polylines around the area covered by only one original polyline.

     
4. If you have chosen the difference mode, choose the polylines to subtract from the selection.
5. Choose if holes should be detected automatically. With this option, smaller contours inside other, larger contours are considered to be holes.
6. When working with large sets of polylines, switch off the automatic preview to avoid delays while the preview is being updated.
7. Click the green tick mark apply button in the options toolbar to apply the operation.
8. The selected polylines are still selected, so they can be easily deleted now if desired. The result is added to the current layer.

13.3.2 Horizontal Construction Line (XH)

Toolbar / Icon:



Menu: `Misc > Draw > Horizontal Construction Line`
Shortcut: `X, H`
Commands: `xlinehorizontal | xh`

Description

Draws an infinite horizontal construction line through a picked point.

13.3.3 Vertical Construction Line (XV)

Toolbar / Icon:



Menu: `Misc > Draw > Vertical Construction Line`
Shortcut: `X, V`
Commands: `xlinevertical | xv`

Description

Draws an infinite vertical construction line through a picked point.

13.3.4 Centerline Cross (CS)

Toolbar / Icon:



Menu: `Misc > Draw > Centerline Cross`
Shortcut: `C, S`
Commands: `cross | cs`

Description

Draws centerline crosses at the centers of selected arcs, circles or ellipses.

13.3.5 Counter (C12)

Toolbar / Icon:



Menu: `Misc > Draw > Counter`
Shortcut: `C, 1, 2`
Commands: `counter | c12`

Description

Places automatically incrementing numbers in the drawing, useful for numbering parts or positions.

13.3.6 Text Along Entity (TG)

Toolbar / Icon:



Menu: `Misc > Draw > Text Along Entity`
Shortcut: `T, G`
Commands: `textalong | tg`

Description

Use this tool to draw text along a line, arc or circle.

Usage

Please refer also the manual of the text function.

After closing the text dialog, choose the entity along which the text should be aligned. Adjust the options in the options toolbar if desired and click the position of the text.

13.3.7 Text Aligned Leader (2T)

Toolbar / Icon:



Menu: `Misc > Draw > Text Aligned Leader`
Shortcut: `2, T`
Commands: `textalignedleader | tal`

Description

Creates a leader whose arrow is aligned with a piece of text.

13.3.8 Leader from Selected Text (T2L)

Toolbar / Icon:



Menu: `Misc > Draw > Leader from Selected Text`
Shortcut: `T, 2, L`
Commands: `LeaderFromText | lftxt`

Description

Creates a leader (arrow) pointing from a selected text entity to a chosen location.

13.3.9 Box Joint from 2 Points (LJ)

Toolbar / Icon:



Menu: `Misc > Draw > Box Joint from 2 Points`
Shortcut: `L, J`
Commands: `boxjoint | bxj`

Description

Draws the fingers, slots and tenons of a box (finger) joint between two points, as used in woodworking.

13.3.10 Dovetail Joint from 2 Points

Toolbar / Icon:



Menu: `Misc > Draw > Dovetail Joint from 2 Points`
Commands: `dovetail`

Description

Draws a dovetail joint between two points, as used in woodworking.

13.3.11 Polyline break symbol from 2 points (S3)

Toolbar / Icon:



Menu: `Misc > Draw > Polyline break symbol from 2 points`
Shortcut: `S, 3`
Commands: `pbreaksymbol | pbrksym`

Description

Draws a polyline break symbol (a short zig-zag gap symbol) between two points to indicate that a part has been shortened or broken out.

13.3.12 Draw Fms Lanes (PF)

Toolbar / Icon:



Menu: `Misc > Draw > Draw Fms Lanes`
Shortcut: `P, F`
Commands: `fms_lanes | pf`

Description

Draws FMS lanes (ribs) used in flexible manufacturing system layouts.

13.3.13 Pole of Line and Circle

Toolbar / Icon:



Menu: `Misc > Draw > Pole of Line and Circle`

Description

Constructs the pole of a line with respect to a circle and marks it with a point.

13.3.14 Power Center of Three Circles

Toolbar / Icon:



Menu: `Misc > Draw > Power Center of Three Circles`

Description

Constructs the power center (radical center) of three circles and marks it with a point.

13.3.15 Radical Axis of 2 Circles

Toolbar / Icon:



Menu: `Misc > Draw > Radical Axis of 2 Circles`

Description

Constructs the radical axis of two circles.

13.3.16 Similarity Axes of 3 Circles

Toolbar / Icon:



Menu: `Misc > Draw > Similarity Axes of 3 Circles`

Description

Constructs the axes of similitude (similarity axes) of three circles.

13.3.17 Draw Bounding Box

Menu: `Misc > Draw > Draw Bounding Box`
Commands: `boundingbox`

Description

Draws a rectangle around the bounding box (extents) of the selected entities.

13.3.18 Polygon Hull

Add-on: **QCAD Professional**
Menu: `Misc > Draw > Polygon Hull`

Description

Creates a polygon that forms the hull (outline) around the selected entities.

13.4 Modify

Toolbar / Icon:



Commands: `miscmodifymenu`

Description

Contains additional, specialized modification tools.

13.4.1 Scale in X

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Modify > Scale in X`
Commands: `scalex`

Description

Scales the selection in the X direction only.

13.4.2 Scale in Y

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Modify > Scale in Y`
Commands: `scaley`

Description

Scales the selection in the Y direction only.

13.4.3 Scale One Way

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Modify > Scale One Way`
Commands: `scaleoneway`

Description

Scales the selection in a single, freely chosen direction.

13.4.4 Offset (Legacy)

Toolbar / Icon:

Menu: `Misc > Modify > Offset (Legacy)`

Description

With this tool you can create parallels to existing lines (or concentric arcs and circles).

Usage

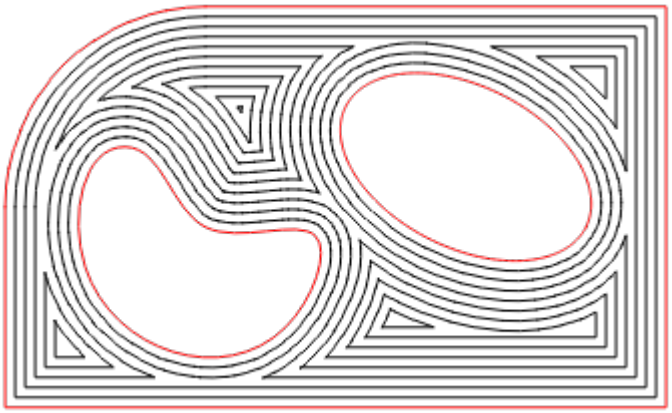
1. Enter the distance of the concentric or parallel entity from the original entity in the options tool bar shown at the top.
2. Enter the number of parallel or concentric entities to create in the options tool bar.
3. Click the base entity. The parallel or concentric entities are created on that side on which the mouse cursor is located while selecting the entity.

QCAD Professional

Additional options for the handling of polylines are shown in the options tool bar:

Mode

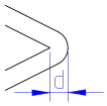
- 
  Polyline Mode: In this mode, the offset tool offsets entire polylines and other contours at once. If multiple polylines or contours are selected, they are all offset together as shown here:



- 
  Segment Mode: In this mode, individual segments of polylines can be offset.

Join Types

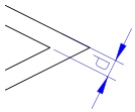
- Round:
  
  Edges are rounded with a radius that equals the offset distance.



- Miter:

  Edges are trimmed.

13.4.5 Polyline Offset (Alternative)

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Modify > Polyline Offset (Alternative)`

Description

Alternative polyline offset tool that creates a parallel (equidistant) copy of a polyline.

13.4.6 Offset (through Point, Legacy)

Toolbar / Icon:




Menu: `Misc > Modify > Offset (through Point, Legacy)`

Description

With this tool you can create parallels to existing lines (or concentric arcs and circles).

Usage

1. Enter the number of entities to create in the options tool bar.
2. Click the base entity.
3. Click the position through which the first parallel or concentric arc or circle runs.

QCAD Professional

The same additional options for the handling of polylines apply as for the Offset (with Distance) tool.

13.4.7 Set Attributes by Layer

Add-on: QCAD Professional
Menu: `Misc > Modify > Set Attributes by Layer`

Description

Sets the attributes (color, linetype and line weight) of the selected entities to 'By Layer'.

13.4.8 Set Attributes by Block

Add-on: QCAD Professional
Menu: `Misc > Modify > Set Attributes by Block`

Description

Sets the attributes (color, linetype and line weight) of the selected entities to 'By Block'.

13.4.9 Purge Unused Layers and Blocks

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Modify > Purge Unused Layers and Blocks`
Commands: `purgelayersandblocks`

Description

Removes all unused layers and block definitions from the drawing in a single step.

13.4.10 Flatten Drawing to 2D

Add-on: QCAD Professional
Menu: `Misc > Modify > Flatten Drawing to 2D`
Commands: `flatten`

Description

Flattens the drawing to 2D by setting the Z coordinate of all geometry to zero.

13.4.11 Reload Linetype Definitions

Menu: `Misc > Modify > Reload Linetype Definitions`

Description

Reloads the linetype (line pattern) definitions from disk, applying any changes made to the linetype files.

13.4.12 Flush Undo History

Menu: `Misc > Modify > Flush Undo History`

Description

Clears the undo / redo history of the current drawing to free memory.

13.4.13 Set Font to Arial

Menu: `Misc > Modify > Set Font to Arial`

Description

Changes the font of all text and dimension entities in the drawing to Arial.

13.4.14 Regenerate Hatch Patterns

Add-on: QCAD Professional
Menu: `Misc > Modify > Regenerate Hatch Patterns`
Commands: `hatchregen`

Description

Regenerates the pattern of existing hatch entities, for example after a pattern definition has changed.

13.4.15 Convert Splines to Bézier Segments

Add-on: QCAD Professional
Menu: `Misc > Modify > Convert Splines to Bézier Segments`
Commands: `splitsplines`

Description

Converts splines into a sequence of Bezier curve segments.

13.4.16 Convert Straight Splines to Lines

Menu: `Misc > Modify > Convert Straight Splines to Lines`

Description

Converts splines that are effectively straight into simple line entities.

13.4.17 Reset Splines Tangents

Add-on: QCAD Professional
Menu: `Misc > Modify > Reset Splines Tangents`
Commands: `resetsplinetangents`

Description

Resets the end tangents of the selected splines.

13.4.18 Split polyline at arc / line segment change

Add-on: QCAD Professional
Menu: `Misc > Modify > Split polyline at arc / line segment change`

Description

Splits polylines at every point where the segment type changes between a line and an arc.

13.5 Projection

Toolbar / Icon:



Commands: `miscprojectionmenu`

Description

Contains additional projection tools.

13.5.1 Map Projection

Add-on: QCAD Professional
Menu: `Misc > Projection > Map Projection`

Description

Reprojects geographic geometry from one map projection (coordinate reference system) to another.

13.6 Layer

Toolbar / Icon:



Commands: `misclayermenu`

Description

Contains additional layer tools.

13.6.1 Make all White Layers Black

Add-on: QCAD Professional
Menu: `Misc > Layer > Make all White Layers Black`

Description

Changes the color of all white layers to black.

13.6.2 Make all Black Layers White

Add-on: QCAD Professional
Menu: `Misc > Layer > Make all Black Layers White`

Description

Changes the color of all black layers to white.

13.6.3 Create New Layer from Hatches

Add-on: QCAD Professional
Menu: `Misc > Layer > Create New Layer from Hatches`

Description

Moves all hatch entities onto a new, separate layer.

13.7 Block

Toolbar / Icon:



Commands: `miscblockmenu`

Description

Contains additional tools for working with blocks.

13.7.1 Fix Block Names

Menu: `Misc > Block > Fix Block Names`

Description

Corrects invalid or duplicate block names in the drawing.

13.7.2 Export Block List...

Menu: `Misc > Block > Export Block List…`

Description

Exports the list of block definitions in the drawing to a file.

13.7.3 Export Block References List...

Menu: `Misc > Block > Export Block References List…`

Description

Exports a list of all block references (inserts) in the drawing to a file.

13.7.4 List Block Attributes

Menu: `Misc > Block > List Block Attributes`

Description

Lists the attributes of the block references (inserts) in the drawing.

13.7.5 Prefix Block Names...

Add-on: QCAD Professional
Menu: `Misc > Block > Prefix Block Names...`

Description

Adds a common prefix to the names of all blocks in the drawing.

13.8 Information

Toolbar / Icon:



Commands: `miscinfomenu`

Description

Contains additional tools for querying information from the drawing.

13.8.1 Store Positions (TT)

Add-on: QCAD Professional
Menu: `Misc > Information > Store Positions`
Shortcut: `T, T`
Commands: `storepos | tt`

Description

Stores the positions of picked reference points so they can be reused later.

13.8.2 Use Positions (TY)

Add-on: QCAD Professional
Menu: `Misc > Information > Use Positions`
Shortcut: `T, Y`
Commands: `usepos | ty`

Description

Recalls positions previously stored with the Store Positions tool.

13.8.3 Bearing / Azimuth

Menu: `Misc > Information > Bearing / Azimuth`

Description

Measures the bearing (azimuth) of the direction between two picked points.

13.9 Development

Toolbar / Icon:



Commands: `miscdevelopmentmenu`

Description

Contains tools intended for script development and debugging.

13.9.1 Script Shell (GE)

Toolbar / Icon:



Menu: `Misc > Development > Script Shell`
Shortcut: `G, E`
Commands: `ge`

Description

Opens an interactive ECMAScript (JavaScript) shell for entering and running script commands directly inside QCAD.

13.9.2 Run Script (XC)

Toolbar / Icon:



Menu: `Misc > Development > Run Script`
Shortcut: `X, C`
Commands: `runscript | rscript | xscript | rsc | xsc`

Description

Opens a script file from disk and runs it inside QCAD.

13.10 Reset Configuration

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Reset Configuration`
Commands: `resetconfig`

Description

Resets the application configuration (settings) back to its defaults.

13.11 Manage RCC Plugins

Toolbar / Icon:



Add-on: QCAD Professional
Menu: `Misc > Manage RCC Plugins`

Description

Manages RCC (Qt resource) plugins, allowing add-on resource packs to be enabled or disabled.

14 CAM

Toolbar / Icon:



Commands: `cammenu`

Description

The CAM menu contains tools for computer-aided manufacturing, such as defining toolpaths and exporting G-code for CNC machines.

14.1 CAM Configuration... (KK)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > CAM Configuration...`
Shortcut: `K, K`
Commands: `camconfig | kk`

Description

Use this function to display a dialog with various preferences for the current configuration. These preferences apply to all tools and all toolpaths.

14.2 CAM Export (KX)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > CAM Export`
Shortcut: `K, X`
Commands: `camexport | kx`

Description

This tool exports all available toolpaths in the current drawing to a CAM output file using the currently selected configuration (combo box at the top left).
If the drawing has been previously exported using the same configuration, the export starts immediately. Otherwise, a file dialog is shown to choose a file name and location.

14.3 G-Code Import...

Toolbar / Icon:

Add-on: QCAD/CAM
Menu: `CAM > G-Code Import...`
Commands: `camimport`

Description

This tool imports the geometry from an existing G-code file. The expected syntax of the G-code can be configured in the application preferences.

14.4 Display Toolpaths (KV)

Toolbar / Icon:

Add-on: QCAD/CAM
Menu: `CAM > Display Toolpaths`
Shortcut: `K, V`
Commands: `togglecam | kv`

Description

This tool toggles the display of all CAM toolpaths. If toolpaths are on, all toolpaths are shown in the drawing area. This can be distracting when changing the geometry of your drawing.

14.5 Add Tool (KA)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Add Tool`
Shortcut: `K, A`
Commands: `camaddtool | addtool | ka`

Description

This function adds a tool for CAM processing to the current drawing. The new tool can then be used to create CAM toolpaths. The main property of a tool is its diameter. The diameter is used to compute the desired toolpath offset (cutter radius compensation).

Usage

1. Start this tool.

2. A dialog for entering the tool parameters is displayed.
   - The tool name is usually a number that is used in the G-code output for tool changes (e.g. "T1").
   - The tool diameter is required for the calculation of the tool radius compensation and for the simulation.

14.6 Edit Tool (KE)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Edit Tool`
Shortcut: `K, E`
Commands: `camedittool | edittool | ke`

Description

This function can be used to edit an existing tool. The same dialog with tool parameters is shown as when a new tool is added.

14.7 Remove Tool (KR)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Remove Tool`
Shortcut: `K, R`
Commands: `camdeletetool | tooldelete | kr`

Description

This function removes the selected tool from the tool list.

14.8 Export Tools (KO)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Export Tools`
Shortcut: `K, O`
Commands: `camexporttools | exporttools | ko`

Description

This function exports all tools of the selected configuration of the current drawing into a separate file. The tools can then for example be imported into another, existing drawing.

14.9 Import Tools (KI)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Import Tools`
Shortcut: `K, I`
Commands: `camimporttools | importtools | ki`

Description

This function imports tools from an external file. This file may have been created previously via CAM < Export Tools or can be any other drawing with tools.

14.10 Add Profile Toolpath (KP)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Add Profile Toolpath`
Shortcut: `K, P`
Commands: `camaddprofiletoolpath | addprofiletoolpath | profiletoolpathadd | kp`

Description

This tool creates a new profile tool path, based on the current selection. A profile tool path is tool path that follows a profile with the tool center either on the geometry or using a tool radius correction to the left or right side of the geometry.

Usage

1. Select the profile(s) you want to use as a basis for the profile toolpath. You can select one or multiple open or closed profiles. A profile typically consists of a polyline or a selection of connected lines, arcs, ellipse arcs and splines. If you need full control of the start point and direction of a profile toolpath, make sure that all your profiles consist of polylines with the desired start point and direction.
2. Start this tool.
3. The profile toolpath dialog is shown with the various parameters that define a toolpath.
   - Name: Name of the toolpath, unique per configuration.
   - Tool: The tool number of the tool used to process the toolpath. You can also add a tool by clicking on the plus symbol.

- Feed Rate: The feed rate to use for this particular tool path. This defaults to the feed rate of the chosen tool but can be adjusted for each toolpath individually if desired.
- Plunge Rate: The feed rate used for downward moves. This defaults to the plunge rate of the chosen tool but can be adjusted for each toolpath individually if desired. Cutting Depths:
- Safe Z (a): The Z level used to approach the toolpath at rapid move (clearance, approaching level).
- Start Depth (b): The Z level at which the material begins (typically 0).
- Cut Depth (c): The final cutting depth, measured from the start depth. Note that cut depth is a distance (positive), not a Z level.
- Passes: The number of passes to use to reach the final cutting depth or 1 to cut to the final depth in a single pass. Side / Direction:
- Side: The side for the tool radius compensation or *On* to cut with the tool center on the profile.
- Direction: Cutting direction climbing (left of profile) or conventional (right of profile). Leads:
- Lead In Type: Shape of lead in to use. Leads are an additional moves inserted before / after the profile to approach or leave the profile. If tool radius compensation is used, leads must be used, so the tool radius compensation can be switched on / off.
- Lead In Type: Shape of lead out to use.
- Lead In Length / Radius: The size of the lead in shape to use.
- Lead Out Length / Radius: The size of the lead out shape to use.
- Overcut: Overlapping element used at the end of a profile to cut slightly further than required. This can lead to a smoother end result.

4. Click OK to create the profile toolpath. QCAD/CAM generates the toolpath and displays the path of the tool center in the graphics view as a dashed line.

14.11 Add Pocket Toolpath (KC)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Add Pocket Toolpath`
Shortcut: `K, C`
Commands: `camaddpockettoolpath | addpockettoolpath | pockettoolpathadd | kc`

Description

This tool creates a new pocket toolpath, based on the current selection. A pocket toolpath clears the area inside closed contours. The tool is moved along offset paths which are generated from the contour, starting in the center of the pocket and moving outwards. The last pass follows the contour at the distance of the tool radius (finishing pass).

Closed contours inside the selected pocket contours are treated as islands and are not cut. Contours inside islands are treated as pockets again (nested islands and pockets).

Usage

1. Select the closed contour(s) of the pocket(s) you want to clear. A contour typically consists of a closed polyline, a circle or a selection of connected lines, arcs, ellipse arcs and splines. Selected contours that are located inside another selected contour are treated as islands and are excluded from the pocket. Contours inside islands are pockets again.
2. Start this tool.
3. The pocket toolpath dialog is shown with the various parameters that define a toolpath.
   - Name: Name of the toolpath, unique per configuration.
   - Tool: The tool number of the tool used to process the toolpath. You can also add a tool by clicking on the plus symbol.
   - Feed Rate: The feed rate to use for this particular tool path. This defaults to the feed rate of the chosen tool but can be adjusted for each toolpath individually if desired.
   - Plunge Rate: The feed rate used for downward moves. This defaults to the plunge rate of the chosen tool but can be adjusted for each toolpath individually if desired. Cutting Depths:
   - Safe Z (a): The Z level used to approach the toolpath at rapid move (clearance, approaching level).
   - Start Depth (b): The Z level at which the material begins (typically 0).
   - Cut Depth (c): The final cutting depth, measured from the start depth. Note that cut depth is a distance (positive), not a Z level.
   - Passes: The number of passes to use to reach the final cutting depth or 1 to cut to the final depth in a single pass. Pocket:
   - Direction: Cutting direction climbing or conventional.
   - Max. Stepover: The maximum distance between two neighbouring passes of the pocket toolpath. This defaults to half the tool diameter. The actual stepover never exceeds this value and the pocket is always cleared completely.
   - Max. Finishing Stepover: The maximum distance between the last pass along the contour (finishing pass) and the pass before it. A small value leaves little material for the finishing pass which results in a smoother surface of the pocket walls.
4. Click OK to create the pocket toolpath. QCAD/CAM generates the toolpath and displays the path of the tool center in the graphics view. The pocket is cleared from inside out. Where possible, the tool moves from one pass to the next pass without retreating to the safe Z level.

14.12 Add Drill Toolpath (KL)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Add Drill Toolpath`
Shortcut: `K, L`
Commands: `camadddrilltoolpath | adddrilltoolpath | drilltoolpathadd | kl`

Description

This tool creates a new drill tool path, based on the current selection. A drill tool path is a tool path for drilled holes. The center of the hole can be defined by a point entity or the center of a circle entity.

Usage

1. Select the points or circles you want to use as a basis for the drill toolpath. You can select one or multiple points or circles.
2. Start this tool.
3. The drill toolpath dialog is shown with the various parameters that define a drill toolpath.
   - Name: Name of the toolpath, unique per configuration.
   - Tool: The tool number of the tool used to process the toolpath. You can also add a tool by clicking on the plus symbol.
   - Plunge Rate: The feed rate used for downward moves. This defaults to the plunge rate of the chosen tool but can be adjusted for each toolpath individually if desired. For upward moves, the feed rate of the tool is used. Cutting Depths:
   - Safe Z (a): The Z level used to approach the toolpath at rapid move (clearance, approaching level).
   - Start Depth (b): The Z level at which the material begins (typically 0).
   - Cut Depth (c): The final cutting depth, measured from the start depth. Note that cut depth is a distance (positive), not a Z level.
   - Passes: The number of passes to use to reach the final cutting depth or 1 to cut to the final depth in a single pass.
4. Click OK to create the drill toolpath. QCAD/CAM generates the toolpath and displays the path of the tool center in the graphics view as a dashed line.

14.13 Edit Toolpath (KD)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Edit Toolpath`
Shortcut: `K, D`
Commands: `camedittoolpath | edittoolpath | toolpathedit | kd`

Description

Use this function to edit the parameters of existing toolpaths.

14.14 Remove Toolpath (KM)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Remove Toolpath`
Shortcut: `K, M`
Commands: `camremoveprofiletoolpath | removeprofiletoolpath | toolpathremove | km`

Description

This tool removes the currently selected toolpath from the toolpath list.

14.15 Move Toolpath Up

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Move Toolpath Up`
Commands: `toolpathmoveup`

Description

This tool moves a toolpath further up in the order of processing. Toolpaths are exported in the exact order as shown in the toolpath list.

14.16 Move Toolpath Down

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Move Toolpath Down`
Commands: `toolpathmovedown`

Description

This tool moves a toolpath further down in the order of processing. Toolpaths are exported in the exact order as shown in the toolpath list.

14.17 Regenerate Toolpath

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Regenerate Toolpath`
Commands: `toolpathregen`

Description

This tool regenerates the toolpath that is currently selected in the toolpath list based on its current parameters. Use this after changing the parameters or geometry of a toolpath.

14.18 Regenerate Toolpath from Tool

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Regenerate Toolpath from Tool`
Commands: `toolpathregentool`

Description

Regenerates the selected toolpath, updating tool parameters from the tool associated with the toolpath, updating the toolpath to use the parameters of that tool if they have changed.

14.19 Regenerate All Toolpaths

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Regenerate All Toolpaths`
Commands: `toolpathregenall`

Description

This tool regenerates all toolpaths based on their current parameters. Use this after changing configuration global parameters or after updating the geometry of your drawing.

14.20 Update Toolpath from Selection

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Update Toolpath from Selection`
Commands: `toolpathupdatefromselection`

Description

This tool updates the currently selected toolpath to use the selected geometry as a new basis.
Use this after deleting (part of) the geometry of a toolpath to regenerate the toolpath based on a different or updated geometry.

14.21 Add Tab (KT)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Add Tab`
Shortcut: `K, T`
Commands: `camaddtab | kt`

Description

This function adds a tab to the current drawing. Tabs are small connections between a cutout shape and its surrounding material. Tabs can be used to hold a piece in place while cutting it out of a sheet material. This prevents the piece from moving while the tool is still cutting.
Tabs are typically very small, so the piece can be easily broken out of the sheet material once the toolpath is complete. To securely hold a piece in place, three or more tabs are typically used.

Usage

1. Start this tool.
2. Enter the desired length and thickness of the tab.
3. Use the mouse to position the tab in your drawing. Tabs are usually positioned on the original contour (not on the offset contour). However, they also affect the toolpath if a radius compensation is used.

14.22 Add Workpiece (KW)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Add Workpiece`
Shortcut: `K, W`
Commands: `camaddworkpiece | kw`

Description

This function adds a workpiece to the current drawing. A workpiece represents the stock material (for now a cuboid) from which the part is machined. Workpieces are 3D mesh entities (see Draw > 3D) flagged as workpiece (property "Workpiece" in the property editor), so CAM tools can tell them apart from other 3D bodies.
In 2D views, a workpiece is shown as its outline (top view). In the 3D view it is shown as solid body.

Usage

1. Start this tool.
2. Enter the thickness of the workpiece (its size in Z) and the Z coordinate of its top face. By default the top face lies in the drawing plane (Z=0) and the material extends downwards, as the toolpaths cut into the material from Z=0 downwards.
3. Click the first corner of the workpiece in your drawing.
4. Click the opposite corner of the workpiece. Hold down the Shift key to create a square footprint.

14.23 Show CAM Tool List (GO)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Show CAM Tool List`
Shortcut: `G, O`
Commands: `camtoollist | go`

Description

The tool list shows all tool that have been created for the current drawing and chosen configuration. You can create as many tools as required, for each configuration. When changing the configuration at the top left, the tool list is automatically updated to only show the tools available for that configuration.

14.24 Show CAM Toolpath List (GA)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Show CAM Toolpath List`
Shortcut: `G, A`
Commands: `camtoolpathlist | ga`

Description

The toolpath list shows all toolpath that have been created for the current drawing and chosen configuration. Toolpaths are shown in the same order as they will be processed in.

14.25 Show Simulation Panel (GI)

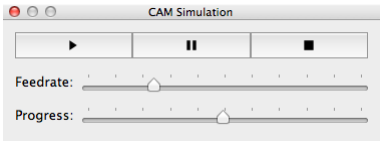
Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Show Simulation Panel`
Shortcut: `G, I`
Commands: `camsimulation | gi`

Description

With the CAM Simulation widget, the order in which entities are processed can be visualized in slow motion.



The play button starts the simulation. The current position of the tool is shown as yellow circle in the drawing area.
The pause button can be used to pause the simulation. Hit the play button to continue the simulation again.
The stop button stops the simulation.
The feedrate slider can be used to adjust the speed of the simulation while the simulation is in progress.
With the progress slider, the progress of the simulation can be adjusted directly to the desired position and continued from there.

14.26 Nesting (NE)

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Nesting`
Shortcut: `N, E`
Commands: `nesting`

Description

This tool automatically arranges parts on one or multiple chosen sheets optimizing the space used on the sheet.

Usage

1. Optional: Define the nesting quantity of the parts to nest. To do this, select all entities of a part, choose *CAM* > *Set Nesting Quantity* and enter the desired quantity.
2. Select the parts to nest. Parts can consist of lines, arcs, circles, ellipses, splines or polylines. Parts may contain islands. Each part is expected to be surrounded by a closed path of connected entities.
3. Start this tool.
4. Adjust the nesting parameters in the options toolbar at the top if desired:
   - Rotations: The number of rotations to try when inserting parts into the sheets. 4 means that each part can be rotated 0 degrees, 90 degrees, 180 degrees or 270 degrees.
   - Sheet Margin: Minimum distance between sheets and parts
   - Margin: Minimum distance between parts
   - Tolerance: Tolerance when converting arcs to polylines. A larger tolerance will not lead to overlapping parts but might lead to larger margins between parts.
   - Alignment: Parts are arranged with the chosen alignment in mind. An alignment of "Left - Bottom" means that parts are shuffled to the left (primary alignment) and towards the bottom (secondary alignment).
5. Click the sheet to be used to arrange the parts in. This can be any entity that is part of a closed shape. Multiple sheets may be chosen. Only one entity of each sheet has to be clicked. Multiple sheets can be chosen. If a sheet lies within another sheet, it is considered to be an island. Islands are not used to place parts.
6. Right-click when all sheets have been chosen.
7. QCAD/CAM starts the nesting process and lists possible solutions as they are found.
8. As soon as the first solutions are found and listed, you can click a solution in the list to preview it. Solutions are listed in order of quality (from best to worst).
9. The nesting process can be stopped when a good solution is identified using the "Stop" button.
10. QCAD/CAM now shows all solutions again and lets you choose the preferred solution.
11. Click "Apply Solution" to apply the chosen solution.

Preferences

Edit > Application Preferences > CAM > Nesting

- *Number of threads:* Number of CPU threads to use for nesting. Default is "Auto" which means the number of threads depends on the number of CPU cores available on the system.
- *Maximum number of iterations:* Number of iterations for which threads are created to compute new, better solutions. This can be set to a lower number to cause QCAD/CAM to stop even when the user never clicks the "Stop" button.
- *Weighting of primary alignment:* The factor by how much the primary alignment is weighted more than the secondary alignment. Set to 1 to weigh both alignments equally. Set to a higher number to squeeze in one direction only.
- *Place small parts into holes of other parts:* If activated, allows placing of smaller parts inside holes of larger parts.
- Align part angles to axis: Automatically rotate parts to be aligned to X / Y axis.
- *Disable AppNap if QCAD runs in background mode:* Keep QCAD/CAM threads active at normal priority when QCAD/CAM runs in the background (i.e. QCAD/CAM is not the top window). This preference is only available on macOS.

14.27 Set Nesting Quantity

Toolbar / Icon:



Add-on: QCAD/CAM
Menu: `CAM > Set Nesting Quantity`
Commands: `nestingquantity`

Description

This tool allows you to assign a quantity to a part to be nested. The part is then inserted several times into the sheet during nesting.

15 Command Line Tools

Description

Command-line tools are standalone scripts run from a terminal instead of from the QCAD graphical interface. They use the same drawing engine as QCAD to process drawing files in batch, without opening a window, which makes them well suited for automation.

15.1 merge

Description

Command-line tool that merges several drawing files into a single drawing.

Usage

```
Usage: ./merge [OPTIONS]... <XML config file>

Merges multiple drawing files (DWG, DXF, ...) into one.

  -f, -force              Overwrite existing output file
  -h, -help               Display this help
  -t TYPE, -type=TYPE     Set file type to TYPE
                          (e.g. R12...R27 or 2000...2013)
  -o FILE, -outfile=FILE  Set output file to FILE
                          default is same path and base name as input file

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.

XML format:
<?xml version="1.0" encoding="UTF-8"?>
<merge xmlns="http://qcad.org/merge/elements/1.0/" unit="Millimeter">
    <item src="file1.dxf">
        <insert x="10" y="0" angle="30" flipx="false" flipy="false" />
    </item>
</merge>
```

15.2 bmp2dwg

Description

Command-line tool that traces a raster bitmap image into vector geometry and saves it as a DWG/DXF drawing.

Usage

```
Usage: ./bmp2dwg [OPTIONS]... <bitmap file>

Traces a bitmap file (BMP, PNG, ...) and saves the result in a drawing file (DWG, DXF, ...).

  -a, -alpha=ALPHA        Set alpha threshold to ALPHA,
                          treat higher transparencies as fully transparent [0-255]
                          (default 200)
  -b, -blurdelta=DELTA    Set blur delta to DELTA (default 200)
  -c, -colors=COLORS      Set number of colors to COLORS (default 16)
  -f, -force              Overwrite existing output file
  -h, -help               Display this help
  -m, -monochrome         Convert bitmap to monochrome before tracing
  -o, -outfile=FILE       Set output file to FILE
                          Default is same path and base name as input file
  -r, -release=RELEASE    Set output DXF/DWG release to RELEASE
  -t, -brightness=B       Set brightness threshold to B,
                          treat brighter pixels as white [0-255] (default 200)

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.3 bbox

Description

Command-line tool that reports the bounding box (extents) of a DWG/DXF drawing.

Usage

```
Usage: ./bbox <drawing file>

Prints the minimum, maximum and size of the given drawing.

  -h, -help               Display this help

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:
```

```
  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.4 dwg2bmp

Description

Command-line tool that renders a DWG/DXF drawing to a raster bitmap image.

Usage

```
Usage: ./dwg2bmp [OPTIONS]... <drawing file>

Converts a drawing file (DWG, DXF, ...) to a bitmap file (BMP, PNG, ...).

  -a, -antialiasing       Enable antialiasing
  -b, -background=C       Set the background color as color name
                          e.g. 'white' or '#ccdd00'.
      -block=BLOCK_NAME   Name of block (or layout) to output
                          (defaults to *Model_Space)
  -c, -color-correction   Enable color correction to prevent white on white
                          or black on black painting
  -d, -recompute-dim      Recompute dimension blocks if empty
                          (dimensions rendered with best compatibility)
  -f, -force              Overwrite existing output file
      -flat               Flatten to 2D (Z==0)
  -fs, -font-substitution FONT1 FONT2
                          Substitute FONT1 with FONT2
  -h, -help               Display this help
      -no-weight-margin   Don't add additional margin to compensate for
                          lineweight
  -k, -grayscale          Grayscale
      -layer=LAYER_NAMES  Comma-separated list of layer(s) to export
                          List may include regular expressions
  -l, -list               List supported formats
      -ltscale=FACTOR     Set linetype scale to FACTOR
  -m, -margin=M           Set auto zoom margin to M (in pixels)
  -n, -monochrome         Monochrome (black / white)
  -o, -outfile=FILE       Set output file to FILE
                          Default is same path and base name as input file
  -q, -quality=Q          Set the image quality / compression
                          (0=best compression..100=best quality)
  -r, -resolution=R       Set rendering resolution to R
                          (in pixels/drawing unit)
                          This overrides the width/height parameters.
  -u, -unit=[in|m|mm]     Assume input data to be in given unit
                          Default is the drawing unit
                          Mainly useful for drawings with unit 'None'
  -w, -window=X,Y,W,H     Zoom to given window
  -x, -width=W            Set bitmap width to W (in pixels)
  -y, -height=H           Set bitmap height to H (in pixels)
                          The maximum size for width*height is 536870911 pixels
      -zoom-all           Auto zoom to maximum extends of all layers
  -p, -origin             Draw zero point (origin)
```

```
macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.5 dwg2csv

Description

Command-line tool that exports data from a DWG/DXF drawing to a CSV file.

Usage

```
Usage: ./dwg2csv [OPTIONS]... <drawing file>

Extracts information from a drawing file (DWG, DXF, ...) to CSV.

  -d, -decimals           Number of decimal places for float value
  -f, -force              Overwrite existing output file
  -l, -list               List available entity types and their properties
      -layer=LAYER_NAMES  Comma-separated list of layer(s) to export
                          Layer names may be regular expressions
                          Entities on other layers are not exported
                          Does not affect non-entity objects (layers, blocks, ...)
  -o FILE, -outfile=FILE  Set output file to FILE
                          default is same path and base name as input file
  -p PROPERTY             Extract given PROPERTY (e.g. 'Start Point:X')
                          Multiple properties may be extracted
                          Defaults to all properties of the given object type(s)
  -t OBJECT_TYPE          Object or entity type (e.g. 'Line', 'Layer', ...)
                          Defaults to all object type(s)

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.6 dwg2dwg

Description

Command-line tool that converts a DWG/DXF drawing to another DWG/DXF version or format.

Usage

```
Usage: ./dwg2dwg [OPTIONS]... <drawing file>

Converts a drawing file (DWG, DXF) or PDF file
to a drawing file (DWG, DXF) with a specified format release.

  -f, -force              Overwrite existing output file
  -h, -help               Display this help
  -l, -list               List supported releases
  -n, -setdimfont=FONT    Set the dimension font to the given font
  -o, -outfile=FILE       Set output file to FILE
                          Default is same path and base name as input file
  -r, -release=RELEASE    Set output DXF/DWG release to RELEASE
  -s, -set=VARIABLE:VALUE Set the given known variable to the given value
                          For example -s DIMASZ:2.5
                          DXF header variables that can be changed are
                          (among others):
                          DIMADEC:  Angular dimension label precision (0-8)
                          DIMASZ:   Dimension arrow size
                          DIMAUNIT: Angular dimension label format
                                    0: decimal degrees
                                    1: deg/min/sec
                                    2: gradians
                                    3: radians
                                    4: surveyors
                          DIMAZIN:  Trailing zeroes for angular dimensions
                                    (0: on, 2: off)
                          DIMDEC:   Set linear dimension label precision (0-8)
                          DIMDLI:   Dimension line increment
                          DIMEXE:   Extension line extension
                          DIMEXO:   Extension line offset
                          DIMGAP:   Dimension line gap
                          DIMLUNIT: Set linear dimension label format
                                    1: scientific
                                    2: decimal
                                    3: engineering
                                    4: architectural
                                    5: fractional
                          DIMTSZ:   Dimension arch tick size (0 for arrow)
                          DIMTXT:   Dimension text height
                          DIMZIN:   Trailing zeroes (0: on, 8: off)
  -t, -substitute=FONT1:FONT2
                          Substitute FONT1 with FONT2 in all texts

PDF import options:

      -noimages           Don't import images from PDF
      -relpath            Store images with relative path
                          Note: relative images paths are not properly
                          supported by the DXF/DWG format. Applications
                          may or may not load images from relative paths.
      -noclipping         Don't clip content to clipping areas
  -p, -page=PAGE          Set page to import from PDF file
```

```
macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.7 dwg2maptiles

Description

Command-line tool that renders a DWG/DXF drawing into a set of map tiles.

Usage

```
Usage: ./dwg2maptiles [OPTIONS]... <drawing file>

Converts a drawing file (DWG, DXF, ...) to a set of
bitmap files (PNG, JPEG, ...) for use with online maps, e.g. leaflet.js.

  -a, -antialiasing       Enable antialiasing
      -skip-empty         Skip empty tiles
  -b, -background=C       Set the background color as color name
                          e.g. 'white', '#ccdd00' or 'transparent'
      -block=BLOCK_NAME   Name of block (or layout) to output
                          (defaults to *Model_Space)
  -c, -color-correction   Enable color correction to prevent white on white
                          or black on black painting
      -center=X,Y         Set center of first tile to X,Y in drawing units
      -dry-run            Don't write any tiles.
  -d, -recompute-dim      Recompute dimension blocks if empty
                          (dimensions rendered with best compatibility)
      -tile-markers       Add tile markers (for testing, debugging)
  -e, -tile=ROW,COL       Generate only tile at ROW,COL
  -f, -force              Overwrite existing output file
      -flat               Flatten to 2D (Z==0)
  -fs, -font-substitution FONT1 FONT2
                          Substitute FONT1 with FONT2
  -h, -help               Display this help
  -k, -grayscale          Grayscale
      -layer=LAYER_NAMES  Comma-separated list of layer(s) to export
                          List may include regular expressions
  -l, -list               List supported formats
      -ltscale=FACTOR     Set linetype scale to FACTOR
  -m, -margin=M           Set auto zoom margin to M (in percent)
                          Default is 10%
      -tms                Y=0 is at the bottom (typically used for TMS,
                          default is Y=0 at the top for WMS)
  -n, -monochrome         Monochrome (black / white)
  -o, -outdir=DIR         Set output dir to DIR
                          default is same path and base name as input file
      -hq-postfix=P       Generate high-res tiles (512px) in addition to
```

```
                          256px tiles and use the given postfix for the
                          high-res output files, e.g. '@2x' for 'xxx@2x.png'
  -q, -quality=Q          Set the image quality / compression
                          (0=best compression..100=best quality)
  -r, -resolution=R       Set rendering resolution of first tile to R
                          (in pixels/drawing unit)
                          This overrides the width/height parameters
      -screen-lt          Use screen based linetypes (thin, visible patterns)
  -s, -subsection=N,M     Generate subsection N of M
                          Examples:
                          1,4: generate first of four subsections
                          4,4: generate fourth of four subsections
                          7,8: generate seventh of eight subsections
  -t, -format             File format (png, jpeg, gif)
      -text-threshold=T   Render texts below this T pixels as bounding box
  -u, -unit=[in|m|mm]     Assume input data to be in given unit
                          Default is the drawing unit
                          Mainly useful for drawings with unit 'None'
  -x, -width=W            Set bitmap width to W (in pixels)
  -y, -height=H           Set bitmap height to H (in pixels)
  -y, -height=H           Set bitmap height to H (in pixels)
      -x-offset=X         X offset for fine tuning
      -y-offset=Y         Y offset for fine tuning
      -opacity=O          Make all tiles (semi-)transparent using the given opacity
                          0.0: transparent
                          1.0: opaque
  -p, -origin             Draw zero point (origin)
      -web-mercator       Input is in Web Mercator coordinates,
                          from example from dwgmapconvert
  -z, -zoomlevels         Number or range of zoom levels, defaults to 0-6
                          Examples:
                              5: zoom levels 0,1,2,3,4,5
                            3-5: zoom levels 3,4,5
                          1,3,5: zoom levels 1,3,5
                            3-3: zoom level 3 only

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.8 dwg2pdf

Description

Command-line tool that exports a DWG/DXF drawing to a PDF file.

Usage

```
Usage: ./dwg2pdf [OPTIONS]... <drawing file>
```

```
Converts a drawing file (DWG, DXF, ...) to a PDF file.

  -a, -auto-fit           Auto fit and center drawing to paper
      -auto-orientation   Auto paper orientation (landscape or portrait)
                          based on document bounding box
      -block=BLOCK_NAMES  Comma-separated list of block(s) or
                          layout(s) to export
                          List may include regular expressions
                          List defines order of export unless
                          -block-sort is used
                          A block may be exported more than once
                          unless -block-unique is used
      -block-sort         Sort blocks in natural order
                          Layouts are sorted by tab order
                          and exported before blocks
                          Blocks are ordered alphanumerically
      -block-unique       Export every matching block only once
  -c, -center             Auto center drawing on paper
      -color-correction   Enable color correction to prevent white on white
                          or black on black painting
  -d, -recompute-dim      Recompute dimension blocks if empty
                          (dimensions rendered with best compatibility)
  -e, -papers             List all supported paper size names
      -point-size=SIZE    Set size of points in mm
  -f, -force              Overwrite existing output file
      -flat               Flatten to 2D (Z==0)
  -fs, -font-substitution FONT1 FONT2
                          Substitute FONT1 with FONT2
  -h, -help               Display this help
  -k, -grayscale          Grayscale
  -l, -landscape          Landscape (default is portrait)
      -layer=LAYER_NAMES  Comma-separated list of layer(s) to export
                          List may include regular expressions
      -list-printers      List all available printers
      -ltscale=FACTOR     Set linetype scale to FACTOR
  -m, -margin=M           Set paper margins to M
  -n, -monochrome         Monochrome (black / white)
  -o, -outfile=FILE       Set output file to FILE
                          default is same path and base name as input file
      -offset=X,Y         Set offset to X,Y in drawing units
                          ('3,4', '7/2,1/4', '0.5,-1.5', ...)
  -p, -paper=WxH          Set paper size to W x H in mm or to named paper size
                          ('A4', 'Letter', ...)
      -pdf-version=VER    Specify PDF version to
                          '1.4' (default) for PDF 1.4
                          '1.6' for PDF 1.6
                          'A' or 'A-1B' or 'A1B' for PDF/A-1B
      -printer=PRINTER    Print directly to PRINTER or default printer
                          if no printer given
      -rows=ROWS          Set number of paper rows to ROWS
  -s, -scale=SCALE        Set drawing scale to given scale
                          ('1', '2', '0.5', '1:1', '1:5', '5:1', ...)
      -cols=COLS          Set number of paper columns to COLS
  -u, -unit=[in|m|mm]     Assume input data to be in given unit
                          Default is the drawing unit
                          Mainly useful for drawings with unit 'None'
  -w, -min-lineweight=LW  Set minimum line weight on paper to LW Millimeter
      -max-lineweight=LW  Set maximum line weight on paper to LW Millimeter

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
```

```
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.9 dwg2svg

Description

Command-line tool that exports a DWG/DXF drawing to an SVG file.

Usage

```
Usage: ./dwg2svg [OPTIONS]... <drawing file>

Converts a drawing file (DWG, DXF, ...) file to an SVG file.

  -a, -adjust             Adjust page size to include thickest line weight
  -b, -bottom=MB          Set page margin at bottom to MB
      -block=BLOCK_NAME   Name of block (or layout) to output
                          (defaults to *Model_Space)
  -c, -circle=CR          Export points as circles with radius CR
      -color-correction   Enable color correction to prevent white on white
                          or black on black painting
  -d, -recompute-dim      Recompute dimension blocks if empty
                          (dimensions rendered with best compatibility)
      -decimals=D         Decimal places for float values (defaults to 4)
  -e, -equal-margins=M    Set all page margins to M
                          -l, -r, -t and -b are ignored
      -embed-images       Embed images (bitmaps)
  -f, -force              Overwrite existing output file
      -flat               Flatten to 2D (Z==0)
  -fs, -font-substitution FONT1 FONT2
                          Substitute FONT1 with FONT2
  -g, -preserve-geometry  Preserve as much of the original geometry as possible
                          Default is identical rendering
  -h, -help               Display this help
  -i, -inkscape           Generate additional information for Inkscape
  -l, -left=ML            Set page margin at left to ML
      -layer=LAYER_NAMES  Comma-separated list of layer(s) to export
                          List may include regular expressions
      -layer-attribute    Export layer attributes into QCAD SVG namespace
      -ltscale=FACTOR     Set linetype scale to FACTOR
  -n, -no-comments        Suppress comments for DXF entities
  -o, -outfile=FILE       Set output file to FILE
                          Default is same path and base name as input file
  -p, -open               Open the converted file after conversion
  -u, -unit=[in|m|mm]     Assume input data to be in given unit
                          Default is the drawing unit
                          Mainly useful for drawings with unit 'None'
  -r, -right=MR           Set page margin at right to MR
  -s, -scale=SF           Set scale factor to SF
  -t, -top=MT             Set page margin at top to MT
```

```
  -w, -min-lineweight=LW  Set minimum line weight to LW drawing units
                          Defaults to 0.001 since 0 in SVG means invisible
      -no-text-transforms Don't use transformations for text entities
  -x, -cross=CS           Export points as crosses with size CS

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.

Points:    If neither -c nor -x is given, points are not exported.
SVG Units: SVG only supports the following CAD units: mm, cm, in.
           - Unsupported Metric drawing units are converted to Millimeters.
           - Unsupported Imperial drawing units are converted to Inches.
           Use -s to scale and avoid large values.
Margins:   Margin values have to be specified in the (converted) SVG unit.

Examples:
Adjust page, set all margins to 1.5, export points as circles with radius 0.5:
./dwg2svg -a -e 1.5 -c 0.5 file.dwg

Set scale factor to '1:200':
./dwg2svg -s 1:200 file.dwg
```

15.10 dwgexplode

Description

Command-line tool that explodes complex entities (blocks, polylines and similar) of a DWG/DXF drawing into their basic components.

Usage

```
Usage: ./dwgexplode [OPTIONS]... <drawing file>

Explodes blocks, ellipses, texts in the given drawing file (DXF, DWG).

  -f, -force              Overwrite existing output file
  -t, -types=TYPES        Comma separated list of entity types that
                          should NOT be exploded any further
                          Default is to explode everything into points,
                          lines, arcs and circles
  -h, -help               Display this help
  -o, -output=FILE        Set CAD output file to FILE

Examples:
./dwgexplode -t Text,Polyline -o out.dxf in.dxf
    Explodes everything except texts and polylines.
    Dimensions for example are exploded into lines, polylines
    (for the arrows) and a text label.
```

```
macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.11 dwghatch

Description

Command-line tool that adds hatches to the closed areas of a DWG/DXF drawing.

Usage

```
Usage: ./dwghatch [OPTIONS]... <drawing file>

Fill all fillable (closed) boundaries of the given drawing
file (DXF, DWG) with a solid fill hatch.

  -f, -force              Overwrite existing output file
  -h, -help               Display this help
      -per-layer          Hatch entities per layer, one layer, one hatch, same layer
      -replace            Delete boundary entities after creating hatch
  -o FILE, -output=FILE   Set CAD output file to FILE

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.12 dwghatcharea

Description

Command-line tool that reports the hatched area of a DWG/DXF drawing.

Usage

```
Usage: ./dwghatcharea [OPTIONS]... <drawing file>

List the area of all hatches (or solid fills) of the given drawing
file (DXF, DWG) per layer. The area of all entities on all blocks are considered.
If hatch loops are intersecting, the result is not correct / not defined.

      -block=BLOCK_NAME   Name of block (or layout) to process
                          (defaults to *Model_Space)
  -f, -factor             Linear factor applied to all lengths
      -flat               Flatten to 2D (Z==0)
  -h, -help               Display this help
      -layer=LAYER_NAMES  Comma-separated list of layer(s) to process
                          List may include regular expressions
                          Defaults to all visible layers
  -s, -segment=LENGTH     Segment length for curve segmentation.
                          Shorter means slower but more precise.
                          Defaults to 0.1 drawing units.

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.13 dwginfo

Description

Command-line tool that prints information about a DWG/DXF drawing, such as its entities, layers and blocks.

Usage

```
Usage: ./dwginfo [OPTION]... <drawing file>

Exracts various information from a DXF/DWG file.

  -h, -help               Display this help
  -b, -blocklist          List all block names
  -l, -layerlist          List all layer names
  -e, -entities           Output number of entities in file

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:
```

```
  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.14 dwgmapconvert

Description

Command-line tool that converts the map projection (coordinate reference system) of a DWG/DXF drawing.

Usage

```
Usage: ./dwgmapconvert [OPTIONS]... <drawing file>

Converts a drawing file (DWG, DXF, ...) from one CRS to another.
Typically used to convert the coordinate system before running
dwg2maptiles.

      -block=BLOCK_NAME   Name of block (or layout) to output
                          (defaults to *Model_Space)
      -crs-in=CRS         Data is in given coordinate reference system (CRS *)
      -crs-out=CRS        Transform to given coordinate reference system (CRS *)
                          * CRS can be either of:
                          - An object code such as 'EPSG:2056'
                          - An object name such as 'LV95'
                          - A WKT string ('PROJCS[...]')
                          - A proj-string ('+proj=...')
      -crs-list           List supported coordinate reference systems (CRS)
      -pipeline=PIPELINE  Transform according to the given transform pipeline PIPELINE
  -d, -recompute-dim      Recompute dimension blocks if empty
                          (dimensions rendered with best compatibility)
  -f, -force              Overwrite existing output file
  -fs, -font-substitution FONT1 FONT2
                          Substitute FONT1 with FONT2
  -h, -help               Display this help
      -layer=LAYER_NAMES  Comma-separated list of layer(s) to export
                          List may include regular expressions
      -ltscale=FACTOR     Set linetype scale to FACTOR
      -retain-texts       Retain (don't segmentate) text labels
                          Only the text position and height is transformed
      -segment-length=L   Segmentate all entities with segment length L
      -simplify           Simplify after segmentation
  -o, -outfile=FILE       Set output file to FILE
  -r, -release=RELEASE    Set output DXF/DWG release to RELEASE.
                          Defaults to R27.
  -u, -unit=[in|m|mm]     Assume input data to be in given unit
                          Default is the drawing unit
                          Mainly useful for drawings with unit 'None'

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:
```

```
  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.15 extracttextinfo

Description

Command-line tool that extracts the text contents of a DWG/DXF drawing.

Usage

```
Usage: ./extracttextinfo [OPTIONS]... <drawing file>

Extracts information from a drawing file (DWG, DXF, ...).

  -f, -force              Overwrite existing output file
  -o FILE, -outfile=FILE  Set output file to FILE
                          default is same path and base name as input file

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

15.16 svg2dwg

Description

Command-line tool that converts an SVG file to a DWG/DXF drawing.

Usage

```
Usage: ./svg2dwg [OPTIONS]... <drawing file>

Converts the given SVG file to DXF or DWG.

  -f, -force              Overwrite existing output file
  -t, -filter=FILTER      Use specified format filter
                          e.g. DXF R15, DWG R27, etc.
      -layer0=color,weight,type
                          Set layer 0 attributes (e.g. 'white,0.05,Continuous')
```

```
      -by-layer           Set all attributes to 'By Layer'
  -h, -help               Display this help
  -o, -output=FILE        Set CAD output file to FILE

macOS specific options:

  -no-dock-icon           Don't show icon in dock. This should only be used
                          in combination with -no-gui.

Qt/X11 platform arguments:

  -platform offscreen     Run without requiring an X11 server or display.
                          Use this argument to run this command line tool in a
                          Linux server environment.

Qt/X11 environment variables:

  QT_QPA_FONTDIR          Set to fonts directory in a server environment, e.g. 'fonts/qt'.
```

16 Tutorials

16.1 Command Line Tools

QCAD ships with command line tools that process drawings without opening the graphical interface, making them ideal for automation, batch processing and integration into larger workflows. This section explains how to put these tools to work, for example to generate map tiles from a drawing for display in online maps such as Leaflet.

16.1.1 Generating Map Tiles

Introduction

QCAD comes with two command line tools to create map tiles for use with online map displays such as Leaflet. Leaflet is a JavaScript library to work with interactive maps in a browser.

Presenting a Drawing Online

The simplest use case for an online map is to provide visitors of a web site with an interactive, zoomable view of a drawing.

Here's an example that generates the map tiles for any given drawing file:

```
dwg2maptiles -z 0-6 -c -b white -hq-postfix=@2x -a -o path/to/output/directory drawing.dxf
```

The `-z` switch specifies the zoom levels to generate, here from zoom level 0 to zoom level 6. This means that starting from level 0 (whole drawing of auto zoom), the user can zoom in 6 times until the limit is reached. Each time the user zooms in, the drawing is enlarged by a factor of 2, so with 6 zoom levels, the user can zoom in by a factor of 2^6 (=64). Generating more zoom levels requires more time and disk space. Each additional zoom level typically requiring four times more time and disk space to generate than the previous one.

The tiles are generated in the given output directory (`-o`) with the standard directory structure and file names for map tiles (z/x/y.png).

In the above example, we want the drawing to be rendered on a white background (`-b white`). White lines should be corrected to black to make sure all lines are visible (`-c`).

The `-hq-postfix=@2x` argument tells dwg2maptiles to generate high-quality (hq, high-res, retina) tiles at 512x512px along with the standard resolution tiles at 256x256px.

To further increase quality, we also use the `-a` switch for antialiasing.

View example with Leaflet online

Generating Georeferenced Map Tiles

dwg2maptiles can also be used to generate exact overlays for world maps. For this, a drawing usually has to be transformed from a local coordinate system into Web Mercator coordinates as commonly used by online maps.

QCAD Professional comes with a separate command line tool to transform drawings from one coordinate reference system (CRS) to another: dwgmapconvert.

dwgmapconvert is typically used to convert all coordinates in a given drawing file from a known, local coordinate system to the EPSG:3857 (sometimes called "Web Mercator") coordinate system. This is the coordinate system used by most online maps, for example Google maps or mapbox. To use a drawing as input for dwg2maptiles and produce georeferenced map tiles, the drawing needs to be in the EPSG:3857 coordinate system.

In the EPSG:3857 coordinate system the whole world map fits into a 20037508.3427892m square at zoom level 0. If a drawing is converted into this coordinate system, the drawing is transformed in such a way that it fits into that world square and that the drawing is located at the correct, projected world coordinates.

For example, if a drawing was created in EPSG:2056 (also LV95, the local Swiss coordinate system), it can be transformed to EPSG:3857 with dwgmapconvert as follows:

```
dwgmapconvert -segment-length=0.5 -crs-in=EPSG:2056 -crs-out=EPSG:3857 -f -o
 drawing_epsg3857.dxf drawing_epsg2056.dxf
```

Since map projections are usually non-linear, curves and longer lines need to be split up into small line segments which can then be projected in a non-linear way. If the switch `-segment-length` is provided, dwgmapconvert does this segmentation automatically, using the given segment length. The segment length is specified in drawing units of the original drawing (here Meter).

`-crs-in` is used to specify the input coordinate system, here EPSG:2056.

`-crs-out` is used to specify the output coordinate system which must be EPSG:3857 if the output should later be used with dwg2maptiles.

dwgmapconvert comes with support for LV95 as input and WebMercator as output.

Other coordinate systems will be added in future releases or can be added by the user.

To add support for a new coordinate system, two equations must be provided to convert the coordinates from the new coordinate system to WGS84 (Longitude/Latitude, like GPS coordinates).

New coordinate systems can be added to a JSON configuration file called CRS.json, located in the "scripts" folder of the QCAD installation that is used. Here's an example CRS.json file that configures the coordinate system LV95. To be able to use LV95 as input for the dwgmapconvert tool, we need to provide the "toWgs84" section with equations for the x and y coordinate conversion. Here, we also use two auxiliary variables, x_ and y_ which must be initialised in the x equation in order to be accessible in both equations.

We could also provide a "fromWgs84" section to allow conversion from other coordinate systems into LV95. Both sections are optional, depending on the conversion direction you want to support.

```
{
  "CRS" : [
    {
      "id" : "EPSG:2056 CH1903+ / LV95",
      "toWgs84" : {
        "x": "x_ = (y-1200000)/1000000; y_ = (x-2600000)/1000000; (2.6779094 + 4.728982*y_ +
 0.791484*y_*x_ + 0.1306*y_*x_*x_ - 0.0436*y_*y_*y_) * 100 / 3",
        "y": "(16.9023892 + 3.238272*x_ - 0.270978*y_*y_ - 0.002528*x_*x_ - 0.0447*y_*y_*x_
 - 0.0140*x_*x_*x_) * 100 / 36"
      },
      "aliases" : [ "LV95", "Swiss" ]
    }
  ]
}
```

Generating Map Tiles

Once we have converted the drawing to EPSG:3857 coordinates, we can use dwg2maptiles to generate the tiles for the overlay layer in leaflet:

```
dwg2maptiles -web-mercator -skip-empty -hq-postfix=@2x -z 22 -b transparent -o path/to/
output/directory drawing_epsg3857.dxf
```

Note that we need to pass the `-web-mercator` switch to dwg2maptiles to let it know that it should not perform an auto zoom but generate a map overlay in exact EPSG:3857 coordinates as present in the file.

`-skip-empty` tells dwg2maptiles to not generate tiles that don't contain any part of the drawing. Without this switch, 2^zoomLevel tiles will be generated per zoom level. That would be 4'194'304 tiles for zoom level 22. Since an overlay drawing very likely only covers a very small part of the world map, this switch should always be used when creating overlay maps.

`-b transparent` sets the background to transparent, so the map can be used as an overlay for an existing world map (e.g. Google Map, mapbox, etc.). To control the color in which black or white lines are displayed, the background can be set to black transparent or white transparent with `-b "#00ffffff"` (white transparent background, black lines) or `-b "#00000000"` (black transparent background, white lines) respectively. Be sure to also pass the `-c` switch in this case to force automatic color correction.

The argument `-hq-postfix=@2x` can be used to generate high-res tiles (512px) in addition to the normal 256px tiles. The value "@2x" is recommended and widely used for high-res tiles, also in leaflet. With this switch, for every low-res tile "N.png", a high-res tile "N@2x.png" is created.

Using Multiple Processes (and CPU Cores)

Generating map tiles can be a CPU intensive task, depending on zoom levels, drawing complexity and other settings.

To generate map tiles faster, you can divide the job among multiple available CPU cores. For this feature, dwg2maptiles offers the `-s` switch to generate only a slice of the required tiles. The `-s` switch expects two comma-separated parameters: the slice to generate and the total number of slices used. For example `-s 1,4` would generate the first slice of four slices in total. To generate all tiles of the above example using four CPU cores, you could run:

```
dwg2maptiles -web-mercator -skip-empty -hq-postfix=@2x -z 22 -s 1,4 -b transparent -o path/
to/output/directory drawing_epsg3857.dxf &
dwg2maptiles -web-mercator -skip-empty -hq-postfix=@2x -z 22 -s 2,4 -b transparent -o path/
to/output/directory drawing_epsg3857.dxf &
dwg2maptiles -web-mercator -skip-empty -hq-postfix=@2x -z 22 -s 3,4 -b transparent -o path/
to/output/directory drawing_epsg3857.dxf &
dwg2maptiles -web-mercator -skip-empty -hq-postfix=@2x -z 22 -s 4,4 -b transparent -o path/
to/output/directory drawing_epsg3857.dxf &
wait
```

Of course you could also divide tasks by zoom levels or use slices for higher zoom levels only and a single process for all the lower (cheaper) zoom levels, etc.

Example

An interactive example provides two overlay layers to the map of the Swiss canton of Zurich: the protected forest area and the wildlife corridors. The original data came in EPSG:2056 (LV95) coordinates and was transformed to EPSG:3857 using dwgmapconvert.

View example with Leaflet online

16.2 Development

QCAD is built to be extended, and this section shows how to tailor it to your own needs through scripting and plugins. The tutorials cover experimenting interactively in the Script Shell, running procedural scripts from a file, writing command line tools that leverage QCAD functionality, building interactive script actions that add menus and handle user input, and extending the application with compiled C/C++ plugins. Whether you want to automate repetitive tasks or add entirely new features, these guides provide a starting point.

16.2.1 Command Line Tool Scripts

Introduction

Scripts can be used to create command line tools that leverage QCAD functionality. Such a script might for example generate a drawing based on parameters, convert a drawing or analyse a drawing.

Example

The basic structure of a command line tool script can look for example as shown below. This is also a fully functional example script that can be used as command line tool to add a line from 0,0 to 100,100 to a given drawing and export that drawing under a new file name:

```
include("scripts/simple.js");
include("scripts/Tools/arguments.js");

function printHelp() {
    print("Usage: " + args[1] + " [OPTION]... <drawing file>");
    print();
    print("This tool does...");
    print();
    print("  -f, -force          Overwrite existing output file");
    print("  -h, -help           Display this help");
    print("  -o, -outfile=FILE   Set output file to FILE");
    print("  -p, -parameter      Some parameter...");
    print("  ...");
    printGenericUsage();
    print();
}

function main() {
    if (testArgument(args, "-h", "-help")) {
        printHelp();
        return;
    }

    if (args.length < 3) {
        print("No input file given. Try -h for help.");
        return;
    }

    var inFile = args[args.length - 1];
    if (inFile.indexOf("-") === 0) {
        print("No input file. Try -h for help.");
        return;
    }
```

```
    var fi = new QFileInfo(inFile);
    if (!fi.isAbsolute()) {
        inFile = RSettings.getLaunchPath() + QDir.separator + inFile;
        fi = new QFileInfo(inFile);
    }

    var outFile = getArgument(args, "-o", "-outfile");
    if (outFile !== undefined) {
        outFile = getAbsolutePathForArg(outFile);
    }
    else {
        print("No output file. Try -h for help.");
    }

    if (new QFileInfo(outFile).exists() && !testArgument(args, "-f", "-force")) {
        print("Output file exists already, not overwriting: ", outFile);
        print("Use -f to force overwrite");
        return;
    }

    // create new document (off screen):
    var doc = new RDocument(new RMemoryStorage(), new RSpatialIndexNavel());
    var di = new RDocumentInterface(doc);

    // import given file:
    if (di.importFile(inFile) != RDocumentInterface.IoErrorNoError) {
        di.destroy();
        print("Cannot import file:", inFile);
        return;
    }

    // add line to drawing (using QCAD Simple API):
    startTransaction(di);
    addLine(0, 0, 100, 100);
    endTransaction();

    // export to given file (-o)
    if (!di.exportFile(outFile)) {
        di.destroy();
        print("Cannot export file:", outFile);
        return;
    }

    di.destroy();
}

if (typeof(including) == 'undefined' || including === false) {
    main();
}
```

Running Command Line Tool Scripts

Let's assume this script is saved under scripts/Tools/ExTool/ExTool.js.

It can then be run as follows:

```
qcad -autostart scripts/Tools/ExTool/ExTool.js -f -o out.dxf in.dxf
```

If the script is run as part of a web service, be sure to also pass the switches `-no-gui` and `-allow-multiple-instances` to tell QCAD that no GUI is available and multiple instances can be run in parallel:

```
qcad -no-gui -allow-multiple-instances -autostart scripts/Tools/ExTool/ExTool.js -f -o
 out.dxf in.dxf
```

On a server you would typically also pass `-platform offscreen` to run headless (i.e. without an X11 server on Linux):

```
qcad -platform offscreen -no-gui -allow-multiple-instances -autostart scripts/Tools/ExTool/
ExTool.js -f -o out.dxf in.dxf
```

16.2.2 Creating a QCAD Plugin

Introduction

QCAD plugins can be used to extend QCAD with functionality implemented in C or C++.

Implementing RPluginInterface

All QCAD plugins must be directly or indirectly derived from QObject and the abstract base class RPluginInterface.

File RExamplePlugin.h

For our first, minimalistic example plugin, we add empty or minimalistic implementations for most methods of RPluginInterface directly to the header file. This initial structure of the header file can be copied to get a first working plugin that can be loaded into QCAD.

```
#include <QObject>
#include <QScriptEngine>

#include "RPluginInterface.h"

class RExamplePlugin : public QObject, public RPluginInterface
{
    Q_OBJECT
    Q_INTERFACES(RPluginInterface)
#if QT_VERSION >= 0x050000
    Q_PLUGIN_METADATA(IID "org.qcad.exampleplugin")
#endif

public:
    virtual bool init() { return true; }
    virtual void uninit(bool) {}
    virtual void postInit(InitStatus status) {}
    virtual void initScriptExtensions(QScriptEngine& engine) {}
    virtual RPluginInfo getPluginInfo();
    virtual bool checkLicense() { return true; }
};
```

File RExamplePlugin.cpp

For method getPluginInfo, we add a simple implementation to the cpp file for the benefit of seeing something in the about dialog of QCAD.

```
#include "RExamplePlugin.h"

RPluginInfo RExamplePlugin::getPluginInfo() {
    RPluginInfo ret;
    ret.set("Version", "1.0");
    ret.set("ID", "EXAMPLE");
    ret.set("Name", "Example Plugin");
    ret.set("License", "GPLv3");
```

```
    ret.set("URL", "http://qcad.org");
    return ret;
}

#if QT_VERSION < 0x050000
QT_BEGIN_NAMESPACE
Q_EXPORT_PLUGIN2(example, RExamplePlugin)
QT_END_NAMESPACE
#endif
```

File exampleplugin.pro

To compile our plugin, we also need a Qt project file that can be turned into a Makefile with Qt qmake.

```
CONFIG += plugin
TARGET = example
include(../../../shared.pri)

TEMPLATE = lib
HEADERS = RExamplePlugin.h
SOURCES = RExamplePlugin.cpp
DESTDIR = ../../../plugins
LIBS += -lqcadcore -lqcadgui -lqcadecmaapi
```

The include on line 3 should point to the file shared.pri of the QCAD source code installation (e.g. qcad/shared.pri). As DESTDIR, you may want to choose the plugins folder of a working QCAD installation, so the plugin is compiled in the right place from where it can also be loaded by QCAD.

Compilation

Your plugin can be compiled with these three files, creating the file qcad/plugins/qcadexample.dll or libqcadexample.so or libqcadexample.dylib.

Note: if you run a QCAD binary compiled in debug mode, the plugin must also be compiled in debug mode. If you run QCAD in release mode, the plugin must be compiled in release mode as well.

To compile the plugin in debug mode run:

Linux / macOS:

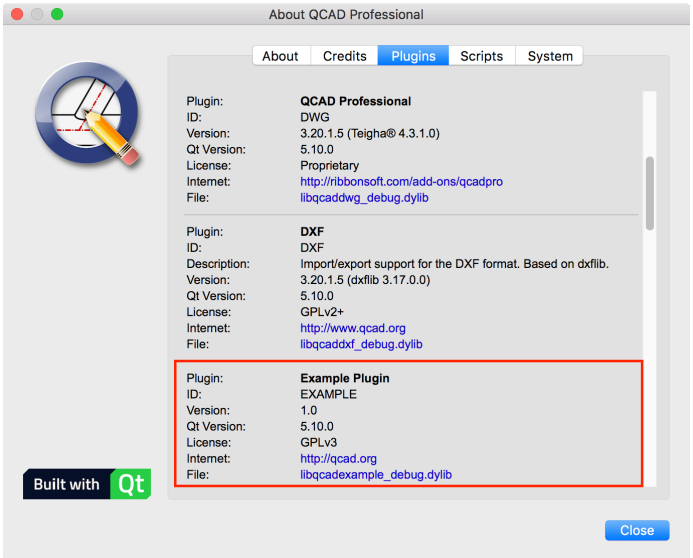
```
qmake
make debug
```

Windows:

```
qmake
nmake debug
```

To compile in release mode, run `make release` or `nmake release` instead.

Obviously, the plugin won't do anything useful yet, but you should be able to see your plugin in the QCAD about dialog:



If you cannot see your plugin, please check for errors in the terminal while starting QCAD. You might also see an error message in the about dialog.

Compiling Scripts as Resources into a Plugin

QCAD plugins may be used to compile a collection of scripts and other resources (ui files, icons, etc.) into a plugin. This has the advantage that the code of multiple ECMAScript files can be bundled inside a single, compiled binary file. Plugins tend to load faster than individual files from disk and take up less disk space.

File scripts/MyTool/MyTool.js

Let's assume there's a folder scripts/MyTool with a file MyTool.js which contains a QCAD tool, implemented in ECMAScript. The tool adds itself at the top of the Misc menu and outputs "Hello World" when triggered.

```
include("scripts/EAction.js");

function MyTool(guiAction) {
    EAction.call(this, guiAction);
}

MyTool.prototype = new EAction();

MyTool.prototype.beginEvent = function() {
    EAction.prototype.beginEvent.call(this);

    EAction.handleUserMessage("Hello World");

    this.terminate();
};

MyTool.init = function(basePath) {
    var action = new RGuiAction("&MyTool", RMainWindowQt.getMainWindow());
    action.setRequiresDocument(true);
    action.setScriptFile(basePath + "/MyTool.js");
    action.setGroupSortOrder(0);
    action.setSortOrder(0);
    action.setWidgetNames(["MiscMenu"]);
};
```

File scripts.qrc

To compile this script tool into our C++ plugin, we need to create a Qt resource file which points qmake to the file we want to compile into the plugin:

```
<!DOCTYPE RCC>
<RCC version="1.0">
    <qresource>
        <file alias="scripts/MyTool/MyTool.js">scripts/MyTool/MyTool.js</file>
    </qresource>
</RCC>
```

The qrc file lists the paths under which resources are to be made available in Qt's resource system (alias="scripts/MyTool/MyTool.js") and the actual path where the file can be found on disk (used for compilation only). The above example means that there is a file on disk under scripts/MyTool/MyTool.js which will be compiled into the plugin. As soon as the plugin is loaded, the file will be available under path ":/scripts/MyTool/MyTool.js". QCAD will find the tool automatically under that path since it scans folder "scripts" as well as ":/scripts" for scripts.

We also need to add the qrc file to the project file (exampleplugin.pro) by adding the following line:

```
RESOURCES = scripts.qrc
```

After editing the file exampleplugin.pro, run qmake and make again and load the plugin. There should now be a menu entry "MyTool" at the top of the Misc menu in QCAD.

Source Code

You can find the complete source code for this example plugin at:

https://github.com/qcad/qcad/tree/master/support/examples/exampleplugin

Troubleshooting

- If your plugin compiles but cannot be loaded by QCAD, please make sure that the same Qt version is used as for QCAD (see *Help* > *About QCAD*). Please make sure that if QCAD is run in release mode, your plugin is also compiled in release mode.
- Launch QCAD from a terminal or the command line and check the console output for error messages. Output messages in your plugin to confirm it is being loaded. For example in your init function.
- Check the about dialog (*Help* > *About QCAD* > *Plugins*) for error messages.

16.2.3 Interactive Script Actions

Introduction

Script actions are scripts that add an entry to a menu and/or tool bar and that can handle user interactions. Script actions stay active until terminated by the user or until they self-terminate.

Events

As soon as the script action is started it handles various events until it is terminated. An event is something that occurs if something is happening. For example if the script action is started, `beginEvent` is called. If the user clicks an entity a `pickEntity` event is triggered, if the user clicks a coordinate, a `pickCoordinate` event occurs, etc.

The minimal structure of a script action is as follows:

```
include("scripts/EAction.js");

function ExMyMinimal(guiAction) {
    EAction.call(this, guiAction);
}

ExMyMinimal.prototype = new EAction();

ExMyMinimal.init = function(basePath) {
    var action = new RGuiAction(qsTr("&Minimal Example"), RMainWindowQt.getMainWindow());
    action.setRequiresDocument(true);
    action.setScriptFile(basePath + "/ExMyMinimal.js");
    action.setGroupSortOrder(100000);
    action.setSortOrder(0);
    action.setWidgetNames(["ExamplesMenu"]);
};
```

This example script adds a menu to the bottom of the menu *Misc* > *Examples*. The menu text is "Minimal Example".

Note that in order for the script to be found, the file name needs to match the class name, i.e. "ExMyMinimal.js" in this case. It also needs to reside inside a directory with the same name "ExMyMinimal", so this script can for example be put into scripts/Misc/ExMyMinimal/ExMyMinimal.js.

You can also put your scripts into a local scripts folder within your user home folder. To find out the exact folder, open the about dialog (*Help* > *About QCAD...*) and navigate to the *System* tab. There you can see the data location under *Data directory*. This is the directory under which you have to create a subfolder called *scripts* and sub folders, one folder per script tool, for example /path/to/data directory/scripts/MyScripts/MyScript1/MyScript1.js

The exact location depends on your system and its configuration.

Adding beginEvent

The script above is fully functional and can be triggered. However, it doesn't actually do anything when triggered. Moreover, once triggered, the script stays active until the user terminates it by clicking the right mouse button. To change this, let's implement beginEvent to print something to the command line history of QCAD and terminate the action:

```
include("scripts/EAction.js");

function ExMyMinimal(guiAction) {
    EAction.call(this, guiAction);
}

ExMyMinimal.prototype = new EAction();

ExMyMinimal.prototype.beginEvent = function() {
    EAction.prototype.beginEvent.call(this);

    EAction.handleUserMessage("Hello World!");

    this.terminate();
};

ExMyMinimal.init = function(basePath) {
    var action = new RGuiAction(qsTr("&Minimal Example"), RMainWindowQt.getMainWindow());
    action.setRequiresDocument(true);
    action.setScriptFile(basePath + "/ExMyMinimal.js");
```

```
    action.setGroupSortOrder(100000);
    action.setSortOrder(0);
    action.setWidgetNames(["ExamplesMenu"]);
};
```

If the tool *Misc* > *Examples* > *Minimal Example* is now started, it prints "Hello World!" to the command line history (line 12) and then terminates (line 14).

If a script does not require any user interaction, such a script can be used to add a menu that does something and then terminates. Examples for such actions are *View* > *Auto Zoom*, *Select* > *Select All*, *Edit* > *Delete*, etc.

Adding Interaction

As soon as a script requires any kind of user interaction, we need to implement more event handlers and tell the script what the user needs to do next (e.g. picking an entity or defining a coordinate). In the next step, we enter a state in which the action expects a coordinate from the user. We then draw a circle at every position the user clicks or enters.

```
include("scripts/EAction.js");

function ExMyMinimal(guiAction) {
    EAction.call(this, guiAction);

    this.pos = undefined;
}

ExMyMinimal.prototype = new EAction();

ExMyMinimal.prototype.beginEvent = function() {
    EAction.prototype.beginEvent.call(this);

    var di = this.getDocumentInterface();
    di.setClickMode(RAction.PickCoordinate);
};

ExMyMinimal.prototype.pickCoordinate = function(event, preview) {
    this.pos = event.getModelPosition();

    if (preview) {
        this.updatePreview();
    }
    else {
        this.applyOperation();
    }
};

ExMyMinimal.prototype.getOperation = function(preview) {
    var doc = this.getDocument();

    var op = new RAddObjectOperation();
    var circle = new RCircle(this.pos, 1);
    op.addObject(shapeToEntity(doc, circle));
    return op;
};

ExMyMinimal.init = function(basePath) {
    var action = new RGuiAction(qsTr("&Minimal Example"), RMainWindowQt.getMainWindow());
    action.setRequiresDocument(true);
    action.setScriptFile(basePath + "/ExMyMinimal.js");
    action.setGroupSortOrder(100000);
```

```
    action.setSortOrder(0);
    action.setWidgetNames(["ExamplesMenu"]);
};
```

In the beginEvent, we no longer terminate the action immediately, but let it run until the user terminates it (right click or Escape). We then implement pickCoordinate to store the position of the mouse cursor or the coordinate entered and to either update the preview or apply the operation (i.e. add the circle). pickCoordinate is called whenever the user moves the mouse to display a preview of the operation that is planned. When the user clicks or enters a coordinate, it is called with parameter preview set to false to indicate that a definitive coordinate has been picked or entered.

updatePreview on line 22 previews the operation returned by getOperation while applyOperation on line 25 actually applies the operation to our document.

getOperation must be implemented to return the operation to preview or apply to the document. This is slightly more complex than what we have seen in the simple API above. This is because a single operation can be used to add multiple objects, modify objects or delete objects.

Adding Widgets to the Options Toolbar

The circle drawn in our example always has a radius of 1 drawing unit (see line 33). In a next step, we want to allow the user to enter a radius for the circle. QCAD usually uses the options toolbar at the top to display and change such tool parameters. For this, we need to define what widgets we want to show in the options toolbar and what parameters they control. This can be done with a UI file, an XML file that defines a widget and its contents. UI files can be comfortably designed using a software called Qt Designer which comes as part of the Qt toolkit. For this example, we use a simple UI file that can also be created in a text editor (file ExMyMinimal.ui):

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>ExMyMinimal</class>
    <widget class="QWidget" name="ExMyMinimal">
        <layout class="QHBoxLayout">
            <item>
                <widget class="QLabel" name="RadiusLabel">
                    <property name="text">
                        <string>&amp;amp;Radius:</string>
                    </property>
                    <property name="buddy">
                        <cstring>Radius</cstring>
                    </property>
                </widget>
            </item>
            <item>
                <widget class="RMathLineEdit" name="Radius">
                    <property name="text">
                        <string notr="true">1</string>
                    </property>
                </widget>
            </item>
        </layout>
    </widget>
    <customwidgets>
        <customwidget>
            <class>RMathLineEdit</class>
            <extends>QLineEdit</extends>
            <header>RMathLineEdit.h</header>
        </customwidget>
    </customwidgets>
```

```
    <resources/>
    <connections/>
</ui>
```

The UI file defines two widgets: a label (QLabel) and a line edit (RMathLineEdit). Important is the name of the line edit ("Radius"). The widget is automatically linked to our script through this name. All we have to do in our script is to define which UI file we want to use (line 9) and to implement a new event handler called slotRadiusChanged, that's "slot" + [the name of our line edit] + "Changed" (line 45):

```
include("scripts/EAction.js");

function ExMyMinimal(guiAction) {
    EAction.call(this, guiAction);

    this.pos = undefined;
    this.radius = undefined;

    this.setUiOptions("ExMyMinimal.ui");
}

ExMyMinimal.prototype = new EAction();

ExMyMinimal.prototype.beginEvent = function() {
    EAction.prototype.beginEvent.call(this);

    var di = this.getDocumentInterface();
    di.setClickMode(RAction.PickCoordinate);
};

ExMyMinimal.prototype.pickCoordinate = function(event, preview) {
    this.pos = event.getModelPosition();

    if (preview) {
        this.updatePreview();
    }
    else {
        this.applyOperation();
    }
};

ExMyMinimal.prototype.getOperation = function(preview) {
    if (isNull(this.pos) || isNull(this.radius)) {
        return undefined;
    }

    var doc = this.getDocument();

    var op = new RAddObjectOperation();
    var circle = new RCircle(this.pos, this.radius);
    op.addObject(shapeToEntity(doc, circle));
    return op;
};

ExMyMinimal.prototype.slotRadiusChanged = function(v) {
    this.radius = v;
    this.updatePreview();
};

ExMyMinimal.init = function(basePath) {
    var action = new RGuiAction(qsTr("&Minimal Example"), RMainWindowQt.getMainWindow());
```

```
    action.setRequiresDocument(true);
    action.setScriptFile(basePath + "/ExMyMinimal.js");
    action.setGroupSortOrder(100000);
    action.setSortOrder(0);
    action.setWidgetNames(["ExamplesMenu"]);
};
```

This new function slotRadiusChanged is called whenever the user enters a new radius. It sets the member variable this.radius which is in turn used when creating the circle in getOperation.

All scripts in QCAD are based on one of these concepts outlined in this tutorial.

Since every tool in QCAD is implemented as a script on the top level, there are plenty of example scripts available. You can find them in our git repository.

16.2.4 Running Scripts from a File

Introduction

It can quickly become tedious to type scripts into the Script Shell of QCAD. As soon as scripts are getting slightly more complex, you might want to write them into a file instead. For procedural scripts which do not require interaction, you can simply create a text file with calls to the QCAD API and run the script using *Misc* > *Development* > *Run Script*.

Examples

Plotting Functions

A typical task for a script would be to plot a mathematical function. This example plots a Lissajous curve.

Note that the lines starting with `//` are comments. Comments are ignored and are purely intended for explanation.

```
// constants:
a = 7; b = 6;
// phase angle:
phi = 0.2 * Math.PI;
// iteration step:
step = Math.PI / 256;
// width of plot (amplitude in X):
w = 100;
// height of plot (amplitude in Y):
h = 100;

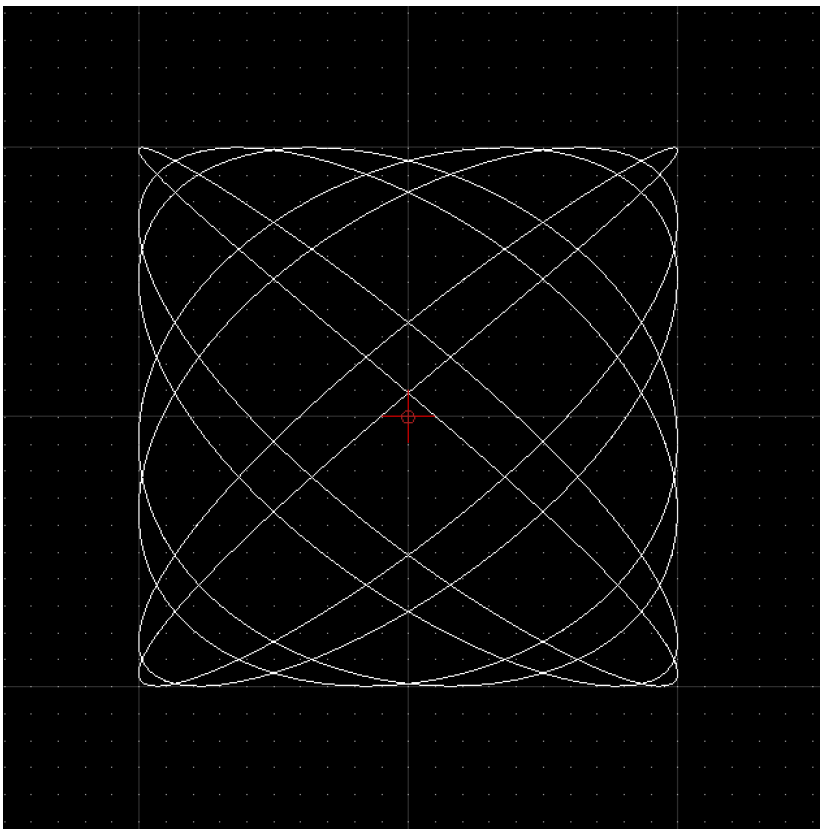
// array of x,y coordinates on the curve:
v = [];

// loop:
for (t = 0.0; t < Math.PI * 2; t += step) {
    // compute next x,y coordinate:
    x = w * Math.sin(a * t + phi);
    y = h * Math.sin(b * t);

    // append coordinate to our array of coordinates:
    v.push([x,y]);
}

// add a spline to the drawing, using
// the computed coordinates as fit points:
addSpline(v, true);
```

```
// auto zoom:
autoZoom();
```



16.2.5 The Simple API

Introduction

The QCAD Simple API is a small set of global script functions that simplify common tasks such as adding entities to a drawing, querying entities and modifying them (moving, trimming, rounding, offsetting, ...). It is mainly designed for use in the Script Shell and for short procedural scripts that are run from a file, but it can be used from any script.

The Simple API covers only a small part of the complete QCAD API. Since it is written in ECMAScript (JavaScript) itself, it is also a good starting point to learn how the underlying QCAD API works. The source code can be found in the files `scripts/simple.js` and `scripts/simple_*.js` of the QCAD source code.

Using the Simple API

In the Script Shell (*Misc* > *Development* > *Script Shell*) and in scripts launched with *Misc* > *Development* > *Run Script*, all Simple API functions are available directly. Type for example:

```
addLine(0,0, 100,100);
```

In your own script files, for example in interactive script actions or command line tools, include the file `simple.js` first:

```
include("scripts/simple.js");
```

```
addLine(0,0, 100,100);
```

All functions operate on the drawing that is currently open in the QCAD application window, unless a transaction is used to redirect them to another document (see Transactions).

All examples on this page can be pasted into the Script Shell of an empty drawing. Each example creates the entities it needs. Use *View* > *Auto Zoom* or the function `autoZoom()` to see the result.

Conventions

Coordinates

Positions can be passed in three different ways:

- As two separate numbers: `addLine(0,0, 100,100)`
- As an array of two numbers: `addLine([0,0], [100,100])`
- As vector object: `addLine(new RVector(0,0), new RVector(100,100))`

Angles are always given in degrees, except for the start and end parameters of ellipses which are given in radians.

Entity References

Functions that work with an existing entity accept the entity in various forms:

- An **entity ID** (number), for example `1234`.
- An **entity object** as returned by the creation functions, for example the return value of `addLine(...)`.
- A **position** in the drawing (`[x,y]` or `RVector`): the entity closest to the given position is used.
- A **shape** (for example `new RLine(0,0, 100,0)`): the operation is applied to the shape only, the drawing is not changed.

```
var l = addLine(0,0, 100,0);
move(l, [0,10]);              // by entity object
move(l.getId(), [0,10]);      // by entity ID
move([50,20], [0,10]);        // by position: the entity closest to 50,20
```

Click Positions

Some operations depend on where an entity is clicked when using the equivalent tool interactively. Trimming keeps the part of the entity at the click position, rounding and beveling round the ends of the entities closest to the click positions and offsetting creates parallels on the side of the click position.

In the Simple API, a click position may follow every entity reference. If an entity is referenced by a position, that position also serves as click position. If no click position is given, the middle point of the entity is used:

```
var line = addLine(0,0, 100,0);
var limit = addLine(70,-20, 70,20);

// trim line to limit, keep the part of the line at 10,0:
trim(line, [10,0], limit);
```

Return Values

- **Creation functions** return the added entity. The entity can be passed to other functions or queried for information (for example `getId()`, `getLength()` or `getStartPoint()`).
- **Transformations** (`move`, `rotate`, `scale`, `mirror`, `lengthen`, `reverse`) return the modified entity.

- **Other modification functions** (`trim`, `round`, `bevel`, `offset`, `breakOut`, `autoTrim`, `divide`, `explode`, `stretch`) return the transaction that was applied to the drawing or `undefined` if the operation failed.
- If **shapes** are passed instead of entities, transformations modify the given shapes and all other functions return the resulting shapes as an array.

Failed operations do not raise an error. Instead, a warning is shown in the Script Shell, for example "Entity cannot be trimmed.".

Creating Entities

addPoint

```
addPoint(x, y)
addPoint([x, y])
```

Adds a point entity at the given position.

```
for (var i=0; i<10; i++) {
    addPoint(i*10, 0);
}
```

addLine

```
addLine(x1, y1, x2, y2)
addLine([x1, y1], [x2, y2])
```

Adds a line from the start point to the end point.

```
// a triangle:
addLine(0,0, 100,0);
addLine(100,0, 50,80);
addLine(50,80, 0,0);
```

addXLine

```
addXLine(x, y, dx, dy)
addXLine([x, y], [dx, dy])
```

Adds an infinite line (construction line) through the base point `x`, `y` with the direction `dx`, `dy`. The direction is given relative to the base point.

```
// horizontal and 45 degree construction lines through 0,0:
addXLine(0,0, 1,0);
addXLine([0,0], [1,1]);
```

addArc

```
addArc(cx, cy, radius, startAngle, endAngle)
addArc(cx, cy, radius, startAngle, endAngle, reversed)
addArc([cx, cy], radius, startAngle, endAngle, reversed)
```

Adds an arc with the given center, radius, start angle and end angle. Angles are in degrees. Arcs run counterclockwise from the start angle to the end angle unless `reversed` is `true` (optional, default is `false`).

```
// quarter arc from 0 to 90 degrees:
addArc(0,0, 50, 0, 90);
```

```
// clockwise arc from 0 to 90 degrees (the other three quarters):
addArc([0,0], 40, 0, 90, true);
```

addCircle

```
addCircle(cx, cy, radius)
addCircle([cx, cy], radius)
```

Adds a circle with the given center and radius.

```
// concentric circles:
for (var r=10; r<=50; r+=10) {
    addCircle(0,0, r);
}
```

addEllipse

```
addEllipse(cx, cy, mx, my, ratio, startParam, endParam, reversed)
addEllipse([cx, cy], [mx, my], ratio, startParam, endParam, reversed)
```

Adds an ellipse or elliptical arc. `mx,my` is the end point of the major axis, relative to the center. `ratio` is the ratio between the minor and the major axis. `startParam` and `endParam` are the ellipse parameters in **radians** where the ellipse starts and ends. Use `0` and `2*Math.PI` for a full ellipse.

```
// full ellipse with a major radius of 50 and a minor radius of 25:
addEllipse(0,0, 50,0, 0.5, 0, 2*Math.PI, false);

// upper half of a rotated ellipse:
addEllipse([0,0], [30,30], 0.3, 0, Math.PI, false);
```

addPolyline

```
addPolyline(points, closed, relative)
```

Adds a polyline through the given vertices. `points` is an array of vertices, each given as `[x, y]`, `[x, y, bulge]` or `[x, y, bulge, relative]`. Instead of `x, y`, an `RVector` can be used.

- `bulge` defines the shape of the segment that starts at this vertex: `0` (default) for a straight segment, `1` for a half circle counterclockwise, `-1` for a half circle clockwise. Other values create arc segments in between (the bulge is the tangent of a quarter of the arc angle).
- `relative`: `true` if the coordinates of this vertex are relative to the previous vertex.
- `closed`: `true` to close the polyline (default is `false`).
- `relative`: `true` to treat the coordinates of all vertices as relative to the previous vertex (default is `false`).

```
// closed rectangle:
addPolyline([[0,0], [100,0], [100,50], [0,50]], true);

// slot shape with two half circles:
addPolyline([[0,60], [80,60,1], [80,80], [0,80,1]], true);

// zigzag line with relative coordinates:
addPolyline([[0,100], [10,10], [10,-10], [10,10], [10,-10]], false, true);
```

addSpline

```
addSpline(fitPoints, closed)
```

Adds a spline through the given fit points. `fitPoints` is an array of `[x, y]` arrays or `RVector` objects. If `closed` is `true`, the spline is closed (periodic).

```
// open spline:
addSpline([[0,0], [30,40], [60,0], [90,40]], false);

// closed spline:
addSpline([[0,60], [40,80], [80,60], [40,100]], true);
```

The tutorial Running Scripts from a File shows another example that plots a Lissajous curve as spline.

addSimpleText

```
addSimpleText(text, x, y)
addSimpleText(text, x, y, height, angle, font, vAlign, hAlign, bold, italic)
addSimpleText(text, [x, y], height, angle, font, vAlign, hAlign, bold, italic)
```

Adds a single line text at the given position. All arguments after the position are optional:

- `height`: text height (default `1`)
- `angle`: text angle in degrees (default `0`)
- `font`: font name (default `"Standard"`)
- `vAlign`: vertical alignment: `RS.VAlignTop` (default), `RS.VAlignMiddle`, `RS.VAlignBase` or `RS.VAlignBottom`
- `hAlign`: horizontal alignment: `RS.HAlignLeft` (default), `RS.HAlignCenter` or `RS.HAlignRight`
- `bold`, `italic`: `true` for bold or italic text (TrueType fonts only)

```
addSimpleText("Hello World", 0,0);
addSimpleText("Title", [0,20], 10);
addSimpleText("Centered", [50,-20], 5, 0, "Standard", RS.VAlignMiddle, RS.HAlignCenter);
addSimpleText("Rotated", 100,0, 5, 45);
```

addSolid

```
addSolid(x1, y1, x2, y2, x3, y3, x4, y4)
addSolid([x1, y1], [x2, y2], [x3, y3], [x4, y4])
```

Adds a solid fill area with three or four corners. Note that the corners of a four sided solid are given in a zigzag order: first corner, second corner, then the corner opposite the second corner and the corner opposite the first corner. To create a triangle, pass the third corner twice.

```
// filled square:
addSolid(0,0, 50,0, 0,50, 50,50);

// filled triangle:
addSolid([60,0], [110,0], [85,50], [85,50]);
```

Adding Shapes, Entities and Objects

The following functions add objects that were created with the full QCAD API to the drawing. They are useful when the specialized creation functions above do not cover a certain case.

addShape

```
addShape(shape)
addShape(shape, color, linetype, lineweight)
```

Adds the given shape (`RLine`, `RArc`, `RCircle`, `REllipse`, `RPolyline`, `RSpline`, `RPoint`, ...) as new entity on the current layer. Optionally, a color (`RColor`), linetype (name or ID) and lineweight (for example `RLineweight.Weight050`) can be given. By default, the current color, linetype `ByLayer` and lineweight `ByLayer` are used.

```
addShape(new RLine(new RVector(0,0), new RVector(100,0)));
addShape(new RCircle(new RVector(50,0), 20), new RColor("red"));
addShape(new RLine(new RVector(0,-20), new RVector(100,-20)), new RColor("blue"), "DASHED",
 RLineweight.Weight050);
```

addShapes

```
addShapes(shapes)
```

Adds all shapes in the given array as new entities.

```
var shapes = [];
for (var i=0; i<5; i++) {
    shapes.push(new RCircle(new RVector(i*20, 0), 8));
}
addShapes(shapes);
```

addEntity

```
addEntity(entity)
addEntity(entity, color, linetype, lineweight)
```

Adds the given entity (`REntity`) to the drawing. If no color is given, the current color is used.

```
var doc = getDocument();
var data = new RLineData(new RVector(0,0), new RVector(100,50));
var entity = new RLineEntity(doc, data);
addEntity(entity);
```

addObject

```
addObject(object)
```

Adds any object (entity, layer, block, ...) to the drawing and returns the added object.

```
// add a block called "Marker" with a cross in it:
var doc = getDocument();
var block = addObject(new RBlock(doc, "Marker", new RVector(0,0)));

var l1 = new RLineEntity(doc, new RLineData(new RVector(-5,0), new RVector(5,0)));
l1.setBlockId(block.getId());
addEntity(l1);
var l2 = new RLineEntity(doc, new RLineData(new RVector(0,-5), new RVector(0,5)));
l2.setBlockId(block.getId());
addEntity(l2);

// insert the block twice:
addEntity(new RBlockReferenceEntity(doc, new RBlockReferenceData(block.getId(), new
 RVector(0,0), new RVector(1,1), 0.0)));
addEntity(new RBlockReferenceEntity(doc, new RBlockReferenceData(block.getId(), new
 RVector(50,0), new RVector(2,2), 0.0)));
```

deleteObject

```
deleteObject(object)
```

Deletes the given object from the drawing. To delete entities, the function deleteEntity is usually more convenient.

```
var line = addLine(0,0, 100,0);
deleteObject(line);
```

Layers and Attributes

addLayer

```
addLayer(name)
addLayer(name, color, linetype, lineweight)
```

Adds a new layer with the given name. Optionally, the color name (default `"white"`, which is displayed black on a white background), linetype name (default `"CONTINUOUS"`) and lineweight (default `RLineweight.Weight000`) can be given. Returns the new layer or `undefined` if a layer with the given name already exists.

```
addLayer("Construction", "gray", "DASHED");
addLayer("Contour", "blue", "CONTINUOUS", RLineweight.Weight050);
```

hasLayer

```
hasLayer(name)
```

Returns `true` if a layer with the given name exists in the drawing.

```
if (!hasLayer("Holes")) {
    addLayer("Holes", "red");
}
```

setCurrentLayer

```
setCurrentLayer(name)
```

Sets the current layer. All entities added afterwards are placed on this layer.

```
addLayer("Holes", "red");
setCurrentLayer("Holes");
addCircle(0,0, 5);
addCircle(50,0, 5);
setCurrentLayer("0");
```

setCurrentColor

```
setCurrentColor(color)
setCurrentColor(r, g, b)
```

Sets the color for entities that are added afterwards. The color can be given as `RColor` object or as red, green, blue values (0-255). Use `new RColor(RColor.ByLayer)` to reset the color to "By Layer".

```
setCurrentColor(255, 0, 0);
addCircle(0,0, 10);
setCurrentColor(new RColor("blue"));
addCircle(30,0, 10);
setCurrentColor(new RColor(RColor.ByLayer));
addCircle(60,0, 10);
```

Querying Entities

getEntity

```
getEntity(id)
getEntity(entity)
getEntity(x, y)
getEntity([x, y])
```

Returns the entity for the given entity reference, i.e. the entity with the given ID, the current state of the given entity or the entity closest to the given position. Returns `undefined` if no entity is found.

```
var circle = addCircle(0,0, 20);
move(circle, [10,0]);

// circle still holds the old center, getEntity returns the current state:
getEntity(circle).getCenter();     // returns 10,0

// entity closest to 30,0:
getEntity(30,0).getRadius();       // returns 20
```

getShape

```
getShape(id)
getShape(entity)
getShape(x, y)
getShape([x, y])
```

Returns the geometry of the given entity as shape (`RLine` for a line, `RArc` for an arc, `RPolyline` for a polyline, ...). Shapes can be used for geometric calculations without changing the drawing.

```
addPolyline([[0,0], [100,0], [100,50]], false);

var pl = getShape(50,0);
pl.getLength();                    // returns 150
```

getClosestEntity

```
getClosestEntity(x, y)
getClosestEntity([x, y], range)
```

Returns the entity closest to the given position. If `range` is given, only entities within that distance of the position are considered. Returns `undefined` if no entity is found.

```
addLine(0,0, 100,0);
addLine(0,50, 100,50);

getClosestEntity(50,10).getStartPoint();     // returns 0,0
getClosestEntity([50,25], 5);               // returns undefined
```

getEntityShape

```
getEntityShape(entity, position)
```

Returns the shape of the given entity object. For complex entities that consist of multiple shapes (block references, hatches, dimensions, texts), the shape closest to the given position is returned. The position is optional.

```
// a block reference of a block with a rectangle:
var doc = getDocument();
```

```
var block = addObject(new RBlock(doc, "Box", new RVector(0,0)));
var pl = new RPolylineEntity(doc, new RPolylineData());
pl.setShape(new RPolyline([new RVector(0,0), new RVector(40,0), new RVector(40,20), new
 RVector(0,20)], true));
pl.setBlockId(block.getId());
addEntity(pl);
var ref = addEntity(new RBlockReferenceEntity(doc, new RBlockReferenceData(block.getId(),
 new RVector(0,0), new RVector(1,1), 0.0)));

// shape of the block reference near 40,10 (the rectangle):
var s = getEntityShape(getEntity(ref), new RVector(41,10));
s.getLength();                    // returns 120
```

getIntersectionPoints

```
getIntersectionPoints(entity1, entity2)
getIntersectionPoints(entity1, entity2, limited)
```

Returns an array with the intersection points of the two given entities or shapes. Entities can be given as entity objects, entity IDs or shapes. If `limited` is `false`, the entities are treated as if they were infinite (lines extended, arcs as full circles). Default is `true`.

```
var line = addLine(-50,0, 50,0);
var circle = addCircle(0,0, 30);
var ips = getIntersectionPoints(line, circle);

// mark the intersections with small circles:
for (var i=0; i<ips.length; i++) {
    addCircle(ips[i], 2);
}
```

isPosition and toVector

```
isPosition(value)
toVector(value)
```

`isPosition` returns `true` if the given value is a position (`RVector` or array of two or three numbers). `toVector` converts a position given as array to an `RVector`. These functions are useful when writing your own functions that accept positions in the same way as the Simple API.

```
function addCross(pos, size) {
    var p = toVector(pos);
    addLine(p.x-size, p.y, p.x+size, p.y);
    addLine(p.x, p.y-size, p.x, p.y+size);
}

addCross([0,0], 5);
addCross(new RVector(20,0), 5);
isPosition([1,2]);                // returns true
isPosition("abc");                // returns false
```

Modifying Entities

Transformations

move

```
move(entity, dx, dy)
move(entity, [dx, dy])
```

Moves the given entity by the given offset.

```
var c = addCircle(0,0, 10);
move(c, 50,0);
```

rotate

```
rotate(entity, angle)
rotate(entity, angle, cx, cy)
rotate(entity, angle, [cx, cy])
```

Rotates the given entity by the given angle in degrees around the given center. If no center is given, the entity is rotated around the origin `0,0`.

```
// a fan of lines, rotated around 0,0:
for (var a=0; a<360; a+=30) {
    var l = addLine(10,0, 50,0);
    rotate(l, a);
}

// rotate a square around its center:
var sq = addPolyline([[100,0], [120,0], [120,20], [100,20]], true);
rotate(sq, 45, [110,10]);
```

scale

```
scale(entity, factor)
scale(entity, factor, cx, cy)
scale(entity, factor, [cx, cy])
scale(entity, [sx, sy], [cx, cy])
```

Scales the given entity by the given factor. The focus point (the point that remains in place) defaults to the origin `0,0`. Different factors for X and Y can be given as array `[sx, sy]`. Note that some entities such as circles and arcs cannot be scaled by different factors in X and Y.

```
var sq = addPolyline([[0,0], [10,0], [10,10], [0,10]], true);
scale(sq, 3);

var pl = addPolyline([[50,0], [60,0], [60,10], [50,10]], true);
scale(pl, [2,1], [50,0]);
```

mirror

```
mirror(entity, x1, y1, x2, y2)
mirror(entity, [x1, y1], [x2, y2])
mirror(entity, new RLine(...))
```

Mirrors the given entity at the axis defined by two points.

```
var pl = addPolyline([[10,0], [40,0], [40,30]], false);
// mirror at the Y axis:
mirror(pl, 0,0, 0,100);
```

To keep the original and create a mirrored copy, add a copy of the entity first:

```
var arc = addArc(20,0, 10, 0, 180);
var copy = addShape(getShape(arc));
mirror(copy, [0,-10], [0,10]);
```

Trimming and Lengthening

trim

```
trim(trimEntity, limitingEntity)
trim(trimEntity, trimPos, limitingEntity, limitingPos)
trim(trimEntity, trimPos, limitingEntity, limitingPos, trimBoth)
trim(trimPos, limitingPos)
```

Trims the entity `trimEntity` to the limiting entity `limitingEntity`. The part of the trimmed entity at the click position `trimPos` is kept. If the two entities do not intersect, the entity is extended to the limiting entity. If there are multiple intersections, the one closest to the click position `limitingPos` is used. If `trimBoth` is `true`, both entities are trimmed to each other.

```
var line = addLine(0,0, 100,0);
var limit = addLine(70,-20, 70,20);

// keep the left part, line ends now at 70,0:
trim(line, [10,0], limit);
```

```
// extend a line to a circle:
var line = addLine(0,0, 50,0);
var circle = addCircle(100,0, 20);

// extend to the intersection with the circle closest to 80,0:
trim(line, [25,0], circle, [80,0]);      // line ends now at 80,0
```

trimBoth

```
trimBoth(entity1, entity2)
trimBoth(entity1, pos1, entity2, pos2)
trimBoth(pos1, pos2)
```

Trims or extends both entities to their intersection point. This is a shortcut for `trim` with `trimBoth` set to `true`.

```
// two lines that do not touch are extended to form a corner at 100,0:
var a = addLine(0,0, 80,0);
var b = addLine(100,20, 100,100);
trimBoth(a, b);
```

lengthen

```
lengthen(entity, amount)
lengthen(entity, start, amount)
lengthen(entity, pos, amount)
lengthen(pos, amount)
```

Lengthens the given entity by the given amount. Negative amounts shorten the entity. By default, the entity is lengthened at its end point. Pass `true` for `start` to lengthen it at its start point instead or pass a position close to the end that should be lengthened.

```
var line = addLine(0,0, 100,0);
lengthen(line, 20);              // line ends now at 120,0
lengthen(line, true, -10);       // line starts now at 10,0

var arc = addArc(0,50, 30, 0, 90);
lengthen(arc, [30,50], 10);      // lengthen the arc at its start point
```

breakOut

```
breakOut(entity, pos)
breakOut(entity, pos, removeSegment)
breakOut(pos)
```

Breaks out the segment of the given entity at the given position. The segment ends at the closest intersections with other entities on either side of the position. If there is no intersection on one side, the segment extends to the end of the entity. If `removeSegment` is `true` (default), the segment is deleted. Otherwise it is kept as a separate entity.

```
var line = addLine(0,0, 100,0);
addLine(30,-10, 30,10);
addLine(70,-10, 70,10);

// remove the part of the line between 30,0 and 70,0:
breakOut(line, [50,0]);
```

autoTrim

```
autoTrim(entity, pos)
autoTrim(pos)
```

The opposite of `breakOut`: keeps only the segment of the given entity at the given position and removes the rest. The segment ends at the closest intersections with other entities on either side of the position.

```
var line = addLine(0,0, 100,0);
addLine(30,-10, 30,10);
addLine(70,-10, 70,10);

// keep only the part between 30,0 and 70,0:
autoTrim(line, [50,0]);
```

divide

```
divide(entity, pos)
divide(entity, pos1, pos2)
divide(pos)
```

Divides the given entity at the given position. An open entity (line, arc, open polyline, ...) is divided into two parts at one position or three parts at two positions. Closed entities (circles, full ellipses, closed polylines) require two positions and are divided into two parts.

```
var line = addLine(0,0, 100,0);
divide(line, [40,0]);            // two lines: 0..40 and 40..100

var circle = addCircle(0,50, 20);
divide(circle, [20,50], [-20,50]);   // two half circles
```

Corners

round

```
round(entity1, entity2, radius)
round(entity1, pos1, entity2, pos2, radius)
round(entity1, pos1, entity2, pos2, radius, trim)
round(pos1, pos2, radius)
```

Rounds the corner between two entities with an arc of the given radius (fillet). The click positions indicate which ends of the entities are rounded. If `trim` is `true` (default), both entities are trimmed to the rounding arc.

```
// a rectangle with rounded corners:
var l1 = addLine(0,0, 100,0);
var l2 = addLine(100,0, 100,50);
var l3 = addLine(100,50, 0,50);
var l4 = addLine(0,50, 0,0);

round(l1, l2, 10);
round(l2, l3, 10);
round(l3, l4, 10);
round(l4, l1, 10);
```

The corners of a polyline can be rounded too, by passing the same polyline twice with a click position on each of the two segments:

```
var pl = addPolyline([[0,0], [100,0], [100,50]], false);
round(pl, [50,0], pl, [100,25], 10);
```

bevel and chamfer

```
bevel(entity1, entity2, distance)
bevel(entity1, entity2, distance1, distance2)
bevel(entity1, pos1, entity2, pos2, distance1, distance2)
bevel(entity1, pos1, entity2, pos2, distance1, distance2, trim)
bevel(pos1, pos2, distance)
```

Bevels the corner between two entities with a straight line (chamfer). `distance1` is the distance of the bevel line from the corner along the first entity, `distance2` along the second entity (defaults to `distance1`). If `trim` is `true` (default), both entities are trimmed to the bevel line. `chamfer` is an alias for `bevel`.

```
var a = addLine(0,0, 100,0);
var b = addLine(100,0, 100,50);
bevel(a, b, 10);

var c = addLine(0,100, 100,100);
var d = addLine(100,100, 100,150);
chamfer(c, d, 20, 10);
```

offset

```
offset(entity, distance)
offset(entity, distance, number)
offset(entity, distance, number, side)
offset(entity, sidePos, distance)
offset(entity, sidePos, distance, number)
```

Creates parallels to lines, concentric arcs and circles or offset curves of ellipses and other entities. `number` is the number of parallels to create (default `1`). The parallels are created on both sides of the entity unless a side is given, either as position `sidePos` next to the entity or as `side` (`RS.LeftHand`, `RS.RightHand` or `RS.BothSides`). Offsetting polylines and splines is only available in QCAD Professional.

```
var line = addLine(0,0, 100,0);
offset(line, 5);                 // one parallel on each side
```

```
var line2 = addLine(0,30, 100,30);
offset(line2, 5, 2, RS.LeftHand);    // two parallels on the left side (above)

var circle = addCircle(50,80, 10);
offset(circle, [50,95], 3, 4);   // four concentric circles outside
```

reverse

```
reverse(entity)
reverse(x, y)
```

Reverses the direction of the given entity, i.e. swaps its start and end point.

```
var line = addLine(0,0, 100,0);
line = reverse(line);
line.getStartPoint();            // returns 100,0
```

explode

```
explode(entity)
explode(entity, options)
explode(x, y)
```

Explodes the given entity into simpler entities: polylines into lines and arcs, block references into their contents, texts into lines and splines, dimensions and hatches into lines, arcs, etc. The original entity is deleted.

`options` is an optional object with settings that are otherwise taken from the application preferences, for example `{ splineSegments: 64, textToPolylines: true }`. Supported options are: `splineTolerance`, `splineSegments`, `ellipseSegments`, `splinesToLineSegments`, `textToPolylines`, `textSplineToLineOrArc`, `multilineTextToSimpleText` and `circlesToPolylines`.

```
var pl = addPolyline([[0,0], [100,0,1], [100,50], [0,50]], true);
explode(pl);                     // three lines and one arc
```

stretch

```
stretch(x1, y1, x2, y2, dx, dy)
stretch([x1, y1], [x2, y2], [dx, dy])
stretch(area, offset)
```

Stretches all entities in the rectangular area defined by two corners by the given offset. Entities that lie completely inside the area are moved, entities that cross the boundary of the area are stretched (only the end points inside the area are moved). The area can also be given as `RBox` or as closed `RPolyline` shape. If entities are selected in the drawing, only selected entities are stretched.

```
// a rectangle made of lines:
addLine(0,0, 100,0);
addLine(100,0, 100,50);
addLine(100,50, 0,50);
addLine(0,50, 0,0);

// make the rectangle 50 units wider by stretching its right side:
stretch([80,-10], [120,60], [50,0]);
```

deleteEntity

```
deleteEntity(entity)
deleteEntity(id)
deleteEntity(x, y)
```

Deletes the given entity from the drawing.

```
var l1 = addLine(0,0, 100,0);
var l2 = addLine(0,10, 100,10);
addLine(0,20, 100,20);

deleteEntity(l1);                // by entity
deleteEntity(l2.getId());        // by entity ID
deleteEntity(50,20);             // by position
```

paste

```
paste(sourceDocument, destination, offset, scale, rotation,
      flipHorizontal, flipVertical, toCurrentLayer, overwriteLayers, overwriteBlocks)
```

Pastes the contents of the document `sourceDocument` into the document interface `destination` or into the current drawing if `destination` is `undefined`. The pasted contents are moved by `offset`, scaled by `scale` and rotated by `rotation` degrees. All arguments are required:

- `flipHorizontal`, `flipVertical`: `true` to flip the contents
- `toCurrentLayer`: `true` to paste all entities onto the current layer
- `overwriteLayers`, `overwriteBlocks`: `true` to overwrite existing layers or blocks with the same name

```
// create a symbol in an off-screen document:
var symbol = createDocument();
startTransaction(symbol);
addCircle(0,0, 10);
addLine(-10,0, 10,0);
addLine(0,-10, 0,10);
endTransaction();

// paste it three times into the current drawing:
for (var i=0; i<3; i++) {
    paste(symbol, undefined, [i*50, 0], 1+i*0.5, i*15, false, false, false, false, false);
}
```

Transactions

startTransaction and endTransaction

```
startTransaction()
startTransaction(document)
endTransaction()
```

By default, every Simple API call is applied to the drawing immediately and can be undone individually. Between `startTransaction()` and `endTransaction()`, all changes are collected and applied together when `endTransaction` is called. This has two advantages:

- It is much faster when adding many entities.
- All changes can be undone in one single step.

If a document (`RDocument`) or document interface (`RDocumentInterface`) is passed to `startTransaction`, all changes are applied to that document instead of the current drawing. This is useful to create drawings in the background, for example to export them to a file (see createDocumentInterface).

```
startTransaction();
for (var x=0; x<=200; x+=10) {
    for (var y=0; y<=200; y+=10) {
        addPoint(x, y);
    }
}
endTransaction();
```

Note the following when working with transactions:

- Within a transaction, the changes are not yet part of the drawing. Modification functions such as `trim` or `round` cannot refer to entities added in the same transaction.
- Entities returned by creation functions within a transaction do not have a valid ID, not even after `endTransaction`. To modify such entities later on, refer to them by position.
- All entities added within a transaction are placed on the layer that is current when `endTransaction` is called. Set the current layer before starting the transaction and use separate transactions for different layers.

```
startTransaction();
addLine(0,0, 100,0);
addLine(100,0, 100,50);
endTransaction();

// refer to the new lines by position to round the corner:
round([50,0], [100,25], 10);
```

getTransactionDocument and getTransactionDocumentInterface

```
getTransactionDocument()
getTransactionDocumentInterface()
```

Return the document or document interface that the current transaction applies to. Outside of transactions, the current document and document interface are returned.

```
var doc = createDocument();
startTransaction(doc);
addLine(0,0, 10,0);
getTransactionDocument() === doc;    // returns true
endTransaction();
```

getOperation

```
getOperation()
```

Returns the operation (`RAddObjectsOperation`) of the current transaction. Outside of a transaction, a new operation is returned that has to be applied explicitly. This is meant for advanced use, for example to combine Simple API calls with the full QCAD API.

```
startTransaction();
addCircle(0,0, 10);
var doc = getTransactionDocument();
var e = new RLineEntity(doc, new RLineData(new RVector(-10,0), new RVector(10,0)));
getOperation().addObject(e);
```

```
endTransaction();
```

View

autoZoom

```
autoZoom()
```

Zooms the current view so that the whole drawing is visible.

```
addCircle(1000,1000, 500);
autoZoom();
```

zoomTo

```
zoomTo(entity)
zoomTo(entity, margin)
zoomTo([x1, y1], [x2, y2])
zoomTo([x1, y1], [x2, y2], margin)
zoomTo(box, margin)
```

Zooms to the given entity, shape or rectangular area. `margin` is the margin around the area in pixels (default `10`).

```
addCircle(0,0, 100);
var small = addCircle(80,0, 2);
zoomTo(small, 50);
```

```
zoomTo([0,0], [100,100]);
```

Documents and Application

getMainWindow

```
getMainWindow()
```

Returns the QCAD application window (`RMainWindowQt`).

```
getMainWindow().windowTitle;
```

getDocumentInterface

```
getDocumentInterface()
```

Returns the document interface (`RDocumentInterface`) of the current drawing. The document interface connects a document with its views and provides functions to import and export files, apply operations, undo and redo.

```
addLine(0,0, 100,0);
getDocumentInterface().undo();       // removes the line again
```

getDocument

```
getDocument()
```

Returns the current document (`RDocument`). The document holds all entities, layers, blocks, etc. of a drawing and provides functions to query them.

```
addLine(0,0, 100,0);
addCircle(0,0, 10);
getDocument().queryAllEntities().length;     // number of entities in the drawing
```

getGraphicsView

```
getGraphicsView()
```

Returns the current or last active view (`RGraphicsView`) of the current drawing.

```
addCircle(0,0, 50);
getGraphicsView().autoZoom();
```

createDocument

```
createDocument()
```

Creates a new, empty document in memory that is not shown in a window. Such an off-screen document can be filled using a transaction and then pasted into a drawing (see paste).

```
var doc = createDocument();
startTransaction(doc);
addLine(0,0, 100,0);
addLine(100,0, 100,100);
endTransaction();

doc.queryAllEntities().length;       // returns 2
```

createDocumentInterface

```
createDocumentInterface()
```

Creates a new, empty document in memory together with a document interface. The document interface can be used to import or export files.

```
var di = createDocumentInterface();
startTransaction(di);
addCircle(0,0, 50);
addSimpleText("Created by script", [-40,-60], 5);
endTransaction();

di.exportFile(QDir.homePath() + "/simple_api_example.dxf", "R27 (2013) DXF");
```

disableInput, enableInput and isInputEnabled

```
disableInput()
enableInput()
isInputEnabled()
```

`disableInput` disables the QCAD application window to prevent user input, for example during long operations or animations. `enableInput` enables the application window again and `isInputEnabled` returns `true` if user input is currently enabled. Always make sure to enable input again at the end of your script.

update

```
update()
```

Processes pending events so that the user interface is updated during long operations. User input must be disabled with `disableInput` before calling `update`.

sleep

```
sleep(milliseconds)
```

Pauses the script for the given time in milliseconds while keeping the user interface up to date. User input must be disabled with `disableInput` before calling `sleep`.

The following example uses `disableInput`, `sleep` and `enableInput` to show a simple animation:

```
disableInput();
var c = addCircle(0,0, 10);
autoZoom();
for (var i=0; i<20; i++) {
    c = move(c, 5,0);
    sleep(50);
}
enableInput();
```

warning

```
warning(message)
```

Shows a warning message. In the Script Shell, warnings are shown in the command history. Otherwise, warnings are printed to the console.

```
if (getDocument().queryAllEntities().length===0) {
    warning("The drawing is empty.");
}
```

User Input

The following functions show a simple dialog to ask the user for a value. If the user cancels the dialog, `undefined` is returned.

getInt

```
getInt(title, prompt, value, step, min, max)
```

Asks the user for an integer number. `value` is the initial value (default `0`), `step` the step of the spin box (default `1`), `min` and `max` the allowed range.

```
var n = getInt("Circles", "Number of circles:", 5, 1, 1, 100);
if (isNull(n)) {
    warning("Canceled.");
}
else {
    for (var i=1; i<=n; i++) {
        addCircle(0,0, i*10);
    }
}
```

getDouble

```
getDouble(title, prompt, value, decimals, min, max)
```

Asks the user for a number. `value` is the initial value (default `0`), `decimals` the number of decimals shown (default `1`), `min` and `max` the allowed range.

```
var r = getDouble("Circle", "Radius:", 25.0, 2, 0.01, 1000);
if (!isNull(r)) {
    addCircle(0,0, r);
}
```

getText

```
getText(title, prompt, text)
```

Asks the user for a text. `text` is the initial text (default empty).

```
var t = getText("Label", "Text:", "Hello");
if (!isNull(t)) {
    addSimpleText(t, 0,0, 10);
}
```

getItem

```
getItem(title, prompt, items, index, separator)
```

Lets the user choose an item from a drop down list. `items` is a string with all items separated by `separator` (default `","`). `index` is the index of the initially selected item (default `0`). Returns the chosen item or `undefined` if the dialog was canceled.

```
var shape = getItem("Shape", "Shape to add:", "Circle,Square,Triangle", 0);
if (shape==="Circle") {
    addCircle(0,0, 50);
}
else if (shape==="Square") {
    addPolyline([[-50,-50], [50,-50], [50,50], [-50,50]], true);
}
else if (shape==="Triangle") {
    addPolyline([[-50,-50], [50,-50], [0,50]], true);
}
```

Example: A Flange

This example combines several functions of the Simple API to draw a flange with a pitch circle, bolt holes on a separate layer and a rounded rectangular cut-out:

```
addLayer("Center Lines", "gray", "CENTER");
addLayer("Holes", "red");

// outline:
setCurrentLayer("0");
addCircle(0,0, 100);

// pitch circle and center lines:
setCurrentLayer("Center Lines");
addCircle(0,0, 75);
addLine(-110,0, 110,0);
addLine(0,-110, 0,110);

// bolt holes, added in one transaction:
setCurrentLayer("Holes");
startTransaction();
for (var a=0; a<360; a+=45) {
    var angle = a * Math.PI / 180;
    addCircle(75*Math.cos(angle), 75*Math.sin(angle), 8);
}
```

```
endTransaction();

// rounded square cut-out in the center:
setCurrentLayer("0");
var l1 = addLine(-30,-30, 30,-30);
var l2 = addLine(30,-30, 30,30);
var l3 = addLine(30,30, -30,30);
var l4 = addLine(-30,30, -30,-30);
round(l1, l2, 8);
round(l2, l3, 8);
round(l3, l4, 8);
round(l4, l1, 8);

autoZoom();
```

The lines of the cut-out are added without transaction, so that they can be passed directly to `round`. The bolt holes are added in a transaction, which is faster and makes them undoable in one step.

16.2.6 The Script Shell

Introduction

QCAD comes with a very powerful and complete script interface, allowing you to access almost the entire QCAD API as well as the Qt API through ECMAScript (JavaScript). A good way to experiment with scripts is the QCAD Script Shell.

Launching the Script Shell

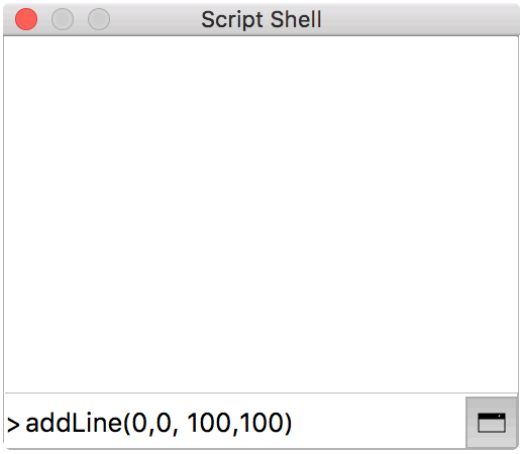
The script shell can be shown using menu *Misc* > *Development* > *Script Shell*.

The script shell shows up as a widget with a command line at the bottom and a command line history that shows previously entered commands and their output.

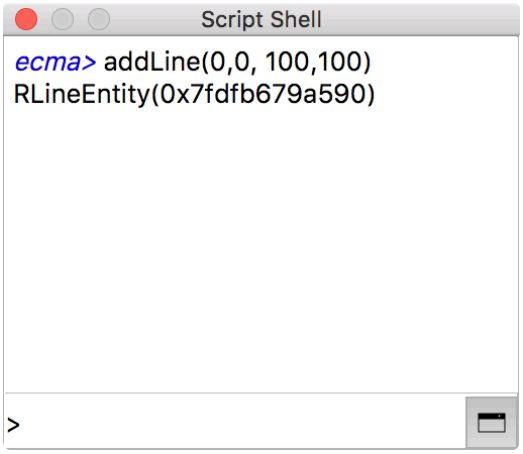
Running Commands

Call the function to add a line by entering this into the command line at the bottom:

```
addLine(0,0, 100,100)
```



Press the Enter key to submit the command.



QCAD adds a line from 0,0 to 100,100 to the current drawing.

The function addLine is part of the QCAD Simple API, a simplified way to access a limited part of the QCAD API. You can find a list of all QCAD Simple API functions at the QCAD Simple API documentation page.

Scripts on Multiple Lines

Loops (if, for, while, ...) are usually entered on multiple lines. You can do this in the same way in the script console by entering these lines, one by one and hitting enter:

```
for (i=0; i<=10; i++) {
    addLine(i*10,0, i*10,100);
}
```

Indentations are not required.

This script adds 11 parallel lines from x=0 to x=100 with a distance of 10 units.

Copy / Pasting Scripts

More complex scripts with multiple lines can be written into an external text editor and then copy / pasted into the script console. The script is executed as soon as it is pasted.

Repeating the Last Used Command

Use the up arrow and down arrow keys to browse through previously entered commands and modify them or launch them again.

Clearing the Command Line History

Use Ctrl-L (⌘L) to clear the command line history.

16.3 Drawing

This section collects techniques for creating and manipulating geometry in QCAD beyond the basic drawing tools. The tutorials show how to work with lists of coordinates, reuse existing positions from a drawing as input for other tools, import labelled points from CSV files, morph one shape into another, and apply custom lineweights that go beyond the standard DXF/DWG set. Together they cover practical workflows for getting precise data into your drawing and transforming it efficiently.

16.3.1 Custom Lineweights

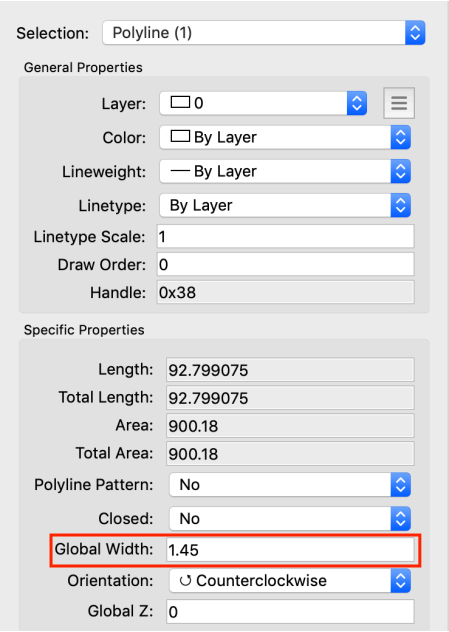
Introduction

QCAD supports a fixed set of lineweights. The lineweights available might seem somewhat arbitrary but are defined in the DXF/DWG standard. Being able to add other, custom lineweights in QCAD would break compatibility with any other DXF/DWG compatible CAD software.

However, there is a possibility to get around this limitation: custom segment widths for polylines. To make use of this feature, make sure that your entity is a polyline. Almost all entities can be easily converted to polylines, for example using *Draw* > *Polyline* > *Create from Segments* for lines or arcs or using *Modify* > *Explode* for splines or ellipses.

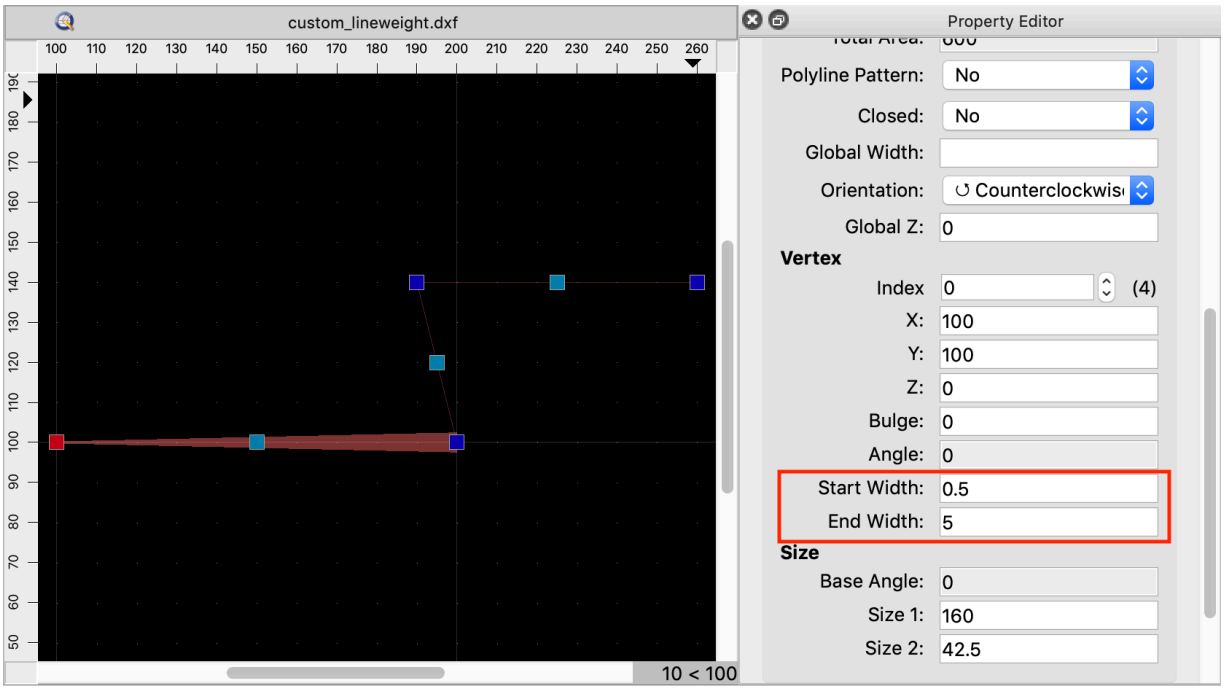
Global Width of a Polyline

You can select one or multiple polylines and use the property editor (*View* > *Property Editor*) to assign any custom lineweight (*global width*):

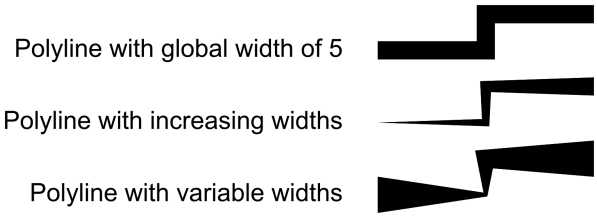


Widths per Segment

Alternatively, you can assign custom lineweights to the start or end of an individual segment of the polyline (properties *start width* and *end width*). In this example, the start width of the first segment was set to 0.5 and the end width to 5:



Examples



16.3.2 Importing Points with Labels from CSV

Introduction

QCAD offers a powerful tool to import point data with labels from CSV files. CSV files are simple text based spreadsheet files. CSV stands for "comma-separated values".

This tool can be accessed from menu *Misc* > *Import/Export* > *Import Points with Labels*.

Example Data

For this tutorial, we use the following table as example data:

N	X	Y	Z	C
P1	127.6	100.5	50	Location A
P2	120	134	48.77	Location B
P3	125	123.5	37.6	Location C
P4	175	80.75	12.6	Location D

The raw contents of the CSV file looks like this:

```
N,X,Y,Z,C
P1,127.6,100.5,50,Location A
P2,120,134,48.77,Location B
P3,125,123.5,37.6,Location C
```

```
P4,175,80.75,12.6,Location D
```

Preparation

Before importing points, make sure that the active layer of your drawing is the layer on which you want the points to be.

Prepare all layers you will need for the imported labels if you are planning to put the labels on separate layers.

Example 1: Points with a Single Label

For this first example, we want to import all points in the table with columns X and Y as point coordinates. Column 'N' will be used as a label at the right side beside each point. Each row in the table is imported as a point entity and a text entity as shown here:



After starting the tool from menu *Misc* > *Import/Export* > *Import Points with Labels*, you first have to choose which CSV file you want to import.

Next, a dialog is shown which lets you configure how exactly points and their labels should be rendered and formatted. You also need to tell QCAD which columns contain the coordinates of the point.

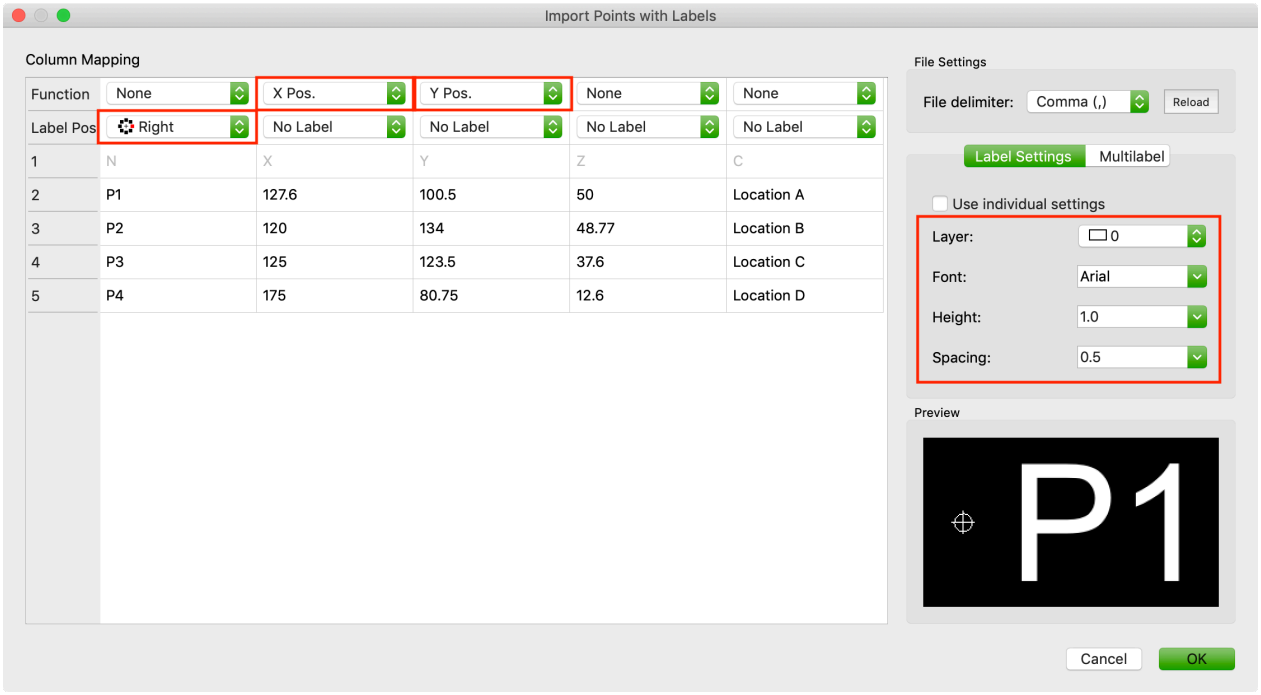
For this example, we only have to adjust or check three fields (the ones marked in red in the screenshot below):

- For the first column, we choose that these values are to be displayed as labels to the right of the point (Label Pos: "Right").
- For the second column, we choose the function "X Pos." as this column contains the X coordinate values of the points.
- For the third column we choose the function "Y Pos.". QCAD tries to detect the X and Y columns automatically, based on the column header names.

Note:

- The fourth and fifth columns with the Z coordinate and the comment are ignored for this example (no function and no label).
- At the right side under "Label Settings", we can adjust the font ("Arial"), the font size (here 1 drawing unit) and the spacing between the point and the text (here 0.5 drawing units). We can also choose which layer the labels are to be placed on.
- In the bottom right corner, we can see a preview of the first data point as it will be imported into the drawing.

Click OK to import all points of the CSV file with their labels.



Example 2: Point with Three Labels

This example shows how multiple labels can be arranged around the imported points.

For this example, we want to show the Z coordinate of the points at the left of the point and the name and comment at the right, below each other:



We can achieve this configuration by setting the label positions as follows:

Function	None	X Pos.	Y Pos.	None	None
Label Pos	Top Right	No Label	No Label	Left	Bottom Right
1	N	X	Y	Z	C
2	P1	127.6	100.5	50	Location A
3	P2	120	134	48.77	Location B
4	P3	125	123.5	37.6	Location C
5	P4	175	80.75	12.6	Location D

Example 3: Displaying Coordinates as Labels

A column can be used as a coordinate as well as a label. In this example, we display the X, Y and Z coordinates at the right of the point, below each other:



Note that we did not assign the function for the Z position, so the point is still placed at X/Y with Z set to 0.0.

Function	None	X Pos.	Y Pos.	None	None
Label Pos	No Label	Top Right	Right	Bottom Right	No Label
1	N	X	Y	Z	C
2	P1	127.6	100.5	50	Location A
3	P2	120	134	48.77	Location B
4	P3	125	123.5	37.6	Location C
5	P4	175	80.75	12.6	Location D

Example 4: Multilabels with Separators

It's also possible to assign multiple columns to the same label position. In this case, QCAD concatenates all values and displays them at the desired location. Here we show the name and the Z coordinate at the same label position ("Right"):

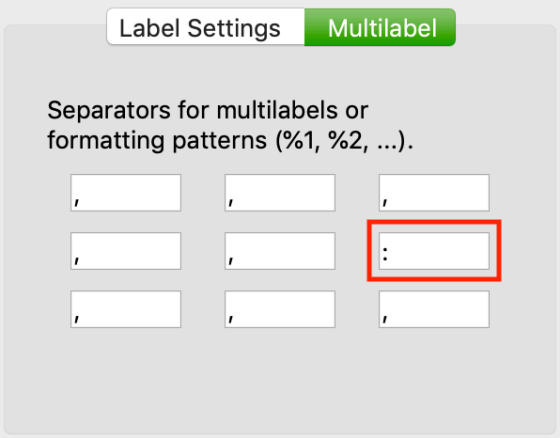


We can achieve this by simply assigning the same label position to multiple columns.

Function	None	X Pos.	Y Pos.	None	None
Label Pos	Right	No Label	No Label	Right	No Label
1	N	X	Y	Z	C
2	P1	127.6	100.5	50	Location A
3	P2	120	134	48.77	Location B
4	P3	125	123.5	37.6	Location C
5	P4	175	80.75	12.6	Location D

You can assign as many columns as desired to the same label position. Labels are concatenated by a comma and a space by default.

The separators used for the various label positions can be adjusted under the Multilabel tab:

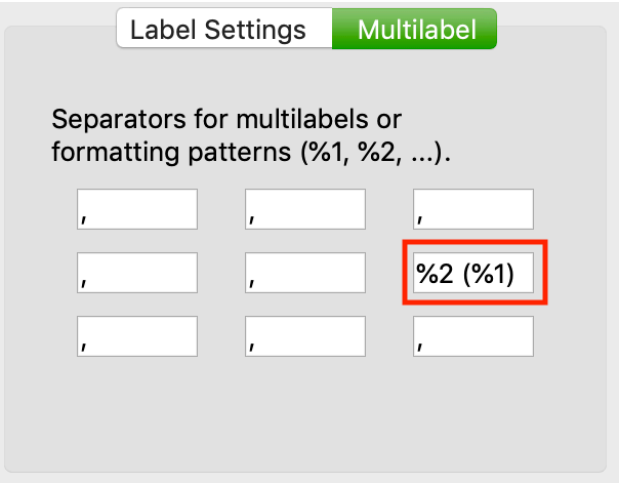


We can for example change the default separator for the label position at the right with a colon followed by a space (": "), leading to a label formatting as shown here:



Example 5: Multilabels with Formatting Pattern

It's also possible to format labels in more advanced ways. To do this, you can enter a formatting pattern in the separator input fields. "%1" is then replaced by the first value of the multilabel, "%2" with the second value, etc. In this example, we use the formatting pattern "%2 (%1)":



Note that in this case, the second value (Z) is displayed first, followed by the first value in brackets:



Z Coordinates

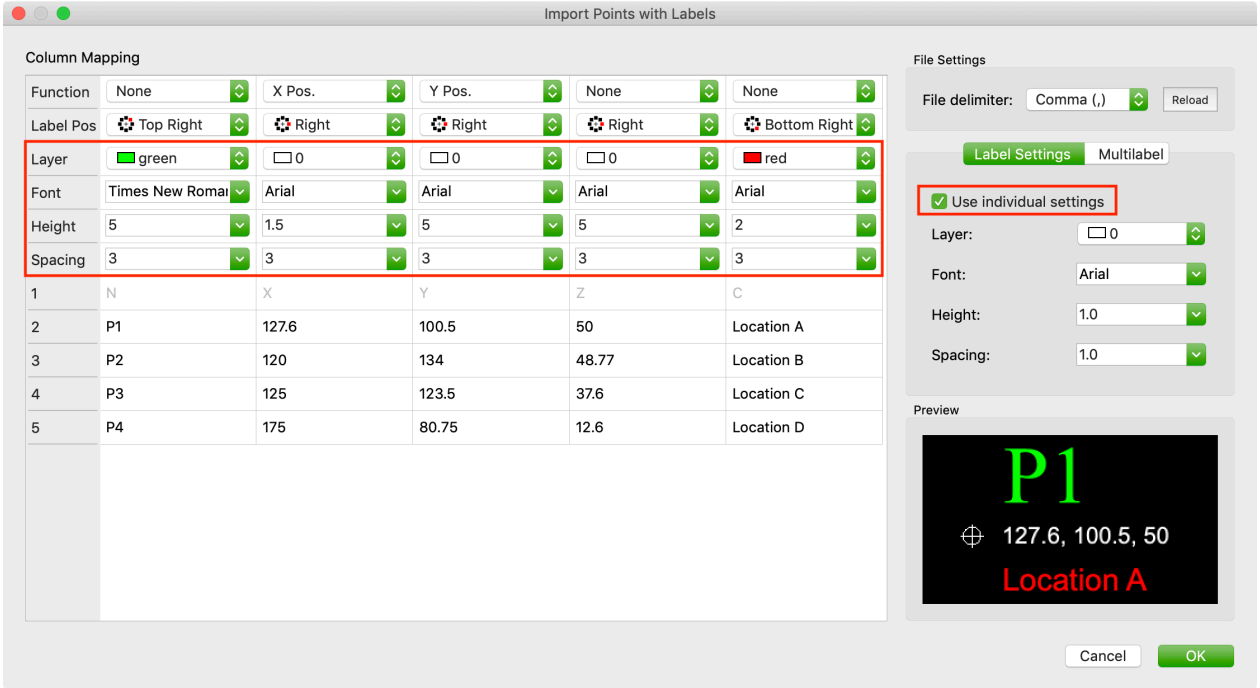
Even though QCAD is a 2D CAD application, you can import Z coordinates for points. These Z coordinates can be displayed by the property editor in QCAD and will also be saved to DXF/DWG. To do this, simply assign the function "Z Pos." for the appropriate column:

Function	None	X Pos.	Y Pos.	Z Pos.	None
Label Pos	Right	No Label	No Label	No Label	No Label
1	N	X	Y	Z	C
2	P1	127.6	100.5	50	Location A
3	P2	120	134	48.77	Location B
4	P3	125	123.5	37.6	Location C
5	P4	175	80.75	12.6	Location D

Labels with Individual Settings

So far, we've used the same layer, font, text height and spacing for all labels. It's also possible to control these aspects for each label position individually. To do this, check the option "Use individual settings" under "Label Settings".

If this option is enabled, additional controls are displayed for each column. With these controls, we can define the layer, font, text height and spacing for each label position individually. This allows us to place different labels on different layers and use a different font for each label as shown in this example:



File Delimiters

Fields in CSV files are typically separated by a comma. However, there are also CSV files which use a tab or a semicolon as delimiter. You can adjust the file delimiter in the dialog at the top right and reload the file if you are dealing with such a file:



16.3.3 Morphing

Introduction

Morphing in QCAD means to transform a polyline into another polyline in a given number of steps.

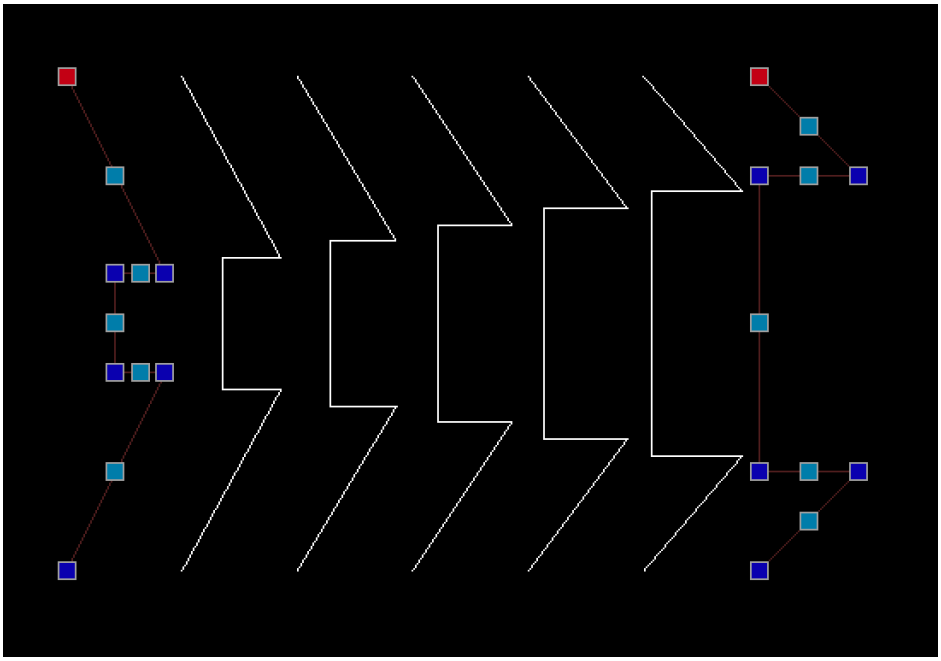
Step 1: Preparation

The source and target shape of the morphing operation need to be polylines. You can convert circles, ellipses and splines into polylines using *Modify* > *Explode*. Connected segments of lines, arcs, ellipse arcs, splines and open polylines can be joined into single polylines by right-clicking on a segment and choosing *Auto Create Polyline*.

Open Polylines

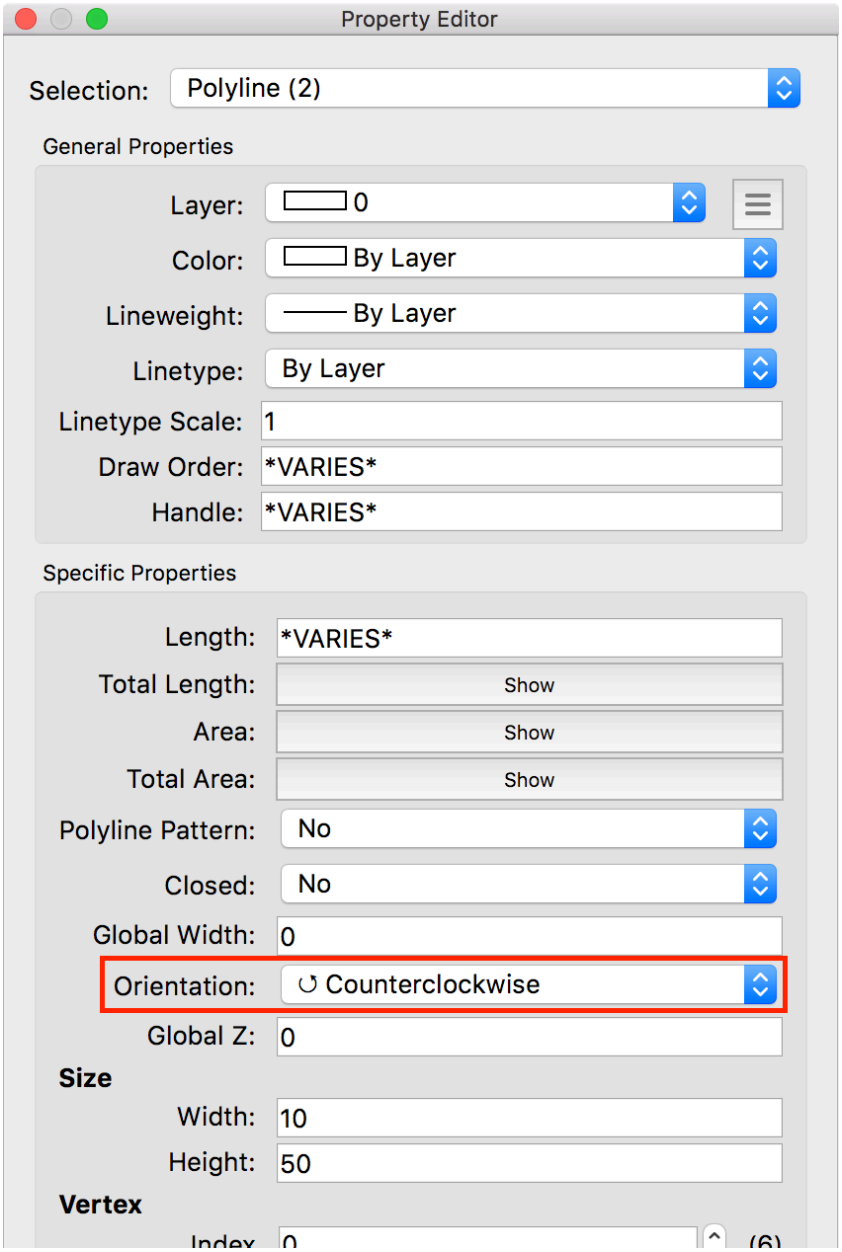
For open polylines, make sure that both polylines have the same direction. You can reverse the direction of a polyline using *Modify* > *Reverse*.

For example, if your polylines are vertical strings of lines as shown below, make sure they both have the start point at the top (or bottom). You can see the start point of a polyline highlighted in red if the polylines are selected as shown below. The result of the morphing operation is shown between the selected polylines:

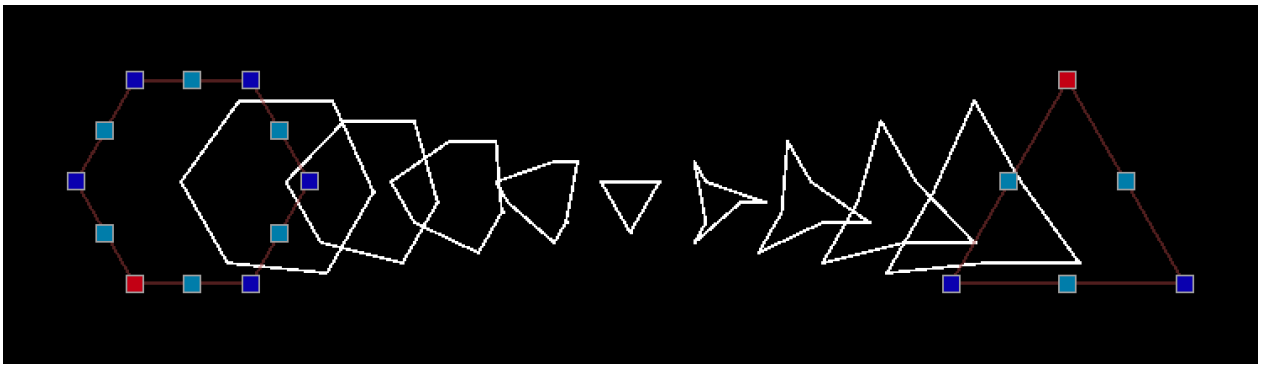


Closed Polylines

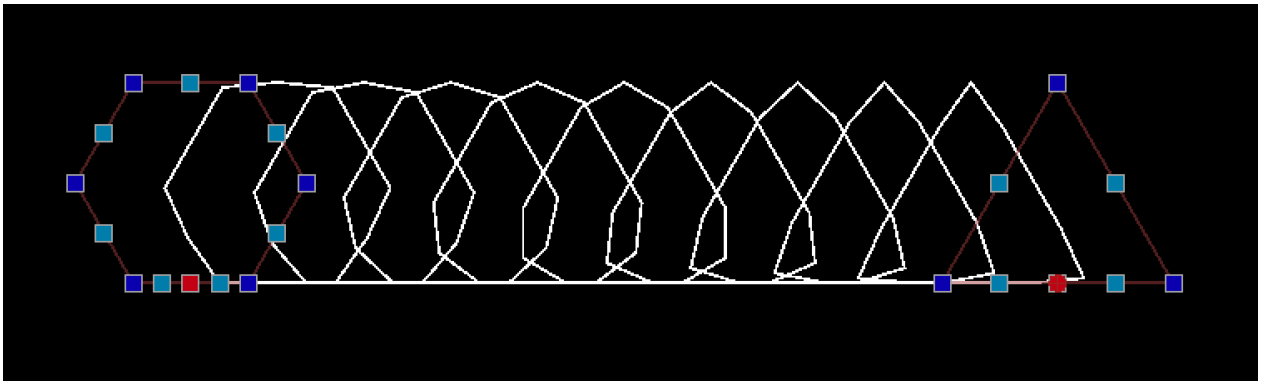
For closed polylines, make sure that both polylines have the same orientation (clockwise or counterclockwise). Select both polylines and use the property editor to change the property *Orientation* to either *Clockwise* or *Counterclockwise*:



In addition, check that the start points of the closed polylines are aligned to prevent any rotation while morphing. In this example, the start points of the two closed polylines are not aligned and the morphing leads to a rotation type transformation:



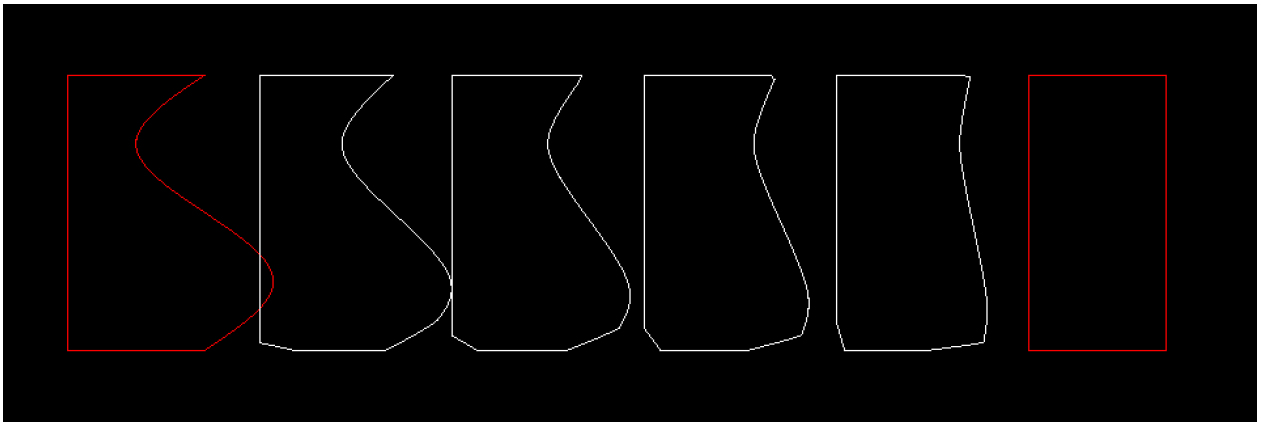
You can relocate the start point of a polyline by right-clicking on it and choosing *Relocate Startpoint*. Then, click the new position of the start point. This can be on any node or on any segment of the polyline. The morphing is now much smoother:



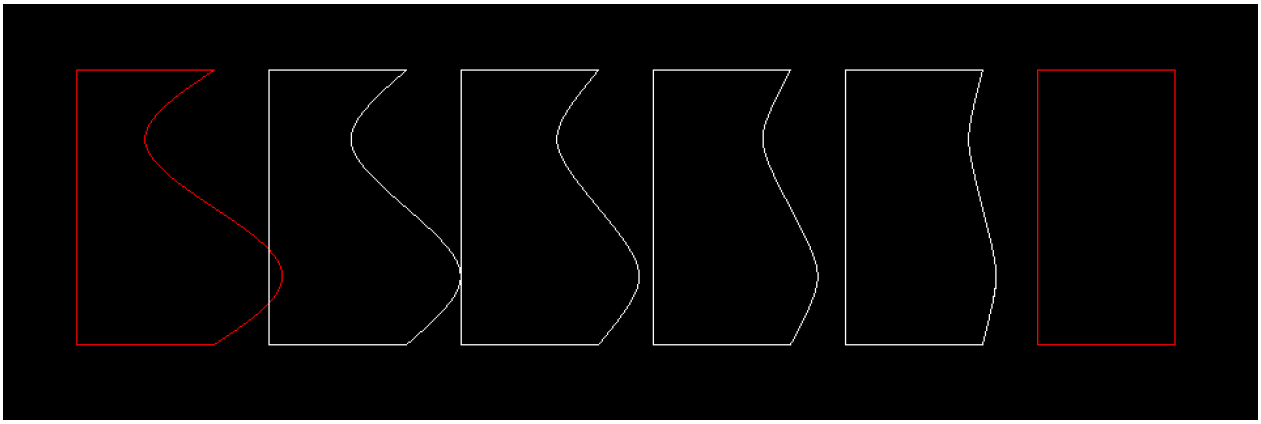
Polylines with Different Numbers of Segments

If the two polylines have the exact same number of segments, each segment of the source polyline is transformed to the corresponding segment of the target polyline.

If the two polylines have a different number of segments, a different approach is applied which can lead to a distortion at sharp corners as shown in this example (note the left bottom border being distorted while morphed):



To prevent this effect, split the polyline into two polylines at the top right and bottom right corners using *Modify* > *Divide*. Then morph the two polylines individually:

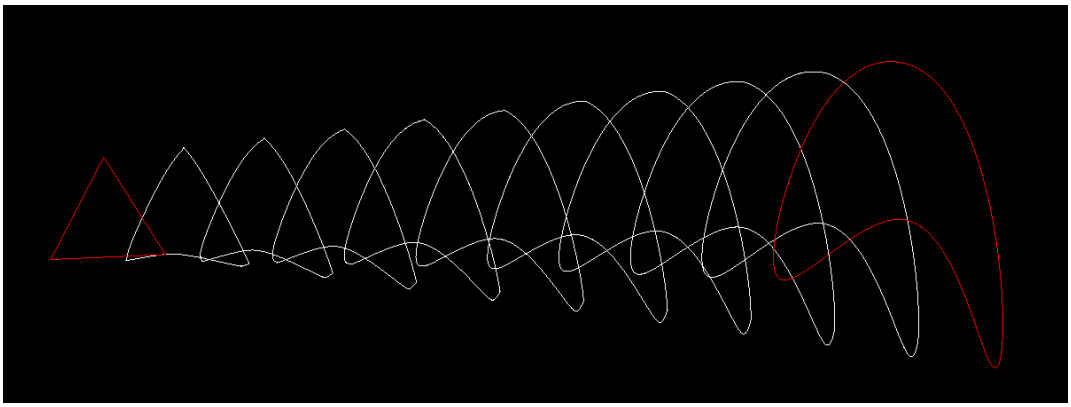


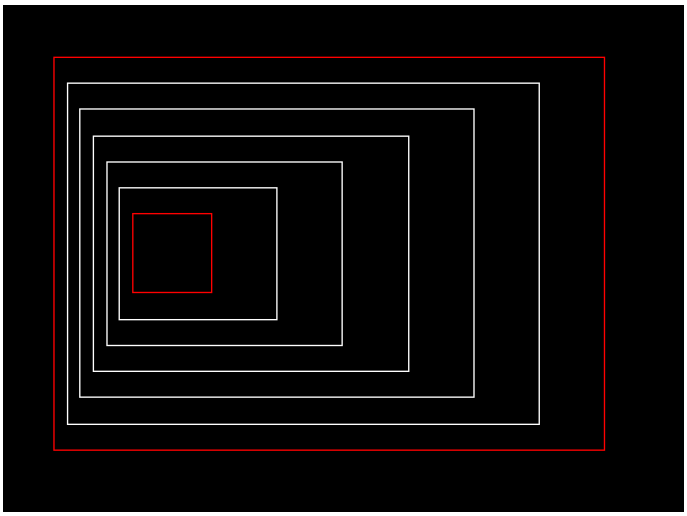
Step 2: Morphing

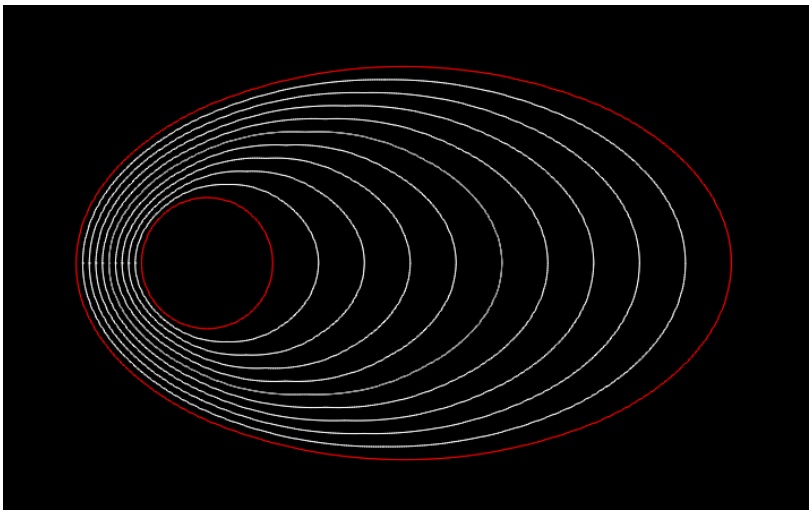
To morph one polyline to another, choose *Polyline* > *Draw* > *Morph*, click the first polyline and then the second.

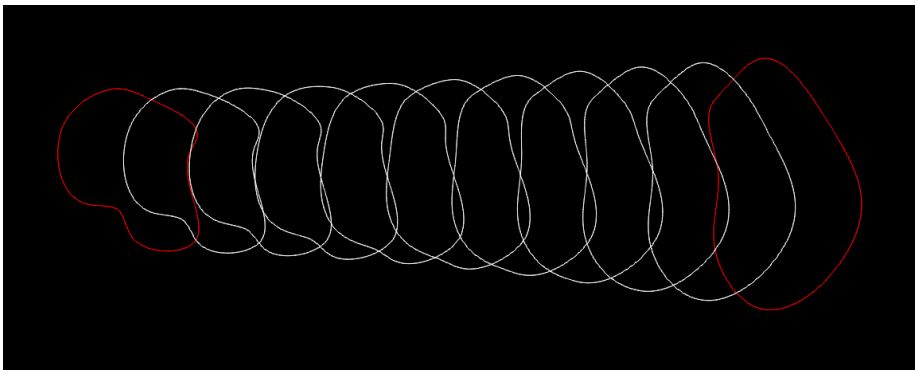
The options toolbar at the top allows you to enter the number of steps and choose to interpolate arc segments (for strictly applying morphing also to points on arcs). You can also enter a known custom factor to apply, where 0.0 is on top of the first polyline and 1.0 matches the second polyline. 0.5 would be exactly in the middle. You can also extrapolate by entering a factor > 1.

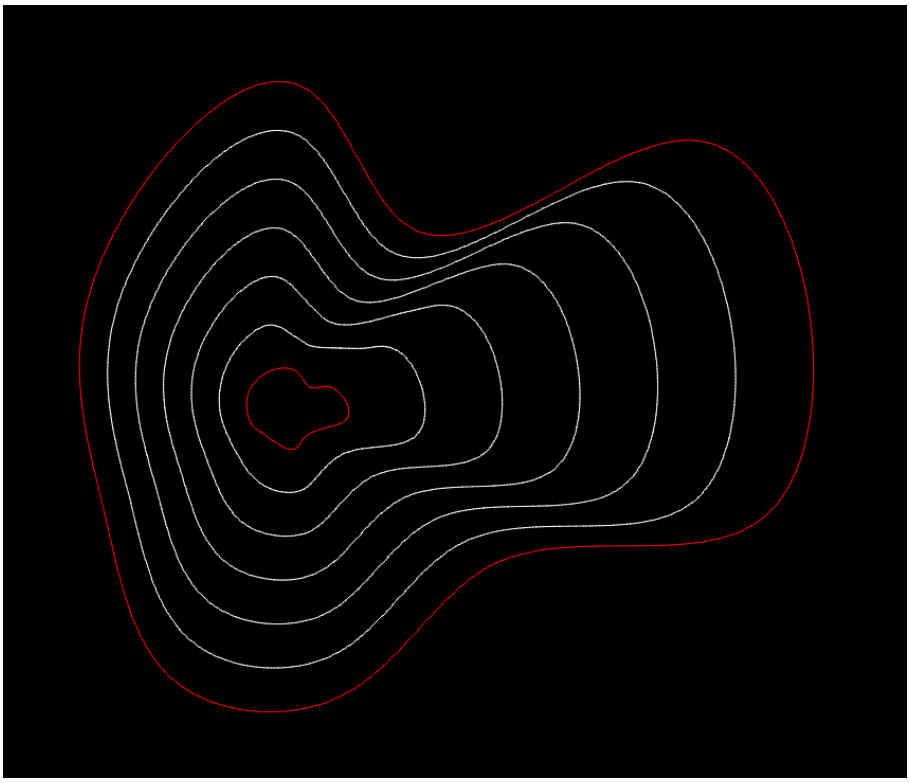
Examples



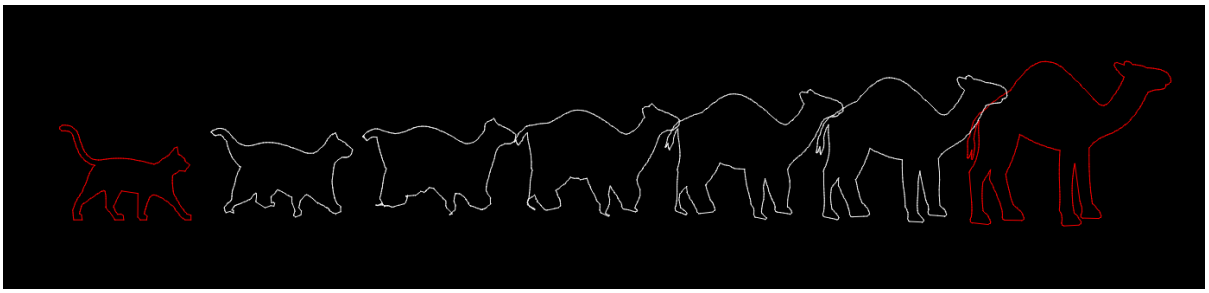




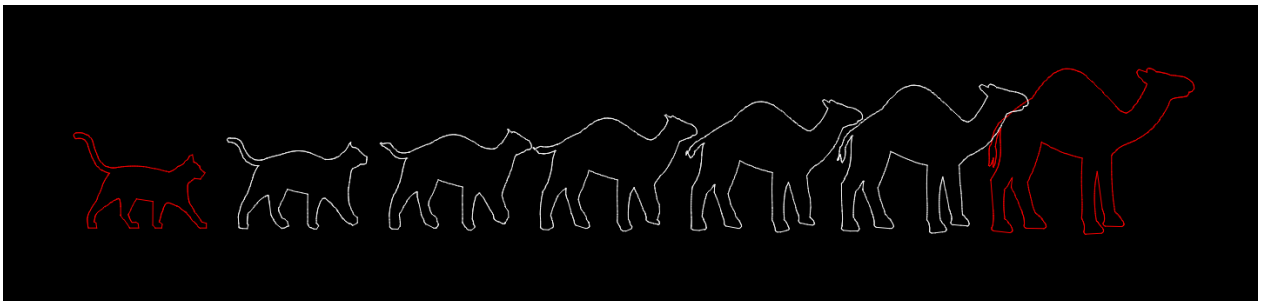




Single polyline:



Five separate polylines, split up at strategic places to separate legs, tails, head:



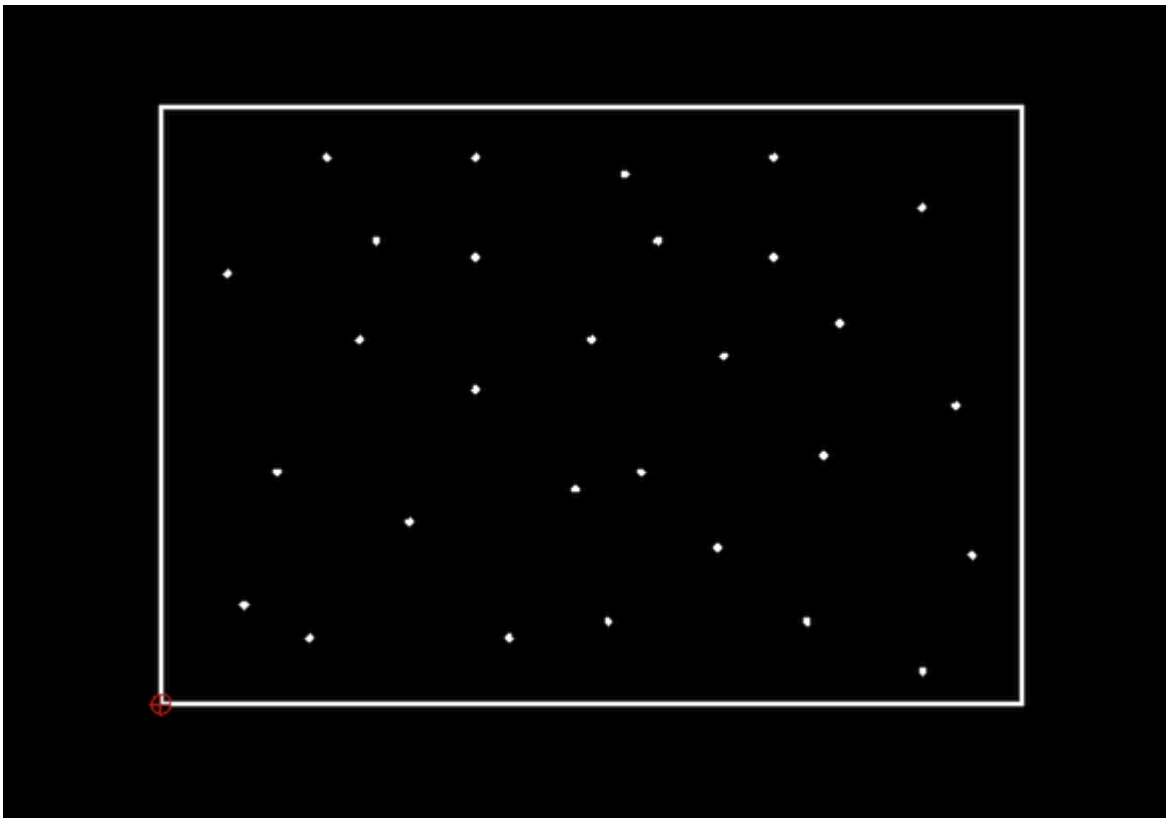
16.3.4 Reusing Drawing Positions

Introduction

There are situations in which existing positions of a drawing are to be used as basis for a new operation. The positions may for example be defined as points or other entities at those positions. The positions are not in any regular shape or pattern but arbitrary coordinates, perhaps as measured from an existing situation.

Example

For the sake of this example, let's assume the points shown here are such positions and we want to add a unique number to each point.

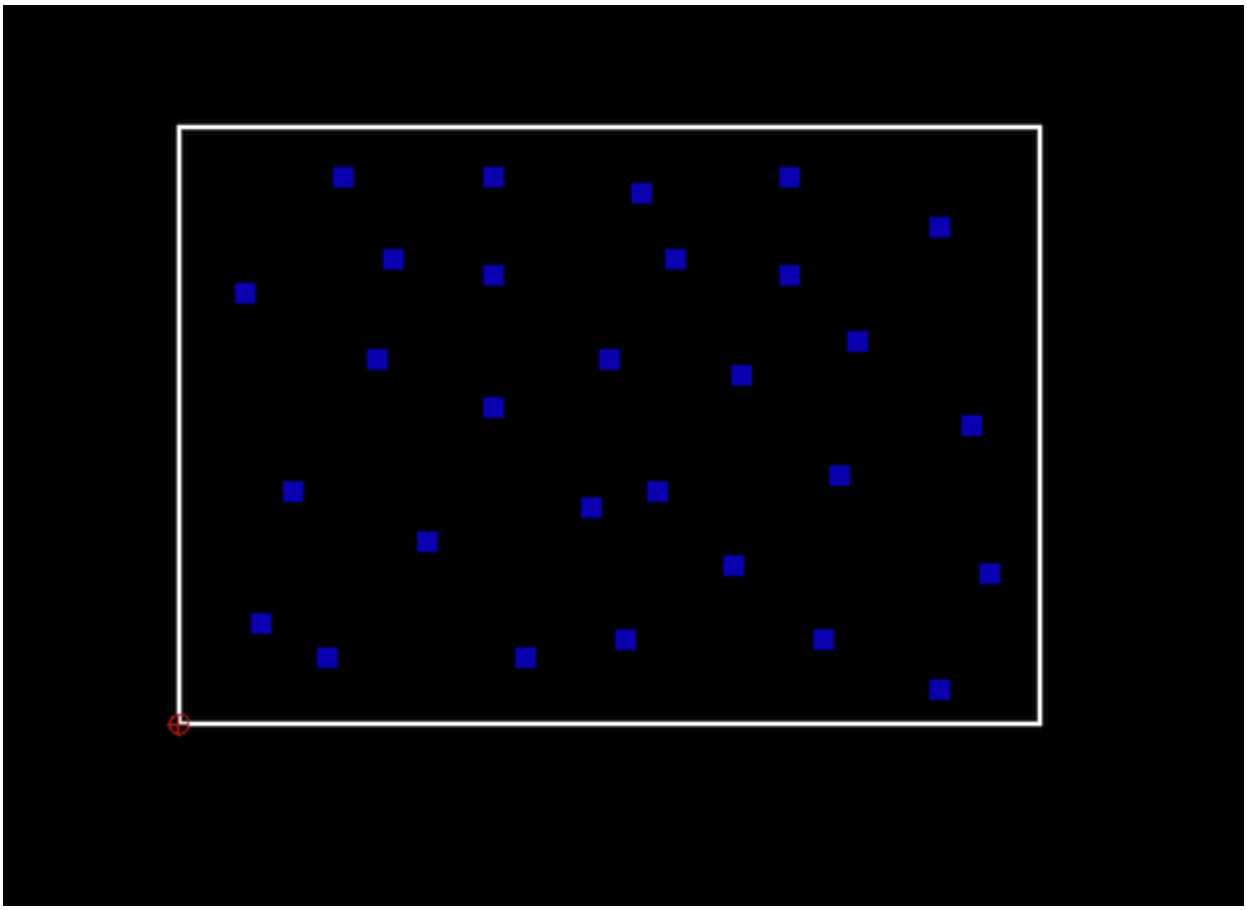


For this example, we add a text label with a counter at each position of a point. However, we could also add a circle around each point or paste the clipboard at each location of a point.

Rather than clicking each point individually, QCAD can record (store) those positions and then replay or reuse them for another tool. This can be any tool that expects coordinates as input, for example a tool for drawing points, lines or circles with given radius, pasting the clipboard, adding text, inserting a block, etc.

Step 1: Selecting the Positions

First, select all the entities that define the positions you want to work with. Here, we select the points.



Step 2: Storing the Positions

Choose *Misc* > *Information* > *Store Positions* to store the positions of the selected entities. Positions are stored in drawing / display order.

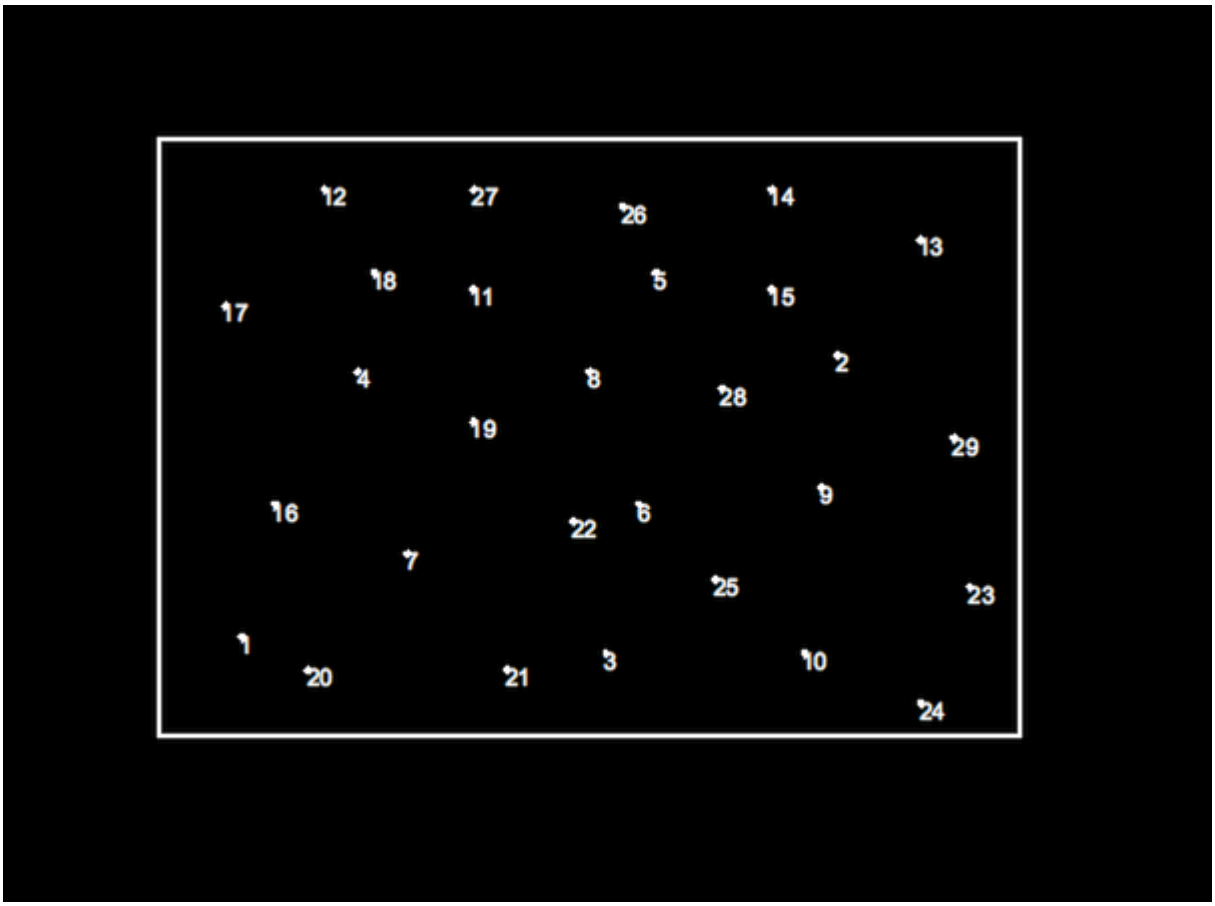
For points, the coordinate of the point is stored. For lines, the start point of each line. For polylines, each vertex. For splines, each fit point or control point.

Step 3: Starting the Tool to Use the Positions

Next, start the tool you wish to use the positions for. In this example, we add a counter to each point in drawing order using *Misc* > *Draw* > *Counter*.

Step 4: Using the Stored Positions

Instead of clicking each point now, we can simply choose *Misc* > *Information* > *Use Positions* and QCAD "clicks" each stored position for us.



Other Example Uses

- Refactoring drawings with many repeated loose patterns of entities to use block references. For example, replacing all identical circles with a block reference of a block containing a circle. All circles can then be changed at once by changing that block.
- Pasting entities at each vertex of a polyline.
- Drawing a point entity at each control point of a spline.
- Adding counters to entities / vertices.
- Creating polylines based on points or circle centers.
- Replacing points with circles.

16.3.5 Using a List of Coordinates

QCAD offers a simple and convenient way to quickly import or use a list of plain coordinates. These coordinates may for example originate from a spreadsheet, CSV file or a 3rd party program. Based on these coordinates, QCAD can create points, lines, a polyline, a spline, circles with a given radius, text labels at the given coordinates, etc.

Step 1: Getting the Coordinates into the Right Format

First, get the coordinates into the following format (for example through a CSV export from a spreadsheet):

```
x1,y1
x2,y2
x3,y3
```

```
...
```

For example:

```
12.5,5.4
7.3,2.5
23.5,5.6
...
```

Step 2: Copying the Coordinates

Copy the whole list of these coordinates to the clipboard (select the multiple lines from the plain text or CSV file and copy to clipboard in a text editor).

Step 3: Using / Pasting the Coordinates

In QCAD, start the tool you want to use. This can for example be *Draw* > *Line* > *Line from 2 Points*, *Draw* > *Polyline* > *Draw Polyline*, *Draw* > *Circle* > *Center, Radius* or any other tool that accepts a sequence of coordinates as input.

Click into the command line input field or press the space bar to make sure the command line has the keyboard focus.

Paste the list of coordinates into the command line input field using Ctrl-V (or ⌘V on macOS).

QCAD now enters the list of coordinates just as if you would have entered them one by one manually, and in the process draws points, lines, a polyline, spline, circles, text labels or whatever the tool creates.

16.4 Fonts

QCAD uses CXF line fonts to render text as true vector geometry, which is essential for engraving, plotting and other CAD workflows. This section explains how to work with these fonts, including how to load, edit and save your own line fonts in the CXF format so you can adjust existing glyphs or create custom characters.

16.4.1 Editing CXF Fonts

Introduction

QCAD comes with a collection of line fonts in its own CXF format. Line fonts are sometimes used in CAD instead of TrueType fonts. Line fonts are also often used for engraving or other further processing. QCAD can use any TTF font installed on the system and CXF fonts available in its fonts directory.

QCAD can load and edit fonts in CXF format.

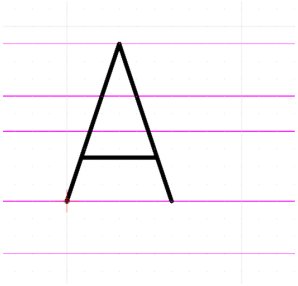
Opening a CXF Font for Editing

To load an existing CXF font file in QCAD to view and edit its glyphs, simply use the menu *File* > *Open* and choose the format filter "CXF Font Files (*.cxf)". Then choose the font file you want to open. The fonts that come with QCAD are stored in directory "fonts" of your QCAD installation.

Note that depending on your installation path, your login credentials and your operating system, this directory might be read-only. In this case, you might want to move the font file to another folder for editing.

Editing a CXF Font

The font glyphs (representation of the various letters and symbols in the font) are loaded into QCAD as block definitions. Make sure the block list is visible (*View* > *Block List*), so you can easily navigate through the font glyphs. Each block representing a glyph has a name in the format "[Hexadecimal Unicode] [Character]", for example "0x0041 A" for a capital "A" with Unicode 65 (41 in hexadecimal). To open and edit a font glyph, double-click on its block. The glyph is then shown in the graphics view of QCAD:



Auxiliary Lines

Auxiliary lines may be shown, depending on the font. These are not part of the font and are only meant to provide some guidance when creating and editing glyphs. You can switch the auxiliary lines off by switching off layer "Auxiliary".

Geometry of Glyphs

By convention of the CXF format, a capital "A" should be constructed with a height of 9 drawing units. When the font is used for a text entity, it is scaled by QCAD according to the text height chosen by the user. Glyphs should have their reference point 0,0 at the lower left corner of the baseline. In the CXF format, glyphs can contain lines, arcs and polylines. Splines may be used, but are converted into polylines during export.

Adding New Glyphs

To add a new glyph currently not present in a font you need to add a new block with the name [Hexadecimal Unicode] [Character], for example "0x0041 A" for a capital A. You can look up Unicode codes in the Code Charts section of the official web site of the Unicode Consortium. If a font already contains glyphs for many characters, you might want to copy an existing block with a similar glyph using *Block* > *Copy Block*.

Saving a CXF Font

Font files can be saved like drawing files using *File* > *Save* or *File* > *Save As*. QCAD needs to be restarted to reinitialize all the CXF fonts in its "fonts" directory.

CXF File Header

CXF fonts store additional information about the font in a simple, human readable font header. This information can be edited with a text editor or within QCAD as follows:

- Make sure that the property editor shows properties from layers: *Edit* > *Application Preferences* > *Widgets* > *Property Editor*: Tick the option "Display properties of current block and current layer".
- Activate layer "0" in the layer list (*View* > *Layer List*).
- In the property editor (*View* > *Property Editor*), change the selection at the top from "All" to "Layer". The property editor now shows properties of layer "0".
- Use the property editor to change the meta information of the CXF font:

- **Authors**: comma separated list of font authors
- **AuxiliaryLines**: comma separated list of auxiliary line positions and angles in the format: x1,y1,angle1, x2,y2,angle2, ...
- **Encoding**: font encoding (should be "UTF-8")
- **LetterSpacing**: space between two letters
- **WordSpacing**: space between two words
- **LineSpacingFactor**: can be used to increase / decrease default line spacing
- **Names**: comma separated list of font names / aliases for this font

16.5 Installation

This section covers getting QCAD up and running on your system. Besides the standard installation, it explains alternative setups such as running QCAD Portable directly from a USB stick or other removable drive, so you can carry your CAD environment and settings with you and use them on any compatible computer.

16.5.1 Installing QCAD under Linux

QCAD is distributed for Linux in three different package formats. Which one you should use depends on your system:

- **Flatpak** (`.flatpak`) - the recommended format for current desktop Linux distributions and the only supported way to install QCAD on ChromeOS. The Flatpak bundles all libraries QCAD needs, so it also runs on systems whose libraries are too old or too new for the other packages.
- **Installer** (`.run`) - a self-extracting installer that unpacks QCAD into your home directory. No administrator rights are required.
- **Archive** (`.tar.gz`) - the same files as the installer, for users who prefer to extract the application manually.

None of these packages install QCAD system-wide, and none of them require root privileges. QCAD is always installed for the current user only and can be removed again by deleting a single directory.

The following pages describe each installation method in detail:

- *Installing QCAD from a Flatpak* - installation from the `.flatpak` package on a desktop Linux distribution.
- *Installing QCAD from the Installer* - installation from the `.run` installer or the `.tar.gz` archive.
- *Installing QCAD on a Chromebook* - installation from the `.flatpak` package inside the Linux environment of ChromeOS.

Throughout these pages, version numbers are written as `3.x.x`. Replace this with the actual version number of the QCAD release you have downloaded, for example `3.32.9`.

Choosing the Correct Package

QCAD packages for Linux are built for different CPU architectures and against different versions of the Qt library. If you are unsure which package to choose, determine your CPU architecture first. Open a terminal and enter:

```
uname -m
```

- `x86_64` means you have a 64-bit Intel or AMD CPU.

- `aarch64` (or `arm64`) means you have a 64-bit ARM CPU.

Then pick the package from the download area accordingly:

Your system	Section in the download area	File name
Current 64-bit Intel/AMD distribution	Linux > Qt 6 > 64-bit Intel	`qcad-3.x.x-pro-linux-qt6-x86_64.flatpak` / `.run` / `.tar.gz`
Current 64-bit ARM distribution	Linux > Qt 6 > ARM (64-bit)	`qcad-3.x.x-pro-linux-qt6-arm64.flatpak` / `.run` / `.tar.gz`
Older 64-bit Intel/AMD distribution	Linux > Qt 5 > 64-bit Intel	`qcad-3.x.x-pro-linux-qt5.14-x86_64.run` / `.tar.gz`
Very old 64-bit Intel/AMD distribution	Linux > 64-bit Intel	`qcad-3.x.x-pro-linux-x86_64.run` / `.tar.gz`
Very old 32-bit Intel/AMD distribution	Linux > 32-bit Intel	`qcad-3.x.x-pro-linux-x86_32.run` / `.tar.gz`

Note: the Qt 5 and 32-bit packages are provided for compatibility with legacy distributions only. Use a Qt 6 package unless QCAD refuses to start on your system.

The file names above are those of the QCAD Professional packages. Trial packages are named `qcad-3.x.x-trial-...` and are otherwise identical.

16.5.1.1 Installing QCAD from a Flatpak

This page guides you through the installation of QCAD from a Flatpak on Linux operating systems. If you are using a Chromebook with ChromeOS, follow the instructions under *Installing QCAD on a Chromebook* instead.

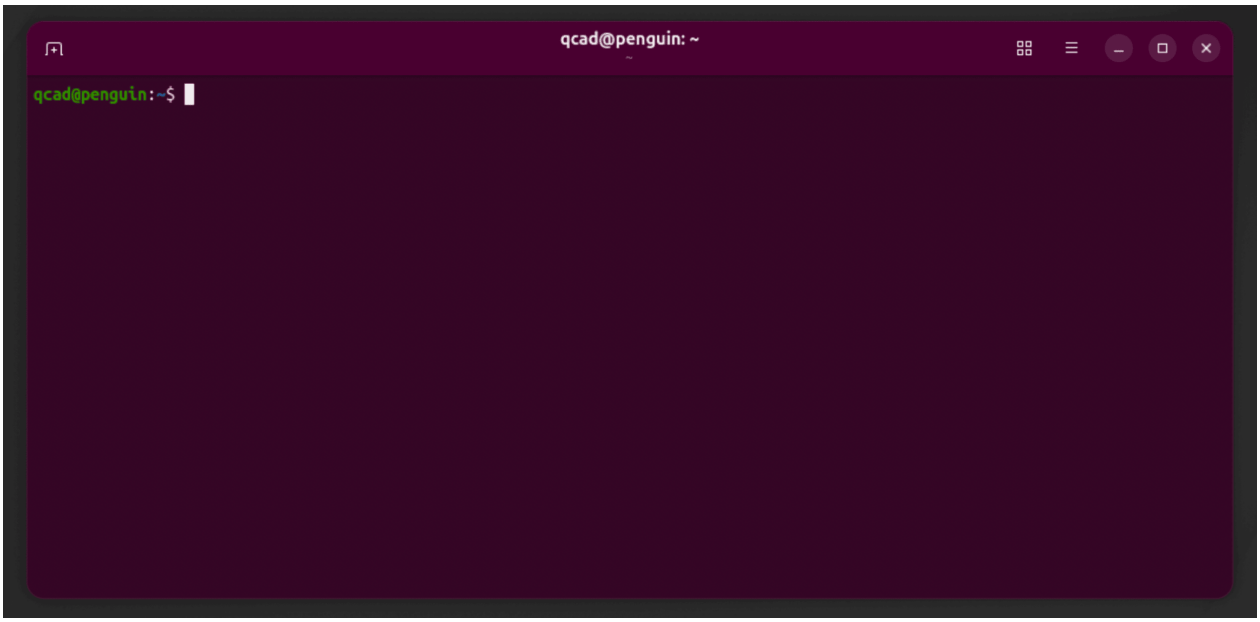
A Flatpak bundles all libraries QCAD needs. It therefore also runs on systems whose libraries are too old or too new for the installer and the archive packages.

We install QCAD using a terminal, as this is the preferred way to manage and maintain a Linux system. There are also graphical applications that can be used to install Flatpaks, but these are not covered here.

Version numbers are written as `3.x.x` below. Replace this with the actual version number of the QCAD release you have downloaded, for example `3.32.9`.

Opening a Terminal

Open the terminal application of your desktop environment. A window opens showing a prompt such as `username@host:~$`, where `username` is your user name (here "qcad") and `host` is the name of your computer (here "penguin"):



Installing Flatpak

Copy the following command to your clipboard and paste it into the terminal. You can paste into a terminal with a right-click:

```
sudo apt install flatpak
```

Hit the Enter key.

If the terminal asks you "Do you want to continue?", hit Enter again. Wait for the installation to complete, which typically takes a few minutes. When the prompt appears again, the installation has finished.

If the terminal tells you that Flatpak is already the newest version, you already have Flatpak and you can continue with the next step.

Note: `apt` is the package manager of Debian and Ubuntu based distributions. On other distributions, use the corresponding command of your package manager, for example `sudo dnf install flatpak` on Fedora or `sudo pacman -S flatpak` on Arch Linux.

Checking Your Flatpak Version

Run:

```
flatpak --version
```

Make sure your Flatpak version is at least 1.12, otherwise the installation will likely fail.

Adding the Flathub Repository

QCAD requires a runtime which is downloaded from Flathub, the central Flatpak repository. Add it with:

```
flatpak --user remote-add --if-not-exists flathub https://dl.flathub.org/repo/
flathub.flatpakrepo
```

Hit Enter.

Installing the Correct Runtime

Determine your CPU type as described under *Installing QCAD under Linux*, then install the matching runtime. If you are unsure, enter `uname -m` in the terminal.

If your CPU is `x86_64`:

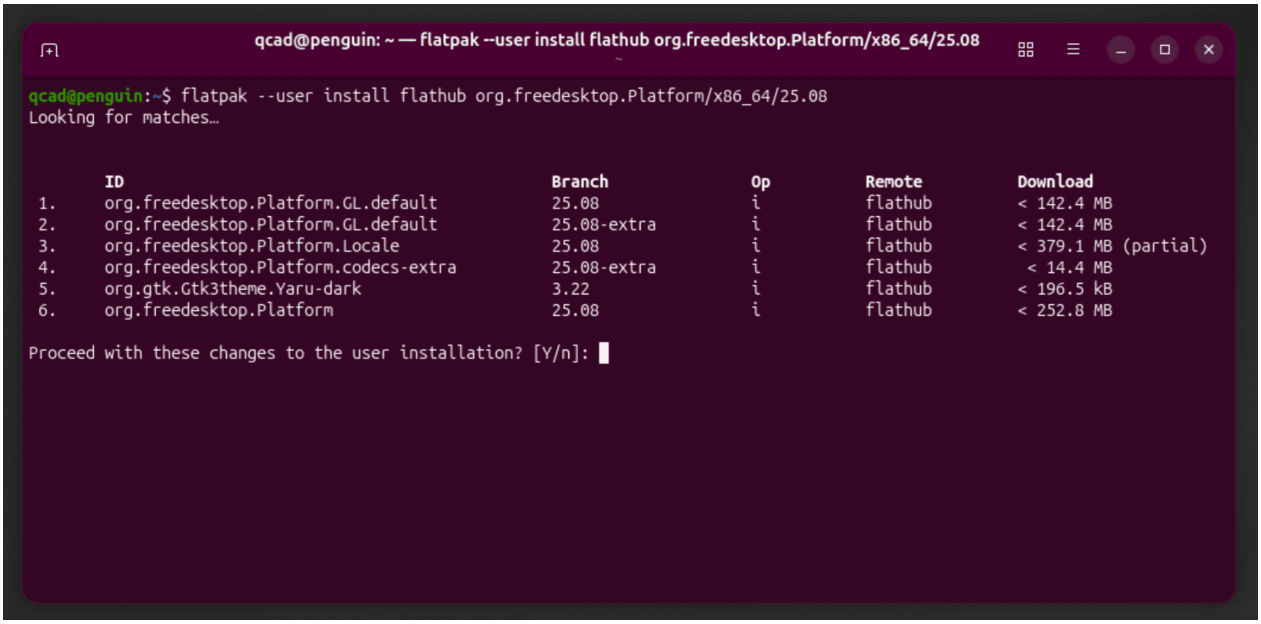
```
flatpak --user install flathub org.freedesktop.Platform/x86_64/25.08
```

If your CPU is `aarch64`:

```
flatpak --user install flathub org.freedesktop.Platform/aarch64/25.08
```

Hit Enter.

Flatpak lists the packages it is about to install and asks for confirmation. Confirm with Enter:



Flatpak installs these packages and reports "Installation complete" when it is done.

Note: if you get the error "Unable to load summary from remote flathub: URI … exceeded maximum size of 10485760 bytes", your Flatpak version is outdated. In this case, run:

```
flatpak --user remote-delete --force flathub
flatpak --user remote-add flathub https://dl.flathub.org/repo/flathub.flatpakrepo
```

and then try the installation again.

Downloading the QCAD Package

Open the download area using the link you have received after your purchase.

Scroll down to the section that matches your CPU: Linux > Qt 6 > 64-bit Intel for Intel and AMD CPUs, or Linux > Qt 6 > ARM (64-bit) for ARM CPUs. Below that title, click on the item with "Flatpak" in its name to download the Flatpak file to your Downloads folder.

Wait for the download to complete.

Installing QCAD

Go back to the terminal and change into your Downloads folder:

```
cd ~/Downloads
```

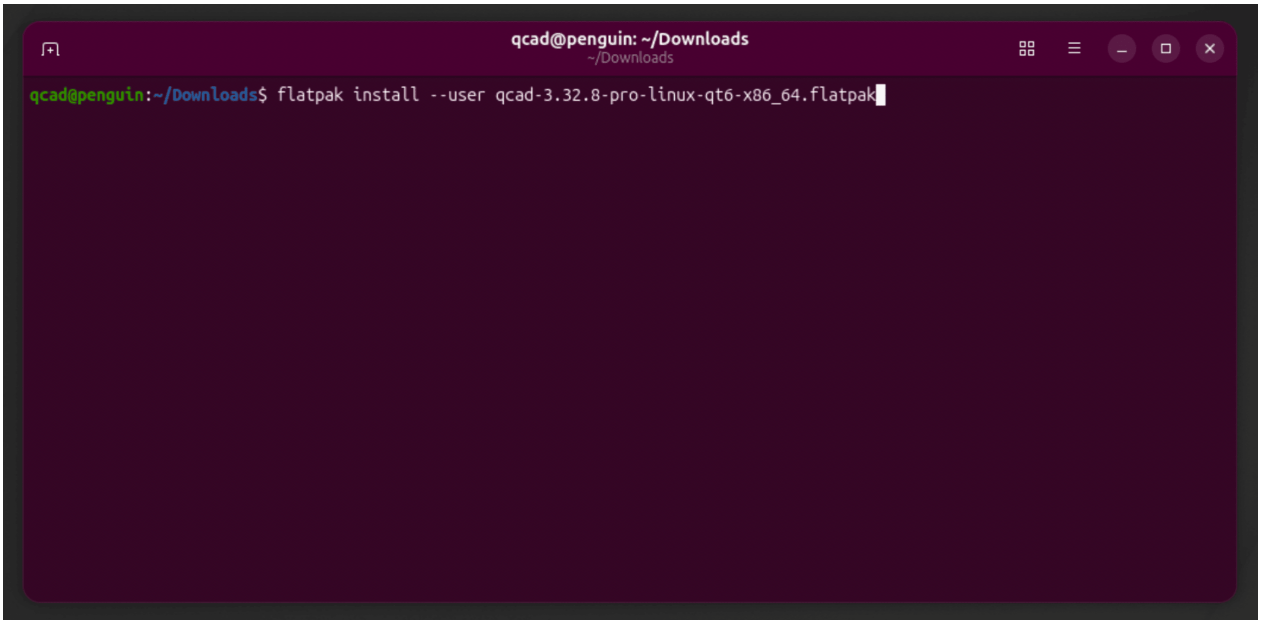
If you have an Intel or AMD CPU, enter:

```
flatpak install --user qcad-3.x.x-pro-linux-qt6-x86_64.flatpak
```

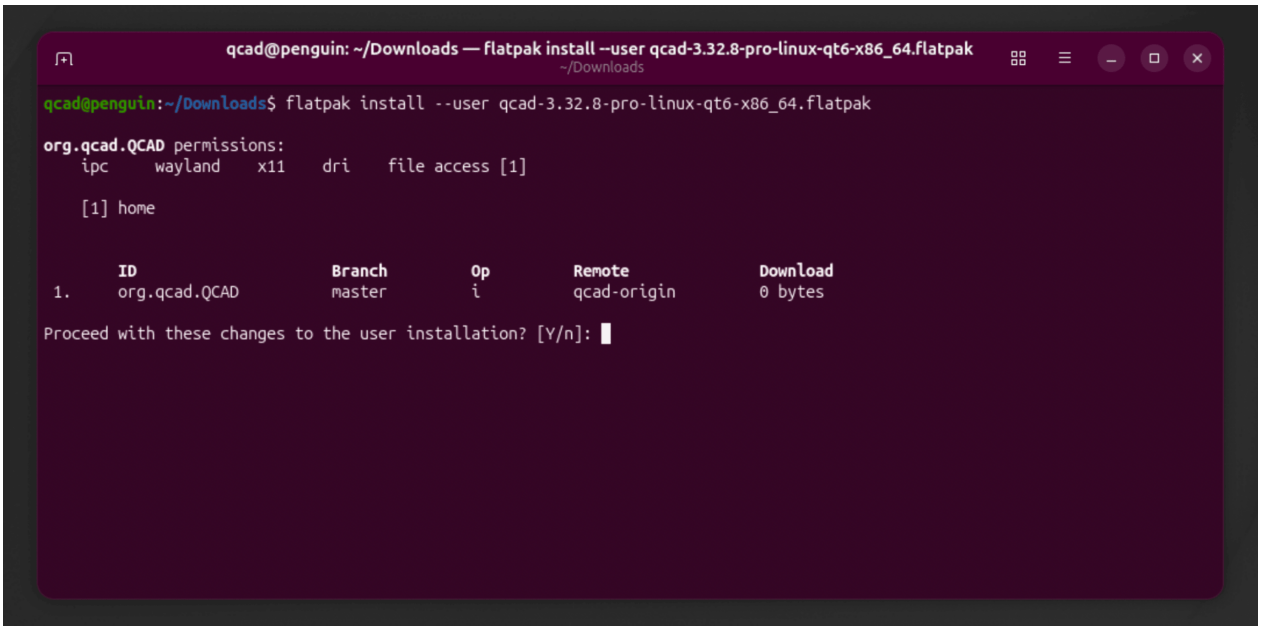
If you have an ARM64 CPU, enter:

```
flatpak install --user qcad-3.x.x-pro-linux-qt6-arm64.flatpak
```

Hit Enter. Remember to replace `3.x.x` with the exact version number of the package you have downloaded.



Flatpak shows the permissions QCAD requests and asks you whether to proceed:



Hit Enter to confirm.

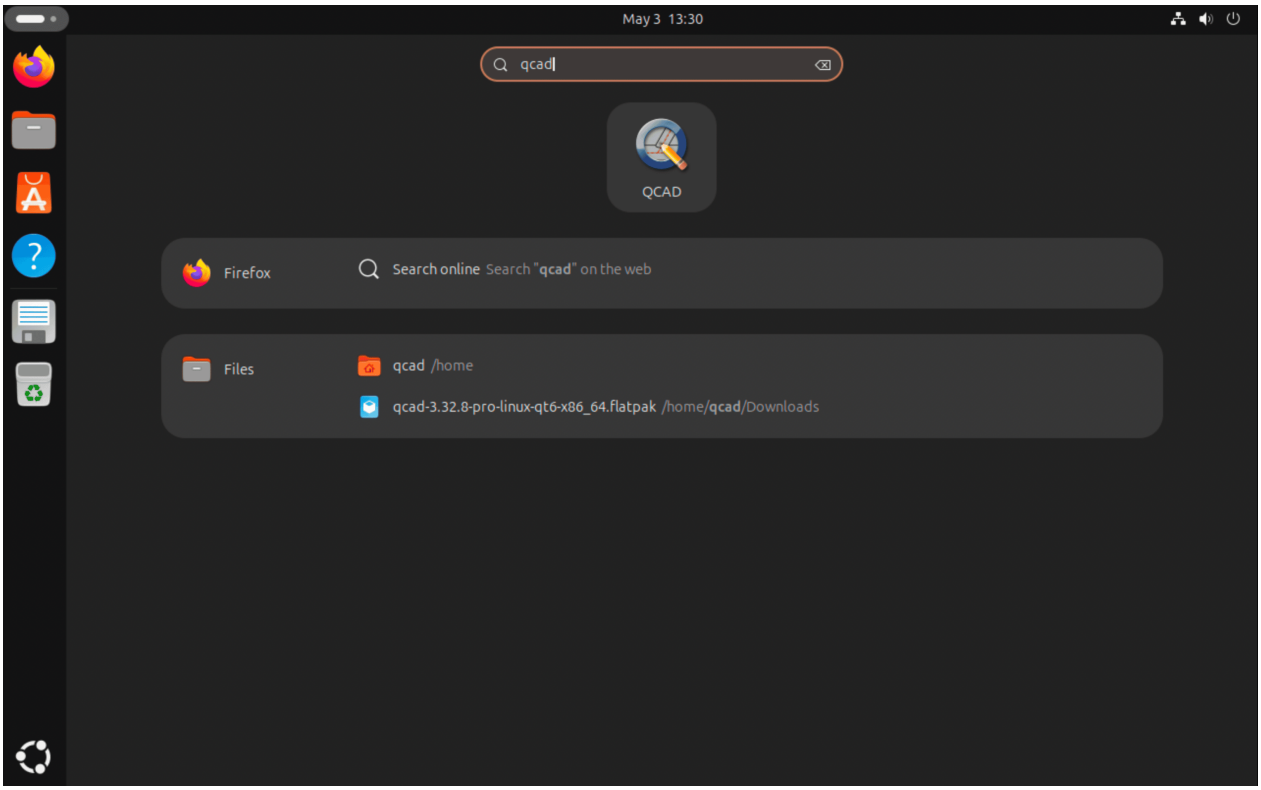
When the installation has finished, the terminal shows "Installation complete."

Running QCAD

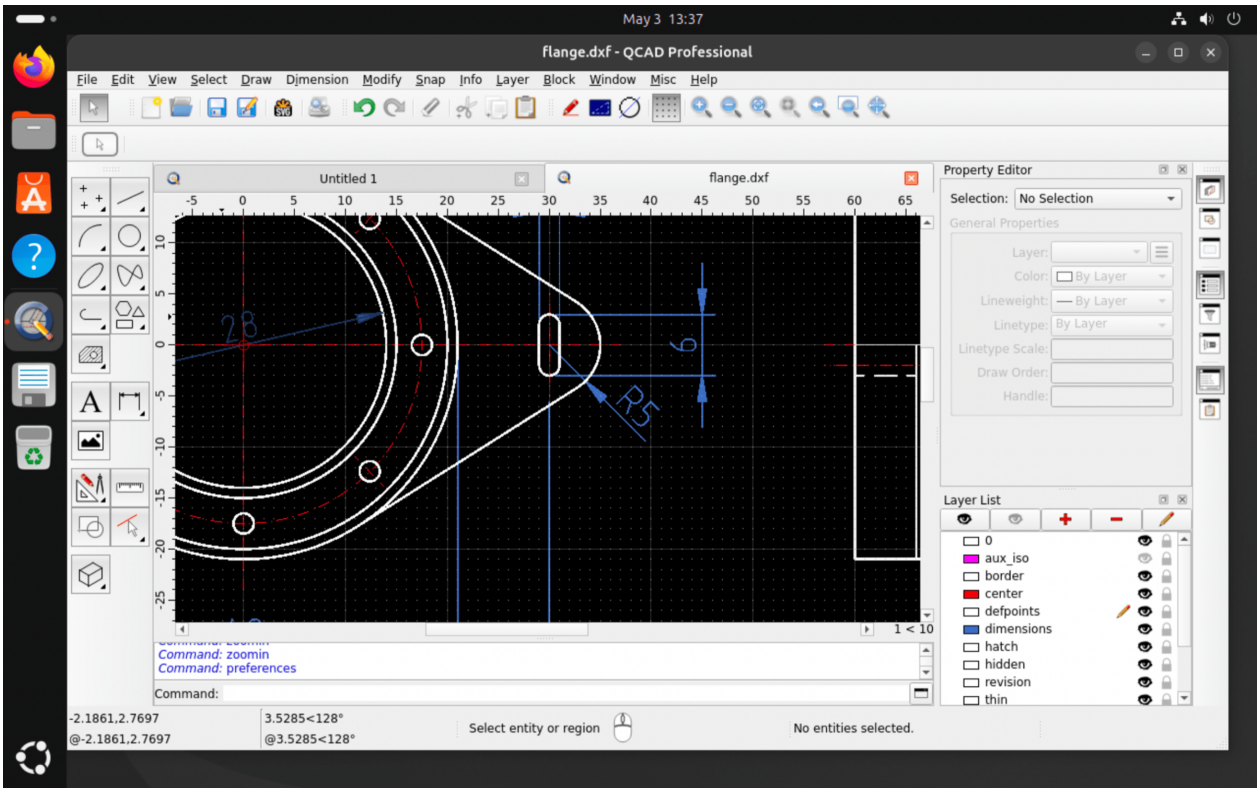
To start QCAD from the terminal, enter:

```
flatpak run org.qcad.QCAD
```

Alternatively, use the application launcher of your desktop environment, where QCAD now appears as a regular application:



QCAD starts with its welcome screen where you can choose some initial settings. Once you confirm, the QCAD application window is shown and you can start working:



Running Command Line Tools from a Flatpak

A Flatpak installation does not give you access to the launcher scripts of the QCAD command line tools (`dwg2pdf`, `dwg2bmp`, ...).

To still run these tools, pass the corresponding launcher script to QCAD using the `-autostart` argument:

```
flatpak run org.qcad.QCAD -autostart scripts/Pro/Tools/Dwg2Pdf/Dwg2Pdf.js
```

The launcher scripts of the various QCAD command line tools are:

Tool	Launcher script
merge	scripts/Tools/MergeDrawings/MergeDrawings.js
bbox	scripts/Pro/Tools/BoundingBox/BoundingBox.js
bmp2dwg	scripts/Pro/Tools/Bmp2Dwg/Bmp2Dwg.js
dwg2bmp	scripts/Pro/Tools/Dwg2Bmp/Dwg2Bmp.js
dwg2csv	scripts/Pro/Tools/Dwg2Csv/Dwg2Csv.js
dwg2dwg	scripts/Pro/Tools/Dwg2Dwg/Dwg2Dwg.js
dwg2maptiles	scripts/Pro/Tools/Dwg2MapTiles/Dwg2MapTiles.js
dwg2pdf	scripts/Pro/Tools/Dwg2Pdf/Dwg2Pdf.js
dwg2svg	scripts/Pro/Tools/Dwg2Svg/Dwg2Svg.js
dwgexplode	scripts/Pro/Tools/DwgExplode/DwgExplode.js
dwghatch	scripts/Pro/Tools/DwgHatch/DwgHatch.js
dwghatcharea	scripts/Pro/Tools/DwgHatchArea/DwgHatchArea.js

Tool	Launcher script
dwginfo	scripts/Pro/Tools/DwgInfo/DwgInfo.js
dwgmapconvert	scripts/Pro/Tools/DwgMapConvert/DwgMapConvert.js
svg2dwg	scripts/Pro/Tools/Svg2Dwg/Svg2Dwg.js
dwgnest (QCAD/CAM)	scripts/Nest/Tools/DwgNest/DwgNest.js

Uninstalling a Flatpak Installation

To remove a Flatpak installation of QCAD, run:

```
flatpak uninstall --user org.qcad.QCAD
```

16.5.1.2 Installing QCAD from the Installer

This page describes the installation of QCAD under Linux from the self-extracting installer (`.run`) or from the archive (`.tar.gz`). Both contain the same files and install QCAD into a single directory in your home directory. Administrator rights are not required.

Choose the package that matches your system as described under *Installing QCAD under Linux*. Version numbers are written as `3.x.x` below. Replace this with the actual version number of the QCAD release you have downloaded, for example `3.32.9`.

Installation Using the Installer (.run)

Open a terminal application, for example GNOME Terminal, Konsole or xterm, and change to the directory the installer was downloaded to:

```
cd ~/Downloads
```

Make the installer executable and run it:

```
chmod a+x qcad-3.x.x-pro-linux-qt6-x86_64.run
./qcad-3.x.x-pro-linux-qt6-x86_64.run
```

This creates the directory `~/opt/qcad-3.x.x-pro-linux-qt6-x86_64` containing the QCAD application. Nothing is written outside your home directory.

Installation Using the Archive (.tar.gz)

Extract the archive into any directory of your choice. In this example we use `~/opt`:

```
mkdir -p ~/opt
cd ~/opt
tar xfvz ~/Downloads/qcad-3.x.x-pro-linux-qt6-x86_64.tar.gz
```

Launching QCAD

Change into the directory QCAD was installed into and start QCAD from there:

```
cd ~/opt/qcad-3.x.x-pro-linux-qt6-x86_64
./qcad
```

Troubleshooting

If QCAD does not start, your system might be missing libraries QCAD depends on. You can check for unmet dependencies as follows:

```
cd ~/opt/qcad-3.x.x-pro-linux-qt6-x86_64
LD_LIBRARY_PATH=. ldd qcad-bin
```

If any lines end with "not found", install the corresponding packages using the package manager of your distribution.

If dependencies cannot be resolved, use the Flatpak package instead, as described under *Installing QCAD from a Flatpak*. A Flatpak ships with all required libraries and is independent of the libraries installed on your system.

Uninstalling

The installer and the archive install QCAD into a single directory. To uninstall QCAD, simply delete that directory:

```
rm -rf ~/opt/qcad-3.x.x-pro-linux-qt6-x86_64
```

Your drawings and your QCAD configuration are stored elsewhere and are not affected.

16.5.1.3 Installing QCAD on a Chromebook

This page guides you through the installation of QCAD on ChromeOS on your Chromebook computer.

On ChromeOS, QCAD runs inside the Linux development environment as a Flatpak. This is the only supported way to install QCAD on a Chromebook.

Version numbers are written as `3.x.x` below. Replace this with the actual version number of the QCAD release you have downloaded, for example `3.32.9`.

Am I Using ChromeOS?

If you are using a laptop which mentions "Chromebook" somewhere, chances are that it has ChromeOS installed on it.

A typical ChromeOS desktop has a shelf at the bottom of the screen with a circle in the bottom left corner for the launcher:

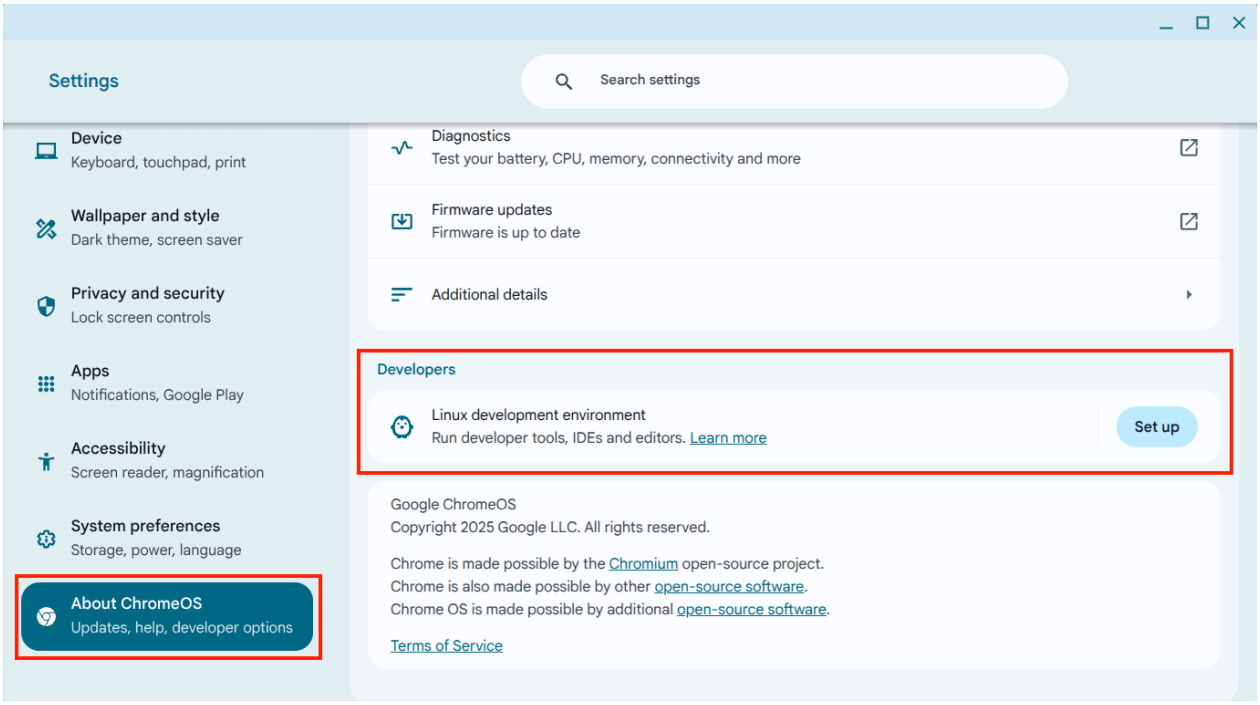


Installing the Linux Environment

QCAD runs inside the Linux development environment of ChromeOS, which has to be set up once.

Click on the circle in the bottom left corner (Launcher) to show the launch menu and start the *Settings* app.

Navigate to *About ChromeOS* and scroll down to *Linux development environment*:



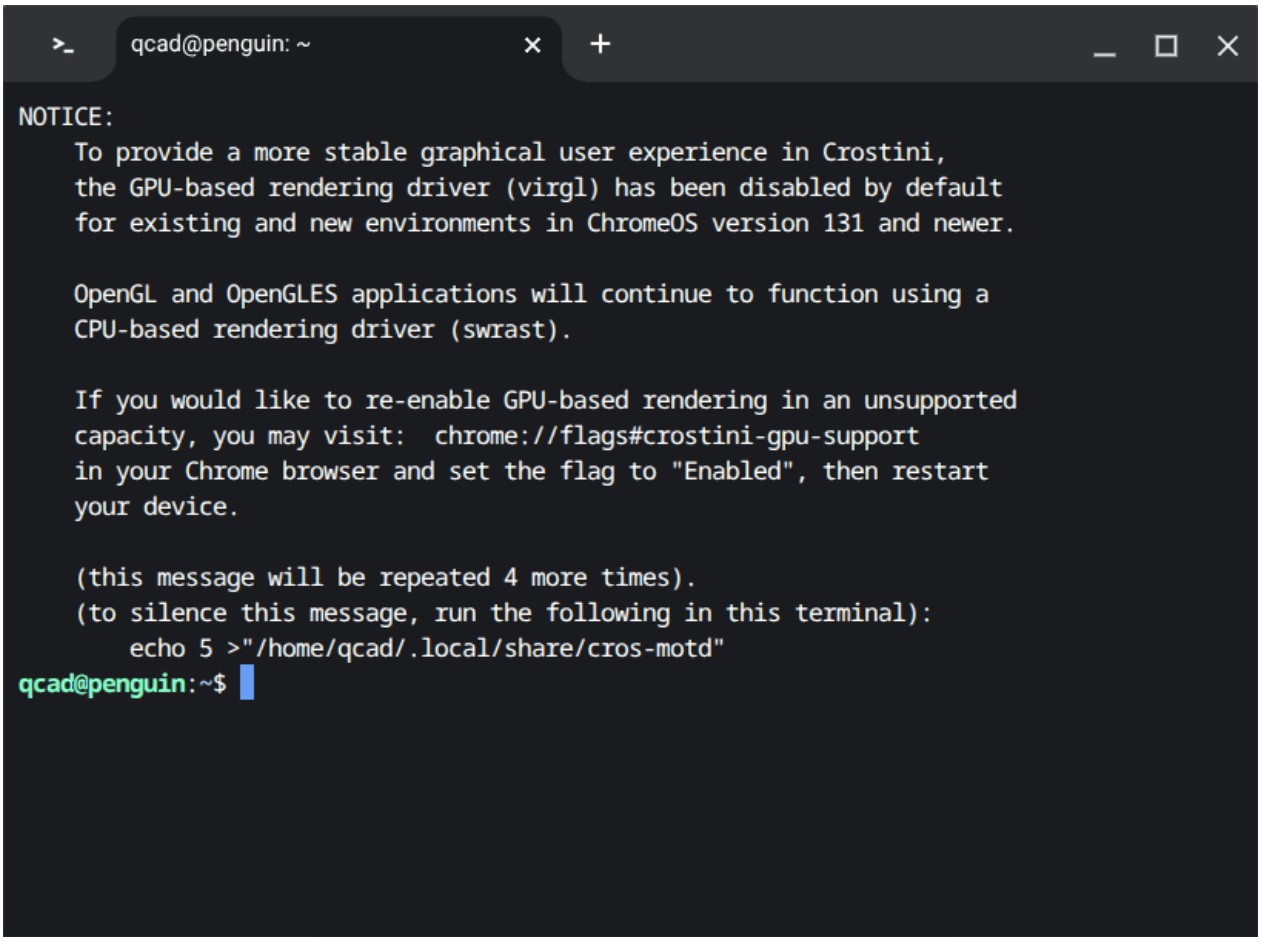
Click *Set up*, then click *Next*.

Enter a user name for the Linux environment. This can be a nickname or, for example, your first name. User names may not contain spaces and are all lowercase.

For the disk size, leave the default of 10.0 GB.

Click *Install* and wait for the installation to complete.

Eventually, a black window opens with the text `username@penguin:~$`, where `username` is the user name you have chosen. This window is a terminal and we will use it to install QCAD:



Installing Flatpak

Copy the following command to your clipboard and paste it into the terminal. You can paste into the terminal with a right-click or by tapping with two fingers on the trackpad of your Chromebook.

```
sudo apt install flatpak
```

Hit the Enter key.

When the terminal asks you "Do you want to continue?", hit Enter again.

Wait for the installation of Flatpak to complete, which typically takes a few minutes. When you see the `username@penguin:~$` line in green again, the installation has completed.

Adding the Flathub Repository

Copy the following command to your clipboard and paste it into the terminal:

```
sudo flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/
flathub.flatpakrepo
```

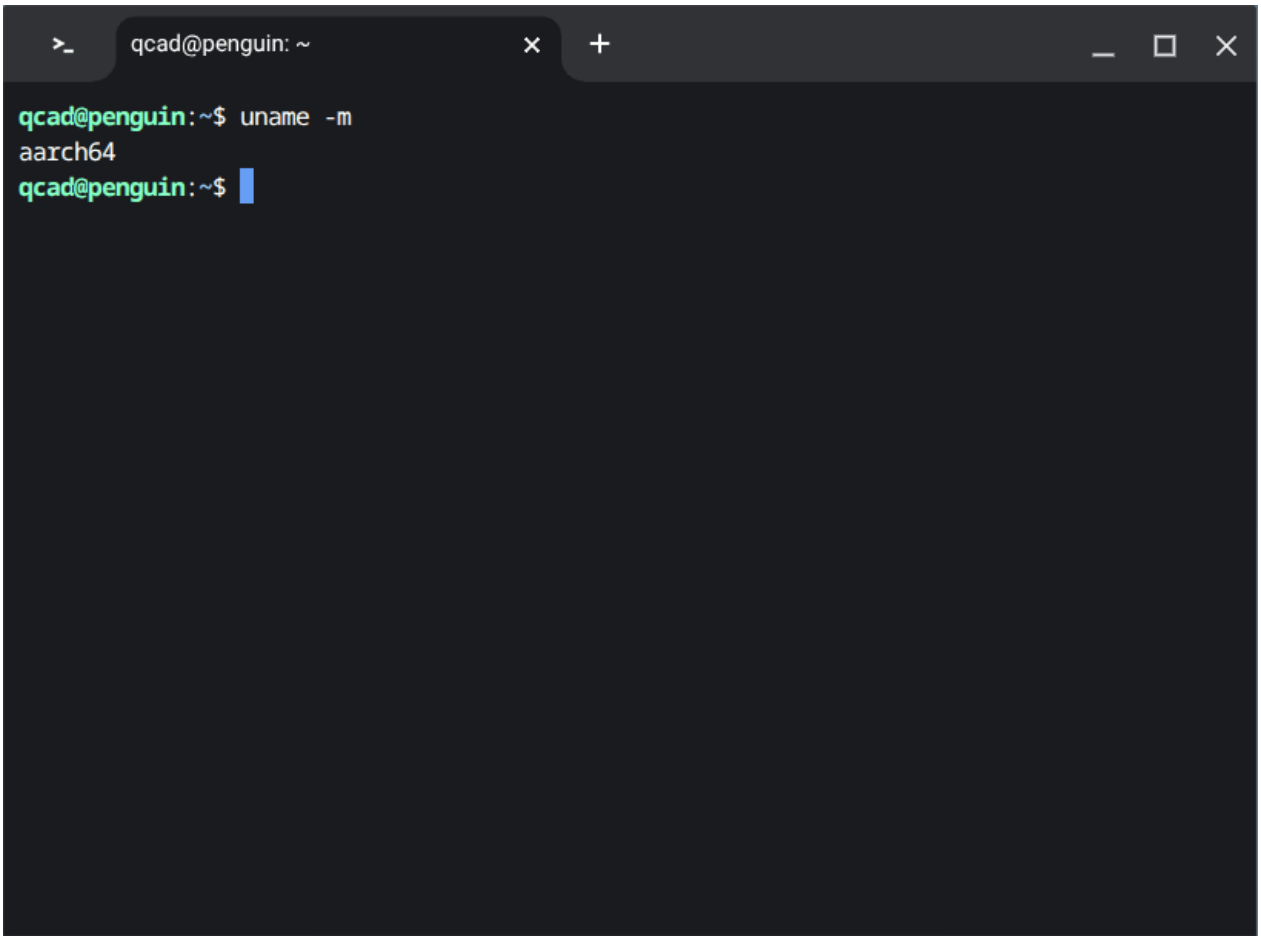
Hit Enter, enter your password and hit Enter again.

Finding Out the CPU Type

Next, we need to find out which CPU your computer is based on. In the terminal, enter:

```
uname -m
```

This displays your CPU type, for example `aarch64` or `x86_64`:



If your computer displays `aarch64`, you have a 64-bit ARM CPU. If it displays `x86_64`, you have a 64-bit Intel or AMD CPU. Keep in mind which CPU type you have.

Installing the Correct Runtime

If your CPU is `aarch64`:

```
sudo flatpak install flathub org.freedesktop.Platform/aarch64/25.08
```

If your CPU is `x86_64`:

```
sudo flatpak install flathub org.freedesktop.Platform/x86_64/25.08
```

Downloading the QCAD Package

Open the download area using the link you have received after your purchase.

Scroll down to the downloads with the `.flatpak` ending: Linux > Qt 6 > 64-bit Intel if you have an Intel or AMD CPU, or Linux > Qt 6 > ARM (64-bit) if you have an ARM CPU.

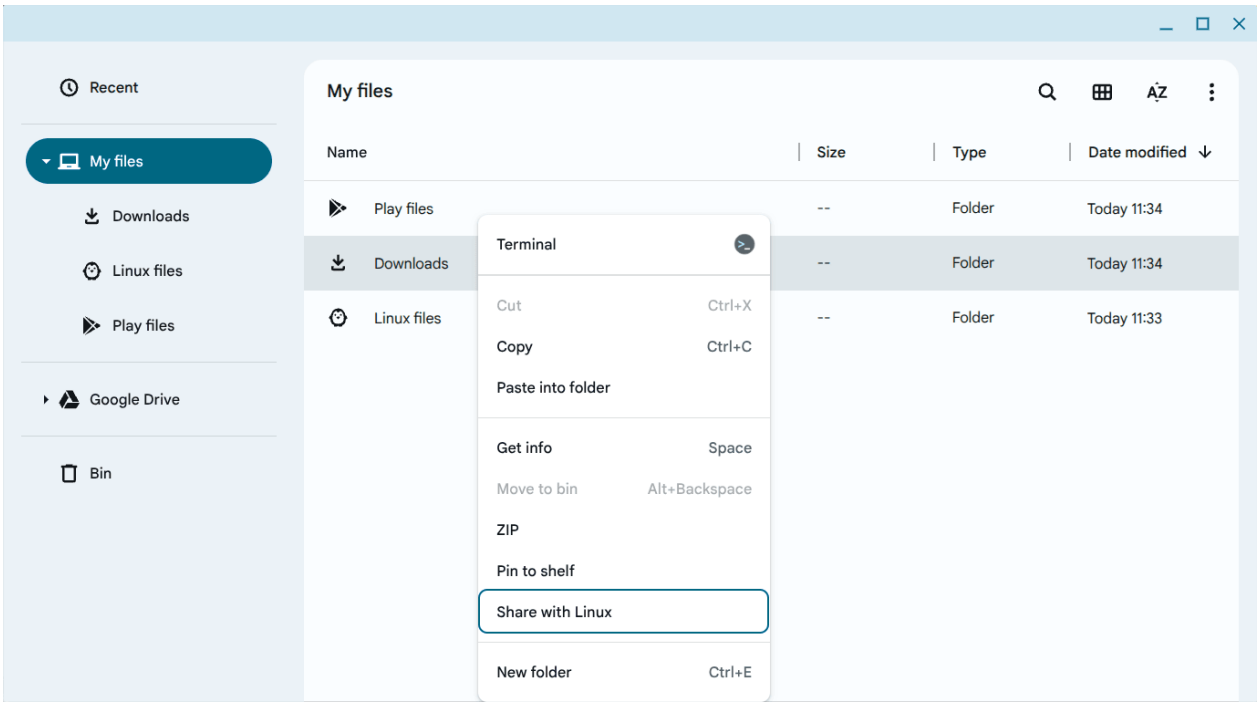
Below the title you have scrolled to, click on the item with "Flatpak" in its name to download the Flatpak file for that platform. The Chrome browser shows a file dialog and suggests saving the file in your Downloads folder. This is fine, so click *Save*.

Wait for the download to complete.

Sharing the Downloads Folder with Linux

The Linux environment cannot access your ChromeOS files until you share them explicitly.

Open the *Files* app (blue folder icon) from the shelf at the bottom of the screen. On the right hand side, right-click on *Downloads* and choose *Share with Linux*:



Close the Files app.

Installing QCAD

Go back to the terminal app or open it again. In the terminal, navigate to the shared Downloads folder:

```
cd /mnt/chromeos/MyFiles/Downloads
```

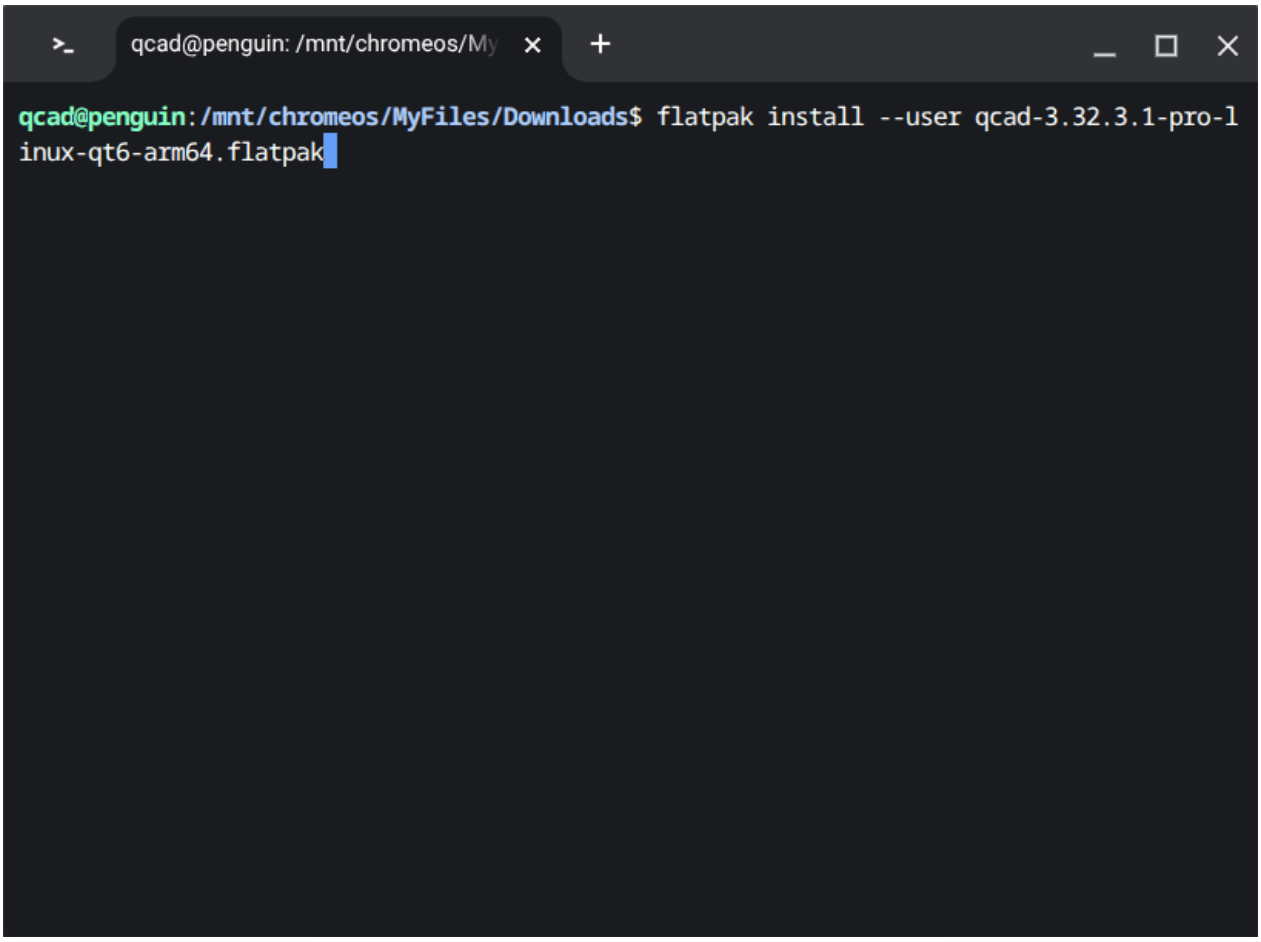
If you have an Intel or AMD CPU, enter:

```
flatpak install --user qcad-3.x.x-pro-linux-qt6-x86_64.flatpak
```

If you have an ARM64 CPU, enter:

```
flatpak install --user qcad-3.x.x-pro-linux-qt6-arm64.flatpak
```

Hit Enter. Remember to replace `3.x.x` with the exact version number of the package you have downloaded.



The terminal prompts you:

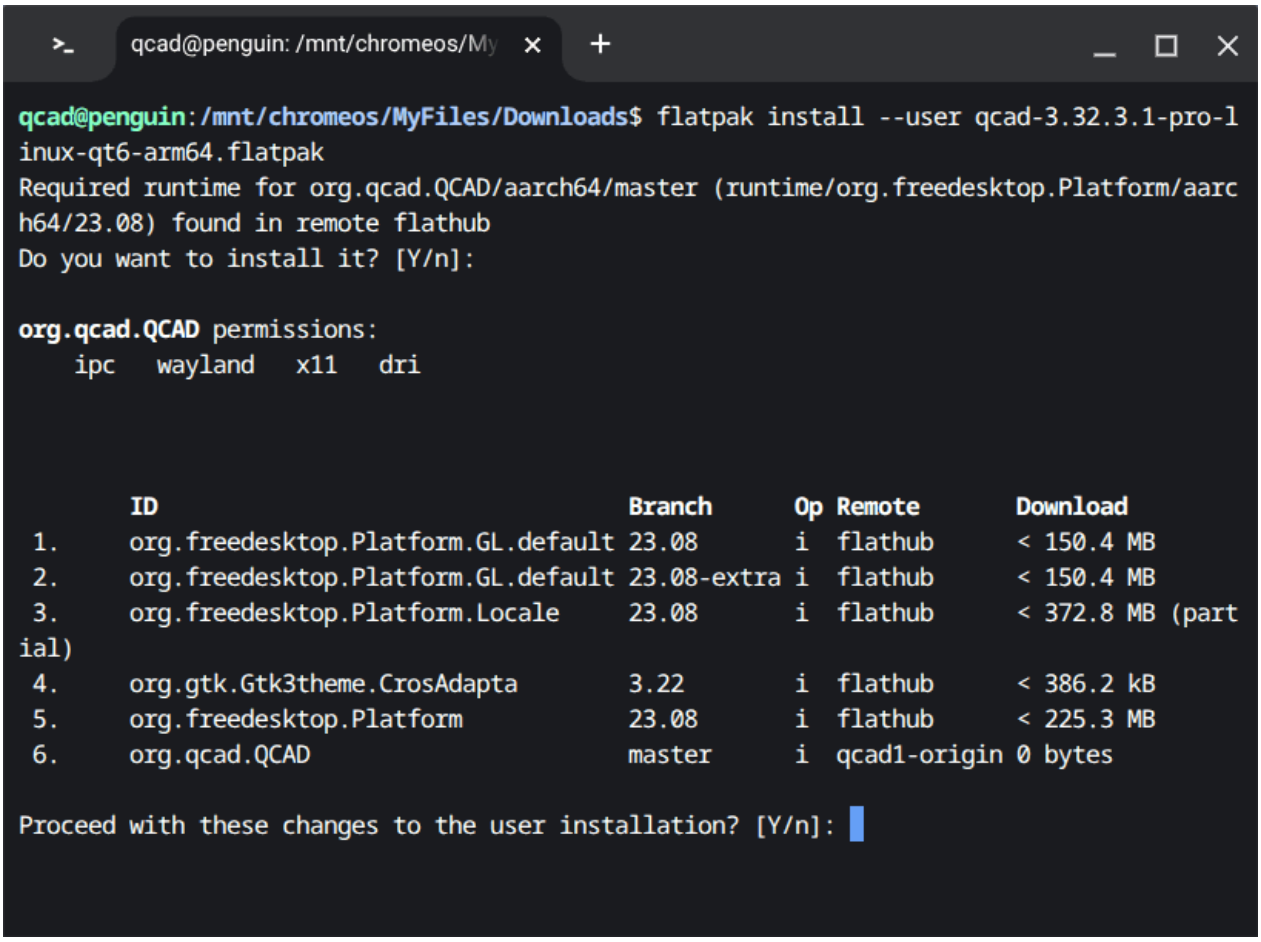
```
Required runtime for org.qcad.QCAD/... (runtime/org.freedesktop.Platform/...) found in
 remote flathub
Do you want to install it? [Y/n]:
```

Hit Enter.

The terminal then prompts you:

```
Proceed with these changes to the user installation? [Y/n]:
```

Hit Enter to confirm.



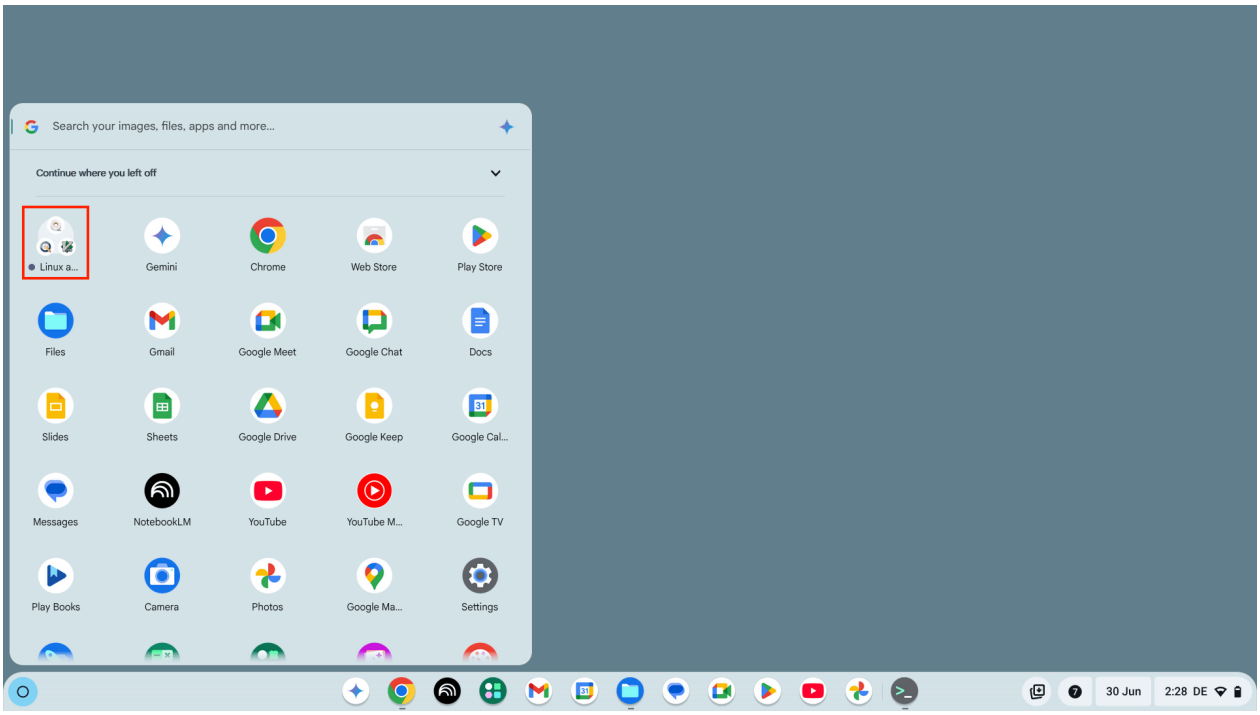
Depending on your Internet speed, this takes a few minutes. When the installation has finished, the terminal shows "Installation complete."

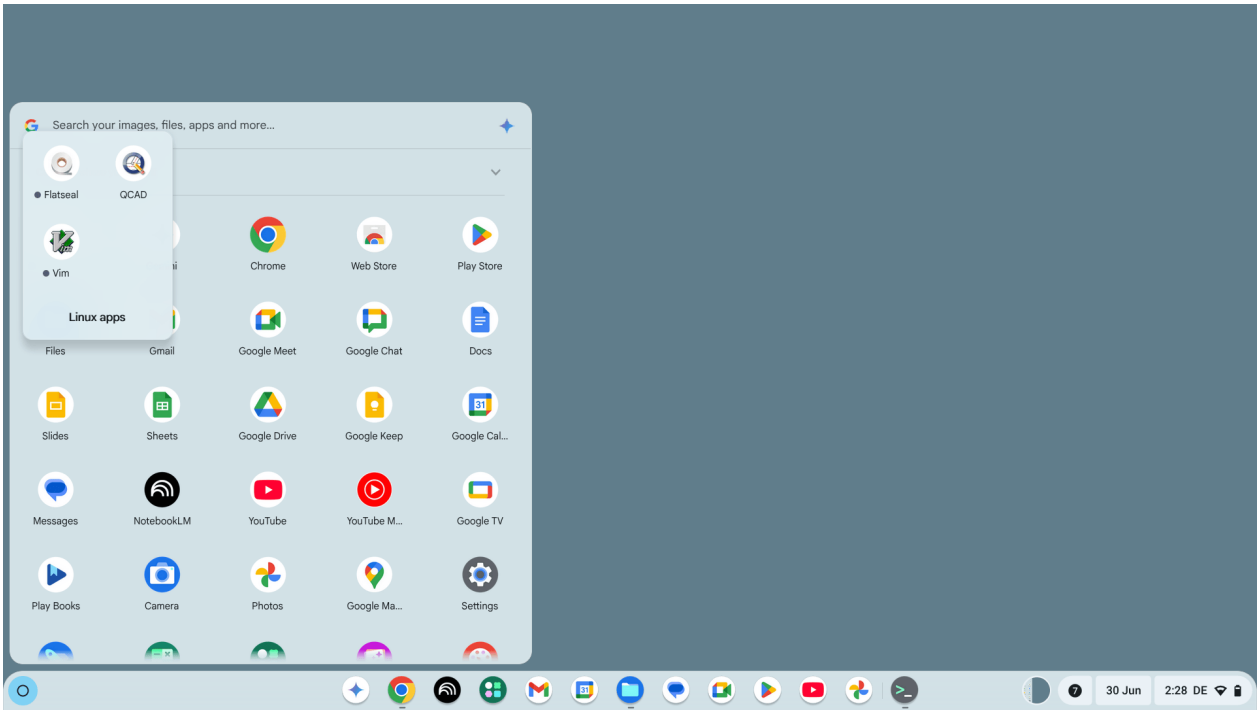
Running QCAD

To start QCAD from the terminal, enter:

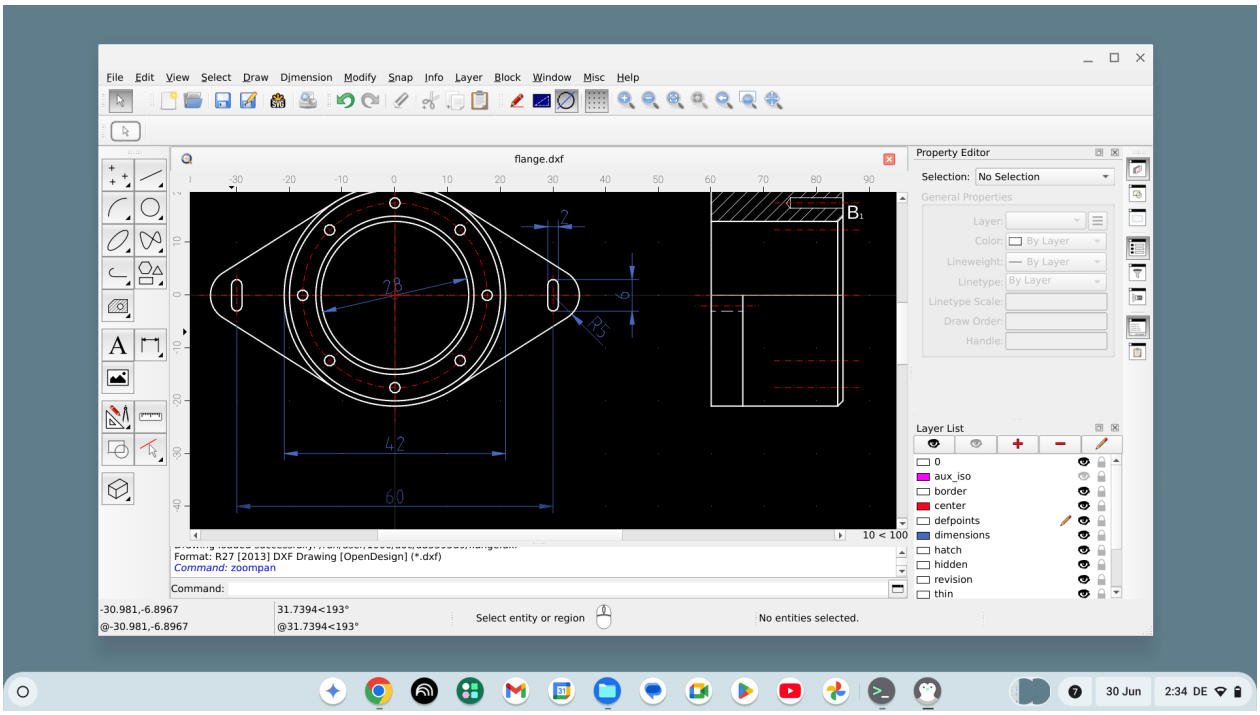
```
flatpak run org.qcad.QCAD
```

Alternatively, use the Launcher, where you can find QCAD in the *Linux apps* folder:





QCAD starts with its welcome screen where you can choose some initial settings. Once you confirm, the QCAD application window is shown and you can start working:



Uninstalling QCAD on ChromeOS

To remove QCAD again, enter the following in the terminal:

```
flatpak uninstall --user org.qcad.QCAD
```

16.5.2 Installing QCAD on macOS

This section is for users of the macOS operating system. The setup file you need depends on your macOS version and on the processor type of your Mac.

For brevity, version numbers are shown as `3.x.x` below. Replace this with the actual version number of the QCAD release you downloaded, for example `3.28.1`.

Finding your macOS Version and Processor Type

- Click the Apple icon in the top left corner of your screen.
- Choose *About This Mac*. The dialog shows the macOS version number underneath or beside the label *macOS*, for example 13.2. Write it down.
- The processor type is shown beside the label *Processor* or *Chip*. Older Macs have an **Intel** CPU, newer ones an **Apple M1, M2, M3, M4 or M5** CPU (Apple Silicon). Note which one you have.

Getting the Correct Setup File

Installation from a USB Flash Drive

Insert the USB flash drive and double-click its icon on the desktop to show its contents.

- If you have an Apple **M1, M2, M3, M4 or M5** processor, install `installation/3.x/macOS/qcad-3.x.x-pro-macos-11-13-qt6-arm64.dmg`
- If you have an **Intel** processor, the file depends on your macOS version:
  - macOS 10.14, or 11.x / 12.x / 13.x: `installation/3.x/macOS/qcad-3.x.x-pro-macos-10.14-13.dmg`

- macOS 10.10, 10.11, 10.12 or 10.13: `installation/macOS/qcad-3.x.x-pro-macos-10.10-10.13.dmg`
- For a lower or higher version, use the installer that covers that version.

Installation from Download

If you purchased the downloadable version, use your browser to access your personal download area (see the e-mail you received after your purchase).

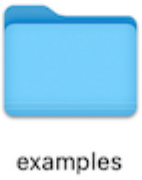
- Apple **M1/M2/M3/M4/M5**: click the link named *"Download for ARM (M1, M2, M3, M4, M5)"*
- **Intel**: click the link labelled *"Download for 64bit Intel"*

After the download has finished, use Finder to locate the file on your hard disk.

Installation

Double-click the file you located in the previous section. It is a disk image that contains the QCAD application. Double-clicking shows its contents:





Drag the QCAD application icon on the left to the Applications folder that is shown on the right in the same Finder window.

Launching QCAD

QCAD is now installed. To launch it, open the Applications folder and double-click the QCAD application icon.

Uninstalling

To remove QCAD, drag the QCAD icon from your Applications folder to the Trash in your dock.

16.5.3 QCAD Portable

QCAD can be installed on a portable drive or USB device without installing it on the computer's hard drive. This allows you to carry QCAD with you and run it on different computers.

To make sure that all configuration files are stored on the portable device instead of on the host computer, QCAD has to be started with the `-config` command line argument, specifying a configuration directory on the portable device.

Windows

Extract the `.zip` package to your portable device and start QCAD with:

```
F:\QCAD\qcad.exe -config F:\config
```

Linux

Extract the `.tar.gz` package to your portable device and start QCAD with:

```
/mnt/myusbstick/qcad/qcad -config /mnt/myusbstick/config
```

macOS

Copy the `.app` bundle from the `.dmg` file to your portable device and start QCAD with:

```
/Volumes/MyUSBStick/QCAD-Pro.app/Contents/MacOS/QCAD-Pro -config /Volumes/MyUSBStick/config
```

16.5.4 Installing QCAD under Windows

System Requirements

- Windows 10 or Windows 11 (64-bit) for the recommended packages
- 1 GHz processor or faster
- 2 GB RAM
- 260 MB of free disk space
- Screen resolution of 1024x768 or higher

Packages for older Windows versions (XP, Vista, 7, 8, 8.1) are still offered as legacy downloads. They are based on an older Qt version, do not receive new features and should only be used if the recommended package cannot be installed.

Choosing the Right Package

The download page offers several variants of the same QCAD release:

- **64-bit, Qt 6** (recommended): for Windows 10 and Windows 11
- **64-bit, Qt 5**: for Windows 7, 8 and 8.1
- **32-bit, Qt 5**: for 32-bit installations of Windows 7, 8 and 8.1
- **32-bit legacy**: for Windows XP and Windows Vista

If you are unsure whether your Windows installation is 32-bit or 64-bit, open *Start* > *Settings* > *System* > *About* and check the entry *System type*.

For each variant, two formats are available:

- an **installer** (`.msi`), which installs QCAD on your computer in the usual way
- a **ZIP archive** (`.zip`), which contains the ready-to-run application folder and can be extracted anywhere without installing anything, for example onto a USB drive (see *QCAD Portable*)

Downloading QCAD

Trial Version

The trial version of QCAD Professional and QCAD/CAM can be downloaded free of charge from:

https://qcad.org/en/download

The installers of the trial version are named according to the pattern:

```
qcad-<version>-trial-win64-qt6-installer.msi
qcadcam-<version>-trial-win64-qt6-installer.msi
```

Purchased Version

If you have purchased QCAD Professional or QCAD/CAM, you have received an e-mail with a personal download link after purchase. Use this link to download the installers.

If you have lost or never received your download link, you can recover it at:

https://qcad.org/qcad-installer-recovery

The installers of the purchased version are named according to the pattern:

```
qcad-<version>-pro-win64-qt6-installer.msi
qcadcam-<version>-pro-win64-qt6-installer.msi
```

QCAD does not use license keys, activation codes or online registration. The package you download from your personal download link *is* the licensed full version.

Browser Download Warnings

Windows or your browser may warn you that the downloaded file *isn't commonly downloaded* and that you should make sure you trust it before opening it. This usually means that we have recently released a new QCAD version and the new download has not yet established a reputation. You can safely ignore this warning or wait about two weeks.

To ignore the warning, click ... next to the warning and choose *Keep*. Then click the down arrow next to the *Delete* button and choose *Keep anyway*. You can then open the file from your Downloads folder or directly from the browser.

From a USB Drive or DVD

If you received QCAD on a physical medium, open the medium in the Windows File Explorer and navigate to the folder for your platform, for example `installation\windows\64bit`, and locate the `.msi` file there.

Installing QCAD

- Double-click the downloaded `.msi` file.
- Confirm the User Account Control prompt. Administrator rights are required because QCAD is installed for all users of the computer.

The prompt shows the verified publisher *RibbonSoft GmbH*. All QCAD installers and program files are digitally signed, so this entry can be used to verify that the package you downloaded is genuine and unmodified.

- Follow the instructions of the setup wizard. The setup is available in several languages and starts in English by default.
- The default installation folder is `C:\Program Files\QCAD` (`C:\Program Files\QCADCAM` for QCAD/CAM). It can be changed in the setup wizard.
- No restart is required after the installation.

The installer creates a start menu entry and a desktop icon, and registers QCAD as an application for `.dxf` and `.dwg` files.

If you do not have administrator rights on your computer, use the ZIP archive instead of the installer. It can be extracted into any folder you have write access to, for example into your user folder, and does not require an installation.

Starting QCAD

After the installation, QCAD can be started:

- from the desktop icon, or
- from the start menu under *Start* > *QCAD* > *QCAD*

The application itself is `qcad.exe` in the installation folder.

When QCAD is started for the first time, a dialog is shown where the language of the user interface can be set (see *User Interface Language*).

Trial Mode

The trial packages contain QCAD together with a trial of the QCAD Professional or QCAD/CAM add-ons. The trial runs for 15 minutes at a time and can then be restarted as often as you like.

If you decide not to purchase a license, you can remove the add-on running in trial mode and continue to use the free QCAD Community Edition: click *Remove* in the *Trial* widget and follow the instructions on screen.

Updating QCAD

To update to a newer version, download the new installer and run it. The previous installation is replaced. Your settings, drawings and part library remain untouched, since they are stored in your user profile and not in the program folder.

When switching between different variants of the same release, for example from the 32-bit to the 64-bit package or from the Qt 5 to the Qt 6 package, uninstall the previous version first.

Uninstalling QCAD

QCAD can be uninstalled using the standard Windows tools. It is listed there under its full product name and version, for example *QCAD Professional 3.32.9*, published by *RibbonSoft GmbH*:

- Windows 11: *Start* > *Settings* > *Apps* > *Installed apps*, then click the three dots next to the QCAD entry and choose *Uninstall*
- Windows 10: *Start* > *Settings* > *Apps* > *Apps & features*, then click the QCAD entry and *Uninstall*
- Alternatively: *Control Panel* > *Programs* > *Uninstall a program*

Uninstalling QCAD does not remove your personal settings. To remove these as well, see *Resetting the QCAD Configuration*.

Where QCAD Stores Your Settings

The configuration file and the user data folder are stored in your user profile (where "username" is your login name):

```
C:\Users\username\AppData\Roaming\QCAD\QCAD3.ini
C:\Users\username\AppData\Roaming\QCAD\QCAD
```

The folder `AppData` is hidden by default. It can be reached by entering `%APPDATA%\QCAD` in the address bar of the Windows File Explorer.

Both paths are also shown in QCAD under *Help* > *About*.

A different configuration file can be used by starting QCAD with the `-config` command line argument. This is mainly useful for portable installations (see *QCAD Portable*).

16.6 Layers

Layers are one of the most important tools for organizing a drawing, letting you group related entities and control their visibility and appearance together. This section explains how to work with layers, how to store and restore their settings using layer states, and how layers and blocks interact to determine an entity's effective attributes such as color, linetype and lineweight.

16.6.1 Layers, Blocks and Attributes

Introduction

In CAD, the attributes of an entity (color, linetype, lineweight) are usually controlled through layers. Entities placed on a colored layer inherit that layer's color. The same principle applies to entities inside a block.

The special attribute value *By Block* lets an entity take on the attributes of the layer on which its containing block is inserted, rather than the attributes of its own layer.

Example

The following example shows a single block reference inserted on layer "2":



The block contains four triangles with text, each set up differently:

- The **top** triangle and text are on layer "0" with the color fixed to *Green*. They stay green regardless of where the block is inserted.
- The **second** triangle and text are on layer "1" with the color set to *By Layer*. They are shown in red, the color of layer "1".
- The **third** triangle and text are on layer "2" (blue) with the color set to *By Layer*. They inherit the blue color of layer "2".
- The **bottom** triangle and text are on layer "2" with the color set to *By Block*. Instead of the color of their own layer, they inherit the color of the layer on which the block reference is inserted.

When the same block reference is instead inserted on an orange layer, only the *By Block* entities change. They now appear orange, following the block reference's insertion layer:



Notes

The property values *By Layer* and *By Block* are not only available for colors, but also for the other entity attributes lineweight and linetype.

16.6.2 Working with Layer States

A layer state stores the state (attributes) of all layers in a drawing at a given moment.

Introduction

Layer states are an advanced concept that can be useful if you operate on drawings with many layers or if you require different layer settings for different uses of your drawing. A drawing might for example use one layer state for printing and another one for bitmap exports or SVG exports. A layer state stores all layers with their attributes (color, linetype, lineweight, etc) at the moment the layer state is created or updated. When a layer state is restored, all layer attributes are restored to the attributes stored in the layer state.

Example

Consider this example drawing of a floorplan:



This drawing was created with the dark background of a typical CAD model space view in mind. The walls are solid filled to make them well visible and the windows are shown in a bright cyan.

When it comes to printing, we might want to replace the solid fill of the walls with a darker hatch pattern and make the window layer darker as well. Further, we might want to quickly switch to a

version without the interior elements and with darker colors, perhaps for a bitmap export for a website on a bright background.

We create our drawing with the following layer structure:

- Walls (contains the solid fills of the walls)
- Walls Hatch (contains a hatch for the walls)
- Windows (contains all windows)
- Interior (contains the furniture and other interior elements)

Before thinking about alternative layer states, the current ("normal") layer state should always be saved, to make it easy to switch back to this state at any time.

Creating a Layer State

1. Choose *Layer* > *Layer States* > *Add...*
2. A dialog is shown to enter the name of the layer state. Make sure to name your layer states in a meaningful way to avoid having to look at all layer states to find the one you need. We name this layer state in this example "Normal".

We can now modify the layers of our drawing to create the desired look for printing. In this example, we switch off layer "Walls" and switch in layer "Walls Hatch" instead. We also adjust the colors and linetypes of our layers as desired. Once we are happy with the look, we store this layer state under the name "Printing".

Additional layer states can be created in the same way, for example a layer state for bitmap export for a web site perhaps without interiors, etc.

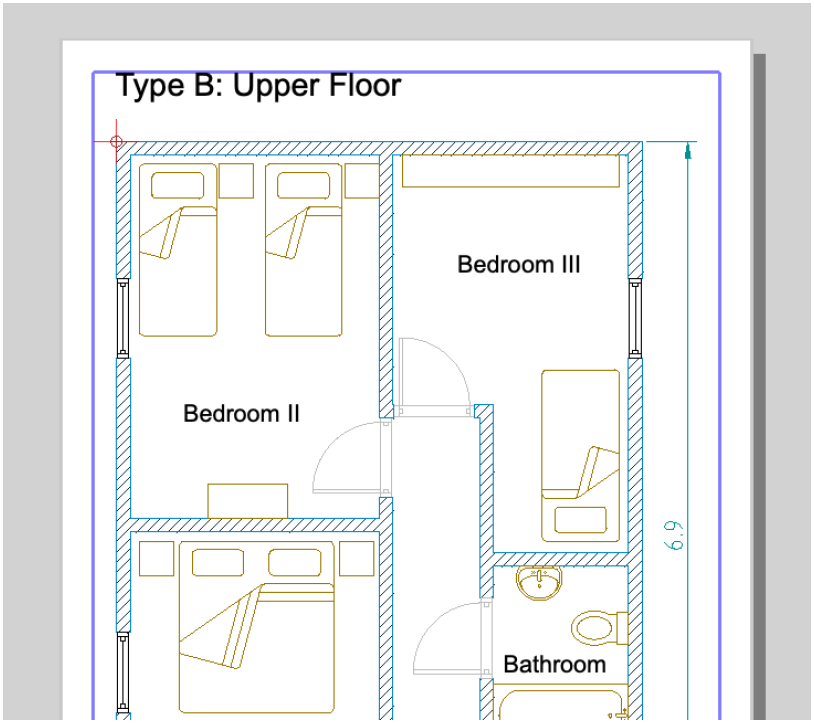
Activating a Layer State

You can now switch between layer states using *Layer* > *Layer States* > *[Layer State Name]* > *Activate*. In our example, we can switch between the layer states "Normal", "Printing" and "Webpage":

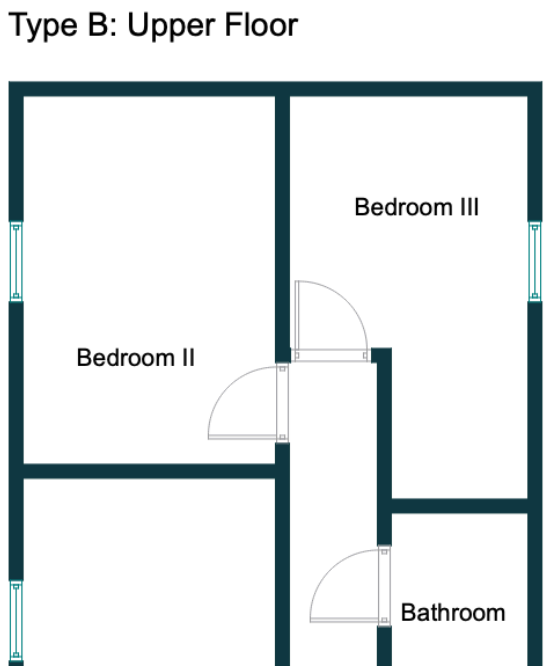
Layer State "Normal" (bright colors, solid fills):



Layer State "Printing" (hatch instead of solid fill, darker colors):



Layer State "Webpage" (darker colors, interior layer switched off):



Updating a Layer State

When a drawing is updated, new layers might be added or layers might be removed or changed. To keep the layer states up to date, you can update an existing layer state using the menu *Layer* > *Layer States* > *[Layer State Name]* > *Update*.

A typical workflow would be to activate a layer state, adjust the layers as desired and then update the stored layer state.

Renaming a Layer State

To rename a layer state, choose *Layer* > *Layer States* > *[Layer State Name]* > *Rename*.

Removing a Layer State

To remove a layer state, choose *Layer* > *Layer States* > *[Layer State Name]* > *Remove*.

16.6.3 Working with Layers

Layers in QCAD define how your entities look like (color, line type and lineweight).

Introduction

Layers are a fundamental concept in CAD. Each line, arc, etc. you draw resides on a layer. Layers are part of your drawing and stored with your drawing, so every drawing has its own set of layers.

All entities with the same type of real world function are typically residing on the same layer. Since the layer defines the look of all entities on it, only entities with the exact same look are usually put on the same layer.

For a mechanical drawing, you might create a layer "hidden" for hidden edges (thin, black, dashed) or "center" for center lines (thin, red, dash-dotted).

For an architectural drawing, you might create a layer for interior walls, one for exterior walls, one for electrical installations, etc.

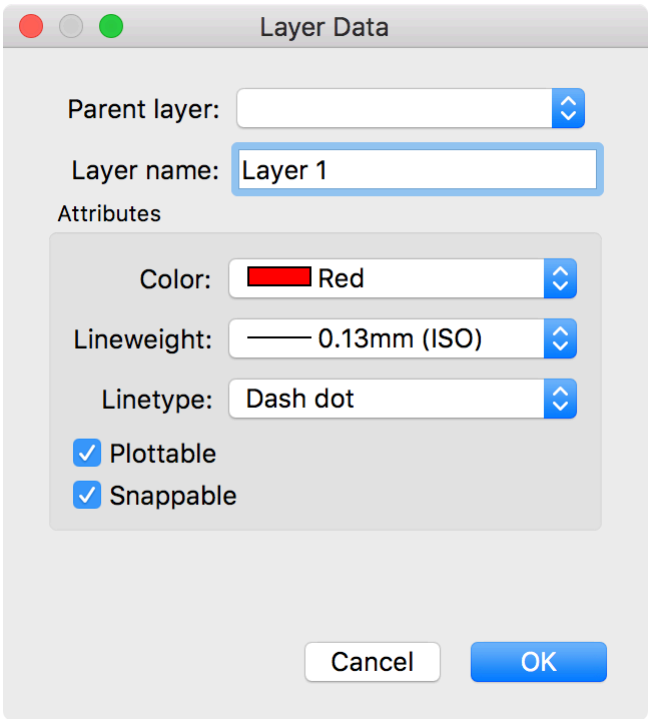
Step 1: Showing the Layer List

When working with layers, make sure that the layer list is visible. If that is not the case choose menu *View* > *Layer List*.

The layer list shows all layers of the current drawing. It also shows which layer is currently active - that means which layer you are drawing on. Layers can also be added, deleted, switched on / off, etc. all from the layer list.

Step 2: Creating a Layer

Use the *Layer* > *Add Layer* tool to create a new layer. Call the layer "Layer 1" and give it a color (red), lineweight (0.13mm) and linetype (dash dot):

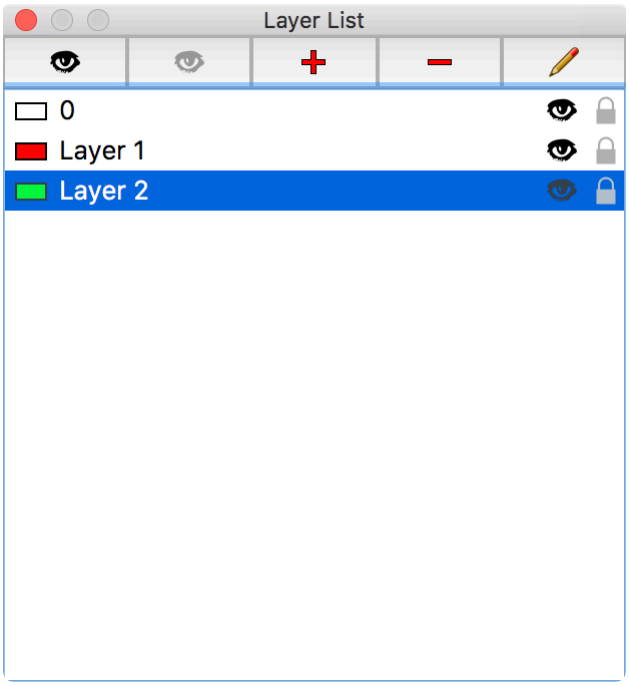


In the same way, add another layer "Layer 2" with different attributes (e.g. "green", 0.2mm, Continuous).

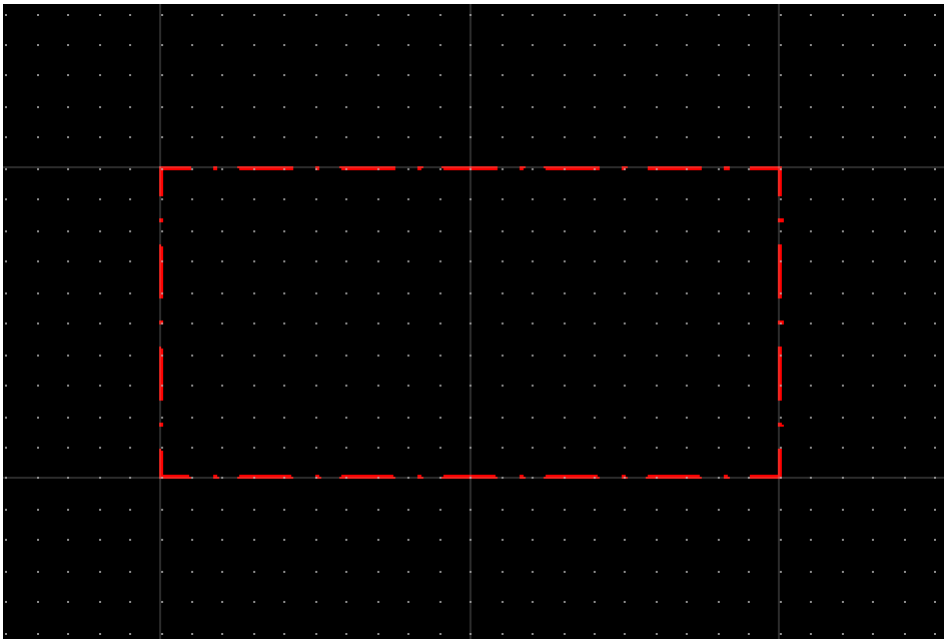
The default color and lineweight for new layers and for layer "0" in new drawings can be configured under *Edit* > *Application Preferences* > *Layer* > *Add Layer*.

Step 3: Using Layers

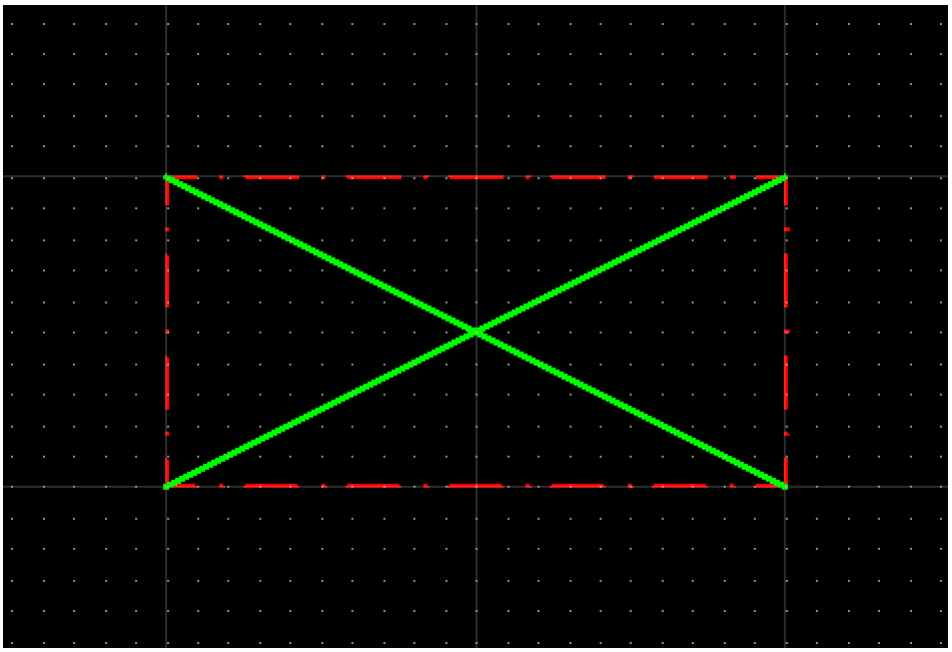
The layer list always shows you which layer is currently active: the one that is highlighted in the list. To activate a different layer, simply click on it in the layer list.



Click on layer "Layer 1" and draw some lines. The lines are drawn in red, with the attributes previously chosen for that layer.



Click on layer "Layer 2" and draw some more lines. These lines are drawn in green:

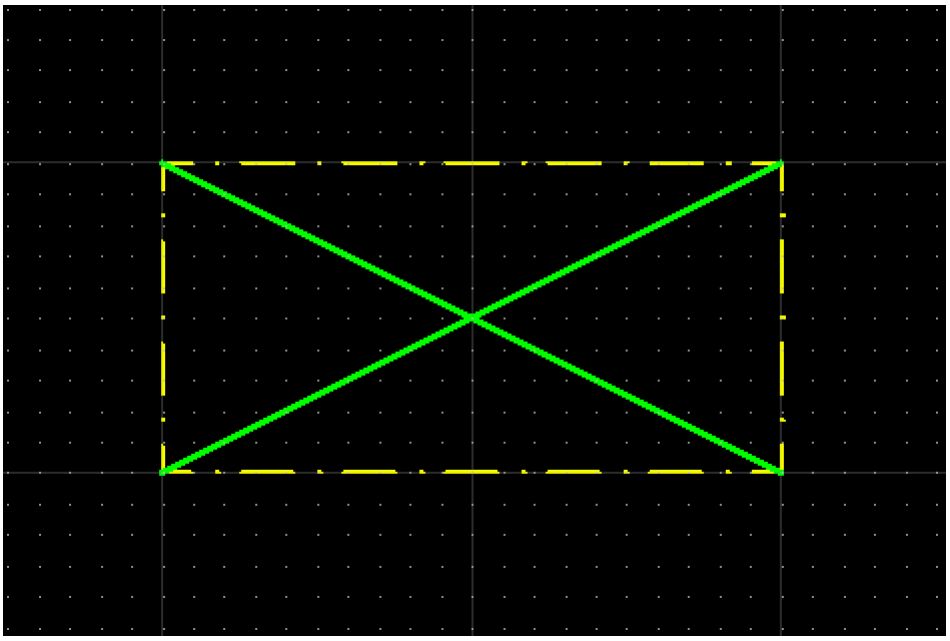


Step 4: Changing Layers

Double-click "Layer 1" in the layer list. The layer dialog is shown again, this time to allow you to edit the layer. You can change the layer name or its attributes.

Change the layer colour to "Yellow" and click OK.

As you can see, the lines that were previously drawn while "Layer 1" was active are now yellow:



Layers allow you to manage the attributes of all related entities at once.

Step 5: Hiding Layers

Click the eye icon next to "Layer 1". It turns grey, meaning that this layer is now turned off. All lines that reside on "Layer 1" are now hidden. They are still part of your drawing and will also be saved. They are just not displayed anymore. This can be useful to hide certain information in a drawing (for

example electrical installations) while working on other parts of the drawing (for example interior walls).

Troubleshooting

Linetypes and Lineweight not Shown

Problem

My entities are not showing any pattern or thickness.

Explanation

Lineweights and linetypes (patterns) are defined at relatively small sizes (a few Millimeters / a fraction of an Inch). If the drawing size is much larger, linetypes and lineweights might not be visible in the drawing model (i.e. the *Model_Space block). Linetypes and lineweights are still visible in the print preview where they are adjusted to print correctly on paper.

Solutions

The solution depends on your working habits: are you printing your drawing directly using *File* > *Print Preview* and *File* > *Print* or are you working with layout blocks (paper spaces) and viewports.

If you are printing your drawing without layout blocks or viewports, you might want to switch on screen-based linetypes to visualize linetypes while working in model space: *View* > *Screen-based Linetypes*. This can also be made a permanent preference under *Edit* > *Application Preferences* > *Graphics View* > *Appearance* > *Auto switch to screen based linetypes in model view*.

If you are working with layout blocks (paper spaces) and viewports, the drawing linetype scale can be increased under *Edit* > *Drawing Preferences* > *General* > *Linetype* > *Global linetype scale*. When working in Meters, set the global linetype scale to 1000.

16.7 MCP

QCAD can be connected to AI assistants through the Model Context Protocol (MCP). Once connected, an assistant such as Claude can read the drawing that is open in QCAD, answer questions about it, check it for problems, draw or modify geometry and export the result. This section explains how to set up and use the QCAD MCP server.

16.7.1 Using QCAD with Claude Desktop

Introduction

QCAD contains an MCP (Model Context Protocol) server. MCP is an open standard that lets AI assistants use tools provided by other applications. With the QCAD MCP server, the Claude Desktop app can work directly with the drawing that is currently open in QCAD. For example, you can ask Claude to:

- describe an unfamiliar drawing, its layers, blocks, dimensions and text
- check a drawing for problems such as unclosed polylines, duplicate entities or geometry on layer 0
- measure distances, lengths and areas
- create a bill of materials from block references
- add layers, lines, arcs, circles, polylines, text, dimensions and hatches
- move, rotate, scale, trim, round or offset existing entities

- export the drawing as PNG, PDF, SVG, DXF or DWG

This tutorial explains how to connect QCAD to Claude Desktop under Windows or macOS.

Requirements

- QCAD with the MCP plugin, installed and running on the same computer as Claude Desktop
- The Claude Desktop app for Windows or macOS

The QCAD extension for Claude Desktop comes with QCAD. It is the file `qcad.mcpb` in the folder `mcp` of your QCAD installation, for example:

- Windows: `C:\Program Files\QCAD\mcp\qcad.mcpb`
- macOS: `/Applications/QCAD-Pro.app/Contents/Resources/mcp/qcad.mcpb`

Under macOS, the folder is inside the application package and is not shown by the Finder when you open the `Applications` folder. *Misc* > *MCP* > *Install in Claude Desktop* in QCAD opens or shows the file for you (see below).

Node.js does not have to be installed separately. Claude Desktop comes with everything that is needed to run the extension.

Starting the MCP Server in QCAD

The MCP server in QCAD does not start by itself. Until it is started, QCAD does not accept any connections.

- Start QCAD.
- Choose *Misc* > *MCP* > *Start*.

The command line at the bottom of the QCAD window confirms that the server is running:

```
MCP: MCP server running on port 3907 (http://127.0.0.1:3907/mcp)
```

The server keeps running until you choose *Misc* > *MCP* > *Stop* or quit QCAD. It has to be started again every time QCAD is started.

The menu *Misc* > *MCP* also offers:

- *Install in Claude Desktop*: helps you install the QCAD extension in Claude Desktop (see below)
- *Status*: shows on the command line whether the server is running and on which port
- *Restart*: stops and restarts the server, for example after changing its port

The server only accepts connections from the same computer (address `127.0.0.1`). It cannot be reached from other computers on your network.

Installing the QCAD Extension in Claude Desktop

The extension is installed only once. Make sure Claude Desktop is installed and you are logged in before you start.

The QCAD extension is a local extension: Claude Desktop runs it on your computer, where it connects to the MCP server in QCAD. Do not use *Add custom connector* in Claude for this. Custom connectors only accept servers that are reachable from the internet through an `https` address, so the address `http://127.0.0.1:3907/mcp` of QCAD is not accepted there.

Windows

- In QCAD, choose *Misc* > *MCP* > *Install in Claude Desktop*. QCAD shows the file `qcad.mcpb` selected in the File Explorer. The command line in QCAD also shows the steps to install it.

- In Claude Desktop, open the menu at the top left of the window and choose *File* > *Settings*.
- Go to *Extensions*, open *Advanced settings* and click *Install Extension*.
- Choose the file `qcad.mcpb`, for example `C:\Program Files\QCAD\mcp\qcad.mcpb`.

Under Windows, do not double-click the file `qcad.mcpb` and do not drag it into the Claude Desktop window. Neither installs the extension: Claude Desktop does not register itself for this file type, so a double-click only makes Windows ask which app to use to open the file.

macOS

- In QCAD, choose *Misc* > *MCP* > *Install in Claude Desktop*. QCAD opens the file `qcad.mcpb` directly in Claude Desktop.
- If Claude Desktop cannot be found, QCAD shows the file in the Finder instead. Drag it into the Claude Desktop window, or choose *Claude* > *Settings* from the menu bar in Claude Desktop, go to *Extensions*, open *Advanced settings*, click *Install Extension* and choose the file.

Completing the Installation

Claude Desktop shows the details of the QCAD extension. Click *Install* and confirm.

After the installation, *QCAD* appears in the list of extensions under *Settings* > *Extensions*. Make sure that it is enabled.

Extension Settings

The QCAD extension has three settings, which can be changed in Claude Desktop under *Settings* > *Extensions* > *QCAD*. Usually, the defaults do not need to be changed:

- **QCAD host**: the address of the QCAD MCP server. Leave this at `127.0.0.1`.
- **QCAD MCP port**: the port of the QCAD MCP server, by default `3907`. This must match the port configured in QCAD.
- **Reply timeout (ms)**: how long Claude waits for an answer from QCAD, by default 60000 ms (one minute). Increase this value if you work with very large drawings.

Working with Claude and QCAD

- Open a drawing in QCAD, or start with a new drawing.
- Make sure the MCP server is running (*Misc* > *MCP* > *Status*).
- Open a new chat in Claude Desktop and ask for what you need, for example:

```
Describe the drawing that is open in QCAD.
```

```
Check the drawing in QCAD for problems and list what you find.
```

```
Create a bill of materials from the block references in the drawing.
```

```
Draw a rectangle of 100 x 50 mm on a new layer "Outline" and round all corners with a radius
 of 5 mm.
```

Claude decides which QCAD tools to use. The first time a tool is used, Claude Desktop asks for your permission. You can allow the tool once or always.

Claude always works with the drawing that is currently active in QCAD. If several drawings are open, switch to the right one in QCAD before asking.

Changes made by Claude appear immediately in the QCAD window. Tools that modify or delete existing entities, rename layers or purge unused layers and blocks only report what they would

change, unless Claude explicitly asks them to apply the change. If you want to review a change first, ask Claude to show you what it would do before applying it. Every change can be undone in QCAD with *Edit* > *Undo*, or by asking Claude to undo it.

Claude can also look at the drawing: it renders the drawing as an image to check what it shows or to verify the result of a change.

Voice Mode

MCP is not available in the voice mode of Claude Desktop. The QCAD tools can only be used in text chats. To work with QCAD, type your requests instead of speaking them.

Privacy

The QCAD MCP server and the extension only communicate with each other on your own computer. However, everything Claude reads from your drawing, for example layer names, text, dimensions and coordinates, is sent to Claude as part of the conversation, like any other content you share in a chat. Do not use the MCP server with drawings that must not leave your computer.

Troubleshooting

Windows Asks Which App to Use to Open qcad.mcpb

Claude Desktop for Windows cannot open `.mcpb` files by double-click. Close the dialog and install the file in Claude Desktop under *File* > *Settings* > *Extensions* > *Advanced settings* > *Install Extension* instead, as described above.

Claude Does Not Accept the Address of QCAD as a Connector

Add custom connector in Claude only works with servers that are reachable from the internet. QCAD is connected through the QCAD extension instead, as described above.

Claude Reports "Cannot reach QCAD"

The extension cannot connect to QCAD. Check that:

- QCAD is running
- the MCP server has been started with *Misc* > *MCP* > *Start*
- the port shown by *Misc* > *MCP* > *Status* matches the port in the extension settings in Claude Desktop

QCAD does not have to be started before Claude Desktop, and Claude Desktop does not have to be restarted after starting the MCP server. However, if Claude does not seem to know about the QCAD tools, start QCAD and its MCP server first and then open a new chat in Claude Desktop.

The Server Cannot Be Started

If *Misc* > *MCP* > *Start* reports that the address is already in use, another program, or another QCAD window, is already using port 3907. Only one QCAD window can run the MCP server on a given port. Stop the server in the other QCAD window with *Misc* > *MCP* > *Stop* and start it in the window you want to work in.

Claude Works on the Wrong Drawing

Claude always works with the active drawing of the QCAD instance that runs the MCP server. Click on the tab or window of the drawing you want Claude to work with in QCAD. If several QCAD instances are running, make sure that the MCP server is running in the instance you are looking at.

A Request Takes a Long Time or Fails with a Timeout

QCAD cannot answer while a dialog is open in QCAD. Close any open dialogs in QCAD and ask again. For very large drawings, increase the *Reply timeout* in the extension settings.

16.8 Part Library and Blocks

Reusing existing components saves time and keeps drawings consistent. This section shows how to make the most of the QCAD part library, including how to find ready-made CAD components and how to turn your own collections of DXF and DWG files into a searchable part library that you can insert into any drawing.

16.8.1 Working with the Part Library

Finding CAD Components

Any collection of DXF or DWG (and to a certain degree SVG) files can be used with QCAD as a part library.

Various QCAD part library items are available from our site at:

https://qcad.org/en/download/

However, there are countless online resources that offer free and paid collections of DXF/DWG files for use in any CAD system. Depending on your specific needs, you can use a search engine to look for existing DXF/DWG files with CAD symbols or blocks.

Here are just a few such sites which have collections of DXF/DWG files:

- **Architecture**
  - https://freecads.com
  - https://www.cadwiesel.de
  - https://www.arcat.com/details/cad_details.shtml
  - https://www.cad-blocks.net
- **Architecture, Electrical, Mapping, Mechanical, Piping, P&ID, Transportation**
  - https://www.cadforum.cz/catalog_en
- **Electronic Circuits**
  - https://github.com/leandromattioli/dxfsymbols
- **Architecture, Mechanical**
  - https://www.tracepartsonline.net
- **Various / Art / Clipart**
  - https://www.dxf-downloads.de
  - https://dxf-world.de
  - https://freedxf.com
  - https://www.cadblocksfree.com
  - https://thenounproject.com

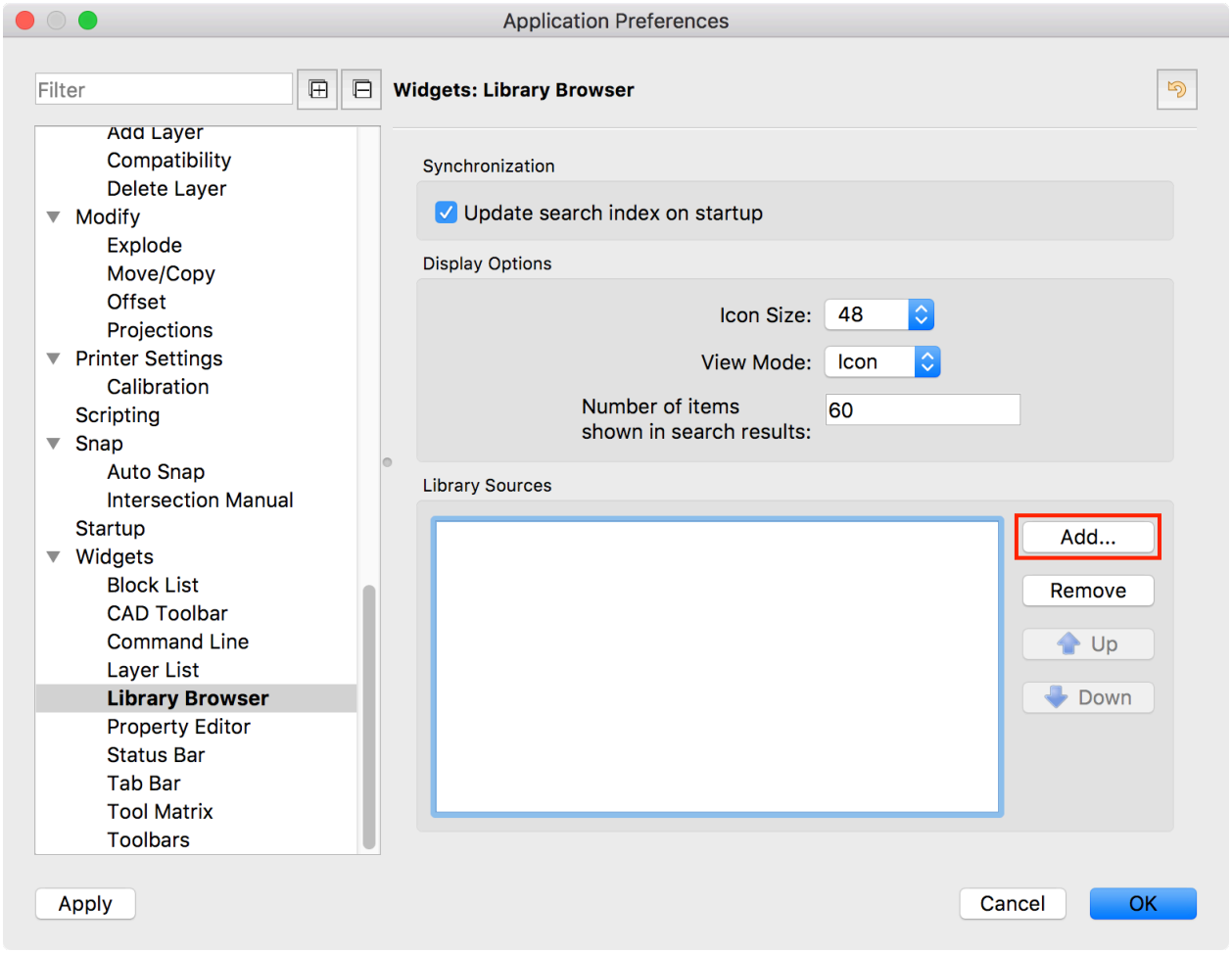
Countless other resources can be found using an Internet browser. Search for "DXF" and a keyword you are interested in, for example "DXF electrical symbols" or more specific "DXF electrical outlet", etc.

Adding Files to the Library Browser

Once you have downloaded DXF/DWG files and extracted them if they came as a ZIP file, you need to tell QCAD where to find these files.

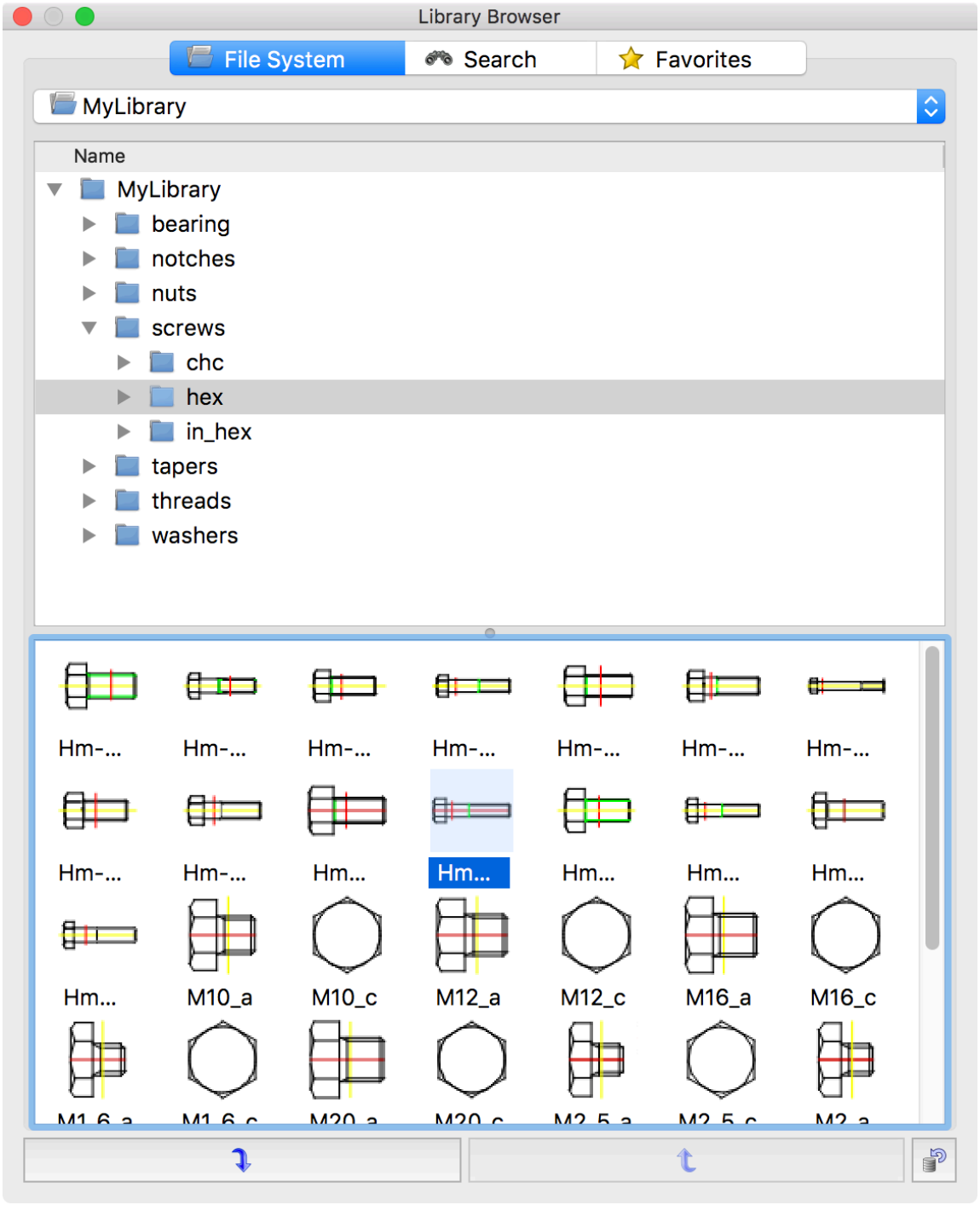
Let's assume you have downloaded various DXF files into `C:\Users\User\MyLibrary` (or `/Users/User/MyLibrary`) and sub folders thereof.

To tell QCAD where to look for this library of CAD files, add this path under *Edit* > *Application Preferences* > *Widgets* > *Library Browser* > *Library Sources* > *Add*:



After restarting QCAD, the library browser will show a new library (e.g. "MyLibrary") with all the subfolders and items in it.

If the library browser is not already shown, you can display it using menu *View* > *Library Browser*:



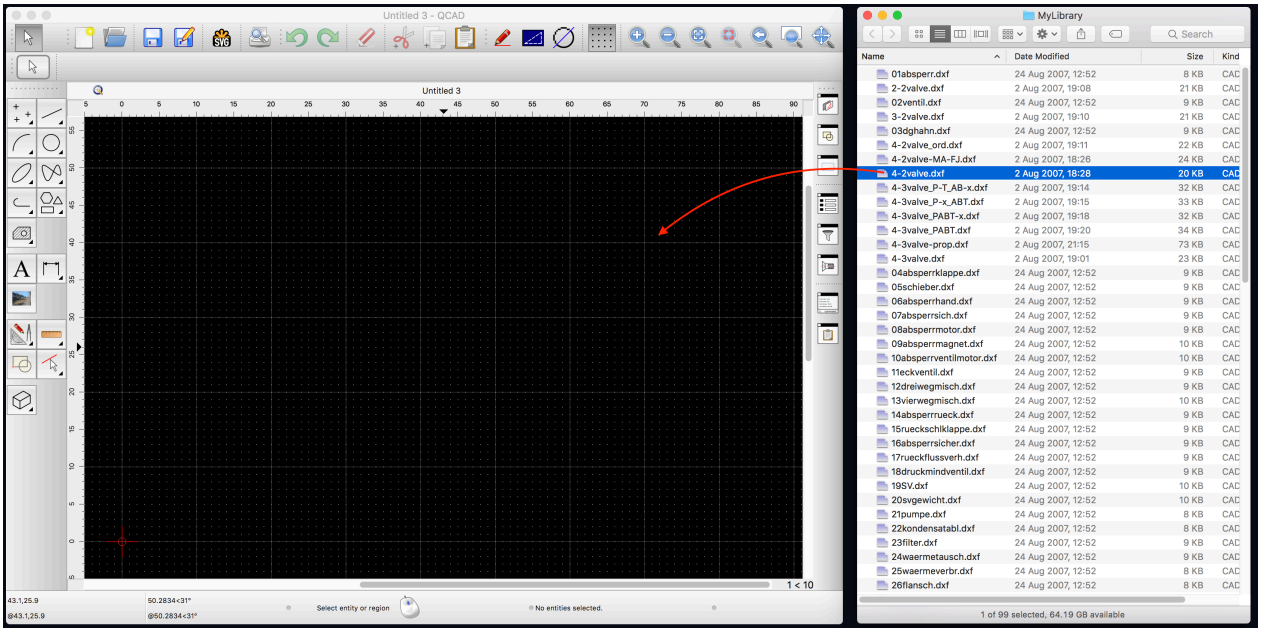
If you cannot see the new folder or items, please click the button at the bottom right to re-synchronize the database from the files on disk. Starting with QCAD ≥ 3.28, this button is no longer available and synchronization is no longer necessary.

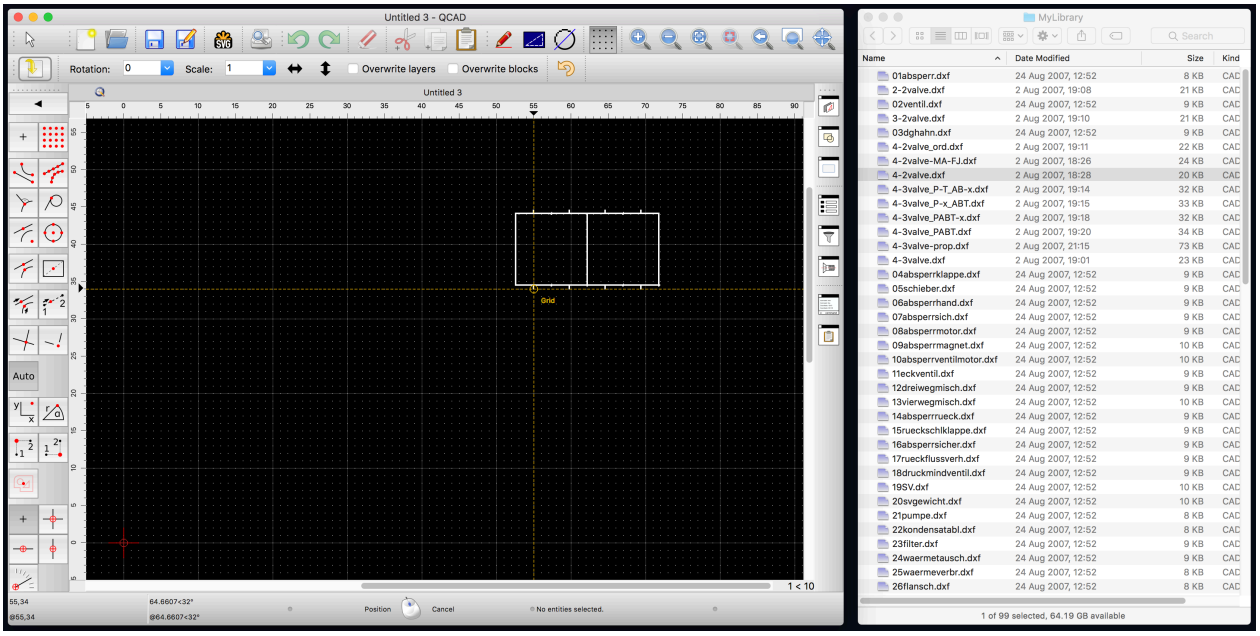
Supported File Formats

The QCAD Library Browser supports all files supported for reading by QCAD. Starting with QCAD 3.28, this includes PDF files and bitmap files.

Working with a File Explorer / Finder

As an alternative to working with the QCAD Library Browser, you might find it more convenient to work with your favorite file browser instead. If you keep QCAD and any file browser open side-by-side, you can simply drag files over into your drawing area to insert them as blocks.





Extracting ZIP Archives

If you need help extracting downloaded ZIP files, please refer to the documentation for extracting ZIP files under Windows, macOS or Linux.

16.9 Preferences

QCAD can be adapted to your personal preferences and working environment in many ways. This section walks through common configuration tasks, such as adjusting the background and entity

colors, changing the language of the user interface, and resetting QCAD to its default settings by removing the configuration file when something goes wrong.

16.9.1 Adjusting Colors

Changing the Background Color

The background color can be adjusted under: *Edit* > *Application Preferences* > *Graphics View* > *Colors* > *Graphics View* > *Background*.

Changing the Color of an Entity

The color of entities is usually controlled by the layer it is placed on. Please refer to the tutorial *Working with Layers* to learn about layers and attributes.

To change the color of an entity (or multiple entities) you can select the entities and use the property editor (*View* > *Property Editor*) to change the color property to any desired fixed color.

Drawing in the Background Color

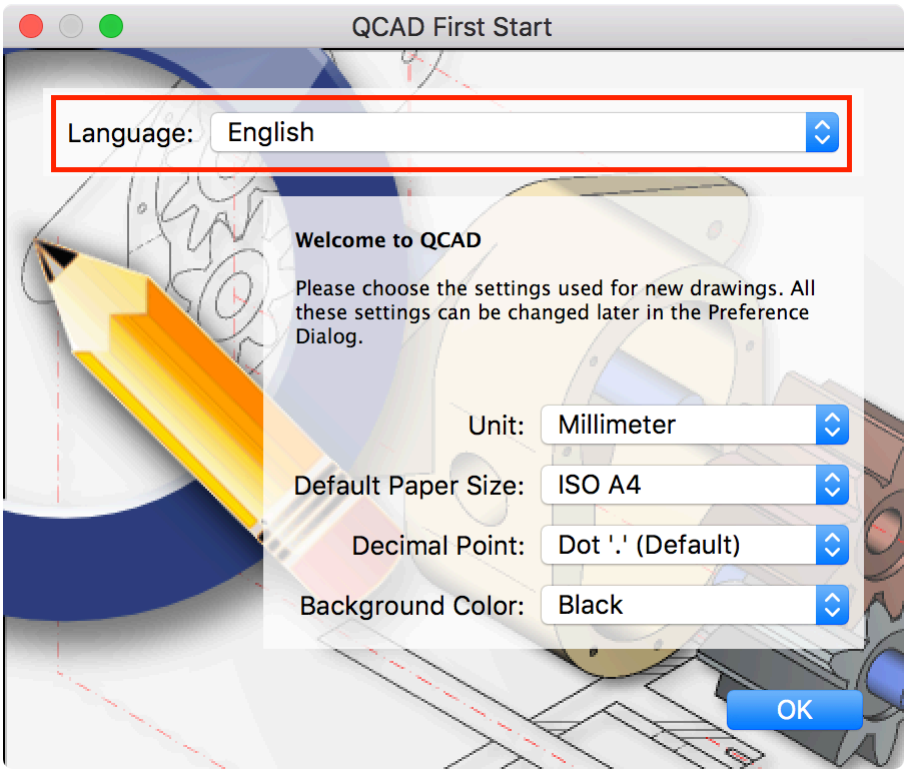
By default, QCAD automatically adjusts the color of white / black entities to never match the background. On a white background, white entities are shown in black and on a black background, black entities are shown in white. This is to make sure that drawings are not "invisible".

This behavior can be adjusted under: *Edit* > *Application Preferences* > *Graphics Views* > *Appearance* > *Prevent white on white / black on black display*.

16.9.2 User Interface Language

Setting the Language on First Start

When QCAD is started for the first time, a dialog is shown where the language of the user interface can be set:



Changing the Language

The user interface language of QCAD can also be changed anytime later under:

Edit > Application Preferences > Language Settings > Language > Language of User Interface

French:

Édition > Préférences générales > Configuration de langue > Langue > Langue de l'interface utilisateur

German:

Bearbeiten > Applikationseinstellungen > Spracheinstellungen > Sprache > Sprache der Benutzeroberfläche

Spanish:

Editar > Preferencias de la aplicación > Configuración de idioma > Idioma > Idioma de la interfaz de usuario

Italian:

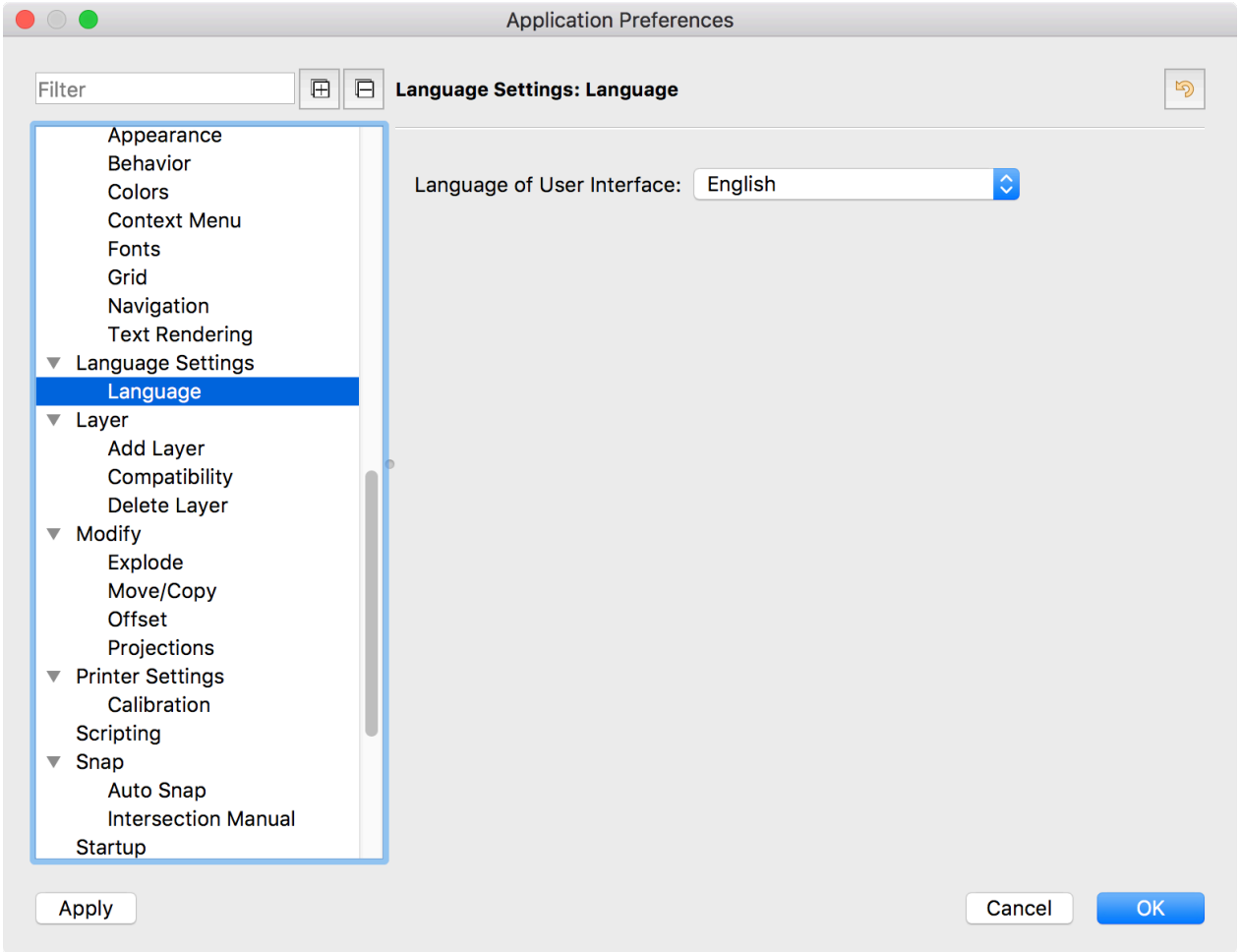
Modifica > Preferenze d'applicazione > Impostazioni della lingua > Lingua > Lingua dell'interfaccia utente

Portuguese:

Edição > Preferências do Aplicativo > Configurações de Linguagem > Linguagem > Idioma da Interface do Usuário

Dutch:

Bewerken > Programma Voorkeursinstellingen > Taal instellingen > Taal > Taal van de gebruikersinterface



Available Languages

The QCAD user interface is available in the following languages:

Tier 1 (up to date and proofread):

- Dutch
- English
- German
- German (Swiss)

Tier 2 (up to date, machine translations):

- Bulgarian
- Chinese (Simplified)
- Chinese (Traditional)
- Croatian
- Czech
- Danish
- Estonian
- Finnish
- French
- Greek
- Hungarian
- Indonesian
- Italian
- Japanese
- Korean
- Latvian
- Lithuanian
- Norwegian (Bokmål)
- Polish
- Portuguese
- Romanian
- Russian
- Slovak
- Slovenian
- Spanish
- Swedish
- Thai
- Turkish
- Ukrainian
- Vietnamese

Setting the Language as Command Line Argument (Advanced)

It's also possible to override the language for an individual session on the command line using the `-locale` command line switch:

```
qcad -locale de
```

This can be useful to create different desktop icons for different languages, one for QCAD in English, one for QCAD in German, etc.

16.9.3 Resetting the QCAD Configuration

Introduction

If your configuration is somehow damaged and you cannot get it back to normal, you might want to reset QCAD to its default settings as they were when QCAD was first installed.

This will affect:

- arrangement and visibility of toolbars and dockable widgets
- arrangement and visibility of tools
- user interface language
- default settings for new drawings
- etc.

Step 1: Quitting QCAD

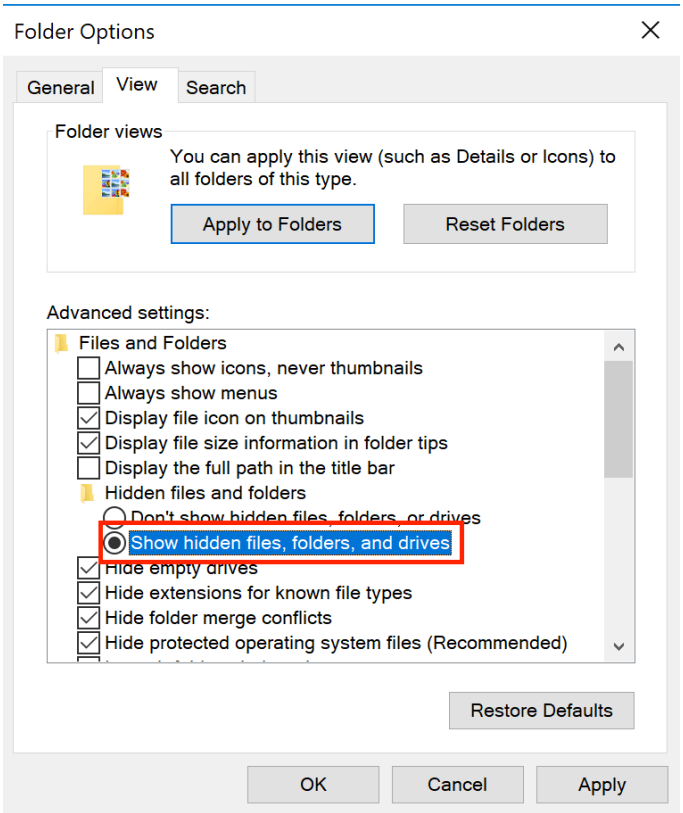
Make sure that QCAD is closed before resetting the configuration.

Step 2: Deleting the QCAD Configuration File

Windows

Make sure that system files and system folders are displayed in Windows Explorer:

- Open Windows Explorer (e.g. *Start* > *All Programs* > *Accessories* > *Windows Explorer*)
- Windows XP: *Tools* > *Folder Options* > *View* > *Show hidden files and folders*
- Windows Vista, 7, 8, 10: *View* > *Options* > *Change folder and search options* > *View* > *Show hidden files, folders, and drives*
- Windows 11: Click the three dots in a Windows Explorer window at the top, then *Options* > *View* > *Show hidden files, folders, and drives*



The file `QCAD3.ini` is located in (where "username" is your login name):

- Windows XP: `C:\Documents and Settings\username\Application Data\QCAD\QCAD3.ini`
- Windows Vista, Windows 7, 8, 10: `C:\Users\username\AppData\Roaming\QCAD\QCAD3.ini`

Note that Windows Explorer might not show the file extension (.ini) but only "QCAD3".

- Delete or rename the file `QCAD3.ini`
- To reset the library browser, delete also the file `library_3_0_0_9.db`
- Start QCAD

Linux

- Delete the file `~/.config/QCAD/QCAD3.conf`
- To reset the library browser, delete also the file `~/.config/QCAD/library_3_0_0_9.db`
- Start QCAD

macOS

- Open Finder
- Choose menu *Go* > *Go to Folder...*
- enter `~/.config/QCAD`
- Click *OK*
- Delete file `QCAD3.ini`
- To reset the library browser, delete also the file `library_3_0_0_9.db`
- Start QCAD

16.10 Printing

Turning a drawing into a printed sheet or a PDF involves setting up scale, paper size and layout so the result matches your intent. This section explains how to prepare a drawing for output, how to use the print preview to print or export to PDF, and how to build more complex, multi-view sheets using layout blocks and viewports.

16.10.1 Printing

Step 1: Before Printing

Drawing 1:1

Drawings are almost always drawn in real size (1:1) with a CAD system. If this is not the case for your drawing, you might want to correct this before proceeding. You can scale your drawing by any given factor using the *Modify* > *Scale* tool if necessary.

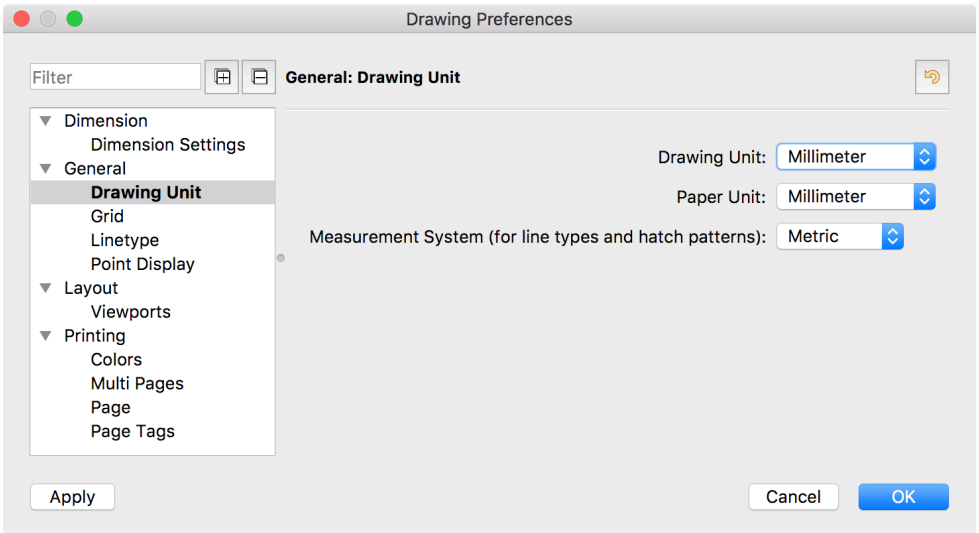
Drawing Borders, Text Labels, Dimensions

If your drawing has a drawing border or header, scale the border or header instead of scaling your drawing. For example, if your drawing is to be printed with a scale of 1:5 (factor 0.2 or five times smaller than real size), scale your drawing border or drawing header with a factor of 5, i.e. insert it five times larger than you want it to print. The same scale should be applied to text labels and dimension properties (dimension arrows, labels, gaps, etc.).

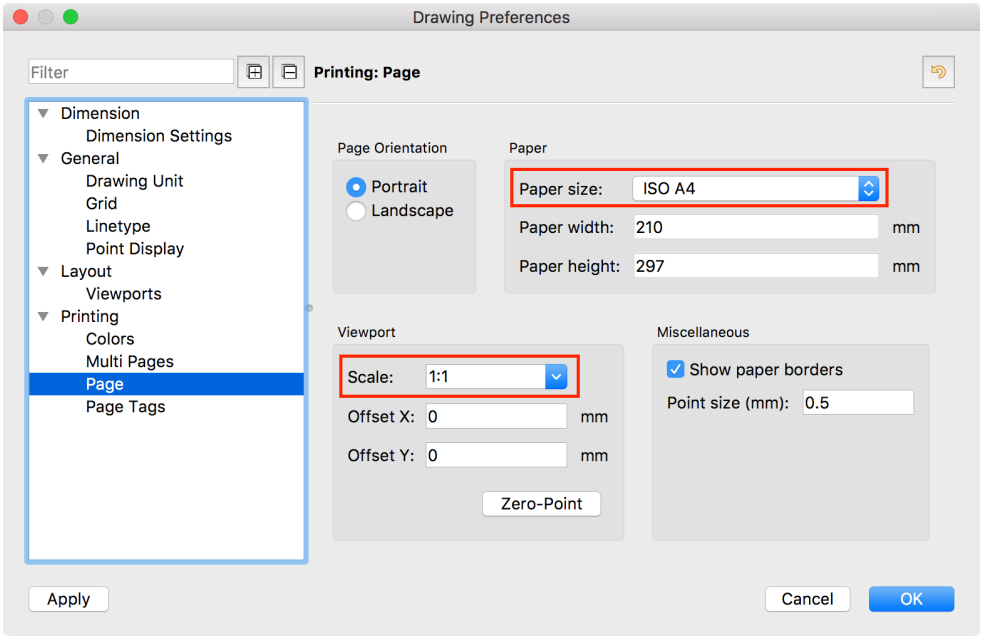
Alternatively, you can use layout blocks and viewports to create a layout of your drawing border and drawing as described in *Tutorial: Layout Blocks and Viewports*.

Drawing Preferences

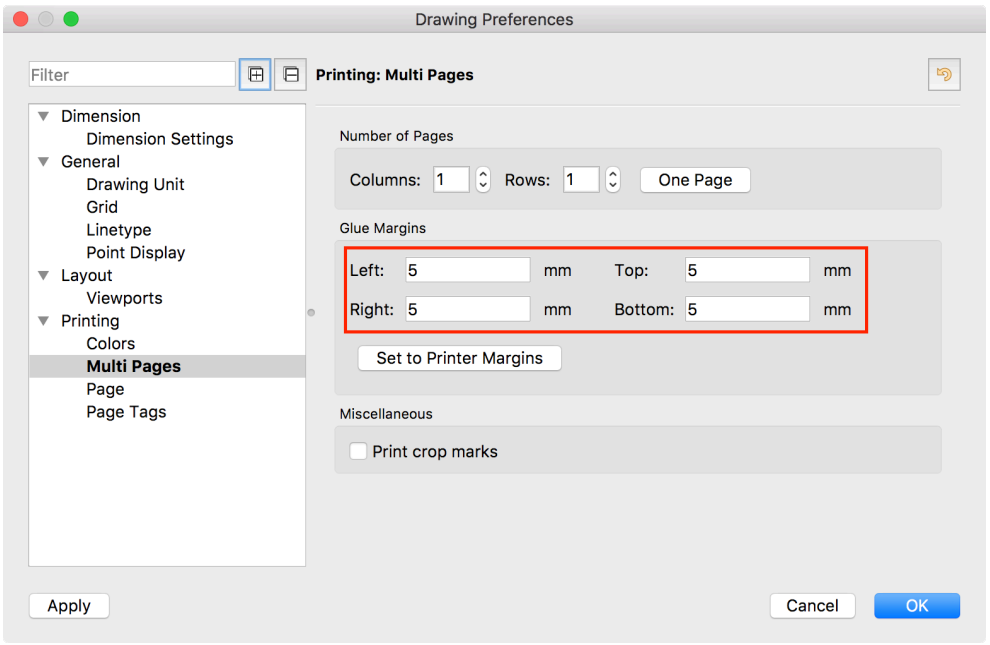
- Open the DXF/DWG file you want to print.
- *Edit* > *Drawing Preferences* (NOT Application Preferences)
- Check and adjust the drawing unit and paper unit under *General* > *Drawing Unit*:



- Check and adjust *Printing* > *Page* > *Paper Size* and Scale (if known):



- Check and adjust *Printing* > *Multi Pages* > *Glue Margins*. These margins are used as overlap when printing to multiple pages but also influence the tools to automatically fit or center the drawing on paper:



- In the same dialog page, you can also adjust the number of columns and rows to use when printing to multiple pages.

Step 2: Print Preview

Choose *File* > *Print Preview* to switch to the print preview mode.

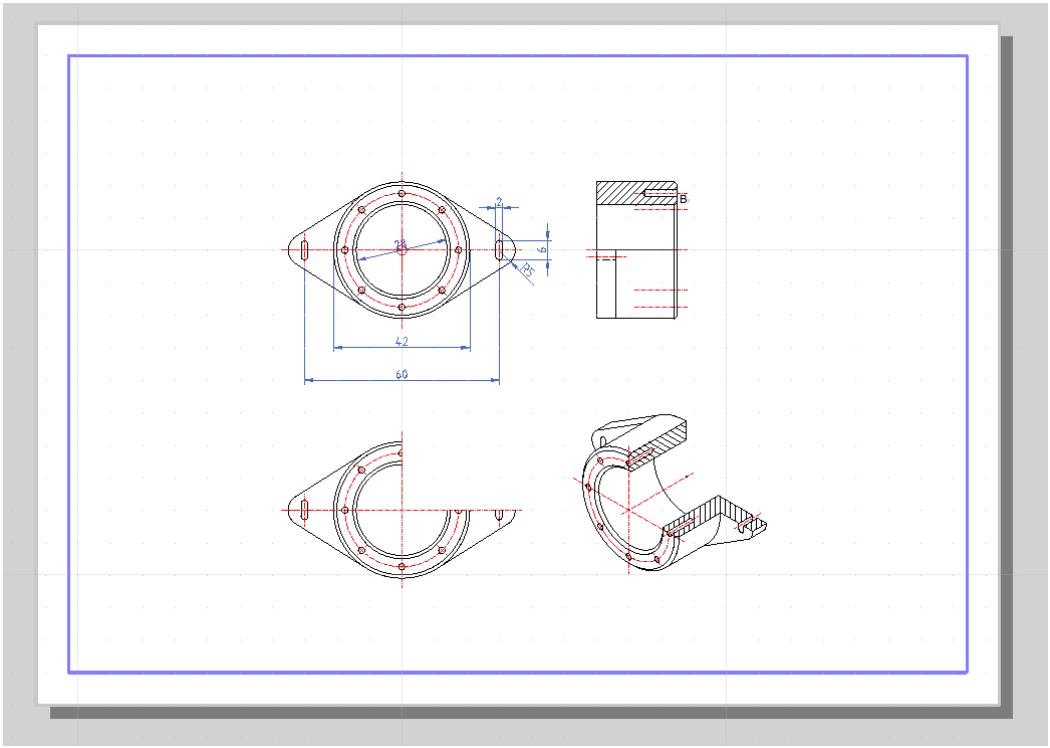
Adjusting the Scale



- If the desired scale for printing is known (e.g. 1:2, 1:10, 2:1, 200:1, etc.), adjust the scale to the accurate, desired scale (2:1 in this example) [1]. If the desired scale of your drawing is *not* known or irrelevant, click the button to adjust the drawing scale to the chosen paper [3].
- Optionally, adjust the paper position using the tool Move Paper Position [2] or use the button to center the drawing on the chosen paper(s) [4].
- To print in black / white or gray scale, click the appropriate icon in the options toolbar:

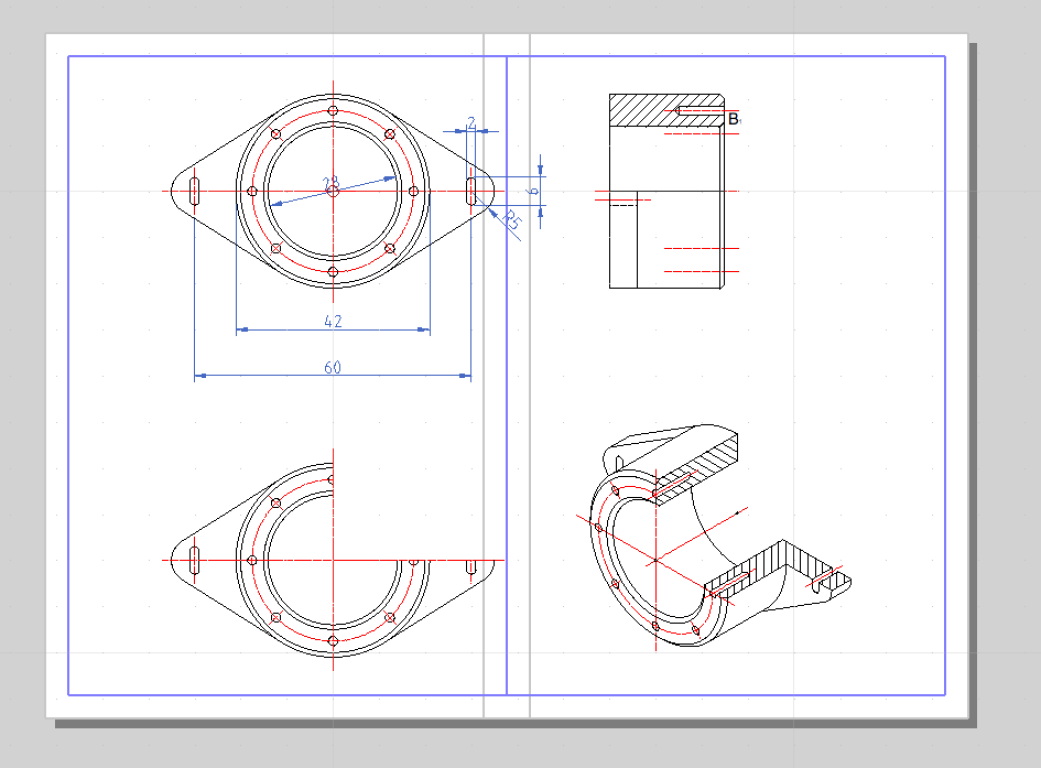


- Your drawing should be shown on a white paper as shown here:



If your drawing is shown on a gray background, please check your paper and scaling settings again. If the purple rectangle crosses your drawing, please check the glue margins in the drawing preferences.

- If you have chosen to print to multiple pages, the pages are shown beside / below each other:



Step 3: Printing / PDF Export

Printing on Paper

To print the drawing, choose *File* > *Print*.

PDF Export

To export the drawing with the chosen layout to PDF, choose *File* > *PDF Export*.

Troubleshooting

Thin or Invisible Lines

If your print out is very light or fine (thin lines), you might want to adjust the widths of your entities. Line widths are usually controlled through layer properties. Select the layer which prints too fine, then choose *Layer* > *Edit Layer* and change the width to something thicker. Avoid line weight "Default" as this prints at the configured default line width (*Edit* > *Application Preferences* > *Graphics View* > *Appearance* > *Linetypes and Lineweights* > *Default Lineweight*) and might not be visible.

If the above does not solve the problem, please make sure that the hairline mode is not enabled for your drawing:

- Load your file
- Choose menu *File* > *Print Preview*
- Make sure that the button to toggle the hairline mode is not checked. In this example it is checked and needs to be unchecked (click the button to uncheck it):



Imprecise Scaling

Printers are not 100% accurate and paper expands / contracts depending on temperature and humidity. You can calibrate your printer under *Edit* > *Application Preferences* > *Printer Settings* > *Calibration*. If for example your print is 2% too small in Y direction, set *Factor Y* to 1.02.

16.10.2 Working with Layout Blocks and Viewports

Introduction

This tutorial introduces an advanced concept in QCAD to print drawings by creating a layout. A layout is an arrangement of parts of the drawing, accompanied with a drawing border and possibly other elements. A layout shows how the drawing will be printed.

Preparation

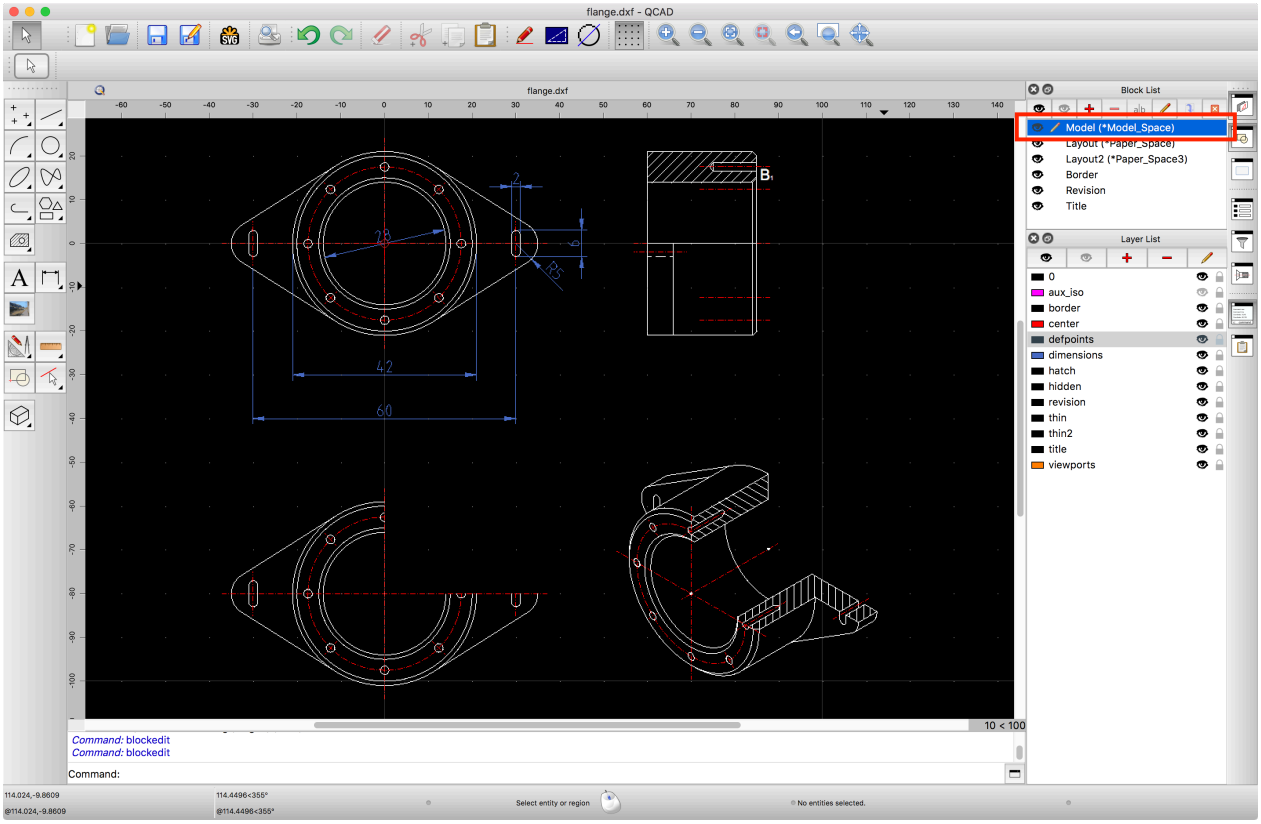
When working with layouts, make sure that the block list is shown. The block list shows all blocks of a drawing, including the model space block and layout blocks. Use *View* > *Block List* to show the block list.

Model Space and Layout Blocks

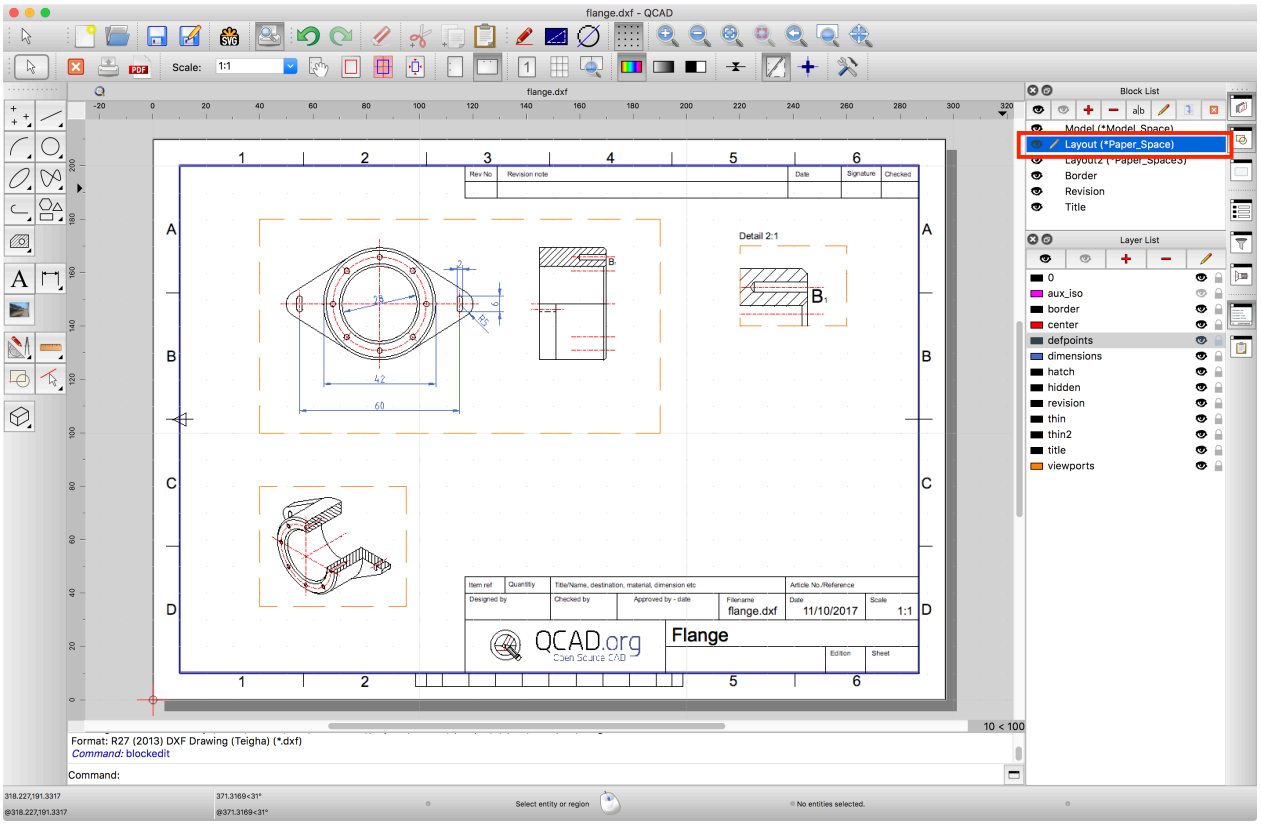
When starting a new drawing in QCAD, the model space (also "model" or "drawing model") is shown. This is a special block which holds the main drawing in real size (1:1). In other words, this is where you create the geometry of your drawing.

A layout (also "layout block" or "paper space") is another special type of block that is usually used for printing a drawing or parts of a drawing. A layout shows one or multiple parts of the drawing, often accompanied by a drawing border and drawing header. Each part of the drawing can be shown at a selected scale and angle.

Consider the example "flange.dxf" which comes with every installation of QCAD. When you open the drawing in QCAD, you can see the model space where all the geometry is created in real size. The geometry is typically arranged in any way that happened to be convenient for constructing it. It is usually not arranged in the way the drawing should be printed.



If you switch to the first layout (listed as "Layout (*Paper_Space)" in the block list), you can see that the same drawing has been re-arranged, a border has been added and a detail of the drawing is shown at a different scale:



It is important to understand at this point, that this is still the same drawing. The objects shown inside the orange, dashed borders have not been copied or scaled and when the drawing model is changed at any point, the layout immediately reflects these changes. A layout block can also contain any kind of own drawing elements, but usually contains a drawing border and a drawing header or legend. In addition, a layout usually contains viewports. These are the orange, dashed rectangles shown above.

Viewports

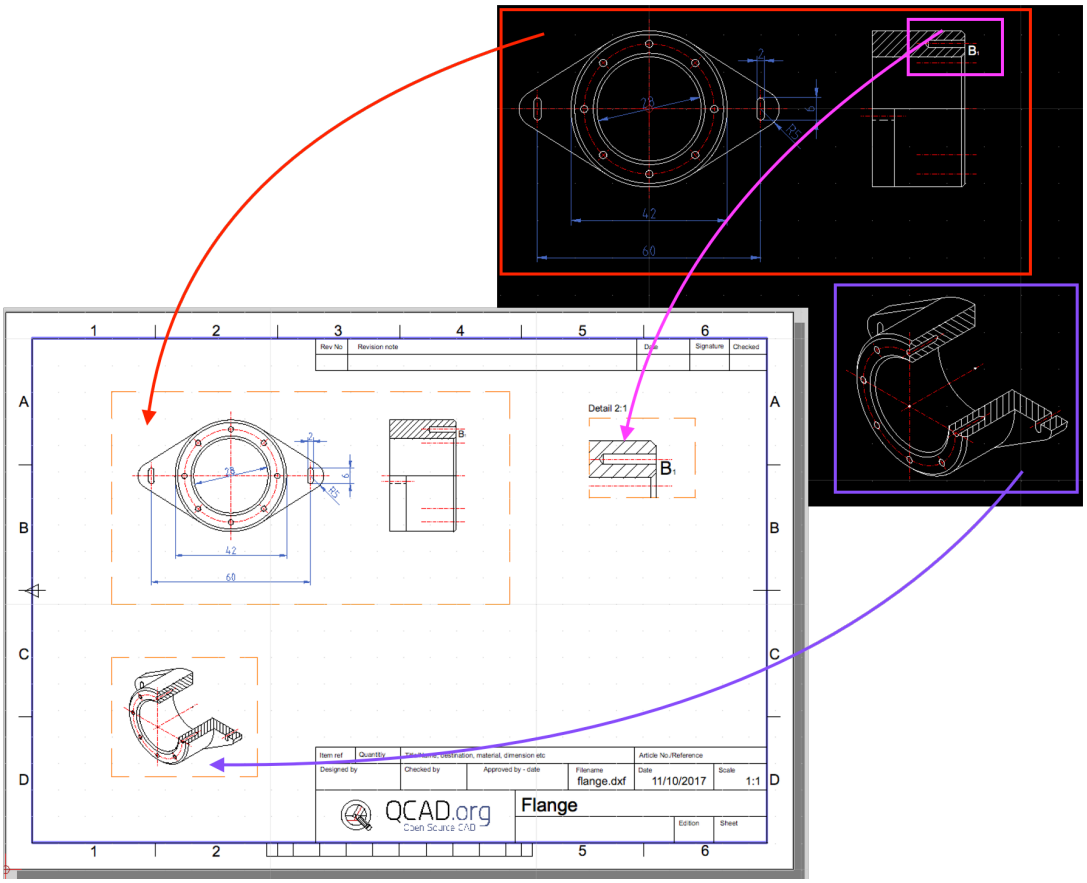
The rectangular areas with a dashed orange border in the layout above are called viewports. A viewport can be described as a window to the drawing model. It shows a part of the drawing model in a layout. You can also think of a viewport as a special kind of a block reference that always references the model space block and is clipped at a given size.

Viewports and Layers

Just like any other entity, every viewport is placed on a layer. The layer controls the appearance of the viewport rectangle (in our example orange and dashed). The layer also controls the visibility of the viewport border. The layer with the viewports on it can be switched off to hide the rectangular viewport borders. The layer with the viewports is also usually made unplottable which means that the rectangular viewport borders are not printed. Note that the layer does not control the visibility of the viewport contents. The contents of a viewport can be hidden using the special viewport property "On".

Viewports and Duplication

The detail B 1 shown in our example is shown twice in the layout: once in the overview at the left and again, at a larger scale at the right. However, there is no duplication of data. The same data is shown twice, through two different viewports at two different scales.



Creating Layouts

Every new drawing automatically has a layout block. New layout blocks can be added as required using *Block* > *Add Layout Block*.

Creating Viewports

Viewports can be created using *Draw* > *Viewport* > *Add Viewport*.

Adjusting Viewports

Just like other entities, viewports can be selected by clicking them. A selected viewport shows reference points at the corners which can be used to adjust the size of the viewport. The reference point at the center of the viewport can be used to move the viewport around on the layout.

There is one other reference point above the center. This reference point can be used to move the contents of the viewport within the viewport. You can drag and drop this reference point for panning within the viewport.

16.11 Properties

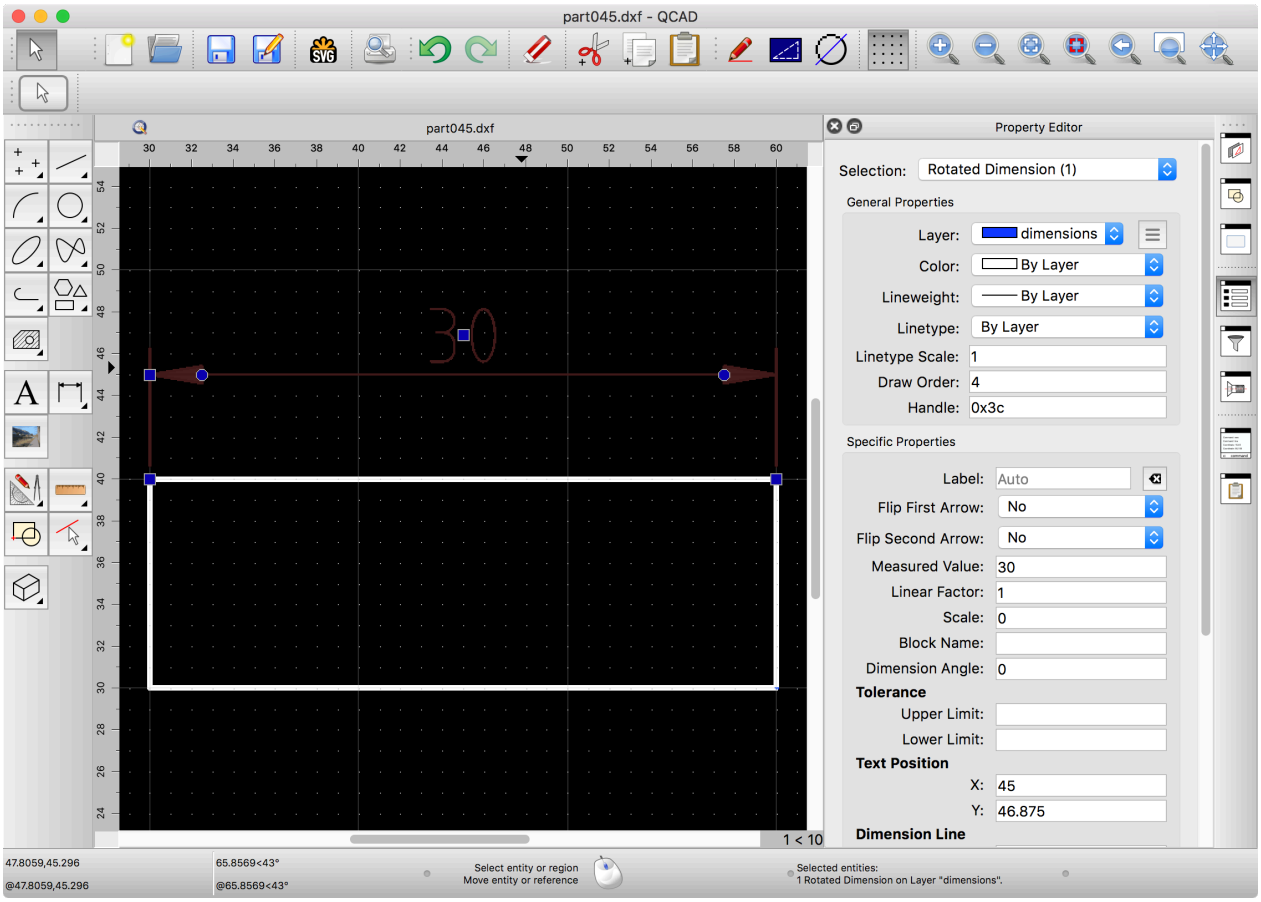
Every entity in a drawing carries a set of properties that define how it looks and where it sits. Some properties are common to all entities, such as the layer an entity belongs to, its color, linetype and lineweight. Others are specific to the entity type, for example the start and end point of a line, the center and radius of a circle, or the text and height of a text entity. This section explains how to view and edit properties, either individually or for many entities at once.

16.11.1 Property Editor

This tutorial introduces the basics of the QCAD property editor.

Property Editor Basics

The property editor is shown at the right side of the QCAD application window after the first launch of QCAD. If you cannot see a property editor, you can show it using the menu *View* > *Property Editor*. Just like most other widgets in QCAD, you can dock it in any desired location, usually somewhere at the right as shown here:



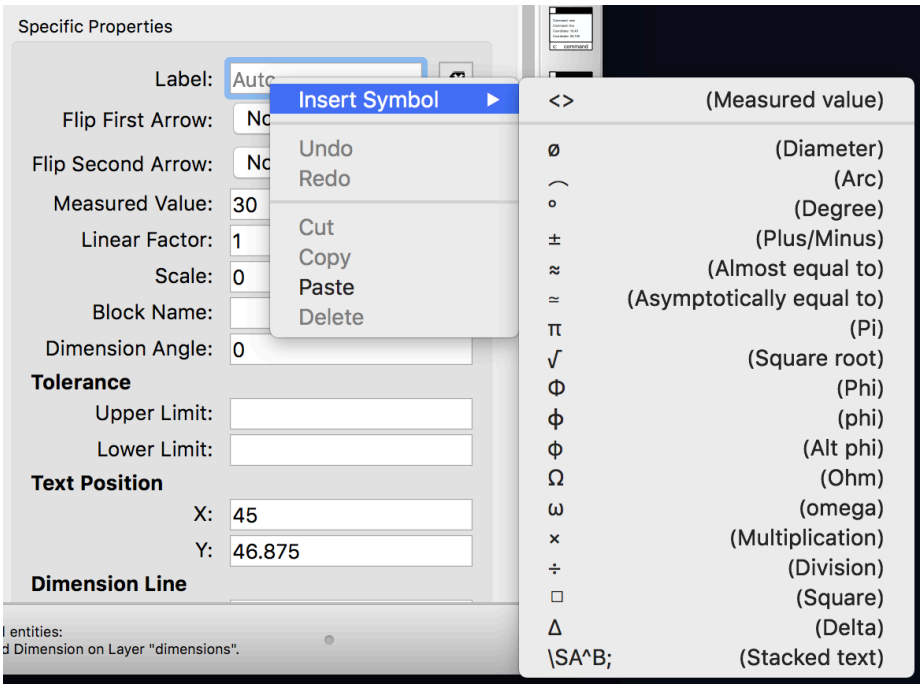
The property editor displays the various properties of the entity or entities that are currently selected and allows you to change them. In the example above, a dimension entity is selected and the property editor shows all properties of this entity.

In this case, there is for example a property called *Label* which shows the value *Auto*. This means that the label of the dimension is not explicitly set and is therefore calculated automatically. We can use the property to enter a fixed label to be used instead.

Dimension Labels and Symbols

Dimension labels often contain special characters that are hard to access on a keyboard (⌀, °, ±, ...).

A right-click into the dimension label text field in the property editor shows a context menu which can be used to quickly insert commonly used symbols and formatting:



Of course, you can also type these codes or symbols directly into the text field. The context menu is merely provided for convenience. The most important code is `<>` (less than, greater than) to use the automatically measured value in combination with a symbol or text. For example `R<>` creates a dimension label R7.5 if the dimensioned radius is 7.5 drawing units.

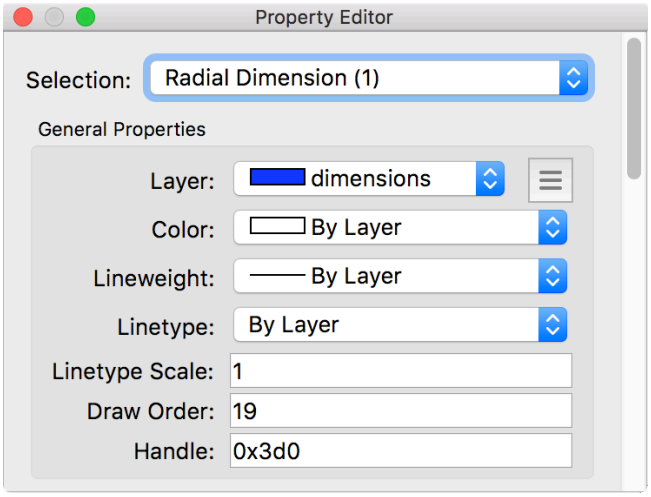
Changing Multiple Entities at Once

You can also change multiple selected entities at once using the property editor. To add an R in front of all radial dimension labels, simply select all radial dimensions and type `R<>` into the value of the label property.

Filtering by Entity Type

Sometimes, selecting all radial dimensions might be a challenge, especially in large and complex drawings.

To assist with that, the property editor offers a filter at the top where you can choose to only display the properties of a certain entity type. To add an R prefix to all radial dimension entities using this feature, we can simply select the entire drawing (*Select* > *Select All*) and then filter the property editor by radial dimensions by choosing *Radial Dimension* from the *Selection* box at the top of the property editor:



This filter also comes in handy when reorganizing drawings. To put for example all texts on a separate layer called text, you could select the whole drawing, choose the filter *Text* in the property editor and then change the layer property. You can even move all selected texts to a new layer by clicking the button at the right of the layer property and choosing *Move Selection to New Layer*.

16.12 QCAD/CAM

QCAD/CAM extends QCAD with computer-aided manufacturing features that convert your drawings into machine instructions. This section introduces the CAM workflow from start to finish: preparing geometry, creating toolpaths and generating G-code, arranging parts efficiently on sheets with nesting to minimize waste, and writing or customizing postprocessor configuration files so the output matches your specific machine.

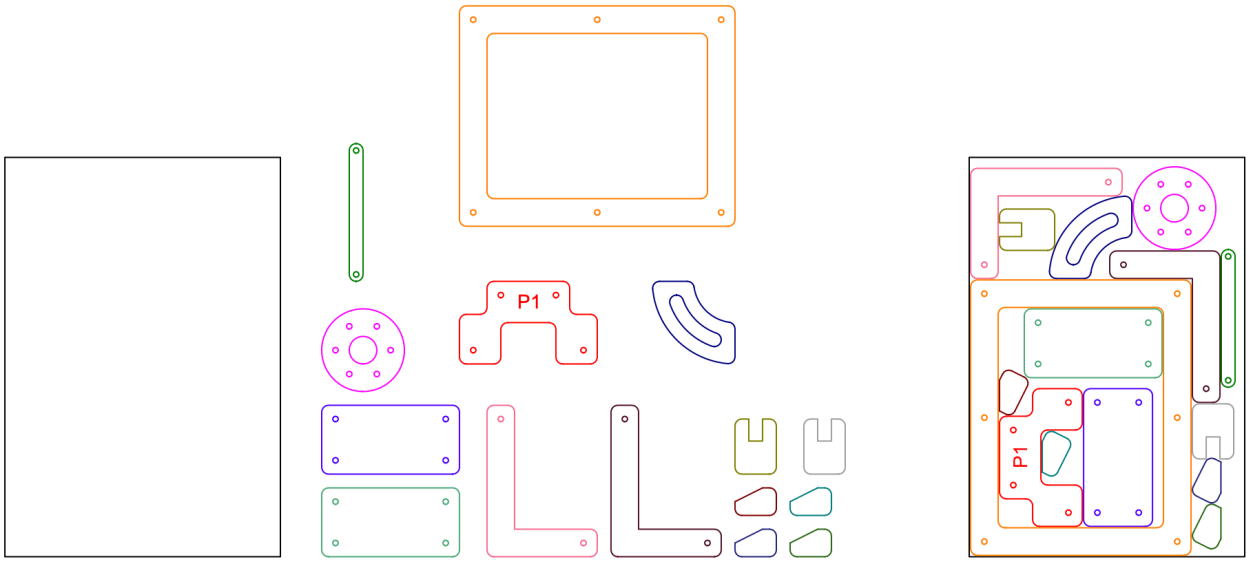
16.12.1 Nesting

Introduction

Nesting is the process of arranging parts on a sheet in such a way that they occupy a minimum of space. The main objective of nesting is usually to minimize waste when cutting sheet material.

QCAD/CAM comes with an integrated nesting tool that allows you to fully automatically arrange parts onto a single or multiple sheets.

In the example below, the sheet is shown in black at the left and the parts in various colors. A possible solution as proposed by QCAD/CAM is shown at the right.



Features

- Placing smaller parts inside holes of larger parts (part-in-part placement)
- Nesting based on the actual shape of a part (true shape nesting)
- Automatic rotation of part at configurable angles
- Automatic alignment of parts to X/Y axes
- Nesting polylines, loose entities, splines, ellipses and texts
- Mark parts to be nested multiple times
- Multithreading

Preparation of Parts

A "part" is a closed shape, optionally with holes. Larger holes in a part may be used for the placement of other, smaller parts as shown above for the hole in the large rectangular orange part.

A part does not necessarily have to be surrounded by a polyline but can also consist of loose, connected entities (lines, arcs, ellipse arcs, splines, circles). A text entity may also be used as a part. If a part contains a text entity, that text entity stays with the part and is not regarded as a hole. This is the case for the red part above. Texts may be used to label or track parts.

To nest the same part multiple times, select the part and choose *CAM* > *Set Nesting Quantity*. A dialog allows you to enter the desired quantity for that part.

Preparation of Sheets

A "sheet" is also expected to be a closed shape, optionally with islands (holes). Sheets can consist of polylines, lines, arcs, ellipse arcs or splines. If a sheet contains an island, that island will not be used to place parts during nesting.

Nesting

Select the parts to nest using the QCAD/CAM selection tools. Start the nesting tool and choose the sheets to nest into by clicking any entity of each closed sheet to use.

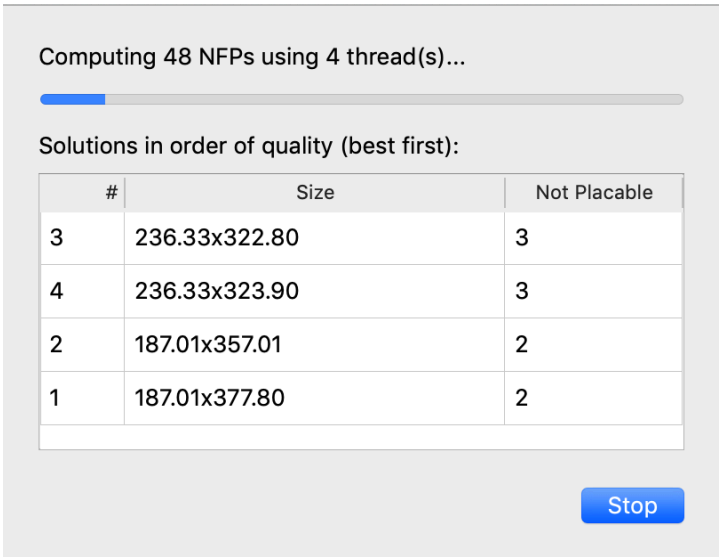
For detailed instructions and an explanation of the various nesting parameters, please refer to the reference manual.

Set rotations to "1" to insert all parts without rotating them. A value of 4 usually yields good results for parts with many right angles and parallel edges.

Right-click when you're done choosing sheets.

QCAD/CAM starts to produce possible solutions to arrange as many parts as possible in the chosen sheets.

As soon as solutions become available, they are listed in the table of the nesting dialog:



The nesting algorithm continuously looks for new, better solutions and lists them in descending order (the best solutions are listed at the top, the worst at the bottom). If a solution is only a partial solution (meaning some parts could not be placed in the chosen sheets), the number of parts that were not placed is shown in the last column.

Previewing Solutions

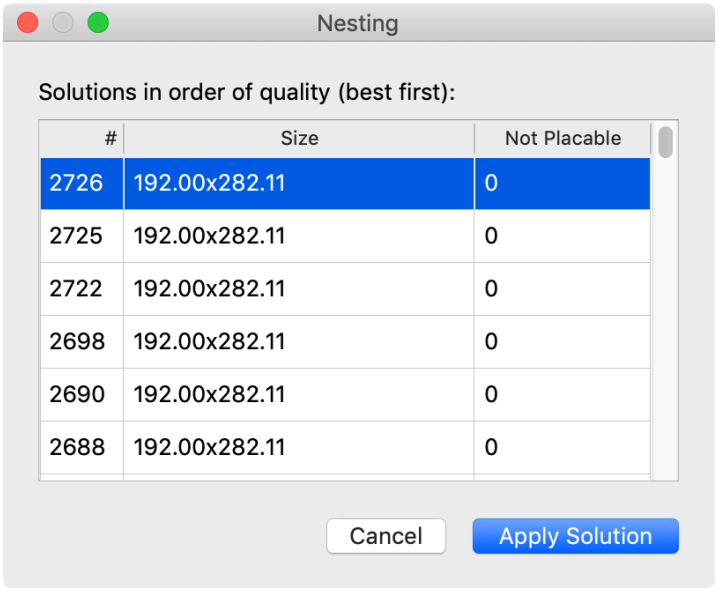
The suggested solutions can be previewed while QCAD/CAM is looking for other solutions by clicking on one of the rows in the table.

Stopping the Nesting Process

If a good solution has been found, click the Stop button to stop looking for better solutions.

Choosing and Applying a Solution

The nesting dialog is now shown again with all the solutions found until the nesting process was stopped. The suggested solutions can now be previewed again until the preferred solution has been identified. To apply the selected solution, click *Apply Solution*.



Examples

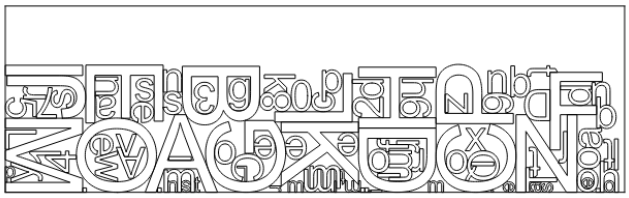
Alignment Left - Bottom:



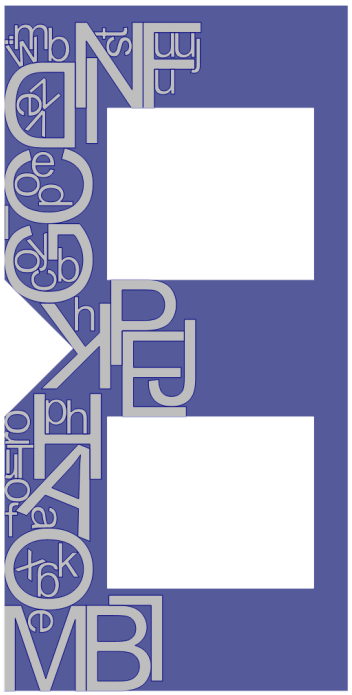
Alignment: Left - Top:



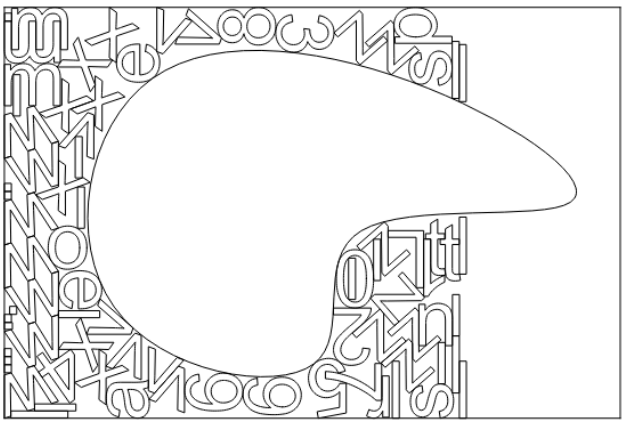
Alignment: Bottom, nesting inside holes of other parts:



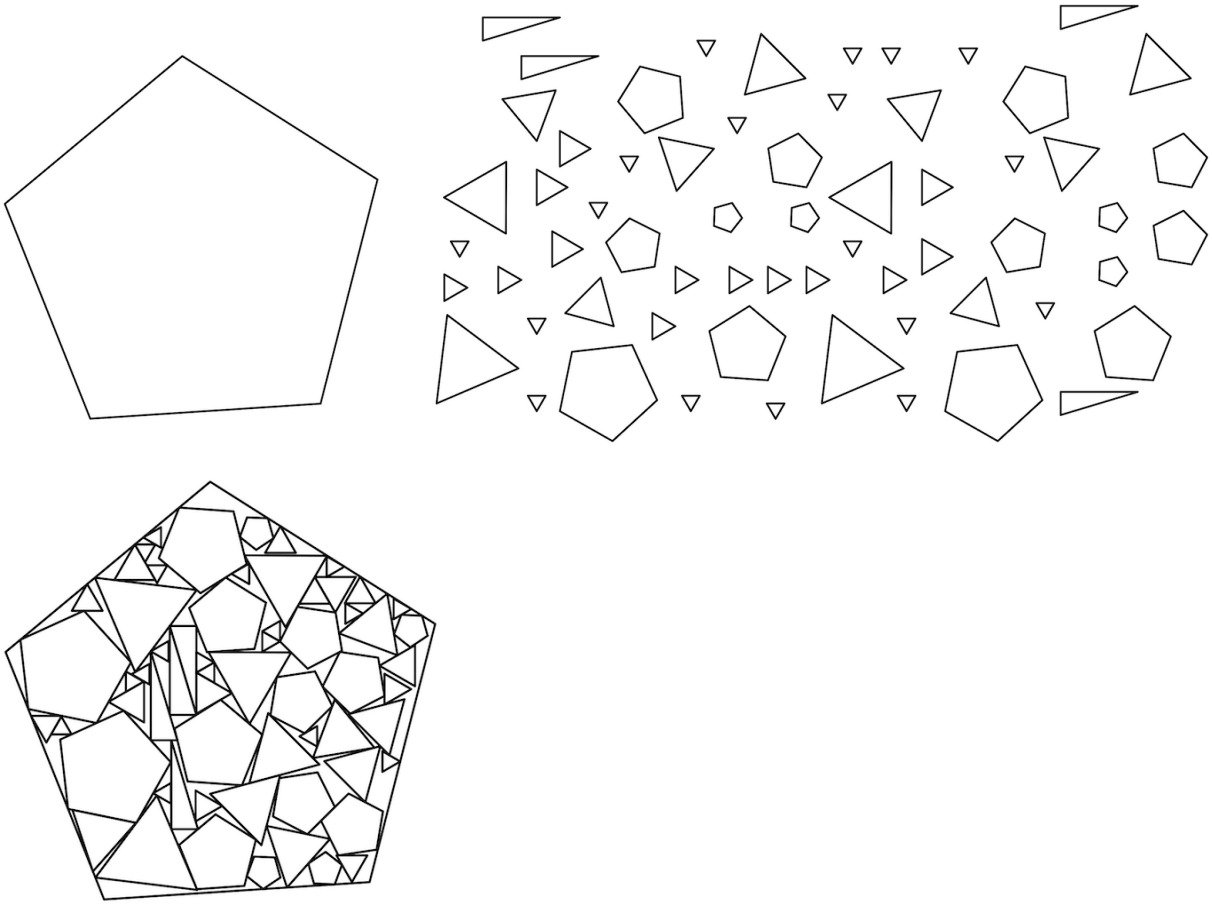
Concave sheet with islands:



Spline shaped island:



Polygons at various angles:



16.12.2 QCAD/CAM Postprocessor

QCAD/CAM can export CAD drawings to various formats used in CAM (computer-aided manufacturing). These formats can be dialects of G-Code or other, completely different formats. Each supported CAM format is defined in a configuration file (or "postprocessor") which is an ECMAScript (JavaScript) file that implements the format details. All available configurations are stored inside the directory `postprocessors`. New format implementations can be added to the same location.

Basic Structure

Each configuration file contains an ECMAScript class with the same name as the script file. If the script file is called `MyGCode.js`, the class defined inside the script must be called `MyGCode`.

The ECMAScript class must derive directly or indirectly from class `CamExporterV2`, which is defined by QCAD/CAM. `CamExporterV2` defines the basic structure of CAM exporter configuration classes. Various methods in `CamExporterV2` are called to export the file header, the various entities in the CAD drawing and the footer. `CamExporterV2` also takes care of the various features of QCAD/CAM, for example tool radius compensation, lead in / lead out, cutting inner contours before outer contours, path optimization, etc.

`CamExporterV2` does not output anything by itself, but would produce an empty file on its own. `CamExporterV2` offers a text stream (`QTextStream`) to write to the exported file. This text stream is exposed through the member variable `this.stream`.

If the desired output is a G-Code dialect (e.g. `G1 X10 Y20` for a linear movement), the configuration can be derived from the existing configuration for G-Code called `GCodeBase`, which is defined in file `GCodeBase.js`. This is the case for most machine controllers. One such example configuration for outputting G-Code in Millimeters is defined in file `GCodeMM.js` and available in the user interface as the configuration called G-Code (G41/G42) [mm].

Deriving a Configuration from GCodeBase

Create a file with a unique name of your choice. For this example, we call our new configuration MyGCode and store it in a file with the same name, i.e. `MyGCode.js` under directory postprocessors of your QCAD/CAM installation.

You can use your favourite plain text editor to create and write the file, for example Notepad under Windows, TextEdit under macOS or vim, Emacs, etc. under Linux systems.

`MyGCode.js` must define a class called MyGCode to be a valid configuration for QCAD/CAM. While there are strictly speaking no classes in ECMAScript, we can create a similar concept by attaching functions to the prototype object of a constructor function. How this works exactly is beyond the scope of this documentation. However, there is a straightforward template you can use for all configuration files.

For our example, the definition of the class MyGCode looks like this (note that lines starting with `//` are comments and are not required):

```
// Include GCodeBase.js which contains the class 'GCodeBase',
// used as base for our own configuration:
include("GCodeBase.js");

// Constructor of our class:
function MyGCode(documentInterface, newDocumentInterface) {
    // Call constructor of base class:
    GCodeBase.call(this, documentInterface, newDocumentInterface);
}

// Derive our class from class GCodeBase:
MyGCode.prototype = new GCodeBase();

// Display name shown in user interface:
MyGCode.displayName = "My G-Code";
```

Now, we have created a new, fully working configuration called "My G-Code" which is ready to be used in QCAD/CAM and produces basic G-Code.

Configuration Preferences

The constructor we defined above calls the constructor of our base class GCodeBase. In the constructor, we can set some parameters for our configuration such as the desired output unit or if we want to use G41/G42 for tool radius compensation or calculate the center of the tool path.

To configure the output unit (that's the unit expected by your machine or controller), we can set the member variable unit to Inch or Millimeter, for example:

```
this.unit = RS.Inch;
```

or

```
this.unit = RS.Millimeter;
```

Other settings that can be configured in the constructor are:

Variable	Possible Values	Default	Description
`this.outputOffsetPath`	true or false	false	Set to true to calculate tool radius compensation instead of using G41/G42.
`this.lineNumber`	Number	10	Initial number used for line numbering (N...).
`this.lineNumberIncrement`	Number	10	Line number incrementation from line to line.
`this.fileExtensions`	Array	`[ "nc", "cnc" ]`	List of file extensions the user can choose from when exporting through this post processor.

A complete constructor might look like this:

```
function MyGCode(documentInterface, newDocumentInterface) {
    GCodeBase.call(this, documentInterface, newDocumentInterface);

    this.unit = RS.Inch;
    this.outputOffsetPath = true;
    this.lineNumber = 1;
    this.lineNumberIncrement = 1;
}
```

Defining the G-Code Output

Code Blocks

G-Code output is structured into code blocks which are written to the output file when using the CAM export function in QCAD/CAM. For example, the first code block that is written is the code block "header". Each code block is defined in a member variable that identifies it by name, for example `this.header` contains the code block for the file header.

Code blocks can be overridden to contain any code / text. Code blocks typically contain variables to output the current X/Y coordinates, line number, feedrate, etc.

Let's say a single line is exported to G-Code. The line is exported with tool radius compensation to the left of the line (G41). The code blocks that are written to the output file are as follows:

Code block	Description	Example output
`header`	Header of the output (always the first code block).	G21 G17 G90 G40 G49 G80
`rapidMoveZ`	Rapid move to safety level.	G0 Z100
`rapidMove`	Rapid move to the first position of the first contour (here the start of the lead in for the line).	G0 X5 Y10
`toolHeader`	First block when a new tool comes into action (tool change).	T1
`rapidMoveZ`	Rapid move down to clearance level (hovering above material).	G0 Z5
`firstLinearMoveZ`	Cutting into material at plunge rate.	G1 Z-5 F50
`toolpathHeader`	First block of a new toolpath.	
`contourHeader`	First block of a new contour within a toolpath.	
`singleZPassHeader`	Header for single Z pass. We're cutting the whole depth in a single pass.	
`zPassFirstHeader`	First header of single or multiple Z passes.	
`linearMoveCompensationLeft`	Linear move to start point of the line along lead in, switching the tool radius compensation on.	G41 X10 Y10 F500
`linearMove`	Linear move along the line to the end point of the line.	X30 Y10
`linearMoveCompensationOff`	Linear move away from end point along lead out, switching the tool radius compensation off.	G40 X35 Y10
`zPassLastFooter`	Last footer of single or several Z passes.	
`singleZPassFooter`	Footer for single Z pass.	
`contourFooter`	End of contour.	
`toolpathFooter`	End of toolpath.	
`linearMoveZ`	Move out of material to clearance level.	Z5
`rapidMoveZ`	Move to safety level.	G0 Z100
`toolFooter`	End for this tool.	
`footer`	End of program.	M30

Many of these code blocks are typically unused and produce no output (empty code block).

Variables

Member variable	Variable name	Description
inputFileName	INFILENAME	Input file name (e.g. DXF file).
inputFilePath	INFILEPATH	Input file name with complete path.
fileName	FILENAME	Output file name.
filePath	FILEPATH	Output file name with complete path.

Member variable	Variable name	Description
dateTime	DATETIME	Date and time of export.
lineNumber	N	Current line number.
spindleSpeed	S	Spindle speed, typically rpm.
feedRate	F	Feedrate.
plungeRate	FP	Plunge rate.
xHomePosition	XH	Home position as configured in CAM configuration.
yHomePosition	YH	
zHomePosition	ZH	
zSafety	ZS	Z safety level for tool changes and start / end.
zClearance	ZU	Z clearance level for approaching the material.
zClearancePass	ZUP	Z clearance level between multiple passes.
zCuttingDepth	ZD	Z cutting level (total).
xFirstPosition	XF	First position of first contour in first toolpath.
yFirstPosition	YF	
xStartPosition	X1	Start of entity (start point of line / arc).
yStartPosition	Y1	
zStartPosition	Z1	
xPosition	X	Current target position. End point of the current line / arc.
yPosition	Y	
zPosition	Z	
xRelPosition	XR	Current target position as relative coordinate, relative to start point.
yRelPosition	YR	
zRelPosition	ZR	
iArcCenterInc	I	Center of arc, relative to start point.
jArcCenterInc	J	
iArcCenterAbs	IA	Center of arc, absolute.
jArcCenterAbs	JA	
arcRadius	RADIUS	Radius of arc.
arcDiameter	DIAMETER	Diameter of arc.
arcSweep	SWEEP	Sweep of arc (angle covered by arc in rad, negative for clockwise).
tool	T	Current tool.
toolRadius	TR	Tool radius.
toolDiameter	TD	Tool diameter.

Member variable	Variable name	Description
toolpathOrder	TOOLPATH_ORDER	Toolpath order in list of toolpaths, starting with 0 for the first toolpath.
toolCalls	TOOL_CALLS	Number of times this tool has been used.
toolpathName	TOOLPATH_NAME	Name of current toolpath (user defined).
programName	PROGRAM_NAME	Program name as configured by user in CAM configuration.
xMin	X_MIN	Minimum extents of drawing.
yMin	Y_MIN	
zMin	Z_MIN	
xMax	X_MAX	Maximum extents of drawing.
yMax	Y_MAX	
zMax	Z_MAX	

Examples

Changing the Header and Footer

To add or change G-Codes or M-Codes or other information in the header or footer, a configuration can override the member variables header and footer in the constructor, for example as follows:

```
this.header = [ "[N] G0 Z100" ];
this.footer = [ "[N] G0 Z100", "[N] M30" ];
```

Absolute I/J

To use absolute values for I/J values for the center of arcs, you can override the arc code blocks as follows:

```
this.firstArcCWMove = "[N] G2 [X] [Y] [IA] [JA] [F]";
this.arcCWMove = "[N] G2 [X] [Y] [IA] [JA] [F]";
this.firstArcCCWMove = "[N] G3 [X] [Y] [IA] [JA] [F]";
this.arcCCWMove = "[N] G3 [X] [Y] [IA] [JA] [F]";
```

Postprocessor Template File

QCAD/CAM automatically merges in a DXF template file with the same name as the postprocessor file (if available) when setting or switching the postprocessor. Let's say, the postprocessor is called "MyGCode", implemented in postprocessors/MyGCode.js. Create a DXF file with the tool definitions in postprocessors/MyGCode.dxf. This file might contain tool definitions, default layers or the outline of the available surface area. Note: Any visible content of the template file is automatically placed on a locked layer when merging, so it is only visible but not selectable.

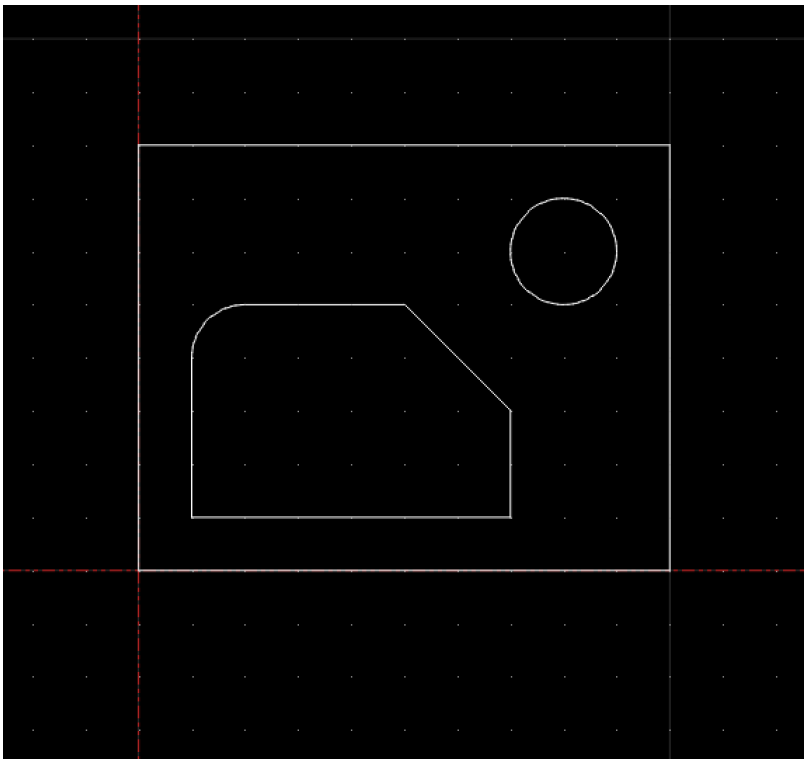
16.12.3 QCAD/CAM

Caution should be exercised when working with hazardous machinery. Simulation is no substitute for the careful verification of the accuracy and safety of your CNC programs. QCAD/CAM or any other software can make mistakes. Programs that appear correctly in the simulations can be very dangerous for operation on real machines. The safety and correctness of any CNC program when working on a real machine is the sole responsibility of the CNC machinist.

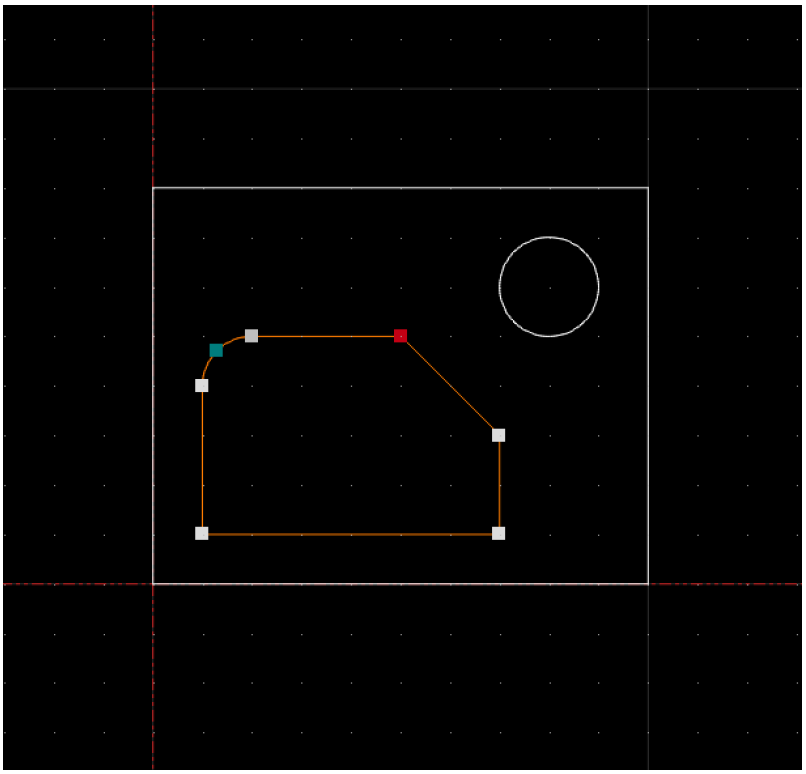
QCAD/CAM is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Creation, Preparation and Selection of the Geometry

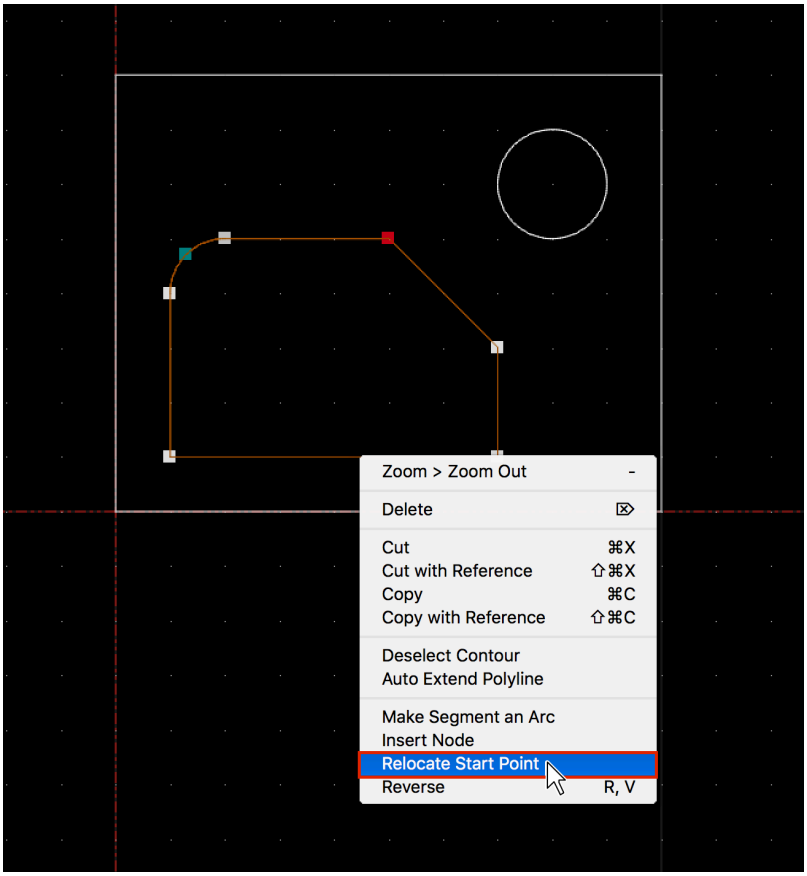
In the first step, the polylines are created which represent the geometry that has to be cut:



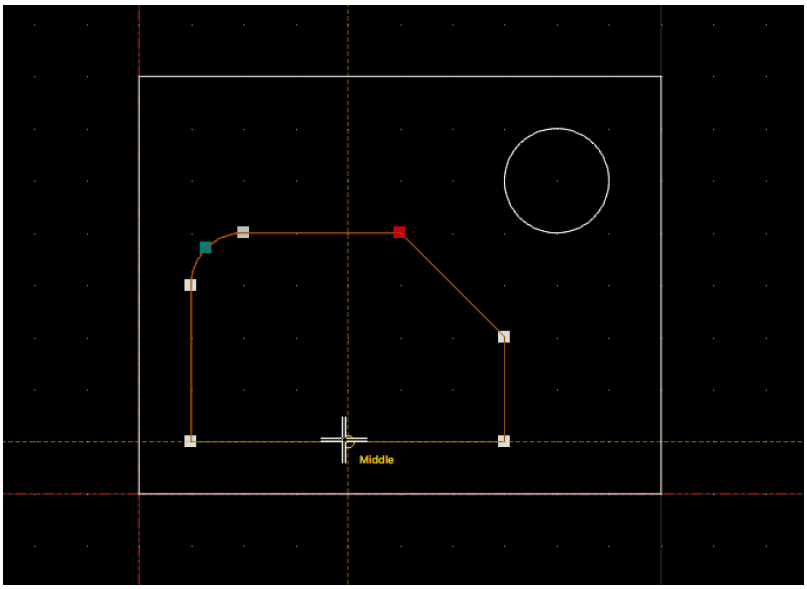
All contours must have a suitable start point. This must be a start point at which the contour can be approached using a lead in. For our example, this is not the case for the inner contour (the red reference point is the start point of the selected polyline):



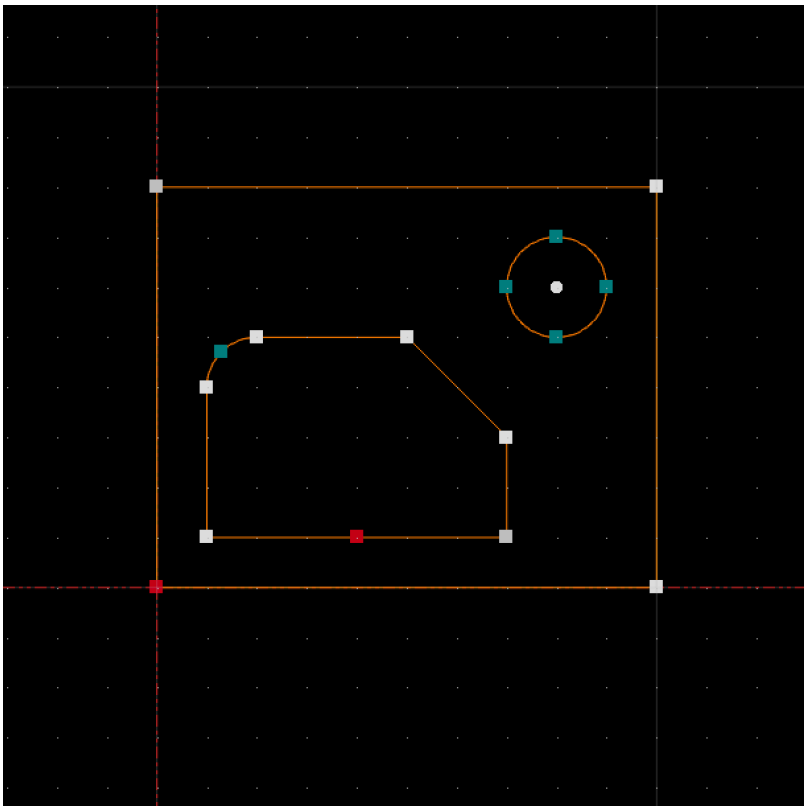
By right-clicking on the contour, the start point can be set to another position:



Click on the new position of the start point. This can be an existing node, or any other point on the polyline. Here the center of the lower segment is selected:



All contours are now ready for processing. With the function *Select* > *Select All* all start points of all contours can be highlighted (red):

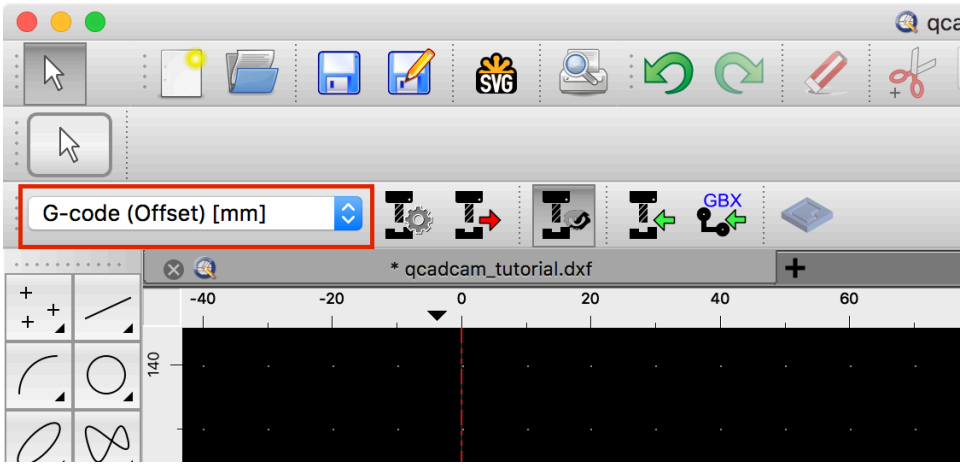


The circle is not a polyline, and may have its start point at any point.

Select all the contours that are to be processed with the same machining parameters and the same tool. In this example, these are all the contours in the drawing.

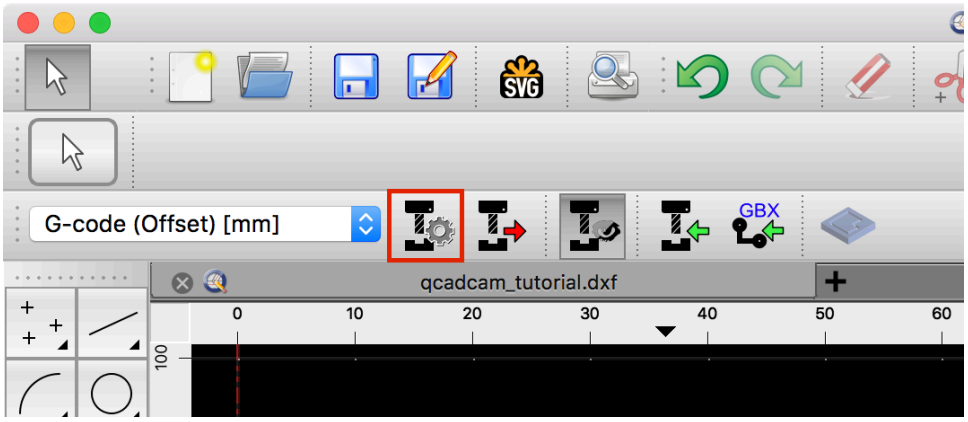
Choosing a Configuration

Choose the desired configuration (postprocessor) at the top left:

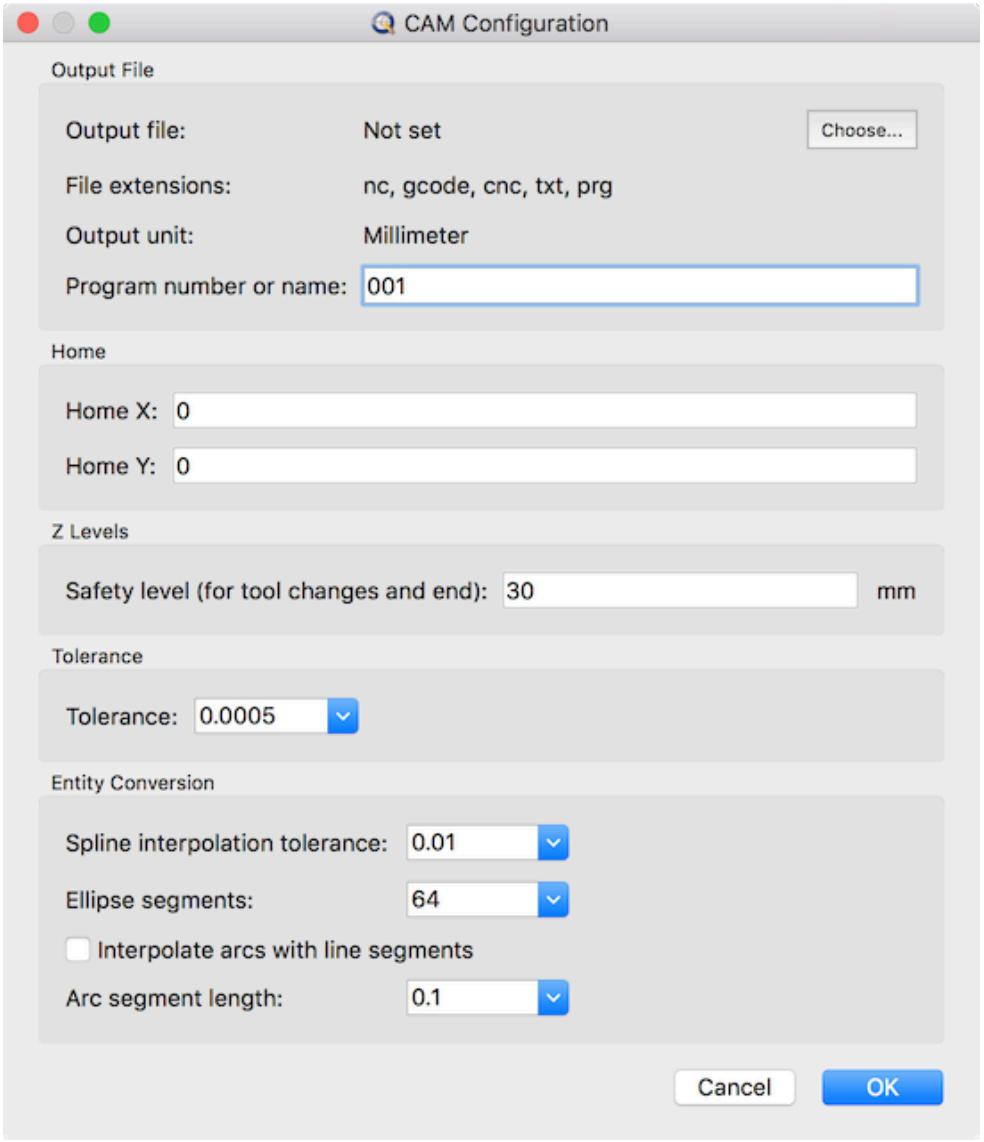


- **G-Code (G41/G42) [mm]**: This configuration can be used when the controller supports the commands G41 and G42 for radius compensation.
- **G-Code (Offset) [mm]**: This configuration can be used if no G41/G42 commands are available or G41/G42 should not be used. QCAD/CAM computes the tool path of the tool center based on the tool radius automatically in this case and outputs the appropriate offset coordinates.

Adjust the parameters of the selected configuration (*CAM* > *CAM Configuration...*):



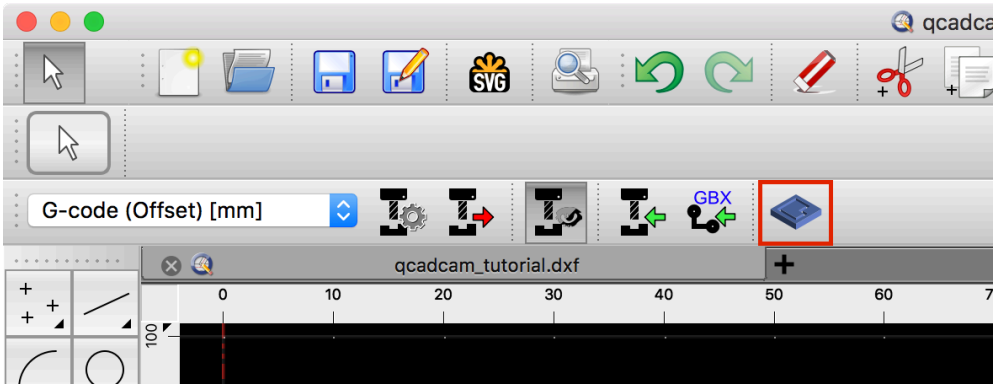
Enter the desired parameters:



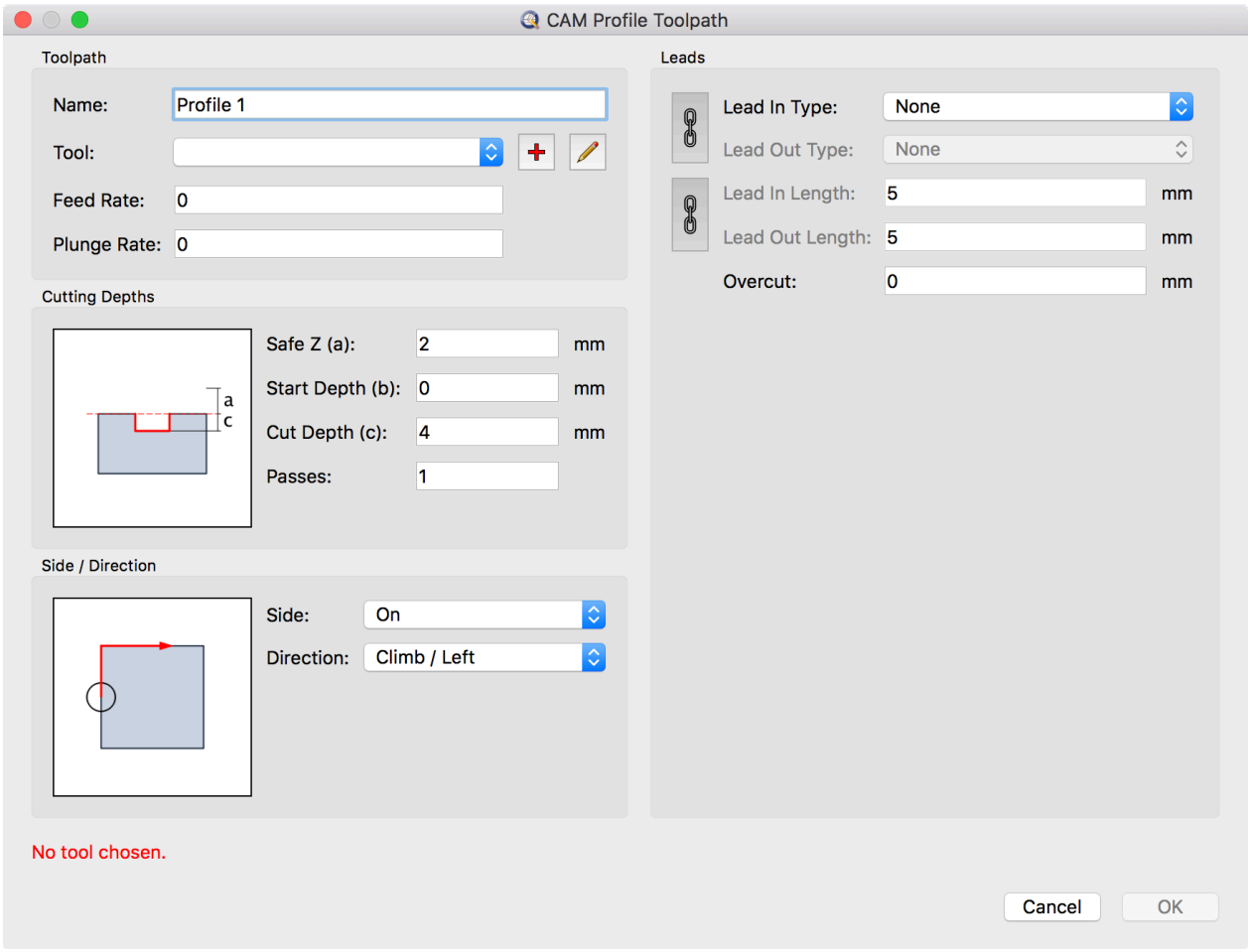
The output file can be set here or later when the output is generated for the first time. The safety Z level is used at the beginning and end of the program as well as before tool changes. Splines are interpolated as tangentially connected arcs within the given tolerance. Ellipses are interpolated as arcs with the given number of segments (per full ellipse). Arcs can optionally be interpolated as lines with the given segment length if the targeted controller does not support circular movements.

Creating a Profile Toolpath

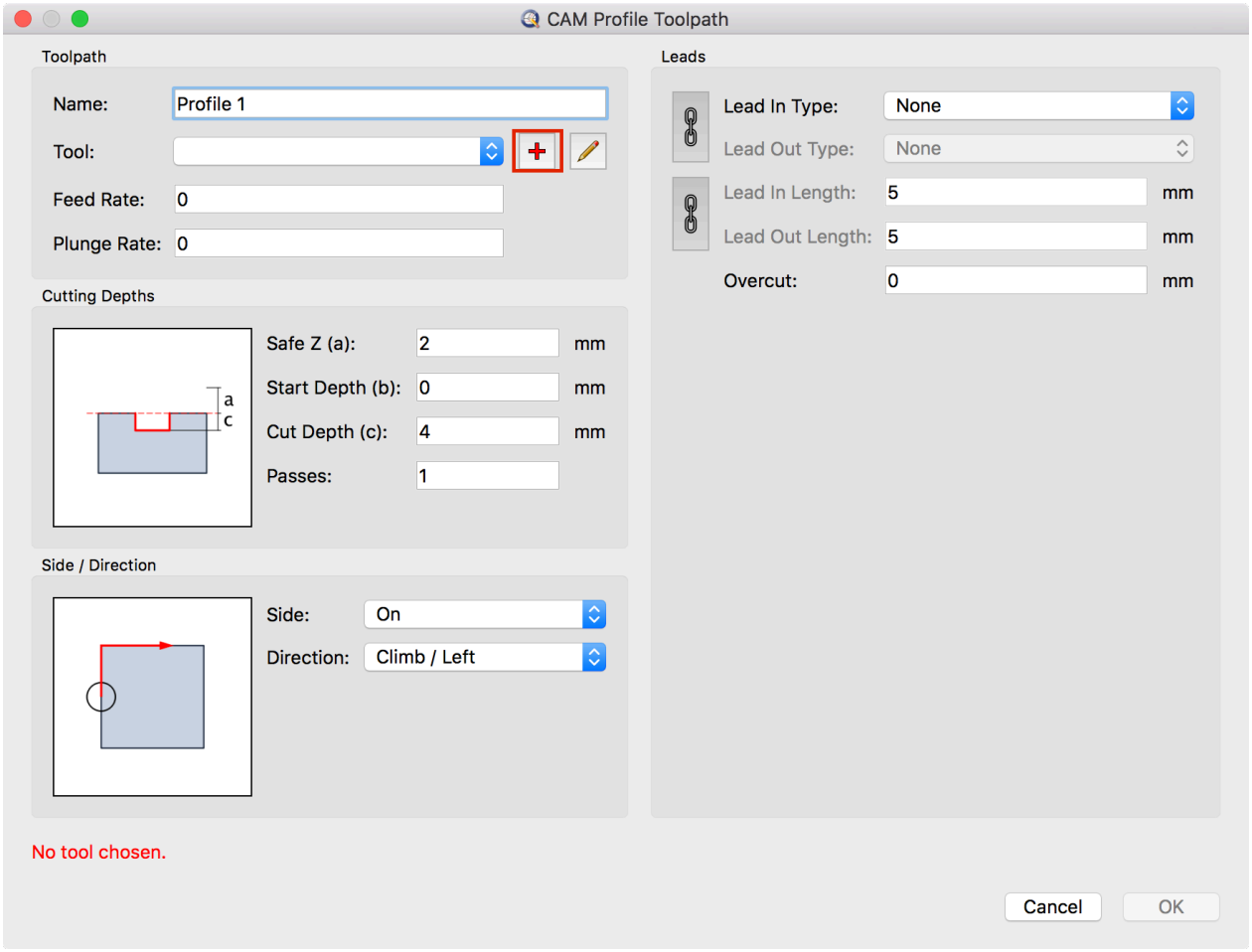
Start the tool to create a profile toolpath (*CAM > Add Profile Toolpath*):



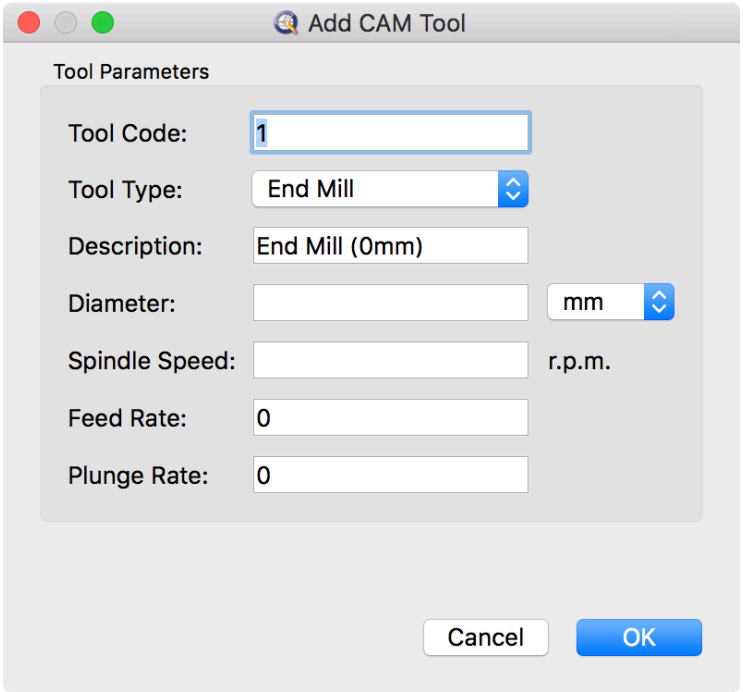
The dialog with profile toolpath parameters appears:



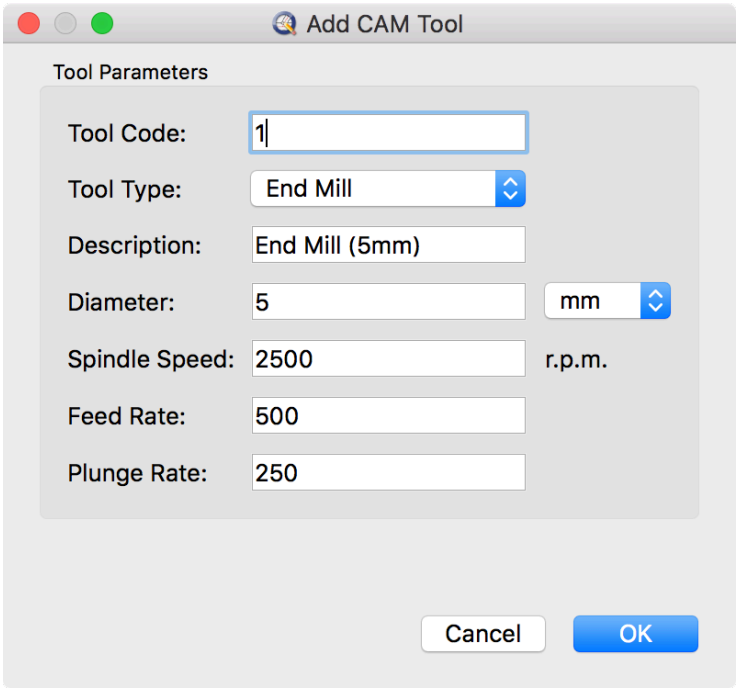
At the moment there are no tools defined yet, so a tool must be added first by clicking the button to add tools:



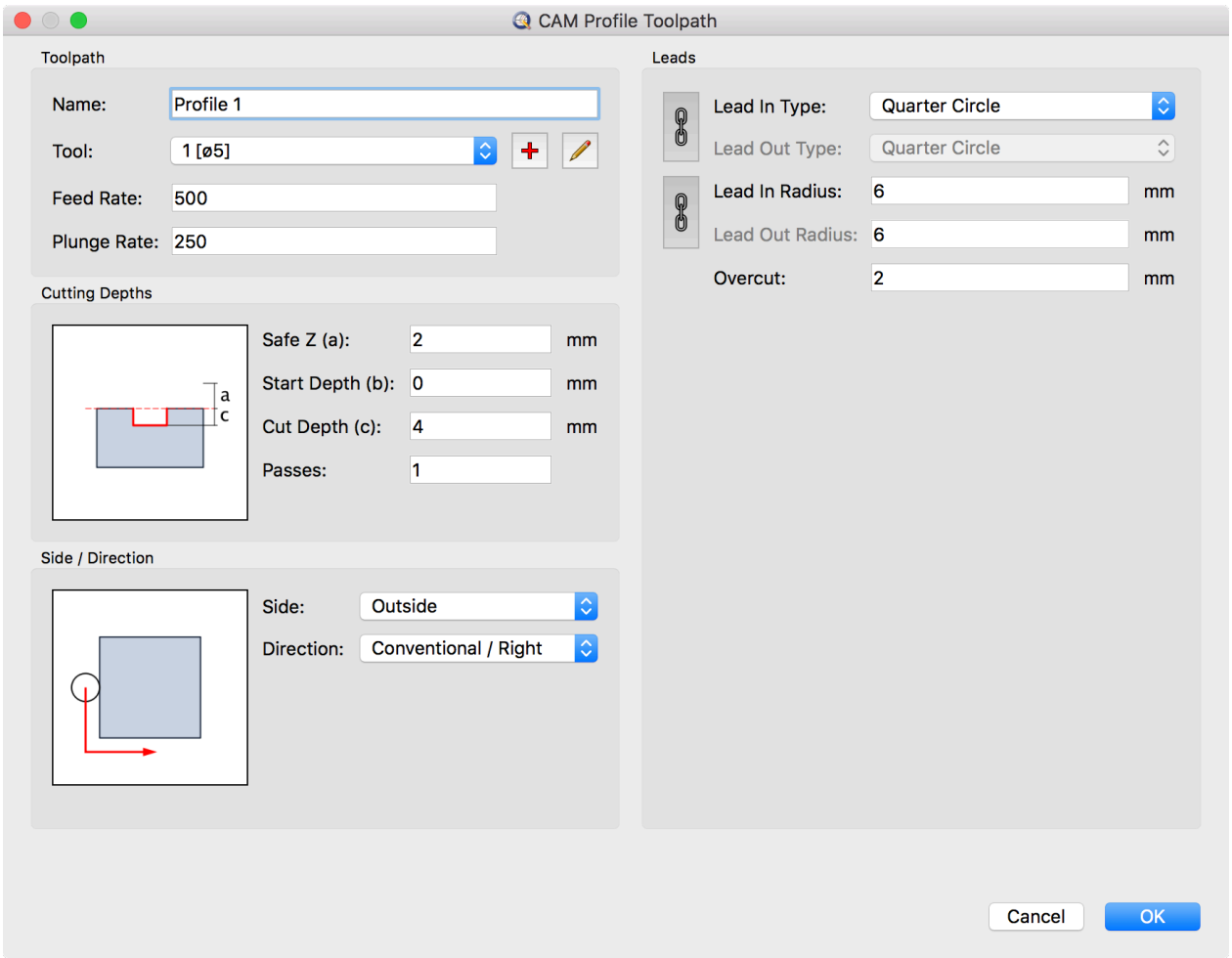
The dialog to add the new tool is displayed:



Enter the desired tool parameters and click *OK*:



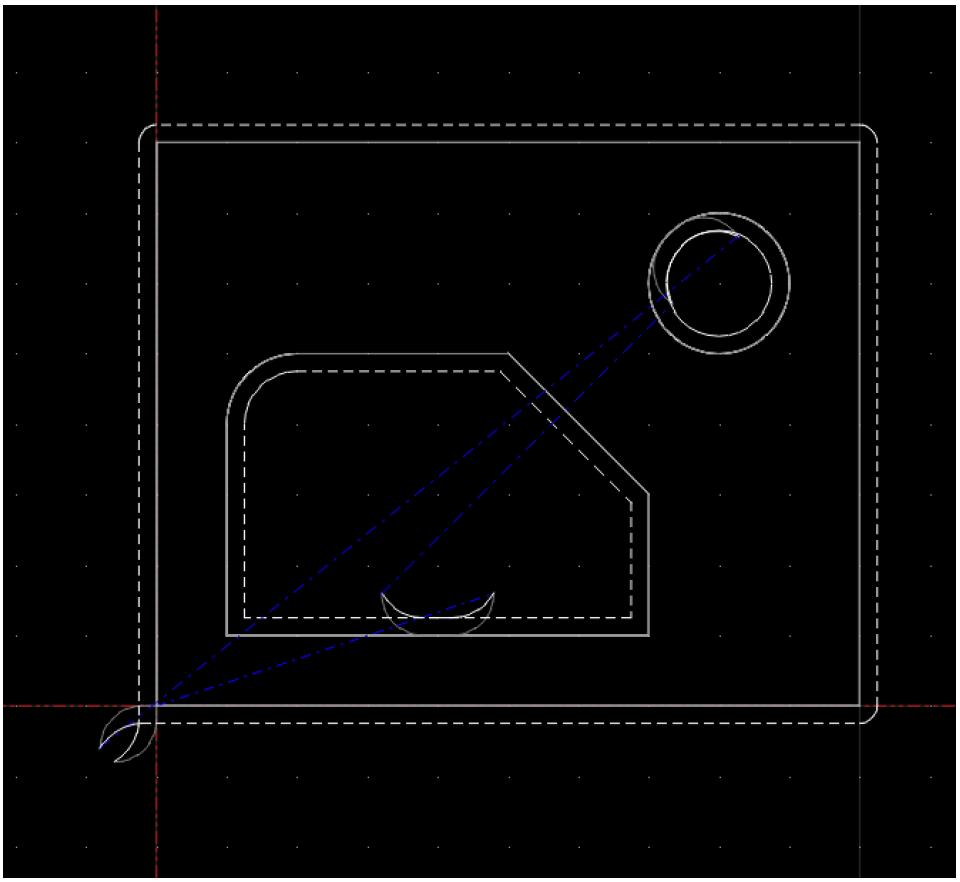
Adjust the parameters for the profile toolpath. For contours with radius compensation (side: Outside or Inside) a lead in path and a lead out path must be used. In this example, these lead paths are quarter circles with a radius which is slightly larger than the tool diameter:



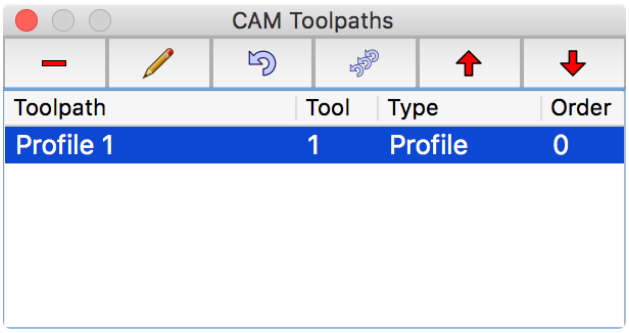
The overcut indicates by how much overlap is used at the start and end when processing the contour.

Click *OK* to generate the toolpath for this profile.

The toolpath is generated and displayed accordingly:



All generated toolpaths are also listed in the widget CAM Toolpaths (*CAM* > *Show CAM Toolpath List*):

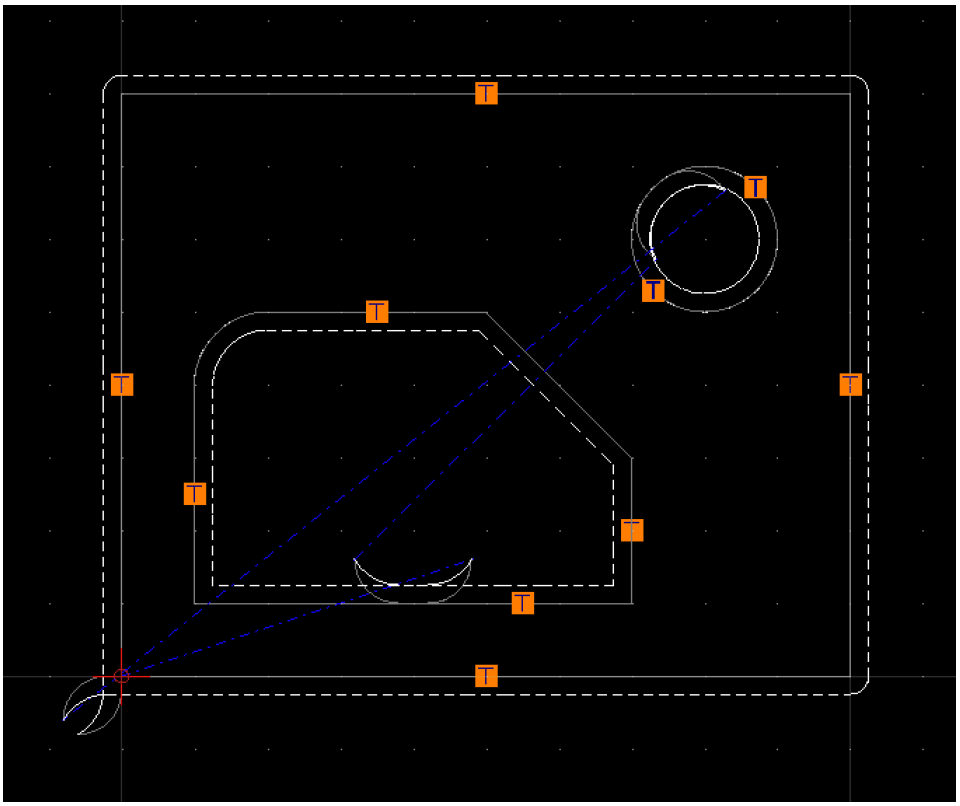


When G-code is generated, these toolpaths are processed in the order shown.

Working with Tabs

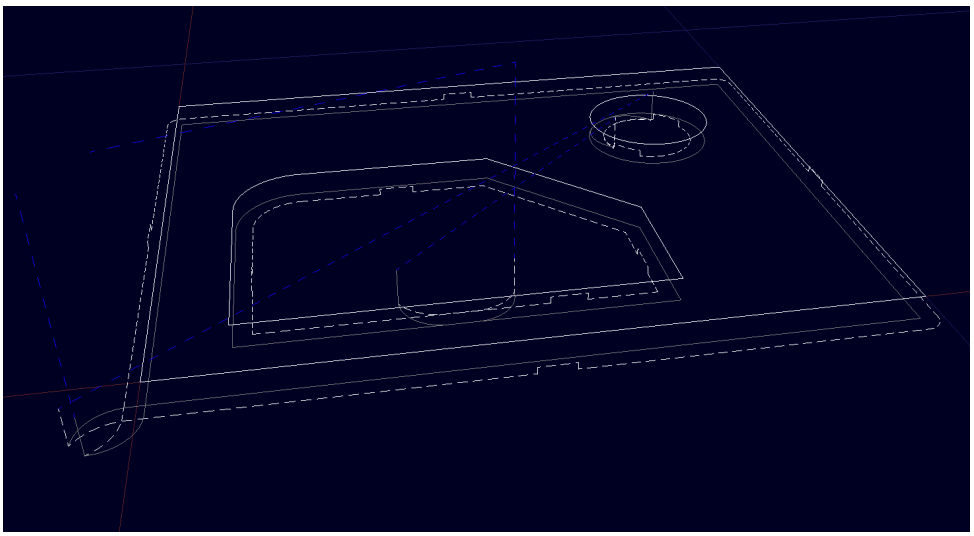
When cutting holes into sheet material, the material inside the hole becomes loose when the cut is almost complete. Such a loose piece of material can move into the cutter and get damaged in the process. Tabs are small connections between a cutout shape and its surrounding material which hold a piece in place while cutting it. Tabs are typically very small, so the piece can be easily broken or cut out of the sheet material once the toolpath is complete. To securely hold a piece in place, three or more tabs are typically used.

- Create a separate layer for the tabs you want to insert (e.g. layer "tab" with a distinguished color).
- Click the menu *CAM* > *Add Tab* to insert tabs.
- Enter the tab length and thickness in the options toolbar. The tab length is the total length of the bridge that is left in place. The thickness is the tab height, measured from the maximum cutting depth of the toolpath.
- Position the tab on the contour. Note that the tab is always placed on the originally drawn contour, even if cutter compensation is used:



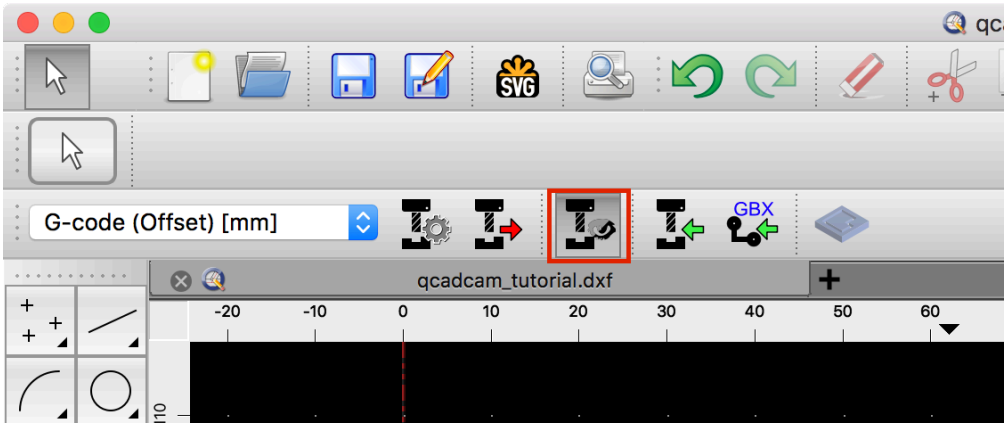
Tabs are shown as small squares with a T inside.

The effect that tabs have on the toolpath can best be observed in the 3D simulation view (see also Simulation below):



Showing / Hiding CAM Output

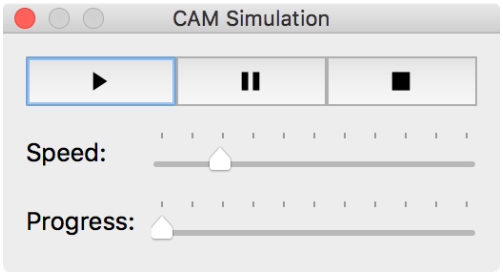
The tool paths can be shown or hidden with the button or menu *CAM* > *Show Toolpaths*:



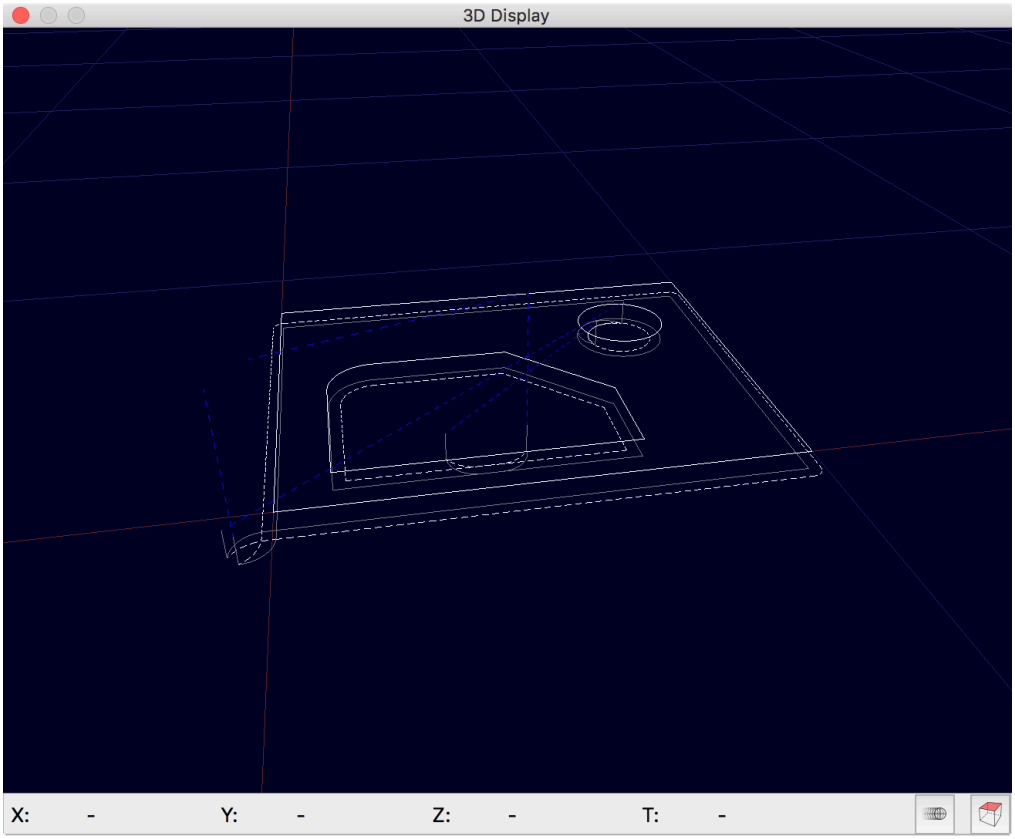
This function toggles the visibility of the layer CAM, the layer onto which all toolpaths are generated.

Simulation

The generated toolpaths can be simulated using the simulation widgets (*CAM* > *Show Simulation Panel*):

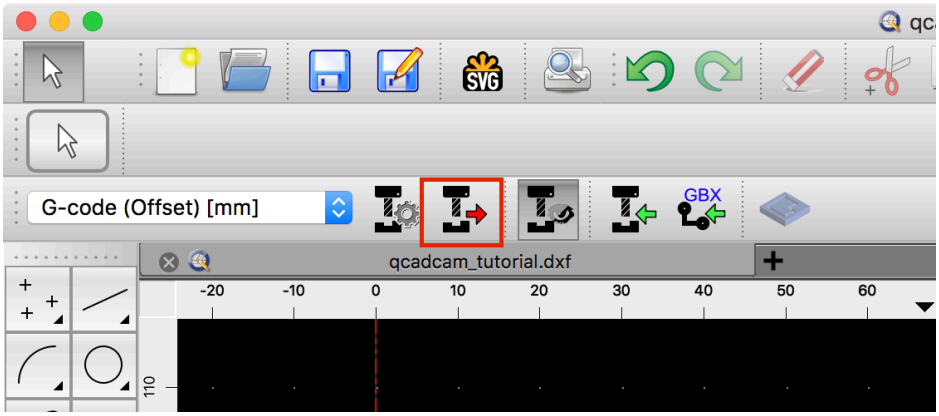


3D movements can sometimes be better observed using the 3D view (*CAM* > *Show 3D Simulation View*):



Generating G-Code

If the simulation appears to be satisfactory, the corresponding G-code can be generated using *CAM > CAM Export*:



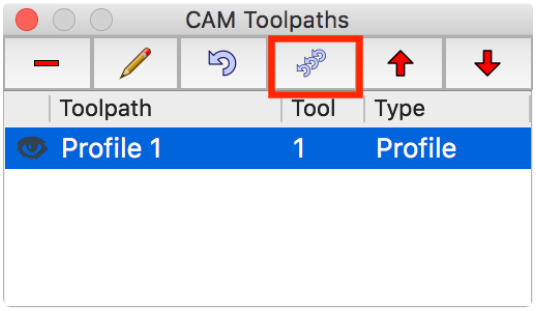
When generating G-Code for the first time, you are prompted for a filename for the output file. All future exports will overwrite that file without warning! The output file can be changed anytime in the configuration *CAM > CAM Configuration...*.

Adjusting the Geometry

If the geometry is adjusted, there are two ways to update the toolpaths that were generated from the old geometry.

Regenerating Toolpaths

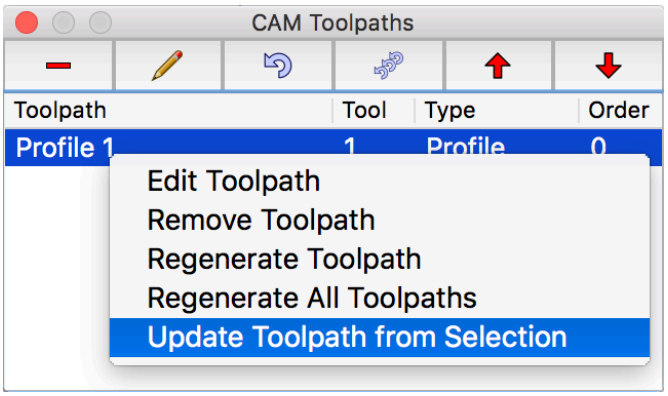
If a polyline has changed, the tool paths can be easily regenerated using *CAM > Regenerate All Toolpaths*:



Regenerating a Toolpath From Selection

To add new entities to an existing toolpath or base an existing toolpath on new entities, the tool path can be regenerated based on the current selection.

- Click the toolpath from the toolpath list.
- Select the entities to use in the graphics view.
- Right-click on the tool path in the toolpath list and choose *Update Toolpath from Selection*:



16.13 Video Tutorials

We offer an ever-expanding collection of video tutorials covering layers, blocks, snapping, printing and much more. You can find them all on our web site at:

https://qcad.org/documentation/video-tutorials