

# Contents

<b>Introduction</b>	<b>19</b>
<b>Preparations</b>	<b>23</b>
<b>1 Basics, Tools, and First Code</b>	<b>25</b>
1.1 Our First Qt Program . . . . .	25
1.1.1 Compiling a Qt Program . . . . .	27
1.2 Layouts, Object Hierarchy, and Memory Management . . . . .	29
1.2.1 How to Arrange Widgets Automatically . . . . .	29
1.2.2 Memory Management in Object Hierarchies . . . . .	31
1.2.3 Other Layout Types . . . . .	33
1.3 Signals and Slots . . . . .	35
1.3.1 The Simplest Case: A Slot Responds to a Signal . . . . .	35
1.3.2 Signals Carrying Additional Information and How They Are Processed . . . . .	36
1.4 Base Classes in Qt . . . . .	39
1.4.1 Classes Derived from QObject . . . . .	39
1.4.2 QString and Other Classes not Derived from QObject . . . . .	40
1.4.3 The Qt Inheritance Hierarchy . . . . .	41
1.5 Qt at a Glance . . . . .	42
1.5.1 The Qt Libraries . . . . .	42
1.5.2 Tools and Utilities . . . . .	47
1.5.3 Examples and Demos . . . . .	58
1.6 How to Use the Documentation . . . . .	59

<b>2</b>	<b>The Tools Needed to Create Dialogs</b>	<b>61</b>
2.1	What's the Difference Between Dialogs and Widgets? . . . . .	62
2.1.1	Inheriting from QObject . . . . .	64
2.1.2	More Complex Layouts . . . . .	65
2.1.3	Increasing Usability . . . . .	68
2.1.4	Implementing Slots . . . . .	70
2.2	Separation of GUI and Processing Logic . . . . .	74
2.2.1	Alternative Design . . . . .	74
2.2.2	Declaring and Sending Out Signals . . . . .	76
2.2.3	Using Your Own Signals . . . . .	79
<b>3</b>	<b>GUI Design Using the Qt Designer</b>	<b>81</b>
3.1	Dialogs "By Mouse Click" . . . . .	81
3.1.1	Making Layouts With the Designer . . . . .	84
3.1.2	The Property Editor . . . . .	85
3.1.3	The Preview . . . . .	88
3.1.4	Signal/Slot Connections . . . . .	88
3.1.5	The Tab Sequence . . . . .	89
3.1.6	Shortcuts and Buddies . . . . .	90
3.2	Integrating Designer-generated Files into Your Qt Project . . . . .	91
3.2.1	Using Designer-generated Classes as Helper Classes . . . . .	92
3.2.2	Always Having Designer-generated Widgets Available . . . . .	94
3.2.3	Multiple Inheritance . . . . .	95
3.3	Automatic Signal/Slot Connections . . . . .	97
3.4	Including Derived Classes in the Designer . . . . .	99
3.5	The Resource Editor . . . . .	99
<b>4</b>	<b>Developing a GUI Application Based on a Main Window</b>	<b>101</b>
4.1	The Anatomy of the Main Window . . . . .	101
4.2	Deriving from QMainWindow . . . . .	103
4.3	Creating a Main Window with the Qt Designer . . . . .	106
4.3.1	Adding Menu Bars . . . . .	107
4.3.2	Recycling Actions in the Toolbar . . . . .	108

---

4.3.3	Integrating the Main Window with Your Source Code . . .	110
4.4	Making the Most of the Status Bar . . . . .	118
4.4.1	Temporary Messages . . . . .	120
4.4.2	Normal Messages . . . . .	120
4.4.3	Permanent Messages . . . . .	121
4.5	Toolbars . . . . .	125
4.6	How Do Actions Work? . . . . .	126
4.6.1	How to Instantiate QAction Manually . . . . .	127
4.6.2	Selectable Actions . . . . .	128
4.6.3	Grouped Actions . . . . .	128
4.7	Dock Windows . . . . .	130
4.7.1	Positioning Dock Windows . . . . .	131
4.7.2	A Dock Window for Our Editor . . . . .	133
4.8	Saving Preferences . . . . .	136
4.8.1	Extending CuteEdit . . . . .	139
<b>5</b>	<b>Laying Out Widgets</b>	<b>141</b>
5.1	Manual Layout . . . . .	141
5.2	Automatic Layout . . . . .	143
5.2.1	Horizontal and Vertical Layout . . . . .	144
5.2.2	Grid Layout . . . . .	148
5.2.3	Nested Layouts . . . . .	149
5.3	Splitter . . . . .	150
5.3.1	Behavior During Size Changes . . . . .	150
5.3.2	Saving Splitter Positions and Determining the Widget Size	151
5.3.3	Defining Relative Sizes . . . . .	152
5.3.4	Customizing Handles . . . . .	153
5.3.5	Layout for Languages Written from Right to Left . . . . .	156
5.4	Stacked Layouts . . . . .	157
5.4.1	The Alternative: Stacked Widgets . . . . .	157
5.4.2	When to Use Stacked Layouts and Widgets . . . . .	157

---

<b>6</b>	<b>Dialogs</b>	<b>161</b>
6.1	Modal Dialogs . . . . .	161
6.2	Non-modal Dialogs . . . . .	163
6.2.1	Usability Problems . . . . .	163
6.3	Semi-modal Dialogs . . . . .	164
6.4	Avoiding Bloated Dialogs . . . . .	164
6.5	Ready-made Dialogs in Qt . . . . .	166
6.5.1	Message Dialogs . . . . .	166
6.5.2	Error Messages That Are Only Visible Once . . . . .	174
6.5.3	File Selection Dialogs . . . . .	175
6.5.4	Input Dialogs . . . . .	179
6.5.5	Font Selection Dialog . . . . .	182
6.5.6	Color Selection and Printing Dialog . . . . .	183
<b>7</b>	<b>Events, Drag and Drop, and the Clipboard</b>	<b>185</b>
7.1	Event Loop and Event Handler . . . . .	185
7.2	Handling Events . . . . .	186
7.2.1	Using Specialized Event Handlers . . . . .	186
7.2.2	Using the General Event Handler . . . . .	189
7.3	Using Event Filters . . . . .	190
7.4	Drag and Drop . . . . .	194
7.4.1	MIME Types . . . . .	194
7.4.2	The Drag Side . . . . .	196
7.4.3	The Drop Side . . . . .	198
7.5	The Clipboard . . . . .	201
<b>8</b>	<b>Displaying Data Using "Interview"</b>	<b>207</b>
8.1	Underlying Concepts . . . . .	208
8.1.1	The View Classes . . . . .	210
8.1.2	The Model Classes . . . . .	211
8.2	Displaying Directory Hierarchies . . . . .	212
8.2.1	Using View Classes in the Designer . . . . .	214
8.2.2	Implementing the Functionality of the File Selection Dialog	216

---

8.3	The String Lists Model . . . . .	221
8.4	Implementing Your Own Models . . . . .	222
8.4.1	An Address Book Model . . . . .	222
8.4.2	Making Your Own Models Writable . . . . .	227
8.5	Sorting and Filtering Data with Proxy Models . . . . .	231
8.5.1	Adjustments to the User Interface . . . . .	232
8.6	Making Entries Selectable with Checkboxes . . . . .	234
8.7	Designing Your Own Proxy Models . . . . .	237
8.8	Implementing Drag and Drop in Models . . . . .	241
8.9	Your Own Delegates . . . . .	245
8.10	Without Your Own Data Source: The Standard Model . . . . .	249
8.11	Element-based Views Without Model Access . . . . .	251
8.11.1	Items . . . . .	251
8.11.2	The List View . . . . .	251
8.11.3	The Tree View . . . . .	252
8.11.4	The Table View . . . . .	253
8.11.5	Cloning Items . . . . .	254
<b>9</b>	<b>The QSql Module</b>	<b>257</b>
9.1	Structure of the QSql Module . . . . .	257
9.2	Selecting the Appropriate Driver . . . . .	258
9.3	Making a Connection . . . . .	260
9.4	Making Queries . . . . .	261
9.5	Transactions . . . . .	264
9.6	Embedded Databases . . . . .	264
9.7	Using SQL Model Classes with Interview . . . . .	265
9.7.1	Displaying SQL Tables Without Foreign Keys in Table and Tree Views . . . . .	265
9.7.2	Resolving Foreign Key Relations . . . . .	266
9.7.3	Displaying Query Results . . . . .	267
9.7.4	Editing Strategies . . . . .	268
9.7.5	Errors in the Table Model . . . . .	270

<b>10 The Graphics Library "Arthur"</b>	<b>271</b>
10.1 Colors . . . . .	271
10.1.1 The RGB Color Space . . . . .	272
10.1.2 Other Color Spaces . . . . .	273
10.1.3 Color Selection Dialog . . . . .	275
10.2 Painting with Qt . . . . .	276
10.3 Geometrical Helper Classes . . . . .	278
10.4 How to Paint on Widgets . . . . .	280
10.4.1 How to Prevent Monitor Flicker . . . . .	282
10.5 Using QPainter in Practice . . . . .	283
10.5.1 Drawing a Pie Chart . . . . .	284
10.5.2 Defining the Widget Size . . . . .	289
10.5.3 The Diagram Application . . . . .	290
10.6 Transformations of the Coordinate System . . . . .	290
10.6.1 Transformations in Practice . . . . .	293
10.7 QImage . . . . .	297
10.7.1 Storage Formats, Transparency, and Color Palettes . . . . .	297
10.7.2 Reading out Pixels Line by Line . . . . .	298
10.8 SVG Support . . . . .	300
10.9 Printing with QPainter . . . . .	302
10.9.1 Digression: Making Screenshots . . . . .	304
10.9.2 Printing an Image File . . . . .	305
10.9.3 Generating PDFs . . . . .	306
10.9.4 The Test Application . . . . .	306
10.10 Complex Graphics . . . . .	307
10.10.1 Clipping . . . . .	307
10.10.2 Painter Paths . . . . .	309
10.10.3 Composition Modes . . . . .	310
<b>11 Input/Output Interfaces</b>	<b>317</b>
11.1 The QIODevice Class Hierarchy . . . . .	317
11.1.1 Derived Classes . . . . .	318
11.1.2 Opening I/O Devices . . . . .	319

---

11.2	Access to Local Files . . . . .	320
11.3	Serializing Objects . . . . .	322
11.3.1	Defining Serialization Operators . . . . .	325
11.3.2	Saving Serialized Data to a File and Reading from It . . . . .	326
11.4	Starting and Controlling Processes . . . . .	328
11.4.1	Synchronous Use of QProcess . . . . .	328
11.4.2	Asynchronous Use of QProcess . . . . .	330
11.5	Communication in the Network . . . . .	332
11.5.1	Name Resolution with QHostInfo . . . . .	333
11.5.2	Using QTcpServer and QTcpSocket . . . . .	333
<b>12</b>	<b>Threading with QThread</b>	<b>337</b>
12.1	Using Threads . . . . .	338
12.2	Synchronizing Threads . . . . .	341
12.2.1	The Consumer/Producer Pattern . . . . .	342
12.3	Thread-dependent Data Structures . . . . .	345
12.4	Using Signals and Slots Between Threads . . . . .	347
12.5	Your Own Event Loops for Threads . . . . .	350
12.5.1	Communication via Events Without a Thread-based Event Loop . . . . .	352
<b>13</b>	<b>Handling XML with QtXml</b>	<b>353</b>
13.1	The SAX2 API . . . . .	354
13.1.1	How It Works . . . . .	354
13.1.2	Reimplementing a Default Handler to Read RSS Feeds . . . . .	355
13.1.3	Digression: Equipping the RSS Reader with a GUI and Network Capability . . . . .	361
13.2	The DOM API . . . . .	366
13.2.1	Reading in and Processing XML Files . . . . .	367
13.2.2	Searching for Specific Elements . . . . .	370
13.2.3	Manipulating the DOM Tree . . . . .	371
13.2.4	The DOM Tree as XML Output . . . . .	372

---

<b>14 Internationalization</b>	<b>375</b>
14.1 Translating Applications into Other Languages . . . . .	375
14.1.1 Preparing the Application . . . . .	376
14.1.2 Processing Translation Sources with Linguist . . . . .	377
14.1.3 Using Translations in the Program . . . . .	378
14.1.4 Adding Notes for the Translation . . . . .	380
14.1.5 Specifying the Translation Context . . . . .	380
14.1.6 Internationalizing Strings Outside Qt Classes . . . . .	381
<b>Appendixes</b>	<b>383</b>
<b>A Debugging Help</b>	<b>385</b>
A.1 Debugging Functions . . . . .	385
A.1.1 Simple Debug Output . . . . .	386
A.1.2 Errors and Warnings . . . . .	387
A.1.3 Customizing the Output of Debugging Functions . . . . .	388
A.2 Ways to Eliminate Errors . . . . .	390
A.2.1 Checking Assertions . . . . .	390
A.2.2 Checking Pointers . . . . .	391
A.2.3 Common Linker Errors . . . . .	392
<b>B Tulip: Containers and Algorithms</b>	<b>393</b>
B.1 Iterators . . . . .	394
B.1.1 STL-Style Iterators . . . . .	395
B.1.2 Java-Style Iterators . . . . .	396
B.2 Lists . . . . .	398
B.2.1 Simple List (QList) . . . . .	400
B.2.2 Linked List (QLinkedList) . . . . .	401
B.2.3 Vectors (QVector) . . . . .	401
B.3 Stacks and Queues . . . . .	403
B.3.1 Stacks (QStack) . . . . .	403
B.3.2 Queues (QQueue) . . . . .	404
B.4 Associative Arrays . . . . .	404
B.4.1 Dictionaries (QMap) . . . . .	404



B.4.2	Allowing Several Identical Keys (QMultiMap) . . . . .	407
B.4.3	Hash Tables with QHash . . . . .	409
B.4.4	Hash-based Amounts with QSet . . . . .	411
B.5	Algorithms . . . . .	412
B.5.1	The foreach Keyword . . . . .	412
B.5.2	Sorting . . . . .	413
B.5.3	Sorting in Unsorted Containers . . . . .	414
B.5.4	Copying Container Areas . . . . .	415
B.5.5	Binary Search in Sorted Containers . . . . .	416
B.5.6	Counting the Number of Occurences of Equal Elements . .	418
B.5.7	Deleting Pointers in Lists . . . . .	418
B.5.8	Checking that Data Structures Have Identical Elements . .	419
B.5.9	Filling Data Structures . . . . .	420
B.5.10	Swapping Values . . . . .	420
B.5.11	Minimum, Maximum, and Threshold Values . . . . .	421
B.5.12	Determining Absolute Value . . . . .	422
B.6	Qt-specific Type Definitions . . . . .	422
B.6.1	Integer types . . . . .	422
B.6.2	Floating-point Values . . . . .	423
B.6.3	Shortcuts for Common Types . . . . .	423

<b>Index</b>		<b>425</b>
--------------	--	------------