Building Portable C++ Applications with YACL

M. A. Sridhar
University of South Carolina
Contents

Preface xv

Chapter 1 Introduction 1
  1.1 About YACL 1
  1.1.1 Design goals 2
  1.2 The YACL classes 4
  1.3 Comparison with other class libraries 5
  1.4 Organization of this book 6

I The Base Classes 9

Chapter 2 Basic Concepts and Overview 11
  2.1 Terminology 11
     2.1.1 Ownership of objects 13
  2.2 Naming conventions 13
     2.2.1 Method names 14
  2.3 Overview of the base classes 15

Chapter 3 Protocol for All Objects 18
  3.1 Methods for CL_Object 19
     3.1.1 Comparison methods 19
     3.1.2 The assignment operator 21
     3.1.3 Copying of objects 22
  3.2 Object storage and retrieval 23
  3.3 Bindings and notification 23
     3.3.1 The Binding classes 24
     3.3.2 Notification of change 28
     3.3.3 Using notification codes 29
     3.3.4 Pre-change notification 30
     3.3.5 Implementing dependent notification 31
  3.4 Run-time type identification 33
  3.5 Summary 34
Chapter 4  Elementary Data Types  35
  4.1  Overriding inherited methods  36
  4.2  Strings  37
    4.2.1  Construction and conversions  37
    4.2.2  Basic operations  40
    4.2.3  Searching and substitution  40
    4.2.4  Case conversions  41
    4.2.5  String modification  42
    4.2.6  String operations  43
    4.2.7  Substrings  44
    4.2.8  Elementary string parsing  44
    4.2.9  Interaction with streams  47
    4.2.10  Implementation of strings  48
  4.3  Date and time classes  48
    4.3.1  Methods for CL_Date  49
    4.3.2  Methods for CL_TimeOfDay  52
    4.3.3  Representation  53
  4.4  A simple reminder program  53
  4.5  The class CL_Integer  54
  4.6  ByteArrays and ByteStrings  55
  4.7  Summary  56

Chapter 5  Container Classes  57
  5.1  Using templates  57
  5.2  Copy or pointer?  58
  5.3  Kinds of containers  59
  5.4  Interaction with template parameters  61
    5.4.1  Null values  62
    5.4.2  Comparison  63
    5.4.3  Object input/output  64
  5.5  Notification by containers  64
  5.6  The Sequence  65
    5.6.1  Construction  66
    5.6.2  Element access and modification  66
    5.6.3  Ordering-related methods  68
    5.6.4  Sequence methods  69
    5.6.5  Using Sequence for queues and stacks  71
    5.6.6  Predefined instantiations of the Sequence template  71
    5.6.7  Another reminder program  72
    5.6.8  Protected methods  74
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.7</td>
<td>The Set</td>
<td>75</td>
</tr>
<tr>
<td>5.7.1</td>
<td>The protocol for Set</td>
<td>75</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Instantiations of the Set class template</td>
<td>77</td>
</tr>
<tr>
<td>5.7.3</td>
<td>Iteration over a Set</td>
<td>79</td>
</tr>
<tr>
<td>5.8</td>
<td>The BitSet</td>
<td>83</td>
</tr>
<tr>
<td>5.8.1</td>
<td>BitSets versus IntegerSets</td>
<td>85</td>
</tr>
<tr>
<td>5.9</td>
<td>The Map class</td>
<td>86</td>
</tr>
<tr>
<td>5.9.1</td>
<td>Protocol for the Map</td>
<td>87</td>
</tr>
<tr>
<td>5.9.2</td>
<td>Using the Map class</td>
<td>88</td>
</tr>
<tr>
<td>5.10</td>
<td>The Tree and Tree-walking classes</td>
<td>90</td>
</tr>
<tr>
<td>5.10.1</td>
<td>Traversing a Tree</td>
<td>92</td>
</tr>
<tr>
<td>5.11</td>
<td>Instantiating new containers</td>
<td>95</td>
</tr>
<tr>
<td>5.12</td>
<td>Containers within containers</td>
<td>98</td>
</tr>
<tr>
<td>5.13</td>
<td>Container implementations</td>
<td>99</td>
</tr>
<tr>
<td>5.13.1</td>
<td>Implementation of Sequence</td>
<td>99</td>
</tr>
<tr>
<td>5.13.2</td>
<td>Representation of the Set class</td>
<td>105</td>
</tr>
<tr>
<td>5.13.3</td>
<td>Implementation of the Map</td>
<td>106</td>
</tr>
<tr>
<td>5.14</td>
<td>Summary</td>
<td>106</td>
</tr>
</tbody>
</table>

**Chapter 6  The B-tree and Related Classes**  107

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Overview of the B-tree classes</td>
<td>108</td>
</tr>
<tr>
<td>6.2</td>
<td>Using the GenericBTree</td>
<td>110</td>
</tr>
<tr>
<td>6.3</td>
<td>The GenericBTree and related classes</td>
<td>112</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Comparators</td>
<td>113</td>
</tr>
<tr>
<td>6.3.2</td>
<td>The class GenericBTree</td>
<td>113</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Iterators for generic B-trees</td>
<td>115</td>
</tr>
<tr>
<td>6.4</td>
<td>The template-based B-tree classes</td>
<td>115</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Some design tradeoffs</td>
<td>116</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Iterators</td>
<td>117</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Instantiating custom templates</td>
<td>117</td>
</tr>
<tr>
<td>6.5</td>
<td>Implementing other containers using B-trees</td>
<td>117</td>
</tr>
<tr>
<td>6.6</td>
<td>Design and implementation</td>
<td>117</td>
</tr>
<tr>
<td>6.6.1</td>
<td>The class CL_BTreeNodeSpace</td>
<td>118</td>
</tr>
<tr>
<td>6.6.2</td>
<td>The class CL_GenericBTreeNode</td>
<td>120</td>
</tr>
<tr>
<td>6.6.3</td>
<td>Traversing a generic B-tree</td>
<td>122</td>
</tr>
<tr>
<td>6.6.4</td>
<td>Implementation of B-tree algorithms</td>
<td>123</td>
</tr>
<tr>
<td>6.6.5</td>
<td>Implementation of the GenericBTreeIterator</td>
<td>125</td>
</tr>
<tr>
<td>6.6.6</td>
<td>Implementation of the template-based B-tree</td>
<td>126</td>
</tr>
<tr>
<td>6.7</td>
<td>Summary</td>
<td>127</td>
</tr>
</tbody>
</table>
Chapter 7  Data Storage and Database Classes  128
7.1  The BinaryFile class  128
7.2  The SlottedFile class  129
   7.2.1  Using the SlottedFile  130
   7.2.2  Protected methods  134
   7.2.3  Implementing the SlottedFile  134
7.3  The ByteStringStore class  138
   7.3.1  Implementing the ByteStringStore  140
7.4  Disk-based B-trees  141
   7.4.1  Fixed-length items  141
   7.4.2  Variable-length items  143
7.5  A simple database  146
   7.5.1  Encapsulating a table row  146
   7.5.2  Encapsulating a table  147
   7.5.3  Implementing the Table class  148
   7.5.4  Indexing  148
7.6  Summary  149

Chapter 8  Object Input/Output and Persistence  150
8.1  Printable representations of objects  151
   8.1.1  Interfacing with C++ iostreams  151
   8.1.2  Using the strstream and related classes  152
8.2  The class CL_Stream  152
8.3  The class ByteStream  154
8.4  Persistent objects  155
8.5  Persistent containers  157
   8.5.1  Duplication  158
   8.5.2  Object identification  161
   8.5.3  Memory leaks  164
8.6  Summary  164

Chapter 9  Designing New Classes  165
9.1  The Book class  166
   9.1.1  Overriding the comparison method  168
   9.1.2  Support for notification  170
   9.1.3  Object input-output and persistence  172
9.2  Designing a new container  173
   9.2.1  Platform dependent issues  174
9.3  Using notification  174
9.4  Summary  176
Chapter 10 Error Handling 178

10.1 The Error class 178
   10.1.1 Changing the error handler 179

10.2 Detecting memory-related errors 180
   10.2.1 Using the module 180
   10.2.2 The MemoryLeakChecker class 181
   10.2.3 Memory usage information 182
   10.2.4 Implementation of the module 183

10.3 Summary 183

II The Graphic User Interface Classes 185

Chapter 11 Basic GUI Concepts 187

11.1 Models, views, and controllers 190
   11.1.1 Models and views 190
   11.1.2 Events and controllers 191

11.2 Overview of the GUI classes 191
   11.2.1 Visual objects 192
   11.2.2 Simple visual objects 192
   11.2.3 Collections of visual objects 193
   11.2.4 The Controller 195
   11.2.5 The Application class 195
   11.2.6 The Menu and related classes 197
   11.2.7 Graphics-related classes 198
   11.2.8 Points, rectangles, and graphics objects 198
   11.2.9 Coordinate conventions 199
   11.2.10 Class hierarchy 200

11.3 A simple GUI application 201
11.4 Implementation considerations 202
   11.4.1 Look and feel 202
   11.4.2 Non-portable methods 203
   11.4.3 Platform-specific hooks 204

11.5 Summary 204

Chapter 12 The Visual Object 205

12.1 Creation and destruction 205
   12.1.1 The visual element 206

12.2 Model access 208
   12.2.1 Refreshing model contents 209
12.3 View-related methods 209
  12.3.1 Shape and position 209
  12.3.2 The title string 210
  12.3.3 The Font method 210
  12.3.4 Visibility and graying 211
12.4 Events and related methods 211
  12.4.1 Event dispatching 212
  12.4.2 Event types 214
  12.4.3 Event dependents 215
12.5 Cursors 218
12.6 The protected interface 219
12.7 Summary 220

Chapter 13 Composite Visual Objects 221
  13.1 Overview 221
  13.2 The CompositeVObject as a main window 222
  13.3 Creation and destruction 224
  13.4 Controlling shape 226
  13.5 Managing children 227
  13.6 Tab stops 227
  13.7 The CompositeVObject’s model 228
  13.8 Dialogs 228
    13.8.1 Modal dialogs 230
    13.8.2 Modeless dialogs 232
  13.9 Dynamic dialogs 234
  13.10 Standard dialogs 238
  13.11 Summary 239

Chapter 14 Data Display Classes 240
  14.1 Protocol for SimpleVObject 241
  14.2 Models 241
  14.3 Selections 242
  14.4 Ownership of models and selections 243
  14.5 Buttons, button groups, and labels 243
    14.5.1 The PushButton 243
    14.5.2 Toggle buttons 244
    14.5.3 Button groups 245
    14.5.4 The Label class 247
    14.5.5 A calculator application 247
  14.6 The StringView classes 251
    14.6.1 Sharing the model 254
14.6.2 A sample application: displaying environment variables 254
14.6.3 Sample application: dual StringViews 257
14.7 The ScrollBar classes 260
14.7.1 A sample application 262
14.8 The StringEditor and TextEditor classes 264
14.9 Summary 264

Chapter 15 Menus 266
15.1 The Menu classes 268
  15.1.1 The Menu protocol 268
  15.1.2 The MenuItem class 272
15.2 An example application 273
  15.2.1 Creating the menu bar 275
  15.2.2 Adding event dependents 277
  15.2.3 Handling events 278
15.3 Summary 279

Chapter 16 Graphics and Drawing 280
16.1 Painting a window 281
16.2 Graphics objects 282
  16.2.1 The Rectangle 283
  16.2.2 Other GraphicObjects 284
16.3 The DisplaySurface class 284
  16.3.1 Creating a DisplaySurface 285
  16.3.2 Drawing tools and operations 285
  16.3.3 Colors 287
16.4 DisplaySurface modes 288
  16.4.1 Rubber banding 289
  16.4.2 An example application 289
16.5 Classes related to DisplaySurface 292
16.6 Text and fonts 292
  16.6.1 Font methods 294
  16.6.2 Font selection 295
  16.6.3 Enumerating fonts 296
  16.6.4 Relationships between the graphics objects 297
16.7 Bitmaps 297
  16.7.1 A simple animation program 298
16.8 Example: Implementing large StringViews 300
  16.8.1 Construction 303
  16.8.2 Initialization and String painting 304
  16.8.3 Handling changes in model contents 306