THE RING PROGRAMMING LANGUAGE

```
changeringoperator + plus
changeringkeyword SEE PRINT

Print 5 plus 5

changeringoperator plus +
changeringkeyword PRINT SEE
```

Syntax Flexibility

```
Load "gameengine.ring"

func main
	oGame = New Game
	{
		title = "My First Game"
		sprite
		{
			type = GE_TYPE_PLAYER
			x=400 y=400 width=100 height=100
			file = "images/player.png"
			transparent = true
			Animate=false
			Move=true
			Scaled=true
		}
	}
```

Declarative Programming

```
# Natural Code
new program {
	Accept 2 numbers then print the sum
}

# Natural Code Implementation
class program
	# Keywords
		Accept=0 numbers=0 then=0 print=0 the=0 sum=0

	# Execution
	func braceexpreval x
		value = x
	func getnumbers
		for x=1 to value
			see "Enter Number ("+x+") :" give nNumber
			aNumbers + nNumber
		next
	func getsum
		nSUm = 0
		for x in aNumbers nSum+= x next
		see "The Sum : " + nSum
	private
		value=0  aNumbers=[]
```

Natural Language Programming

The Complete Reference

Ring Documentation

Release 1.19

Ring Team

Dec 07, 2023

CONTENTS

1 Applications developed in a few hours **2**
- 1.1 Quotes about Ring . . . 2
- 1.2 FetchStockData Application . . . 10
- 1.3 Google API Shortener Application . . . 11
- 1.4 Video-Music-Player Application . . . 12
- 1.5 Fifteen Puzzle Game . . . 12
- 1.6 Analog Clock . . . 13
- 1.7 Squares Puzzle Game . . . 14
- 1.8 Samples in this book . . . 15
- 1.9 Innovative . . . 22
- 1.10 Practical . . . 23

2 Introduction **25**
- 2.1 Motivation (1) . . . 25
- 2.2 Motivation (2) . . . 26
- 2.3 Motivation (3) . . . 26
- 2.4 Motivation (4) . . . 27
- 2.5 Motivation (5) . . . 27
- 2.6 Ring and other languages . . . 27
- 2.7 History . . . 28
- 2.8 Features . . . 29
- 2.9 License . . . 32

3 Using Ring Notepad **33**
- 3.1 Ring Notepad - Main Window . . . 33
- 3.2 Creating and running your first Console Application . . . 34
- 3.3 Creating and running your first GUI/WebAssembly/Mobile Application . . . 36
- 3.4 Creating and running your first Web Application . . . 38
- 3.5 Creating and running your first Desktop/Mobile Game . . . 39
- 3.6 The Main File in the Project . . . 40
- 3.7 The File Menu . . . 41
- 3.8 The Edit Menu . . . 42
- 3.9 The View Menu . . . 43
- 3.10 The Program Menu . . . 44
- 3.11 The Browser Menu . . . 45
- 3.12 The Tools Menu . . . 45
- 3.13 The Distribute Menu . . . 45
- 3.14 The Help Menu . . . 46

4 Getting Started - First Style **47**

4.1 Hello World . . . 47
4.2 Run the program . . . 47
4.3 Create Executable File . . . 47
4.4 Not Case-Sensitive . . . 47
4.5 Multi-Line literals . . . 48
4.6 Getting Input . . . 48
4.7 No Explicit End For Statements . . . 48
4.8 Using ? to print expression then new line . . . 48
4.9 Writing Comments . . . 49

5 Getting Started - Second Style **50**
5.1 Hello World . . . 50
5.2 Run the program . . . 50
5.3 Create Executable File . . . 50
5.4 Not Case-Sensitive . . . 50
5.5 Multi-Line literals . . . 51
5.6 Getting Input . . . 51
5.7 No Explicit End For Statements . . . 51
5.8 Writing Comments . . . 51

6 Getting Started - Third Style **53**
6.1 Hello World . . . 53
6.2 Run the program . . . 53
6.3 Create Executable File . . . 53
6.4 Not Case-Sensitive . . . 53
6.5 Multi-Line literals . . . 54
6.6 Getting Input . . . 54
6.7 No Explicit End For Statements . . . 54
6.8 Writing Comments . . . 54
6.9 Puts() function . . . 55
6.10 Print() function . . . 55
6.11 Print2Str() Function . . . 55
6.12 GetString() function . . . 56
6.13 GetNumber() function . . . 56

7 Variables **57**
7.1 Dynamic Typing . . . 57
7.2 Deep Copy . . . 58
7.3 Implicit Conversion . . . 58

8 Operators **59**
8.1 Arithmetic Operators . . . 59
8.2 Relational Operators . . . 59
8.3 Logical Operators . . . 60
8.4 Bitwise Operators . . . 60
8.5 Assignment Operators . . . 60
8.6 Misc Operators . . . 61
8.7 Operators Precedence . . . 61

9 Control Structures - First Style **63**
9.1 Branching . . . 63
9.2 Looping . . . 64
9.3 Using The Step option with For in . . . 66
9.4 Using For in to modify lists . . . 66
9.5 Do Again Loop . . . 66

9.6 Exit Command . . . 67
9.7 Exit from two loops . . . 67
9.8 Loop Command . . . 67
9.9 Short-circuit evaluation . . . 68
9.10 Comments about evaluation . . . 69

10 Control Structures - Second Style **70**
10.1 Branching . . . 70
10.2 Looping . . . 71
10.3 Exceptions . . . 73

11 Control Structures - Third Style **74**
11.1 Branching . . . 74
11.2 Looping . . . 75
11.3 Exceptions . . . 78

12 Getting Input **79**
12.1 Give Command . . . 79
12.2 GetChar() Function . . . 79
12.3 Input() Function . . . 80

13 Functions - First Style **81**
13.1 Define Functions . . . 81
13.2 Call Functions . . . 82
13.3 Declare parameters . . . 82
13.4 Send Parameters . . . 82
13.5 Main Function . . . 83
13.6 Variables Scope . . . 83
13.7 Return Value . . . 84
13.8 Recursion . . . 84

14 Functions - Second Style **85**
14.1 Define Functions . . . 85
14.2 Call Functions . . . 86
14.3 Declare parameters . . . 86
14.4 Send Parameters . . . 86
14.5 Main Function . . . 87
14.6 Variables Scope . . . 87
14.7 Return Value . . . 88
14.8 Recursion . . . 88

15 Functions - Third Style **89**
15.1 Define Functions . . . 89
15.2 Call Functions . . . 90
15.3 Declare parameters . . . 90
15.4 Send Parameters . . . 90
15.5 Main Function . . . 91
15.6 Variables Scope . . . 91
15.7 Return Value . . . 92
15.8 Recursion . . . 92

16 Program Structure **93**
16.1 Source Code File Sections . . . 93
16.2 Using Many Source Code Files . . . 93
16.3 Load Package . . . 94

16.4 Load Again . . . 95

17 Lists **96**
17.1 Create Lists . . . 96
17.2 Add Items . . . 97
17.3 Get List Size . . . 97
17.4 Delete Item From List . . . 97
17.5 Get List Item . . . 98
17.6 Set List Item . . . 98
17.7 Search . . . 98
17.8 Sort . . . 99
17.9 Reverse . . . 100
17.10 Insert Items . . . 100
17.11 Nested Lists . . . 101
17.12 Copy Lists . . . 101
17.13 First-class lists . . . 101
17.14 Using Lists during definition . . . 102
17.15 Passing Lists to Functions . . . 103
17.16 Access List Items by String Index . . . 103
17.17 Passing Parameters or Arguments Using List . . . 104
17.18 Passing Parameters or Arguments Using List Array . . . 104
17.19 Return Parameters as List or Hash Table . . . 104
17.20 Creating a Multi-Dimensional Array using List . . . 106
17.21 Swap Items . . . 107

18 Strings **108**
18.1 String Literals . . . 108
18.2 Get String Length . . . 108
18.3 Convert Letters Case . . . 108
18.4 Access String Letters . . . 109
18.5 Left() Function . . . 109
18.6 Right() Function . . . 110
18.7 Trim() Function . . . 110
18.8 Copy() Function . . . 110
18.9 Lines() Function . . . 110
18.10 Substr() Function . . . 111
18.11 Find substring . . . 111
18.12 Get substring from position to end . . . 111
18.13 Get Number of Characters From Position . . . 112
18.14 Transform Substring To Another Substring . . . 112
18.15 strcmp() Function . . . 112
18.16 str2list() and list2str() Functions . . . 113
18.17 Merge binary characters . . . 114

19 Date and Time **116**
19.1 Clock() Function . . . 116
19.2 ClocksPerSecond() Function . . . 116
19.3 Time() Function . . . 117
19.4 Date() Function . . . 117
19.5 TimeList() Function . . . 117
19.6 AddDays() Function . . . 119
19.7 DiffDays() Function . . . 119
19.8 EpochTime() Function . . . 119

20 Check Data Type and Conversion **121**

20.1 Check Data Type . . . 121
20.2 IsString() Function . . . 121
20.3 IsNumber() Function . . . 122
20.4 IsList() Function . . . 122
20.5 Type() Function . . . 122
20.6 IsNULL() Function . . . 122
20.7 Check Character . . . 123
20.8 IsAlNum() Function . . . 123
20.9 IsAlpha() Function . . . 124
20.10 IsCntrl() Function . . . 124
20.11 IsDigit() Function . . . 124
20.12 IsGraph() Function . . . 125
20.13 IsLower() Function . . . 125
20.14 IsPrint() Function . . . 125
20.15 IsPunct() Function . . . 125
20.16 IsSpace() Function . . . 126
20.17 IsUpper() Function . . . 126
20.18 IsXdigit() Function . . . 126
20.19 Conversion . . . 127
20.20 Number() Function . . . 127
20.21 String() Function . . . 127
20.22 Ascii() Function . . . 128
20.23 Char() Function . . . 128
20.24 Hex() Function . . . 128
20.25 Dec() Function . . . 128
20.26 Str2hex() Function . . . 129
20.27 Hex2str() Function . . . 129

21 Mathematical Functions **130**
21.1 List of functions . . . 130
21.2 Example . . . 131
21.3 Random() Function . . . 132
21.4 SRandom() Function . . . 133
21.5 Unsigned() Function . . . 134
21.6 Decimals() Functions . . . 134
21.7 Using _ in numbers . . . 135
21.8 Using f after numbers . . . 135
21.9 Using Hexadecimal Numbers . . . 135

22 Files **137**
22.1 Read() Function . . . 138
22.2 Write() Function . . . 138
22.3 Dir() Function . . . 139
22.4 Rename() Function . . . 139
22.5 Remove() Function . . . 139
22.6 Fopen() Function . . . 140
22.7 Fclose() Function . . . 140
22.8 Fflush() Function . . . 140
22.9 Freopen() Function . . . 140
22.10 Tempfile() Function . . . 141
22.11 Tempname() Function . . . 141
22.12 Fseek() Function . . . 142
22.13 Ftell() Function . . . 142
22.14 Rewind() Function . . . 142

22.15 Fgetpos() Function . . . 142
22.16 Fsetpos() Function . . . 143
22.17 Clearerr() Function . . . 143
22.18 Feof() Function . . . 143
22.19 Ferror() Function . . . 143
22.20 Perror() Function . . . 143
22.21 Fgetc() Function . . . 144
22.22 Fgets() Function . . . 144
22.23 Fputc() Function . . . 144
22.24 Fputs() Function . . . 144
22.25 Ungetc() Function . . . 144
22.26 Fread() Function . . . 145
22.27 Fwrite() Function . . . 145
22.28 Fexists() Function . . . 145
22.29 Direxists() Function . . . 145
22.30 Getpathtype() Function . . . 146
22.31 Getfilesize() Function . . . 146
22.32 Example . . . 147
22.33 Numbers and Bytes . . . 148

23 System Functions **150**
23.1 System() Function . . . 151
23.2 SysGet() Function . . . 151
23.3 SysSet() Function . . . 151
23.4 SysUnset() Function . . . 151
23.5 IsMSDOS() Function . . . 151
23.6 IsWindows() Function . . . 152
23.7 IsWindows64() Function . . . 152
23.8 IsUnix() Function . . . 152
23.9 IsMacOSX() Function . . . 152
23.10 IsLinux() Function . . . 152
23.11 IsFreeBSD() Function . . . 153
23.12 IsAndroid() Function . . . 153
23.13 Example . . . 153
23.14 GetArch() Function . . . 153
23.15 Windowsnl() Function . . . 154
23.16 Get Command Line Arguments . . . 155
23.17 Get Active Source File Name . . . 155
23.18 PrevFileName() Function . . . 156
23.19 CurrentDir() Function . . . 156
23.20 ExeFileName() Function . . . 157
23.21 ChDir() Function . . . 157
23.22 ExeFolder() Function . . . 157
23.23 Version() Function . . . 157
23.24 Shutdown() Function . . . 157
23.25 NofProcessors() Function . . . 158

24 Eval() and Debugging **159**
24.1 Try/Catch/Done . . . 159
24.2 Eval() Function . . . 160
24.3 Raise() Function . . . 160
24.4 Assert() Function . . . 161

25 Demo Programs **163**

25.1 Language Shell . . . 163
25.2 Main Menu . . . 164

26 ODBC Functions **167**
26.1 odbc_init() Function . . . 167
26.2 odbc_drivers() Function . . . 168
26.3 odbc_datasources() Function . . . 168
26.4 odbc_close() Function . . . 168
26.5 Print List of ODBC Drivers . . . 168
26.6 Print List of ODBC Data Sources . . . 169
26.7 odbc_connect() Function . . . 170
26.8 odbc_disconnect() Function . . . 170
26.9 Open and Close Connection . . . 170
26.10 odbc_execute() Function . . . 171
26.11 odbc_colcount() Function . . . 171
26.12 odbc_fetch() Function . . . 171
26.13 odbc_getdata() Function . . . 171
26.14 Execute Query and Print Result . . . 171
26.15 odbc_tables() Function . . . 172
26.16 odbc_columns() Function . . . 173
26.17 odbc_autocommit() Function . . . 173
26.18 odbc_commit() Function . . . 174
26.19 odbc_rollback() Function . . . 174
26.20 Transactions and Using Commit and Rollback . . . 174
26.21 Save and Restore images . . . 175

27 MySQL Functions **176**
27.1 MySQL_Info() Function . . . 176
27.2 MySQL_Init() Function . . . 177
27.3 MySQL_Error() Function . . . 177
27.4 MySQL_Connect() Function . . . 177
27.5 MySQL_Close() Function . . . 177
27.6 MySQL_Query() Function . . . 178
27.7 Create Database . . . 178
27.8 Create Table and Insert Data . . . 178
27.9 MySQL_Insert_ID() Function . . . 179
27.10 MySQL_Result() Function . . . 180
27.11 MySQL_Next_Result() Function . . . 180
27.12 Print Query Result . . . 181
27.13 MySQL_Columns() Function . . . 181
27.14 MySQL_Result2() Function . . . 182
27.15 MySQL_Escape_String() Function . . . 183
27.16 Save Image inside the database . . . 183
27.17 Restore Image From The Database . . . 183
27.18 MySQL_AutoCommit() Function . . . 184
27.19 MySQL_Commit() Function . . . 184
27.20 MySQL_Rollback() Function . . . 184
27.21 Transaction Example . . . 184

28 SQLite Functions **186**
28.1 sqlite_init() function . . . 186
28.2 sqlite_open() function . . . 186
28.3 sqlite_execute() function . . . 186
28.4 sqlite_close() function . . . 186

28.5 Example . . . 187

29 PostgreSQL Functions **189**
29.1 Loading the library . . . 189
29.2 Examples . . . 189
29.3 RingPostgreSQL Constants . . . 192
29.4 RingPostgreSQL Functions . . . 193

30 Security and Internet Functions **197**
30.1 MD5() Function . . . 198
30.2 SHA1() Function . . . 199
30.3 SHA256() Function . . . 199
30.4 SHA512() Function . . . 199
30.5 SHA384() Function . . . 200
30.6 SHA224() Function . . . 200
30.7 SupportedCiphers() Function . . . 201
30.8 Encrypt() Function . . . 201
30.9 Decrypt() Function . . . 201
30.10 Encryption and Decryption Example . . . 201
30.11 File Hash . . . 202
30.12 Randbytes() Function . . . 203
30.13 rsa_generate() Function . . . 203
30.14 rsa_export_params() Function . . . 204
30.15 rsa_import_params() Function . . . 205
30.16 rsa_export_pem() Function . . . 205
30.17 rsa_import_pem() Function . . . 206
30.18 rsa_is_privatekey() Function . . . 206
30.19 rsa_encrypt_pkcs() Function . . . 207
30.20 rsa_decrypt_pkcs() Function . . . 208
30.21 rsa_encrypt_oaep() Function . . . 209
30.22 rsa_decrypt_oaep() Function . . . 210
30.23 rsa_encrypt_raw() Function . . . 211
30.24 rsa_decrypt_raw() Function . . . 212
30.25 rsa_sign_pkcs() Function . . . 214
30.26 rsa_signhash_pkcs() Function . . . 214
30.27 rsa_verify_pkcs() Function . . . 215
30.28 rsa_verifyhash_pkcs() Function . . . 216
30.29 rsa_sign_pss() Function . . . 217
30.30 rsa_signhash_pss() Function . . . 218
30.31 rsa_verify_pss() Function . . . 219
30.32 rsa_verifyhash_pss() Function . . . 220
30.33 openssl_versiontext() Function . . . 221
30.34 openssl_version() Function . . . 221
30.35 Large Files Hash . . . 221
30.36 Download() Function . . . 222
30.37 SendEmail() Function . . . 222

31 Object Oriented Programming (OOP) **224**
31.1 Classes and Objects . . . 224
31.2 Access Objects Using Braces . . . 226
31.3 Composition . . . 227
31.4 Setter and Getter . . . 228
31.5 Private Attributes and Methods . . . 229
31.6 Operator Overloading . . . 230

31.7 Inheritance . . . 233
31.8 Dynamic Attributes . . . 234
31.9 Packages . . . 234
31.10 Printing Objects . . . 235
31.11 Find() and List of Objects . . . 235
31.12 Sort() and List of Objects . . . 236
31.13 Using Self.Attribute and Self.Method() . . . 238
31.14 Using This.Attribute and This.Method() . . . 239
31.15 Using This in the class region as Self . . . 240
31.16 Default value for object attributes . . . 241
31.17 Command: New From . . . 241

32 Functional Programming **243**
32.1 Pure Functions . . . 243
32.2 First-class Functions . . . 244
32.3 Higher-order Functions . . . 245
32.4 Anonymous and Nested Functions . . . 245
32.5 Equality of functions . . . 247

33 Reflection and Meta-programming **248**
33.1 locals() Function . . . 249
33.2 globals() Function . . . 249
33.3 functions() Function . . . 250
33.4 cfunctions() Function . . . 251
33.5 islocal() Function . . . 251
33.6 isglobal() Function . . . 252
33.7 isfunction() Function . . . 252
33.8 iscfunction() Function . . . 253
33.9 packages() Function . . . 253
33.10 ispackage() Function . . . 254
33.11 classes() Function . . . 255
33.12 isclass() Function . . . 255
33.13 packageclasses() Function . . . 256
33.14 ispackageclass() Function . . . 256
33.15 classname() Function . . . 257
33.16 objectid() Function . . . 258
33.17 isobject() Function . . . 258
33.18 attributes() Function . . . 258
33.19 methods() Function . . . 259
33.20 isattribute() Function . . . 259
33.21 isprivateattribute() Function . . . 260
33.22 ismethod() Function . . . 260
33.23 isprivatemethod() Function . . . 261
33.24 addattribute() Function . . . 261
33.25 addmethod() Function . . . 262
33.26 getattribute() function . . . 264
33.27 setattribute() function . . . 265
33.28 mergemethods() Function . . . 266
33.29 packagename() Function . . . 267
33.30 importpackage() Function . . . 267
33.31 Nothing() function . . . 268
33.32 OptionalFunc() function . . . 268

34 Declarative Programming using Nested Structures **270**

34.1 Creating Objects inside Lists . . . 270
34.2 Composition and Returning Objects and Lists by Reference . . . 271
34.3 Executing code after the end of object access . . . 273
34.4 Declarative Programming on the top of Object-Oriented . . . 274
34.5 More Beautiful Code . . . 275

35 Natural Language Programming 276
35.1 History . . . 276
35.2 Example . . . 276
35.3 Change the Ring Keyword 'And' . . . 277
35.4 Change the Ring Operator '+' . . . 279
35.5 Change the '=' operator to 'is' . . . 279
35.6 Using Eval() with our Natural Code . . . 281
35.7 BraceStart and BraceEnd Methods . . . 282
35.8 BraceExprEval Method . . . 283
35.9 Real Natural Code . . . 283
35.10 BraceError() Method . . . 284
35.11 Clean Natural Code . . . 285

36 Using the Natural Library 287
36.1 Natural Library - Demo Program . . . 287
36.2 Defining Commands . . . 290
36.3 Natural Library - Operators . . . 292
36.4 Defining commands using classes . . . 293

37 Scope Rules for Variables and Attributes 295
37.1 Three Scopes . . . 295
37.2 Defining Variables and Variables Access . . . 295
37.3 How Ring find the variable? . . . 296
37.4 Using Object.Attribute . . . 296
37.5 The Self Object . . . 296
37.6 How Ring Define Variables and Attributes . . . 297
37.7 Conflict between Global Variables and Class Attributes . . . 297
37.8 Conflict between Class Attributes and Local Variables . . . 299
37.9 Using Braces to access objects inside Class Methods . . . 300
37.10 Accessing the class attributes from braces inside class methods . . . 302
37.11 Creating a Class for each Window in GUI applications . . . 304
37.12 Conflict between self inside braces and self in the class region . . . 305
37.13 Using braces to escape from the current object scope . . . 308
37.14 The For Loops uses the local scope . . . 309
37.15 Summary of Scope Rules . . . 310

38 Scope Rules for Functions and Methods 312
38.1 How Ring find the Functions and Methods? . . . 312
38.2 Example about Sharing Names between Functions and Methods . . . 313
38.3 Calling a function sharing the name with a method in the current class . . . 314

39 Syntax Flexibility 316
39.1 Change Language Keywords . . . 316
39.2 Change Language Operators . . . 317
39.3 Load Syntax Files . . . 318
39.4 Using "()" around the function parameters . . . 319
39.5 Using Semi-colon after and between statements . . . 320
39.6 Using $ and @ in the start of the variable name . . . 320
39.7 Using the 'elseif' keyword as 'but' in if statement . . . 321

39.8 Using the 'else' keyword as 'other' in switch statement . . . 321
39.9 Using the 'end' keyword in different control structures . . . 322
39.10 Using braces to start and end different control structures . . . 323
39.11 Using 'put' and 'get' as 'see' and 'give' . . . 324
39.12 Using 'case' as 'on' in switch statements . . . 324
39.13 Using 'def' as 'func' in functions/methods definition . . . 325
39.14 Using braces { } in Packages/Classes/Functions . . . 325
39.15 Using 'break'/'continue' keywords . . . 325
39.16 Using 'end' keyword after Packages/Classes/Functions . . . 326
39.17 Using 'function'/'endfunction' keywords . . . 326
39.18 Using 'endif'/'endfor'/'endwhile'/'endswitch'/'endtry' keywords . . . 326
39.19 Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions . . . 327
39.20 Ignore new lines after keywords . . . 327
39.21 Automatic loading for syntax files . . . 328

40 Introduction to the Type Hints Library 330
40.1 Why Type Hints? . . . 330
40.2 Example . . . 330
40.3 User Types . . . 331
40.4 Using Types inside Code . . . 331
40.5 Using Override . . . 332
40.6 Rules . . . 332

41 The Trace Library and the Interactive Debugger 334
41.1 Loading the Trace library . . . 334
41.2 Trace All Events . . . 334
41.3 Trace control flow between functions . . . 335
41.4 Pass Error . . . 335
41.5 Interactive Debugger . . . 335
41.6 Execute Program Line by Line . . . 336
41.7 BreakPoint . . . 336
41.8 Disable BreakPoints . . . 336
41.9 Using the Interactive Debugger . . . 337

42 Embedding Ring in Ring 339
42.1 Embedding Ring in Ring without sharing the State . . . 339
42.2 Serial Execution of Programs . . . 340
42.3 ring_state_setvar() . . . 340
42.4 ring_state_new() and ring_state_mainfile() . . . 341
42.5 Runtime Errors when Embedding Ring in Ring . . . 342
42.6 ring_state_filetokens() function . . . 343

43 Stdlib Functions 345
43.1 IsAppCompiled() function . . . 345
43.2 AppArguments() function . . . 345
43.3 AppPath() function . . . 346
43.4 JustFilePath() function . . . 346
43.5 JustFileName() function . . . 347
43.6 Value() function . . . 347
43.7 Times() function . . . 347
43.8 Map() function . . . 348
43.9 Filter() function . . . 348
43.10 Reduce() function . . . 348
43.11 Split() function . . . 349
43.12 SplitMany() function . . . 349

43.13 Capitalized() function . . . 349
43.14 IsSpecial() function . . . 350
43.15 IsVowel() function . . . 350
43.16 LineCount() function . . . 350
43.17 Factorial() function . . . 351
43.18 Fibonacci() function . . . 351
43.19 IsPrime() function . . . 351
43.20 Sign() function . . . 352
43.21 List2File() function . . . 352
43.22 File2List() function . . . 352
43.23 StartsWith() function . . . 353
43.24 EndsWith() function . . . 353
43.25 GCD() function . . . 353
43.26 LCM() function . . . 354
43.27 SumList() function . . . 354
43.28 ProdList() function . . . 354
43.29 EvenOrOdd() function . . . 355
43.30 Factors() function . . . 355
43.31 IsPalindrome() function . . . 355
43.32 IsLeapYear() function . . . 356
43.33 BinaryDigits() function . . . 356
43.34 MatrixMulti() function . . . 356
43.35 MatrixTrans() function . . . 357
43.36 DayOfWeek() function . . . 357
43.37 Permutation() function . . . 358
43.38 ReadLine() function . . . 358
43.39 SubString() function . . . 358
43.40 ChangeString() function . . . 359
43.41 Sleep() function . . . 359
43.42 IsMainSourceFile() function . . . 360
43.43 MakeDir() function . . . 360
43.44 Fsize() function . . . 360
43.45 TrimAll() function . . . 360
43.46 TrimLeft() function . . . 361
43.47 TrimRight() function . . . 361
43.48 EpochTime() function . . . 361
43.49 SystemCmd() Function . . . 361
43.50 ListAllFiles() Function . . . 362
43.51 SystemSilent() Function . . . 362
43.52 OSCreateOpenFolder() Function . . . 362
43.53 OSCopyFolder() Function . . . 362
43.54 OSDeleteFolder() Function . . . 363
43.55 OSCopyFile() Function . . . 363
43.56 OSDeleteFile() Function . . . 363
43.57 OSRenameFile() Function . . . 363
43.58 List2Code() Function . . . 363
43.59 Str2ASCIIList() . . . 364
43.60 ASCIIList2Str() . . . 364
43.61 IsListContainsItems() . . . 364
43.62 IsBetween() . . . 365
43.63 TimeInfo() . . . 365
43.64 RandomList() . . . 366
43.65 RandomItem() . . . 366
43.66 CheckEquality() . . . 366

44 Stdlib Classes **367**
44.1 StdBase Class . . . 368
44.2 String Class . . . 368
44.3 List Class . . . 370
44.4 Stack Class . . . 372
44.5 Queue Class . . . 373
44.6 HashTable Class . . . 374
44.7 Tree Class . . . 375
44.8 Math Class . . . 377
44.9 DateTime Class . . . 380
44.10 File Class . . . 382
44.11 System Class . . . 383
44.12 Debug Class . . . 383
44.13 DataType Class . . . 384
44.14 Conversion Class . . . 386
44.15 ODBC Class . . . 387
44.16 MySQL Class . . . 388
44.17 SQLite Class . . . 389
44.18 PostgreSQL Class . . . 390
44.19 Security Class . . . 392
44.20 Internet Class . . . 393

45 Desktop, WebAssembly and Mobile Development using RingQt **394**
45.1 The First GUI Application . . . 394
45.2 The Events Loop . . . 396
45.3 Using Layout . . . 397
45.4 Using the QTextEdit Class . . . 398
45.5 Using the QListWidget Class . . . 399
45.6 Using QTreeView and QFileSystemModel . . . 402
45.7 Using QTreeWidget and QTreeWidgetItem . . . 404
45.8 Using QComboBox Class . . . 405
45.9 Creating Menubar . . . 406
45.10 Context Menu . . . 409
45.11 Creating Toolbar . . . 410
45.12 Creating StatusBar . . . 411
45.13 Using QDockWidget . . . 412
45.14 Using QTabWidget . . . 414
45.15 Using QTableWidget . . . 415
45.16 Using QProgressBar . . . 416
45.17 Using QSpinBox . . . 417
45.18 Using QSlider . . . 418
45.19 Using QDateEdit . . . 419
45.20 Using QDial . . . 420
45.21 Using QWebView . . . 423
45.22 Using QCheckBox . . . 424
45.23 Using QRadioButton and QButtonGroup . . . 426
45.24 Adding Hyperlink to QLabel . . . 427
45.25 QVideoWidget and QMediaPlayer . . . 428
45.26 Using QFrame . . . 430
45.27 Display Image using QLabel . . . 431
45.28 Menubar and StyleSheet Example . . . 433
45.29 QLineEdit Events and QMessageBox . . . 434
45.30 Other Widgets Events . . . 437
45.31 Using the QTimer Class . . . 439

45.32 Using QProgressBar and Timer . . . 440
45.33 Display Scaled Image using QLabel . . . 441
45.34 Using the QFileDialog Class . . . 443
45.35 Drawing using QPainter . . . 444
45.36 Printing using QPrinter . . . 445
45.37 Using QPrintPreviewDialog . . . 446
45.38 Creating More than one Window . . . 449
45.39 Playing Sound . . . 450
45.40 Using the QColorDialog Class . . . 450
45.41 Using qLCDNumber Class . . . 452
45.42 Movable Label Example . . . 453
45.43 QMessagebox Example . . . 454
45.44 Using QInputDialog Class . . . 455
45.45 Dialog Functions . . . 456
45.46 KeyPress and Mouse Move Events . . . 457
45.47 Moving Objects using the Mouse . . . 458
45.48 Inheritance from GUI Classes . . . 462
45.49 Using QDesktopWidget Class . . . 463
45.50 Rotate Text . . . 464
45.51 Change Focus . . . 466
45.52 Regular Expressions . . . 467
45.53 Simple Client and Server Example . . . 468
45.54 Dynamic Objects . . . 471
45.55 Weight History Application . . . 472
45.56 Notepad Application . . . 476
45.57 The Cards Game . . . 493
45.58 Classes and their Methods to use the default events . . . 499
45.59 Methods to use Events with Events Filter . . . 502
45.60 The Difference between Qt and RingQt . . . 503
45.61 RingQt Classes and their Qt Documentation . . . 504
45.62 New Classes names - Index Start from 1 . . . 504
45.63 TableWidget - AddList() Method . . . 505
45.64 Creating Reports using the WebLib and the GUILib . . . 506

46 Using the Form Designer **510**
46.1 The Designer Windows . . . 511
46.2 The Toolbox . . . 511
46.3 The Properties . . . 511
46.4 Running Forms . . . 512
46.5 Events Code . . . 513
46.6 Keyboard Shortcuts . . . 515
46.7 Menubar Designer . . . 515
46.8 Window Flags . . . 516
46.9 Entering Items . . . 517
46.10 Using Layouts . . . 517
46.11 Objects Order and Layouts . . . 517
46.12 More Samples and Tests . . . 520

47 Graphics Programming using RingQt3D **521**
47.1 Drawing Cube . . . 521
47.2 Drawing Torus . . . 523
47.3 Drawing Sphere . . . 525
47.4 Drawing Cylinder . . . 527
47.5 Drawing Cone . . . 529

47.6 Drawing Plane . . . 531
47.7 Texture . . . 533
47.8 Key Press . . . 535
47.9 Object Picker . . . 537
47.10 Frame Action . . . 540
47.11 Text 2D . . . 543
47.12 Extruded Text . . . 545
47.13 Model . . . 548
47.14 Model Texture . . . 549
47.15 Draw Office . . . 551
47.16 Many Objects . . . 553
47.17 Camera . . . 556
47.18 Scence . . . 559

48 Objects Library for RingQt Application 564
48.1 Library Usage with GUI Applications . . . 564
48.2 Example . . . 565
48.3 openWindowAndLink() Function . . . 567
48.4 openWindowInPackages() Function . . . 568
48.5 Using ObjectID() and GetWindowByID() . . . 569
48.6 Using ObjectID() and OpenWindowAndLink() . . . 571
48.7 Objects Library Source Code . . . 573

49 Multi-language Applications 574
49.1 Using String2Constant . . . 574
49.2 Form Designer Translation . . . 575
49.3 Forms Translation . . . 576

50 Building RingQt Applications for Mobile 578
50.1 Download Requirements . . . 578
50.2 Install Qt for Android . . . 578
50.3 Using Ring2EXE . . . 579
50.4 The Qt project for your Ring application . . . 580
50.5 Comments about developing for Android using RingQt . . . 583

51 Building RingQt Applications for WebAssembly 586
51.1 Download Requirements . . . 586
51.2 Using Ring2EXE . . . 587
51.3 The Qt project for your Ring application . . . 588
51.4 Comments about developing for WebAssembly using RingQt . . . 591
51.5 Dialogs . . . 593
51.6 Online Applications . . . 593

52 Web Development (CGI Library) 594
52.1 Configure the Apache web server . . . 594
52.2 Setting the shebang line . . . 595
52.3 Grant Ring files execution permission . . . 596
52.4 Ring CGI Hello World Program . . . 596
52.5 Hello World Program using the Web Library . . . 596
52.6 Web Library Features . . . 597
52.7 HTTP Get Example . . . 597
52.8 HTTP POST Example . . . 602
52.9 Upload Files . . . 604
52.10 Cookies . . . 608
52.11 URL Encode . . . 609

52.12 Templates . . . 610
52.13 HTML Special Characters . . . 613
52.14 Hash Functions . . . 613
52.15 Random Image . . . 615
52.16 HTML Lists . . . 616
52.17 HTML Tables . . . 619
52.18 Gradient . . . 620
52.19 Generating Pages using Objects . . . 621
52.20 HtmlPage Class . . . 625
52.21 Using Bootstrap Library using Functions . . . 627
52.22 Using Bootstrap Library using Objects . . . 628
52.23 CRUD Example using MVC . . . 631
52.24 Users registration and Login . . . 634
52.25 Database, ModelBase & ControllerBase classes . . . 640
52.26 WebLib API . . . 646
52.27 Application Class . . . 647
52.28 Page Class . . . 648
52.29 ScriptFunctions Class . . . 650
52.30 StyleFunctions Class . . . 651
52.31 WebPage Class . . . 651
52.32 HtmlPage Class . . . 652

53 Deploying Web Applications in the Cloud 653
53.1 Introduction . . . 653
53.2 Usage . . . 654
53.3 Ring source code files and permissions . . . 654
53.4 Hello World program . . . 655
53.5 Application Database . . . 655
53.6 Deploying after updates . . . 655
53.7 Local Tests . . . 656

54 Graphics and 2D Games programming using RingAllegro 657
54.1 Drawing, Animation and Input . . . 657
54.2 Using TrueType Fonts . . . 664
54.3 Playing Sound Files . . . 665
54.4 Scaling and Rotating Images . . . 666
54.5 Display Transparent Image . . . 667
54.6 Using Threads . . . 668

55 Demo Project - Game Engine for 2D Games 672
55.1 Project Layers . . . 672
55.2 Graphics Library bindings . . . 672
55.3 Interface to graphics library . . . 673
55.4 Game Engine Classes . . . 673
55.5 Games Layer . . . 674
55.6 Game Class . . . 674
55.7 GameObject Class . . . 675
55.8 Sprite Class . . . 675
55.9 Text Class . . . 676
55.10 Animate Class . . . 676
55.11 Sound Class . . . 677
55.12 Map Class . . . 677
55.13 Using the Game Engine - Creating the Game Window . . . 678
55.14 Using the Game Engine - Drawing Text . . . 679

55.15 Using the Game Engine - Moving Text . . . 680
55.16 Using the Game Engine - Playing Sound . . . 681
55.17 Using the Game Engine - Animation . . . 682
55.18 Using the Game Engine - Animation and Functions . . . 683
55.19 Using the Game Engine - Sprite - Automatic Movement using Keyboard . . . 685
55.20 Using the Game Engine - Sprite - Keypress event . . . 686
55.21 Using the Game Engine - Sprite - Mouse event . . . 687
55.22 Using the Game Engine - Sprite - State event . . . 688
55.23 Using the Game Engine - Animate - Events . . . 689
55.24 Using the Game Engine - Map . . . 691
55.25 Using the Game Engine - Map Events . . . 693
55.26 Using the Game Engine - Object and Drawing . . . 694
55.27 Stars Fighter Game . . . 698
55.28 Flappy Bird 3000 Game . . . 707
55.29 Super Man 2016 Game . . . 714

56 Building Games For Android **727**
56.1 Download Requirements and Update the Android SDK . . . 727
56.2 Project Folder . . . 728
56.3 Building the project . . . 729

57 Developing Games using RingRayLib **730**
57.1 Introduction . . . 730
57.2 Basic Window . . . 730
57.3 Input Keys . . . 731
57.4 Input Mouse . . . 732
57.5 3D Camera . . . 733
57.6 3D Camera Free . . . 735
57.7 Mouse Wheel . . . 737
57.8 Input Multi-touch . . . 738
57.9 Camera First Person . . . 740
57.10 3D Picking . . . 742
57.11 Full Screen . . . 744
57.12 Two Cubes . . . 745
57.13 Basic Shapes . . . 748
57.14 Draw Ring . . . 750
57.15 Bezier Lines . . . 752
57.16 Collision Area . . . 753
57.17 Following Eyes . . . 756
57.18 Colors Palette . . . 758
57.19 Rectangle Scaling . . . 760
57.20 Music Playing . . . 762
57.21 Sound Loading . . . 764
57.22 Image Drawing . . . 766
57.23 Image Generation . . . 767
57.24 Texture Source . . . 769
57.25 Geometric Shapes . . . 771
57.26 Cubic Map . . . 774
57.27 Functions . . . 775
57.28 More Functions . . . 785

58 Using RingOpenGL and RingFreeGLUT for 3D Graphics **787**
58.1 Samples Source (Authors) . . . 787
58.2 What is RingOpenGL? . . . 787

58.3 What is RingFreeGLUT? . . . 788
58.4 The First Window using RingFreeGLUT . . . 788
58.5 Drawing using RingOpenGL . . . 789
58.6 The First Triangle . . . 791
58.7 Window Resize Event . . . 792
58.8 Triangle Rotation . . . 794
58.9 Keyboard Events and Colors . . . 796
58.10 The Camera . . . 798
58.11 Mouse Events . . . 806
58.12 Menu Events . . . 810
58.13 Using Fonts . . . 818
58.14 Frames Per Second . . . 826
58.15 Make a Cube using RingOpenGL and RingFreeGLUT . . . 837

59 Using RingOpenGL and RingAllegro for 3D Graphics 841
59.1 3D Cube and Texture . . . 841
59.2 Many Cubes . . . 846
59.3 TicTacToe 3D Game . . . 852
59.4 More 3D Samples . . . 863

60 The Gold Magic 800 Game 866
60.1 The Game Story . . . 866
60.2 How to play? . . . 866
60.3 What will you learn? . . . 866
60.4 Screen Shots . . . 867
60.5 Source Code . . . 868

61 RingTilengine Extension 870
61.1 Getting Started . . . 870
61.2 More Samples . . . 871
61.3 Using LibSDL and Tilengine . . . 892
61.4 Reference . . . 897

62 Performance Tips 905
62.1 Introduction . . . 905
62.2 Creating Lists . . . 907
62.3 Arithmetic Operations . . . 908
62.4 Using len() and For Loops . . . 909
62.5 Calling Functions and Methods . . . 909

63 Command Line Options 911
63.1 Printing Tokens . . . 911
63.2 Printing Rules . . . 913
63.3 Printing Intermediate Code . . . 918
63.4 Printing Final Intermediate Code . . . 921
63.5 CGI Support . . . 925
63.6 No Run . . . 926
63.7 Printing Instruction Operation Code . . . 926
63.8 Performance . . . 926
63.9 Generate Object File . . . 927
63.10 Generate Embedded Object File . . . 927

64 Distributing Ring Applications (Manual) 928
64.1 Distributing Applications for Microsoft Windows . . . 928
64.2 Protecting the Source Code . . . 929

64.3 The files ring.ring and ring.ringo . . . 929
64.4 Creating Windows Installer . . . 929
64.5 Using C/C++ Compiler and Linker . . . 929
64.6 Distributing Applications and Games for Mobile . . . 930

65 Distributing Ring Applications using Ring2EXE 931
65.1 Using Ring2EXE . . . 931
65.2 How Ring2EXE works? . . . 931
65.3 Example . . . 932
65.4 Options . . . 932
65.5 Building standalone console application . . . 933
65.6 Distributing RingAllegro Applications . . . 933
65.7 Distributing RingQt Applications . . . 934
65.8 Distributing Applications for Mobile using RingQt . . . 935
65.9 Distributing Applications for WebAssembly using RingQt . . . 935
65.10 Building the Cards Game for Mobile using RingQt . . . 935
65.11 Building the Weight History Application for Mobile using RingQt . . . 936
65.12 Building the Form Designer for Mobile using RingQt . . . 937
65.13 Creating the Qt resource file using Folder2qrc . . . 939
65.14 Important Information about Ring2EXE . . . 939

66 The Ring Package Manager (RingPM) 941
66.1 Features . . . 941
66.2 Discovering Packages . . . 941
66.3 Updating the RingPM Registry . . . 942
66.4 Installing Packages . . . 942
66.5 Printing List of Installed Packages . . . 943
66.6 Run Package . . . 943
66.7 Update Package . . . 944
66.8 Remove Package . . . 944
66.9 Deleting All Packages . . . 944
66.10 Creating New Package . . . 944
66.11 The Package Description File . . . 947
66.12 Create Package in the Current Folder . . . 948
66.13 The RingPM Registry File . . . 950

67 Using ZeroLib 951
67.1 Introduction . . . 951
67.2 Z() function . . . 951
67.3 ZeroBasedList Class . . . 951
67.4 ZeroBasedString Class . . . 952
67.5 Source Code . . . 954

68 FoxRing Functions Reference 955
68.1 FoxRing functions . . . 955
68.2 frAbs() function . . . 956
68.3 frAsc() function . . . 956
68.4 frAddBs() function . . . 957
68.5 frAt() function . . . 957
68.6 frAtC() function . . . 958
68.7 frChr() function . . . 958
68.8 frEmpty() function . . . 959
68.9 frFile() function . . . 959
68.10 frFileToStr() function . . . 959
68.11 frStr() function . . . 960

68.12 frSetIfEmpty() function . . . 961
68.13 frSpace() function . . . 961
68.14 frInList() function . . . 961
68.15 frForcePath() function . . . 962
68.16 frAllTrim() function . . . 962
68.17 frLTrim() function . . . 962
68.18 frJustDrive() function . . . 962
68.19 frJustExt() function . . . 962
68.20 frJustStem() function . . . 963
68.21 frRTrim() function . . . 963
68.22 frJustPath() function . . . 963
68.23 frForceExt() function . . . 963
68.24 frALines() function . . . 963
68.25 frJustFName() function . . . 963
68.26 frPadL() function . . . 964
68.27 frPadR() function . . . 964
68.28 frProper() function . . . 964
68.29 frReplicate() function . . . 964
68.30 frLen() function . . . 964
68.31 frStuff() function . . . 964
68.32 frSubStr() function . . . 965
68.33 frStrTran() function . . . 965
68.34 frListToString() function . . . 965
68.35 frInt() function . . . 965
68.36 frStringToList() function . . . 965
68.37 frIIf() function . . . 966
68.38 frVal() function . . . 966
68.39 frBetween() function . . . 966
68.40 frSetSeparatorTo() function . . . 967
68.41 frTransform() function . . . 967
68.42 frVarType() function . . . 968
68.43 Example . . . 968

69 BigNumber Library **971**
69.1 Loading the library . . . 971
69.2 Examples . . . 971
69.3 BigNumber Functions . . . 973
69.4 BigNumber Class . . . 973
69.5 Library Source Code . . . 973

70 Using CSVLib **974**
70.1 Introduction . . . 974
70.2 Functions . . . 974
70.3 Examples . . . 974

71 Using JSONLib **977**
71.1 Introduction . . . 977
71.2 Functions . . . 977
71.3 Examples . . . 977

72 Using HTTPLib **979**
72.1 Introduction . . . 979
72.2 Server Class Methods . . . 979
72.3 Example . . . 979
72.4 Samples . . . 980

72.5 Printing Constants . . . 980
72.6 Using HTTP GET . . . 980
72.7 Using WebLib . . . 982
72.8 Using HTTP Post . . . 983
72.9 More Samples . . . 986
72.10 Using Cookies . . . 989
72.11 Uploading Files . . . 990
72.12 Using Templates . . . 992
72.13 Regular Expressions . . . 993
72.14 Stop the Server . . . 993

73 Using TokensLib **995**
73.1 Introduction . . . 995
73.2 RingTokens Class . . . 995
73.3 Example . . . 995
73.4 Constants . . . 996

74 Using RingLibCurl **999**
74.1 Get Request . . . 999
74.2 Post Request . . . 999
74.3 Facebook Login . . . 1000
74.4 Save Output to String . . . 1001
74.5 Get Stock Data From Yahoo . . . 1001

75 RingLibCurl Functions Reference **1003**
75.1 Introduction . . . 1003
75.2 Reference . . . 1003

76 RingSockets Extension **1012**
76.1 TCP Server . . . 1012
76.2 TCP Client . . . 1012
76.3 Functions . . . 1013
76.4 Constants . . . 1013

77 Threads Extension **1014**
77.1 Creating Threads . . . 1014
77.2 More Examples . . . 1014
77.3 Reference . . . 1016

78 RingLibUI Extension **1018**
78.1 Hello World . . . 1018
78.2 Say Hello . . . 1020
78.3 Control Gallery . . . 1021
78.4 Say Something . . . 1029
78.5 Using the Menubar . . . 1030
78.6 Drawing Sample . . . 1034
78.7 Draw Gradient . . . 1036
78.8 Histogram . . . 1037
78.9 Text Drawing . . . 1043
78.10 More Drawing Samples . . . 1048

79 Using RingZip **1063**
79.1 Create Zip File . . . 1063
79.2 Extract Zip File . . . 1063
79.3 Print Files in Zip file . . . 1063

79.4 Using RingZip Classes . . . 1064
79.5 Zip Class Reference . . . 1065
79.6 ZipEntry Class Reference . . . 1065

80 RingLibZip Functions Reference 1066
80.1 Introduction . . . 1066
80.2 Reference . . . 1066

81 RingMurmurHash Functions Reference 1067
81.1 MurmurHash1 functions . . . 1067
81.2 MurmurHash2 functions . . . 1067
81.3 MurmurHash3 functions . . . 1067
81.4 Example . . . 1068

82 RingConsoleColors Functions Reference 1069
82.1 Introduction . . . 1069
82.2 Reference . . . 1069

83 RingAllegro Functions Reference 1071
83.1 Introduction . . . 1071
83.2 Reference . . . 1071

84 Using RingLibSDL 1091
84.1 Create Window . . . 1091
84.2 Display Image . . . 1091
84.3 Switch between two images . . . 1092
84.4 Draw Rectangle . . . 1092
84.5 Display PNG Images . . . 1093
84.6 Use TTF Fonts . . . 1093
84.7 Display Transparent Images . . . 1094
84.8 Close Window Event . . . 1095
84.9 Mouse Events . . . 1096
84.10 Play Sound . . . 1097

85 RingLibSDL Functions Reference 1098
85.1 Introduction . . . 1098
85.2 Reference . . . 1098

86 Using RingLibuv 1123
86.1 First Application using RingLibuv . . . 1123
86.2 The Events Loop . . . 1124
86.3 Server Example . . . 1124
86.4 Client Example . . . 1126
86.5 Server Example Using Classes . . . 1127
86.6 Client Example Using Classes . . . 1129
86.7 Threads Example . . . 1130
86.8 Threads Example - Using Classes . . . 1131

87 RingLibuv Functions Reference 1132
87.1 Introduction . . . 1132
87.2 Reference . . . 1132

88 RingFreeGLUT Functions Reference 1141
88.1 Introduction . . . 1141
88.2 Reference . . . 1141

89 RingStbImage Functions Reference **1152**
89.1 Introduction . . . 1152
89.2 Constants . . . 1154
89.3 Functions . . . 1154

90 RingOpenGL (OpenGL 3.2) Functions Reference **1156**
90.1 Introduction . . . 1156
90.2 Reference . . . 1156

91 RingQt Classes Reference **1203**
91.1 AbstractAxis Class . . . 1203
91.2 AbstractBarSeries Class . . . 1206
91.3 CodeEditor Class . . . 1207
91.4 QAbstractAspect Class . . . 1208
91.5 QAbstractButton Class . . . 1208
91.6 QAbstractCameraController Class . . . 1209
91.7 QAbstractItemView Class . . . 1209
91.8 QAbstractPrintDialog Class . . . 1211
91.9 QAbstractScrollArea Class . . . 1211
91.10 QAbstractSeries Class . . . 1212
91.11 QAbstractSlider Class . . . 1213
91.12 QAbstractSocket Class . . . 1214
91.13 QAbstractSpinBox Class . . . 1215
91.14 QAction Class . . . 1216
91.15 QAllEvents Class . . . 1218
91.16 QApp Class . . . 1222
91.17 QAreaLegendMarker Class . . . 1222
91.18 QAreaSeries Class . . . 1223
91.19 QAspectEngine Class . . . 1224
91.20 QAxBase Class . . . 1225
91.21 QAxObject Class . . . 1226
91.22 QAxWidget Class . . . 1226
91.23 QAxWidget2 Class . . . 1226
91.24 QBarCategoryAxis Class . . . 1226
91.25 QBarLegendMarker Class . . . 1227
91.26 QBarSeries Class . . . 1227
91.27 QBarSet Class . . . 1227
91.28 QBitmap Class . . . 1229
91.29 QBluetoothAddress Class . . . 1229
91.30 QBluetoothDeviceDiscoveryAgent Class . . . 1230
91.31 QBluetoothDeviceInfo Class . . . 1230
91.32 QBluetoothHostInfo Class . . . 1231
91.33 QBluetoothLocalDevice Class . . . 1231
91.34 QBluetoothServer Class . . . 1232
91.35 QBluetoothServiceDiscoveryAgent Class . . . 1232
91.36 QBluetoothServiceInfo Class . . . 1233
91.37 QBluetoothSocket Class . . . 1234
91.38 QBluetoothTransferManager Class . . . 1235
91.39 QBluetoothTransferReply Class . . . 1235
91.40 QBluetoothTransferRequest Class . . . 1236
91.41 QBluetoothUuid Class . . . 1236
91.42 QBoxLayout Class . . . 1236
91.43 QBoxPlotLegendMarker Class . . . 1237
91.44 QBoxPlotSeries Class . . . 1237

91.45 QBoxSet Class . . . 1238
91.46 QBrush Class . . . 1239
91.47 QBuffer Class . . . 1240
91.48 QButtonGroup Class . . . 1240
91.49 QByteArray Class . . . 1241
91.50 QCalendarWidget Class . . . 1243
91.51 QCamera Class . . . 1244
91.52 QCameraImageCapture Class . . . 1245
91.53 QCameraLens Class . . . 1246
91.54 QCameraSelector Class . . . 1247
91.55 QCameraViewfinder Class . . . 1247
91.56 QCandlestickLegendMarker Class . . . 1247
91.57 QCandlestickModelMapper Class . . . 1247
91.58 QCandlestickSeries Class . . . 1248
91.59 QCandlestickSet Class . . . 1250
91.60 QCategoryAxis Class . . . 1251
91.61 QChar Class . . . 1251
91.62 QChart Class . . . 1254
91.63 QChartView Class . . . 1255
91.64 QCheckBox Class . . . 1256
91.65 QChildEvent Class . . . 1256
91.66 QClipboard Class . . . 1257
91.67 QColor Class . . . 1257
91.68 QColorDialog Class . . . 1260
91.69 QComboBox Class . . . 1260
91.70 QCompleter Class . . . 1262
91.71 QCompleter2 Class . . . 1263
91.72 QCompleter3 Class . . . 1263
91.73 QCompleter4 Class . . . 1263
91.74 QConeGeometry Class . . . 1264
91.75 QConeMesh Class . . . 1264
91.76 QCoreApplication Class . . . 1265
91.77 QCuboidMesh Class . . . 1266
91.78 QCullFace Class . . . 1267
91.79 QCursor Class . . . 1267
91.80 QCylinderMesh Class . . . 1267
91.81 QDate Class . . . 1268
91.82 QDateEdit Class . . . 1268
91.83 QDateTime Class . . . 1269
91.84 QDateTimeAxis Class . . . 1270
91.85 QDateTimeEdit Class . . . 1270
91.86 QDepthTest Class . . . 1272
91.87 QDesktopServices Class . . . 1272
91.88 QDesktopWidget Class . . . 1272
91.89 QDial Class . . . 1272
91.90 QDialog Class . . . 1273
91.91 QDiffuseSpecularMaterial Class . . . 1274
91.92 QDir Class . . . 1274
91.93 QDockWidget Class . . . 1276
91.94 QDrag Class . . . 1277
91.95 QDragEnterEvent Class . . . 1278
91.96 QDragLeaveEvent Class . . . 1278
91.97 QDragMoveEvent Class . . . 1278
91.98 QDropEvent Class . . . 1278

91.99 QEffect Class . . . 1279
91.100 QEntity Class . . . 1279
91.101 QEvent Class . . . 1279
91.102 QExtrudedTextMesh Class . . . 1280
91.103 QFile Class . . . 1280
91.104 QFile2 Class . . . 1281
91.105 QFileDevice Class . . . 1281
91.106 QFileDialog Class . . . 1281
91.107 QFileInfo Class . . . 1284
91.108 QFileSystemModel Class . . . 1285
91.109 QFirstPersonCameraController Class . . . 1286
91.110 QFont Class . . . 1287
91.111 QFontDialog Class . . . 1288
91.112 QFontMetrics Class . . . 1289
91.113 QForwardRenderer Class . . . 1290
91.114 QFrame Class . . . 1290
91.115 QFrame2 Class . . . 1291
91.116 QFrame3 Class . . . 1291
91.117 QFrameAction Class . . . 1291
91.118 QGeoAddress Class . . . 1291
91.119 QGeoAreaMonitorInfo Class . . . 1292
91.120 QGeoAreaMonitorSource Class . . . 1293
91.121 QGeoCircle Class . . . 1293
91.122 QGeoCoordinate Class . . . 1293
91.123 QGeoPositionInfo Class . . . 1294
91.124 QGeoPositionInfoSource Class . . . 1294
91.125 QGeoRectangle Class . . . 1295
91.126 QGeoSatelliteInfo Class . . . 1295
91.127 QGeoSatelliteInfoSource Class . . . 1296
91.128 QGeoShape Class . . . 1296
91.129 QGoochMaterial Class . . . 1296
91.130 QGradient Class . . . 1297
91.131 QGraphicsScene Class . . . 1297
91.132 QGraphicsVideoItem Class . . . 1300
91.133 QGraphicsView Class . . . 1300
91.134 QGridLayout Class . . . 1302
91.135 QGuiApplication Class . . . 1303
91.136 QHBarModelMapper Class . . . 1305
91.137 QHBoxLayout Class . . . 1306
91.138 QHBoxPlotModelMapper Class . . . 1306
91.139 QHCandlestickModelMapper Class . . . 1307
91.140 QHPieModelMapper Class . . . 1308
91.141 QHXYModelMapper Class . . . 1309
91.142 QHeaderView Class . . . 1310
91.143 QHorizontalBarSeries Class . . . 1312
91.144 QHorizontalPercentBarSeries Class . . . 1312
91.145 QHorizontalStackedBarSeries Class . . . 1312
91.146 QHostAddress Class . . . 1313
91.147 QHostInfo Class . . . 1313
91.148 QIODevice Class . . . 1314
91.149 QIcon Class . . . 1315
91.150 QImage Class . . . 1315
91.151 QInputAspect Class . . . 1316
91.152 QInputDialog Class . . . 1317

91.153QJsonArray Class . . . 1318
91.154QJsonDocument Class . . . 1319
91.155QJsonObject Class . . . 1319
91.156QJsonParseError Class . . . 1320
91.157QJsonValue Class . . . 1320
91.158QKeySequence Class . . . 1320
91.159QLCDNumber Class . . . 1321
91.160QLabel Class . . . 1321
91.161QLayout Class . . . 1322
91.162QLegend Class . . . 1323
91.163QLegendMarker Class . . . 1324
91.164QLineEdit Class . . . 1325
91.165QLineSeries Class . . . 1327
91.166QLinearGradient Class . . . 1327
91.167QListView Class . . . 1328
91.168QListWidget Class . . . 1329
91.169QListWidgetItem Class . . . 1330
91.170QLocale Class . . . 1331
91.171QLogValueAxis Class . . . 1333
91.172QLogicAspect Class . . . 1334
91.173QMainWindow Class . . . 1334
91.174QMaterial Class . . . 1335
91.175QMatrix4x4 Class . . . 1336
91.176QMdiArea Class . . . 1337
91.177QMdiSubWindow Class . . . 1338
91.178QMediaObject Class . . . 1338
91.179QMediaPlayer Class . . . 1339
91.180QMediaPlaylist Class . . . 1340
91.181QMenu Class . . . 1340
91.182QMenuBar Class . . . 1341
91.183QMesh Class . . . 1342
91.184QMessageBox Class . . . 1342
91.185QMetalRoughMaterial Class . . . 1343
91.186QMimeData Class . . . 1343
91.187QMorphPhongMaterial Class . . . 1344
91.188QMovie Class . . . 1345
91.189QMutex Class . . . 1346
91.190QMutexLocker Class . . . 1346
91.191QNetworkAccessManager Class . . . 1346
91.192QNetworkProxy Class . . . 1347
91.193QNetworkReply Class . . . 1348
91.194QNetworkRequest Class . . . 1348
91.195QNmeaPositionInfoSource Class . . . 1349
91.196QNode Class . . . 1349
91.197QObject Class . . . 1350
91.198QObjectPicker Class . . . 1351
91.199QOpenGLBuffer Class . . . 1352
91.200QOpenGLContext Class . . . 1352
91.201QOpenGLDebugLogger Class . . . 1353
91.202QOpenGLFramebufferObject Class . . . 1354
91.203QOpenGLFunctions Class . . . 1354
91.204QOpenGLFunctions_3_2_Core Class . . . 1359
91.205QOpenGLPaintDevice Class . . . 1366
91.206QOpenGLShader Class . . . 1367

91.207QOpenGLShaderProgram Class . . . 1367
91.208QOpenGLTexture Class . . . 1371
91.209QOpenGLTimerQuery Class . . . 1373
91.210QOpenGLVersionProfile Class . . . 1374
91.211QOpenGLVertexArrayObject Class . . . 1374
91.212QOpenGLWidget Class . . . 1374
91.213QOrbitCameraController Class . . . 1375
91.214QPageSetupDialog Class . . . 1375
91.215QPaintDevice Class . . . 1375
91.216QPainter Class . . . 1376
91.217QPainter2 Class . . . 1378
91.218QPainterPath Class . . . 1379
91.219QPen Class . . . 1380
91.220QPerVertexColorMaterial Class . . . 1381
91.221QPercentBarSeries Class . . . 1381
91.222QPhongMaterial Class . . . 1381
91.223QPicture Class . . . 1381
91.224QPieLegendMarker Class . . . 1382
91.225QPieSeries Class . . . 1382
91.226QPieSlice Class . . . 1383
91.227QPixmap Class . . . 1385
91.228QPixmap2 Class . . . 1387
91.229QPlainTextEdit Class . . . 1387
91.230QPlaneMesh Class . . . 1389
91.231QPoint Class . . . 1390
91.232QPointF Class . . . 1390
91.233QPointLight Class . . . 1390
91.234QPolarChart Class . . . 1391
91.235QPrintDialog Class . . . 1391
91.236QPrintPreviewDialog Class . . . 1391
91.237QPrintPreviewWidget Class . . . 1392
91.238QPrinter Class . . . 1393
91.239QPrinterInfo Class . . . 1394
91.240QProcess Class . . . 1394
91.241QProgressBar Class . . . 1396
91.242QPushButton Class . . . 1397
91.243QQmlEngine Class . . . 1397
91.244QQmlError Class . . . 1398
91.245QQuaternion Class . . . 1398
91.246QQuickView Class . . . 1399
91.247QQuickWidget Class . . . 1400
91.248QRadioButton Class . . . 1400
91.249QRect Class . . . 1401
91.250QRegion Class . . . 1402
91.251QRegularExpression Class . . . 1403
91.252QRegularExpressionMatch Class . . . 1404
91.253QRegularExpressionMatchIterator Class . . . 1404
91.254QRenderAspect Class . . . 1405
91.255QRenderPass Class . . . 1405
91.256QScatterSeries Class . . . 1405
91.257QSceneLoader Class . . . 1406
91.258QScreen Class . . . 1406
91.259QScrollArea Class . . . 1407
91.260QScrollBar Class . . . 1407

91.261QSerialPort Class . . . 1408
91.262QSerialPortInfo Class . . . 1409
91.263QSize Class . . . 1409
91.264QSkyboxEntity Class . . . 1410
91.265QSlider Class . . . 1410
91.266QSphereMesh Class . . . 1411
91.267QSpinBox Class . . . 1411
91.268QSplashScreen Class . . . 1412
91.269QSplineSeries Class . . . 1412
91.270QSplitter Class . . . 1412
91.271QSqlDatabase Class . . . 1413
91.272QSqlDriver Class . . . 1414
91.273QSqlDriverCreatorBase Class . . . 1414
91.274QSqlError Class . . . 1415
91.275QSqlField Class . . . 1415
91.276QSqlIndex Class . . . 1416
91.277QSqlQuery Class . . . 1416
91.278QSqlRecord Class . . . 1417
91.279QStackedBarSeries Class . . . 1418
91.280QStackedWidget Class . . . 1418
91.281QStandardPaths Class . . . 1419
91.282QStatusBar Class . . . 1419
91.283QString2 Class . . . 1419
91.284QStringList Class . . . 1421
91.285QStringRef Class . . . 1422
91.286QStyle Class . . . 1424
91.287QSurfaceFormat Class . . . 1425
91.288QSystemTrayIcon Class . . . 1426
91.289QTabBar Class . . . 1426
91.290QTabWidget Class . . . 1428
91.291QTableView Class . . . 1429
91.292QTableWidget Class . . . 1431
91.293QTableWidgetItem Class . . . 1433
91.294QTcpServer Class . . . 1434
91.295QTcpSocket Class . . . 1435
91.296QTechnique Class . . . 1436
91.297QTest Class . . . 1436
91.298QText2DEntity Class . . . 1436
91.299QTextBlock Class . . . 1437
91.300QTextBrowser Class . . . 1438
91.301QTextCharFormat Class . . . 1439
91.302QTextCodec Class . . . 1440
91.303QTextCursor Class . . . 1440
91.304QTextDocument Class . . . 1442
91.305QTextEdit Class . . . 1444
91.306QTextOption Class . . . 1447
91.307QTextStream Class . . . 1447
91.308QTextStream2 Class . . . 1449
91.309QTextStream3 Class . . . 1449
91.310QTextStream4 Class . . . 1449
91.311QTextStream5 Class . . . 1449
91.312QTextToSpeech Class . . . 1449
91.313QTextureLoader Class . . . 1450
91.314QTextureMaterial Class . . . 1450

91.315QThread Class . . . 1451
91.316QThreadPool Class . . . 1452
91.317QTime Class . . . 1452
91.318QTimer Class . . . 1453
91.319QToolBar Class . . . 1453
91.320QToolButton Class . . . 1454
91.321QTorusMesh Class . . . 1455
91.322QTransform Class . . . 1455
91.323QTreeView Class . . . 1456
91.324QTreeWidget Class . . . 1458
91.325QTreeWidgetItem Class . . . 1460
91.326QUrl Class . . . 1462
91.327QUuid Class . . . 1463
91.328QVBarModelMapper Class . . . 1463
91.329QVBoxLayout Class . . . 1464
91.330QVBoxPlotModelMapper Class . . . 1464
91.331QVCandlestickModelMapper Class . . . 1465
91.332QVPieModelMapper Class . . . 1466
91.333QVXYModelMapper Class . . . 1467
91.334QValueAxis Class . . . 1468
91.335QVariant Class . . . 1469
91.336QVariant2 Class . . . 1470
91.337QVariant3 Class . . . 1470
91.338QVariant4 Class . . . 1470
91.339QVariant5 Class . . . 1471
91.340QVariantDouble Class . . . 1471
91.341QVariantFloat Class . . . 1471
91.342QVariantInt Class . . . 1471
91.343QVariantString Class . . . 1471
91.344QVector2D Class . . . 1471
91.345QVector3D Class . . . 1472
91.346QVector4D Class . . . 1473
91.347QVectorQVoice Class . . . 1474
91.348QVideoWidget Class . . . 1474
91.349QVideoWidgetControl Class . . . 1475
91.350QViewport Class . . . 1475
91.351QVoice Class . . . 1475
91.352QWebEnginePage Class . . . 1475
91.353QWebEngineView Class . . . 1478
91.354QWebView Class . . . 1479
91.355QWebView Class . . . 1480
91.356QWidget Class . . . 1480
91.357QWindow Class . . . 1486
91.358QXYLegendMarker Class . . . 1489
91.359QXYSeries Class . . . 1490
91.360QXmlStreamAttribute Class . . . 1492
91.361QXmlStreamAttributes Class . . . 1492
91.362QXmlStreamEntityDeclaration Class . . . 1493
91.363QXmlStreamEntityResolver Class . . . 1493
91.364QXmlStreamNamespaceDeclaration Class . . . 1493
91.365QXmlStreamNotationDeclaration Class . . . 1493
91.366QXmlStreamReader Class . . . 1493
91.367QXmlStreamWriter Class . . . 1495
91.368Qt3DCamera Class . . . 1496

91.369 Qt3DWindow Class . . . 1498
91.370 RingCodeHighlighter Class . . . 1498

92 Using FastPro **1499**
92.1 Bytes2List() function . . . 1499
92.2 List2Bytes() function . . . 1499
92.3 UpdateList() function . . . 1499
92.4 UpdateColumn() function . . . 1502
92.5 Generate Image . . . 1503

93 Using References **1506**
93.1 Introduction . . . 1506
93.2 ref()/reference() function . . . 1507
93.3 refcount() function . . . 1508
93.4 Circular References . . . 1508
93.5 The Tree Class . . . 1509
93.6 Linked list . . . 1510
93.7 Dependency injection . . . 1511
93.8 Passing lists to functions . . . 1512
93.9 Ref() and temp. lists . . . 1513
93.10 Nested Ref() . . . 1514
93.11 Ref() in left side . . . 1514
93.12 Ref() and sub lists . . . 1514

94 Low Level Functions **1516**
94.1 callgc() function . . . 1517
94.2 varptr() function . . . 1517
94.3 space() function . . . 1518
94.4 nullpointer() function . . . 1519
94.5 object2pointer() function . . . 1519
94.6 pointer2object() function . . . 1519
94.7 ispointer() function . . . 1521
94.8 ptrcmp() function . . . 1521
94.9 setpointer() function . . . 1522
94.10 getpointer() function . . . 1522
94.11 pointer2string() function . . . 1523
94.12 memcpy() function . . . 1524
94.13 ringvm_cfunctionslist() function . . . 1524
94.14 ringvm_functionslist() function . . . 1525
94.15 ringvm_classeslist() function . . . 1525
94.16 ringvm_packageslist() function . . . 1526
94.17 ringvm_memorylist() function . . . 1527
94.18 ringvm_calllist() function . . . 1529
94.19 ringvm_fileslist() function . . . 1530
94.20 ringvm_settrace() . . . 1531
94.21 ringvm_tracedata() . . . 1531
94.22 ringvm_traceevent() . . . 1531
94.23 ringvm_tracefunc() . . . 1532
94.24 ringvm_scopescount() . . . 1532
94.25 ringvm_evalinscope() . . . 1532
94.26 ringvm_passerror() . . . 1532
94.27 ringvm_hideerrormsg() . . . 1533
94.28 ringvm_callfunc() . . . 1533
94.29 Example - Using the Trace Functions . . . 1533

94.30 Example - The Trace Library . . . 1537
94.31 ringvm_see() function . . . 1541
94.32 ringvm_give() function . . . 1542
94.33 ringvm_codelist() function . . . 1542
94.34 ringvm_info() function . . . 1542
94.35 ringvm_ismempool() function . . . 1542
94.36 ringvm_runcode() function . . . 1543

95 Tutorial: Ring Extensions in C/C++ 1544
95.1 Hello World . . . 1544
95.2 Build the extension on different platforms . . . 1545
95.3 Sum Two Numbers . . . 1546
95.4 Say Hello . . . 1547
95.5 Sum List of Numbers . . . 1548
95.6 Increment List Items . . . 1550
95.7 Filter List Items . . . 1552
95.8 Replicate List Items . . . 1554
95.9 Generate List . . . 1556
95.10 Display List . . . 1558
95.11 Update Table . . . 1562
95.12 Create Table . . . 1565

96 Extension using the C/C++ languages 1570
96.1 ring_ext.h . . . 1570
96.2 ring_ext.c . . . 1570
96.3 Module Organization . . . 1571
96.4 Function Structure . . . 1572
96.5 Check Parameters Count . . . 1573
96.6 Display Error Message . . . 1573
96.7 Check Parameters Type . . . 1574
96.8 Get Parameters Values . . . 1574
96.9 Return Value . . . 1574
96.10 Function Prototype . . . 1575
96.11 Sin() Function Implementation . . . 1575
96.12 Fopen() and Fclose() Functions Implementation . . . 1575
96.13 Ring API - List Functions . . . 1576
96.14 RING_API_STATE . . . 1579
96.15 Ring API - String Functions . . . 1580
96.16 MySQL_Columns() Function Implementation . . . 1581
96.17 Dynamic/Shared Libraries (DLL/So/Dylib) and LoadLib() function . . . 1582
96.18 Using RING_API_RETMANAGEDCPOINTER() . . . 1583
96.19 Memory Functions . . . 1584

97 Embedding Ring Language in C/C++ Programs 1585
97.1 Ring State . . . 1585
97.2 Ring State Functions . . . 1585
97.3 Ring State Variables . . . 1586

98 Code Generator for wrapping C/C++ Libraries 1588
98.1 Using the tool . . . 1588
98.2 Configuration file . . . 1588
98.3 Using the function prototype . . . 1588
98.4 Adding code to the generated code . . . 1590
98.5 Prefix for Functions Names . . . 1590
98.6 Generate function to wrap structures . . . 1591

98.7 Determine Structure Members Types . . . 1591
98.8 Defining Constants . . . 1591
98.9 Register New Functions . . . 1592
98.10 Writing comments in the configuration file . . . 1592
98.11 Executing code during code generation . . . 1592
98.12 Enum and Numbers . . . 1593
98.13 Filtering using Expressions . . . 1593
98.14 Constants Type . . . 1593
98.15 Configuration file for the Allegro Library . . . 1594
98.16 Threads Support . . . 1595
98.17 Code Generator Rules for Wrapping C++ Classes . . . 1597
98.18 Using configuration file that wrap C++ Library . . . 1598
98.19 Configuration file for the Qt Framework . . . 1598
98.20 Static Methods . . . 1609
98.21 Loading Files . . . 1609
98.22 Managed Classes . . . 1610
98.23 Extra names for functions . . . 1610
98.24 Configuration Files Examples . . . 1610

99 Create your first extension 1612
99.1 Location . . . 1612
99.2 Steps to create the extension . . . 1612
99.3 Testing the extension . . . 1613

100 Language Design 1614
100.1 Why Ring? . . . 1614
100.2 Designed for a Clear Goal . . . 1614
100.3 Simple . . . 1614
100.4 Trying to be natural . . . 1615
100.5 Encourage Organization . . . 1616
100.6 Data Representation . . . 1617
100.7 Compact Syntax . . . 1617
100.8 Define Natural Statements . . . 1618
100.9 Define Declarative Languages . . . 1620
100.10 Transparent Implementation . . . 1621
100.11 Visual Implementation . . . 1622
100.12 Smart Garbage Collector . . . 1624
100.13 No Global Interpreter (VM) Lock - No GIL . . . 1624
100.14 Fast Enough For Many Applications . . . 1624

101 What is new in Ring 1.1? 1625
101.1 List of changes and new features . . . 1625
101.2 Better Natural Language Programming Support . . . 1625
101.3 Generate/Execute Ring Object Files (*.ringo) . . . 1626
101.4 Syntax Flexibility and different styles for I/O and Control Structures . . . 1626
101.5 New Functions and Changes . . . 1629
101.6 StdLib functions and classes written in Ring . . . 1629
101.7 RingLibSDL . . . 1632
101.8 Demo Project - Game Engine for 2D Games . . . 1632
101.9 RingSQLite . . . 1633
101.10 Better Code Generator for Extensions . . . 1634
101.11 Using Self.Attribute in the Class Region to define new attributes . . . 1634
101.12 Using This.Attribute in nested Braces inside the Class Methods . . . 1634
101.13 Better Documentation . . . 1635

102 What is new in Ring 1.2? 1636
102.1 List of changes and new features . . . 1636
102.2 New Functions . . . 1636
102.3 Better Functions . . . 1637
102.4 Better Ring Notepad . . . 1638
102.5 Better RingQt . . . 1638
102.6 Objects Library for RingQt . . . 1639
102.7 RingLibCurl . . . 1640
102.8 Better Call Command . . . 1640
102.9 Using NULL instead of NULLPointer() . . . 1640
102.10 Display Warnings Option . . . 1641
102.11 Better Quality . . . 1641

103 What is new in Ring 1.3? 1642
103.1 List of changes and new features . . . 1642
103.2 Better RingQt . . . 1642
103.3 Better Ring Notepad . . . 1646
103.4 Ring mode for Emacs Editor . . . 1650
103.5 Better StdLib . . . 1650
103.6 Better Loop|Exit Command . . . 1650
103.7 New Functions . . . 1651
103.8 Return Self by Reference . . . 1651
103.9 Using '<' and ':' operators as 'from' keyword . . . 1652
103.10 Embedding Ring in Ring without sharing the State . . . 1652
103.11 RingZip Library . . . 1653
103.12 Form Designer . . . 1654

104 What is new in Ring 1.4? 1656
104.1 List of changes and new features . . . 1656
104.2 Change: Basic Extensions are separated from RingVM . . . 1656
104.3 The Natural Library . . . 1657
104.4 New Style is added to Ring Notepad . . . 1659
104.5 RingREPL . . . 1660
104.6 Convert between Numbers and Bytes . . . 1661
104.7 Better StdLib . . . 1662
104.8 Better WebLib . . . 1662
104.9 Better RingQt . . . 1664
104.10 Qt Class Converter . . . 1664
104.11 What is new in Ring 1.4.1? . . . 1665

105 What is new in Ring 1.5? 1669
105.1 List of changes and new features . . . 1669
105.2 Video-Music-Player Application . . . 1670
105.3 Windows StartUp Manager Application . . . 1670
105.4 Calculator Application . . . 1671
105.5 Better Ring Notepad . . . 1672
105.6 Better StdLib . . . 1675
105.7 Better WebLib . . . 1676
105.8 Better RingQt . . . 1679
105.9 Better Objects Library . . . 1682
105.10 RingFreeGLUT Extension . . . 1682
105.11 RingOpenGL Extension . . . 1693
105.12 Better Code Generator for Extensions . . . 1697
105.13 Better Documentation Generator for Extensions . . . 1697

105.14 Ring VM - Tracing Functions . . . 1697
105.15 Trace Library and Interactive Debugger . . . 1702
105.16 More Syntax Flexibility . . . 1704
105.17 Type Hints Library . . . 1705
105.18 Better Quality . . . 1707
105.19 What is new in Ring 1.5.1? . . . 1707
105.20 What is new in Ring 1.5.2? . . . 1712
105.21 What is new in Ring 1.5.3? . . . 1713
105.22 What is new in Ring 1.5.4? . . . 1716

106 What is new in Ring 1.6? 1718
106.1 List of changes and new features . . . 1718
106.2 Employee Application . . . 1719
106.3 New Tool: Ring2EXE . . . 1719
106.4 Better Ring For Android . . . 1719
106.5 New Tool: Folder2qrc . . . 1721
106.6 Better Scripts for building Ring . . . 1721
106.7 RingConsoleColors Extension . . . 1721
106.8 RingMurmurHash Extension . . . 1722
106.9 Better Ring Notepad . . . 1722
106.10 Better RingQt . . . 1724
106.11 Better StdLib . . . 1725
106.12 Better RingVM . . . 1725
106.13 Better RingREPL . . . 1726
106.14 Using Tab instead of char(9) . . . 1726
106.15 Using CR as Carriage return . . . 1726
106.16 Using the ! operator as not . . . 1727
106.17 Using && and || operators . . . 1727
106.18 Using ? to print expression then new line . . . 1728

107 What is new in Ring 1.7? 1729
107.1 List of changes and new features . . . 1729
107.2 New Command: Load Package . . . 1729
107.3 ringvm_see() and ringvm_give() functions . . . 1730
107.4 ring_state_new() and ring_state_mainfile() functions . . . 1731
107.5 Better Trace Library . . . 1731
107.6 Better Ring Notepad . . . 1732
107.7 Better RingQt . . . 1733
107.8 Better Ring2EXE . . . 1733
107.9 Better RingZip . . . 1734
107.10 Better Documentation . . . 1734
107.11 Better Ring VM . . . 1734
107.12 RingLibuv Extension . . . 1734

108 What is new in Ring 1.8? 1738
108.1 List of changes and new features . . . 1738
108.2 Better Performance . . . 1739
108.3 Find in files Application . . . 1739
108.4 String2Constant Application . . . 1740
108.5 StopWatch Application . . . 1741
108.6 More 3D Samples . . . 1741
108.7 Compiling on Manjaro Linux . . . 1743
108.8 Using This in the class region as Self . . . 1744
108.9 Default value for object attributes is NULL . . . 1745

108.10 The For Loops uses the local scope . . . 1746
108.11 Merge binary characters . . . 1746
108.12 FoxRing Library . . . 1747
108.13 Better Form Designer . . . 1748
108.14 Better Cards Game . . . 1748
108.15 Better RingQt . . . 1749
108.16 Better Code Generator For Extensions . . . 1750
108.17 Better Ring Compiler and VM . . . 1750
108.18 Notes to extensions creators . . . 1751

109 What is new in Ring 1.9? 1752
109.1 List of changes and new features . . . 1752
109.2 New Game : Gold Magic 800 . . . 1752
109.3 More Games . . . 1754
109.4 Better Ring Notepad . . . 1757
109.5 Better StdLib . . . 1758
109.6 BigNumber Library . . . 1759
109.7 RingPostgreSQL Extension . . . 1761
109.8 Deploying Web applications in the Cloud . . . 1763
109.9 Better RingQt . . . 1763
109.10 Better Memory Management . . . 1767
109.11 Better Code Generator for Extensions . . . 1767
109.12 More Improvements . . . 1768

110 What is new in Ring 1.10? 1770
110.1 List of changes and new features . . . 1770
110.2 Chess Game . . . 1770
110.3 Minesweeper Game . . . 1771
110.4 Knight Tour Game . . . 1772
110.5 Game of Life Game . . . 1773
110.6 Pong Game . . . 1774
110.7 Snakes And Ladder Game . . . 1775
110.8 More Games . . . 1776
110.9 Ring Extension for Visual Studio Code . . . 1777
110.10 The Ring Package Manager (RingPM) . . . 1778
110.11 Better Tests . . . 1779
110.12 More Improvements . . . 1779

111 What is new in Ring 1.11? 1781
111.1 List of changes and new features . . . 1781
111.2 More 3D Samples . . . 1781
111.3 Checkers Game . . . 1782
111.4 Sokoban Game . . . 1783
111.5 Maze Game . . . 1784
111.6 Snake Game . . . 1785
111.7 Sudoku Game . . . 1786
111.8 Desktop Screen Shot Application . . . 1788
111.9 Text To Speech Application . . . 1788
111.10 RingRayLib Extension . . . 1789
111.11 ZeroLib Library . . . 1791
111.12 StdLib - More Functions . . . 1792
111.13 Better RingQt . . . 1793
111.14 Better Performance . . . 1795
111.15 Better Documentation . . . 1795

111.16 More Improvements . . . 1795

112 What is new in Ring 1.12? 1797
112.1 List of changes and new features . . . 1797
112.2 Go Game . . . 1797
112.3 ASCII Table application . . . 1798
112.4 BMI Calculator application . . . 1799
112.5 Calendar application . . . 1799
112.6 Julian Day Calendar application . . . 1800
112.7 Tutorial: Number to Words . . . 1801
112.8 Load Again Command . . . 1802
112.9 ring_state_filetokens() function . . . 1803
112.10 Generate Embedded Ring Object File . . . 1804
112.11 Better RingRayLib . . . 1805
112.12 More Improvements . . . 1806

113 What is new in Ring 1.13? 1808
113.1 List of changes and new features . . . 1808
113.2 New Book by Apress . . . 1809
113.3 Ring for WebAssembly . . . 1810
113.4 Better Threads Support . . . 1810
113.5 Laser Game . . . 1811
113.6 Magic Balls Game . . . 1812
113.7 Money Boxes Game . . . 1813
113.8 Matching Game . . . 1813
113.9 Pairs Game . . . 1814
113.10 Word Game . . . 1815
113.11 Tetris Game . . . 1816
113.12 Escape Game . . . 1817
113.13 Hassouna Course Samples . . . 1818
113.14 Ring support in SpaceVim . . . 1819
113.15 Better RingQt . . . 1820
113.16 Better RingRayLib . . . 1822
113.17 RingStbImage Extension . . . 1823
113.18 More Low Level Functions . . . 1824
113.19 Better Organization . . . 1824
113.20 More Improvements . . . 1824

114 What is new in Ring 1.14? 1828
114.1 List of changes and new features . . . 1828
114.2 MyCoffee (Web Application) . . . 1829
114.3 Web Development Samples . . . 1830
114.4 More Samples . . . 1831
114.5 Erlang B Formula . . . 1833
114.6 Customers Form . . . 1834
114.7 RingTilengine Extension . . . 1835
114.8 RingLibui Extension . . . 1838
114.9 RingSockets Extension . . . 1842
114.10 RingThreads Extension . . . 1842
114.11 Better RingOpenSSL . . . 1843
114.12 More Functions . . . 1844
114.13 Better Functions . . . 1845
114.14 Better Performance For Strings . . . 1845
114.15 Better Handling For Numbers . . . 1846

114.16 Using CLOC (Count Lines of Code) . . . 1847
114.17 More Improvements . . . 1848

115 What is new in Ring 1.15? 1850
115.1 List of changes and new features . . . 1850
115.2 New Arabic Book . . . 1850
115.3 Chess Endgame Application . . . 1851
115.4 Chess 3D (Qt3D Sample) . . . 1853
115.5 Better Ring Notepad . . . 1855
115.6 Better StdLib . . . 1856
115.7 Better RingQt . . . 1857
115.8 Better RingLibCurl . . . 1858
115.9 Heroku (Better support) . . . 1858
115.10 Using ':' then Numbers . . . 1859
115.11 More Samples . . . 1859
115.12 More Improvements . . . 1860

116 What is new in Ring 1.16? 1862
116.1 List of changes and new features . . . 1862
116.2 Light GUILib . . . 1862
116.3 UTF-8 File Names in Microsoft Windows . . . 1863
116.4 Nested Methods Call in Separate Lines . . . 1864
116.5 Code Runner Extension support Ring . . . 1865
116.6 Zero and Strings . . . 1866
116.7 Better Installation Scripts . . . 1867
116.8 Better Documentation . . . 1867
116.9 MDI Windows Sample . . . 1867
116.10 More Improvements . . . 1867

117 What is new in Ring 1.17 1869
117.1 List of changes and new features . . . 1869
117.2 Merge Two Files . . . 1869
117.3 Poetry Analysis . . . 1870
117.4 Citations Prediction . . . 1871
117.5 TokensLib Library . . . 1872
117.6 CSVLib Library . . . 1873
117.7 JSONLib Library . . . 1875
117.8 HTTPLib Library . . . 1876
117.9 Better GUILib . . . 1877
117.10 Better RingOpenSSL . . . 1879
117.11 Better RingWinAPI . . . 1879
117.12 Better Syntax Flexibility . . . 1880
117.13 Better I/O Functions . . . 1880
117.14 Better Ring API . . . 1881
117.15 Better Code Generator for Extensions . . . 1882
117.16 More Improvements . . . 1882

118 What is new in Ring 1.18 1885
118.1 List of changes and new features . . . 1885
118.2 Read Files application . . . 1886
118.3 Random User application . . . 1886
118.4 Better CitPre application . . . 1888
118.5 Better Tools . . . 1888
118.6 More Samples . . . 1889
118.7 Syntax Files . . . 1890

118.8 The Power Operator . . . 1891
118.9 Using References . . . 1891
118.10 Ring for MS-DOS . . . 1892
118.11 Faster (For-In) Loop . . . 1893
118.12 Faster BraceError() . . . 1893
118.13 Better RingRayLib . . . 1894
118.14 Better RingSockets . . . 1896
118.15 Heroku (Better support) . . . 1896
118.16 Command: New From . . . 1896
118.17 ImportPackage() Function . . . 1897
118.18 More Low Level Functions . . . 1897
118.19 Better WebLib . . . 1897
118.20 Better Ring API . . . 1898
118.21 Better Documentation . . . 1899
118.22 MatrixLib Library . . . 1899
118.23 More Improvements . . . 1900

119 What is new in Ring 1.19 . . . 1903
119.1 List of changes and new features . . . 1903
119.2 Image Pixel application . . . 1904
119.3 Hours Counter application . . . 1904
119.4 Planetary Orbits application . . . 1905
119.5 More Samples . . . 1905
119.6 Ring for Windows 64bit . . . 1906
119.7 Better batch files for building Ring . . . 1907
119.8 For Loop - Better Performance . . . 1907
119.9 Calling Functions - Better Performance . . . 1907
119.10 Nothing() function . . . 1908
119.11 OptionalFunc() function . . . 1909
119.12 FastPro Extension . . . 1909
119.13 Better TypeHints . . . 1910
119.14 Better RingRayLib . . . 1911
119.15 Better RingStbImage . . . 1912
119.16 Better Extensions Generator . . . 1914
119.17 Better Documentation . . . 1915
119.18 More Improvements . . . 1915

120 Using Other Code Editors . . . 1918
120.1 Using Notepad++ . . . 1918
120.2 Using Geany . . . 1919
120.3 Using Atom . . . 1920
120.4 Using Sublime Text 2 . . . 1921
120.5 Using Visual Studio IDE . . . 1922
120.6 Using Emacs Editor . . . 1923
120.7 Visual Studio Code . . . 1923
120.8 SpaceVim . . . 1924

121 Frequently Asked Questions (FAQ) . . . 1925
121.1 Why do we need Yet Another Programming Language (YAPL)? . . . 1925
121.2 Which of 3 coding styles are commonly used or recommended by the community? . . . 1926
121.3 What is the Ring Architecture? . . . 1927
121.4 What about Memory Management in Ring? . . . 1927
121.5 What about Data Representation in Ring? . . . 1928
121.6 Why is Ring weakly typed? . . . 1930

121.7 What are the advantages to using Ring over Lisp or Smalltalk? . . . 1930
121.8 Why is Ring largely focused on UI creation? . . . 1931
121.9 Is Ring some sort of an improvement of PHP? . . . 1931
121.10 What are the advantages of using Ring over native C or C++? . . . 1931
121.11 What is the difference between Ring and Python? And is Ring Open Source? . . . 1931
121.12 What are the advantages to using Ring over Perl, PHP, Python or Ruby? . . . 1932
121.13 What are the advantages to using Ring over Tcl or Lua? . . . 1933
121.14 What are the advantages to using Ring over C# or Java? . . . 1933
121.15 The documentation says functional programming is supported, but then this happens? . . . 1933
121.16 Why the ability to define your own languages Instead of parsing? . . . 1934
121.17 Why you can specify the number of loops you want to break out of? . . . 1934
121.18 Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords? . . . 1934
121.19 What is the philosophy behind data types in Ring? . . . 1935
121.20 What about the Boolean values in Ring? . . . 1935
121.21 What is the goal of including the "Main" function in Ring? . . . 1936
121.22 Why the list index start from 1 in Ring? . . . 1937
121.23 Why Ring is not case-sensitive? . . . 1938
121.24 Why the Assignment operator uses Deep Copy? . . . 1938
121.25 Is there constructor methods in Ring? . . . 1939
121.26 What happens when we create a new object? . . . 1940
121.27 Can we use the attributes by accessing the Getter and Setter methods? . . . 1941
121.28 Why should a search of global names be made while defining the class attributes? . . . 1942
121.29 Why Ring doesn't avoid the conflict between Global Variables and Class Attributes Names? . . . 1943
121.30 Where can I write a program and execute it? . . . 1943
121.31 How to get the file size using ftell() and fseek() functions? . . . 1944
121.32 How to get the current source file path? . . . 1944
121.33 What about predefined parameters or optional parameters in functions? . . . 1945
121.34 How to print keys or values only in List/Dictionary? . . . 1945
121.35 Why I get a strange result when printing nl with lists? . . . 1946
121.36 Could you explain the output of the StrCmp() function? . . . 1946
121.37 How to use many source code files in the project? . . . 1947
121.38 Why this example use the GetChar() twice? . . . 1948
121.39 How to use NULL and ISNULL() function? . . . 1949
121.40 How to print lists that contains objects? . . . 1950
121.41 How to insert an item to the first position in the list? . . . 1950
121.42 How to print new lines and other characters? . . . 1951
121.43 Why I get Calling Function without definition Error? . . . 1951
121.44 Can Ring work on Windows XP? . . . 1952
121.45 Why we don't use () after the QApp class name? . . . 1952
121.46 Why the window title bar is going outside the screen? . . . 1952
121.47 How to create an array of buttons in GUI applications? . . . 1953
121.48 How to Close a window then displaying another one? . . . 1954
121.49 How to create a Modal Window? . . . 1955
121.50 How can I disable maximize button and resize window? . . . 1956
121.51 Why setClickEvent() doesn't see the object methods directly? . . . 1956
121.52 How to extend RingQt and add more classes? . . . 1957
121.53 How to add Combobox and other elements to the cells of a QTableWidget? . . . 1959
121.54 How to perform some manipulations on selected cells in QTableWidget? . . . 1960
121.55 How to use SQLite using ODBC? . . . 1961
121.56 Can I connect to dbase/harbour database? . . . 1962

122 Building From Source Code **1964**
122.1 Building using Microsoft Windows . . . 1964
122.2 Complete Build (Compiler/VM, Extensions, Tools, etc.) . . . 1964

122.3 64bit version . . . 1964
122.4 Custom Build . . . 1965
122.5 Building using Ubuntu Linux . . . 1967
122.6 Building using MacOS X . . . 1970
122.7 Building using CMake . . . 1973

123How to contribute? 1974
123.1 Special thanks to contributors . . . 1974
123.2 Documentation . . . 1974
123.3 Testing . . . 1975
123.4 Samples . . . 1975
123.5 Applications . . . 1975
123.6 Editors Support . . . 1975
123.7 Libraries in Ring . . . 1975
123.8 Extensions in C/C++ . . . 1975
123.9 Compiler and Virtual Machine (VM) . . . 1976

124Language Specification 1977
124.1 Language Keywords . . . 1977
124.2 Language Functions . . . 1979
124.3 Compiler Errors . . . 1984
124.4 Runtime Errors . . . 1985
124.5 Environment Errors . . . 1987
124.6 Language Grammar . . . 1987
124.7 Virtual Machine (VM) Instructions . . . 1989

125Resources 1994
125.1 Ring Language Website . . . 1994
125.2 Source Code . . . 1994
125.3 Ring Group . . . 1994
125.4 Ring Team . . . 1994

Contents:

CHAPTER

ONE

APPLICATIONS DEVELOPED IN A FEW HOURS

Ring is a new programming language that focuses on the Natural Language Programming and Declarative Programming paradigms and will let you think different about programming and how to solve your problems in a better way. In little days after the first release we got thousands of downloads and many developers started learning and using the language. Their feedback are the secret behind the language progress and success. They said that Ring is powerful, beautiful and easy to learn, Some of them provided good examples about what can be done using Ring in a few hours. They are very happy with the language productivity.

1.1 Quotes about Ring

"I like Smalltalk very much but now I like Ring better!"
, Gal Zsolt (Hungary)

"I find the language and its syntax very natural and easy to follow."
, Bhudda (United States)

"Very nice approach for a new language."
, Matth Moestl (Austria)

"Very interesting! I will keep an eye on it!"
, Eslipak (Argentina)

"I'd like to see some benchmarks. Otherwise, at first glance, it looks really promising."
, Alex Deva (Sweden)

"Excellent"
, Liju Sankar (United States)

"I wish you the best with this project."
, David O'Neil (United States)

"Just fantastic."

, Jose Antonio (Mexico)

"This looks like it was developed by some very competent people."
, Jim Clack (United States)

"The Ring programming language seems pretty interesting so far."
, Eric Johnson (United States)

"Thank you for this awesome language and wonderful ready to use Qt binding."
, Martial FAESSEL

"I think it's great what he does for the community of developers and novice programming."
, Marino Esteban

"Ring is just awesome. The language is so cool and fluent. I am sure it's going to be BIG."
, Ahmed Zain

"What a proud, really wish you Eng. Mahmoud Samir and Your Team moving forward ^_^ and from now , considered me a big fan of the Ring programming language."
, Zainab Mahmoud

"Well guys I love this language and it appears that you have created perfect language."
, Moemen Ghulmi

"Good work Mahmoud, I've installed ring pl, and it's very perfect language."
, Ahmed Omar (Egypt)

"Thanks for this great startup programming language. I wish you best of luck."
, Elkhouaja Khalid

"Congratulations! I am very happy and I wish you Success and good luck."
, Abobasmla Hassan

"Good Features of multi-use language on the Web, Mobile and Desktop."
, Abdelrhman Haider

"Very interesting effort."
, Giannakakis Kostas (Greece)

"I am too lazy to open comment window and write message. But in your case I must say (Perfect) Really, create new remarkable language like your Ring is really perfect job. Even create editor for your language in your language with only few rows… Even noticed in rosettacode.org ! I read your previous articles and I tried these examples a few days ago and I will continue. I love Ring. P.S.: Anders Hejlsberg, Niklaus Wirth, Bjarne Stroustrup, Ada Lovelace Hall of fame is waiting…."

, Martin Nedopil (Czech Republic)

"Ring seems very attractive to me through its very easy design and the Qt bindings. I like its declarative approach and the generous documentation."

, Shalok Shalom (Austria)

"Ring (and plenty of extension library + Qt) is wonderful."

, Kovacs Attila (Hungary)

"Since two days I'm trying Ring and I'm really impressed, in add to power commands and easy use, it's really very efficient and very fast. Each day I hope to find the couple of the year PWCT+RING … Maybe for my Christmas gift!!!! HO HO HO HO Continue your fantastic job and congratulations."

, Jose Le Roux (France)

"There are 3 different styles, it looks like Python and C"

, 64remy

"I was taking a tour around Rosettacode and have found Ring. I like the syntax a lot. It's clean and easy to understand. It looks like a very clean BASIC dialect without sigils. I can say that this is the easiest and the most BASIC-like language I've ever tried."

, Tomaaz

"Thanks for your effort. I took a quick look and found it interesting. You are trying to follow more or less like Clipper with simple command and no rigid declaration rules. Good."

, Anand Gupta

"Thanks for this wonderful language"

, Vinod kc (India)

"Very enlightening. good job!"

, Southmountain (United States)

"The thing I liked was the loop exiting"

, Leon de boer (Australia)

"An outstanding and easy language to program with."

, Kenneth Burgo (Philippines)

"I chose your language as I feel I can understand it better than other languages"

, Harry Singh

"I like the totality of the language, far more features than expected and the freedom of expressiveness is unique."

, Evikone

"Thank you very much Mahmoud! I am using ring for many experiments and so far I love it. I really want to continue using ring and contribute what I can."

, John (SienSystem)

"Sir, Very Good"

, Kamlesh Patel

"That's more than a "cool" syntax, the example of writing free-form text between curly-brackets such that each word calls a function. Which could be interesting (A syntax like that would be nice for declaring text styles)"

, LaurieCheers

"If you browse around you see they have listed 160 contributors. This year they have entered Top 100 in the TIOBE index. Lot of effort seem to have been made to make this language pop out and catch the attention of masses."

, Htuhola

"I like the idea of The Ring being in ANSI C it's an impressive creation, and a lot of skill went into it"

, Garry Taylor

"Very innovative language! Syntactically clean"

, CodeProject Member

"The author must be commended for the readily-obvious hard work and effort that has gone into creating a rich ecosystem for his language. It seems that the language is quite extensive as well. I would find it useful to see a BNF grammar and concise coverage of its semantics."

, Xx-Leninist-1917-Xx (Reddit)

"I can see the AI of the future using this technology to solve computational problems for….. the humans."

, Cryptonite (United States)

"I like your programming language, I like you are going to develop mobile app using RingQt and also I appreciate your web library."

, Domenico D'Oria (Italy)

"Congratulations for the great work with this new programming language."

, Kenny Silva (Venezuela)

"Ring is an amazingly full-featured language and so well documented (the bane of most newer languages out there!)"

, Alex McCullie

" I found the language yesterday, and liked the Qt bindings, as they give a declarative way to create a QtWidgets GUI."

, Cochise Cesar

"Ring does look intriguing, and I'll be reading more of the documentation soon"

, Jamie Cooper

"I was recently considering designing my own dynamically-typed, prototypical language and then developing a means to compile it into C/C++. However, last night I was surfing the web and noticed a little-known language called "Ring" which you've recently created. I began reading the Motivation section in Wikibooks to see why the language was designed and implemented, and I was shocked to see that someone else had created a language with the same intention and need as myself. I mean, it's mind-blowing that someone would have addressed every issue I have with the currently accepted languages. It's crazy to think it's only been around for a year and yet, it's already, practically a batteries-included language."

, Gedalya (YouTube)

"Thank you Mr. Mahmoud for all the wonderful work, whenever I dive in the Source code I see the great effort, further development more excellence, God will reward you with what you wish"

, Azzedine Ramal

"What a great joy to find this surprisingly genius language !!!. It was a total joy to go through the documentation and look at the samples in Rosetta code. Marvelous work. I would even leave my fortune to the development of this language. Keep the good work going and wishing this language will go viral. ps: Thanks for keeping array index to start with 1. It means a lot."

, Nehemiah Jacob (Sydney, Australia)

"I find it very interesting. Especially the fact that it is cross platform makes it something to keep an eye on."

, Boudewijn Lutgerink

"The language I like the most. Efficient, simple, easy, flexible and wonderful language."

, Roshan Ali

"I like variety. The richer the toolbox, the more appropriate the tool."

, Jonathan Day (Quora)

"I greatly appreciated your work and congratulations on what has been achieved."
, Umberto Meglio (Italy)

"Thanks a lot I just found Ring two days ago and I decided to learn it, it is amazing, the samples are helping a lot."
, Zaraki Ken

"Extraordinary. well done sir"
, RugbyLeague (CodeProject Member - United Kingdom)

"The Ring language is pleasant. You get ahead very quickly."
, Neskuk (CodeProject Member - Switzerland)

"I'm happy to use your language."
, Akhil Reddy

"I am enjoying using the Ring Programming language."
, Gabriel Wendt

"I'm loving this language!"
, Pablo Prieto

"Ring Notepad is a an example of the power of Ring. It is like NotePad++ with built in form design capability, drag and drop, object attributes etc. It is a marvelous application"
, Bert Mariani (Italy)

"Ring is the language that offers the greatest potential for converting programmers frustrated with the amount of time it takes to develop apps in C/C++, C# and other OOP-based languages. Considering the extent to which Ring has evolved since the first release in 2016, the Ring team has proven itself worthy of a very high achievement award in the world of programming languages. The extent to which Ring has simplified the development database apps, web apps and GUI apps is a great credit to the Ring team. Ring's implementation of OOP and GUI based apps is far superior to C++ and C#. Another major achievement of the Ring team is the ease at which programmers can get on-line access to Ring documentation, compared to on-line C++ and C# documentation. The Ring Game Engine is truly elegant and it's designer (s) deserve lots of credit for such an impressive bit of software."
, L Godioleskky

"Strongly speaking, it is a strong and new game programming language."
, isVowel (Japan)

"The language seems to have some interesting features, specially the use of braces to access object fields and methods."

, Nuaua (Reddit)

"The code that implements the Ring VM actually looks quite nice. There is a bunch of test code – great!"
, Peterfirefly (Reddit)

"Ring seems promising. It first appeared in 2016 although it's concept is older. It is meant to be portable and can be embedded in C/C++ projects"
, Wim ten Brink (Quora - Top Writer (2018))

"Interesting alternative to Lua."
, djxtc (Sourceforge)

"Ring is a serious thing in the programming language landscape. One should understand its foundation before she can shape an opinion worth considering. There are several innovations outthere with a clear commitment from its designer to simplicity, flexibility and learnability. Natural language as its imagined by Ring opens a window for an unlimited set of applications otherwise impossible to think about. One can built an interactive chatbot based on a rich domain specific language in a matter of hours not months. Combined with a smart yet effective implementation of a declarative programming paradigm, Ring empowers the creation of any kind of programming language on top of it, with any set of syntax. Even the language keywords can be overwritten to serve one's own keywords in any human language not only english. Ring is a disruption, a big intellectual step forward but also a beautiful reincarnation of several legacy ideas and best practices from other old languages like Basic, Lisp and others. Those ideas are a humanity heritage that new visionaries like Mahmoud S. Fayed, the Ring designer, but also Rick Hickey, the Clojure designer, are trying to callback in today's world of complex programmability. An other dimension of Ring, which formed one of its basic motivations, relates to visual programming. Software is a fluid and evolutionary creature and textual code is not accessible to the most of common humans. The promise of visual programming is to empower people, enhance understandability and invite business users to the arena of software development more intuitively. Ring was made by design with an objective of having a programming language capable of better supporting the visual paradigm. Gamification is a first-class citizen in the language. This is a tremendous advantage compared to other general-purpose programming languages. Ring supports a large number of gaming libraries and open a large window of capabilities to use, not only in gaming but also in business and education applications. The best way to assessing the power of Ring is to looking into it and reflecting about the number of problems it is capable to solve."
, Mansour Ayouni (Tunisia)

"I have only known ring-lang for few weeks, really love features and visions of ring: bracket access, class region, declarative and natural programming. It's simple and beautiful the way ring deal with these paradigms. I think it's really great work what you're done, and doing. I want to develop ethereum client in ring language, not only but it's style. It'll have natural and declarative code on it's main part. Furthermore, It will be fantastic if there is "natural language -> evm bytecode" compiler, maybe ring have the solution"
, Asta

"The language looks great!"
, Eric Hawk

"You put a lot of work in the Ring and you're a great programmer. I saw your examples in 3D I've looked at

part of your project, it's really great"

, Rafal Jopek

"Looks like a big project and an impressive piece of work"

, Rochus Keller (PhD From ETH Zurich, Switzerland)

"I see Ring a very useful and easy in PROGRAMMING IN REGIONAL LANGUAGE. I tried to program in my mother language Hindi (India) using UNICODE It is working. No other language give me such an easy platform."

, Negi Manoj (India)

"I'm glad I could help improve the Ring language. It has good and yet easy to understand design and architecture. I believe it has big potential ahead."

, Mounir IDRASSI (Creator of VeraCrypt)

"What I like most about Ring Lang is its ease to create complex applications in WebAssembly."

, Luis Lozad

"I just discovered Ring today and immediately liked it and decided to give it a try. As a senior JAVA / ADA and Python developer I am very curious about the multi-paradigm"

, Bertrand Pivaty

"Wow, that is pretty amazing! It's not every day to hear about a new programming language that looks actually cool! Oh wow, didn't even see the VB-like features there. This has some real potential!"

, EternityForest (reddit)

"I am a big fan of your works I look forward to contributing to some of your projects in the future."

, Adewale Azeez (Nigeria)

"I am a software development manager working in a Canadian Firm. I have been around for 25 years playing with different Programming languages like Python, Perl, VB, C#, Java, REBOL, Euphoria etc. Thanks a lot for the wonderful software and the great documentations"

, Lijo Joseph (Canada)

"I am studying Ring (I am a senior programmer). I certainly never cease to be amazed about this language; it seems to be a wish from a marvellous lamp of computer genius"

, Antonio F.S. (Spain)

"I'm learning the language and so far I'm finding it wonderful. It has great potential."

, Jose Luis Cuesta (Spain)

"Ring is a breath of fresh air in language designs for application development."
, Gary Chike

1.2 FetchStockData Application

It will fetch stock data from Yahoo and draw various types of charts.

1.3 Google API Shortener Application

A small but very nice tool to shorten urls using the Google shortener API.

1.4 Video-Music-Player Application

An application to play multi-media files.

1.5 Fifteen Puzzle Game

An implementation for the Fifteen Puzzle game.

1.6 Analog Clock

Simple implementation for the Analog Clock.

1.7 Squares Puzzle Game

An implementation for the popular Squares Puzzle game.

1.8 Samples in this book

The next samples are developed in a few hours and we will introduce them through this book.

The next screen shot for the Ring Notepad application

The next screen shot for the Cards Game

The next screen shot for the Cards Game (Android)

The next screen shot from the Web Development chapter

The next screen shots for simple 2D Games that we will present in the Game Engine Chapter.

Stars Fighter Game

Flappy Bird 3000 Game

Super Man 2016 Game

The next screen shot for the TicTacToe 3D Game

Screen Shot:

The next screen shot for the Gold Magic 800 Game

The next screen shot from the RingQt3D chapter

1.9 Innovative

The language comes with better support for Natural Language Programming and Declarative Programming. The innovation comes in supporting these paradigms with new practical techniques on the top of Object-Oriented Programming and Functional Programming. No need to know anything about (Compilers and Parsing). You get the language constructs ready for use to create domain-specific languages in a fraction of time.

Articles:-

- Natural Language Programming Library :

https://www.codeproject.com/Articles/1200766/Using-the-Natural-Language-Programming-Library-NLP

- Natural Language Programming :

https://www.codeproject.com/Articles/1138605/Natural-Language-Programming-in-the-Ring-Programmi

- The Declarative Approach :

https://www.codeproject.com/Articles/1222105/The-declarative-approach-of-the-Ring-programming-l

- Syntax Flexibility :

https://www.codeproject.com/Articles/1137388/Syntax-Flexibility-in-the-Ring-Programming-Languag

- The Ring Programming Language :

https://www.codeproject.com/Articles/1089887/The-Ring-Programming-Language

1.10 Practical

Many of the Ring libraries (StdLib, WebLib, Natural Library, Games Engine, etc.) and the Ring IDE (Ring Notepad, Form Designer, etc.) are written in the Ring language itself. Ring is ready for use in production and increase the developers productivity.

Check the Form Designer source code : https://github.com/ring-lang/ring/tree/master/tools/formdesigner

We can run the Form Designer as Android application!

Also we can run it in the Web Browser as a web application using WebAssembly

CHAPTER

TWO

INTRODUCTION

Welcome to the Ring programming language!

Ring is an Innovative and practical general-purpose multi-paradigm language that can be embedded in C/C++ projects, extended using C/C++ code and/or used as standalone language. The supported programming paradigms are Imperative, Procedural, Object-Oriented, Functional, Meta programming, Declarative programming using nested structures, and Natural programming. The language is portable (MS-DOS, Windows, Linux, macOS, Android, WebAssembly, etc.) and can be used to create Console, GUI, Web, Games and Mobile applications. The language is designed to be Simple, Small and Flexible. Also, It's fast enough for many applications. Its Dynamic Language that compile the source code to byte code then execute it by the Ring Virtual Machine, which is integrated with the Ring Compiler in one program.

In this chapter we are going to discuss the goals behind the language design and implementation.

2.1 Motivation (1)

In Nov. 2011, I started to think about creating a new version of the Programming Without Coding Technology (PWCT) software from scratch.

I was interested in creating multi-platform edition of the software beside adding support for Web & Mobile development.

What I was looking for is a programming language that can be used to build the development environment, provides multi-platform support, more productivity, better performance, can be used for components scripting and developing different kinds of applications.

Instead of using a mix of programming languages, I decided to use one programming language.

I looked at many programming languages, but I discovered that I need a different language that comes with new ideas and intelligent implementation.

Note: The importance of the PWCT2 project to the Ring language design is similar to the importance of the Unix operating system to the C language design. Having a specific project in the mind of a programming language designer help a lot in taking the right decisions towards a clear goal. For example, We know that the PWCT2 visual programming

language will provide readability, this let us focus in Ring design on writability. So, we can get maximum readability and maximum writability at the same time since both languages are designed together. Also, PWCT2 as a visual language requires powerful GUI library and fast graphics. That's why Ring comes with support for the Qt framework as it's primary GUI toolkit.

2.2 Motivation (2)

In 2009-2010, I developed a small domain-specific programming language for GUI development called Supernova. I invested many months of my time to create this language and get something that I can use for small programs. After creating Supernova, I realized that if the programming language have some new features on the top of object-oriented programming, it could help me create a domain-specific language like Supernova in days instead of many months. So Ring as a language provide these features, and using Ring we can create Supernova and enjoy its features in real-world applications development. Ring is designed to be a superset of Supernova and enable us to create many domain-specific languages.

Note: The role that Supernova language played for the Ring language design is similar to the role played by the ABC language for the Python language design. When we create a programming language and try to use it in practical, we will learn a lot of lessons that help us when designing the next language.

2.3 Motivation (3)

We can think of a programming language as a user interface between the user and the machine. The user could be a programmer have deep knowledge about the machine architecture, a developer interested in the application and the requirements of his/her users, or a child wants to discover programming. Also, we have many types of other users like researchers, people with deep knowledge about specific domains and so on.

Providing a programming language that uses just a specific syntax doesn't match what we learned about user interfaces. Ring is a language designed to bring translation, and customization at the level of the language syntax. It's not about providing multiple versions of the software where each version uses a different human language. It's about the ability to change the syntax at the project level, where using one version of Ring, we can switch between different human languages, and different customization of syntax. Instead of developing different programming languages that uses the same VM, Ring uses one compiler that support changing the syntax and this can be done many times in the same project.

Note: Choosing the programming language syntax is a huge power and provide a great joy. Using Ring we transfer this power from the programming language designer to the language users.

2.4 Motivation (4)

Many programming languages for application development come and go. i.e., Becomes no longer under development!

I remember having a good time while using Clipper, Classic Visual Basic, Visual FoxPro, etc.

Ring is designed to be a language for application development that can stay for many decades in the future.

To achieve this, Ring is a free and open-source project that works on different platforms and has a small implementation that other programmers could understand and improve. It comes with a visual implementation that could help in this process. Ring is designed and implemented in a way that enables other programmers to easily understand its design and implementation in a short time.

The selection of data types, How the Garbage Collector works, How does the language support threads? All of these decisions are made in a specific way to keep the language very portable and ready for the future (As much as we can, Since we don't know the future, but we can try to invent/predict it!).

Note: We can run Ring on a very old operating system like MS-DOS. Also, we can run it using a modern platform like WebAssembly.

2.5 Motivation (5)

I love programming languages, and I have used a lot of them during my work. When I think about a programming language, I feel something. It's not about the language features; it's about the language spirit and the unique things that can be done using this language. I am not talking about the final result, where we can use many languages in the same group to achieve the same goal. I am talking about the solution itself. the design and the code. How does it look like? Is it beautiful? What about the code size? Is it simple or complex? and things like that, which have a direct effect on source code comprehension and software development productivity.

Ring is designed to include the spirit of many other languages. We don't do this by copying features or mixing things that don't match together! Our technique is different. It's similar to how a child could look like his or her parents. You get the feeling, not a copy!

Note: Ring is designed to be small in size, like Lua, but it comes with standard support for many programming paradigms, like Python and Ruby. This is a challenge when we talk about implementation and development.

2.6 Ring and other languages

Ring is a programming language that comes with better support for Natural Language Programming and Declarative Programming. The innovation comes in supporting these paradigms with new practical techniques on the top of Object-Oriented Programming and Functional Programming.

Also, Ring is influenced by the next programming languages

- Lua
- Python
- Ruby
- C

- C#
- BASIC
- QML
- xBase
- Supernova

2.7 History

In Sept. 2013 I started the design and the implementation of the Ring programming language. After 21 months of development, In May 2015 the language Compiler & Virtual Machine were ready for use!

After that I spent three months testing the language again, trying to discover any bug to fix, writing better tests, by the end of August 2015, all know bugs were fixed, Writing many tests and testing automation helped a lot in getting a stable product.

In September 12, 2015, most of the documentation was written. Before releasing the language I started the marketing by writing a post in Arabic language about it to my facebook profile page asking for contributors interested in the language idea after reading a short description, In the same day I got a lot of emails from developers and friends interested to contribute!

Ring 1.0 is released on January 25, 2016

Ring 1.1 is released on October 6, 2016

Ring 1.2 is released on January 25, 2017

Ring 1.3 is released on May 15, 2017

Ring 1.4 is released on June 29, 2017

Ring 1.5 is released on August 21, 2017

Ring 1.6 is released on November 30, 2017

Ring 1.7 is released on January 25, 2018

Ring 1.8 is released on June 25, 2018

Ring 1.9 is released on October 6, 2018

Ring 1.10 is released on January 25, 2019

Ring 1.11 is released on September 15, 2019

Ring 1.12 is released on January 25, 2020

Ring 1.13 is released on September 15, 2020

Ring 1.14 is released on January 25, 2021

Ring 1.15 is released on September 24, 2021

Ring 1.16 is released on October 23, 2021

Ring 1.17 is released on May 14, 2022

Ring 1.18 is released on July 12, 2023

This is Ring 1.19 documentation.

2.8 Features

The Ring language comes with the next features

Tip: The language is ready for production!

- Free Open Source (MIT License)
- Hybrid Implementation (Compiler + Virtual Machine)
- Declarative programming on the top of Object-Oriented programming
- Natural Language Programming on the top of Object-Oriented programming
- Natural Language Programming Library
- Three different styles for writing the code and you can create your style
- Syntax Flexibility (You can change the language keywords and operators)
- The language keywords can be translated from English to other languages (Arabic, French, etc)
- Compact Syntax, No explicit end for statements (No ; or ENTER is required)
- Using braces { } we can access objects and use attributes/methods as variables/functions
- Transparent Implementation (See the Tokens, Grammar, and Byte Code for each program)
- Visual Implementation - Developed using Visual Programming (PWCT)
- Written in ANSI C (The code is generated + Looks identical to Handwritten Code)
- **A small language**
  - The Compiler + The Virtual Machine + Standard Functions are less than 25,000 lines of C code
  - The other 500,000 lines of code are related to libraries!
- Portable (MS-DOS, Windows, Linux, macOS, Android, WebAssembly, etc)
- Comments (One line & Multi-lines)
- Not Case-Sensitive
- Dynamic Typing
- Weakly typed (Automatic conversion between numbers and strings only)
- Lexical Scoping (Global, Local & Object State)
- Default scope for variables inside functions (Local)
- Default scope for variables outside functions (global)
- We can have separate global scope for each library or sub project
- Garbage Collector - Automatic Memory Management (Escape Analysis and Reference Counting)
- In most cases (90%) Ring VM uses Escape Analysis an no need to run the Garbage Collector (Faster)
- Structure Programming
- Rich control structures & Operators
- For in get item by reference not value, you can read/edit the item
- Use exit to go outside from more than one loop (Use it for programming in the small only)

- Procedures/Functions
- Main Function (Optional - To avoid using the Global Scope)
- Call Function before the definition (Top-Down Programming)
- Recursion
- Multi-line literals
- Access (read/write) string letter by index
- The list index start by 1
- No keyword to end Functions/Classes/Packages
- Range operator ex: 1:10 and "a":"z"
- First Class Variables, Lists, Objects and Functions
- Store/Copy Lists/Objects by value (Deep Copy)
- Pass Lists/Objects by reference
- Create references at any time using Ref() function
- Native Object-Oriented Support
  - Encapsulation
  - Setter/Getter (optional)
  - private state (optional)
  - Instantiation
  - Polymorphism
  - Composition
  - Inheritance (Single Inheritance)
  - Operator Overloading
  - Packages
- Reflection and Meta-programming
- Clear program structure (Statements then functions then packages & classes)
- Exception Handling
- Eval() to execute code during run-time
- 8-bit clean, work on binary data directly
- I/O commands
- Math functions
- String functions
- List functions
- File processing functions
- Database support (ODBC, SQLite, MySQL & PostgreSQL)
- Security Functions (OpenSSL)
- Internet Functions (LibCurl)

- Zip Functions
- CGI Library (Written in Ring)
  - HTTP Get
  - HTTP Post
  - File upload
  - Cookies
  - URL Encode
  - HTML Templates
  - HTML Special Characters
  - HTML Generation using Functions
  - HTML Generation using Classes
  - CRUD Example (using MVC)
  - Users Example (Register, Login and Check)
- Deploying web applications in the Cloud
- Extension using C/C++ (Simple API)
- Embedding the language in C/C++ programs
- Embedding Ring in Ring
- **Comes with code generator (Written in Ring) to quickly wrap C/C++ Libraries**
  - Used to Support Allegro by creating RingAllegro
  - Used to Support LibSDL by creating RingLibSDL
  - Used to Support Qt by creating RingQt
- Create 2D Games for Desktop and Mobile (Using the Allegro Library)
- RingLibSDL Extension
- Comes with simple Game Engine for 2D Games
- RingOpenGL Extension
- RingFreeGLUT Extension
- RingRayLib Extension
- RingTilengine Extension
- Create GUI Applications for Desktop, WebAssembly and Mobile (Using the Qt Framework)
- Comes with IDE contains the Code Editor (Ring Notepad) and the Form Designer
- RingREPL (Read-Eval-Print-Loop)
- Tracing and Debugging
- Type Hints Library
- Comes with Ring2EXE to distribute applications
- RingLibuv Extension
- RingLibui Extension

- RingSockets Extension
- RingThreads Extension
- No Global Interpreter (VM) lock (No GIL) - Better for threads and concurrency (Faster)
- Comes with RingPM (Package Manager)
- Many Samples and Applications
- Complete Documentation.

2.9 License

The Ring Programming Language

http://ring-lang.net/

Version 1.19

The MIT License (MIT)

Copyright (c) Mahmoud Fayed

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER

THREE

USING RING NOTEPAD

In this chapter we will learn about using Ring Notepad to write and execute Ring programs quickly

Ring Notepad is just an application developed using the Ring programming language.

The source code exist in the ring/tools/ringnotepad folder.

3.1 Ring Notepad - Main Window

When we run the Ring Notepad we get the next dockable windows

- Project Files Window : where we can select and open any ring file (*.ring) quickly.
- Source Code Window : Where we write the source code.
- Form Designer Window : The Form Designer to create GUI application forms.
- Web Browser Window : Where we read the documentation or quickly open any website.
- Output Window : Output when we run programs that print to the standard output
- Function Window : List of functions in the current source file
- Classes Window : List of classes in the current source file

3.2 Creating and running your first Console Application

At first we will type the source code

```
see "Hello, World!"
```

As in the next image

Then we will click on the "Save" button from the toolbar (or press CTRL+S)

Determine the source code file name and location.

For example type : hello

This will create a new source code file called : hello.ring

To run the program click on "Debug (Run then wait!)" button from the toolbar

The next screen shot present the application during the runtime

Press Enter to continue and return to the Ring Notepad.

3.3 Creating and running your first GUI/WebAssembly/Mobile Application

To learn how to create GUI applications using Ring check the "Desktop, WebAssembly and Mobile development using RingQt" chapter.

Source Code:

```
load "guilib.ring"

new qApp {
        new qWidget() {
                resize(400,400)
                setWindowTitle("Hello, World!")
                show()
        }
        exec()
}
```

In Ring notepad we have a special button to run GUI applications without displaying the console window.

The next screen shot present the application during the runtime

3.4 Creating and running your first Web Application

To learn how to support Ring in your web server and how to create web applications check the "Web Development (CGI Library)" chapter.

Note: You need to support the Ring language in your web server to be able to run the next example.

Tip: For Windows users, Ring comes with Apache Web server! Using Ring Notepad we can run any web application from any folder directly without doing any configuration.

Source Code:

```
#!ring -cgi

load "weblib.ring"

import System.Web

new Page {

        text("Hello, World!")

}
```

We can run the application in any web browser or in the browser that are embedded in Ring Notepad.

We can run the web application using the Web icon.

3.5 Creating and running your first Desktop/Mobile Game

To learn about creating 2D Games using Ring check the "Demo Project - Game Engine for 2D Games" chapter.

Source Code:

```
load "gameengine.ring"

func main

        oGame = New Game
        {
                title = "My First Game"
                sprite
                {
                        type = GE_TYPE_PLAYER
                        x=400 y=400 width=100 height=100
                        file = "images/player.png"
                        transparent = true
                        Animate=false
                        Move=true
```

(continues on next page)

(continued from previous page)

```
                        Scaled=true
                }
        }
```

We can run the application as any GUI application.

3.6 The Main File in the Project

The idea of the Main File ToolBar is to determine the main file in the project When the project contains many source code files

Using this feature we can run the project (Main File) at any time while opening other files in the project without the need to switch to the Main File to run the project.

To quickly use this feature

(Open the project main file)

Press Ctrl+Shift+M to set the current source code file as the main file

Open and modify other source code files in the project

To run the project (Main File) at any time press Ctrl+Shift+F5 (GUI) or Ctrl+Shift+D (Console)

Screen Shots:

3.7 The File Menu

From this menu we can create, open and save the source code files.

Another feature in this menu is "Print to PDF"

3.8 The Edit Menu

From the Edit menu we can Cut, Copy and Paste text.

Also we can change the font and the colors.

We can Go to a specific line or use the Find and Replace window to find and replace text.

Also We can set the Tab Width (Number of Spaces)

Starting from Ring 1.8 we have the (Find in Files) option.

Starting from Ring 1.11 we have the (Insert Text, Lower Case, Upper Case & Capitalize) options.

3.9 The View Menu

From this menu we can show/hide the dockable windows

Also we can change the Style of the Ring Notepad

Common Styles are (Fusion White and Modern Black)

Also we can quickly show/hide group of dockable windows based on the context

3.10 The Program Menu

From this menu we can run the programs

Also we can set the Main file in the project

3.11 The Browser Menu

From this menu we can quickly open common links in the browser

3.12 The Tools Menu

From this menu we can run the Form Designer in separate window

Also we can run the REPL (Read-Eval-Print-Loop) application

3.13 The Distribute Menu

From this menu we can build an executable file for the application

Also we can prepare the application for distribution

3.14 The Help Menu

From this menu we can get the help files (CHM & PDF)

CHAPTER

FOUR

GETTING STARTED - FIRST STYLE

4.1 Hello World

The next program prints the Hello World message on the screen (std-out).

```
see "Hello World"
```

4.2 Run the program

to run the program, save the code in a file, for example : hello.ring then from the command line or terminal, run it using Ring

```
ring hello.ring
```

4.3 Create Executable File

Using Ring2EXE we can create executable file for our application

```
ring2exe hello.ring -static
```

4.4 Not Case-Sensitive

Since the Ring language is not case-sensitive, the same program can be written in different styles

Tip: It's better to select one style and use it in all of the program source code

```
SEE "Hello World"
```

```
See "Hello World"
```

4.5 Multi-Line literals

Using Ring we can write multi-line literal, see the next example

```
See "
        Hello
        Welcome to the Ring programming language
        How are you?

    "
```

Also you can use the nl variable to insert new line and you can use the + operator to concatenate strings

As we have NL for new lines, we have Tab and CR (Carriage return) too!

Note: nl value means a new line and the actual codes that represent a newline is different between operating systems

```
See "Hello" + nl + "Welcome to the Ring programming language" +
    nl + "How are you?"
```

4.6 Getting Input

You can get the input from the user using the give command

```
See "What is your name? "
Give cName
See "Hello " + cName
```

4.7 No Explicit End For Statements

You don't need to use ';' or press ENTER to separate statements. The previous program can be written in one line.

```
See "What is your name? " give cName see "Hello " + cName
```

4.8 Using ? to print expression then new line

It's common to print new line after printing an expression, We can use the ? operator to do that!

Example:

```
? "Hello, World!"
for x = 1 to 10
        ? x
next
```

Output:

```
Hello, World!
1
2
3
4
5
6
7
8
9
10
```

4.9 Writing Comments

We can write one line comments and multi-line comments

The comment starts with # or //

Multi-lines comments are written between /* and */

```
/*
        Program Name : My first program using Ring
        Date         : 2016.09.09
        Author       : Mahmoud Fayed
*/

See "What is your name? "       # print message on screen
give cName                      # get input from the user
see "Hello " + cName            # say hello!

// See "Bye!"
```

Note: Using // to comment a lines of code is just a code style.

CHAPTER

FIVE

GETTING STARTED - SECOND STYLE

5.1 Hello World

The next program prints the Hello World message on the screen (std-out).

```
put "Hello World"
```

5.2 Run the program

to run the program, save the code in a file, for example : hello.ring then from the command line or terminal, run it using Ring

```
ring hello.ring
```

5.3 Create Executable File

Using Ring2EXE we can create executable file for our application

```
ring2exe hello.ring -static
```

5.4 Not Case-Sensitive

Since the Ring language is not case-sensitive, the same program can be written in different styles

Tip: It's better to select one style and use it in all of the program source code

```
PUT "Hello World"
```

```
Put "Hello World"
```

5.5 Multi-Line literals

Using Ring we can write multi-line literal, see the next example

```
Put "
        Hello
        Welcome to the Ring programming language
        How are you?

    "
```

Also you can use the nl variable to insert new line and you can use the + operator to concatenate strings

As we have NL for new lines, we have Tab and CR (Carriage return) too!

Note: nl value means a new line and the actual codes that represent a newline is different between operating systems

```
Put "Hello" + nl + "Welcome to the Ring programming language" +
    nl + "How are you?"
```

5.6 Getting Input

You can get the input from the user using the get command

```
Put "What is your name? "
Get cName
Put "Hello " + cName
```

5.7 No Explicit End For Statements

You don't need to use ';' or press ENTER to separate statements. The previous program can be written in one line.

```
Put "What is your name? " get cName put "Hello " + cName
```

5.8 Writing Comments

We can write one line comments and multi-line comments

The comment starts with # or //

Multi-lines comments are written between /* and */

```
/*
        Program Name : My first program using Ring
        Date         : 2016.09.09
        Author       : Mahmoud Fayed
*/
```

(continues on next page)

(continued from previous page)

```
Put "What is your name? "       # print message on screen
get cName                       # get input from the user
put "Hello " + cName            # say hello!

// Put "Bye!"
```

Note: Using // to comment a lines of code is just a code style.

CHAPTER

SIX

GETTING STARTED - THIRD STYLE

6.1 Hello World

The next program prints the Hello World message on the screen (std-out).

```
print("Hello, World!\n")
```

6.2 Run the program

to run the program, save the code in a file, for example : hello.ring then from the command line or terminal, run it using Ring

```
ring hello.ring
```

6.3 Create Executable File

Using Ring2EXE we can create executable file for our application

```
ring2exe hello.ring -static
```

The -static option will avoid the need to ring.dll|ring.so|ring.dylib

```
ring2exe hello.ring -dist -allruntime -noqt -noallegro
```

6.4 Not Case-Sensitive

Since the Ring language is not case-sensitive, the same program can be written in different styles

Tip: It's better to select one style and use it in all of the program source code

```
PRINT("Hello World")
```

```
Print("Hello World")
```

6.5 Multi-Line literals

Using Ring we can write multi-line literal, see the next example

```
Print("
        Hello
        Welcome to the Ring programming language
        How are you?

    ")
```

Also you can use the \n to insert new line and you can use #{variable_name} to insert variables values.

```
Print( "Hello\nWelcome to the Ring programming language\nHow are you?")
```

6.6 Getting Input

You can get the input from the user using the getstring() function

```
Print("What is your name? ")
cName = GetString()
Print("Hello #{cName}")
```

6.7 No Explicit End For Statements

You don't need to use ';' or press ENTER to separate statements. The previous program can be written in one line.

```
Print("What is your name? ") cName=getstring() print("Hello #{cName}")
```

6.8 Writing Comments

We can write one line comments and multi-line comments

The comment starts with # or //

Multi-lines comments are written between /* and */

```
/*
        Program Name : My first program using Ring
        Date         : 2016.09.09
        Author       : Mahmoud Fayed
*/

Print("What is your name? ")    # print message on screen
cName=GetString()               # get input from the user
print("Hello #{cName}")         # say hello!

// print("Bye!")
```

Note: Using // to comment a lines of code is just a code style.

6.9 Puts() function

print the value then print new line (nl)

Syntax:

```
puts(expr)
```

Example:

```
Puts("Hello, World!")
```

6.10 Print() function

print string - support \n,\t and \r

Also we can use #{variable_name} to insert variables values.

Syntax:

```
print(string) ---> String
```

Example:

```
print("\nHello, World\n\nHow are you? \t\t I'm fine!\n")
x=10 y=20
print("\nx value = #{x} , y value = #{y} \n")
```

6.11 Print2Str() Function

Syntax:

```
print2Str(string) ---> String
```

Example:

```
world = "World!"
mystring = print2str("Hello, #{world} \nIn Year \n#{2000+17} \n")

see mystring + nl
```

Output:

```
Hello, World!
In Year
2017
```

6.12 GetString() function

Get input from the keyboard - return value as string

```
getstring() ---> string
```

6.13 GetNumber() function

Get input from the keyboard - return value as number

```
getnumber() ---> number
```

CHAPTER

SEVEN

VARIABLES

To create a new variable, you just need to determine the variable name & value. The value will determine the variable type and you can change the value to switch between the types using the same variable name.

Syntax:

```
<Variable Name> = <Value>
```

Tip: The operator '=' is used here as an Assignment operator and the same operator can be used in conditions, but for testing equality of expressions.

Note: The Variable will contains the real value (not a reference). This means that once you change the variable value, the old value will be removed from memory (even if the variable contains a list or object).

7.1 Dynamic Typing

Ring is a dynamic programming language that uses Dynamic Typing.

```
x = "Hello"             # x is a string
see x + nl
x = 5                   # x is a number (int)
see x + nl
x = 1.2                 # x is a number (double)
see x + nl
x = [1,2,3,4]           # x is a list
see x                   # print list items
x = date()              # x is a string contains date
see x + nl
x = time()              # x is a string contains time
see x + nl
x = true                # x is a number (logical value = 1)
see x + nl
x = false               # x is a number (logical value = 0)
see x + nl
```

7.2 Deep Copy

We can use the assignment operator '=' to copy variables. We can do that to copy values like strings & numbers. Also, we can copy complete lists & objects. The assignment operator will do a complete duplication for us. This operation called Deep Copy

```
list = [1,2,3,"four","five"]
list2 = list
list = []
See list        # print the first list - no items to print
See "********" + nl
See list2       # print the second list - contains 5 items
```

Note: To copy lists/objects by reference, See the (Using References) chapter about the Ref() function

7.3 Implicit Conversion

The language can automatically convert between numbers and strings.

Rules:

```
<NUMBER> + <STRING> --> <NUMBER>
<STRING> + <NUMBER> --> <STRING>
```

Note: The same operator '+' can be used as an arithmetic operator or for string concatenation.

Example:

```
x = 10                  # x is a number
y = "20"                # y is a string
sum = x + y             # sum is a number (y will be converted to a number)
Msg = "Sum = " + sum    # Msg is a string (sum will be converted to a string)
see Msg + nl
```

CHAPTER

EIGHT

OPERATORS

In this chapter we will introduce the operators provided by the Ring programming language.

8.1 Arithmetic Operators

The next table presents all of the arithmetic operators provided by the Ring language. Assume variable X=50 and variable Y=10 then:

Operator	Description	Example	Result
+	Add	x+y	60
-	Subtract	x-y	40
*	Multiplies	x*y	500
/	Divide	x/y	5
%	Modulus	x%y	0
++	Increment	x++	51
- -	Decrement	x- -	49
** OR ^^	Power	x**3	125000

8.2 Relational Operators

The next table presents all of the relational operators provided by the Ring language. Assume variable X=50 and variable Y=10 then:

Operator	Description	Example	Result
=	Equal	x = y	False
!=	Not Equal	x != y	True
>	Greater than	x > y	True
<	Less than	x < y	False
>=	Greater or Equal	x >= y	True
<=	Less than or Equal	x <= y	False

8.3 Logical Operators

The next table presents all of the logical operators provided by the Ring language. Assume variable X=True and variable Y=False then:

Operator	Description	Example	Result
and	Logical AND	x and y	False
or	Logical OR	x or y	True
not	Logical Not	not x	False

Another style

Operator	Description	Example	Result
&&	Logical AND	x && y	False
||	Logical OR	x || y	True
!	Logical Not	! x	False

8.4 Bitwise Operators

The next table presents all of the bitwise operators provided by the Ring language. Assume variable X=8 and variable Y=2 then:

Operator	Description	Example	Result
&	Binary AND	x & y	0
|	Binary OR	x | y	10
^	Binary XOR	x ^ y	10
~	Binary Ones Complement	~x	-9
<<	Binary Left Shift	x << y	32
>>	Binary Right Shift	x >> y	2

8.5 Assignment Operators

The next table presents all of the assignment operators provided by the Ring language.

Assume variable X=8 then:

Operator	Description	Example	Result
=	Assignment	x = 10	x=10
+=	Add AND assignment	x += 5	x=13
-=	Subtract AND assignment	x -= 3	x=5
*=	Multiply AND assignment	x *= 2	x=16
/=	Divide AND assignment	x /= 3	x=2.67
%=	Modulus AND assignment	x %= 2	x=0
<<=	Left shift AND assignment	x <<= 2	x=32
>>=	Right shift AND assignment	x >>= 2	x=2
&=	Bitwise AND assignment	x &= 4	x=0
|=	Bitwise OR and assignment	x |= 3	x=11
^=	Bitwise XOR and assignment	x ^= 4	x=12

8.6 Misc Operators

Operator	Description
:literal	using : before identifier mean literal
Start:End	create list contains items from start to end
[list items]	define list items
list[index]	access list item
obj.name	using the dot operator to access object members (attributes/methods).
obj {stmts}	execute statements with direct access to object attributes & methods
func(para,...)	call function using parameters separated by comma
? <expr>	Print expression then new line

8.7 Operators Precedence

The next table present operators from higher precedence (Evaluated first) to lower precedence.

Operator
. [] () {}
- ~ :Literal [list items]
++ - -
Start:End
* / %
+ -
<< >>
&
| ^
< > <= >=
= !=
not !
and or && ||
Assignment = += -= *= /= %=>>= <<= &= ^= |=
?

Example:

```
See 3+5*4       # prints 23
```

CHAPTER

NINE

CONTROL STRUCTURES - FIRST STYLE

In this chapter we are going to learn about the control structures provided by the Ring programming language.

9.1 Branching

- If Statement

Syntax:

```
if Expression
        Block of statements
but Expression
        Block of statements
else
        Block of statements
ok
```

Example:

```
see "
        Main Menu
        ---------
        (1) Say Hello
        (2) About
        (3) Exit

    " give nOption

if nOption = 1  see "Enter your name : " give name see "Hello " + name + nl
but nOption = 2 see "Sample : using if statement" + nl
but nOption = 3 bye
else see "bad option..." + nl
ok
```

- Switch Statement

Syntax:

```
switch Expression
on Expression
        Block of statements
```

(continues on next page)

(continued from previous page)

```
other
        Block of statements
off
```

Example:

```
See "
        Main Menu
        ---------
        (1) Say Hello
        (2) About
        (3) Exit

    " Give nOption

Switch nOption
On 1 See "Enter your name : " Give name See "Hello " + name + nl
On 2 See "Sample : using switch statement" + nl
On 3 Bye
Other See "bad option..." + nl
Off
```

9.2 Looping

- While Loop

Syntax:

```
while Expression
        Block of statements
end
```

Example:

```
While True

        See "
                Main Menu
                ---------
                (1) Say Hello
                (2) About
                (3) Exit

            " Give nOption

        Switch nOption
        On 1
                See "Enter your name : "
                Give name
                See "Hello " + name + nl
        On 2
```

(continues on next page)

(continued from previous page)

```
                See "Sample : using while loop" + nl
        On 3
                Bye
        Other
                See "bad option..." + nl
        Off
End
```

- For Loop

Syntax:

```
for identifier=expression to expression [step expression]
        Block of statements
next
```

Example:

```
# print numbers from 1 to 10
for x = 1 to 10  see x + nl  next
```

Example:

```
# Dynamic loop
See "Start : " give nStart
See "End   : " give nEnd
See "Step  : " give nStep
For x = nStart to nEnd Step nStep
        see x + nl
Next
```

Example:

```
# print even numbers from 0 to 10
for x = 0 to 10 step 2
        see x + nl
next
```

Example:

```
# print even numbers from 10 to 0
for x = 10 to 0 step -2
        see x + nl
next
```

- For in Loop

Syntax:

```
for identifier in List/String  [step expression]
        Block of statements
next
```

Example:

```
aList = 1:10    # create list contains numbers from 1 to 10
for x in aList  see x + nl  next  # print numbers from 1 to 10
```

9.3 Using The Step option with For in

We can use the Step option with For in to skip number of items in each iteration

Example:

```
aList = 1:10    # create list contains numbers from 1 to 10
# print odd items inside the list
for x in aList step 2
        see x + nl
next
```

9.4 Using For in to modify lists

When we use (For in) we get items by reference.

This means that we can read/edit items inside the loop.

Example:

```
aList = 1:5     # create list contains numbers from 1 to 5
# replace list numbers with strings
for x in aList
        switch x
        on 1  x = "one"
        on 2  x = "two"
        on 3  x = "three"
        on 4  x = "four"
        on 5  x = "five"
        off
next
see aList       # print the list items
```

9.5 Do Again Loop

Syntax:

```
do
        Block of statements
again expression
```

Example:

```
x = 1
do
```

(continues on next page)

(continued from previous page)

```
        see x + nl
        x++
again x <= 10
```

9.6 Exit Command

Used to go outside one or more of loops.

Syntax:

```
exit [expression]       # inside loop
```

Example:

```
for x = 1 to 10
        see x + nl
        if x = 5 exit ok
next
```

9.7 Exit from two loops

The next example presents how to use the exit command to exit from two loops in one jump.

Example:

```
for x = 1 to 10
        for y = 1 to 10
                see "x=" + x + " y=" + y + nl
                if x = 3 and y = 5
                        exit 2     # exit from 2 loops
                ok
        next
next
```

9.8 Loop Command

Used to jump to the next iteration in the loop.

Syntax:

```
loop [expression]       # inside loop
```

Example:

```
for x = 1 to 10
        if x = 3
                see "Number Three" + nl
                loop
```

(continues on next page)

(continued from previous page)

```
        ok
        see x + nl
next
```

9.9 Short-circuit evaluation

The logical operators and/or follow the short-circuit evaluation.

If the first argument of the AND operator is zero, then there is no need to evaluate the second argument and the result will be zero.

If the first argument of the OR operator is one, then there is no need to evaluate the second argument and the result will be one.

Example:

```
/* output
** nice
** nice
** great
*/

x = 0 y = 10

if (x = 0 and nice()) and (y = 10 and nice())
        see "great" + nl
ok

func nice  see "nice" + nl   return 1
```

Example:

```
# No output

x = 0 y = 10

if (x = 1 and nice()) and (y = 10 and nice())
        see "great" + nl
ok

func nice  see "nice" + nl   return 1
```

Example:

```
/* output
** nice
** great
*/

x = 0 y = 10

if (x = 0 and nice()) or (y = 10 and nice())
```

(continues on next page)

(continued from previous page)

```
        see "great" + nl
ok

func nice  see "nice" + nl  return 1
```

9.10 Comments about evaluation

- True, False, nl & NULL are variables defined by the language
- True = 1
- False = 0
- nl = new line
- NULL = empty string = “”
- Everything evaluates to true except 0 (False), NULL (Empty String) and Empty List.

Example:

```
# output = message from the if statement

if 5    # 5 evaluates to true because it's not zero (0).
        see "message from the if statement" + nl
ok
```

CHAPTER

TEN

CONTROL STRUCTURES - SECOND STYLE

In this chapter we are going to learn about the second style of control structures provided by the Ring programming language.

10.1 Branching

- If Statement

Syntax:

```
if Expression
        Block of statements
elseif Expression
        Block of statements
else
        Block of statements
end
```

Example:

```
put "
        Main Menu
        ---------
        (1) Say Hello
        (2) About
        (3) Exit

    " get nOption

if nOption = 1  put "Enter your name : " get name put "Hello " + name + nl
elseif nOption = 2 put "Sample : using if statement" + nl
elseif nOption = 3 bye
else put "bad option..." + nl
end
```

- Switch Statement

Syntax:

```
switch Expression
case Expression
```

(continues on next page)

(continued from previous page)

```
        Block of statements
else
        Block of statements
end
```

Example:

```
Put "
        Main Menu
        ---------
        (1) Say Hello
        (2) About
        (3) Exit

    " Get nOption

Switch nOption
Case 1 Put "Enter your name : " Get name Put "Hello " + name + nl
Case 2 Put "Sample : using switch statement" + nl
Case 3 Bye
Else Put "bad option..." + nl
End
```

10.2 Looping

- While Loop

Syntax:

```
while Expression
        Block of statements
end
```

Example:

```
While True

        Put "
                Main Menu
                ---------
                (1) Say Hello
                (2) About
                (3) Exit

            " Get nOption

        Switch nOption
        Case 1
                Put "Enter your name : "
                Get name
                Put "Hello " + name + nl
```

(continues on next page)

(continued from previous page)

```
        Case 2
                Put "Sample : using while loop" + nl
        Case 3
                Bye
        Else
                Put "bad option..." + nl
        End
End
```

- For Loop

Syntax:

```
for identifier=expression to expression [step expression]
        Block of statements
end
```

Example:

```
# print numbers from 1 to 10
for x = 1 to 10  put x + nl  end
```

Example:

```
# Dynamic loop
Put "Start : " get nStart
Put "End   : " get nEnd
Put "Step  : " get nStep
For x = nStart to nEnd Step nStep
        Put x + nl
End
```

Example:

```
# print even numbers from 0 to 10
for x = 0 to 10 step 2
        Put x + nl
end
```

Example:

```
# print even numbers from 10 to 0
for x = 10 to 0 step -2
        put x + nl
end
```

- For in Loop

Syntax:

```
for identifier in List/String  [step expression]
        Block of statements
end
```

Example:

```
aList = 1:10    # create list contains numbers from 1 to 10
for x in aList  put x + nl  end  # print numbers from 1 to 10
```

10.3 Exceptions

```
try
        Block of statements
catch
        Block of statements
end
```

CHAPTER

ELEVEN

CONTROL STRUCTURES - THIRD STYLE

In this chapter we are going to learn about the third style of control structures provided by the Ring programming language.

11.1 Branching

- If Statement

Syntax:

```
if Expression {
        Block of statements
elseif Expression
        Block of statements
else
        Block of statements
}
```

Example:

```
print("
        Main Menu
        ---------
        (1) Say Hello
        (2) About
        (3) Exit
    ")

nOption = getnumber()

if nOption = 1  {
        print("Enter your name : ")
        name = getstring()
        print("Hello #{name}\n")
elseif nOption = 2
        print("Sample : using if statement\n")
elseif nOption = 3
        bye
else
        print("bad option...\n")
}
```

- Switch Statement

Syntax:

```
switch Expression {
case Expression
        Block of statements
else
        Block of statements
}
```

Example:

```
print("
        Main Menu
        ---------
        (1) Say Hello
        (2) About
        (3) Exit

      ")

nOption = GetString()

switch nOption {
case 1
        print("Enter your name : ")
        name = getstring()
        print("Hello #{name}\n")
case 2
        print("Sample : using switch statement\n")
case 3
        Bye
else
        print("bad option...\n")
}
```

11.2 Looping

- While Loop

Syntax:

```
while Expression {
        Block of statements
}
```

Example:

```
While True {

        print("
                Main Menu
```

(continues on next page)

(continued from previous page)

```
                ---------
                (1) Say Hello
                (2) About
                (3) Exit

                  ")

        nOption = GetString()

        switch nOption {
        case 1
                print("Enter your name : ")
                name = getstring()
                print("Hello #{name}\n")
        case 2
                print("Sample : using switch statement\n")
        case 3
                Bye
        else
                print("bad option...\n")
        }

}
```

- For Loop

Syntax:

```
for identifier=expression to expression [step expression] {
        Block of statements
}
```

Example:

```
# print numbers from 1 to 10
for x = 1 to 10  {
        print("#{x}\n")
}
```

Example:

```
# Dynamic loop
print("Start : ") nStart = getnumber()
print("End   : ") nEnd = getnumber()
print("Step  : ") nStep = getnumber()
for x = nStart to nEnd step nStep {
        print("#{x}\n")
}
```

Example:

```
# print even numbers from 0 to 10
for x = 0 to 10 step 2 {
```

(continues on next page)

(continued from previous page)

```
        print("#{x}\n")
}
```

Example:

```
# print even numbers from 10 to 0
for x = 10 to 0 step -2 {
        print("#{x}\n")
}
```

- For in Loop

Syntax:

```
for identifier in List/String  [step expression] {
        Block of statements
}
```

Example:

```
aList = 1:10    # create list contains numbers from 1 to 10
for x in aList { print("#{x}\n") }  # print numbers from 1 to 10
```

Example:

```
aList = 1:10    # create list contains numbers from 1 to 10
# print odd items inside the list
for x in aList step 2 {
        print("#{x}\n")
}
```

When we use (For in) we get items by reference.

This means that we can read/edit items inside the loop.

Example:

```
aList = 1:5     # create list contains numbers from 1 to 5
# replace list numbers with strings
for x in aList {
        switch x {
        case 1  x = "one"
        case 2  x = "two"
        case 3  x = "three"
        case 4  x = "four"
        case 5  x = "five"
        }
}
print(aList)    # print the list items
```

11.3 Exceptions

```
try {
        Block of statements
catch
        Block of statements
}
```

CHAPTER

TWELVE

GETTING INPUT

We can get input from the keyboard using

- The Give Command
- The GetChar() Function
- The Input() Function

12.1 Give Command

Syntax:

```
Give VariableName
```

Example:

```
See "Enter the first number : " Give nNum1
See "Enter the second number : " Give nNum2
See "Sum : " + ( 0 + nNum1 + nNum2 )
```

Output:

```
Enter the first number : 3
Enter the second number : 4
Sum : 7
```

12.2 GetChar() Function

We can get one character from the standard input using the GetChar() function

Syntax:

```
GetChar() ---> Character
```

Example:

```
While True
        See "
                Main Menu
```

(continues on next page)

(continued from previous page)

```
                (1) Say Hello
                (2) Exit
            "
        Option = GetChar()
        GetChar() GetChar()  # End of line

        # the previous two lines can be replaced with the next line
        # Give Option

        if Option = 1
                see "Enter your name : " give cName
                see "Hello " + cName
        else
                bye
        ok
End
```

12.3 Input() Function

We can get input from the keyboard using the Input() function

Syntax:

```
Input(nCount) ---> string
```

The function will wait until nCount characters (at least) are read

Example:

```
See "Enter message (30 characters) : " cMsg = input(30)
See "Message : " + cMsg
```

CHAPTER

THIRTEEN

FUNCTIONS - FIRST STYLE

In this chapter we are going to learn about the next topics :-

- Define functions
- Call functions
- Declare parameters
- Send parameters
- Main Function
- Variables Scope
- Return Value
- Recursion

13.1 Define Functions

To define new function

Syntax:

```
func <function_name> [parameters]
        Block of statements
```

Note: No keyword is required to end the function definition.

Example:

```
func hello
        see "Hello from function" + nl
```

13.2 Call Functions

To call function without parameters, we type the function name then ()

Tip: We can call the function before the function definition and the function code.

Example:

```
hello()

func hello
        see "Hello from function" + nl
```

Example:

```
first()  second()

func first   see "message from the first function" + nl

func second  see "message from the second function" + nl
```

13.3 Declare parameters

To declare the function parameters, after the function name type the list of parameters as a group of identifiers separated by comma.

Example:

```
func sum x,y
        see x+y+nl
```

13.4 Send Parameters

To send parameters to function, type the parameters inside () after the function name

Syntax:

```
funcname(parameters)
```

Example:

```
/* output
** 8
** 3000
*/

sum(3,5) sum(1000,2000)

func sum x,y see x+y+nl
```

13.5 Main Function

Using the Ring programming language, the Main Function is optional, when it's defined, it will be executed after the end of other statements.

if no other statements comes alone, the main function will be the first entry point

Example:

```
# this program will print the hello world message first then execute the main function

See "Hello World!" + nl

func main
        see "Message from the main function" + nl
```

13.6 Variables Scope

The Ring programming language uses lexical scoping to determine the scope of a variable.

Variables defined inside functions (including function parameters) are local variables. Variables defined outside functions (before any function) are global variables.

Inside any function we can access the variables defined inside this function beside the global variables.

Example:

```
# the program will print numbers from 10 to 1

x = 10                          # x is a global variable.

func main

        for t = 1 to 10         # t is a local variable
                mycounter()     # call function
        next

func mycounter

        see x + nl              # print the global variable value
        x--                     # decrement
```

Note: Using the main function before the for loop declare the t variable as a local variable, It's recommended to use the main functions instead of typing the instructions directly to set the scope of the new variables to local.

13.7 Return Value

The function can return a value using the Return command.

Syntax:

```
Return [Expression]
```

Tip: the Expression after the return command is optional and we can use the return command to end the function execution without returning any value.

Note: if the function doesn't return explicit value, it will return NULL (empty string = "").

Example:

```
if novalue() = NULL
        See "the function doesn't return a value" + nl
ok

func novalue
```

13.8 Recursion

The Ring programming language support Recursion and the function can call itself using different parameters.

Example:

```
see fact(5)     # output = 120

func fact x if x = 0 return 1 else return x * fact(x-1) ok
```

CHAPTER

FOURTEEN

FUNCTIONS - SECOND STYLE

In this chapter we are going to learn about the next topics :-

- Define functions
- Call functions
- Declare parameters
- Send parameters
- Main Function
- Variables Scope
- Return Value
- Recursion

14.1 Define Functions

To define new function

Syntax:

```
def <function_name> [parameters]
        Block of statements
[end]
```

Note: the keyword 'end' is optional.

Example:

```
def hello
        put "Hello from function" + nl
end
```

14.2 Call Functions

To call function without parameters, we type the function name then ()

Tip: We can call the function before the function definition and the function code.

Example:

```
hello()

def hello
        put "Hello from function" + nl
end
```

Example:

```
first()  second()

def first   put "message from the first function" + nl

def second  put "message from the second function" + nl
```

14.3 Declare parameters

To declare the function parameters, after the function name type the list of parameters as a group of identifiers separated by comma.

Example:

```
def sum x,y
        put x+y+nl
end
```

14.4 Send Parameters

To send parameters to function, type the parameters inside () after the function name

Syntax:

```
funcname(parameters)
```

Example:

```
/* output
** 8
** 3000
*/

sum(3,5) sum(1000,2000)
```

(continues on next page)

(continued from previous page)

```
def sum x,y put x+y+nl
```

14.5 Main Function

Using the Ring programming language, the Main Function is optional, when it's defined, it will be executed after the end of other statements.

if no other statements comes alone, the main function will be the first entry point

Example:

```
# this program will print the hello world message first then execute the main function

put "Hello World!" + nl

def main
        put "Message from the main function" + nl
end
```

14.6 Variables Scope

The Ring programming language uses lexical scoping to determine the scope of a variable.

Variables defined inside functions (including function parameters) are local variables. Variables defined outside functions (before any function) are global variables.

Inside any function we can access the variables defined inside this function beside the global variables.

Example:

```
# the program will print numbers from 10 to 1

x = 10                          # x is a global variable.

def main
        for t = 1 to 10         # t is a local variable
                mycounter()     # call function
        end
end

def mycounter
        put x + nl              # print the global variable value
        x--                     # decrement
end
```

Note: Using the main function before the for loop declare the t variable as a local variable, It's recommended to use the main functions instead of typing the instructions directly to set the scope of the new variables to local.

14.7 Return Value

The function can return a value using the Return command.

Syntax:

```
Return [Expression]
```

Tip: the Expression after the return command is optional and we can use the return command to end the function execution without returning any value.

Note: if the function doesn't return explicit value, it will return NULL (empty string = "").

Example:

```
if novalue() = NULL
        put "the function doesn't return a value" + nl
end

def novalue
```

14.8 Recursion

The Ring programming language support Recursion and the function can call itself using different parameters.

Example:

```
put fact(5)     # output = 120

def fact x if x = 0 return 1 else return x * fact(x-1) end
```

CHAPTER

FIFTEEN

FUNCTIONS - THIRD STYLE

In this chapter we are going to learn about the next topics :-

- Define functions
- Call functions
- Declare parameters
- Send parameters
- Main Function
- Variables Scope
- Return Value
- Recursion

15.1 Define Functions

To define new function

Syntax:

```
func <function_name> [parameters] ['{']
        Block of statements
['}']
```

Example:

```
func hello {
        print("Hello from function \n")
}
```

15.2 Call Functions

To call function without parameters, we type the function name then ()

Tip: We can call the function before the function definition and the function code.

Example:

```
hello()

func hello {
        print("Hello from function \n")
}
```

Example:

```
first()  second()

func first {  print("message from the first function \n")  }

func second { print("message from the second function \n") }
```

15.3 Declare parameters

To declare the function parameters, after the function name type the list of parameters as a group of identifiers separated by comma.

Example:

```
func sum(x,y) {
        print(x+y)
}
```

15.4 Send Parameters

To send parameters to function, type the parameters inside () after the function name

Syntax:

```
funcname(parameters)
```

Example:

```
/* output
** 8
** 3000
*/

sum(3,5) sum(1000,2000)
```

(continues on next page)

(continued from previous page)

```
func sum(x,y) { print(x+y) }
```

15.5 Main Function

Using the Ring programming language, the Main Function is optional, when it's defined, it will be executed after the end of other statements.

if no other statements comes alone, the main function will be the first entry point

Example:

```
# this program will print the hello world message first then execute the main function

print("Hello, World! \n")

func main {
        print("Message from the main function \n")
}
```

15.6 Variables Scope

The Ring programming language uses lexical scoping to determine the scope of a variable.

Variables defined inside functions (including function parameters) are local variables. Variables defined outside functions (before any function) are global variables.

Inside any function we can access the variables defined inside this function beside the global variables.

Example:

```
# the program will print numbers from 10 to 1

x = 10                          # x is a global variable.

func main {
        for t = 1 to 10 {       # t is a local variable
                mycounter()     # call function
        }
}

func mycounter {
        print("#{x}\n")         # print the global variable value
        x--                     # decrement
}
```

Note: Using the main function before the for loop declare the t variable as a local variable, It's recommended to use the main functions instead of typing the instructions directly to set the scope of the new variables to local.

15.7 Return Value

The function can return a value using the Return command.

Syntax:

```
Return [Expression]
```

Tip: the Expression after the return command is optional and we can use the return command to end the function execution without returning any value.

Note: if the function doesn't return explicit value, it will return NULL (empty string = "").

Example:

```
if novalue() = NULL {
        print("the function doesn't return a value\n")
}

func novalue { }
```

15.8 Recursion

The Ring programming language support Recursion and the function can call itself using different parameters.

Example:

```
print( fact(5) )        # output = 120

func fact(x) { if x = 0 { return 1 else return x * fact(x-1) } }
```

CHAPTER

SIXTEEN

PROGRAM STRUCTURE

In this chapter we will learn about using many source code files in the same project.

16.1 Source Code File Sections

Each source code file may contains the next sections (in the same order).

Source Code File Sections
Load Files
Statements and Global Variables
Functions
Packages and Classes

The application maybe one or more of files.

16.2 Using Many Source Code Files

To include another source file in the project, just use the load command.

Syntax:

```
Load  "filename.ring"
```

Note: The Load command is executed directly by the compiler in the parsing stage

Tip: if you don't know the file name until the runtime, or you need to use functions to get the file path, just use eval().

Example:

```
# File : Start.ring

Load "sub.ring"

sayhello("Mahmoud")
```

```
# File : sub.ring

func sayhello cName
        see "Hello " + cName + nl
```

16.3 Load Package

Using the 'load' command we can use many ring source files in the same project

But all of these files will share the same global scope

We have also the "Load Package" command

Using "Load Package" we can load a library (*.ring file) in new global scope

This is very useful to create libraries that avoid conflicts in global variables

Example:

File: loadpackage.ring

```
x = 100
? "Hello, World!"
load package "testloadpackage.ring"

? x
test()
```

File: testloadpackage.ring

```
? "Hello from testloadpackage.ring"

x = 1000

test()

func test
        ? x
```

Output:

```
Hello, World!
Hello from testloadpackage.ring
1000
100
1000
```

16.4 Load Again

Ring 1.12 comes with the Load Again command

Using this command we can load the Ring source file which contains constants more than one time.

This is useful when using Ring source files for translations through global constants.

Example:

The next function is part from a project which support Arabic and English languages

The files english.ring and arabic.ring contains constants for translation

One of these files is loaded in the start of the program

Loading the same file again using the (Load) command is not possible

Because the (Load) command load the same source file only for the first time and ignore next times.

So we have to use the (Load Again) command.

Where we can use these files again during the runtime as in the next code

```
func setLang nLanguage
        if C_ENV_DEFAULT_LANG = nLanguage
                return
        ok
        C_ENV_DEFAULT_LANG = nLanguage
        # Change the language
                switch nLanguage
                        on C_TRANSLATION_ENGLISH
                                load again "translation/english.ring"
                        on C_TRANSLATION_ARABIC
                                load again "translation/arabic.ring"
                off
```

CHAPTER

SEVENTEEN

LISTS

In this chapter we are going to learn how to deal with lists.

17.1 Create Lists

We can create new lists by defining the list items inside square brackets.

Example:

```
aList = [1,2,3,4,5]
```

Also we can create new lists using the : operator

Example:

```
aList = 1:5
aList2 = "a":"z"
```

Example:

```
aList = 5:1
aList2 = "z":"a"
```

Also we can create lists using the list() function

Syntax:

```
list = list(size)
```

To create 2D list

```
list = list(nRows,nCols)
```

Example (1)

```
aList = list(10)        # aList contains 10 items
```

Example (2)

```
aList = list(5,4)       # Create 2D List contains 5 rows and 4 columns
```

Note: the list index start from 1

17.2 Add Items

To add new items to the list, we can use the Add() function.

Syntax:

```
Add(List,Item)
```

Example:

```
aList = ["one","two"]
add(aList,"three")
see aList
```

Also we can do that using the + operator.

Syntax:

```
List + item
```

Example:

```
aList = 1:10    # create list contains numbers from 1 to 10
aList + 11      # add number 11 to the list
see aList       # print the list
```

17.3 Get List Size

We can get the list size using the len() function

Syntax:

```
Len(List)
```

Example:

```
aList = 1:20  see len(aList)  # print 20
```

17.4 Delete Item From List

To delete an item from the list, we can use the del() function

Syntax:

```
del(list,index)
```

Example:

```
aList = ["one","two","other","three"]
Del(aList,3)    # delete item number three
see aList       # print one two three
```

17.5 Get List Item

To get an item from the list, we uses the next syntax

```
List[Index]
```

Example:

```
aList = ["Cairo","Riyadh"]
see "Egypt : " + aList[1] + nl +
    "KSA   : " + aList[2] + nl
```

17.6 Set List Item

To set the value of an item inside the list, we can use the next syntax

```
List[Index] = Expression
```

Example:

```
aList = list(3) # create list contains three items
aList[1] = "one" aList[2] = "two" aList[3] = "three"
see aList
```

17.7 Search

To find an item inside the list we can use the find() function

Syntax:

```
Find(List,ItemValue) ---> Item Index
Find(List,ItemValue,nColumn) ---> Search in nColumn, returns the Item Index
Find(List,ItemValue,nColumn,cAttribute) ---> Item Index
```

Example:

```
aList = ["one","two","three","four","five"]
see find(aList,"three")         # print 3
```

Example:

```
mylist = [["one",1],
          ["two",2],
          ["three",3]]

see find(mylist,"two",1) + nl           # print 2
see find(mylist,2,2) + nl               # print 2
```

Also we can use the binarysearch() function to search in sorted list.

Syntax:

```
BinarySearch(List,ItemValue) ---> Item Index
BinarySearch(List,ItemValue,nColumn) ---> Search in nColumn, returns the Item Index
```

Example:

```
aList = ["one","two","three","four","five"]
aList = sort(aList)
see binarysearch(aList,"three")
```

Output:

```
five
four
one
three
two
4
```

17.8 Sort

We can sort the list using the sort() function.

Syntax:

```
Sort(List) ---> Sorted List
Sort(List,nColumn) ---> Sorted List based on nColumn
Sort(List,nColumn,cAttribute) ---> Sorted List based on Object Attribute
```

Example:

```
aList = [10,12,3,5,31,15]
aList = sort(aList) see aList # print 3 5 10 12 15 31
```

We can sort list of strings

Example:

```
mylist = ["mahmoud","samir","ahmed","ibrahim","mohammed"]
see mylist                # print list before sorting
mylist = sort(mylist)     # sort list
see "list after sort"+nl
see mylist                # print ahmed ibrahim mahmoud mohammed samir
```

We can sort a list based on a specific column.

Example:

```
aList = [ ["mahmoud",15000] ,
          ["ahmed", 14000 ] ,
          ["samir", 16000 ] ,
          ["mohammed", 12000 ] ,
          ["ibrahim",11000 ] ]
```

(continues on next page)

(continued from previous page)

```
aList2 = sort(aList,1)
see aList2
```

Output:

```
ahmed
14000
ibrahim
11000
mahmoud
15000
mohammed
12000
samir
16000
```

17.9 Reverse

We can reverse a list using the reverse() function.

Syntax:

```
Reverse(List) ---> Reversed List
```

Example:

```
aList = [10,20,30,40,50]
aList = reverse(aList)
see aList       # print 50 40 30 20 10
```

17.10 Insert Items

To insert an item in the list we can use the insert() function.

Syntax:

```
Insert(List,Index,Item)
```

The inserted item will be AFTER the Index

Example:

```
aList = ["A","B","D","E"]
insert(aList,2,"C")    # Inserts AFTER Index 2, "C" into Position 3
see aList              # print A B C D E
```

17.11 Nested Lists

The list may contain other lists

Example:

```
aList = [ 1 , [10,20,30] , 5 , [100,1000,5000] ]
aList2 = [
"one","two",
[3,4],
[20,30], ["three",
          "four",
          "five",[100,200,300]
         ]
]

see aList[2]            # print 10 20 30
see aList[4][3] + nl    # print 5000
see aList2[5][2] + nl   # print four
see aList2[5][4][3]     # print 300
```

17.12 Copy Lists

We can copy lists (including nested lists) using the Assignment operator.

Example:

```
aList = [
"one","two",
[3,4],
[20,30], ["three",
          "four",
          "five",[100,200,300]
         ]
]

aList2 = aList          # Copy aList to aList2
aList2[5] = "other"     # modify item number five
see aList2[5] + nl      # print other
see aList[5]            # print three four five 100 200 300
```

17.13 First-class lists

Lists are first-class citizens where we can store lists in variables, pass lists to functions, and return lists from functions.

Example:

```
aList = duplicate( [1,2,3,4,5] )
see aList[10] + nl                # print 5
```

(continues on next page)

(continued from previous page)

```
see mylist()                        # print 10 20 30 40 50

func duplicate list
        nMax = len(list)
        for x = 1 to nMax
                list + list[x]
        next
        return list

func mylist return [10,20,30,40,50]
```

17.14 Using Lists during definition

We can use the list and the list items while we are defining the list for the first time.

Example:

```
aList = [ [1,2,3,4,5] , aList[1] , aList[1] ]
see aList       # print 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

Example:

```
x = [ 1, 2, x ]
? x             # print 1 2 1 2
? len(x)        # print 3
? x[1]          # print 1
? x[2]          # print 2
? x[3]          # print 1 2
```

Output:

```
1
2
1
2

3
1
2
1
2
```

17.15 Passing Lists to Functions

Lists are passed to functions by reference, This means that the called function will work on the same list and can modify it.

Example:

```
func main
        aList = [1,2,3,4,5]     # create list, local in function main
        myfunc(aList)           # call function, pass list by reference
        see aList               # print 1 2 3 4 5 6 7 8 9 10

func myfunc list
        list + [6,7,8,9,10]
```

17.16 Access List Items by String Index

Instead of using numbers to determine the item index when we get item value or set item value, We can access items using string index if the item is a list contains two items and the first item is a string.

Example:

```
aList = [ ["one",1] , ["two",2] , ["three",3] ]
see aList["one"] + nl +
    aList["two"] + nl +
    aList["three"]      # print 1 2 3
```

This type of lists can be defined in a better syntax using the : and = operators.

Example:

```
aList = [ :one = 1 , :two = 2 , :three = 3 ]
see aList["one"] + nl +
    aList["two"] + nl +
    aList["three"] + nl # print 1 2 3
see aList[1]            # print one 1
```

Tip: using : before identifier (one word) means literal

Note: using = inside list definition create a list of two items where the first item is the left side and the second item is the right side.

We can add new items to the list using the string index

Example:

```
aList = []
aList["Egypt"] = "Cairo"
aList["KSA"] = "Riyadh"
see aList["Egypt"] + nl +       # print Cairo
    aList["KSA"] + nl           # print Riyadh
```

17.17 Passing Parameters or Arguments Using List

This type of lists is very good for passing parameters to functions Where the order of parameters will not be important (we can change the order).

Also some parameters maybe optional.

Example:

```
myconnect (  [ :server = "myserver.com" , :port = 80 ,
               :username = "mahmoud" , :password = "password" ] )

func myconnect mypara

        # print connection details
        see "User Name : " + mypara[:username] + nl +
            "Password  : " + mypara[:password] + nl +
            "Server    : " + mypara[:server] + nl +
            "Port      : " + mypara[:port]
```

17.18 Passing Parameters or Arguments Using List Array

Passing Arguments or Parameters to a Function in an array format

Example:

```
myList = [5,7,3,9]    ### list with args or parms in  an array
result = sum(myList)
See "Sum result: "+ result +n

func sum(aList)
  acc = 0
  sizeList = len(aList)

  for i = 1 to sizeList
     See aList[i] +nl
     acc = acc + aList[i]
  next
return acc
```

17.19 Return Parameters as List or Hash Table

Return Parameters from a Function in an Array or Hash Format

Example:

```
sudoku = [  [2,9,0],
            [0,0,1],
            [0,0,0] ]

aOutput = myFunctionArray(sudoku)
```

(continues on next page)

(continued from previous page)

```
        See "Return Array: T/F: "+ aOutput[1] +" Row: "+ aOutput[2] +" Col: "+␣
↪aOutput[3] +nl

aOutput = myFunctionHash(sudoku)
        See "Return Hash.: T/F: "+ aOutput[:lValue] +" Row: "+ aOutput[:nRow] +" Col: "+␣
↪aOutput[:nCol] +nl

###----------------------------------
### isSolvedSoduku - Return ARRAY

Func myFunctionArray(sudoku)
    for Row = 1 to 9
        for Col = 1 to 9
            if sudoku[Row][Col] = 0

                //----------------------------
                // Return Array with 3 fields
                return [False, Row, Col]
            ok
        next
    next
return [True, Row, Col]

###----------------------------------
### isSolvedSoduku - Return HASH

Func myFunctionHash(sudoku)
    for Row = 1 to 3
        for Col = 1 to 3
            if sudoku[Row][Col] = 0

                //----------------------------------
                // Return Hash Table with 3 fields
                return  [   :lValue = False,
                            :nRow   = Row,
                            :nCol   = Col
                        ]
            ok
        next
    next

return  [ :lValue = False, :nRow = Row, :nCol = Col ]

###-----------------------------
```

17.20 Creating a Multi-Dimensional Array using List

A Multi-Dimensional Array of any size can be built using recursion in a Function

Example:

```
###-------------------------------------------------------
### Create Array -- Dimensions Any Size:  3D, 4D, 5D etc

dimList = [4,3,4]
bList   = createDimList(dimList)

###-------------------------------------------------------
### Populate the arrays using a counter 1 ,  4x4x4 = 256 , 2x3x4x5x6 = 720

Counter = 1

for Col=1 to dimList[1]
  for Row=1 to dimList[2]
    for Dep=1 to dimList[3]
            blist[Col][Row][Dep] = Counter
            Counter++
    next
  next
next

###-----------------------------------------------
### Print the array elements in block format

for Col=1 to dimList[1]
  for Row=1 to dimList[2]
    for Dep=1 to dimList[3]
            See bList[Col][Row][Dep] See " "
    next
    See nl
  next
    See nl
next

###==========================
### FUNCTIONS

###-------------------------------------------------------------------
### Recursive Create a Dimension Array
### Call by passing an array of dimensions: dimList = [2,3,4,5]
### Drop the first entry every iteration call, making newParms
###
### Example:
###    dimList = [4,2,3,2]               <<< Number and size of dimensions in array␣
↪format
###    bList   = createDimList(dimList)   <<< Call using the array as input

func createDimList(dimArray)
```

(continues on next page)

(continued from previous page)

```
    sizeList = len(dimArray)

    newParms = []
    for i = 2 to sizeList
       Add(newParms, dimArray[i])
    next

    alist = list(dimArray[1])

    if sizeList = 1
       return aList
    ok

    for t in alist
        t = createDimList(newParms)
    next

return alist
```

17.21 Swap Items

We can swap the list items using the Swap() function.

Example:

```
aList = [:one,:two,:four,:three]
see aList
see copy("*",50) + nl
swap(aList,3,4)
see aList
```

Output

```
one
two
four
three
**************************************************
one
two
three
four
```

CHAPTER

EIGHTEEN

STRINGS

In this chapter we are going to learn about strings creation and manipulation.

18.1 String Literals

Syntax:

```
cStr = "This is a string"
cStr2 = 'Another string'
cStr3 = :JustAnotherString
cStr4 = `Yet "another" 'string' ! `
```

18.2 Get String Length

We can get the string length (letters count inside a string) using the len() function

Syntax:

```
len(string) ---> string length
```

Example:

```
cStr = "How are you?"
see cStr + nl
see "String size : " + len(cStr) + nl
```

18.3 Convert Letters Case

Syntax:

```
lower(string) ---> convert string letters to lower case
upper(string) ---> convert string letters to UPPER case
```

Example:

```
cStr = "Welcome To The Ring Programming Language"
see cStr + nl + upper(cStr) + nl + lower(cStr)
```

18.4 Access String Letters

We can access a letter inside a string by the letter index

Syntax:

```
string[index] ---> get string letter
string[index] = letter  # set string letter
```

Example:

```
# print user name letter by letter (each letter in new line)

See "Hello, Enter your name : " give cName
for x = 1 to len(cName)
        see nl + cName[x]
next
```

We can use for in to get string letters.

Example:

```
# print user name letter by letter (each letter in new line)

See "Hello, Enter your name : " give cName
for x in cName
        see nl + x
next
```

We can modify the string letters

Example:

```
# convert the first letter to UPPER case

See "Enter your name : " give cName
cName[1] = upper(cName[1])
see "Hello " + cName
```

18.5 Left() Function

We can get a specified number of characters from a string using the Left() function.

The starting position is 1.

Syntax:

```
Left(string,count)
```

Example:

```
see left("Hello World!",5) # print Hello
```

18.6 Right() Function

We can get a specified number of characters from a string using the Right() function.

The starting position is the last character on the right.

Syntax:

```
Right(string,count)
```

Example:

```
see Right("Hello World!",6) # print World!
```

18.7 Trim() Function

We can remove all leading and trailing spaces from a string using the Trim() function.

Syntax:

```
trim(string)
```

Example:

```
cMsg = "     Welcome      "
see trim(cMsg)                  # print Welcome
```

18.8 Copy() Function

We can duplicate a string more than one time using the copy() function.

Syntax:

```
copy(string,nCount) ---> string replicated nCount times
```

Example

```
see copy("***hello***",3) # print ***hello******hello******hello***
```

18.9 Lines() Function

We can count the number of lines inside a string using the Lines() function.

Syntax:

```
lines(string) ---> Number of lines inside the string
```

Example:

```
cStr = "Hello
How are you?
are you fine?"
see lines(cStr)         # print 3
```

18.10 Substr() Function

We can work on sub strings inside a string using the substr() function. Using Substr() we can

- Find substring
- Get substring from position to end
- Get Number of characters from position
- Transform Substring To Another Substring

18.11 Find substring

Syntax:

```
substr(string,substring)  ---> the starting position of substring in string
```

Example:

```
cStr = "Welcome to the Ring programming language"
see substr(cStr,"Ring")         # print 16
```

18.12 Get substring from position to end

Syntax:

```
substr(string,position)  ---> Get substring starting from position to end
```

Example:

```
cStr = "Welcome to the Ring programming language"
nPos = substr(cStr,"Ring")      # nPos = 16
see substr(cStr,nPos)           # print Ring programming language
```

18.13 Get Number of Characters From Position

Syntax:

```
substr(string,position,count)  ---> Get characters starting from position
```

Example:

```
cStr = "Welcome to the Ring programming language"
nPos = substr(cStr,"Ring")      # nPos = 16
see substr(cStr,nPos,4)         # print Ring
```

18.14 Transform Substring To Another Substring

Syntax:

```
substr(string,substring,newsubstring)  ---> Transformed string (Match case)
substr(string,substring,newsubstring,1)  ---> Transformed string (Ignore case)
```

Example:

```
cStr = "Welcome to the New programming language"
see substr(cStr,"New","Ring") + nl  # print Welcome to the Ring programming language
see substr(cStr,"new","Ring",1)+ nl # print Welcome to the Ring programming language
```

18.15 strcmp() Function

We can compare between two strings using the strcmp() function.

Syntax:

```
strcmp(cString1,cString2) ---> value = 0 if cString1 = cString2
                               value < 0 if cString1 < cString2
                               value > 0 if cString1 > cString2
```

Example:

```
see strcmp("hello","hello") + nl +
    strcmp("abc","bcd") + nl +
    strcmp("bcd","abc") + nl
```

Output:

```
0
-1
1
```

18.16 str2list() and list2str() Functions

We can convert string lines to list items using the str2list() function. Also we can convert the list to a string using list2str() function.

Syntax:

```
str2list(string) ---> list contains the string lines
list2str(list)   ---> string contains the list items
```

Example:

```
/* output:
** Items : 4
** Item : Hello
** Item : How are you ?
** Item : are you fine ?
** Item : ok
** list2Str result = Hello
** How are you ?
** are you fine ?
** ok
** Done
*/

mystr = "Hello
How are you ?
are you fine ?
ok"

mylist = str2list(mystr)
see "Items : " + len(mylist) + nl

for x in mylist
        see "Item : " + x + nl
next

newstr = list2str(mylist)
see "list2Str result = " + newstr

if mystr = newstr
        see nl + "Done"
else
        see nl + "Error!"
ok
```

18.17 Merge binary characters

From Ring 1.0 we can create binary strings and do operations on these strings.

Starting from Ring 1.8, we can get individual characters from these strings and merge them together using the '+' operator.

Example:

```
cStr = "Welcome"
? cstr[1] + cstr[2] + cStr[5]
v = cstr[1] + cstr[2] + cStr[5]
? v
? len(v)
c1 = cStr[1]
? c1
aList = [1,2,3]
cStr = ""
for item in aList
        cStr += int2bytes(item)
next
? "All String"
? len(cStr)
? "First Part"
n1 = cStr[1] + cStr[2] + cStr[3] + cStr[4]
? len(n1)
? "Second Part"
n2 = cStr[5] + cStr[6] + cStr[7] + cStr[8]
? len(n2)
? "Third Part"
n3 = cStr[9] + cStr[10] + cStr[11] + cStr[12]
? len(n3)
? "All String"
cString = cStr[1] + cStr[2] + cStr[3] + cStr[4] +
          cStr[5] + cStr[6] + cStr[7] + cStr[8] +
          cStr[9] + cStr[10] + cStr[11] + cStr[12]
? len(cString)
? ascii(cStr[1])
? len(cStr[2])
```

Output:

```
Weo
Weo
3
W
All String
12
First Part
4
Second Part
4       }
Third Part
4
```

(continues on next page)

(continued from previous page)

```
All String
12
1
1
```

CHAPTER

NINETEEN

DATE AND TIME

In this chapter we are going to learn about the date and time functions.

19.1 Clock() Function

Syntax:

```
Clock() ---> The number of clock ticks from program start
```

Example:

```
See "Calculate performance" + nl
t1 = clock()
for x = 1 to 1000000 next
see clock() - t1
```

19.2 ClocksPerSecond() Function

Return how many clocks in one second

Syntax:

```
clockspersecond() ---> Number of clocks in one second
```

Example:

```
# Wait 1 second
t = clock()
while clock() - t <= clockspersecond() end
```

19.3 Time() Function

We can get the system time using the Time() function.

Example:

```
See "Time : " + time()
```

19.4 Date() Function

We can get the date using the Date() function.

Syntax:

```
Date() ---> String represent the date "dd/mm/yyyy"
```

Example:

```
See "Date : " + date()  # Date : 24/05/2015
```

19.5 TimeList() Function

We can print the date and the time information using the TimeList() function.

Syntax:

```
TimeList() ---> List contains the time and date information.
```

The next table presents the list items

index	value
1	abbreviated weekday name
2	full weekday name
3	abbreviated month name
4	full month name
5	Date & Time
6	Day of the month
7	Hour (24)
8	Hour (12)
9	Day of the year
10	Month of the year
11	Minutes after hour
12	AM or PM
13	Seconds after the hour
14	Week of the year (sun-sat)
15	day of the week
16	date
17	time
18	year of the century
19	year
20	time zone
21	percent sign

Example:

```
/* Output:
** Sun                  abbreviated weekday name
** Sunday               full weekday name
** May                  abbreviated month name
** May                  full month name
** 05/24/15 09:58:38    Date & Time
** 24                   Day of the month
** 09                   Hour (24)
** 09                   Hour (12)
** 144                  Day of the year
** 05                   Month of the year
** 58                   Minutes after hour
** AM                   AM or PM
** 38                   Seconds after the hour
** 21                   Week of the year (sun-sat)
** 0                    day of the week
** 05/24/15             date
** 09:58:38             time
** 15                   year of the century
** 2015                 year
** Arab Standard Time   time zone
** %                    percent sign
*/

See TimeList()
```

Example:

```
See "Day Name : " + TimeList()[2]       # Sunday
```

Example:

```
See "Month Name : " + TimeList()[4]     # May
```

19.6 AddDays() Function

Syntax:

```
AddDays(cDate,nDays) ---> Date from cDate and after nDays
```

Example:

```
cDate = date()
see cDate + nl                  # 24/05/2015
cDate = adddays(cDate,10)
see cDate + nl                  # 03/06/2015
```

19.7 DiffDays() Function

Syntax:

```
DiffDays(cDate1,cDate2) ---> number of days (Date1 - Date2)
```

Example:

```
cDate1 = date()
see cDate1 + nl                                         # 24/05/2015
cDate2 = adddays(cDate1,10)
see cDate2 + nl                                         # 03/06/2015
see "DiffDays = " + diffdays(cDate1,cDate2) + nl        # -10
see "DiffDays = " + diffdays(cDate2,cDate1) + nl        # 10
```

19.8 EpochTime() Function

Syntax:

```
EpochTime( cDate, cTime ) ---> Epoch Seconds
```

Example:

```
###-------------------------------------------------------------
# EpochTime()
# Example ---  EpochSec = EpochTime( Date(), Time() )
# Call Format: EpochSec = EpochTime( "15/07/2016", "10:15:30" )
#              EpochSec = 1468577730
```

(continues on next page)

(continued from previous page)

```
#-------------------------------------------------------------

Func EpochTime(Date, Time)

    arrayDate = split(Date, "/")
    arrayTime = split(Time, ":")

    Year = arrayDate[3] ; Month  = arrayDate[2] ; Day    = arrayDate[1]
    Hour = arrayTime[1] ; Minute = arrayTime[2] ; Second = arrayTime[3]

    cDate1    = Day +"/"+ Month +"/"+ Year
    cDate2    = "01/01/" + Year
    DayOfYear = DiffDays( cDate1, cDate2)

    ### Formula
    tm_sec  = Second    * 1
    tm_min  = Minute    * 60
    tm_hour = Hour      * 3600
    tm_yday = DayOfYear * 86400
    tm_year = Year      - 1900

    tm_year1 =         ( tm_year -  70)          * 31536000
    tm_year2 = ( floor(( tm_year -  69) /   4 )) * 86400
    tm_year3 = ( floor(( tm_year -   1) / 100 )) * 86400
    tm_year4 = ( floor(( tm_year + 299) / 400 )) * 86400

    ### Result
    EpochSec = tm_sec + tm_min + tm_hour + tm_yday +
               tm_year1 + tm_year2 - tm_year3 + tm_year4

return EpochSec
```

CHAPTER

TWENTY

CHECK DATA TYPE AND CONVERSION

In this chapter we are going to learn about the functions that can be used for

- Checking Data Type
- Checking Character
- Conversion

20.1 Check Data Type

The next functions can be used to check the data type

- isstring()
- isnumber()
- islist()
- type()
- isnull()

20.2 IsString() Function

Using the IsString() function we can know if the value is a string or not

Syntax:

```
IsString(value) ---> 1 if the value is a string or 0 if not
```

Example:

```
see isstring(5) + nl +          # print 0
    isstring("hello") + nl      # print 1
```

20.3 IsNumber() Function

Using the IsNumber() function we can know if the value is a number or not

Syntax:

```
IsNumber(value) ---> 1 if the value is a number or 0 if not
```

Example:

```
see isnumber(5) + nl +          # print 1
    isnumber("hello") + nl      # print 0
```

20.4 IsList() Function

Using the IsList() function we can know if the value is a list or not

Syntax:

```
IsList(value) ---> 1 if the value is a list or 0 if not
```

Example:

```
see islist(5) + nl +            # print 0
    islist("hello") + nl +      # print 0
    islist([1,3,5])             # print 1
```

20.5 Type() Function

We can know the type of a value using the Type() Function.

Syntax:

```
Type(value) ---> The Type as String
```

Example:

```
see Type(5) + nl +          # print NUMBER
Type("hello") + nl +        # print STRING
Type([1,3,5])               # print LIST
```

20.6 IsNULL() Function

We can check the value to know if it's null or not using the IsNULL() function

Syntax:

```
IsNULL(value) ---> 1 if the value is NULL or 0 if not
```

Example:

```
see isnull(5) + nl +        # print 0
isnull("hello") + nl +      # print 0
isnull([1,3,5]) + nl +      # print 0
isnull("") + nl +           # print 1
isnull("NULL")              # print 1
```

20.7 Check Character

The next functions can be used to check character

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()

20.8 IsAlNum() Function

We can test a character or a string using the IsAlNum() Function

Syntax:

```
IsAlNum(value) ---> 1 if the value is digit/letter or 0 if not
```

Example:

```
see isalnum("Hello") + nl +     # print 1
    isalnum("123456") + nl +    # print 1
    isalnum("ABCabc123") + nl + # print 1
    isalnum("How are you")      # print 0 because of spaces
```

20.9 IsAlpha() Function

We can test a character or a string using the IsAlpha() Function

Syntax:

```
IsAlpha(value) ---> 1 if the value is a letter or 0 if not
```

Example:

```
see isalpha("Hello") + nl +     # print 1
    isalpha("123456") + nl +    # print 0
    isalpha("ABCabc123") + nl + # print 0
    isalpha("How are you")      # print 0
```

20.10 IsCntrl() Function

We can test a character or a string using the IsCntrl() Function

Syntax:

```
IsCntrl(value) ---> 1 if the value is a control character (no printing position)
                    or 0 if not
```

Example:

```
See iscntrl("hello") + nl +     # print 0
    iscntrl(nl)                 # print 1
```

20.11 IsDigit() Function

We can test a character or a string using the IsDigit() Function

Syntax:

```
IsDigit(value) ---> 1 if the value is a digit or 0 if not
```

Example:

```
see isdigit("0123456789") + nl +        # print 1
    isdigit("0123a")                    # print 0
```

20.12 IsGraph() Function

We can test a character or a string using the IsGraph() Function

Syntax:

```
IsGraph(value) ---> 1 if the value can be printed (Except space) or 0 if not
```

Example:

```
see isgraph("abcdef") + nl +    # print 1
    isgraph("abc def")          # print 0
```

20.13 IsLower() Function

We can test a character or a string using the IsLower() Function

Syntax:

```
IsLower(value) ---> 1 if the value is lowercase letter or 0 if not
```

Example:

```
see islower("abcDEF") + nl +    # print 0
    islower("ghi")              # print 1
```

20.14 IsPrint() Function

We can test a character or a string using the IsPrint() Function

Syntax:

```
IsPrint(value) ---> 1 if the value occupies a printing position or 0 if not
```

Example:

```
see isprint("Hello") + nl +             # print 1
    isprint("Nice to see you") + nl +   # print 1
    isprint(nl)                         # print 0
```

20.15 IsPunct() Function

We can test a character or a string using the IsPunct() Function

Syntax:

```
IsPunct(value) ---> 1 if the value is a punctuation character or 0 if not
```

Example:

```
see ispunct("hello") + nl +     # print 0
    ispunct(",")                # print 1
```

20.16 IsSpace() Function

We can test a character or a string using the IsSpace() Function

Syntax:

```
IsSpace(value) ---> 1 if the value is a white-space or 0 if not
```

Example:

```
see isspace("   ") + nl +       # print 1
    isspace("test")             # print 0
```

20.17 IsUpper() Function

We can test a character or a string using the IsUpper() Function

Syntax:

```
IsUpper(value) ---> 1 if the value is an uppercase alphabetic letter or 0 if not
```

Example:

```
see isupper("welcome") + nl +    # print 0
    isupper("WELCOME")           # print 1
```

20.18 IsXdigit() Function

We can test a character or a string using the IsXdigit() Function

Syntax:

```
IsXdigit(value) ---> 1 if the value is a hexadecimal digit character or 0 if not
```

Example:

```
see isxdigit("0123456789abcdef") + nl +  # print 1
    isxdigit("123z")                     # print 0
```

20.19 Conversion

The next functions can be used for conversion

- number()
- string()
- ascii()
- char()
- hex()
- dec()
- str2hex()
- hex2str()

20.20 Number() Function

We can convert strings to numbers using the Number() function or the + operator.

Syntax:

```
Number(string) ---> Number
0 + string ---> Number
```

Example:

```
see number("5") + 5 + nl        # print 10
see 0 + "10" + 2                # print 12
```

20.21 String() Function

We can convert numbers to strings using the String() function or the + operator.

Syntax:

```
String(number) ---> String
"" + number ---> String
```

Example:

```
see string(5) + 5 + nl          # print 55
see "" + 10 + 2                 # print 102
```

20.22 Ascii() Function

We can get the ASCII code for a letter using the Ascii() function

Syntax:

```
Ascii(character) ---> ASCII Code
```

Example:

```
See ascii("m") + nl +   # print 109
    ascii("M")          # print 77
```

20.23 Char() Function

We can convert the ASCII code to character using the Char() function.

Syntax:

```
Char(ASCII Code) ---> character
```

Example:

```
See char(109) + nl +    # print m
    char(77)            # print M
```

20.24 Hex() Function

We can convert decimal to hexadecimal using the Hex() function.

Syntax:

```
Hex(decimal) ---> hexadecimal
```

Example:

```
See hex(10) + nl +      # print a
    hex(200)            # print c8
```

20.25 Dec() Function

We can convert hexadecimal to decimal using the Dec() function

Syntax:

```
Dec(hexadecimal) ---> decimal
```

Example:

```
See dec("a") + nl +      # print 10
    dec("c8")            # print 200
```

20.26 Str2hex() Function

We can convert string characters to hexadecimal characters using the Str2hex() function.

Syntax:

```
Str2hex(string) ---> hexadecimal string
```

Example:

```
See str2hex("hello")     # print 68656c6c6f
```

20.27 Hex2str() Function

We can convert hexadecimal characters to string using the Hex2str() function

Syntax:

```
Hex2Str(Hexadecimal string) ---> string
```

Example:

```
See hex2str("68656c6c6f")       # print hello
```

CHAPTER

TWENTYONE

MATHEMATICAL FUNCTIONS

In this chapter we are going to learn about the mathematical functions

21.1 List of functions

The Ring programming language comes with the next mathematical functions

Function	Description
sin(x)	Returns the sine of an angle of x radians
cos(x)	Returns the cosine of an angle of x radians
tan(x)	Returns the tangent of an angle of x radians
asin(x)	Returns the principal value of the arc sine of x, expressed in radians
acos(x)	Returns the principal value of the arc cosine of x, expressed in radians
atan(x)	Returns the principal value of the arc tangent of x, expressed in radians
atan2(y,x)	Returns the principal arc tangent of y/x, in the interval [-pi,+pi] radians
sinh(x)	Returns the hyperbolic sine of x radians
cosh(x)	Returns the hyperbolic cosine of x radians
tanh(x)	Returns the hyperbolic tangent of x radians
exp(x)	Returns the value of e raised to the xth power
log(x)	Returns the natural logarithm of x (the base of e)
log(x,b)	Returns the logarithm of x to the base of b
log10(x)	Returns the common logarithm (base-10 logarithm) of x
ceil(x)	Returns the smallest integer value greater than or equal to x
floor(x)	Returns the largest integer value less than or equal to x
fabs(x)	Returns the absolute value of x.
pow(x,y)	Returns x raised to the power of y
sqrt(x)	Returns the square root of x
random(x)	Returns a random number in the range [0,x]
srandom(x)	Initialize random number generator
unsigned(n,n,c)	Perform operation using unsigned numbers
decimals(n)	Determine the decimals digits after the point in float/double numbers

21.2 Example

```
See "Mathematical Functions" + nl
See "Sin(0) = " + sin(0) + nl
See "Sin(90) radians = " + sin(90) + nl
See "Sin(90) degree = " + sin(90*3.14/180) + nl

See "Cos(0) = " + cos(0) + nl
See "Cos(90) radians = " + cos(90) + nl
See "Cos(90) degree = " + cos(90*3.14/180) + nl

See "Tan(0) = " + tan(0) + nl
See "Tan(90) radians = " + tan(90) + nl
See "Tan(90) degree = " + tan(90*3.14/180) + nl

See "asin(0) = " + asin(0) + nl
See "acos(0) = " + acos(0) + nl
See "atan(0) = " + atan(0) + nl
See "atan2(1,1) = " + atan2(1,1) + nl

See "sinh(0) = " + sinh(0) + nl
See "sinh(1) = " + sinh(1) + nl
See "cosh(0) = " + cosh(0) + nl
See "cosh(1) = " + cosh(1) + nl
See "tanh(0) = " + tanh(0) + nl
See "tanh(1) = " + tanh(1) + nl

See "exp(0) = " + exp(0) + nl
See "exp(1) = " + exp(1) + nl
See "log(1) = " + log(1) + nl
See "log(2) = " + log(2) + nl
See "log10(1) = " + log10(1) + nl
See "log10(2) = " + log10(2) + nl
See "log10(10) = " + log10(10) + nl

See "Ceil(1.12) = " + Ceil(1.12) + nl
See "Ceil(1.72) = " + Ceil(1.72) + nl

See "Floor(1.12) = " + floor(1.12) + nl
See "Floor(1.72) = " + floor(1.72) + nl

See "fabs(1.12) = " + fabs(1.12) + nl
See "fabs(1.72) = " + fabs(1.72) + nl

See "pow(2,3) = " + pow(2,3) + nl

see "sqrt(16) = " + sqrt(16) + nl
```

Program Output:

```
Mathematical Functions
Sin(0) = 0
Sin(90) radians = 0.89
```

(continues on next page)

(continued from previous page)

```
Sin(90) degree = 1.00
Cos(0) = 1
Cos(90) radians = -0.45
Cos(90) degree = 0.00
Tan(0) = 0
Tan(90) radians = -2.00
Tan(90) degree = 1255.77
asin(0) = 0
acos(0) = 1.57
atan(0) = 0
atan2(1,1) = 0.79
sinh(0) = 0
sinh(1) = 1.18
cosh(0) = 1
cosh(1) = 1.54
tanh(0) = 0
tanh(1) = 0.76
exp(0) = 1
exp(1) = 2.72
log(1) = 0
log(2) = 0.69
log10(1) = 0
log10(2) = 0.30
log10(10) = 1
Ceil(1.12) = 2
Ceil(1.72) = 2
Floor(1.12) = 1
Floor(1.72) = 1
fabs(1.12) = 1.12
fabs(1.72) = 1.72
pow(2,3) = 8
sqrt(16) = 4
```

21.3 Random() Function

The Random() function generate a random number and we can set the maximum value (optional).

Syntax:

```
Random(x) ---> Random number in the range [0,x]
```

Example:

```
for x = 1 to 20
        see  "Random number : " + random() + nl +
             "Random number Max (100) : " + random(100) + nl
next
```

Program Output:

```
Random number : 31881
Random number Max (100) : 80
Random number : 5573
Random number Max (100) : 63
Random number : 2231
Random number Max (100) : 43
Random number : 12946
Random number Max (100) : 39
Random number : 22934
Random number Max (100) : 48
Random number : 4690
Random number Max (100) : 52
Random number : 13196
Random number Max (100) : 65
Random number : 30390
Random number Max (100) : 87
Random number : 4327
Random number Max (100) : 77
Random number : 12456
Random number Max (100) : 17
Random number : 28438
Random number Max (100) : 13
Random number : 30503
Random number Max (100) : 6
Random number : 31769
Random number Max (100) : 94
Random number : 8274
Random number Max (100) : 65
Random number : 14390
Random number Max (100) : 90
Random number : 28866
Random number Max (100) : 12
Random number : 24558
Random number Max (100) : 70
Random number : 29981
Random number Max (100) : 77
Random number : 12847
Random number Max (100) : 63
Random number : 6632
Random number Max (100) : 60
```

21.4 SRandom() Function

The SRandom() function initialize random number generator.

Syntax:

```
SRandom(x)
```

21.5 Unsigned() Function

We can use unsigned numbers using the Unsigned() function.

Syntax:

```
Unsigned(nNum1,nNum2,cOperator) --> result of cOperator operation on nNum1,nNum2
```

Example:

```
see oat_hash("hello") + nl

# Jenkins hash function - https://en.wikipedia.org/wiki/Jenkins_hash_function
func oat_hash cKey
        h = 0
        for x in cKey
                h = unsigned(h,ascii(x),"+")
                h = unsigned(h,unsigned(h,10,"<<"),"+")
                r = unsigned(h,6,">>")
                h = unsigned(h, r,"^")
        next
        h = unsigned(h,unsigned(h,3,"<<"),"+")
        h = unsigned(h,unsigned(h,11,">>"),"^")
        h = unsigned(h,unsigned(h,15,"<<"),"+")
        return h
```

Output:

```
3372029979.00
```

21.6 Decimals() Functions

We can determine the decimals numbers count after the point in float/double numbers using the decimals() function.

Syntax:

```
Decimals(nDecimalsCount)
```

Example:

```
x = 1.1234567890123
for d = 0 to 14
        decimals(d)
        see x + nl
next
```

Output:

```
1
1.1
1.12
1.123
1.1235
```

(continues on next page)

(continued from previous page)

```
1.12346
1.123457
1.1234568
1.12345679
1.123456789
1.1234567890
1.12345678901
1.123456789012
1.1234567890123
1.12345678901230
```

21.7 Using _ in numbers

We can use _ between numbers digits.

Example:

```
x = 1_000_000
see type(x)+nl
see x+1+nl
```

Output:

```
NUMBER
100000001
```

21.8 Using f after numbers

We can use the 'f' letter after numbers.

Example:

```
x = 19.99f
see type(x) + nl
```

Output:

```
NUMBER
```

21.9 Using Hexadecimal Numbers

We can write Hexadecimal number by preceding with "0x" or "0X"

Example:

```
x = 0x10
? x              # 16
x = 0xff
```

(continues on next page)

(continued from previous page)

```
? x             # 255
x = 0x0A
? x             # 10
? 0xFFFF        # 65535
? 0x0A + 1      # 10+1 = 11
```

CHAPTER

TWENTYTWO

FILES

In this chapter we are going to learn about files functions.

- Read()
- Write()
- Dir()
- Rename()
- Remove()
- fopen()
- fclose()
- fflush()
- freopen()
- tempfile()
- tempname()
- fseek()
- ftell()
- rewind()
- fgetpos()
- fsetpos()
- clearerr()
- feof()
- ferror()
- perror()
- fgetc()
- fgets()
- fputc()
- fputs()
- ungetc()
- fread()

- fwrite()
- fexists()
- direxists()
- getpathtype()
- getfilesize()
- Numbers and Bytes

22.1 Read() Function

We can read the file content using the Read() function

Syntax:

```
Read(cFileName) ---> String contains the file content
```

Example:

```
see read("myfile.txt")
```

The read function can read binary files too

Example:

```
see read("myapp.exe")
```

22.2 Write() Function

We can write string to file using the Write() function

The write function can write binary data to binary files.

Syntax:

```
Write(cFileName,cString)        # write string cString to file cFileName
```

Example:

```
# copy file
cFile = read("ring.exe")
write("ring2.exe",cFile)
```

22.3 Dir() Function

We can get the folder contents (files & sub folders) using the Dir() function.

Syntax:

```
Dir(cFolderPath) ---> List contains files & sub folders.
```

This function returns a list and each list item is a list of two items

- File/sub folder name
- Type (0 = File , 1 = Folder/Directory)

Example:

```
see "Testing DIR() " + nl
mylist = dir("C:\myfolder")
for x in mylist
        if x[2]
                see "Directory : " + x[1] + nl
        else
                see "File : " + x[1] + nl
        ok
next
see "Files count : " + len(mylist)
```

22.4 Rename() Function

We can rename files using the Rename() function

Syntax:

```
Rename(cOldFileName,cNewFileName) ---> Number ( Status: Success (0) , Error (-1) )
```

Example:

```
rename("file.txt","help.txt")
```

22.5 Remove() Function

We can delete a file using the Remove() function

Syntax:

```
Remove(cFileName)
```

Example:

```
remove("test.txt")
```

22.6 Fopen() Function

We can open a file using the Fopen() function

Syntax:

```
Fopen(cFileName,cMode) ---> File Handle
```

Mode	Description
"r"	Reading (The file must exist)
"w"	Writing (create empty file / overwrite)
"a"	Appends (create file if it doesn't exist)
"r+"	update (reading/writing)
"w+"	Create empty file (reading/writing)
"a+"	reading & appending

22.7 Fclose() Function

When we open a file using fopen() function, we can close it using the Fclose() function

Syntax:

```
Fclose(file handle)
```

22.8 Fflush() Function

We can flushes the output buffer of a stream using the Fflush() function

Syntax:

```
Fflush(file handle)
```

22.9 Freopen() Function

We can open another file using the same file handle and at the same time close the old file

Syntax:

```
Freopen(cFileName,cMode,file handle) ---> file handle
```

Example:

```
freopen("myprogoutput.txt","w+",stdout)
see "welcome" + nl
for x = 1 to 10
        see x + nl
next
```

(continues on next page)

(continued from previous page)

```
/*
** Read : https://en.wikipedia.org/wiki/Device_file#Device_files
** The next code is not portable, we can use iswindows() before
** using it and we can write special code for each operating system.
*/

freopen("CON","w",stdout)       # For Microsoft Windows
see "Done" + nl                 # print to stdout again
```

Output:

```
# Output to stdout
Done

# Output to file : myprogoutput.txt
welcome
1
2
3
4
5
6
7
8
9
10
```

22.10 Tempfile() Function

The function Tempfile() creates a temp. file (binary).

The file will be deleted automatically when the stream is closed

Syntax:

```
TempFile() ---> file handle
```

22.11 Tempname() Function

We can generate temp. file name using the Tempname() function

The generated name will be different from the name of any existing file

Syntax:

```
Tempname() ---> generated file name as string
```

22.12 Fseek() Function

We can set the file position of the stream using the Fseek() function

Syntax:

```
Fseek(file handle, nOffset, nWhence) ---> zero if successful
```

The next table presents the nWhence values

Value	Description
0	Beginning of file
1	Current position
2	End of file

22.13 Ftell() Function

We can know the current file position of a stream using the Ftell() function

Syntax:

```
Ftell(file handle) ---> file position as number
```

22.14 Rewind() Function

We can set the file position to the beginning of the file using the Rewind() function

Syntax:

```
Rewind(file handle)
```

22.15 Fgetpos() Function

We can get handle to the current file position using the Fgetpos() function

Syntax:

```
Fgetpos(file handle) ---> position handle
```

22.16 Fsetpos() Function

We can set the current file position using the Fsetpos() function

Syntax:

```
Fsetpos(file handle,position handle)
```

22.17 Clearerr() Function

We can clear the EOF error and the error indicators of a stream using the clearerr() function

Syntax:

```
Clearerr(file handle)
```

22.18 Feof() Function

We can test the end-of-file indicator using the Feof() function

Syntax:

```
Feof(file handle) ---> returns 1 if EOF and 0 if not
```

22.19 Ferror() Function

We can test the error indicator of a given stream using the Ferror() function

Syntax:

```
Ferror(file handle) ---> returns 1 if error and 0 if not
```

22.20 Perror() Function

We can print error message to the stderr using the Perror() function

Syntax:

```
Perror(cErrorMessage)
```

22.21 Fgetc() Function

We can get the next character from the stream using the Fgetc() function

Syntax:

```
Fgetc(file handle) ---> returns character or EOF
```

22.22 Fgets() Function

We can read new line from the stream using the Fgets() function

Syntax:

```
Fgets(file handle,nSize) ---> string
```

The function stop when nSize characters are read, new line character is read or EOF.

22.23 Fputc() Function

We can write a character to the stream using the Fputc() function

Syntax:

```
Fputc(file handle,cChar)
```

22.24 Fputs() Function

We can write a string to the stream using the Fputs() function

Syntax:

```
Fputs(file handle,cString)
```

22.25 Ungetc() Function

We can push a character to the stream using the Ungetc() function

The character will be available for the next read

Syntax:

```
Ungetc(file handle,character)
```

22.26 Fread() Function

We can read data from a stream using the Fread() function

Syntax:

```
Fread(file handle,nSize)
```

22.27 Fwrite() Function

We can write data to a stream using the Fwrite() function

Syntax:

```
Fwrite(file handle,cString)
```

22.28 Fexists() Function

We can check if a file exists using the Fexists() function

Syntax:

```
Fexists(cFileName) ---> returns 1 if the file exists
```

Example:

```
see fexists("b:\mahmoud\apps\ring\ring.exe") + nl +
    fexists("b:\mahmoud\apps\ring\ring2.exe") + nl
```

Output:

```
1
0
```

22.29 Direxists() Function

Syntax:

```
Direxists(cDirPath) ---> returns 1 if the directory exists
```

Example:

```
? direxists("b:\ring") + nl +
  direxists("b:\ring\bin2")
```

Output:

```
1
0
```

22.30 Getpathtype() Function

We can get the type a given path (file or directory) using the Getpathtype() function

Syntax:

```
Getpathtype(cPath) ---> 0 if the path doesn't exists
                        1 if it corresponds to existing file
                        2 if it corresponds to existing directory
                        -1 if the path exists but has
                           an unknown type (e.g. a pipe)
```

Example:

```
? Getpathtype("b:\ring\bin\ring.exe") + nl +
  Getpathtype("b:\ring") + nl +
  Getpathtype("b:\ring\bin2")
```

Output:

```
1
2
0
```

22.31 Getfilesize() Function

We can get the size in bytes of a given file using the Getfilesize() function

Syntax:

```
Getfilesize(cFilePath) ---> file size in bytes as a positive Number
                            or -1 in case of failure
                            (e.g. path doesn't exist or not a regular file)
```

Example:

```
? Getfilesize("b:\ring\bin\ring.exe") + nl +
  Getfilesize("b:\ring") + nl +
  Getfilesize("b:\ring\ring2.exe")
```

Output:

```
80384
-1
-1
```

22.32 Example

The next program test some of the file functions

```
See "testing file functions" + nl

See "open file" + nl
fp = fopen(exefolder() + "../tests/scripts/s65.ring","r")

See "reopen" + nl
fp = freopen(exefolder() + "../tests/scripts/s78.ring","r",fp)
See "close file" + nl
fclose(fp)

see "temp file" + nl
fp = tempfile()
fclose(fp)

see "temp name" + nl
see tempname() + nl

remove(exefolder() + "../tests/scripts/mytest2.txt")
write(exefolder() + "../tests/scripts/tests1.txt","hello")
rename(exefolder() + "../tests/scripts/test1.txt",exefolder() +
                                        "../tests/scripts/mytests2.txt")

see "print file" + nl
fp = fopen(exefolder() + "../samples/fromdoc/filefuncs.ring","r")
r = fgetc(fp)
while isstring(r)
                see r
                r = fgetc(fp)
end
fclose(fp)

see nl+"print line from the file" + nl
fp = fopen(exefolder() + "../samples/fromdoc/filefuncs.ring","r")
r = fgets(fp,33)
see r + nl
fclose(fp)
fp = fopen(exefolder() + "../tests/scripts/test78.txt","w+")
fseek(fp,0,2) # goto end of file
fputc(fp,"t")
fputc(fp,"e")
fputc(fp,"s")
fputc(fp,"t")
fputs(fp,"tests2")
fclose(fp)

see "print file" + nl
see read(exefolder() + "../tests/scripts/test78.txt")

fp = fopen(exefolder() + "../tests/scripts/test78.txt","r")
```

(continues on next page)

(continued from previous page)

```
see "testing ungetc() " + nl
for x = 1 to 3
                r = fgetc(fp)
                see r + nl
                ungetc(fp,r)
next
fclose(fp)

see "testing fread() " + nl
fp = fopen(exefilename(),"rb")
r = fread(fp,100)
see r + nl
fclose(fp)

see "testing fwrite() " + nl
fp = fopen(exefolder() + "../tests/scripts/test1.txt","wb")
fwrite(fp,r)
fclose(fp)
```

The next example print part of the content of a binary file

```
see "Testing: fread()" +" FileName: "+ exefilename() +nl +nl
fp = fopen(exefilename(),"rb")
r = fread(fp,800)
for n =1 to len(r)
        if isprint(substr(r, n, 1))
                see substr(r, n, 1)
        else
                see "."
        ok
        ### 80 char per line
        if n % 80 = 0
                see nl
        ok
next
fclose(fp)
```

22.33 Numbers and Bytes

The next functions to convert between Numbers and Bytes.

- Int2Bytes()
- Float2Bytes()
- Double2Bytes()
- Bytes2Int()
- Bytes2Float()
- Bytes2Double()

Example:

```
see "Test Int2Bytes() and Bytes2Int() - Value : 77" + nl
r = Int2Bytes(77)
see "Int Size : " + len(r) + nl
see r + nl
see Bytes2Int(r) + nl
see "Test Float2Bytes() and Bytes2Float() - Value 77.12" + nl
r = Float2Bytes(77.12)
see "Float Size : " + len(r) + nl
see r + nl
see Bytes2Float(r) + nl
see "Test Double2Bytes() and Bytes2Double() - Value 9999977.12345" + nl
r = Double2Bytes(9999977.12345)
see "Double Size : " + len(r) + nl
see r + nl
decimals(5)
see Bytes2Double(r) + nl
```

CHAPTER

TWENTYTHREE

SYSTEM FUNCTIONS

In this chapter we are going to learn about the system functions

- System()
- SysGet()
- SysSet()
- SysUnset()
- IsMSDOS()
- IsWindows()
- IsWindows64()
- IsUnix()
- IsMacOSX()
- IsLinux()
- IsFreeBSD()
- IsAndroid()
- GetArch()
- Windowsnl()
- Get Command Line Arguments
- Get Active Source File Name
- CurrentDir()
- ExeFileName()
- ChDir()
- ExeFolder()
- Version()
- Shutdown()

23.1 System() Function

We can execute system commands using the system() function

Syntax:

```
System(cCommand)
```

Example:

```
System("myapp.exe")     # Run myapp.exe
System("ls")            # print list of files
```

23.2 SysGet() Function

We can get environment variables using the Get() function

Syntax:

```
SysGet(cVariable)
```

Example:

```
see sysget("path")              # print system path information
```

23.3 SysSet() Function

We can set environment variables using the SysSet() function

```
SysSet(cVariable, cValue) ---> Returns 1 for success and return 0 for failure
```

23.4 SysUnset() Function

We can delete an environment variables using the SysUnset() function

```
SysUnset(cVariable) ---> Returns 1 for success and return 0 for failure
```

23.5 IsMSDOS() Function

We can check if the operating system is MSDOS or not using the IsMSDOS() function

Syntax:

```
IsMSDOS() ---> Returns 1 if the operating system is MS-DOS, Returns 0 if it's not
```

23.6 IsWindows() Function

We can check if the operating system is Windows or not using the IsWindows() function

Syntax:

```
IsWindows() ---> Returns 1 if the operating system is Windows, Returns 0 if it's not
```

23.7 IsWindows64() Function

We can check if the operating system is Windows 64bit or not using the IsWindows64() function

Syntax:

```
IsWindows64() ---> Returns 1 if the operating system is Windows64, Returns 0 if it's not
```

23.8 IsUnix() Function

We can check if the operating system is Unix or not using the IsUnix() function

Syntax:

```
IsUnix() ---> Returns 1 if the operating system is Unix, Returns 0 if it's not
```

23.9 IsMacOSX() Function

We can check if the operating system is macOS or not using the IsMacOSX() function

Syntax:

```
IsMacOSX() ---> Returns 1 if the operating system is Mac OS X, Returns 0 if it's not
```

23.10 IsLinux() Function

We can check if the operating system is Linux or not using the IsLinux() function

Syntax:

```
IsLinux() ---> Returns 1 if the operating system is Linux, Returns 0 if it's not
```

23.11 IsFreeBSD() Function

We can check if the operating system is FreeBSD or not using the IsFreeBSD() function

Syntax:

```
IsFreeBSD() ---> Returns 1 if the operating system is FreeBSD, Returns 0 if it's not
```

23.12 IsAndroid() Function

We can check if the operating system is Android or not using the IsAndroid() function

Syntax:

```
IsAndroid() ---> Returns 1 if the operating system is Android, Returns 0 if it's not
```

23.13 Example

```
see "IsMSDOS()     --> " + ismsdos()     + nl
see "IsWindows()   --> " + iswindows()   + nl
see "IsWindows64() --> " + iswindows64() + nl
see "IsUnix()      --> " + isunix()      + nl
see "IsMacOSX()    --> " + ismacosx()    + nl
see "IsLinux()     --> " + islinux()     + nl
see "IsFreeBSD()   --> " + isfreebsd()   + nl
see "IsAndroid()   --> " + isandroid()   + nl
```

Output:

```
IsMSDOS()     --> 0
IsWindows()   --> 1
IsWindows64() --> 0
IsUnix()      --> 0
IsMacOSX()    --> 0
IsLinux()     --> 0
IsFreeBSD()   --> 0
IsAndroid()   --> 0
```

23.14 GetArch() Function

We can detect the architecture of the Ring executable using the GetArch() function

Syntax:

```
GetArch() ---> cString (The name of the architecture of the Ring executable)
```

Possible values are:

```
x86
x64
arm64
arm
unknown
```

Example:

```
switch getarch()
on "x86"
        ? "x86 32bit architecture"
on "x64"
        ? "x64 64bit architecture"
on "arm64"
        ? "ARM64 64bit architecture"
on "arm"
        ? "ARM 32bit architecture"
other
        ? "Unknown architecture"
off
```

23.15 Windowsnl() Function

We can get the windows new line string using the Windowsnl() function.

Syntax:

```
WindowsNL() ---> Returns a string contains CR+LF = CHAR(13) + CHAR(10)
```

Example:

```
cStr = read("input.txt")

if iswindows()
        cStr = substr(cStr,windowsnl(),nl)
ok

aList = str2list(cStr)
# to do - list items processing using "for in"
cStr = list2str(aList)

if iswindows()
        cStr = substr(cStr,nl,windowsnl())
ok

write("output.txt",cStr)
```

23.16 Get Command Line Arguments

We can get the command line arguments passed to the ring script using the sysargv variable.

The sysargv variable is a list contains the command line parameters.

Example

```
see copy("=",30) + nl
see "Command Line Parameters" + nl
see "Size : " + len(sysargv) + nl
see sysargv
see copy("=",30) + nl
if len(sysargv) < 4 return ok
nStart = sysargv[3]
nEnd = sysargv[4]
for x = nStart to nEnd
        see x + nl
next
```

Output

```
b:\mahmoud\apps\ring>ring tests\syspara.ring 1 10
==============================
Command Line Parameters
Size : 4
ring
tests\syspara.ring
1
10
==============================
1
2
3
4
5
6
7
8
9
10
```

23.17 Get Active Source File Name

We can get the active source file name (*.ring) using the filename() function

Syntax:

```
filename() ---> String contains the active source file name.
```

Example:

```
see "Active Source File Name : " + filename() + nl
```

Output:

```
Active Source File Name : tests\filename.ring
```

Example:

```
if sysargv[2] = filename()
        see "I'm the main program file!" + nl
        # we can run tests here!
else
        see "I'm a sub file in a program" + nl
ok
```

23.18 PrevFileName() Function

Using the PrevFileName() function we can get the previous active source file name.

The previous file would be the file of the caller function, Or the file of the function that we called before calling PrevFileName().

Syntax:

```
prevfilename() ---> String contains the previous source file name.
```

Example:

The next function in stdlib.ring uses the PrevFileName() to know if the file of the caller function is the main source file of the program or not.

```
Func IsMainSourceFile
        if PrevFileName() = sysargv[2]
                return true
        ok
        return false
```

23.19 CurrentDir() Function

Return the path of the current directory

Syntax:

```
CurrentDir() ---> String contains the path of the current directory
```

23.20 ExeFileName() Function

Return the Ring executable file name

Syntax:

```
exefilename() ---> String contains the Ring executable file name
```

23.21 ChDir() Function

Change the current directory

Syntax:

```
ChDir(cNewPath)
```

23.22 ExeFolder() Function

Return the Ring executable file path

Syntax:

```
exefolder() ---> String contains the Ring executable path
```

23.23 Version() Function

Return the Ring version

Syntax:

```
version() ---> String contains the Ring version
```

Output:

```
1.19
```

23.24 Shutdown() Function

Close the application

Syntax:

```
shutdown(nStatus) ---> Close the application
```

23.25 NofProcessors() Function

We can detect the number of processors using the NofProcessors() Function

Syntax:

```
NofProcessors() ---> nProcessors
```

Example:

```
? NofProcessors()
```

CHAPTER

TWENTYFOUR

EVAL() AND DEBUGGING

In this chapter we are going to learn about

- Error Handling using Try/Catch/Done
- Eval() function
- Raise() function
- Assert() function

24.1 Try/Catch/Done

Syntax:

```
Try
        Statements...
Catch
        Statements...
Done
```

The statements in the Try block will be executed, if any error happens then the statements in the catch block will be executed.

Inside the catch block we can use the variable cCatchError to get the error message

Example:

```
Try
        see 5/0
Catch
        see "Catch!" + nl + cCatchError
Done
```

Output:

```
Catch!
Error (R1) : Can't divide by zero !
```

24.2 Eval() Function

We can execute code during the runtime from string using the Eval() function

Syntax:

```
Eval(cCode)
```

Example:

```
Eval("nOutput = 5+2*5 " )
See "5+2*5 = " + nOutput + nl
Eval("for x = 1 to 10 see x + nl next")
Eval("func test see 'message from test!' ")
test()
```

Output:

```
5+2*5 = 15
1
2
3
4
5
6
7
8
9
10
message from test!
```

We can use the Return command to return a value

Example:

```
see Eval("return 5*5")
```

Output:

```
25
```

24.3 Raise() Function

We can raise an exception using the Raise() function

Syntax:

```
Raise(cErrorMessage)
```

The function will display the error message then end the execution of the program.

We can use Try/Catch/Done to avoid exceptions generated by raise() function.

Example:

```
nMode = 10

if nMode < 0 or nMode > 5
        raise("Error : nMode not in the range 1:4")
ok
```

Output:

```
Line 4 Error : nMode not in the range 1:4
In raise in file tests\raise.ring
```

Example:

```
try
        testmode(6)
catch
        see "avoid raise!"
done

testmode(-1)

func testmode nMode

        if nMode < 0 or nMode > 5
                raise("Error : nMode not in the range 1:4")
        ok
```

Output:

```
avoid raise!
Line 12 Error : nMode not in the range 1:4
In raise In function testmode() in file tests\raise2.ring
called from line 7  in file tests\raise2.ring
```

24.4 Assert() Function

We can use the Assert() function to test conditions before executing the code

If the test fail the program will be terminated with an error message contains the assert condition.

Syntax:

```
Assert( condition )
```

Example:

```
x = 10
assert( x = 10)
assert( x = 100 )
```

Output:

```
Line 3 Assertion Failed!
In assert in file tests\assert.ring
```

CHAPTER

TWENTYFIVE

DEMO PROGRAMS

In this chapter we will see simple demo programs

- Language Shell
- Main Menu

25.1 Language Shell

We can create simple interactive programming environment using the next program

```
while true
        see nl + "code:> "
        give cCode
        try
                eval(cCode)
        catch
                see cCatchError
        done
end
```

Output:

```
code:> see "hello world"
hello world
code:> for x = 1 to 10 see x + nl next
1
2
3
4
5
6
7
8
9
10

code:> func test see "Hello from test" + nl

code:> test()
Hello from test
```

(continues on next page)

(continued from previous page)

```
code:> bye
```

25.2 Main Menu

Example:

```
# Demo Program

while true

        see "

        Main Menu
        ===========
        [1] Say Hello
        [2] Sum two numbers
        [3] Stars
        [4] Fact
        [5] Exit

        " give nMenu see nl

        # we can use Switch-ON-Other-OFF instead of IF-BUT-ELSE-OK

        Switch nMenu
        On 1 sayhello()
        On 2 Sum()
        On 3 Stars()
        On 4
                see "Enter Number : " give x
                see "Output : "

                Try
                        see Fact(number(x))
                Catch
                        see "Error in parameters!" + nl
                Done

        On "5" return
        Other see "bad option" + nl
        Off

end

func sayhello
        see "Enter your name ? " give fname
        see "Hello " + fname + nl

func sum
```

(continues on next page)

(continued from previous page)

```
        see "number 1 : " give num1 see "number 2 : " give num2
        see "Sum : " see 0 + num1 + num2

func stars
        for x = 1 to 10
                see space(8)
                for y = 1 to x see "*" next see nl
        next

func fact x if x = 0 return 1 else return x * fact(x-1) ok

func space x y = "" for t=1 to x y += " " next return y
```

Output:

```
        Main Menu
        ===========
        [1] Say Hello
        [2] Sum two numbers
        [3] Stars
        [4] Fact
        [5] Exit

        1

Enter your name ? Mahmoud Fayed
Hello Mahmoud Fayed


        Main Menu
        ===========
        [1] Say Hello
        [2] Sum two numbers
        [3] Stars
        [4] Fact
        [5] Exit

        2

number 1 : 3
number 2 : 4
Sum : 7

        Main Menu
        ===========
        [1] Say Hello
        [2] Sum two numbers
        [3] Stars
        [4] Fact
        [5] Exit

        3
```

(continues on next page)

(continued from previous page)

```
        *
        **
        ***
        ****
        *****
        ******
        *******
        ********
        *********
        **********


        Main Menu
        ===========
        [1] Say Hello
        [2] Sum two numbers
        [3] Stars
        [4] Fact
        [5] Exit

        4

Enter Number : 5
Output : 120

        Main Menu
        ===========
        [1] Say Hello
        [2] Sum two numbers
        [3] Stars
        [4] Fact
        [5] Exit

        5
```

CHAPTER

TWENTYSIX

ODBC FUNCTIONS

This chapter contains the ODBC functions provided by the Ring programming language.

- odbc_init()
- odbc_drivers()
- odbc_datasources()
- odbc_close()
- odbc_connect()
- odbc_disconnect()
- odbc_execute()
- odbc_colcount()
- odbc_fetch()
- odbc_getdata()
- odbc_tables()
- odbc_columns()
- odbc_autocommit()
- odbc_commit()
- odbc_rollback()

Before using the next function load the odbclib.ring library

```
load "odbclib.ring"
# Use ODBC functions
```

26.1 odbc_init() Function

We can create ODBC Handle using the odbc_init() function

Syntax:

```
odbc_init() ---> ODBC Handle
```

26.2 odbc_drivers() Function

We can get a list of ODBC drivers using the odbc_drivers() function

Syntax:

```
odbc_drivers(ODBC Handle) ---> List of Drivers
```

26.3 odbc_datasources() Function

We can get a list of ODBC data sources using the odbc_datasources() function

Syntax:

```
odbc_datasources(ODBC Handle) ---> List of Data sources
```

26.4 odbc_close() Function

After the end of using ODBC functions we can free resources using ODBC_Close() function

Syntax:

```
odbc_close(ODBC Handle)
```

26.5 Print List of ODBC Drivers

The next example print a list of ODBC drivers.

```
See "ODBC test 1" + nl
oODBC = odbc_init()
See "Drivers " + nl
see odbc_drivers(oODBC)
odbc_close(oODBC)
```

Output:

```
ODBC test 1
Drivers
Microsoft Access-Treiber (*.mdb) - SQLLevel=0
Driver do Microsoft Paradox (*.db ) - SQLLevel=0
Driver do Microsoft Excel(*.xls) - SQLLevel=0
Microsoft Text Driver (*.txt; *.csv) - SQLLevel=0
Driver da Microsoft para arquivos texto (*.txt; *.csv) - SQLLevel=0
Microsoft dBase-Treiber (*.dbf) - SQLLevel=0
SQL Server - CPTimeout=60
Microsoft Excel Driver (*.xls) - SQLLevel=0
Driver do Microsoft dBase (*.dbf) - SQLLevel=0
Microsoft Paradox-Treiber (*.db ) - SQLLevel=0
```

(continues on next page)

(continued from previous page)

```
Microsoft ODBC for Oracle - CPTimeout=120
Microsoft Text-Treiber (*.txt; *.csv) - SQLLevel=0
Microsoft Excel-Treiber (*.xls) - SQLLevel=0
Microsoft Access Driver (*.mdb) - SQLLevel=0
Driver do Microsoft Access (*.mdb) - SQLLevel=0
Microsoft Paradox Driver (*.db ) - SQLLevel=0
Microsoft dBase Driver (*.dbf) - SQLLevel=0
Microsoft Access Driver (*.mdb, *.accdb) - UsageCount=3
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb) - UsageCount=3
Microsoft Access Text Driver (*.txt, *.csv) - UsageCount=3
SQL Server Native Client 10.0 - UsageCount=1
SQL Server Native Client 11.0 - UsageCount=1
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx) - UsageCount=3
Microsoft Access Paradox Driver (*.db) - UsageCount=3
MySQL ODBC 5.3 ANSI Driver - UsageCount=1
MySQL ODBC 5.3 Unicode Driver - UsageCount=1
ODBC Driver 11 for SQL Server - UsageCount=1
Lianja ODBC Driver - CPTimeout=60
Microsoft Visual FoxPro Driver - UsageCount=1
Microsoft Visual FoxPro-Treiber - UsageCount=1
Driver para o Microsoft Visual FoxPro - UsageCount=1
Microsoft FoxPro VFP Driver (*.dbf) - UsageCount=1
```

26.6 Print List of ODBC Data Sources

The next example print a list of ODBC data sources.

```
See "ODBC test 2" + nl
pODBC = odbc_init()
See "Data Sources " + nl
see odbc_datasources(pODBC)
odbc_close(pODBC)
```

Output:

```
ODBC test 2
Data Sources
Excel Files - Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)
MS Access Database - Microsoft Access Driver (*.mdb, *.accdb)
Customer - Microsoft Access Driver (*.mdb)
IdCardData - Microsoft Access Driver (*.mdb)
MyProjectData2 - Microsoft Access Driver (*.mdb)
MyData - Microsoft Access Driver (*.mdb)
MonprojetData - Microsoft Access Driver (*.mdb)
dBASE Files - Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)
myvfpdata - Microsoft Visual FoxPro Driver
FACTORYDATA - Microsoft Access Driver (*.mdb)
TRAININGSYSDATA - Microsoft Access Driver (*.mdb)
RVCSYSDATASQLDB - SQL Server Native Client 11.0
PWCTRVCDATA - Microsoft Access Driver (*.mdb)
```

(continues on next page)

(continued from previous page)

```
MyCompany - Microsoft Access Driver (*.mdb)
HCS - Microsoft Access Driver (*.mdb)
HCS2 - Microsoft Access Driver (*.mdb, *.accdb)
MyProjectData - Microsoft Access Driver (*.mdb)
Xtreme Sample Database 2008 - Microsoft Access Driver (*.mdb)
Lianja_Southwind - Lianja ODBC Driver
Visual FoxPro Database - Microsoft Visual FoxPro Driver
Visual FoxPro Tables - Microsoft Visual FoxPro Driver
```

26.7 odbc_connect() Function

We can connect to the database using the odbc_connect() function.

Syntax:

```
odbc_connect(ODBC Handle, cConnectionString)
```

26.8 odbc_disconnect() Function

We can close the connection to the database using the odbc_disconnect() function.

Syntax:

```
odbc_disconnect(ODBC Handle)
```

26.9 Open and Close Connection

The next example connect to the database then close the connection

```
See "ODBC test 3" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "disconnect" + nl
odbc_disconnect(pODBC)
See "Close database..." + nl
odbc_close(pODBC)
```

Output:

```
ODBC test 3
Connect to database
1
disconnect
Close database...
```

26.10 odbc_execute() Function

We can execute SQL Statements on the database using the odbc_execute() function.

Syntax:

```
odbc_execute(ODBC Handle, cSQLStatement)
```

26.11 odbc_colcount() Function

We can get columns count in the query result using the odbc_colcount() function.

Syntax:

```
odbc_colcount(ODBC Handle) ---> Columns Count as Number
```

26.12 odbc_fetch() Function

We can fetch a row from the query result using the odbc_fetch() function.

Syntax:

```
odbc_fetch(ODBC Handle)
```

26.13 odbc_getdata() Function

We can get column value from the fetched row using the odbc_getdata() function.

Syntax:

```
odbc_getdata(ODBC Handle, nColumnNumber) ---> Column Value
```

26.14 Execute Query and Print Result

The next example execute query then print the query result.

```
See "ODBC test 4" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Select data" + nl
see odbc_execute(pODBC,"select * from person") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
        See "Row data:" + nl
        for x = 1 to nMax
```

(continues on next page)

(continued from previous page)

```
                see odbc_getdata(pODBC,x) + " - "
        next
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

26.15 odbc_tables() Function

We can get a list of tables inside the database using the odbc_tables() function.

We can access the result of this function as we get any query result.

Syntax:

```
odbc_tables(ODBC Handle)
```

Example:

```
See "ODBC test - Get Database Tables" + nl
pODBC = odbc_init()
See "Connect to database" + nl
odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Select data" + nl
odbc_tables(pODBC) + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
        for x = 1 to nMax
                see odbc_getdata(pODBC,x)
                if x != nMax see " - " ok
        next
        See nl
end
See "Close database..."
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

Output:

```
ODBC test - Get Database Tables
Connect to database
Select data
Columns Count : 5
.\test - NULL - Customer - TABLE - NULL
.\test - NULL - employee - TABLE - NULL
.\test - NULL - person - TABLE - NULL
.\test - NULL - tel - TABLE - NULL
Close database...
```

26.16 odbc_columns() Function

We can get a list of columns inside the table using the odbc_columns() function.

Syntax:

```
odbc_columns(ODBC Handle, cTableName)
```

Example:

```
See "ODBC test - Get Table Columns" + nl
pODBC = odbc_init()
See "Connect to database" + nl
odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Get Columns inside the Person Table" + nl
odbc_columns(pODBC,"person") + nl
while odbc_fetch(pODBC)
        see odbc_getdata(pODBC,4) + nl
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

Output:

```
ODBC test - Get Table Columns
Connect to database
Get Columns inside the Person Table
FIRST
LAST
STREET
CITY
STATE
ZIP
HIREDATE
MARRIED
AGE
SALARY
NOTES
Close database...
```

26.17 odbc_autocommit() Function

We can enable or disable the auto commit feature using the odbc_autocommit() function.

Syntax:

```
odbc_autocommit(ODBC Handle, lStatus)   # lStatus can be True or False
```

26.18 odbc_commit() Function

We can commit updates to the database using the odbc_commit() function.

Syntax:

```
odbc_commit(ODBC Handle)
```

26.19 odbc_rollback() Function

We can rollback updates to the database using the odbc_rollback() function.

Syntax:

```
odbc_rollback(ODBC Handle)
```

26.20 Transactions and Using Commit and Rollback

Example:

```
See "ODBC Test - Transactions and using Commit and Rollback" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
see "insert data..." + nl
odbc_autocommit(pODBC,0)
for x = 1 to 10000
        odbc_execute(pODBC,"insert into tel values (" + x + ",'mahmoud')")
next
for x = 10001 to 15000
        odbc_execute(pODBC,"insert into tel values (" + x + ",'samir')")
next
odbc_commit(pODBC)

for x = 15001 to 20000
        odbc_execute(pODBC,"insert into tel values (" + x + ",'fayed')")
next

ODBC_ROLLBACK(pODBC)
odbc_execute(pODBC,"insert into tel values (" + x + ",'fayed')")
odbc_commit(pODBC)

See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

Output:

```
ODBC Test - Transactions and using Commit and Rollback
Connect to database
```

(continues on next page)

(continued from previous page)

```
1
insert data...
Close database...
```

26.21 Save and Restore images

The next example save an image inside the database

```
See "ODBC test - Save image in the database" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
see "Read Image File..." + nl
cFile = str2hex(read("tests\mahmoud.jpg"))
see "size " + len(CFile)+nl
see "Save image in the database..." + nl
stmt = "insert into tel values (20000,'mahmoud','" + cFile + "');"
odbc_execute(pODBC,stmt)
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

The next example restore the image from the database

```
See "ODBC Test - Restore image from the database" + nl
pODBC = odbc_init()
See "Connect to database" + nl
see odbc_connect(pODBC,"DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
See "Select data" + nl
see odbc_execute(pODBC,"select * from tel where id = 20000") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
if odbc_fetch(pODBC)
        See "Write image file" + nl
        write("tests\great.jpg",hex2str( odbc_getdata(pODBC,3) ) )
ok
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

CHAPTER

TWENTYSEVEN

MYSQL FUNCTIONS

In this chapter we are going to learn about the MySQL functions provided by the Ring programming language.

- MySQL_Info()
- MySQL_Init()
- MySQL_Error()
- MySQL_Connect()
- MySQL_Close()
- MySQL_Query()
- MySQL_Insert_ID()
- MySQL_Result()
- MySQL_Next_Result()
- MySQL_Columns()
- MySQL_Result2()
- MySQL_Escape_String()
- MySQL_AutoCommit()
- MySQL_Commit()
- MySQL_Rollback()

Before using the next function load the mysqllib.ring library

```
load "mysqllib.ring"
# Use MySQL functions
```

27.1 MySQL_Info() Function

We can get the MySQL Client version using the MySQL_Info() function.

Syntax:

```
MySQL_Info() ---> string contains the MySQL Client version
```

Example:

```
see "MySQL Client Version : " + mysql_info()
```

Output:

```
MySQL Client Version : 6.1.5
```

27.2 MySQL_Init() Function

We can start using MySQL Client through the MySQL_Init() function.

Syntax:

```
MySQL_Init() ---> MySQL Handle
```

27.3 MySQL_Error() Function

We can get the error message from the MySQL Client using the MySQL_Error() function.

Syntax:

```
MySQL_Error(MySQL Handle) ---> Error message as string
```

27.4 MySQL_Connect() Function

We can connect to the MySQL database server using the MySQL_Connect() function.

Syntax:

```
MySQL_Connect(MySQL Handle, cServer, cUserName, cPassword) ---> lStatus
```

27.5 MySQL_Close() Function

We can close the connection to the MySQL database using the MySQL_Close() function

Syntax:

```
MySQL_Close(MySQL Handle)
```

27.6 MySQL_Query() Function

We can execute SQL queries using the MySQL_Query() function

Syntax:

```
MySQL_Query(MySQL Handle, cSQLQuery)
```

27.7 Create Database

The next example connect to MySQL Server then create new database.

```
See "MySQL Test - Create Database" + nl
con = mysql_init()

See "Connect" + nl
if mysql_connect(con,"localhost","root","root") = 0
        see "Can't connect" + nl
        see "Error : " + mysql_error(con) + nl
        mysql_close(con)
        bye
ok

See "Create Database..." + nl
mysql_query(con,"CREATE DATABASE mahdb")

See "Close Connection" + nl
mysql_close(con)
```

Output:

```
MySQL Test - Create Database
Connect
Create Database...
Close Connection
```

27.8 Create Table and Insert Data

The next example create new table and insert records

```
func main
        see "Create Table and Insert Records" + nl
        con = mysql_init()

        see "Connect" + nl
        if mysql_connect(con, "localhost", "root", "root","mahdb") = 0
                system_error(con)
        ok
```

(continues on next page)

(continued from previous page)

```
        see "Drop table" + nl
        if mysql_query(con, "DROP TABLE IF EXISTS Employee")  system_error(con) ok

        see "Create table" + nl
        if mysql_query(con, "CREATE TABLE Employee(Id INT, Name TEXT, Salary INT)")
           system_error(con) ok

        see "Insert data" + nl
        if mysql_query(con, "INSERT INTO Employee VALUES(1,'Mahmoud',15000)")
           system_error(con) ok

        if mysql_query(con, "INSERT INTO Employee VALUES(2,'Samir',16000)")
           system_error(con) ok

        if mysql_query(con, "INSERT INTO Employee VALUES(3,'Fayed',17000)")
           system_error(con) ok

        see "Close connection" + nl
        mysql_close(con)

func system_error con
        see mysql_error(con)  mysql_close(con)  bye
```

Output:

```
Create Table and Insert Records
Connect
Drop table
Create table
Insert data
Close connection
```

27.9 MySQL_Insert_ID() Function

We can get the inserted row id using the MySQL_Insert_ID() function

Syntax:

```
MySQL_Insert_ID() ---> Inserted row id as number
```

Example:

```
con = mysql_init()
see "connect to database" + nl
mysql_connect(con,"localhost","root","root","mahdb")
see "drop table" + nl
mysql_query(con, "DROP TABLE IF EXISTS Customers")
see "create table" + nl
mysql_query(con, "CREATE TABLE Customers(Id INT PRIMARY KEY AUTO_INCREMENT, Name TEXT)")
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES('Mahmoud')")
```

(continues on next page)

(continued from previous page)

```
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES('Samir')")
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES('Fayed')")
see "insert record" + nl
mysql_query(con, "INSERT INTO Customers(Name) VALUES('Test 2015')")

see "inserted row id : " + mysql_insert_id(con) + nl
see "close database" + nl
mysql_close(con)
```

Output:

```
connect to database
drop table
create table
insert record
insert record
insert record
insert record
inserted row id : 4
close database
```

27.10 MySQL_Result() Function

We can get the query result (data without column names) using the MySQL_Result() function.

Syntax:

```
MySQL_Result(MySQL Handle) ---> List contains the query result
```

27.11 MySQL_Next_Result() Function

We can move to the next query result using the MySQL_Next_Result() function. We use this function when we have multiple SQL statements in the same query.

Syntax:

```
MySQL_Next_Result(MySQL Handle)
```

27.12 Print Query Result

The next example execute a query on the database then print the result.

```
con = mysql_init()
see "Connect to database" + nl
mysql_connect(con, "localhost", "root", "root","mahdb")
see "Execute Query" + nl
mysql_query(con, "SELECT Name FROM Employee WHERE Id=1;"+
                 "SELECT Name FROM Employee WHERE Id=3")
see "Print Result" + nl
see mysql_result(con)
mysql_next_result(con)
see mysql_result(con)
see "close database" + nl
mysql_close(con)
```

Output:

```
Connect to database
Execute Query
Print Result
Mahmoud
Fayed
close database
```

27.13 MySQL_Columns() Function

We can get a list of columns names using the MySQL_Columns() function.

Syntax:

```
MySQL_Columns(MySQL Handle) ---> List contains columns information
```

Example:

```
con = mysql_init()
see "Connect to database" + nl
mysql_connect(con, "localhost", "root", "root","mahdb")
see "Execute Query" + nl
mysql_query(con, "SELECT * FROM Employee")
see "Result" + nl
see mysql_columns(con)
see "Close database" + nl
mysql_close(con)
```

Output:

```
Connect to database
Execute Query
Result
Id
```

(continues on next page)

(continued from previous page)

```
11
3
32768
Name
65535
252
16
Salary
11
3
32768
Close database
```

27.14 MySQL_Result2() Function

Instead of using MySQL_Result() to get the result data without columns names, we can use the MySQL_Result2() to get all of the column names then the query result in one list.

Syntax:

```
MySQL_Result2(MySQL Handle) ---> List (query result starts with columns names)
```

Example:

```
con = mysql_init()
see "Connect to database" + nl
mysql_connect(con, "localhost", "root", "root","mahdb")
see "Execute Query" + nl
mysql_query(con, "SELECT * FROM Employee")
see "Print Result" + nl
see mysql_result2(con)
see "Close database" + nl
mysql_close(con)
```

Output:

```
Connect to database
Execute Query
Print Result
Id
Name
Salary
1
Mahmoud
15000
2
Samir
16000
3
Fayed
```

(continues on next page)

(continued from previous page)

```
17000
Close database
```

27.15 MySQL_Escape_String() Function

We can store binary data and special characters in the database after processing using MySQL_Escape_String() function

Syntax:

```
MySQL_Escape_String(MySQL Handle, cString) ---> String after processing
```

27.16 Save Image inside the database

Example:

```
See "Read file" + nl
cFile = read("tests\mahmoud.jpg")
con = mysql_init()
See "Connect to database..." + nl
mysql_connect(con, "localhost", "root", "root","mahdb")
See "Escape string..." + nl
cFile = mysql_escape_string(con,cFile)
stmt = "INSERT INTO photo(id, data) VALUES(1, '" + cFile + "')"
See "Insert data..." + nl
mysql_query(con,stmt)
See "Close database..." + nl
mysql_close(con)
```

Output:

```
Read file
Connect to database...
Escape string...
Insert data...
Close database...
```

27.17 Restore Image From The Database

Example:

```
con = mysql_init()
See "Connect to database..." + nl
mysql_connect(con, "localhost", "root", "root","mahdb")
See "Read data from database..." + nl
mysql_query(con,"SELECT data FROM photo WHERE id=1")
See "Write new file" + nl
result = mysql_result(con)
```

(continues on next page)

(continued from previous page)

```
write("tests\mahmoud2.jpg",result[1][1])
See "Close database..." + nl
mysql_close(con)
```

Output:

```
Connect to database...
Read data from database...
Write new file
Close database...
```

27.18 MySQL_AutoCommit() Function

We can enable or disable the auto commit feature using the MySQL_AutoCommit() function.

Syntax:

```
MySQL_AutoCommit(MySQL Handle, lStatus)  # lstatus can be True/False
```

27.19 MySQL_Commit() Function

We can commit updates to the database using the MySQL_Commit() function.

Syntax:

```
MySQL_Commit(MySQL Handle)
```

27.20 MySQL_Rollback() Function

We can rollback updates to the database using the MySQL_Rollback() function.

Syntax:

```
MySQL_Rollback(MySQL Handle)
```

27.21 Transaction Example

The next example presents the usage of MySQL_Autocommit(), MySQL_Commit() & MySQL_RollBack() functions.

Example:

```
func main

        con = mysql_init()

        see "Connect" + nl
```

(continues on next page)

(continued from previous page)

```
        if mysql_connect(con, "localhost", "root", "root","mahdb") = 0
                system_error(con) ok

        see "Drop table" + nl
        if mysql_query(con, "DROP TABLE IF EXISTS Employee2")
                system_error(con) ok

        see "Create table" + nl
        if mysql_query(con, "CREATE TABLE Employee2(Id INT, Name TEXT, Salary INT)")
                system_error(con) ok

        see "Insert data" + nl
        if mysql_query(con, "INSERT INTO Employee2 VALUES(1,'Mahmoud',15000)")
                system_error(con) ok

        if mysql_query(con, "INSERT INTO Employee2 VALUES(2,'Samir',16000)")
                system_error(con) ok

        if mysql_query(con, "INSERT INTO Employee2 VALUES(3,'Fayed',17000)")
                system_error(con) ok

        mysql_autocommit(con,False)
        mysql_query(con, "INSERT INTO Employee2 VALUES(4,'Ahmed',5000)")
        mysql_query(con, "INSERT INTO Employee2 VALUES(5,'Ibrahim',50000)")
        mysql_query(con, "INSERT INTO Employee2 VALUES(6,'Mohammed',50000)")
        See "Save transaction (y/n) " give nChoice
        if upper(nChoice) = "Y"
                mysql_commit(con)
        else
                mysql_rollback(con)
        ok

        see "Close connection" + nl
        mysql_close(con)

func system_error con

        see mysql_error(con)
        mysql_close(con)
        bye
```

Output:

```
Connect
Drop table
Create table
Insert data
Save transaction (y/n) y
Close connection
```

CHAPTER

TWENTYEIGHT

SQLITE FUNCTIONS

In this chapter we will learn about using the SQLite database in the Ring programming language.

Before using the next function load the sqlitelib.ring library

```
load "sqlitelib.ring"
# Use SQLite functions
```

28.1 sqlite_init() function

Syntax:

```
sqlite_init() ---> SQLite Object
```

28.2 sqlite_open() function

Syntax:

```
sqlite_open(SQLite Object,cFileName)
```

28.3 sqlite_execute() function

Syntax:

```
sqlite_execute(SQLite Object,cSQLStatement)
```

28.4 sqlite_close() function

Syntax:

```
sqlite_close(SQLite Object)
```

28.5 Example

The next code create a SQLite database, add new records then display the data.

```
load "sqlitelib.ring"

oSQLite = sqlite_init()

sqlite_open(oSQLite,"mytest.db")

sql = "
         CREATE TABLE COMPANY (
         ID INT PRIMARY KEY     NOT NULL,
         NAME           TEXT    NOT NULL,
         AGE            INT     NOT NULL,
         ADDRESS        CHAR(50),
         SALARY         REAL );
"
sqlite_execute(oSQLite,sql)

sql = "
        INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
        VALUES  (1, 'Mahmoud' , 29, 'Jeddah', 20000.00 ),
                (2, 'Ahmed'   , 27, 'Jeddah', 15000.00 ),
                (3, 'Mohammed', 31, 'Egypt' , 20000.00 ),
                (4, 'Ibrahim' , 24, 'Egypt ', 65000.00 );
"

sqlite_execute(oSQLite,sql)

aResult =  sqlite_execute(oSQLite,"select * from COMPANY")
for x in aResult
        for t in x
                ? t[2] + nl
        next
next
? copy("*",50)
for x in aResult
        ? x[:name]
next
sqlite_close(oSQLite)
```

Output:

```
1
Mahmoud
29
Jeddah
20000.0
2
Ahmed
27
Jeddah
```

(continues on next page)

(continued from previous page)

```
15000.0
3
Mohammed
31
Egypt
20000.0
4
Ibrahim
24
Egypt
65000.0
**************************************************
Mahmoud
Ahmed
Mohammed
Ibrahim
```

CHAPTER

TWENTYNINE

POSTGRESQL FUNCTIONS

In this chapter we will learn about using the PostgreSQL database in the Ring programming language.

29.1 Loading the library

Before using the next function load the postgresqllib.ring library

```
load "postgresqllib.ring"
# Use PostgreSQL functions
```

29.2 Examples

Example (1):

```
load "postgresqllib.ring"

conninfo = "user=postgres password=sa dbname = postgres"

exit_nicely = func conn {
        PQfinish(conn)
        shutdown(1)
}

conn = PQconnectdb(conninfo)

if (PQstatus(conn) != CONNECTION_OK)
        fputs(stderr, "Connection to database failed: "+PQerrorMessage(conn))
                call exit_nicely(conn)
ok

res = PQexec(conn, "select * from pg_database")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Select failed: " + PQerrorMessage(conn))
        PQclear(res)
        exit_nicely(conn)
ok

nFields = PQnfields(res)
```

(continues on next page)

(continued from previous page)

```
for i = 1 to nFields
                ? PQfname(res, i-1)
next

? copy("*",60)

for i = 1 to PQntuples(res)
        for j=1 to nFields
                see PQgetvalue(res, i-1, j-1) + " "
        next
        see nl
next

PQclear(res)

PQfinish(conn)
```

Output:

```
datname
datdba
encoding
datcollate
datctype
datistemplate
datallowconn
datconnlimit
datlastsysoid
datfrozenxid
datminmxid
dattablespace
datacl
************************************************************
postgres 10 6 English_United States.1252
        English_United States.1252 f t -1 12937 549 1 1663
template1 10 6 English_United States.1252 English_United States.1252
        t t -1 12937 549 1 1663 {=c/postgres,postgres=CTc/postgres}
template0 10 6 English_United States.1252 English_United States.1252
        t f -1 12937 549 1 1663 {=c/postgres,postgres=CTc/postgres}
mahdb 10 6 English_United States.1252 English_United States.1252
        f t -1 12937 549 1 1663
```

Example(2):

```
load "postgresqllib.ring"

conninfo = "user=postgres password=sa"

exit_nicely = func conn {
        PQfinish(conn)
        shutdown(1)
}
```

(continues on next page)

(continued from previous page)

```
conn = PQconnectdb(conninfo)

if (PQstatus(conn) != CONNECTION_OK)
        fputs(stderr, "Connection to database failed: "+PQerrorMessage(conn))
                call exit_nicely(conn)
ok

res = PQexec(conn, "DROP DATABASE IF EXISTS mahdb;")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Remove failed: " + PQerrorMessage(conn))
ok
PQclear(res)


res = PQexec(conn, "CREATE DATABASE mahdb;")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Create database failed: " + PQerrorMessage(conn))
ok
PQclear(res)


res = PQexec(conn, "
CREATE TABLE COMPANY (
                ID INT PRIMARY KEY     NOT NULL,
                NAME           TEXT    NOT NULL,
                AGE            INT     NOT NULL,
                ADDRESS        CHAR(50),
                SALARY         REAL );
")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Create Table failed: " + PQerrorMessage(conn))
ok
PQclear(res)

res = PQexec(conn, "
                INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
                VALUES  (1, 'Mahmoud' , 31, 'Jeddah', 10.00 ),
                                (2, 'Ahmed'   , 27, 'Jeddah', 20.00 ),
                                (3, 'Mohammed', 33, 'Egypt' , 30.00 ),
                                (4, 'Ibrahim' , 24, 'Egypt ', 40.00 );
")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Insert Table failed: " + PQerrorMessage(conn))
ok
PQclear(res)

res = PQexec(conn, "
           select * from COMPANY
")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Select failed: " + PQerrorMessage(conn))
```

(continues on next page)

(continued from previous page)

```
        PQclear(res)
        call exit_nicely(conn)
ok


nFields = PQnfields(res)
for i = 1 to nFields
                ? PQfname(res, i-1)
next

? copy("*",60)

for i = 1 to PQntuples(res)
        for j=1 to nFields
                see PQgetvalue(res, i-1, j-1) + " "
        next
        see nl
next

PQclear(res)

PQfinish(conn)
```

Output:

```
id
name
age
address
salary
************************************************************
1 Mahmoud  31 Jeddah  10
2 Ahmed    27 Jeddah  20
3 Mohammed 31 Egypt   30
4 Ibrahim  24 Egypt   40
```

29.3 RingPostgreSQL Constants

The next constants are define by the RingPostgreSQL Library

```
CONNECTION_STARTED
CONNECTION_MADE
CONNECTION_AWAITING_RESPONSE
CONNECTION_AUTH_OK
CONNECTION_SSL_STARTUP
CONNECTION_SETENV
CONNECTION_OK

PQPING_OK
PQPING_REJECT
```

(continues on next page)

(continued from previous page)

```
PQPING_NO_RESPONSE
PQPING_NO_ATTEMPT

PGRES_EMPTY_QUERY
PGRES_COMMAND_OK
PGRES_TUPLES_OK
PGRES_COPY_OUT
PGRES_COPY_IN
PGRES_BAD_RESPONSE
PGRES_NONFATAL_ERROR
PGRES_FATAL_ERROR
PGRES_COPY_BOTH
PGRES_SINGLE_TUPLE

PG_DIAG_SEVERITY
PG_DIAG_SQLSTATE
PG_DIAG_MESSAGE_PRIMARY
PG_DIAG_MESSAGE_DETAIL
PG_DIAG_MESSAGE_HINT
PG_DIAG_STATEMENT_POSITION
PG_DIAG_INTERNAL_POSITION
PG_DIAG_INTERNAL_QUERY
PG_DIAG_CONTEXT
PG_DIAG_SCHEMA_NAME
PG_DIAG_TABLE_NAME
PG_DIAG_COLUMN_NAME
PG_DIAG_DATATYPE_NAME
PG_DIAG_CONSTRAINT_NAME
PG_DIAG_SOURCE_FILE
PG_DIAG_SOURCE_LINE
PG_DIAG_SOURCE_FUNCTION
```

29.4 RingPostgreSQL Functions

The next functions are define by the RingPostgreSQL Library

Reference : https://www.postgresql.org/docs/9.1/static/libpq.html

```
PGconn *PQconnectdbParams(const char **keywords,
        const char **values,int expand_dbname);
PGconn *PQconnectdb(const char *conninfo)
PGconn *PQsetdbLogin(const char *pghost,const char *pgport,
        const char *pgoptions,const char *pgtty,
        const char *dbName,const char *login,const char *pwd)
PGconn *PQsetdb(char *pghost,char *pgport,char *pgoptions,
        char *pgtty,char *dbName)
PGconn *PQconnectStartParams(const char **keywords,
        const char **values,int expand_dbname)
PGconn *PQconnectStart(const char *conninfo)
PostgresPollingStatusType PQconnectPoll(PGconn *conn)
PQconninfoOption *PQconndefaults(void)
```

(continues on next page)

(continued from previous page)

```
PQconninfoOption *PQconninfo(PGconn *conn)
PQconninfoOption *PQconninfoParse(const char *conninfo, char **errmsg)
void PQfinish(PGconn *conn)
void PQreset(PGconn *conn)
int PQresetStart(PGconn *conn)
PostgresPollingStatusType PQresetPoll(PGconn *conn)
PGPing PQpingParams(const char **keywords,const char **values,
        int expand_dbname)
PGPing PQping(const char *conninfo)
char *PQdb(const PGconn *conn)
char *PQuser(const PGconn *conn)
char *PQpass(const PGconn *conn)
char *PQhost(const PGconn *conn)
char *PQport(const PGconn *conn)
char *PQtty(const PGconn *conn)
char *PQoptions(const PGconn *conn)
ConnStatusType PQstatus(const PGconn *conn)
PGTransactionStatusType PQtransactionStatus(const PGconn *conn)
const char *PQparameterStatus(const PGconn *conn, const char *paramName)
int PQprotocolVersion(const PGconn *conn)
int PQserverVersion(const PGconn *conn)
char *PQerrorMessage(const PGconn *conn)
int PQsocket(const PGconn *conn)
int PQbackendPID(const PGconn *conn)
int PQconnectionNeedsPassword(const PGconn *conn)
int PQconnectionUsedPassword(const PGconn *conn)
int PQsslInUse(const PGconn *conn)
const char *PQsslAttribute(const PGconn *conn, const char *attribute_name)
const char **PQsslAttributeNames(const PGconn *conn)
void *PQsslStruct(const PGconn *conn, const char *struct_name)
void *PQgetssl(const PGconn *conn)
PGresult *PQexec(PGconn *conn, const char *command);
PGresult *PQexecParams(PGconn *conn,const char *command,int nParams,
        const Oid *paramTypes,const char **paramValues,
        const int *paramLengths,const int *paramFormats,int resultFormat)
PGresult *PQprepare(PGconn *conn,const char *stmtName,
        const char *query,int nParams,const Oid *paramTypes)
PGresult *PQexecPrepared(PGconn *conn,const char *stmtName,
        int nParams,const char **paramValues,
        const int *paramLengths,const int *paramFormats,int resultFormat)
PGresult *PQdescribePrepared(PGconn *conn, const char *stmtName)
PGresult *PQdescribePortal(PGconn *conn, const char *portalName)
ExecStatusType PQresultStatus(const PGresult *res)
char *PQresStatus(ExecStatusType status)
char *PQresultErrorMessage(const PGresult *res)
char *PQresultErrorField(const PGresult *res, int fieldcode)
void PQclear(PGresult *res)
int PQntuples(const PGresult *res)
int PQnfields(const PGresult *res)
char *PQfname(const PGresult *res,int column_number)
int PQfnumber(const PGresult *res,const char *column_name)
Oid PQftable(const PGresult *res,int column_number)
```

(continues on next page)

(continued from previous page)

```
int PQftablecol(const PGresult *res,int column_number)
int PQfformat(const PGresult *res,int column_number)
Oid PQftype(const PGresult *res,int column_number)
int PQfmod(const PGresult *res,int column_number)
int PQfsize(const PGresult *res,int column_number)
int PQbinaryTuples(const PGresult *res)
char *PQgetvalue(const PGresult *res,int row_number,int column_number)
int PQgetisnull(const PGresult *res,int row_number,int column_number)
int PQgetlength(const PGresult *res,int row_number,int column_number)
int PQnparams(const PGresult *res)
Oid PQparamtype(const PGresult *res, int param_number)
void PQprint(FILE *fout,const PGresult *res,const PQprintOpt *po)
char *PQcmdStatus(PGresult *res)
char *PQcmdTuples(PGresult *res)
Oid PQoidValue(const PGresult *res)
char *PQoidStatus(const PGresult *res)
char *PQescapeLiteral(PGconn *conn, const char *str, size_t length)
char *PQescapeIdentifier(PGconn *conn, const char *str, size_t length)
size_t PQescapeStringConn(PGconn *conn,char *to,
        const char *from, size_t length,int *error)
size_t PQescapeString(char *to, const char *from, size_t length)
unsigned char *PQescapeByteaConn(PGconn *conn,
        const unsigned char *from,size_t from_length,size_t *to_length)
unsigned char *PQescapeBytea(const unsigned char *from,
        size_t from_length,size_t *to_length)
unsigned char *PQunescapeBytea(const unsigned char *from, size_t *to_length)
int PQsendQuery(PGconn *conn, const char *command)
int PQsendQueryParams(PGconn *conn,const char *command,
        int nParams,const Oid *paramTypes,const char **paramValues,
        const int *paramLengths,const int *paramFormats,int resultFormat)
int PQsendPrepare(PGconn *conn,const char *stmtName,
        const char *query,int nParams,const Oid *paramTypes)
int PQsendQueryPrepared(PGconn *conn,const char *stmtName,
        int nParams,const char **paramValues,
        const int *paramLengths,const int *paramFormats,int resultFormat)
int PQsendDescribePrepared(PGconn *conn, const char *stmtName)
int PQsendDescribePortal(PGconn *conn, const char *portalName)
PGresult *PQgetResult(PGconn *conn)
int PQconsumeInput(PGconn *conn)
int PQisBusy(PGconn *conn)
int PQsetnonblocking(PGconn *conn, int arg)
int PQisnonblocking(const PGconn *conn)
int PQflush(PGconn *conn)
int PQsetSingleRowMode(PGconn *conn)
PGcancel *PQgetCancel(PGconn *conn)
void PQfreeCancel(PGcancel *cancel)
int PQcancel(PGcancel *cancel, char *errbuf, int errbufsize)
int PQrequestCancel(PGconn *conn)
PGresult *PQfn(PGconn *conn,int fnid,int *result_buf,
        int *result_len,int result_is_int,const PQArgBlock *args,int nargs)
PGnotify *PQnotifies(PGconn *conn)
int PQputCopyData(PGconn *conn,const char *buffer,int nbytes)
```

(continues on next page)

(continued from previous page)

```
int PQputCopyEnd(PGconn *conn,const char *errormsg)
int PQgetCopyData(PGconn *conn,char **buffer,int async)
int PQgetline(PGconn *conn,char *buffer,int length)
int PQgetlineAsync(PGconn *conn,char *buffer,int bufsize)
int PQputline(PGconn *conn,const char *string)
int PQputnbytes(PGconn *conn,const char *buffer,int nbytes)
int PQendcopy(PGconn *conn)
int PQclientEncoding(const PGconn *conn)
char *pg_encoding_to_char(int encoding_id)
int PQsetClientEncoding(PGconn *conn, const char *encoding)
void PQtrace(PGconn *conn, FILE *stream)
void PQuntrace(PGconn *conn)
void PQfreemem(void *ptr)
void PQconninfoFree(PQconninfoOption *connOptions)
char *PQencryptPasswordConn(PGconn *conn, const char *passwd,
         const char *user, const char *algorithm)
char *PQencryptPassword(const char *passwd, const char *user)
PGresult *PQmakeEmptyPGresult(PGconn *conn, ExecStatusType status)
int PQfireResultCreateEvents(PGconn *conn, PGresult *res)
PGresult *PQcopyResult(const PGresult *src, int flags)
int PQsetResultAttrs(PGresult *res, int numAttributes, PGresAttDesc *attDescs)
int PQsetvalue(PGresult *res, int tup_num, int field_num,
                char *value, int len)
void *PQresultAlloc(PGresult *res, size_t nBytes)
int PQlibVersion(void)
PQnoticeReceiver PQsetNoticeReceiver(PGconn *conn,
                PQnoticeReceiver proc,void *arg)
PQnoticeProcessor PQsetNoticeProcessor(PGconn *conn,
                PQnoticeProcessor proc,void *arg)
void PQinitOpenSSL(int do_ssl, int do_crypto)
void PQinitSSL(int do_ssl)
int PQisthreadsafe(void)
```

CHAPTER

THIRTY

SECURITY AND INTERNET FUNCTIONS

This chapter contains the security and internet functions provided by the Ring programming language for Hashing, Encryption & Decryption.

Before using the next functions load the openssllib.ring library

```
load "openssllib.ring"
# Use OpenSSL functions
```

- MD5()
- SHA1()
- SHA256()
- SHA512()
- SHA384()
- SHA224()
- SupportedCiphers()
- Encrypt()
- Decrypt()
- Randbytes()
- rsa_generate
- rsa_export_params
- rsa_import_params
- rsa_export_pem
- rsa_import_pem
- rsa_is_privatekey
- rsa_encrypt_pkcs
- rsa_decrypt_pkcs
- rsa_encrypt_oaep
- rsa_decrypt_oaep
- rsa_encrypt_raw
- rsa_decrypt_raw
- rsa_sign_pkcs

- rsa_signhash_pkcs
- rsa_verify_pkcs
- rsa_verifyhash_pkcs
- rsa_sign_pss
- rsa_signhash_pss
- rsa_verify_pss
- rsa_verifyhash_pss
- openssl_versiontext
- openssl_version
- MD5Init(), MD5Update(), MD5Final()
- SHA1Init(), SHA1Update(), SHA1Final()
- SHA256Init(), SHA256Update(), SHA256Final()
- SHA512Init(), SHA512Update(), SHA512Final()
- SHA384Init(), SHA384Update(), SHA384Final()
- SHA224Init(), SHA224Update(), SHA224Final()

Before using the next functions load the internetlib.ring library

```
load "internetlib.ring"
# Use the Internet functions
```

- Download()
- SendEmail()

30.1 MD5() Function

We can calculate the MD5 hash using the MD5() Function

Syntax:

```
MD5(cString) ---> String contains the MD5 hash of the string cString
```

Example:

```
see "md5('happy') = " + md5("happy") + nl +
    "md5('Hello') = " + md5("Hello") + nl
```

Output:

```
md5('happy') = 56ab24c15b72a457069c5ea42fcfc640
md5('Hello') = 8b1a9953c4611296a827abf8c47804d7
```

30.2 SHA1() Function

We can calculate the SHA1 hash using the SHA1() Function

Syntax:

```
SHA1(cString) ---> String contains the SHA1 hash of the string cString
```

Example:

```
see "sha1('hello') : " + sha1("hello") + nl +
    "sha1('apple') : " + sha1("apple") + nl
```

Output:

```
sha1('hello') : aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d
sha1('apple') : d0be2dc421be4fcd0172e5afceea3970e2f3d940
```

30.3 SHA256() Function

We can calculate the SHA256 hash using the SHA256() Function

Syntax:

```
SHA256(cString) ---> String contains the SHA256 hash of the string cString
```

Example:

```
see "sha256('hello') : " + sha256("hello") + nl +
    "sha256('apple') : " + sha256("apple") + nl
```

Output:

```
sha256('hello') : 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824
sha256('apple') : 3a7bd3e2360a3d29eea436fcfb7e44c735d117c42d1c1835420b6b9942dd4f1b
```

30.4 SHA512() Function

We can calculate the SHA512 hash using the SHA512() Function

Syntax:

```
SHA512(cString) ---> String contains the SHA512 hash of the string cString
```

Example:

```
see "sha512('hello') : " + sha512("hello") + nl +
    "sha512('apple') : " + sha512("apple") + nl +
    "sha512('hello world') : " + sha512("hello world") + nl
```

Output:

```
sha512('hello') : 9b71d224bd62f3785d96d46ad3ea3d73319bfbc2890caadae2dff72519673c
a72323c3d99ba5c11d7c7acc6e14b8c5da0c4663475c2e5c3adef46f73bcdec043
sha512('apple') : 844d8779103b94c18f4aa4cc0c3b4474058580a991fba85d3ca698a0bc9e52
c5940feb7a65a3a290e17e6b23ee943ecc4f73e7490327245b4fe5d5efb590feb2
sha512('hello world') : 309ecc489c12d6eb4cc40f50c902f2b4d0ed77ee511a7c7a9bcd3ca8
6d4cd86f989dd35bc5ff499670da34255b45b0cfd830e81f605dcf7dc5542e93ae9cd76f
```

30.5 SHA384() Function

We can calculate the SHA384 hash using the SHA384() Function

Syntax:

```
SHA384(cString) ---> String contains the SHA384 hash of the string cString
```

Example:

```
see "sha384('hello') : " + sha384("hello") + nl +
    "sha384('apple') : " + sha384("apple") + nl +
    "sha384('hello world') : " + sha384("hello world") + nl
```

Output:

```
sha384('hello') : 59e1748777448c69de6b800d7a33bbfb9ff1b463e44354c3553bcdb9c666fa
90125a3c79f90397bdf5f6a13de828684f
sha384('apple') : 3d8786fcb588c93348756c6429717dc6c374a14f7029362281a3b21dc10250
ddf0d0578052749822eb08bc0dc1e68b0f
sha384('hello world') : fdbd8e75a67f29f701a4e040385e2e23986303ea10239211af907fcb
b83578b3e417cb71ce646efd0819dd8c088de1bd
```

30.6 SHA224() Function

We can calculate the SHA224 hash using the SHA224() Function

Syntax:

```
SHA224(cString) ---> String contains the SHA224 hash of the string cString
```

Example:

```
see "sha224('hello') : " + sha224("hello") + nl +
    "sha224('apple') : " + sha224("apple") + nl +
    "sha224('hello world') : " + sha224("hello world") + nl
```

Output:

```
sha224('hello') : ea09ae9cc6768c50fcee903ed054556e5bfc8347907f12598aa24193
sha224('apple') : b7bbfdf1a1012999b3c466fdeb906a629caa5e3e022428d1eb702281
sha224('hello world') : 2f05477fc24bb4faefd86517156dafdecec45b8ad3cf2522a563582b
```

30.7 SupportedCiphers() Function

The function SupportedCiphers() retrieves the list of all algorithms supported by Encrypt()/Decrypt() functions.

Syntax:

```
SupportedCiphers() ---> List of names of cipher algorithms supported by Encrypt()/
↪Decrypt() functions
```

30.8 Encrypt() Function

We can use the Encrypt() function to encrypt the data using the specified algorithm. If no algorithm is specified, Blowfish algorithm is used in CBC mode. Typical algorithm values: "bf", "des", "des3", "aes128", "aes192", "aes256" which all use CBC mode. The function CipherAlgorithms() return the list of all supported cipher algorithms.

Syntax:

```
Encrypt(cString, cKey, cIV[, cCipherAlgorithmName]) ---> Encrypted string
```

30.9 Decrypt() Function

We can use the Decrypt() function to decrypt the data encrypted using the Encrypt() function. If no algorithm is specified, Blowfish algorithm is used in CBC mode. Typical algorithm values: "bf", "des", "des3", "aes128", "aes192", "aes256" which all use CBC mode. The function CipherAlgorithms() return the list of all supported cipher algorithms.

Syntax:

```
Decrypt(cCipher, cKey, cIV[, cCipherAlgorithm]) ---> Decrypted string
```

30.10 Encryption and Decryption Example

The next example demonstrates how to use the Encrypt() and Decrypt() functions.

These functions use the AES-128 algorithm (AES with 128-bits key)

```
See "Enter a string : " give cStr
list = 0:15  cKey=""    for x in list cKey += char(x) next
list = 1:16  cIV = ""   for x in list cIV += char(x) next
cStr = Encrypt(cStr,cKey,cIV,"aes128")
See "Cipher Text    : " + cStr + nl +
    "Plain Text     : " + Decrypt(cStr,cKey,cIV,"aes128") + nl
```

We can write the same example using normal for loop

```
See "Enter a string : " give cStr

cKey=""                         # 16 bytes
for x = 0 to 15
        cKey += char(x)
```

(continues on next page)

(continued from previous page)

```
next

cIV = ""
for x = 1 to 16
        cIV += char(x)
next


cStr = Encrypt(cStr,cKey,cIV,"aes128")
See "Cipher Text    : " + cStr + nl +
    "Plain Text     : " + Decrypt(cStr,cKey,cIV,"aes128") + nl
```

Also we can write the key and the IV directly using strings

```
See "Enter a string : " give cStr

# Note: Don't use simple key in real applications!
cKey = "1234567890@#$%^&"
cIV  = "FEDCBA0987654321"

cStr = Encrypt(cStr,cKey,cIV,"aes128")
See "Cipher Text    : " + cStr + nl +
    "Plain Text     : " + Decrypt(cStr,cKey,cIV,"aes128") + nl
```

Finally we can specify the key and the IV values using hexadecimal notation

```
See "Enter a string : " give cStr

# Note: Don't use simple key in real applications!
cKey = hex2str("A0A1A2A3A5A6A7A8AAABACADAFB0B1B2")
cIV  = hex2str("00112233445566778899AABBCCDDEEFF")

cStr = Encrypt(cStr,cKey,cIV,"aes128")
See "Cipher Text    : " + cStr + nl +
    "Plain Text     : " + Decrypt(cStr,cKey,cIV,"aes128") + nl
```

30.11 File Hash

The next example demonstrates how to calculate the hash functions for files

```
cStr = read("myapp.exe")
see "Size : " + len(cStr) + nl +
    "md5 : " + md5(cStr) + nl +
    "sha1 : " + sha1(cStr) + nl +
    "sha256 : " + sha256(cStr) + nl +
    "sha224 : " + sha224(cStr) + nl +
    "sha384 : " + sha384(cStr) + nl +
    "sha512 : " + sha512(cStr) + nl
```

Output:

```
Size : 58079876
md5 : 762eee15d8d2fd73b71ea52538b28667
sha1 : 9212c0c7258bad89a62bd239e1358a9276a9d070
sha256 : 7d6724e69b6c553da749ba31b6185dddc965129b64d9e9bf3de88f67df3b1cdc
sha224 : 5a9c8a7d662bce4f880ba94f90a79362b672528b9efd5abc718c7a3d
sha384 : 18e23f973abedbeb3981c423f12aeadecf96f9c6fb28aeabe3be4c484f8540afcc3861b
b370ce2b59cf3c99c130b856b
sha512 : da3d5e997d06f8b2a7a9964b77f7d82eedb76b245c611082c1639f83f51d83880bcd08f
cd53dcab1167bdca0b82fec5071971ac17c76479d76985ced4ab0d18e
```

30.12 Randbytes() Function

We can generate a string of cryptographically secure pseudo-random bytes using the Randbytes() function.

Syntax:

```
Randbytes(nSize) ---> String contains random bytes (bytes count = nSize)
```

Example:

```
salt =  randbytes(32)
password = "SecretPassWord@$%123"
see salt + nl
see sha256("test" + salt) + nl
```

30.13 rsa_generate() Function

We can generate a random RSA key pair using the rsa_generate() function.

Syntax:

```
rsa_generate(nBits[,nPublicExponent]) ---> a random RSA key pair with nBits as size in␣
↪bits
                If nPublicExponent is omited, then the standard public exponent value␣
↪65537 is used.
```

Example:

```
/* generate a new 2048-bit RSA key pair */
try
        rsaKey = rsa_generate(2048)
        rsaKeyParams = rsa_export_params(rsaKey)
        See "Modulus = " + rsaKeyParams[:n] + nl
catch
        See "Failed to generate the RSA key pair: " + cCatchError + nl
done
```

30.14 rsa_export_params() Function

We can export the parameters of an RSA key to a string-indexed list using the rsa_export_params() function. The list contains the following string indexes:

- "type" for the key type as a string equal to "RSA" in our case
- "bits" for the bot length of the key as an integer
- "n" for the Modulus as a hexadecimal string
- "e" for the Public Exponent as a hexadecimal string
- "d" for the Private Exponent as a hexadecimal string
- "p" for the first prime as a hexadecimal string
- "q" for the second prime as a hexadecimal string
- "dmp1" for the first CRT exponent as a hexadecimal string
- "dmq1" for the second CRT exponent as a hexadecimal string
- "iqmp" for the CRT coefficent as a hexadecimal string

If the key contains only the public part, then "d", "p", "q", "dmp1", "dmq1" and "iqmp" will be empty strings.

Syntax:

```
rsa_export_params(pRsaKey) ---> list of the key parameters
```

Example:

```
/* generate a new 2048-bit RSA key pair */
try
        rsaKey = rsa_generate(2048)
        rsaKeyParams = rsa_export_params(rsaKey)
        See "Key Type = " + rsaKeyParams[:type] + nl
        See "Key Size = " + rsaKeyParams[:bits] + " bits" + nl
        See "Modulus = " + rsaKeyParams[:n] + nl
        See "Public Exponent = " + rsaKeyParams[:e] + nl
        See "Private Exponent = " + rsaKeyParams[:d] + nl
        See "Prime 1 = " + rsaKeyParams[:p] + nl
        See "Prime 2 = " + rsaKeyParams[:q] + nl
        See "CRT Exponent 1 = " + rsaKeyParams[:dmp1] + nl
        See "CRT Exponent 2 = " + rsaKeyParams[:dmq1] + nl
        See "CRT Coefficient = " + rsaKeyParams[:iqmp] + nl
catch
        See "Failed to generate the RSA key pair: " + cCatchError + nl
done
```

30.15 rsa_import_params() Function

We can create a new RSA key from parameters stored in a string-indexed list using the rsa_import_params() function. The format of the input list is the one described in the function rsa_export_params

The indexes "n" and "e" must not be empty, otherwise an exception is thrown. If we need to import only an RSA public key, then the indexes "d", "p", "q", "dmp1", "dmq1" and "iqmp" must be empty.

Syntax:

```
rsa_import_params(pParamsList) ---> a new RSA key
```

Example:

```
/* create an RSA public key from a generated RSA key pair */
try
        rsaKey = rsa_generate(2048)
        rsaKeyParams = rsa_export_params(rsaKey)

        /* create parameters of public key: modulus and public exponent */
        rsaPublicKeyParam = [:n = rsaKeyParams[:n], :e = rsaKeyParams[:e]]
        /* create the public key using rsa_import_params */
        rsaPublicKey = rsa_import_params(rsaPublicKeyParam)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.16 rsa_export_pem() Function

We can export an RSA key to a string in PEM format using the rsa_export_pem() function. If the RSA key contains both public and private parts, then returned string will start with "——BEGIN PRIVATE KEY——" If the RSA key contains only the public part, then returned string will start with "——BEGIN PUBLIC KEY——"

Syntax:

```
rsa_export_pem(pRsaKey) ---> string encoding of the key in PEM format
```

Example:

```
/* generate an RSA key and save it to a file in PEM format */
try
        rsaKey = rsa_generate(2048)
        rsaKeyPEM = rsa_export_pem(rsaKey)
        /* save private key to a file */
        write ("privateKey.pem", rsaKeyPEM)

        /* save public key to a file */
        rsaKeyParams = rsa_export_params(rsaKey)
        rsaPublicKeyParam = [:n = rsaKeyParams[:n], :e = rsaKeyParams[:e]]
        rsaPublicKey = rsa_import_params(rsaPublicKeyParam)
        rsaPublicKeyPEM = rsa_export_pem(rsaPublicKey)
        write ("publicKey.pem", rsaPublicKeyPEM)
```

(continues on next page)

(continued from previous page)

```
catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.17 rsa_import_pem() Function

We can create an RSA key from PEM encoded string using the rsa_import_pem() function. If the PEM string starts with "——BEGIN PRIVATE KEY——", then a full RSA key pair will be created. if the PEM string starts with "——BEGIN PUBLIC KEY——", then an RSA public key will be created.

Syntax:

```
rsa_import_pem(cStrPEM) ---> a new RSA key
```

Example:

```
/* create an RSA key from a PEM file */
try
        rsaKeyPEM = Read("privateKey.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        rsaPublicKeyPEM = Read("publicKey.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)
catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.18 rsa_is_privatekey() Function

We can check whether an RSA key is a private key or public key using the rsa_is_privatekey() function.

Syntax:

```
rsa_is_privatekey(pRsaKey) ---> returns 1 if pRsaKey is an RSA private key and 0 if it␣
↪is an RSA public key
```

Example:

```
/* create an RSA key from a PEM file and check if it is a private key */
try
        rsaKeyPEM = Read("key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        if rsa_is_privatekey(rsaKey)
                See "an RSA private key was loaded" + nl
        else
                See "an RSA public key was loaded" + nl
        ok
```

(continues on next page)

(continued from previous page)

```
catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.19 rsa_encrypt_pkcs() Function

We can encrypt data with an RSA key and PKCS#1 v1.5 padding using the rsa_encrypt_pkcs() function. The maximum size of data that can be encrypted by rsa_encrypt_pkcs is (modulusLen - 11), with modulusLen the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the maximum size of data that can be encrypted is 256 - 11 = 245 bytes. RSA encryption is usually applied to a symmetric key (e.g. AES) which is used to encrypt much larger data. RSA encryption needs only the public part of an RSA key, so rsa_encrypt_pkcs can be used with both RSA private key and RSA public key

Syntax:

```
rsa_encrypt_pkcs(pRsaKey,cPlainData) ---> return a string containing the encryption of␣
↪cPlainData
```

Example:

```
/* encrypt a file using AES key and then encrypt the AES key using an RSA public key */
try
        /* read Alice public key */
        rsaPublicKeyPEM = Read("alice_public_key.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)

        /* encrypt file with random AES-128 key */
        cData = Read ("secret_document.txt")
        cKey = RandBytes(16)
        cIV = RandBytes(16)
        cEncryptedData = Encrypt(cData,cKey,cIV,"aes128")

        /* encrypt the AES-128 key with the RSA public key */
        cEncryptedKey = rsa_encrypt_pkcs(rsaPublicKey,cKey)

        /* store IV, encrypted AES key and encrypted data in a file to be sent to Alice*/
        Write("encrypted_document.enc", cIV + cEncryptedKey + cEncryptedData)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.20 rsa_decrypt_pkcs() Function

We can decrypt data encrypted with an RSA key and PKCS#1 v1.5 padding using the rsa_decrypt_pkcs() function. The size of data that can be decrypted by rsa_decrypt_pkcs must be equal to modulusLen which is the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of encrypted data that can be decrypted must be 256 bytes. For RSA decryption, the RSA key must contain the private key part.

Syntax:

```
rsa_decrypt_pkcs(pRsaKey,cEncryptedData) ---> return a string containing the decryption␣
↪of cEncryptedData
```

Example:

```
/* decrypt a file by first decrypting AES key that was used to encrypt it
 * and then decrypt the whole content using the AES key
 */
try
        /* read Alice private key */
        rsaKeyPEM = Read("alice_private_key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        /* calculate the modulus length */
        rsaKeyParams = rsa_export_params(rsaKey)
        modulusLen = rsaKeyParams[:bits]/ 8

        /* read encrypted file */
        cEncryptedContent = Read ("encrypted_document.enc")

        /* IV is the first 16 bytes if the file */
        cIV = substr(cEncryptedContent, 1, 16)

        /* encrypted key follows IV and its length is modulusLen */
        cEncryptedKey = substr(cEncryptedContent, 17, modulusLen)

        /* encrypted data follows the key */
        cEncryptedData = substr(cEncryptedContent, 17 + modulusLen)

        /* decrypt the AES-128 key */
        cKey = rsa_decrypt_pkcs(rsaKey,cEncryptedKey)

        /* decrypt the data using the AES-128 key */

        cPlainData = Decrypt(cEncryptedData,cKey,cIV,"aes128")

        /* store the decrypted data to a file */
        Write("decrypted_document.txt", cPlainData)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.21 rsa_encrypt_oaep() Function

We can encrypt data with an RSA key and OAEP padding using the rsa_encrypt_oaep() function. The maximum size of data that can be encrypted by rsa_encrypt_oaep is (modulusLen - 2*hashLen -2), with modulusLen the length of the RSA key modulus in bytes and hashLen and the length of hash algorithm used. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the maximum size of data that can be encrypted using OAEP padding with SHA-1 is 256 - 2*20 - 2 = 214 bytes. RSA encryption is usually applied to a symmetric key (e.g. AES) which is used to encrypt much larger data. RSA encryption needs only the public part of an RSA key, so rsa_encrypt_oaep can be used with both RSA private key and RSA public key.

Syntax:

```
rsa_encrypt_oaep(pRsaKey,cPlainData[,nHashAlgorithm]) ---> return a string containing␣
↪the OAEP encryption of cPlainData
        nHashAlgorithm indicates the hash algorithm to use for OAEP padding. If omited,␣
↪SHA-1 is used by default.
        Possible values for nHashAlgorithm argument are:
         - $OSSL_HASH_MD5 which is equal to 0
         - $OSSL_HASH_SHA1 which is equal to 1
         - $OSSL_HASH_SHA256 which is equal to 2
         - $OSSL_HASH_SHA384 which is equal to 3
         - $OSSL_HASH_SHA512 which is equal to 4
```

Example:

```
/* encrypt a file using AES key and then encrypt the AES key using an RSA public key␣
↪using OAEP padding */
try
        /* read Alice public key */
        rsaPublicKeyPEM = Read("alice_public_key.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)

        /* encrypt file with random AES-128 key */
        cData = Read ("secret_document.txt")
        cKey = RandBytes(16)
        cIV = RandBytes(16)
        cEncryptedData = Encrypt(cData,cKey,cIV,"aes128")

        /* encrypt the AES-128 key with the RSA public key */
        cEncryptedKey = rsa_encrypt_oaep(rsaPublicKey,cKey)

        /* store IV, encrypted AES key and encrypted data in a file to be sent to Alice*/
        Write("oaep_encrypted_document.enc", cIV + cEncryptedKey + cEncryptedData)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.22 rsa_decrypt_oaep() Function

We can decrypt data encrypted with an RSA key and OAEP padding using the rsa_decrypt_oaep() function. The size of data that can be decrypted by rsa_decrypt_oaep must be equal to modulusLen which is the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of encrypted data that can be decrypted must be 256 bytes. For RSA decryption, the RSA key must contain the private key part. The hash algorithm specified in rsa_decrypt_oaep() call must be the same as the one used during OAEP encryption.

Syntax:

```
rsa_decrypt_oaep(pRsaKey,cEncryptedData[,nHashAlgorithm]) ---> return a string␣
↪containing the decryption of cEncryptedData
        nHashAlgorithm indicates the hash algorithm to use for OAEP padding. If omited,␣
↪SHA-1 is used by default.
        Possible values for nHashAlgorithm argument are:
         - $OSSL_HASH_MD5 which is equal to 0
         - $OSSL_HASH_SHA1 which is equal to 1
         - $OSSL_HASH_SHA256 which is equal to 2
         - $OSSL_HASH_SHA384 which is equal to 3
         - $OSSL_HASH_SHA512 which is equal to 4
```

Example:

```
/* decrypt a file by first decrypting AES key that was used to encrypt it
 * and then decrypt the whole content using the AES key
 */
try
        /* read Alice private key */
        rsaKeyPEM = Read("alice_private_key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        /* calculate the modulus length */
        rsaKeyParams = rsa_export_params(rsaKey)
        modulusLen = rsaKeyParams[:bits]/ 8

        /* read encrypted file */
        cEncryptedContent = Read ("oaep_encrypted_document.enc")

        /* IV is the first 16 bytes if the file */
        cIV = substr(cEncryptedContent, 1, 16)

        /* encrypted key follows IV and its length is modulusLen */
        cEncryptedKey = substr(cEncryptedContent, 17, modulusLen)

        /* encrypted data follows the key */
        cEncryptedData = substr(cEncryptedContent, 17 + modulusLen)

        /* decrypt the AES-128 key */
        cKey = rsa_decrypt_oaep(rsaKey,cEncryptedKey)

        /* decrypt the data using the AES-128 key */
```

(continues on next page)

(continued from previous page)

```
        cPlainData = Decrypt(cEncryptedData,cKey,cIV,"aes128")

        /* store the decrypted data to a file */
        Write("oaep_decrypted_document.txt", cPlainData)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.23 rsa_encrypt_raw() Function

We can perform raw RSA encryption on data using the function rsa_encrypt_raw() The size of data must be equal to the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of input data that can be encrypted using raw RSA is 256 bytes. Raw RSA encryption needs only the public part of an RSA key, so rsa_encrypt_raw can be used with both RSA private key and RSA public key. Raw RSA should only be used to implement secure cryptographic protocols. Encrypting user data directly with raw RSA is insecure.

Syntax:

```
rsa_encrypt_raw(pRsaKey,cPlainData) ---> return a string containing the raw RSA␣
↪encryption of cPlainData
```

Example:

```
/* encrypt a file using AES key and then encrypt the AES key using an RSA public key␣
↪using PKCS1 padding */
/* we manually add PKCS1 padding and then perform raw RSA encryption */
try
        /* read Alice public key */
        rsaPublicKeyPEM = Read("alice_public_key.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)

        /* encrypt file with random AES-128 key */
        cData = Read ("secret_document.txt")
        cKey = RandBytes(16)
        cIV = RandBytes(16)
        cEncryptedData = Encrypt(cData,cKey,cIV,"aes128")

        /* encrypt the AES-128 key with the RSA public key */

        /* calculate the modulus length */
        rsaKeyParams = rsa_export_params(rsaPublicKey)
        modulusLen = rsaKeyParams[:bits]/ 8

        /* we manually add PKCS1 padding */
        paddingSize = modulusLen - Len(cKey) - 2 - 1
        paddingStr = space (paddingSize)

        /* encryption case. Add random bytes */
        for i=1 to paddingSize
```

(continues on next page)

(continued from previous page)

```
                paddingStr[i] = Char (1 + Random(254))
        next

        paddedData = Char(0) + Char(2) + paddingStr + Char (0) + cKey

        cEncryptedKey = rsa_encrypt_raw(rsaPublicKey,paddedData)

        /* store IV, encrypted AES key and encrypted data in a file to be sent to Alice*/
        Write("raw_encrypted_document.enc", cIV + cEncryptedKey + cEncryptedData)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.24 rsa_decrypt_raw() Function

We can perform raw RSA decryption of data using the rsa_decrypt_pkcs() function. The size of data that can be decrypted by rsa_decrypt_raw must be equal to modulusLen which is the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of encrypted data that can be decrypted must be 256 bytes. For raw RSA decryption, the RSA key must contain the private key part. The size of the result of raw RSA decryption is equal to the length of RSA modulus in bytes.

Syntax:

```
rsa_decrypt_raw(pRsaKey,cEncryptedData) ---> return a string containing the decryption␣
↪of cEncryptedData
```

Example:

```
/* decrypt a file by first decrypting AES key that was used to encrypt it
 * and then decrypt the whole content using the AES key
 /* We decrypt AES using rsa_decrypt_raw and then remove padding manually
 */
try
        /* read Alice private key */
        rsaKeyPEM = Read("alice_private_key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        /* calculate the modulus length */
        rsaKeyParams = rsa_export_params(rsaKey)
        modulusLen = rsaKeyParams[:bits]/ 8

        /* read encrypted file */
        cEncryptedContent = Read ("encrypted_document.enc")

        /* IV is the first 16 bytes if the file */
        cIV = substr(cEncryptedContent, 1, 16)

        /* encrypted key follows IV and its length is modulusLen */
        cEncryptedKey = substr(cEncryptedContent, 17, modulusLen)
```

(continues on next page)

(continued from previous page)

```
        /* encrypted data follows the key */
        cEncryptedData = substr(cEncryptedContent, 17 + modulusLen)

        /* decrypt the AES-128 key */
        cPaddedKey = rsa_decrypt_raw(rsaKey,cEncryptedKey)

        /* remove PKCS1 padding */
        paddedInputLength = len(cPaddedKey)
        cKey = ""
        if paddedInputLength > 11 AND Ascii(cPaddedKey[1]) = 0 AND Ascii(cPaddedKey[2])␣
↪= 2
                zeroFound = false
                for j = 3 to paddedInputLength
                        if Ascii(cPaddedKey[j]) = 0
                                i = j
                                zeroFound = true
                                exit
                        ok
                next

                if zeroFound
                        if i = paddedInputLength
                                /* unpadded data is empty */
                                Raise("Empty data recovered from padding")
                        else
                                cKey = substr(cPaddedKey,i+1)
                        ok
                else
                        Raise ("Invalid data padding")
                ok
        else
                Raise("the decrypted data is invalid")
        ok

        /* decrypt the data using the AES-128 key */

        cPlainData = Decrypt(cEncryptedData,cKey,cIV,"aes128")

        /* store the decrypted data to a file */
        Write("decrypted_document.txt", cPlainData)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.25 rsa_sign_pkcs() Function

We can sign data with RSA PKCS#1 v1.5 padding using the function rsa_sign_pkcs() The maximum size of data that can be signed by rsa_sign_pkcs is (modulusLen - 11), with modulusLen the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the maximum size of data that can be signed is 256 - 11 = 245 bytes. For RSA PKCS signature, the RSA key must contain the private key part. The size of the result of RSA PKCS signature is equal to the length of RSA modulus in bytes.

Syntax:

```
rsa_sign_pkcs(pRsaKey,cData) ---> return a string containing RSA PKCS signature
```

Example:

```
/* sign a document using  RSA-PKCS with SHA256.
 * digest OID added manually
 */
try
        /* read Alice private key */
        rsaKeyPEM = Read("alice_private_key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        /* read file content */
        cFileContent = Read ("document.txt")

        /* hash content */
        digest = SHA256(cFileContent)

        /* digest OID of SHA256 */
        digestOID = hex2str("3031300d0609608648016503040201050004 20")

        /* perform PKCS signing */
        dataToSign = digestOID + digest
        cSignature = rsa_sign_pkcs(rsaKey,dataToSign)

        /* store the signature */
        Write("document.txt.pkcs1.sig", cSignature)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.26 rsa_signhash_pkcs() Function

We can sign a hash value with RSA PKCS#1 v1.5 padding using the function rsa_signhash_pkcs() This function infers the hash algorithm from hash value size and it automatically adds OID of hash algorithm before applying the PKCS#1 v1.5 padding. For RSA PKCS signature, the RSA key must contain the private key part. The size of the result of RSA PKCS signature is equal to the length of RSA modulus in bytes.

Syntax:

```
rsa_signhash_pkcs(pRsaKey,cHashValue) ---> return a string containing RSA PKCS signature
```

Example:

```
/* sign a document using  RSA-PKCS with SHA256.
 */
try
        /* read Alice private key */
        rsaKeyPEM = Read("alice_private_key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        /* read file content */
        cFileContent = Read ("document.txt")

        /* hash content */
        digest = SHA256(cFileContent)

        /* perform PKCS signing */
        cSignature = rsa_signhash_pkcs(rsaKey,digest)

        /* store the signature */
        Write("document.txt.pkcs1.sig", cSignature)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.27 rsa_verify_pkcs() Function

We can verify an RSA-PKCS signature of data using the function rsa_verify_pkcs() The size of signature must be equal to the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of input signature that can be verified using RSA-PKCS is 256 bytes. RSA-PKCS verification needs only the public part of an RSA key, so rsa_verify_pkcs can be used with both RSA private key and RSA public key.

Syntax:

```
rsa_verify_pkcs(pRsaKey,cData,cSignature) ---> returns 1 if signature is valid and 0␣
↪otherwise
```

Example:

```
/* verify a document signature using RSA-PKCS with SHA256
 * digest OID is added manually
 */
try
        /* read Alice public key */
        rsaPublicKeyPEM = Read("alice_public_key.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)

        /* read file content */
        cFileContent = Read ("document.txt")

        /* hash content */
```

(continues on next page)

(continued from previous page)

```
        digest = SHA256(cFileContent)

        /* digest OID of SHA256 */
        digestOID = hex2str("3031300d0609608648016503040201050004 20")

        /* read file signature */
        cSignature = Read ("document.txt.pkcs1.sig")

        /* perform PKCS verification */
        dataToVerify = digestOID + digest
        if rsa_verify_pkcs(rsaPublicKey,dataToVerify,cSignature)
                See "file signature is valid" + nl
        else
                See "file signature is INVALID" + nl
        ok

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.28 rsa_verifyhash_pkcs() Function

We can verify the RSA-PKCS signature of a hash value using the function rsa_verifyhash_pkcs() This function infers the hash algorithm from hash value size and it automatically uses the OID of hash algorithm during verification. The size of signature must be equal to the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of input signature that can be verified using RSA-PKCS is 256 bytes. RSA-PKCS verification needs only the public part of an RSA key, so rsa_verifyhash_pkcs can be used with both RSA private key and RSA public key.

Syntax:

```
rsa_verifyhash_pkcs(pRsaKey,cHashValue,cSignature) ---> returns 1 if signature is valid␣
↪and 0 otherwise
```

Example:

```
/* verify a document signature using RSA-PKCS with SHA256
 */
try
        /* read Alice public key */
        rsaPublicKeyPEM = Read("alice_public_key.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)

        /* read file content */
        cFileContent = Read ("document.txt")

        /* hash content */
        digest = SHA256(cFileContent)

        /* read file signature */
        cSignature = Read ("document.txt.pkcs1.sig")
```

(continues on next page)

(continued from previous page)

```
        /* perform PKCS verification */
        if rsa_verifyhash_pkcs(rsaPublicKey,digest,cSignature)
                See "file signature is valid" + nl
        else
                See "file signature is INVALID" + nl
        ok

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.29 rsa_sign_pss() Function

We can sign data with RSA PSS using the function rsa_sign_pss() The input data will be first hashed using the specified hash algorithm then RSA PSS signing will be applied to the computed hash value. For RSA PSS signature, the RSA key must contain the private key part. The size of the result of RSA PSS signature is equal to the length of RSA modulus in bytes.

Syntax:

```
rsa_sign_pss(pRsaKey,cData,nHashAlgorithm[,nSaltLength]) ---> return a string containing␣
↪RSA PSS signature
        nHashAlgorithm indicates the hash algorithm to use for hashing and PSS padding.
        nSaltLength indicates the length of PSS salt to use. If ommited, then maximum␣
↪salt length is used.
        nSaltLength can have the special values -1 and -2: -1 indicates that salt length␣
↪is equal to hash size
        and -2 indicates that maximum salt length is used.
        Possible values for nHashAlgorithm argument are:
         - $OSSL_HASH_MD5 which is equal to 0
         - $OSSL_HASH_SHA1 which is equal to 1
         - $OSSL_HASH_SHA256 which is equal to 2
         - $OSSL_HASH_SHA384 which is equal to 3
         - $OSSL_HASH_SHA512 which is equal to 4
```

Example:

```
/* sign a document using  RSA-PSS with SHA256 and maximal salt length
 */
try
        /* read Alice private key */
        rsaKeyPEM = Read("alice_private_key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        /* read file content */
        cFileContent = Read ("document.txt")

        /* perform PSS signing */
        cSignature = rsa_sign_pss(rsaKey,cFileContent,$OSSL_HASH_SHA256)
```

(continues on next page)

(continued from previous page)

```
        /* store the signature */
        Write("document.txt.sig", cSignature)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.30 rsa_signhash_pss() Function

We can sign a hash value with RSA PSS using the function rsa_signhash_pss() This function infers the hash algorithm from hash value size. For RSA PSS signature, the RSA key must contain the private key part. The size of the result of RSA PSS signature is equal to the length of RSA modulus in bytes.

Syntax:

```
rsa_signhash_pss(pRsaKey,cHashValue[,nSaltLength]) ---> return a string containing RSA␣
↪PSS signature
        nSaltLength indicates the length of PSS salt to use. If ommited, then maximum␣
↪salt length is used.
        nSaltLength can have the special values -1 and -2: -1 indicates that salt length␣
↪is equal to hash size
        and -2 indicates that maximum salt length is used.
```

Example:

```
/* sign a document using  RSA-PSS with SHA256 and maximal salt length
 */
try
        /* read Alice private key */
        rsaKeyPEM = Read("alice_private_key.pem")
        rsaKey = rsa_import_pem(rsaKeyPEM)

        /* hash file content */
        ctx = SHA256Init()
        cFileContent = Read ("document.txt")
        SHA256Update(ctx, cFileContent)
        digest = SHA256Final(ctx)

        /* perform PSS signing */
        cSignature = rsa_signhash_pss(rsaKey,digest)

        /* store the signature */
        Write("document.txt.sig", cSignature)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.31 rsa_verify_pss() Function

We can verify the RSA-PSS signature of data using the function rsa_verify_pss() The input data will be first hashed using the specified hash algorithm then RSA PSS verification will be applied to the computed hash value and the given signature to check if they match or not. The size of signature must be equal to the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of input signature that can be verified using RSA-PSS is 256 bytes. RSA-PSS verification needs only the public part of an RSA key, so rsa_verify_pss can be used with both RSA private key and RSA public key.

Syntax:

```
rsa_verify_pss(pRsaKey,cData,cSignature,nHashAlgorithm[,nSaltLength]) ---> returns 1 if␣
↪signature is valid and 0 otherwise
        nHashAlgorithm indicates the hash algorithm to use for hashing and PSS padding.
        nSaltLength indicates the length of PSS salt to use. If ommited, then maximum␣
↪salt length is used.
        nSaltLength can have the special values -1 and -2: -1 indicates that salt length␣
↪is equal to hash size
        and -2 indicates that maximum salt length is used.
        Possible values for nHashAlgorithm argument are:
         - $OSSL_HASH_MD5 which is equal to 0
         - $OSSL_HASH_SHA1 which is equal to 1
         - $OSSL_HASH_SHA256 which is equal to 2
         - $OSSL_HASH_SHA384 which is equal to 3
         - $OSSL_HASH_SHA512 which is equal to 4
```

Example:

```
/* verify a document signature using RSA-PSS with SHA256 and maximal salt length
 */
try
        /* read Alice public key */
        rsaPublicKeyPEM = Read("alice_public_key.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)

        /* read file content */
        cFileContent = Read ("document.txt")

        /* read file signature */
        cSignature = Read ("document.txt.sig")

        /* perform PSS verification */
        if rsa_verify_pss(rsaPublicKey,cFileContent,cSignature,$OSSL_HASH_SHA256)
                See "file signature is valid" + nl
        else
                See "file signature is INVALID" + nl
        ok

        /* store the signature */
        Write("document.txt.sig", cSignature)

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.32 rsa_verifyhash_pss() Function

We can verify the RSA-PSS signature of a hash value using the function rsa_verifyhash_pss() This function infers the hash algorithm from hash value size. The size of signature must be equal to the length of the RSA key modulus in bytes. For example, for 2048-bit RSA key, the length of modulus is 2048/8 = 256 bytes and so the size of input signature that can be verified using RSA-PSS is 256 bytes. RSA-PSS verification needs only the public part of an RSA key, so rsa_verifyhash_pss can be used with both RSA private key and RSA public key.

Syntax:

```
rsa_verifyhash_pss(pRsaKey,cHashValue,cSignature[,nSaltLength]) ---> returns 1 if␣
↪signature is valid and 0 otherwise
        nSaltLength indicates the length of PSS salt to use. If ommited, then maximum␣
↪salt length is used.
        nSaltLength can have the special values -1 and -2: -1 indicates that salt length␣
↪is equal to hash size
        and -2 indicates that maximum salt length is used.
```

Example:

```
/* verify a document signature using RSA-PSS with SHA256 and maximal salt length
 */
try
        /* read Alice public key */
        rsaPublicKeyPEM = Read("alice_public_key.pem")
        rsaPublicKey = rsa_import_pem(rsaPublicKeyPEM)

        /* hash file content */
        ctx = SHA256Init()
        cFileContent = Read ("document.txt")
        SHA256Update(ctx, cFileContent)
        digest = SHA256Final(ctx)

        /* read file signature */
        cSignature = Read ("document.txt.sig")

        /* perform PSS verification */
        if rsa_verifyhash_pss(rsaPublicKey,digest,cSignature)
                See "file signature is valid" + nl
        else
                See "file signature is INVALID" + nl
        ok

catch
        See "Unexpected error occured: " + cCatchError + nl
done
```

30.33 openssl_versiontext() Function

We can get the full version text of the OpenSSL library using the function openssl_versiontext(). The returned string is equal to the value returned by the command "openssl.exe version"

Syntax:

```
openssl_versiontext() ---> return a string containing the full version text of OpenSSL␣
↪library
```

Example:

```
/* Display the version of OpenSSL library used by Ring
 */
See "Ring is using " + openssl_versionText() + nl
```

30.34 openssl_version() Function

We can get the version numbers (Major,Minor,Fix) of the OpenSSL library using the function openssl_version(). The returned list contains three items corresponding to the the three part of the version. For example, for OpenSSL 1.0.2, openssl_version() returns the list [1,0,2]

Syntax:

```
openssl_version() ---> return a list containing the version numbers of the OpenSSL␣
↪library
        First list item holds the version major number
        Second list item holds the version minor number
        Third list item holds the version fix number
```

Example:

```
/* Display the version number of OpenSSL library used by Ring
 */
ver = openssl_version()
OpenSSLVersionMajor = ver[1]
OpenSSLVersionMinor = ver[2]
OpenSSLVersionFix = ver[3]
See "Ring is using OpenSSL version " + OpenSSLVersionMajor + "." + OpenSSLVersionMinor +
↪"." + OpenSSLVersionFix + nl
```

30.35 Large Files Hash

These functions compute the hash of large files/data without the need to load all of the content in a single string.

```
md5init() -> MD5_CTX
md5update (MD5_CTX, string) -> 1 for success or 0 for failure
md5final (MD5_CTX) -> string

sha1init() -> SHA_CTX
```

(continues on next page)

(continued from previous page)

```
sha1update (SHA_CTX, string) -> 1 for success or 0 for failure
sha1final (SHA_CTX) -> string

sha224init() -> SHA224_CTX
sha224update (SHA224_CTX, string) -> 1 for success or 0 for failure
sha224final (SHA224_CTX) -> string

sha256init() -> SHA256_CTX
sha256update (SHA256_CTX, string) -> 1 for success or 0 for failure
sha256final (SHA256_CTX) -> string

sha384init() -> SHA384_CTX
sha384update (SHA384_CTX, string) -> 1 for success or 0 for failure
sha384final (SHA384_CTX) -> string

sha512init() -> SHA512_CTX
sha512update (SHA512_CTX, string) -> 1 for success or 0 for failure
sha512final (SHA512_CTX) -> string
```

30.36 Download() Function

Syntax:

```
Download(cURL) ---> String contains the server response
```

Example:

```
cStr= download("http://doublesvsoop.sourceforge.net/")
see cStr
write("download.txt",cStr)
```

30.37 SendEmail() Function

Syntax:

```
SendEmail(cSMTPServer,cEmail,cPassword,cSender,cReceiver,cCC,cTitle,cContent)
```

Example:

```
See "Send email..." + nl
sendemail("smtp://smtp.gmail.com:587",
        "email@gmail.com",
        "password",
        "email@gmail.com",
        "somebody@yahoo.com",
        "somebodyelse@yahoo.com",
        "Sending email from Ring",
        "Hello
```

(continues on next page)

(continued from previous page)

```
        How are you?
        Are you fine?
        Thank you!
        Greetings,
        Mahmoud")
see "Done.." + nl
```

CHAPTER

THIRTYONE

OBJECT ORIENTED PROGRAMMING (OOP)

In this chapter we are going to learn how to use the Object-Oriented programming paradigm inside the Ring programming language.

We will learn about

- Classes and Objects
- Access Objects Using Braces
- Composition
- Setter and Getter
- Private Attributes and Methods
- Operator Overloading
- Inheritance
- Dynamic Attributes
- Packages
- Printing Objects
- Find() and List of Objects
- Sort() and List of Objects
- Using Self.Attribute and Self.Method()
- Using This.Attribute and This.Method()
- Command: New From

31.1 Classes and Objects

We can define new classes using the next syntax

Syntax:

```
Class <Class Name> [From|<|: <Parent Class Name>]
        [Attributes]
        [Methods]
        [Private
          [Attributes]
          [Methods]
        ]
```

And we can create objects using the next syntax

Syntax:

```
New <Object Name> [ (init method parameters) ] |
                  [ { access object data and methods } ]   ---> Object
```

Example:

```
New point { x=10  y=20  z=30  print() }
Class Point x y z func print see x + nl + y + nl + z + nl
```

Note: We can use { } to access object data and methods.

Tip: we can declare the class attributes directly after the class name.

Output:

```
10
20
30
```

We can rewrite the same program in another style

```
New point                       # create new object using the point class
{                               # access the new object attributes and methods
        x = 10                  # set the x attribute to 10
        y = 20                  # set the y attribute to 20
        z = 30                  # set the z attribute to 30
        print()                 # call the print method
}                               # end of object access


Class Point                     # define the Point class
        x y z                   # the class contains three attributes x, y & z
        func print              # define the print method
                see x + nl +    # print the x attribute
                    y + nl +    # print the y attribute
                    z + nl      # print the z attribute
```

Also we can write the same program in another way

```
P1 = New Point
P1.x = 10
P1.y = 20
P1.z = 30
P1.Print()
Class Point x y z func print see x + nl + y + nl + z + nl
```

Note: we can use the dot operator after the object name to access object members.

Also we can write the same program in another way

```
new point { print() }
Class Point
        x = 10  y = 20  z = 30
        func print see x + nl + y + nl + z + nl
```

Note: we can set the default values for the class attributes when we declare them.

Also we can write the same program in another way

```
new point(10,20,30)
Class Point
        x y z
        func init p1,p2,p3 x=p1 y=p2 z=p3 print()
        func print see x + nl + y + nl + z + nl
```

Note: we can call the init method directly using () when we create new objects

Also we can write the same program in another way

```
new point( [ :x = 10 , :y = 20 , :z = 30 ] )
Class Point x y z
      func init aPara x = aPara[:x] y = aPara[:y] z = aPara[:z] print()
      func print see x + nl + y + nl + z + nl
```

Tip: using Hash for passing method parameters enable us to create optional parameters and change the order of parameters when adding them to the Hash.

31.2 Access Objects Using Braces

We can access the object at any time using braces { }

Inside the braces we can use the object attributes and methods directly

This can be done when we create the object using the New keyword or at any time using the next syntax

```
ObjectName { access object data and methods }
```

Example:

```
See "Creating the Object" + nl
o1 = new Point
See "Using the Object" + nl
o1 {
        x=5
        y=15
        z=25
```

(continues on next page)

(continued from previous page)

```
        print()
}
Class Point x y z func print see x + nl + y + nl + z
```

We can use braces to access objects when we call functions or methods

Example:

```
o1 = new Point

print( o1 { x=10 y=20 z=30 } )

func print object
        see object.x + nl +
            object.y + nl +
            object.z

Class Point x y z
```

We can mix between using braces and the dot operator to access the object in the same expression.

Example:

```
o1 = new Point

O1 { x=10 y=20 z=30 }.print()

Class Point x y z
        func print see x + nl + y + nl + z
```

31.3 Composition

The object may contains other objects as attributes.

Using braces to access objects can be nested.

Example:

```
R1 = New Rectangle
{

        Name = "Rectangle 1"

        P1
        {
                X = 10
                Y = 20
        }

        P2
        {
                X = 200
```

(continues on next page)

(continued from previous page)

```
                Y = 300
        }

        Color = "Blue"

}

see "Name : " + R1.Name + nl +
    "Color: " + R1.Color + nl +
    "P1   : (" + R1.P1.X + "," + R1.P1.Y + ")" + nl +
    "P2   : (" + R1.P2.X + "," + R1.P2.Y + ")"

Class Rectangle
        name  color
        p1 = new Point
        p2 = new Point

Class Point x y
```

Output:

```
Name : Rectangle 1
Color: Blue
P1   : (10,20)
P2   : (200,300)
```

31.4 Setter and Getter

We can define methods to be used when we set and get object attributes.

Syntax:

```
Class ClassName

        AttributeName
        ...

        Func SetAttributeName
                ...

        Func GetAttributeName
                ...
```

Example:

```
o1 = new person

o1.name = "Mahmoud"  see o1.name + nl

o1 { name = "Ahmed"  see name }
```

(continues on next page)

(continued from previous page)

```
Class Person

        name family = "Fayed"

        func setname value
                see "Message from SetName() Function!" + nl
                name = value + " " + family

        func getname
                see "Message from GetName() Function!" + nl
                return "Mr. " + name
```

Output:

```
Message from SetName() Function!
Message from GetName() Function!
Mr. Mahmoud Fayed
Message from SetName() Function!
Message from GetName() Function!
Mr. Ahmed Fayed
```

31.5 Private Attributes and Methods

We can define private attributes and methods after the keyword private inside the class body

Example:

```
o1 = new person {
        name = "Test"
        age = 20
        print()
        o1.printsalary()
}

try
        see o1.salary
catch
        see cCatchError + nl
done

try
        o1.increasesalary(1000)
catch
        see cCatchError + nl
done

Class Person

        name age

        func print
```

(continues on next page)

(continued from previous page)

```
                see "Name   : " + name + nl +
                    "Age    : " + age + nl

        func printsalary
                see "Salary : " + salary + nl

        private

        salary = 15000

        func increasesalary x
                salary += x
```

Output:

```
Name   : Test
Age    : 20
Salary : 15000
Error (R27) : Using private attribute from outside the class : salary
Error (R26) : Calling private method from outside the class : increasesalary
```

31.6 Operator Overloading

We can add the operator method to our class to enable using operators with the class objects.

Syntax:

```
Class ClassName

        ...

        Func operator cOperator,Para

                ...
```

The function operator takes two parameters, the first represent the operator and the second represent the second parameter after the operator.

Example:

```
o1 = new point { x = 10 y = 10 print("P1    : ") }
o2 = new point { x = 20 y = 40 print("P2    : ") }

o3 = o1 + o2
o3.print("P1+P2 : ")

class point x y

        func operator cOperator,Para
                result = new point
                switch cOperator
```

(continues on next page)

(continued from previous page)

```
                on "+"
                        result.x = x + Para.x
                        result.y = y + Para.y
                on "-"
                        result.x = x - Para.x
                        result.y = y - Para.y
                off
                return result

        func print cPoint
                see cPoint + "X : " + x + " Y : " + y + nl
```

Output:

```
P1    : X : 10 Y : 10
P2    : X : 20 Y : 40
P1+P2 : X : 30 Y : 50
```

The next example from the List class in the stdlib.ring

```
Func operator cOperator,Para
        result = new list
        switch cOperator
                on "+"
                        if isobject(para)
                                for t in Para.vValue
                                        vValue + t
                                next
                        but islist(para)
                                for t in Para
                                        vValue + t
                                next
                        ok
                on "len"
                        return len( vValue )
                on "[]"
                        return &vValue[para]
        off
        return result
```

The "len" operator is used with (for in) control structure.

The "[]" operator is used when we try to access the list items, In this case we use the & operator to return the item values like strings an numbers by reference, so we can update it when we access the items.

Another Example

```
func main

See "----1"+nl
    a1 = new BigNumber( "123" )
    a2 = new BigNumber( "456" )
    a3 = new BigNumber( "789" )
See nl+"----2"+nl
```

(continues on next page)

(continued from previous page)

```
        a1.print()
        a2.print()
        a3.print()
See nl+"----3"+nl
    a2 = a1 + "45"
See nl+"----4"+nl
    a2.print()
See nl+"----5"+nl
    a3 = a1 + a2
See nl+"----6"+nl
    a3.print()
See nl+"----7"+nl

###==================================
Func FuncAdd( num1, num2)
   Sum = 0 + num1 + num2     ### Para.aData isNumber
   Sum = "" +Sum             ### Para.adata isString
return Sum                   ### return to Class
###===================================

class BigNumber

    ### Variables
    aData = "468"

    ### Functions INIT default values
    func init aPara
        ? "INIT aPara: " ? aPara
        if isString(aPara)
            aData = aPara
        else
            aData = "" + aPara
        ok

    ### Other Functions
    func operator cOperator, Para
            whatType = Type(Para)
            ? nl+"WhatType-PARA: "+ whatType ? Para
            ? nl+"Operator: " ? cOperator  ? nl+"PARA: " ? Para ? "    ______" ? nl
                            if whatType = "STRING"
                               dataInfo = Para
                               ? "dataInfo String: " ? dataInfo
                            but whatType = "NUMBER"
                               datinfo  = "" + Para
                               ? "dataInfo Number: " ? dataInfo
                            else whatType = "OBJECT"
                               dataInfo = "" + para.aData
                               ? "dataInfo OBJECT: " ? dataInfo
                            ok
                         ? "dataInfo USING: " ? dataInfo
            ### Para.aData does NOT exist on first pass ( Object with member)
            ### Result isObject when assigned "self"
```

(continues on next page)

(continued from previous page)

```
            result = self
            switch cOperator
                on "+"
                     answer = FuncAdd( aData, dataInfo )
                     ? nl+"AnswerString - FunAdd aData, dataInfo: " ? answer
                     ### result = self, is Object, populate Object with aData member
                     result.aData = answer
                off
            ### Result = Self is Object
            return result

    func print
        ? nl+"ClassPrint aData: " ? aData
```

31.7 Inheritance

We can create class from another class in the class definition using the keyword from.

Syntax:

```
Class <Class Name> [From <Parent Class Name>]
```

We can call a method in the parent class from the child class using the super object.

Syntax:

```
func methodname
        ...
        super.methodname()
        ...
```

Example:

```
Func main
        e1 = new Employee {
                Name = "test"
                age = 20
                job = "programmer"
                salary = 20000000
                print()
        }


Class Human
        Name Age
        func print
                see "Name : " + name + nl + "Age  : " + age + nl

Class Employee from Human
        Job Salary
        func print
```

(continues on next page)

(continued from previous page)

```
                super.print()
                see "Job  : " + job + nl + "Salary : " + salary + nl
```

Output:

```
Name : test
Age  : 20
Job  : programmer
Salary : 20000000
```

31.8 Dynamic Attributes

We can write instructions after the class name to be executed when we create new objects

Example:

```
o1 = new dynamicClass
see o1.var5 + nl        # output 5

Class DynamicClass
        for x = 1 to 10
                cStr = "var" + x + " = " + x
                eval(cStr)
        next
```

Tip: in the previous example var1, var2, . . . , var10 will be defined as attributes.

Tip: The problem with the previous example is that x and cStr will be defined as attributes too!

Note: we can write class definitions inside a string then using eval() we can execute the string to define the classes

31.9 Packages

We can create a package (a group of classes under a common name) using the next syntax

```
package PackageName
        Class Class1
                ...
        Class Class2
                ...
        Class Class3
                ...
        ...
```

Example

```
o1 = new System.output.console
o1.print("Hello World")

Package System.Output
        Class Console
                Func Print cText
                        see cText + nl
```

Note: we can use the dot operator as part of the package name

Instead of typing the long name PackageName.ClassName we can use the import command

When we import a package, we can use any class inside this package directly.

Example

```
import system.output
o1 = new console {
        print("Hello World")
}
Package System.Output
        Class Console
                Func Print cText
                        see cText + nl
```

31.10 Printing Objects

We can print the object state (attributes and values) using the see command.

Example:

```
see new point { x=10 y=20 z=30 }
class point x y z
```

Output:

```
x: 10.000000
y: 20.000000
z: 30.000000
```

31.11 Find() and List of Objects

We can use the find() function to search inside a list of objects.

Syntax:

```
Find(List,ItemValue,nColumn,cAttribute) ---> Item Index
```

Example:

```
myList1 = [new Company {position=3 name="Mahmoud" symbol="MHD"},
           new Company {position=2 name="Bert" symbol="BRT"},
           new Company {position=1 name="Ring" symbol="RNG"}
          ]

see find(mylist1,"Bert",1,"name") + nl
see find(mylist1,"Ring",1,"name") + nl
see find(mylist1,"Mahmoud",1,"name") + nl
see find(mylist1,"RNG",1,"symbol") + nl
see find(mylist1,"MHD",1,"symbol") + nl
see find(mylist1,"BRT",1,"symbol") + nl
see find(mylist1,3,1,"position") + nl
see find(mylist1,1,1,"position") + nl
see "Other" + nl
see find(mylist1,"test",1,"name") + nl
see find(mylist1,"test",0,"name") + nl
see find(mylist1,"test",5,"name") + nl

class company position name symbol
```

Output:

```
2
3
1
3
1
2
1
3
Other
0
0
0
```

31.12 Sort() and List of Objects

We can sort a list of objects based on an object attribute using the Sort() function.

Syntax:

```
Sort(List,nColumn,cAttribute) ---> Sorted List based on Object Attribute
```

Example:

```
myList1 = [
                 new Company {position=3 name="Mahmoud" symbol="MHD"},
                 new Company {position=2 name="Bert" symbol="BRT"},
                 new Company {position=8 name="Charlie" symbol="CHR"},
                 new Company {position=6 name="Easy" symbol="FEAS"},
                 new Company {position=7 name="Fox" symbol="EFOX"},
                 new Company {position=5 name="Dog" symbol="GDOG"},
```

(continues on next page)

(continued from previous page)

```
                new Company {position=4 name="George" symbol="DGRG"},
                new Company {position=1 name="Ring" symbol="RNG"}
          ]

see sort(mylist1,1,"name")
see copy("*",70) + nl
see sort(mylist1,1,"symbol")
see copy("*",70) + nl
see sort(mylist1,1,"position")

class company position name symbol
```

Output:

```
position: 2.000000
name: Bert
symbol: BRT
position: 8.000000
name: Charlie
symbol: CHR
position: 5.000000
name: Dog
symbol: GDOG
position: 6.000000
name: Easy
symbol: FEAS
position: 7.000000
name: Fox
symbol: EFOX
position: 4.000000
name: George
symbol: DGRG
position: 3.000000
name: Mahmoud
symbol: MHD
position: 1.000000
name: Ring
symbol: RNG
**********************************************************************
position: 2.000000
name: Bert
symbol: BRT
position: 8.000000
name: Charlie
symbol: CHR
position: 4.000000
name: George
symbol: DGRG
position: 7.000000
name: Fox
symbol: EFOX
position: 6.000000
```

(continues on next page)

(continued from previous page)

```
name: Easy
symbol: FEAS
position: 5.000000
name: Dog
symbol: GDOG
position: 3.000000
name: Mahmoud
symbol: MHD
position: 1.000000
name: Ring
symbol: RNG
***************************************************************************
position: 1.000000
name: Ring
symbol: RNG
position: 2.000000
name: Bert
symbol: BRT
position: 3.000000
name: Mahmoud
symbol: MHD
position: 4.000000
name: George
symbol: DGRG
position: 5.000000
name: Dog
symbol: GDOG
position: 6.000000
name: Easy
symbol: FEAS
position: 7.000000
name: Fox
symbol: EFOX
position: 8.000000
name: Charlie
symbol: CHR
```

31.13 Using Self.Attribute and Self.Method()

Inside the class region (After the class name and before any method) and the class methods we can use self.attribute and self.method()

```
Class Point
        self.x = 10
        self.y = 20
        self.z = 30
        func print
                see self.x + nl + self.y + nl + self.z + nl
```

Note: using self.attribute in the class region to define the class attribute protect the class attributes from conflict with

global variables.

Tip: if you typed the class attributes with self.attribute and there are a global variable with the same name it will be used and the attribute will not be defined.

Check the "Scope Rules" chapter to know about the conflict between the global variable name and the attribute name

What this may happens?

Because

- Because in the class region we can access global variables.
- Before defining any variable, Ring try to find the variable and use it if it's found.

Note: Try to avoid the global variables, use the main function and start their names with $

Tip: In large programs protect your classes and define their members using self.attribute

31.14 Using This.Attribute and This.Method()

Inside class methods we have access to the object scope directly. we don't need to use Self.attribute or Self.method to read/write attribute and call methods.

But we can use braces {} while we are inside methods to access another object, In this case the current object scope will be changed while we are inside the brace.

How we can get access to our class attributes and methods while we are inside braces?

This can be done using This.Attribute and This.Method()

Example:

```
new point

class point
        x=10 y=20 z=30
        print()
        func print
                new UI {
                        display(this.x,this.y,this.z)
                }

Class UI
        func display x,y,z
                see x + nl + y + nl + z + nl
```

31.15 Using This in the class region as Self

The class region is the region that comes after the class name and before any method.

We can use This in the class region as Self.

Example:

```
func main

        o1 = new program {
                test()
        }

        ? o1

class program

        this.name = "My Application"
        this.version = "1.0"
        ? name ? version

        func test
                ? "Name    = " + name
                ? "Version = " + version
```

Output

```
My Application
1.0
Name    = My Application
Version = 1.0
name: My Application
version: 1.0
```

Note: When we use braces to change the current active object, Using This we can still point to the class.

Tip: The difference between This and Self is that Self point to the current active object that we can change using braces.

Remember that in most cases we don't need to use This or Self in the class region

We can write

```
class program name version
```

Or

```
class program name="My Application" version="1.0"
```

Note: We use This or Self in the class region just to avoid conflict with global variables that are defined with the same

name.

31.16 Default value for object attributes

The default value for object attributes is NULL

In Ring, the NULL value is just an empty string or a string that contains "NULL"

We can check for NULL values using the isNULL() function

Example:

```
oProgram = new Program
? oProgram.name
? oProgram.version
? isNULL(oProgram.name)
? isNULL(oProgram.version)
oProgram { name="My Application" version="1.0" }
? isNULL(oProgram.name)
? isNULL(oProgram.version)
? oProgram

class program
        name
        version
```

Output:

```
NULL
NULL
1
1
0
0
name: My Application
version: 1.0
```

31.17 Command: New From

Using (new) we can create a new object from a specific class

Using (new from) we provide a variable which contains the class name

Example:

```
cClassName = "myclass2"
myobj = new from cClassName

cClassName = "myclass"
myobj = new from cClassName
```

(continues on next page)

(continued from previous page)

```
class myclass
        ? :hello

class myclass2
        ? :wow
```

Output:

```
wow
hello
```

CHAPTER

THIRTYTWO

FUNCTIONAL PROGRAMMING

In previous chapters we learned about Functions and Recursion.

In this chapter we are going to learn about more Functional Programming (FP) concepts like

- Pure Functions
- First-class functions
- Higher-order functions
- Anonymous and nested functions.
- Equality of functions

32.1 Pure Functions

We can create pure functions (functions that doesn't change the state) by the help of the assignment operator to copy variables (Lists & Objects) by value to create new variables instead of working on the original data that are passed to the function by reference.

Example:

```
Func Main
        aList = [1,2,3,4,5]
        aList2 = square(aList)
        see "aList" + nl
        see aList
        see "aList2" + nl
        see aList2

Func Square aPara
        a1 = aPara              # copy the list
        for x in a1
                x *= x
        next
        return a1               # return new list
```

Output:

```
aList
1
2
```

(continues on next page)

(continued from previous page)

```
3
4
5
aList2
1
4
9
16
25
```

32.2 First-class Functions

Functions inside the Ring programming language are first-class citizens, you can pass functions as parameters, return them as value or store them in variables.

We can pass/return the function by typing the function name as literal like "FunctionName" or :FunctionName for example.

We can pass/return functions using the variable that contains the function name.

We can call function from variables contains the function name using the Call command

Syntax:

```
Call Variable([Parameters])
```

Example:

```
Func Main
        see "before test2()" + nl
        f = Test2(:Test)
        see "after test2()" + nl
        call f()

Func Test
        see "Message from test!" + nl

Func Test2 f1
        call f1()
        See "Message from test2!" + nl
        return f1
```

Output:

```
before test2()
Message from test!
Message from test2!
after test2()
Message from test!
```

32.3 Higher-order Functions

Higher-order functions are the functions that takes other functions as parameters.

Example:

```
Func Main
        times(5,:test)

Func Test
        see "Message from the test function!" + nl

Func Times nCount,F

        for x = 1 to nCount
                Call F()
        next
```

Output:

```
Message from the test function!
Message from the test function!
Message from the test function!
Message from the test function!
Message from the test function!
```

32.4 Anonymous and Nested Functions

Anonymous Functions are functions without names that can be passed as parameters to other functions or stored in variables.

Syntax:

```
Func [Parameters] { [statements] }
```

Example:

```
test( func x,y {
                        see "hello" + nl
                        see "Sum : " + (x+y) + nl
                } )

new great { f1() }

times(3, func { see "hello world" + nl } )

func test x
        call x(3,3)
        see "wow!" + nl

func times n,x
        for t=1 to n
```

(continues on next page)

(continued from previous page)

```
                call x()
        next

Class great
        func f1
                f2( func { see "Message from f1" + nl } )

        func f2 x
                call x()
```

Output:

```
hello
Sum : 6
wow!
Message from f1
hello world
hello world
hello world
```

Example:

```
Func Main
        aList = [1,2,3,4]
        Map (aList , func x {
                                return x*x
                            } )
        see aList
        aList = [4,9,14,25]
        Map(aList, :myfilter )
        see aList
        aList = [11,12,13,14]
        Map (aList , func x {
                if x%2=0
                        return "even"
                else
                        return "odd"
                ok
        })
        see aList

Func myfilter x
        if x = 9
                return "True"
        else
                return "False"
        ok

Func Map aList,cFunc
        for x in aList
                x = call cFunc(x)
        next
```

Output:

```
1
4
9
16
False
True
False
False
odd
even
odd
even
```

32.5 Equality of functions

We can test if function = function or not using the '=' or '!=' operators

Example:

```
f1 = func { see "hello" + nl }

f2 = func { see "how are you?" + nl }

f3 = f1

call f1()
call f2()
call f3()

see (f1 = f2) + nl
see (f2 = f3) + nl
see (f1 = f3) + nl
```

Output:

```
hello
how are you?
hello
0
0
1
```

CHAPTER

THIRTYTHREE

REFLECTION AND META-PROGRAMMING

Since the Ring programming language is a dynamic language, we can get answers about the program code and we can modify our code during the runtime.

In this chapter we will learn about this and the available functions to use.

- locals()
- globals()
- functions()
- cfunctions()
- islocal()
- isglobal()
- isfunction()
- iscfunction()
- packages()
- ispackage()
- classes()
- isclass()
- packageclasses()
- ispackageclass()
- classname()
- objectid()
- isobject()
- attributes()
- methods()
- isattribute()
- isprivateattribute()
- ismethod()
- isprivatemethod()
- addattribute()
- addmethod()

- getattribute()
- setattribute()
- mergemethods()
- packagename()
- importpackage()
- nothing()
- optionalfunc()

33.1 locals() Function

We can get a list of variables names in the current scope using the locals() function.

Syntax:

```
locals() --> a list contains the variables names in the current scope
```

Example:

```
test("hello")

func test cMsg

        see cMsg + nl

        x = 10
        y = 20
        z = 30

        see locals()
```

Output:

```
hello
cmsg
x
y
z
```

33.2 globals() Function

We can get a list of variables names in the global scope using the globals() function.

Syntax:

```
globals() --> a list contains variables names in the global scope
```

Example:

```
x=10 y=20 z=30
test()

func test
        see "message from test()" + nl +
            "Global Variables:" + nl
        see globals()
```

Output:

```
message from test()
Global Variables:
x
y
z
```

33.3 functions() Function

We can get a list of functions names written in the Ring language using the functions() function.

Syntax:

```
functions() --> a list contains functions names
```

Example:

```
see functions()

func f1
        see "f1" + nl

func f2
        see "f2" + nl

func f3
        see "f3" + nl
```

Output:

```
f1
f2
f3
```

33.4 cfunctions() Function

We can get a list of functions names written in the C language using the cfunctions() function.

Syntax:

```
cfunctions() --> a list contains functions names
```

Example:

```
aList =  cfunctions()
See "Count : " + len(aList) + nl
for x in aList
        see x + "()" + nl
next
```

Output:

```
Count : 251
len()
add()
del()
get()
clock()
...
```

Note: The complete list is removed from the previous output.

33.5 islocal() Function

We can check if a variable is defined in the local scope or not using the islocal() function.

Syntax:

```
islocal(cVariableName) --> returns 1 if the variable is defined in the local scope
                           returns 0 if the variable is not defined in the local scope
```

Example:

```
test()

func test
        x=10 y=20
        see islocal("x") + nl +
            islocal("y") + nl +
            islocal("z") + nl
```

Output:

```
1
1
0
```

33.6 isglobal() Function

We can check if a variable is defined in the global scope or not using the isglobal() function.

Syntax:

```
isglobal(cVariableName) --> returns 1 if the variable is defined in the global scope
                            returns 0 if the variable is not defined in the global scope
```

Example:

```
x=10 y=20

test()

func test
        see isglobal("x") + nl +
            isglobal("y") + nl +
            isglobal("z") + nl
```

Output:

```
1
1
0
```

33.7 isfunction() Function

We can check if a Ring function is defined or not using the isfunction() function.

Syntax:

```
isfunction(cFunctionName) --> returns 1 if the Ring function is defined
                              returns 0 if the Ring function is not defined
```

Example:

```
see isfunction("f1") + nl +
    isfunction("f2") + nl +
    isfunction("f3") + nl

func f1
        see "message from f1()" + nl

func f2
        see "message from f2()" + nl
```

Output:

```
1
1
0
```

33.8 iscfunction() Function

We can check if a C function is defined or not using the iscfunction() function.

Syntax:

```
iscfunction(cFunctionName) --> returns 1 if the C function is defined
                               returns 0 if the C function is not defined
```

Example:

```
see iscfunction("len") + nl +
    iscfunction("add") + nl +
    iscfunction("test") + nl
```

Output:

```
1
1
0
```

33.9 packages() Function

We can get a list of packages names using the packages() function.

Syntax:

```
packages() --> a list contains packages names
```

Example:

```
See packages()

Package Package1
        Class class1
                Func f1

Package Package2
        Class class1
                Func f1

Package Package3
        Class class1
                Func f1

Package Package4
        Class class1
                Func f1
```

Output:

```
package1
package2
package3
package4
```

33.10 ispackage() Function

We can check if a package is defined or not using the ispackage() function.

Syntax:

```
ispackage(cPackageName) --> returns 1 if the Package is defined
                            returns 0 if the Package is not defined
```

Example:

```
See ispackage("package1") + nl +
    ispackage("package4") + nl +
    ispackage("package5") + nl +
    ispackage("package3") + nl

Package Package1
        Class class1
                Func f1

Package Package2
        Class class1
                Func f1

Package Package3
        Class class1
                Func f1

Package Package4
        Class class1
                Func f1
```

Output:

```
1
1
0
1
```

33.11 classes() Function

We can get a list of classes names using the classes() function.

Syntax:

```
classes() --> a list contains classes names
```

Example:

```
See classes()

Class class1
        Func f1

Class class2
        Func f1

Class class3
        Func f1
```

Output:

```
class1
class2
class3
```

33.12 isclass() Function

We can check if a class is defined or not using the isclass() function.

Syntax:

```
isclass(cClassName) -->  returns 1 if the Class is defined
                         returns 0 if the Class is not defined
```

Example:

```
see isclass("class4") + nl +
    isclass("class3") + nl +
    isclass("class2") + nl

Class class1
        func f1

class class2
        func f1

class class3
        func f1
```

Output:

```
0
1
1
```

33.13 packageclasses() Function

We can get a list of classes names inside a package using the packageclasses() function.

Syntax:

```
packageclasses(cPackageName) --> a list contains classes names inside the package
```

Example:

```
see "classes in Package1" + nl
see packageclasses("Package1")
see "classes in Package2" + nl
see packageclasses("Package2")

Package Package1
        Class class1
                Func f1

Package Package2
        Class class1
                Func f1
        Class class2
                Func f1
        Class class3
                func f1
```

Output:

```
classes in Package1
class1
classes in Package2
class1
class2
class3
```

33.14 ispackageclass() Function

We can check if a class is defined inside package or not using the ispackageclass() function.

Syntax:

```
ispackageclass(cPackageName,cClassName) -->  returns 1 if the Class is defined
                                             returns 0 if the Class is not defined
```

Example:

```
see ispackageclass("package1","class1") + nl +
    ispackageclass("package1","class2") + nl +
    ispackageclass("package2","class1") + nl +
    ispackageclass("package2","class2") + nl

Package Package1
        Class class1
                Func f1

Package Package2
        Class class1
                Func f1
        Class class2
                Func f1
        Class class3
                func f1
```

Output:

```
1
0
1
1
```

33.15 classname() Function

We can know the class name of an object using the classname() function

Syntax:

```
classname(object) --> Returns the object class name
```

Example:

```
o1 = new point
o2 = new rect

see classname(o1) + nl          # print point
see classname(o2) + nl          # print rect

class point
class rect
```

33.16 objectid() Function

We can know the object id using the objectid() function

Syntax:

```
objectid(object) --> Returns the object id
```

Example:

```
o1 = new point
see objectid(o1) + nl
test(o1)

func test v
        see objectid(v) + nl

Class point x y z
```

Output:

```
021B5808
021B5808
```

33.17 isobject() Function

We can check the variable to know if it's an object or not using the isobject() function

Syntax:

```
isobject(variable) --> Returns True if it's an object, False if it's not
```

33.18 attributes() Function

We can get the object attributes using the attributes() function

Syntax:

```
attributes(object) --> Returns a list contains the object attributes
```

Example:

```
o1 = new point
aList = attributes(o1)          # we can use see attributes(o1)
for t in aList see t next       # print xyz
Class Point x y z
```

33.19 methods() Function

We can get the object methods using the methods() function

Syntax:

```
methods(object) --> Returns a list contains the object methods
```

Example:

```
o1 = new test
aList = methods(o1)

for x in aList
        cCode = "o1."+x+"()"
        eval(cCode)
next

Class Test
        func f1
                see "hello from f1" + nl
        func f2
                see "hello from f2" + nl
        func f3
                see "hello from f3" + nl
        func f4
                see "hello from f4" + nl
```

Output:

```
hello from f1
hello from f2
hello from f3
hello from f4
```

33.20 isattribute() Function

We can test if the object contains an attribute or not using the isattribute() function

Syntax:

```
isattribute(object,cAttributeName) --> Returns True if the object contains the attribute
```

Example:

```
o1 = new point

see isattribute(o1,"x") + nl    # print 1
see isattribute(o1,"t") + nl    # print 0
see isattribute(o1,"y") + nl    # print 1
see isattribute(o1,"z") + nl    # print 1

class point x y z
```

33.21 isprivateattribute() Function

We can test if the object contains a private attribute or not using the isprivateattribute() function

Syntax:

```
isprivateattribute(object,cAttributeName) --> Returns True if the object
                                              contains the private attribute
```

Example:

```
o1 = new person

see isprivateattribute(o1,"name") + nl +
    isprivateattribute(o1,"address") + nl +
    isprivateattribute(o1,"phone") + nl +
    isprivateattribute(o1,"job") + nl +
    isprivateattribute(o1,"salary")

Class Person
        name address phone
        private
                job salary
```

Output:

```
0
0
0
1
1
```

33.22 ismethod() Function

We can test if the object class contains a method or not using the ismethod() function

Syntax:

```
ismethod(object,cMethodName) --> Returns True if the object class contains the method
```

Example:

```
o1 = new point

see ismethod(o1,"print") + nl           # print 1

mylist = []
mylist + new point

see ismethod(mylist[1],"print") + nl    # print 1

class point x y z
```

(continues on next page)

(continued from previous page)

```
        func print
                see x + nl + y + nl + z + nl
```

33.23 isprivatemethod() Function

We can test if the object class contains a private method or not using the isprivatemethod() function

Syntax:

```
isprivatemethod(object,cMethodName) --> Returns True if the object class contains
                                        the private method
```

Example:

```
o1 = new Test

see isprivatemethod(o1,"f1") + nl +
    isprivatemethod(o1,"f2")

Class Test
        func  f1
                see "message from f1()" + nl
        private
                func f2
                        see "message from f2()" + nl
```

Output:

```
0
1
```

33.24 addattribute() Function

We can add an attribute (or a group of attributes) to the object state (not the class) using the addattribute() function

Syntax:

```
AddAttribute(object,cAttributeName|aAttributesList)
```

Example(1):

```
see new point {x=10 y=20 z=30}
Class Point
        AddAttribute(self,["x","y","z"])
```

Example(2):

```
o1 = new point
addattribute(o1,"x")
addattribute(o1,"y")
```

(continues on next page)

(continued from previous page)

```
addattribute(o1,"z")
see o1 {x=10 y=20 z=30}
class point
```

Output:

```
x: 10.000000
y: 20.000000
z: 30.000000
```

33.25 addmethod() Function

We can add a method to the object class using the addmethod() function This method can be used with any object from the same class.

Syntax:

```
AddMethod(Object,cNewMethodName,cMethodName|AnonymousFunction)
```

Example:

```
o1 = new point { x=10 y=20 z=30 }

addmethod(o1,"print", func { see x + nl + y + nl + z + nl } )

o1.print()

Class point
        x y z
```

Output:

```
10
20
30
```

Instead of using anonymous function to add new method to the class, we can use the function name

Example:

```
o1 = new point { x=10 y=20 z=30 }

myfunc = func { see x + nl + y + nl + z + nl }

addmethod(o1,"print", myfunc )
addmethod(o1,"display", myfunc )
addmethod(o1,"show", myfunc )

o1.print()
o1.display()
o1.show()
```

(continues on next page)

(continued from previous page)

```
Class point
        x y z
```

Output:

```
10
20
30
10
20
30
10
20
30
```

Since we add the method to the class, any object from that class can use this method

Example:

```
o1 = new point { x=10 y=20 z=30 }
o2 = new point { x=100 y=200 z=300 }
o3 = new point { x=50 y=150 z=250 }

addmethod(o1,"print", func { see x + nl + y + nl + z + nl } )

o1.print()
o2.print()
o3.print()

Class point
        x y z
```

Output:

```
10
20
30
100
200
300
50
150
250
```

33.26 getattribute() function

We can get the object attribute value using the getattribute() function

Syntax:

```
GetAttribute(oObject,cAttributeName) ---> Attribute Value
```

Example:

```
o1 = new point

see getattribute(o1,"name") + nl +
    getattribute(o1,"x") + nl +
    getattribute(o1,"y") + nl +
    getattribute(o1,"z") + nl

Class Point
        x=10 y=20 z=30
        name = "3D-Point"
```

Output:

```
3D-Point
10
20
30
```

Example:

We can Find a Class List Member using GetAttribute() using a function findclass() The Find uses the member name, rather than the column number

```
myList =
          [new Company {position=3 name="Mahmoud" symbol="MHD"},
           new Company {position=2 name="Bert" symbol="BRT"},
           new Company {position=1 name="Ring" symbol="RNG"}
          ]

see myList
see nl +"=====================" + nl + nl

for i = 1 to len(myList)
     see  "Pos: "+ i +" | "+ myList[i].position +" | "+ myList[i].name +
          " | "+  myList[i].symbol    +" | "+ nl
next


See findclass(myList, "MHD", "symbol") +nl   ### Specify Member class name

###----------------------------------------

func findclass classList, cValue, classMember
```

(continues on next page)

(continued from previous page)

```
        See nl + "FindClass: "  +" "+ cValue + nl + nl

        for i = 1 to len(classList)
             result = getattribute( classList[i], classMember )

            See "Result-Attr: " + i +" "+  result +nl
            if result = cValue
                j = i
             ok
        next
return j

###-------------------------------------

class company position name symbol
```

Output:

```
Pos: 1 | 3 | Mahmoud | MHD |
Pos: 2 | 2 | Bert | BRT |
Pos: 3 | 1 | Ring | RNG |

FindClass:  MHD

Result-Attr: 1 MHD
Result-Attr: 2 BRT
Result-Attr: 3 RNG

1
```

33.27 setattribute() function

We can set the object attribute value using the setattribute() function

Syntax:

```
SetAttribute(oObject,cAttributeName,Value)
```

Example:

```
o1 = new person
setattribute(o1,"cName","Mahmoud")
setattribute(o1,"nSalary",1000000)
setattribute(o1,"aColors",["white","blue","yellow"])

see o1
see o1.aColors

Class Person
        cName
```

(continues on next page)

(continued from previous page)

```
        nSalary
        aColors
```

Output:

```
cname: Mahmoud
nsalary: 1000000.000000
acolors: List...
white
blue
yellow
```

33.28 mergemethods() Function

We can share methods between classes without inheritance using the MergeMethods() function

This function merge class methods to another class.

Syntax:

```
MergeMethods(cClassNameDestination,cClassNameSource)
```

Example:

```
mergemethods("count","share")
mergemethods("count2","share")

o1 = new count  { test() }
o1 = new count2 { test() }

Class Share
        func one
                see "one" + nl
        func two
                see "two" + nl
        func three
                see "three" + nl

Class Display
        Func printline
                see copy("*",20) + nl

Class Count from Display
        Func test
                printline()
                one()
                two()
                three()
                printline()

Class Count2 from Display
```

(continues on next page)

(continued from previous page)

```
        Func test
                three()
                two()
                one()
                printline()
```

Output:

```
********************
one
two
three
********************
three
two
one
********************
```

33.29 packagename() Function

We can know the package name of the latest successful import command using the packagename() function

Syntax:

```
packagename() --> Returns the package name of the latest successful import
```

Example:

```
load "weblib.ring"
import System.web
see packagename()       # system.web
```

33.30 importpackage() Function

Instead of using the import command we can use the importpackage() function

This function get the package name through a string or variable

This is useful if the package name will be known only during the runtime

Syntax:

```
importpackage(cPackageName)
```

Example:

```
importpackage(:mypackage)
new myclass { myfunction() }

package mypackage
        class myclass
```

(continues on next page)

(continued from previous page)

```
                function myfunction
                        ? "Hello, World!"
```

33.31 Nothing() function

This function does nothing and can accept any number/type of parameters. The output will be Zero.

Some of the Use Cases

(1) Performance measurements, where we can test the performance of calling functions written in C and we can change the number of parameters during tests.

(2) In places of code that you want to write a function name, and it's not defined yet.

(3) To disable some feature/code by just changing the function name to nothing without changing the parameters or commenting the code.

(4) In small programs, where you want to write a function that you can override from a Test program.

33.32 OptionalFunc() function

Using this function we can define functions similar to Nothing() but with a different name.

Syntax:

```
OptionalFunc(cFunctionName)
```

Example:

File: Question.ring

```
optionalFunc(:reply)

? "I love Programming, What about you?"
reply()
? "Ok, Thanks!"
```

Output:

```
I love Programming, What about you?
Ok, Thanks!
```

File: Answer.ring

```
load "Question.ring"

func reply
        ? "Me too!"
```

Output:

```
I love Programming, What about you?
Me too!
Ok, Thanks!
```

CHAPTER

THIRTYFOUR

DECLARATIVE PROGRAMMING USING NESTED STRUCTURES

In this chapter we are going to learn how to build declarative programming world using nested structures on the top of object oriented.

We will learn about

- Creating Objects inside Lists
- Composition and Returning Objects and Lists by Reference
- Executing code after the end of object access
- Declarative Programming on the top of Object-Oriented

34.1 Creating Objects inside Lists

We can create objects inside lists during list definition. Also we can add objects to the list at any time using the Add() function or the + operator.

Example:

```
alist = [new point, new point, new point]       # create list contains three objects

alist + [1,2,3]                                 # add another item to the list

see "Item 4 is a list contains 3 items" + nl
see alist[4]

add(alist , new point)
alist + new point

alist[5] { x = 100 y = 200 z = 300 }
alist[6] { x = 50 y = 150 z = 250 }

see "Object inside item 5" + nl
see alist[5]
see "Object inside item 6" + nl
see alist[6]

class point x y z
```

Output:

```
Item 4 is a list contains 3 items
1
2
3
Object inside item 5
x: 100.000000
y: 200.000000
z: 300.000000
Object inside item 6
x: 50.000000
y: 150.000000
z: 250.000000
```

34.2 Composition and Returning Objects and Lists by Reference

When we use composition and have object as one of the class attributes, when we return that object it will be returned by reference.

if the caller used the assignment operator, another copy of the object will be created.

The caller can avoid using the assignment operator and use the returned reference directly to access the object.

The same is done also if the attribute is a list (not object).

Note: Objects and Lists are treated using the same rules. When you pass them to function they are passed by reference, when you return them from functions they are returned by value except if it's an object attribute where a return by reference will be done.

Example:

```
o1 = new Container
myobj = o1.addobj()     # the assignment will create another copy
myobj.x = 100
myobj.y = 200
myobj.z = 300
see o1.aobjs[1]         # print the object inside the container
see myobj               # print the copy

Class Container
        aObjs = []
        func addobj
                aobjs + new point
                return aobjs[len(aobjs)]        # return object by reference

Class point
        x  = 10
        y  = 20
        z  = 30
```

Output:

```
x: 10.000000
y: 20.000000
z: 30.000000
x: 100.000000
y: 200.000000
z: 300.000000
```

Example(2):

```
func main
        o1 = new screen  {
                content[point()] {
                        x = 100
                        y = 200
                        z = 300
                }
                content[point()] {
                        x = 50
                        y = 150
                        z = 250
                }
        }
        see o1.content[1]
        see o1.content[2]

Class Screen
        content = []
        func point
                content + new point
                return len(content)

Class point
        x  = 10
        y  = 20
        z  = 30
```

Output:

```
x: 100.000000
y: 200.000000
z: 300.000000
x: 50.000000
y: 150.000000
z: 250.000000
```

Example(3):

```
func main
        o1 = New Screen  {
                point() {               # access the object using reference
                        x = 100
                        y = 200
                        z = 300
```

(continues on next page)

(continued from previous page)

```
                }
                point() {               # access the object using reference
                        x = 50
                        y = 150
                        z = 250
                }
        }
        see o1.content[1]
        see o1.content[2]

Class Screen
        content = []
        func point
                content + new point
                return content[len(content)]    # return the object by reference

Class point x=10 y=20 z=30
```

Output:

```
x: 100.000000
y: 200.000000
z: 300.000000
x: 50.000000
y: 150.000000
z: 250.000000
```

34.3 Executing code after the end of object access

We can access an object using { } to use object attributes and methods.

if the object contains a method called BraceEnd(), it will be executed before the end of the object access.

Example:

```
New Point { See "How are you?" + nl }

Class Point x y z
        func braceend
                see "I'm fine, Thank you!" + nl
```

Output:

```
How are you?
I'm fine, Thank you!
```

34.4 Declarative Programming on the top of Object-Oriented

The next features enable us to build and use declarative programming environment using nested structures on the top of object oriented

- using { } to access the object attributes and methods
- BraceEnd() Method
- returning objects by reference
- Setter/Getter Methods (optional)

Example:

```
# Declarative Programming (Nested Structures)

Screen()
{

        point()
        {
                x = 100
                y = 200
                z = 300
        }

        point()
        {
                x = 50
                y = 150
                z = 250
        }
}

# Functions and Classes

Func screen return new screen

Class Screen

        content = []

        func point
                content + new point
                return content[len(content)]

        func braceend
                see "I have " + len(content) + " points!"

Class point

        x=10 y=20 z=30

        func braceend
```

(continues on next page)

(continued from previous page)

```
                see self
```

Output:

```
x: 100.000000
y: 200.000000
z: 300.000000
x: 50.000000
y: 150.000000
z: 250.000000
I have 2 points!
```

34.5 More Beautiful Code

We can get better results and a more beautiful code when we can avoid writing () after the method name when the methods doesn't take parameters. This feature is not provided directly by the Ring language because there is a difference between object methods and object attributes. We can get a similar effect on the syntax of the code when we define a getter method for the object attribute. For example instead of defining the point() method. we will define the point attribute then the getpoint() method that will be executed once you try to get the value of the point attribute. since we write the variable name directly without () we can write point instead of point() and the method getpoint() will create the object and return the object reference for us.

Example:

```
new Container
{
        Point
        {
                x=10
                y=20
                z=30
        }
}

Class Container
        aObjs = []
        point
        func getpoint
                aObjs + new Point
                return aObjs[len(aObjs)]

Class Point x y z
        func braceend
                see "3D Point" + nl + x + nl + y + nl + z + nl
```

Output

```
3D Point
10
20
30
```

CHAPTER

THIRTYFIVE

NATURAL LANGUAGE PROGRAMMING

Using the Ring programming language, we can create Natural programming languages based on classes and objects.

35.1 History

In 2010, I developed a new programming language called Supernova (developed using PWCT). This language uses a code that looks similar to Natural Language statements to create simple GUI applications. Now after five years, In the Ring programming language, we can get similar results, but now we have the ability to create/use code similar to Natural language statements in any domain that we like or need.

The Ring programming language comes with the Supernova spirit, but with more generalization and with mix of other languages spirits.

35.2 Example

The next example presents how to create a class that define two instructions

The first instruction is : I want window

The second instruction is : Window title = <expr>

Also keywords that can be ignored like the 'the' keyword

```
New App
{
        I want window
        The window title = "hello world"
}

Class App

        # Attributes for the instruction I want window
                i want window
                nIwantwindow = 0
        # Attributes for the instruction Window title
        # Here we don't define the window attribute again
                title
                nWindowTitle = 0
        # Keywords to ignore, just give them any value
                the=0
```

(continues on next page)

(continued from previous page)

```
        func geti
                if nIwantwindow = 0
                        nIwantwindow++
                ok

        func getwant
                if nIwantwindow = 1
                        nIwantwindow++
                ok

        func getwindow
                if nIwantwindow = 2
                        nIwantwindow= 0
                        see "Instruction : I want window" + nl
                ok
                if nWindowTitle = 0
                        nWindowTitle++
                ok

        func settitle cValue
                if nWindowTitle = 1
                        nWindowTitle=0
                        see "Instruction : Window Title = " + cValue + nl
                ok
```

Output:

```
Instruction : I want window
Instruction : Window Title = hello world
```

35.3 Change the Ring Keyword 'And'

What if we want to connect between the two instructions using 'and'

We have a problem because in Ring 'and' is a keyword

We can change that using the ChangeRingKeyword command.

Syntax:

```
ChangeRingKeyword  <oldkeyword>  <newkeyword>
```

Note: remember to restore the keyword again

Tip: The ChangeRingKeyword command is executed in the scanner stage by the compiler (before parsing).

Example:

```
ChangeRingKeyword       and  _and

New App
{
                I want window and the window title = "hello world"
}

Class App

                # Attributes for the instruction I want window
                                i want window
                                nIwantwindow = 0
                # Attributes for the instruction Window title
                # Here we don't define the window attribute again
                                title
                                nWindowTitle = 0
                # Keywords to ignore, just give them any value
                                the=0  and=0

ChangeRingKeyword       _and  and

                func geti
                        if nIwantwindow = 0
                                nIwantwindow++
                        ok

                func getwant
                        if nIwantwindow = 1
                                nIwantwindow++
                        ok

                func getwindow
                        if nIwantwindow = 2
                                nIwantwindow= 0
                                see "Instruction : I want window" + nl
                        ok
                        if nWindowTitle = 0
                                nWindowTitle++
                        ok

                func settitle cValue
                        if nWindowTitle = 1
                                nWindowTitle=0
                                see "Instruction : Window Title = " + cValue + nl
                        ok

                func getand
                        see "Using : and" + nl
```

Output:

```
Instruction : I want window
Using : and
```

(continues on next page)

(continued from previous page)

```
Instruction : Window Title = hello world
```

35.4 Change the Ring Operator '+'

What if we want to define a new behavior for any operator like the "+" operator.

We can do this change using the ChangeRingOperator command to hide operator (change it's name)

Then we can use the operator as identifier that we can handle it's behaviour

Syntax:

```
ChangeRingOperator  <oldoperator>  <newoperator>
```

Note: remember to restore the operator again

Tip: The ChangeRingOperator command is executed in the scanner stage by the compiler (before parsing).

Example:

```
ChangeRingOperator + _+

New App {
        +
}

Class App
        +
        func get+
                see "Plus operator"

ChangeRingOperator _+ +
```

Output:

```
Plus operator
```

35.5 Change the '=' operator to 'is'

Example:

```
ChangeRingKeyword       and  _and
ChangeRingOperator      =    is

New App
{
                I want window and the window title is "hello world"
```

(continues on next page)

(continued from previous page)

```
}

ChangeRingOperator      is    =

Class App

                # Attributes for the instruction I want window
                                i want window
                                nIwantwindow = 0
                # Attributes for the instruction Window title
                # Here we don't define the window attribute again
                                title
                                nWindowTitle = 0
                # Keywords to ignore, just give them any value
                                the=0  and=0

ChangeRingKeyword       _and  and

                func geti
                        if nIwantwindow = 0
                                nIwantwindow++
                        ok

                func getwant
                        if nIwantwindow = 1
                                nIwantwindow++
                        ok

                func getwindow
                        if nIwantwindow = 2
                                nIwantwindow= 0
                                see "Instruction : I want window" + nl
                        ok
                        if nWindowTitle = 0
                                nWindowTitle++
                        ok

                func settitle cValue
                        if nWindowTitle = 1
                                nWindowTitle=0
                                see "Instruction : Window Title = " + cValue + nl
                        ok
```

35.6 Using Eval() with our Natural Code

Example:

```
func Main

  cProgram = ' I want window and the window title is "hello world" '

  MyLanguage(cProgram)

Func MyLanguage cCode

  # We add to the code the instructions that change keywords and operators
  # Because Eval() uses a new Compiler Object (the original keywords and operators).

  cCode = '
        ChangeRingKeyword  and  _and
        ChangeRingOperator  =    is
  ' + cCode

  New App
  {
          eval(cCode)
  }


  Class App

          # Attributes for the instruction I want window
                  i want window
                  nIwantwindow = 0
          # Attributes for the instruction Window title
          # Here we don't define the window attribute again
                  title
                  nWindowTitle = 0
          # Keywords to ignore, just give them any value
                  the=0

          ChangeRingKeyword  and  _and
                  and=0
          ChangeRingKeyword  _and  and

          func geti
                if nIwantwindow = 0
                  nIwantwindow++
                ok

          func getwant
                if nIwantwindow = 1
                  nIwantwindow++
                ok

          func getwindow
```

(continues on next page)

(continued from previous page)

```
                if nIwantwindow = 2
                  nIwantwindow= 0
                  see "Instruction : I want window" + nl
                ok
                if nWindowTitle = 0
                  nWindowTitle++
                ok

          func settitle cValue
                if nWindowTitle = 1
                  nWindowTitle=0
                  see "Instruction : Window Title = " + cValue + nl
                ok
```

35.7 BraceStart and BraceEnd Methods

We can write code that will be executed before/after using { }

Example:

```
o1 = new test {
        see "Hello" + nl
}

o1 {}

class test

        func bracestart
                see "start" + nl

        func braceend
                see "end" + nl
```

Output:

```
start
Hello
end
start
end
```

35.8 BraceExprEval Method

The next example demonstrates how to use the "BraceExprEval" method to get expressions in Natural code.

Example:

```
new natural {
        create 5
}

class natural
        create=0
        lkeyword = false
        func braceexpreval r
                if lkeyword lkeyword=false return ok
                see "expr eval" + nl
                see "type: " + type(r) see nl
                see "value : " see r see nl
        func getcreate
                lkeyword = true
                see "create" + nl
```

Output:

```
create
expr eval
type: NUMBER
value : 5
```

35.9 Real Natural Code

The next example is a more advanced example

```
# Natural Code
new program {
        Accept 2 numbers then print the sum
}

# Natural Code Implementation
class program
        # Keywords
                Accept=0 numbers=0 then=0 print=0 the=0 sum=0

        # Execution
        func braceexpreval x
                value = x
        func getnumbers
                for x=1 to value
                        see "Enter Number ("+x+") :" give nNumber
                        aNumbers + nNumber
                next
```

(continues on next page)

(continued from previous page)

```
        func getsum
                nSUm = 0
                for x in aNumbers nSum+= x next
                see "The Sum : " + nSum
        private
                value=0 aNumbers=[]
```

Output:

```
Enter Number (1) :3
Enter Number (2) :4
The Sum : 7
```

35.10 BraceError() Method

The next examples demonstrates how to use the "BraceError" method to handle errors when accessing the object using braces {}.

Example:

```
func main
        o1 = new point {
                x=10 y=20 z=30
                TEST
                SEE test
        }

class point x y z
        func braceerror
                see "Handle Error!" + nl
                SEE "Message :" + cCatchError + nl
                if ( left(cCatchError,11) = "Error (R24)" ) and not isattribute(self,
↪"test")
                        see "add attribute" + nl
                        addattribute(self,"test")
                        test = 10
                ok
                see "done" + nl
                return
```

Output:

```
Handle Error!
Message :Error (R24) : Using uninitialized variable : test
add attribute
done
10
```

Example:

```
new point {
        x=10 y=20 z=30
        test()
        see "mmm..." + NL
}

class point x y z
        func braceerror
                see "Handle Error!" + nl
                see "Message :" + cCatchError + nl
                see self
                see "Done" + NL
```

Output:

```
Handle Error!
Message :Error (R3) : Calling Function without definition !: test
x: 10.000000
y: 20.000000
z: 30.000000
Done
mmm...
```

35.11 Clean Natural Code

Instead of typing the literal as "literal" we can accept the words directly.

Example:

The next example accept hello world instead of "hello world"

But this example uses braceend() to check the end of the instruction

This means that this class process only one natural statement that end with literal.

```
ChangeRingKeyword       and  _and

New App
{
                I want window and the window title is hello world
}

Class App

                # Attributes for the instruction I want window
                                i want window
                                nIwantwindow = 0
                # Attributes for the instruction Window title
                # Here we don't define the window attribute again
                                title is
                                nWindowTitle = 0
                # Keywords to ignore, just give them any value
                                the=0  and=0
```

(continues on next page)

(continued from previous page)

```
                # Data
                                literal = ""

ChangeRingKeyword       _and  and

                func geti
                        if nIwantwindow = 0
                                nIwantwindow++
                        ok

                func getwant
                        if nIwantwindow = 1
                                nIwantwindow++
                        ok

                func getwindow
                        if nIwantwindow = 2
                                nIwantwindow= 0
                                see "Instruction : I want window" + nl
                        ok
                        if nWindowTitle = 0
                                nWindowTitle++
                        ok

                func gettitle
                        if nWindowTitle = 1
                                nWindowTitle=2
                        ok

                func getis
                        if nWindowTitle = 2
                                nWindowTitle=3
                        ok

                func braceend
                        if nWindowTitle = 3
                                see "Instruction : Window Title = " + literal + nl
                                nWindowTitle = 0
                        ok

                func braceerror
                        c= substr(cCatchError,":")
                        while c > 0
                                c= substr(cCatchError,":")
                                cCatchError=substr(cCatchError,c+1)
                        end
                        literal += substr(cCatchError,1)
```

CHAPTER

THIRTYSIX

USING THE NATURAL LIBRARY

In this chapter we will learn how to use the Natural Library to quickly define a language that contains a group of commands.

To start using the library, We need to call naturallib.ring

```
load "naturallib.ring"
```

After loading the library, We can use the NaturalLanguage class that contains the next methods :-

- SetLanguageName(cLanguageName)
- setCommandsPath(cFolder)
- SetPackageName(cPackageName)
- UseCommand(cCommandName)
- SetOperators(cOperators)
- RunFile(cFileName)
- RunString(cString)

36.1 Natural Library - Demo Program

We will write the natural code in a Text file, for example program.txt

File: program.txt

```
Welcome to the Ring programming language!
What you are reading now is not comments, I swear!

After many years of programming I decided to think different about
programming and solve the problems in a better way.

We are writing commands or code and the Ring language is reading
it to understand us! Sure, What you are seeing now is
just ***part of the code - Not the Complete Program***
You have to write little things before and after this
part to be able to run it!

It is the natural part of our code where we can write in English,
Arabic or any Natural Language Then we will tell the computer
```

(continues on next page)

(continued from previous page)

```
through the Ring language what must happens! in a way that we can scale
for large frameworks and programs.

Just imagine what will happens to the world of programming once
we create many powerful frameworks using the Ring language that
uses this way (Natural Programming).

For example When we say Hello to the Machine, It can reply! and when we
say count from 1 to 5 it will understand us, Also if
we said count from 5 to 1 it will
understand us too! You can see the Output window!

This Goal is not new, but the Ring language comes
with an innovative solution to this problem.
```

Output:

```
Hello, Sir!


The Numbers!

1

2

3

4

5

I will count Again!

5

4

3

2

1
```

To execute the natural code, We have start.ring

In start.ring we define the language and the commands.

File: start.ring

```
load "stdlib.ring"
load "naturallib.ring"
```

(continues on next page)

(continued from previous page)

```
New NaturalLanguage {
        SetLanguageName(:MyLanguage)
        SetCommandsPath(CurrentDir()+"/../command")
        SetPackageName("MyLanguage.Natural")
        UseCommand(:Hello)
        UseCommand(:Count)
        RunFile("program.txt")
}
```

We defined a language called MyLanguage, We have folder for the language commands.

Each command will define a class that belong to the MyLanguage.Natural package.

We will define two commands, Hello and Count.

So we must have two files for defining the commands in the CurrentDir()+"/../command" folder

File: hello.ring

```
DefineNaturalCommand.SyntaxIsKeyword([
        :Package = "MyLanguage.Natural",
        :Keyword = :hello,
        :Function = func {
                See  "Hello, Sir!" + nl + nl
        }
])
```

File: count.ring

```
DefineNaturalCommand.SyntaxIsKeywordNumberNumber([
        :Package = "MyLanguage.Natural",
        :Keyword = :count,
        :Function = func {
                if not isattribute(self,:count_times) {
                        AddAttribute(self,:count_times)
                        Count_Times = 0
                }
                if Expr(1) > Expr(2) {
                        nStep = -1
                else
                        nStep = 1
                }
                if Count_Times = 0 {
                        see nl+"The Numbers!" + nl
                        Count_Times++
                else
                        see nl + "I will count Again!" +nl
                }
                for x = Expr(1) to Expr(2) step nStep {
                        see nl+x+nl
                }
                CommandReturn(fabs(Expr(1)-Expr(2))+1)
        }
])
```

36.2 Defining Commands

To define new command we can use the DefineNaturalCommand object

This object provides the next methods :-

- SyntaxIsKeyword(aPara)
- SyntaxIsKeywordNumber(aPara)
- SyntaxIsKeywordNumberNumber(aPara)
- SyntaxIsKeywordNumbers(aPara,nCount)
- SyntaxIsKeywordString(aPara)
- SyntaxIsKeywordStringString(aPara)
- SyntaxIsKeywordStrings(aPara,nCount)
- SyntaxIsKeywordExpression(aPara)
- SyntaxIsKeywordExpressionExpression(aPara)
- SyntaxIsKeywordExpressions(aPara,nCount)
- SyntaxIsCommand(aPara)
- SyntaxIsCommandNumber(aPara)
- SyntaxIsCommandNumberNumber(aPara)
- SyntaxIsCommandNumbers(aPara,nCount)
- SyntaxIsCommandString(aPara)
- SyntaxIsCommandStringString(aPara)
- SyntaxIsCommandStrings(aPara,nCount)
- SyntaxIsCommandExpression(aPara)
- SyntaxIsCommandExpressionExpression(aPara)
- SyntaxIsCommandExpressions(aPara,nCount)

File: mylanguage.ring

```
load "stdlib.ring"
load "naturallib.ring"

MyLanguage = New NaturalLanguage {
        SetLanguageName(:MyLanguage)
        setCommandsPath(CurrentDir()+"/../command")
        SetPackageName("MyLanguage.Natural")
        UseCommand(:Hello)
        UseCommand(:Count)
        UseCommand(:Print)
        UseCommand(:IWantWindow)
        UseCommand(:WindowTitleIs)
        UseCommand(:IWantButton)
}
```

Example (1)

In the next example we will define the Print command.

We will use the SyntaxIsKeywordExpression() Method.

We pass list (as Hash) to the method. We determine the package name, the keyword and the function that will be executed.

Inside this function we uses the Expr(nExprNumber) function to get the expression value that the user will write after the keyword.

File: print.ring

```
DefineNaturalCommand.SyntaxIsKeywordExpression([
        :Package = "MyLanguage.Natural",
        :Keyword = :print,
        :Function = func {
                See  Expr(1)
        }
])
```

Usage:

```
load "mylanguage.ring"

MyLanguage.RunString('
        print "Hello, World!"
')
```

Output:

```
Hello, World!
```

Example (2)

File: iwantwindow.ring

```
DefineNaturalCommand.SyntaxIsCommand([
        :Package = "MyLanguage.Natural",
        :Command = "i want window",
        :Function = func {
                See  "Command: I want window" + nl
        }
])
```

Usage:

```
load "mylanguage.ring"

MyLanguage.RunString('
        i want window
')
```

Output:

```
Command: I want window
```

Example (3)

File: windowtitleis.ring

```
DefineNaturalCommand.SyntaxIsCommandString([
        :Package = "MyLanguage.Natural",
        :Command = "window title is",
        :Function = func {
                See  "Command: Window title is " + Expr(1) + nl
        }
])
```

Usage:

```
load "mylanguage.ring"

MyLanguage.RunString('
        I want window and the window title is "Hello World"
')
```

Output:

```
Command: I want window
Command: Window title is Hello World
```

36.3 Natural Library - Operators

In the next example we uses the Count command without using operators

```
load "mylanguage.ring"

MyLanguage.RunString("
        Hello
        Count 1 5
        Count 5 1
")
```

We can add more description

```
load "mylanguage.ring"

MyLanguage.RunString("
        Hello, Please   Count from 1 to 5 then count from 5 to 1
")
```

Also we can use operators like "(" and ")" around the instruction

```
load "mylanguage.ring"

MyLanguage {
        SetOperators("()")
        RunString("
                Here we will play and will try something
```

(continues on next page)

(continued from previous page)

```
                that looks like Lisp Syntax
                (count  (count 1 5)  (count 20 15))
                Just for fun!
        ")
}
```

36.4 Defining commands using classes

This section is related to the implementation details.

When we define new command, Each command is defined by the Natural Library as a class.

We have the choice to define commands using the simple interface provided by the DefineNaturalCommand object or by defining new class as in the next examples.

If we used DefineNaturalCommand (More Simple), The class will be defined during the runtime.

File: hello.ring

```
Package MyLanguage.Natural

class Hello

        func AddAttributes_Hello
                AddAttribute(self,:hello)

        func GetHello
                See  "Hello, Sir!" + nl + nl
```

File: count.ring

```
Package MyLanguage.Natural

class Count

        func Getcount
                StartCommand()
                CommandData()[:name] = :Count
                CommandData()[:nExpr] = 0
                CommandData()[:aExpr] = []

        func BraceExprEval_Count nValue
                if isCommand() and CommandData()[:name] = :Count {
                        if isNumber(nValue) {
                                CommandData()[:nExpr]++
                                CommandData()[:aExpr] + nValue
                                if CommandData()[:nExpr] = 2 {
                                        Count_Execute()
                                }
                        }
                }
```

(continues on next page)

(continued from previous page)

```
        func AddAttributes_Count
                AddAttribute(self,:count)

        func Count_Execute
                if not isattribute(self,:count_times) {
                        AddAttribute(self,:count_times)
                        Count_Times = 0
                }
                if Expr(1) > Expr(2) {
                        nStep = -1
                else
                        nStep = 1
                }
                if Count_Times = 0 {
                        see nl+"The Numbers!" + nl
                        Count_Times++
                else
                        see nl + "I will count Again!" +nl
                }
                for x = Expr(1) to Expr(2) step nStep {
                        see nl+x+nl
                }
                CommandReturn(fabs(Expr(1)-Expr(2))+1)
```

CHAPTER

THIRTYSEVEN

SCOPE RULES FOR VARIABLES AND ATTRIBUTES

In this chapter we will learn about scope rules and how Ring find variables.

Also we will learn about conflicts and how to solve/avoid them.

The next information are important once you start developing large applications using Ring

These application may uses

- Global variables (Try to avoid them)
- Classes (Object-Oriented)
- braces { } to access objects
- Declarative Programming
- Natural Programming

37.1 Three Scopes

In Ring we have three scopes :-

(1) Public/Global Scope - Each variable you define in the statements part (before functions and classes)

(2) Object Scope - When you are inside an object (Inside class method or using { } to access the object)

(3) Local Scope - Related to functions and methods

37.2 Defining Variables and Variables Access

(1) Ring uses lexical scoping, i.e. the scope of the variable is based on where we defined the variable.

(2) Inside braces { } when you access an object, You will change the current active object scope to this object scope but you still can access the global scope and the local scope.

(3) After the 'Class' keyword and the class name, when you write variable names to be defined as attributes, You still can access the global scope.

In this region (class region - after the class name and before methods) we have

- Global Scope —-> The Global Scope
- Object Scope —-> The Object Scope
- Local Scope —-> The Object Scope

Note: Since the local scope in the class region point also to the object scope in this region, we can use nested braces and still have access to the object scope of the class through the local scope.

Tip: You can create windows and controls as attributes by defining them in this region.

Tip: In the class region if you created objects and used braces { } to access them then using self.attribute inside braces will use the class (not the object that you access) because you have access to the class through the local scope.

(4) Function Parameters are automatically defined in the local scope.

37.3 How Ring find the variable?

1 - Search First in the Local Scope

if not found !

2 - Search in the Object Scope

if not found !

3 - Search in the public scope

if not found —-> Runtime Error

if found —-> Check if we can do optimization to avoid searching next time (Cache / Pointers for performance).

37.4 Using Object.Attribute

When we use object.attribute the search will be in the object attributes only.

I.e. no search will be done in the local scope or in the global scope for the object attribute.

Note: Using self.attribute will search for the first self before searching for attributes.

37.5 The Self Object

The self object is a reference to the current object that we can use from the class methods.

When we are inside class method and use Self we mean the object that will be created from this class.

Inside the class methods if we used Braces { } this will change the current object scope and self will be changed also inside braces to reference the object that we access using Braces.

Inside the Class Region (after the class name and before any method) we have access to the object through the object scope and the local scope also. In this region using Self will always be a reference to the class object. if we used Braces to change the object scope then used Self inside Braces, Also self will be a reference to the class object (not the object that we already access using braces) because in the class region we have :-

- Global Scope —> Global Scope
- Object Scope —> Object Scope
- Local Scope —> Object Scope

And using Braces changes the object scope only (not the local scope) and when Ring search for variables it will search in the Local Scope first so it will find self in the class that we are inside.

37.6 How Ring Define Variables and Attributes

Ring will use the variable name in the Assignment operation

1 - Search using the variable name

2 - If not found —> Avoid the runtime error and define the variable in the current scope

3 - If found —> Use the variable and don't define anything in the current scope

- In the global region (before any function or class) the current scope is the global scope.
- In the class region (after the class name and before any method) the current scope is the object attributes.
- In Functions and methods the current scope is the local scope.

37.7 Conflict between Global Variables and Class Attributes

Look at this example:

```
name = "test"
o1 = new person
see o1

class person
        name
        address
        phone
```

In the previous example we have a global variable called 'name' inside the class person.

when we use the variable 'name', Ring will start the search operation and will try to find it.

if found —> Use it

if not found —> Define new attribute

But the variable name is a global variable, so it will be found and used!

We will not have the attribute name! added to the object.

Solution (1) - Use the Main Function

```
func main
        name = "test"
        o1 = new person
        see o1

class person
```

(continues on next page)

(continued from previous page)

```
        name
        address
        phone
```

Solution (2) - Use special mark for global variable names like $

```
$name = "test"
o1 = new person
see o1

class person
        name
        address
        phone
```

Solution (3) - Use the AddAttribute() Method

```
name = "test"
o1 = new person
see o1

class person
        AddAttribute(self,"name")
        address
        phone
```

Solution (4) - Use self before the attribute name

```
name = "test"
o1 = new person
see o1

class person
        self.name
        address
        phone
```

So what is the best solution to this conflict?

1 - Use the $ Mark for global variables

2 - Optional : Try to avoid global variables and use the Main function

In practice i do both of them.

The other solution

- Use self before the attribute name or use AddAttribute()

Note: Ring have a nice feature through the (load package) command which load a Ring source code file in a new global scope different from the current global scope used by the caller.

37.8 Conflict between Class Attributes and Local Variables

This conflict may happen when we access the object using braces

Example:

```
func main
        name = "nice"
        o1 = new person {name="mahmoud" address="Egypt"  phone = 000 }
        see o1

class person
        name
        address
        phone
```

In the previous example we have the local variable name.

The value of this variable will be set to "mahmoud" instead of the object attribute.

Solution (1) : Just use Self

```
func main
        name = "nice"
        o1 = new person {self.name="mahmoud" address="Egypt"  phone = 000 }
        see o1

class person
        name
        address
        phone
```

Solution (2) : Change the Local variable name

```
func main
        cName = "nice"
        o1 = new person {name="mahmoud" address="Egypt"  phone = 000 }
        see o1

class person
        name
        address
        phone
```

Solution (3) : Change Braces and use the Dot operator

```
func main
        name = "nice"
        o1 = new person
        o1.name ="mahmoud"
        o1.address ="Egypt"
        o1.phone = 000
        see o1

class person
```

(continues on next page)

(continued from previous page)

```
        name
        address
        phone
```

37.9 Using Braces to access objects inside Class Methods

Remember that we have Three scopes (Local Scope, Object Scope and Global Scope) and when we are inside a class method, we expect that we have access to the object attributes and methods and this is true until we use braces to access another object attributes and methods because in this case our object scope will be switched to another object.

```
new point { test() }

class point
        x=10 y=20
        func test
                see x + nl + y + nl # works fine
                myobj = new otherclass {
                        see name + nl
                        see x + nl + y + nl # error !
                }

class otherclass
        name = "test"
```

Output:

```
10
20
test

Line 8 Error (R24) : Using uninitialized variable : x
In method test() in file methodbraceerror.ring
called from line 5  in file methodbraceerror.ring
```

Now what we will do to solve the previous problem?

Solution (1) : Write the code that access the class attributes outside braces.

```
new point { test() }

class point
        x=10 y=20
        func test
                see x + nl + y + nl # works fine
                myobj = new otherclass {
                        see name + nl
                }
                see x + nl + y + nl # Outside braces - works fine
```

(continues on next page)

(continued from previous page)

```
class otherclass
        name = "test"
```

Output:

```
10
20
test
10
20
```

Solution (2) : Don't Use Braces

```
new point { test() }

class point
        x=10 y=20
        func test
                see x + nl + y + nl
                myobj = new otherclass
                see myobj.name
                see x + nl + y + nl

class otherclass
        name = "test"
```

Solution (3) : Copy the self object

We may use this solution if we want to use braces and get access to the class attributes (Just Reading).

```
new point { test() }

class point
        x=10 y=20
        func test
                oSelf = self
                see x + nl + y + nl
                myobj = new otherclass {
                        see name + nl
                        see oself.x + nl + oself.y + nl
                }

class otherclass
        name = "test"
```

Output:

```
10
20
test
10
20
```

Now look at this line

```
oself = self
```

The problem with the previous line is that we will have a new copy from the object Because in Ring the assignment operator copy lists and objects by value (not by reference).

When we access the new object attributes (reading) we don't have problems

But if we modified the object attributes (Then we will modify the copy!).

Note: We can use braces again with the copy

```
new point { test() }

class point
        x=10 y=20
        func test
                oSelf = self
                see x + nl + y + nl
                myobj = new otherclass {
                        see name + nl
                        oSelf {
                                see x + nl + y + nl
                        }
                }

class otherclass
        name = "test"
```

In a GUI application, we may create a class contains the window objects as attributes to be able to access the controls from different methods. Remember the previous information when you try to access objects using braces inside methods because in this case you can't access the object attributes directly and if you copied the self object you will work on a copy and the new controls that you create will be related to the copy and you can't access them.

37.10 Accessing the class attributes from braces inside class methods

We access the class attributes directly from the class methods, also we have the choice to use the Self reference before the attribute/method name. Using Braces {} inside class method change the active object scope and prevent us from getting direct access to the class attributes. Also using Self will not help because the Self reference will be changed to the object that we access using Braces.

In this case if you want to read an attribute you have to copy the Self object before using Braces and if you want to modify an attribute you have to the copy from local variable to the object attribute after using Braces.

This case happens when you want to read/modify attribute instead braces.

```
Class MyApp

        oCon    # Attribute

        # some code here
```

(continues on next page)

(continued from previous page)

```
        Func OpenDatabase
                # some code here
                new QSqlDatabase() {
                        oCon = addDatabase("QSQLITE") {
                                setDatabaseName("weighthistory.db")
                                open()
                        }
                }
                self.oCon = oCon
                # some code here
```

In the previous example we want to create the connection object and save it inside the oCon attribute.

The object is an output from the addDatabase() method that we use after accessing the QSQLDatabase() object.

Inside braces we can't use the Self reference to use the object created from the MyApp class, Because the Self reference here will be to the object that we access using Braces.

We solved the problem in the previous example by creating a local variable called oCon then after Braces we copied that variable to the oCon attribute.

The next code is another solution.

```
Class MyApp

        oCon   # Attribute

        # some code here

        Func OpenDatabase
                # some code here
                oCon = new QSqlDatabase()
                oCon = oCon.addDatabase("QSQLITE") {
                        setDatabaseName("weighthistory.db")
                        Open()
                }
                # some code here
```

The next code is a better solution.

```
Class MyApp

        oCon   # Attribute

        # some code here

        Func OpenDatabase
                # some code here
                new QSqlDatabase() {
                        this.oCon = addDatabase("QSQLITE") {
                                setDatabaseName("weighthistory.db")
                                Open()
                        }
```

(continues on next page)

(continued from previous page)

```
                }
                # some code here
```

Note: We used this.attribute to access the class attribute (oCon) while we are inside Braces.

37.11 Creating a Class for each Window in GUI applications

A good way for creating classes for windows is to define the window directly after the class name

In this area you can use nested braces without problems to define the window and the controls, and they will be attributes that you can access from methods.

Example:

```
Load "guilib.ring"

new qApp
{
        $ObjectName = "oFirstWindow"
        oFirstWindow = new FirstWindow

        $ObjectName = "oSecondWindow"
        oSecondWindow = new SecondWindow

        exec()
}

Class FirstWindow

        win = new qWidget() {
                setgeometry(0,50,300,200)
                setWindowTitle("First Window")
                label1 = new qLabel(win)
                {
                        setgeometry(10,10,300,30)
                        setText("0")
                }
                btn1 = new qPushButton(win)
                {
                        move(100,100)
                        setText("Increment")
                        setClickEvent($ObjectName+".increment()")
                }
                show()
        }

        Func Increment
                label1 {
                        setText( "" + ( 0 + text() + 1 ) )
```

(continues on next page)

(continued from previous page)

```
                }


Class SecondWindow

        win = new qWidget() {
                setgeometry(400,50,300,200)
                setWindowTitle("Second Window")
                label1 = new qLabel(win)
                {
                        setgeometry(10,10,300,30)
                        setText("0")
                }
                btn1 = new qPushButton(win)
                {
                        move(100,100)
                        setText("Decrement")
                        setClickEvent($ObjectName+".decrement()")
                }
                show()
        }

        Func Decrement
                label1 {
                        setText( "" + ( 0 + text() - 1 ) )
                }
```

37.12 Conflict between self inside braces and self in the class region

In the class region (after the class name and before any methods) we define the attributes.

In this region we have access to the global scope and the local scope will point to the object scope.

Three Scopes

- Global Scope —> Global Scope
- Object Scope —> Object Scope
- Local Scope —> Object Scope

Look at this example

```
New Account {
        see aFriends
}

Class Account
        name = "Mahmoud"
        aFriends = []
        aFriends + new Friend {
                name = "Gal"
        }
```

(continues on next page)

(continued from previous page)

```
        aFriends + new Friend {
                name = "Bert"
        }

Class Friend
        name
```

Output:

```
name: NULL
name: NULL
```

The problem in the previous example is that the Class account contains an attribute called "name" and the Friend class contains an attribute called "name" also.

If you tried using self.name inside braces you will get the same result!

```
New Account {
        see aFriends
}

Class Account
        name = "Mahmoud"
        aFriends = []
        aFriends + new Friend {
                self.name = "Gal"
        }
        aFriends + new Friend {
                self.name = "Bert"
        }

Class Friend
        name
```

So why using self.name inside braces doesn't solve this conflict?

Because after the class region we have

- global scope —> global scope
- object scope —> object scope (Account Class)
- local scope —> local scope (Account Class)

When we use braces we change the object scope, so we have

- global scope —> global scope
- object scope —> object scope (Friend Class)
- local scope —> local scope (Account Class)

Ring search in the local scope first, so using self.name will use the Account class.

There are many solution

Solution (1) : Access the object through the list

```
New Account {
        see aFriends
}

Class Account
        name = "Mahmoud"
        aFriends = []
        aFriends + new Friend
        aFriends[len(aFriends)] {
                aFriends[len(aFriends)].name = "Gal"
        }
        aFriends + new Friend
        aFriends[len(aFriends)] {
                aFriends[len(aFriends)].name = "Bert"
        }

Class Friend
        name
```

Solution (2) : Create Method in the friend class to set the name attribute.

```
New Account {
        see aFriends
}

Class Account
        name = "Mahmoud"
        aFriends = []
        aFriends + new Friend {
                setname("Gal")
        }
        aFriends + new Friend {
                setname("Bert")
        }

Class Friend
        name
        func setname cName
                name = cName
```

Solution (3) : Create a method in the account class to set the attribute

```
New Account {
        see aFriends
}

Class Account
        name = "Mahmoud"
        aFriends = []
        friend("Gal")
        friend("Bert")

        func friend cName
```

(continues on next page)

(continued from previous page)

```
                aFriends + new Friend {
                        name = cName
                }

Class Friend
        name
```

Solution (4) : Declarative Programming

```
New Account {
        name = "mahmoud"
        friend {
                name = "Gal"
        }
        friend {
                name = "Bert"
        }
        see aFriends
}

Class Account
        name
        aFriends = []
        friend
        func getfriend
                aFriends + new Friend
                return aFriends[len(aFriends)]

Class Friend
        name
```

Output:

```
name: Gal
name: Bert
```

37.13 Using braces to escape from the current object scope

Since braces change the current object scope to another object. we can use it to do some work without modifying the class attributes and using the same variable names.

```
new point {x=10 y=20 z=30 start() }
class point x y z
        func start
                see self # print the x y z values (10,20,30)
                new Local {
                        x = 100
                        y = 200
                        z = 300
                }
```

(continues on next page)

(continued from previous page)

```
                see self # print the x y z values (10,20,30)
                see x + nl # will print 100
                see y + nl # will print 200
                see z + nl # will print 300
                Self {  # NO Advantage - Search is done in local scope first
                        see x + nl # will print 100
                        see y + nl # will print 200
                        see z + nl # will print 300
                }
                see self.x + nl # will print 10
                see self.y + nl # will print 20
                see self.z + nl # will print 30

class Local
```

Output:

```
x: 10.000000
y: 20.000000
z: 30.000000
x: 10.000000
y: 20.000000
z: 30.000000
100
200
300
100
200
300
10
20
30
```

37.14 The For Loops uses the local scope

Starting from Ring 1.8, when the For Loop defines new identifier (variable) it will define it in the local scope.

Example:

```
x = 10
? x             # Print 10
test1()
? x             # Print 10
test2()
? x             # Print 10

func test1
        for x = 1 to 5
        next
        ? x     # Print 6
```

(continues on next page)

(continued from previous page)

```
func test2
        list = 1:5
        for x in list
        next
        ? x     # Print NULL (The "For In" loop will kill the reference after the loop)
```

Output:

```
10
6
10
NULL
10
```

37.15 Summary of Scope Rules

At first remember that

1 - Each programming language comes with it's scope rules based on the language goals

2 - Programming in the small is different than Programming in the Large

3 - Some programming language are designed for developing small programs while others are designed for large programs

4 - In programming, If we have access to more than one scope - Then problems may come if we don't manage things correctly

5 - It's always more secure to reduce the number of visible scopes

6 - Some programming languages force you to manage the scope in some way, while others not!

In Ring

1 - Special and *very simple* scope rules that are designed for Flexibility first then Security

2 - Ring is designed to support programming in the small and programming in the large.

3 - The language provide the different programming paradigms that you may select from based on the project size. Errors comes only if you selected a bad paradigm for the target project or you are using the paradigm in a way that is not correct or at least not common.

4 - In Ring you have the choice, you can use global variables or avoid them. you can give them a special $ mark or leave them. you can use object-oriented or stay with procedures. you can use the class region (after the class name and before any method) just for attributes or use it for code too.

5 - Just read the next scope rules and think about them then use them in your favorite way.

Scope Rules:

1 - At any place in our program code we have only at maximum Three Scopes (Local Scope, Object Scope and Global Scope).

2 - When Ring find a variable it will search in the local scope first then in the object scope then in the global scope.

3 - At any time inside procedures or methods you can use braces { } to access an object and change the current object scope.

4 - In the class region (After the class name and before any method) this is a special region where both of the object scope and the local scope point to the object scope. I.e. No local variables where each variable you define in this region will become an attribute.

5 - Before defining any variable (in any scope and in the class region too) a search process will be done to use the variable if it's found.

6 - Functions and Methods parameters are defined automatically as local variables to these functions or methods.

7 - Using Object.Attribute will search in the object attributes only.

8 - Using Self.Attribute will lead to a search for Self first then search in Self Attributes.

9 - The Self reference inside class region (after the class name and before any method) always point to the object scope created from the class.

10- The Self reference inside methods will be changed when we uses Braces to be a reference to the object that we access.

11- Writing variable names directly in the class region (after the class name and before any method) means using them or define then (in order).

12- Using self.attribute in the class region reduce search to the object scope (avoid conflict with global scope).

From these rules you can understand all types of conflicts and why you may have them and how to avoid them

Simple advices to avoid any conflict and use the scope rules in a better way

1 - Try to avoid global variables

2 - Use the Main Function - This will help you to avoid global variables

3 - If you are going to use many global variables use the $ mark before the variable name

4 - In the class region if you don't respect the advice number three ($) then use self.attribute when you define your attributes

5 - You can use object.attribute and object.method() instead of object { attribute } and object { method() } if you don't like changing the object scope.

6 - If you will use nested braces in a class - think about using the class region if possible because in this region you will have access to the object that you access using { } + access to the class attributes

7 - If you are inside a class method and used nested braces you will change the object scope with each brace and you will loss the access to the class attributes directly but you have access to the local scope before and after using brace { } , if you will read/modify the class attribute from braces then use This.Attribute because using 'This' means (The object created from this class) while using 'Self' means (The object in the current object scope).

After understanding all of the previous points, You will master this topic.

CHAPTER

THIRTYEIGHT

SCOPE RULES FOR FUNCTIONS AND METHODS

In this chapter we will learn about the scope rules for functions and methods.

You need to know the next information once you started using Ring for large applications.

These applications may contains and use

- Many Packages and Classes written in Ring
- Many Functions written in Ring
- Standard Ring Functions (Written in C language)
- Functions and Classes written in C/C++ languages

38.1 How Ring find the Functions and Methods?

When you call a method or function, Ring will start a search process to find this function

If found –> Call the function and store the function pointer in the cache so Ring can use it again with doing another search.

If not found —> Runtime error message (That you can avoid using Try/Catch)

How the search process is done?

Search for functions/methods follow the next order

1 - Search in methods (if we are inside class method or object using braces {})

2 - Search in functions written by the programmer using Ring Code

3 - Search in functions written in C/C++ like standard Ring functions

This enable us to write clean code inside classes methods and avoid any conflict with functions.

If we want to call a function with the same name as a method in the class we will need a wrapper function or we will access a temp. object using { } then call that function there.

We can replace C/C++ Functions with Ring Functions.

We can replace Ring Functions with Ring Methods.

Note: Using self.method() is not necessary in any use case.

Tip: We can use this.method() to escape from the current active scope that we access using braces { } and call a method in the class that we are inside.

38.2 Example about Sharing Names between Functions and Methods

Look at the next example

```
func main
        o1 = new myclass { test() test2() }
        test2()

func f1
        see "f1 function" + nl

func f2
        see "f2 function" + nl

func f3
        see "f3 function" + nl

func test2
        myline()
        see "test2 function" + nl
        new myclass {
                f1()
                f2()
                f3()
                self.f3()
        }
        myobj = new myclass
        myobj.f3()
        myline()

func myline
        see copy("=",40) + nl

Class myclass

        func test
                myline()
                see "test method" + nl
                f1()
                f2()
                f3()
                myline()

        func f3
                see "f3 method" + nl

        func test2
```

(continues on next page)

(continued from previous page)

```
                myline()
                see "test2 method" + nl
                self {
                        f1()
                        f2()
                        f3()
                }
                myline()
```

Output:

```
========================================
test method
f1 function
f2 function
f3 method
========================================
========================================
test2 method
f1 function
f2 function
f3 method
========================================
========================================
test2 function
f1 function
f2 function
f3 method
f3 method
f3 method
========================================
```

38.3 Calling a function sharing the name with a method in the current class

In the previous example we have a function called f3() and we have a method called f3()

How we can call the f3() function from the test() method ?

Solution (1) : Change the current object scope to another object scope

In this solution we will have an empty class called local that we will use to change the current object scope.

```
func main
        o1 = new myclass { test()}

func f1
        see "f1 function" + nl

func f2
        see "f2 function" + nl
```

(continues on next page)

(continued from previous page)

```
func f3
        see "f3 function" + nl

func myline
        see copy("=",40) + nl

Class myclass

        func test
                myline()
                see "test method" + nl
                f1()
                f2()
                f3()                # call f3() method
                new local { f3() } # call f3() function
                myline()

        func f3
                see "f3 method" + nl

class local
```

Output:

```
========================================
test method
f1 function
f2 function
f3 method
f3 function
========================================
```

CHAPTER

THIRTYNINE

SYNTAX FLEXIBILITY

In this chapter we will learn about some options that are provided automatically by the Ring compiler for syntax flexibility.

39.1 Change Language Keywords

We can change any keyword using the ChangeRingKeyword command.

Note: Remember to restore the keyword again if the team will mix between styles in the same project.

Tip: The ChangeRingKeyword command is executed in the scanner stage by the compiler (before parsing).

Syntax:

```
ChangeRingKeyword  <oldkeyword>  <newkeyword>
```

Example:

```
ChangeRingKeyword see print

print "welcome" + nl

ChangeRingKeyword print see

see "Welcome" + nl
```

Example:

```
ChangeRingKeyword  func function
ChangeRingKeyword  see  print
ChangeRingKeyword  ok   endif
ChangeRingKeyword  next endfor
ChangeRingKeyword  end  endwhile

x = 10
while x > 0
        print "x = " + x + nl
        for t = 1 to 10
```

(continues on next page)

(continued from previous page)

```
                if t = 3
                        print "number three" + nl
                endif
        endfor

x--
endwhile

test()

function test
        print "message from test" + nl

ChangeRingKeyword  function func
ChangeRingKeyword  print see
ChangeRingKeyword  endif ok
ChangeRingKeyword  endfor next
ChangeRingKeyword  endwhile end
```

39.2 Change Language Operators

We can change any operator using the ChangeRingOperator command.

Note: Remember to restore the operator again if the team will mix between styles in the same project.

Tip: The ChangeRingOperartor command is executed in the scanner stage by the compiler (before parsing).

Syntax:

```
ChangeRingOperator  <oldkeyword>  <newkeyword>
```

Example:

The next program hide the + operator by changing it to _+

```
changeringoperator + _+
changeringkeyword SEE PRINT

try
        print 5 + 10
catch
        print nl print "error" print nl
done

changeringoperator _+ +
```

The next program change the + operator to “plus”.

```
changeringoperator + plus
changeringkeyword SEE PRINT

Print 5 plus 5

changeringoperator plus +
changeringkeyword PRINT SEE
```

39.3 Load Syntax Files

You may store a group of ChangeRingKeyword and ChangeRingOperator commands in a file to use later in many source files. You can't use the Load command to call these files because

- ChangeRingKeyword and ChangeRingOperator commands are executed in the scanner phase by the compiler (before parsing).
- The load command is executed in the parsing phase (after the scanner phase).

Solution: Use the LoadSyntax Command which is executed in the scanner phase.

Syntax:

```
LoadSyntax      "syntaxfile.ring"
```

Example:

File : StyleBasicOn.ring

```
ChangeRingKeyword       see     print
ChangeRingKeyword       ok      endif
ChangeRingKeyword       next    endfor
ChangeRingKeyword       end     endwhile
```

File : StyleBasicOff.ring

```
ChangeRingKeyword  print        see
ChangeRingKeyword  endif        ok
ChangeRingKeyword  endfor       next
ChangeRingKeyword  endwhile     end
```

File : UseStyleBasic.ring

```
LoadSyntax "stylebasicon.ring"

x = 10
while x > 0
        print "x = " + x + nl
        for t = 1 to 10
                if t = 3
                        print "number three" + nl
                endif
        endfor
        x--
endwhile
```

(continues on next page)

(continued from previous page)

```
LoadSyntax "stylebasicoff.ring"

see "done" + nl
```

Note: files called by the LoadSyntax command must contains ChangeRingKeyword and ChangeRingOperator commands only.

Tip: files called by the LoadSyntax command doesn't support functions, packages and classes. just imperative commands only.

Note: Using this feature you can create many styles that you can use in the same project and you can support Ring translation to other languages like Arabic, French and so on.

Tip: The effect of LoadSyntax command is related to the current source code file only.

Tip: Using LoadSyntax command is optional, See the (Automatic loading for syntax files) section.

39.4 Using "()" around the function parameters

We can use () around the function parameters (optional).

Example:

```
hello()
sum(3,4)

func hello()
        see "Hello" + nl

func sum(x,y)
        see x+y+nl
```

Output:

```
Hello
7
```

Example:

```
myfunc = func x,y { see x + y + nl }

call myfunc (3,4)
```

(continues on next page)

(continued from previous page)

```
myfunc2 = func (x,y) { see x+y+nl }

call myfunc(3,4)
```

Output:

```
7
7
```

39.5 Using Semi-colon after and between statements

In Ring we can use semi-colon after and between statements (optional).

Example:

```
# Using semi-colon is optional

see "Hello" + nl ; see "How are you?" + nl  ; see "Welcome to Ring" + nl ;
one() ; two() ; three() ;
func one ; see "one" + nl ;
func two ; see "two" + nl ;
func three ; see "three" + nl ;
```

Output:

```
Hello
How are you?
Welcome to Ring
one
two
three
```

39.6 Using $ and @ in the start of the variable name

You can use any unicode character in the variable name also we can use $ and @ in the name.

This feature may help, for example we can start global variables with $ and the object attributes with @.

In other languages like Ruby this is the rule, In the Ring language this is just an option without any force from the Compiler.

example:

```
$global_variable = 5

new test { hello() }

class test
```

(continues on next page)

(continued from previous page)

```
        @instance_variable = 10

        func hello

                local_variable = 15

                see "Global   : " + $global_variable + nl +
                    "Instance : " + @instance_variable + nl +
                    "Local    : " + local_variable + nl
```

Output:

```
Global   : 5
Instance : 10
Local    : 15
```

39.7 Using the 'elseif' keyword as 'but' in if statement

if you don't like the 'but' keyword in if statement Then you can use the 'elseif' keyword.

Example:

```
give x
if x = 1 see "one"
elseif x=2 see "two"
elseif x=3 see "three"
elseif x=4 see "four"
else see "other"
ok
see nl
```

39.8 Using the 'else' keyword as 'other' in switch statement

if you don't like the 'other' keyword in switch statement Then you can use the 'else' keyword.

Also you can replace 'else' with 'other' in if statement.

i.e. 'other' keyword is the same as 'else' keyword.

Example:

```
x = 1
switch x
        on 10
                see "10" + nl
        else
                see "not 10" + nl
end
```

Output:

```
not 10
```

39.9 Using the 'end' keyword in different control structures

We can use the 'end' keyword to close different control structures

- If statement
- For loop
- Switch
- While
- Try-Catch

Example:

```
see "if statement.." + nl
x = 1
if x = 1
        see "one" + nl
elseif x=2
        see "two" + nl
elseif x=3
        see "three" + nl
end
see "for loop.." + nl
for t = 1 to 10
        see t
end
see nl
see "switch..." + nl
x = 1

switch x
        on 1 see "one" + nl
        on 2 see "two" + nl
end

see "try catch..." + nl
try
        x = 1 / 0
catch
        see "catching error" + nl
end
```

Output:

```
if statement..
one
for loop..
12345678910
switch...
```

(continues on next page)

(continued from previous page)

```
one
try catch...
catching error
```

39.10 Using braces to start and end different control structures

We can use braces { } to start and end different control structures

- If statement
- For loop
- Switch
- While
- Try-Catch

Example:

```
see "if statement.." + nl
x = 1
if x = 1 {
        see "one" + nl
elseif x=2
        see "two" + nl
elseif x=3
        see "three" + nl
}
see "for loop.." + nl
for t = 1 to 10 {
        see t
}
see nl
see "switch..." + nl
x = 1

switch x {
        on 1 see "one" + nl
        on 2 see "two" + nl
}

see "try catch..." + nl
try {
        x = 1 / 0
catch
        see "catching error" + nl
}
```

Output:

```
if statement..
one
```

(continues on next page)

(continued from previous page)

```
for loop..
12345678910
switch...
one
try catch...
catching error
```

39.11 Using 'put' and 'get' as 'see' and 'give'

We can replace the 'see' keyword with the 'put' keyword.

Also we can replace the 'give' keyword with the 'get' keyword.

Example:

```
put "Hello World" + nl
put "Enter Your Name ? " Get Name
Put "Hello " + Name
```

39.12 Using 'case' as 'on' in switch statements

We can replace the 'on' keyword with 'case' keyword in the switch statement.

Example (1) :

```
for x=1 to 10
        switch x
        case 1 put "one" + nl
        case 2 put "two" + nl
        case 3 put "three" + nl
        else put "else" + nl
        end
end
```

Example (2) :

```
for x=1 to 10 {
        switch x {
        case 1 put "one" + nl
        case 2 put "two" + nl
        case 3 put "three" + nl
        else put "else" + nl
        }
}
```

39.13 Using 'def' as 'func' in functions/methods definition

We can use the 'def' keyword as the 'func' keyword to define functions and methods.

Example:

```
one() two()

def one put "one" + nl
def two put "two" + nl
```

39.14 Using braces { } in Packages/Classes/Functions

Example:

```
load "stdlib.ring"

import mypackage

new myclass {
        myfunc()
}

package mypackage
{
        class myclass
        {
                func myfunc
                {
                        print("Hello, World!\n")
                }
        }
}
```

39.15 Using 'break'/'continue' keywords

Instead of using Exit/Loop commands we can use Break/Continue

Example:

```
for t=1 to 10 {
        if t=3 {
                continue
        elseif t=5
                break
        }
        ? t
}
```

Output:

```
1
2
4
```

39.16 Using 'end' keyword after Packages/Classes/Functions

Example:

```
import mypackage

new myclass {
        myfunc()
}

package mypackage
        class myclass
                def myfunc
                        put "Hello, World!"
                end
        end
end
```

39.17 Using 'function'/'endfunction' keywords

Example:

```
one() two() three()
function one
        ? :one
endfunction
function two
        ? :two
endfunction
function three
        ? :three
endfunction
```

39.18 Using 'endif'/'endfor'/'endwhile'/'endswitch'/'endtry' keywords

Example:

```
for t=1 to 10
        if t=3
                ? :three
        endif
endfor
```

39.19 Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions

Example:

```
import mypackage

new myclass { myfunc() }

package mypackage
        class myclass
                func myfunc
                        see "welcome"  + nl
                endfunc
        endclass
endpackage
```

39.20 Ignore new lines after keywords

Starting from Ring 1.8 the compiler will ignore new lines after keywords that expect tokens after it

Example:

```
see
"
        Hello, World!
"
test()

func
#======================#
        Test
#======================#

        ?
        "

        Hello from the Test function

        "
```

Output:

```
Hello, World!


Hello from the Test function
```

39.21 Automatic loading for syntax files

Starting from Ring 1.18 we have better syntax flexibility

- Support running source code files with any extension
- Automatic loading for (ringsyntax.ring) file that exist in the current folder

For example in the the next screen shot

We have (ringsyntax.ring) that translate some of the Ring keywords to Arabic language

When we execute the file with Arabic name which means in English (hello.ring)

Ring will automatically execute (ringsyntax.ring) using Load Syntax command

Each Folder in the program could have it's optional (ringsyntax.ring) file

We can mix styles in the same project

For Windows users, To use Arabic source code files with Ring, Set the language settings.

Time & language › Language & region
English (United Kingdom)
language pack, text-to-speech, speech recognition, handwriting, basic typing
Arabic (Saudi Arabia)
language pack, text-to-speech, basic typing
Region
Country or region
Windows and apps might use your country or region to give you local content
Regional format
Related settings
Typing
Spell check, autocorrect, text suggestions
Administrative language settings
Windows Back up
Language and regional format save to account while Language preferences is ticked.
Region
Formats
Administrative
Welcome screen and new user accounts
View and copy your international settings to the welcome screen, system accounts and new user accounts.
Copy settings...
Language for non-Unicode programs
This setting (system locale) controls the language used when displaying text in programs that do not support Unicode.
Current language for non-Unicode programs:
English (United Kingdom)
Change system locale...
OK
Cancel
Apply
Region Settings
Select which language (system locale) to use when displaying text in programs that do not support Unicode. This setting affects all user accounts on the computer.
Current system locale:
English (United Kingdom)
Beta: Use Unicode UTF-8 for worldwide language support
OK
Cancel

CHAPTER

FORTY

INTRODUCTION TO THE TYPE HINTS LIBRARY

In this chapter we will learn about the Type Hints Library

40.1 Why Type Hints?

Using this library we can add the type information to the source code which will be very useful for tools like

- Code Editors
- Static-Analysis

Note: Ring is a dynamic language, No type checking will be done by the compiler.

40.2 Example

The next example will use the Type Hints library

```
load "typehints.ring"

see sum(3,4) + nl ;
see sayHello("Mahmoud");

int func sum(int x,int y) {
        return x+y ;
}

string func sayHello(string name) {
        return "Hello " + name ;
}
```

40.3 User Types

The Type Hints library is very powerful and will support user types (Classes) automatically

Example:

```
load "typehints.ring"

import mypackage

test()  { main([:one,:two,:three]) }

myclass func test() {
        see "Testing User Types!" + nl
        return new myclass
}

package mypackage {
        public class myclass {
                public static void func main(list args) {
                        see "welcome" + nl
                        see args
                }
        }
}
```

40.4 Using Types inside Code

Also you can use the types inside the code (not only the function prototype)

Example:

```
load "typehints.ring"

int     sum = sum(3,4)
string  msg = sayHello("Mahmoud")

see "Sum = " + sum + nl + msg + nl


int func sum(int x,int y) {
        return x+y ;
}

string func sayHello(string name) {
        return "Hello " + name ;
}
```

40.5 Using Override

We can use override or @override

Example:

```
load "typehints.ring"

o = new MyNewLib {
        ? isGreaterThanTwo(10)
        ? isGreaterThanTwo(1)
}

class MyLib {
        boolean func isGreaterThanTwo(int x) {
                if x > 2
                        return true
                else
                        return false
                ok
        }
}

class MyNewLib < MyLib {
        @override
        boolean func isGreaterThanTwo(int x) {
                ? "Using override"
                return x > 2
        }
}
```

Output:

```
Using override
1
Using override
0
```

40.6 Rules

- To use the types in the function prototype you must use '(' and ')' around parameters
- To use the types in the function code, You must set the variable value (Assignment).

The next types are defined by the library

```
# Low Level Types
char
unsigned
signed
int
short
```

(continues on next page)

(continued from previous page)

```
long
float
double
void
byte
boolean


# High Level Types
string
list
number
object

# Other
public
static
abstract
protected
override
@override
```

CHAPTER

FORTYONE

THE TRACE LIBRARY AND THE INTERACTIVE DEBUGGER

In this chapter we will learn about the Trace Library and the Interactive Debugger

41.1 Loading the Trace library

To start using the Trace library, We must load it first!

```
load "tracelib.ring"
```

41.2 Trace All Events

The next example demonstrates the Trace library usage to trace all events.

```
# Trace All Events
trace(:AllEvents)

see "Hello, world!" + nl
see "Welcome" + nl
see "How are you?" +nl

mytest()

new myclass { mymethod() }

func mytest
        see "Message from mytest" + nl

class myclass
        func mymethod
                see "Message from mymethod" + nl
```

41.3 Trace control flow between functions

The next example demonstrates the Trace library usage to trace the control flow between functions.

```
Trace(:Functions)

test1()

func test1
        see :test1 + nl
        test2()

func test2
        see :test2 + nl
        see test3() + nl

func test3
        see :test3 + nl
        return "test 3 output"
```

41.4 Pass Error

The next example demonstrates the Trace library usage to pass an error!

```
Trace(:PassError)

test1()

func test1
        x = 10
        see :test1 + nl
        test2() # Runtime Error!
        see "We can continue!"
```

41.5 Interactive Debugger

The next example demonstrates the Trace library usage to use the Interactive Debugger

```
Trace(:Debugger)

test1()
see "good bye!" + nl

func test1
        x = 10
        see :test1 + nl
        t = 12
        test2() # Runtime Error!
        see "After Error!" +nl
```

(continues on next page)

(continued from previous page)

```
        see "t = " see t see nl
        see "x = " see x see nl
```

41.6 Execute Program Line by Line

The next example demonstrates the Trace library usage to execute the program line by line!

```
Trace(:LineByLine)

test1()

func test1
        x = 10
        see :test1 + nl
        t = 12
        test2()
        see "After Error!" +nl
        see "t = " + t + nl
```

41.7 BreakPoint

The next example demonstrates the Trace library usage to stop at a breakpoint!

```
test1()

func test1
        x = 10
        see :test1 + nl
        t = 12
        BreakPoint()
        see "After breakpoint!" +nl
        see "t = " + t + nl
        see "End of program!" + nl
```

41.8 Disable BreakPoints

The next example demonstrates the Trace library usage and how to disable the Breakpoints!

```
NoBreakPoints()

test1()

func test1
        x = 10
        see :test1 + nl
        t = 12
```

(continues on next page)

(continued from previous page)

```
        BreakPoint()
        see "After breakpoint!" +nl
        see "t = " + t + nl
        see "End of program!" + nl
```

41.9 Using the Interactive Debugger

The next example uses a Breakpoint to open the Interactive Debugger!

```
load "tracelib.ring"

test1()

func test1
        x = 10
        see :test1 + nl
        t = 12
        BreakPoint()
        see "After breakpoint!" +nl
        see "t = " + t + nl
        see "End of program!" + nl
```

Screen Shots:

We have the Interactive Debugger at the Breakpoint!

We can print the variables values

```
load "tracelib.ring"

test1()

func test1
        x = 10
        see :test1 + nl
        t = 12
        BreakPoint()
        see "After breakpoint!" +nl
        see "t = " + t + nl
        see "End of program!" + nl
```

```
C:\WINDOWS\system32\cmd.exe - B:\ring\applications\rnote/batch\run "B:/ring/ringlibs/tracelib/samples/sample6.ring"
============================================================
Command (Exit)        : End Program
Command (Cont)        : Continue Execution
Command (Locals)      : Print local variables names
Command (LocalsData)  : Print local variables data
Command (Globals)     : Print global variables names
Command (CallStack)   : Print call stack
We can execute Ring code
============================================================

code:> localsdata

Variable : x      Type : NUMBER          Value : 10
Variable : t      Type : NUMBER          Value : 12

code:>
```

We can change the variables values then continue execution

```
# BreakPoint

load "tracelib.ring"

test1()

func test1
        x = 10
        see :test1 + nl
        t = 12
        BreakPoint()
        see "After breakpoint!" +nl
        see "t = " + t + nl
        see "End of program!" + nl
```

```
C:\WINDOWS\system32\cmd.exe - B:\ring\applications\rnote/batch\run "B:/ring/ringlibs/tracelib/samples/sample6.ring"

code:> localsdata

Variable : x      Type : NUMBER          Value : 10
Variable : t      Type : NUMBER          Value : 12

code:> x = 100

code:> t = 200

code:> cont
After breakpoint!
t = 200
End of program!
```

We can run the Interactive Debugger in the Output Window

CHAPTER

FORTYTWO

EMBEDDING RING IN RING

In this chapter we will learn about embedding Ring in Ring programs and applications.

42.1 Embedding Ring in Ring without sharing the State

From Ring 1.0 we already have functions for embedding Ring in the C language. Also we can execute Ring code inside Ring programs using the eval() function. In this release we provide functions for embedding Ring in Ring programs without sharing the state.

Advantages:

(1) Quick integration for Ring programs and applications together without conflicts.

(2) Execute and run Ring code in safe environments that we can trace.

Example:

```
pState = ring_state_init()
ring_state_runcode(pState,"See 'Hello, World!'+nl")
ring_state_runcode(pState,"x = 10")

pState2 = ring_state_init()
ring_state_runcode(pState2,"See 'Hello, World!'+nl")
ring_state_runcode(pState2,"x = 20")

ring_state_runcode(pState,"see x +nl")
ring_state_runcode(pState2,"see x +nl")

v1 = ring_state_findvar(pState,"x")
v2 = ring_state_findvar(pState2,"x")

see v1[3] + nl
see V2[3] + nl

ring_state_delete(pState)
ring_state_delete(pState2)
```

Output:

```
Hello, World!
Hello, World!
10
```

(continues on next page)

(continued from previous page)

```
20
10
20
```

42.2 Serial Execution of Programs

We can execute application after another application using ring_state_main()

Example:

```
chdir(exefolder()+"/../applications/formdesigner")
ring_state_main('formdesigner.ring')
chdir(exefolder()+"/../applications/cards")
ring_state_main('cards.ring')
```

42.3 ring_state_setvar()

Using ring_state_setvar() we can set variables value

The value could be (String, Number, List or C Pointer)

We need this function to quickly pass lists and C pointers to the Sub Ring Environment

Syntax:

```
ring_state_setvar(oState,cVariableName,Value)
```

Example:

```
load "guilib.ring"

myapp   = null
win     = null

func main
        myapp = new qApp {
                win = new qWidget() {
                        setWindowTitle("Advanced Example on using ring_state_setvar()")
                        move(100,100)
                        resize(600,400)
                        new qPushButton(win) {
                                setText("Test")
                                setClickEvent("Test()")
                        }
                        # We need this because using load 'guilib.ring' in
                        # the sub environment
                        # Will create timers by Qt and closing the window
                        # will not be enough to close the application
                        oFilter = new qAllEvents(win)
                        oFilter.setCloseEvent("myapp.quit()")
```

(continues on next page)

(continued from previous page)

```
                        win.installeventfilter(oFilter)
                        show()
                }
                exec()
        }

func test
        pState = ring_state_init()
        ring_state_runcode(pstate,"load 'guilib.ring'")
        ring_state_runcode(pState,"x = NULL")
        # Pass String
                ring_state_setvar(pState,"x","hello")
                ring_state_runcode(pState,"? x")
        # Pass Number
                ring_state_setvar(pState,"x",100)
                ring_state_runcode(pState,"? x")
        # Pass List
                ring_state_setvar(pState,"x",["one","two","three"])
                ring_state_runcode(pState,"? x")
        # Pass Object
        # We can't pass the Ring Object (win)
        # Because Objects store pointers to the Class Information
        # And the class is related to the Parent Ring Environment
        # And the sub Ring environment can't access it
        # But we can pass C pointers like win.pObject
                ring_state_setvar(pState,"x",win.pObject)
        # Now we create the object again but using the same C pointer
        # So we have access to the Same window in the parent Ring environment
                ring_state_runcode(pState,"
                        new qWidget {
                                pObject = x
                                setwindowtitle('Message from the Sub Ring Environment')
                        }
                ")
        ring_state_delete(pState)
```

42.4 ring_state_new() and ring_state_mainfile()

Using ring_state_new() and ring_state_mainfile() we can run Ring programs from Ring programs

But unlike ring_state_main(), Here we can control when to delete the Ring state!

This is important when we run GUI programs from GUI programs

Because they will share the GUI Library (RingQt), And In this case the caller will call

qApp.Exec()

So the sub program, will not stop and will return to the Main program

Here deleting the State of the sub programs will lead to a problem when we run the sub program events

So keeping the state is important for sub GUI programs hosted in GUI programs.

Example:

```
load "guilib.ring"

func main
        new qApp {
                win = new qWidget() {
                        setWindowTitle("Test ring_state_mainfile()")
                        resize(400,400) move(100,100)
                        btn = new qPushButton(Win) {
                                settext("test")
                                setclickevent("mytest()")
                        }
                        show()
                }
                exec()
        }

func mytest
        pState = ring_state_new()
        ring_state_mainfile(pState,"runprogram.ring")
        # Here we don't delete the state if we will run GUI application
        # So we can run the GUI application events
                // ring_state_delete(pState)
```

If you will use this feature, remember to update the previous example based on your application needs

So you can call ring_state_delete() at some point to avoid the memory leak!

42.5 Runtime Errors when Embedding Ring in Ring

Starting from Ring 1.8

When embedding Ring in Ring, the error in the hosted environment will not close the host

Example:

```
? "Start the test!"

pState = ring_state_init()

ring_state_runcode(pState," ? 'Let us try having an error' ? x")

ring_state_delete(pState)

? ""
? "End of test!"
```

Output:

```
Start the test!
Let us try having an error

Line 1 Error (R24) : Using uninitialized variable : x
```

(continues on next page)

(continued from previous page)

```
in file Ring_EmbeddedCode
End of test!
```

42.6 ring_state_filetokens() function

Starting from Ring 1.12 we have the ring_state_filetokens() function

Using this function we can get all the tokens in the ring source code file.

```
C_FILENAME      = "test_tokens.ring"
C_WIDTH         = 12

# write the file
        write(C_FILENAME,'
                        see "Hello, World!"
                        ? 3*2+3
                        Name = "Ring"
                        ? Name
')

# Token Type
        C_KEYWORD       = 0
        C_OPERATOR      = 1
        C_LITERAL       = 2
        C_NUMBER        = 3
        C_IDENTIFIER    = 4
        C_ENDLINE       = 5

# Keywords List
aKEYWORDS = ["IF","TO","OR","AND","NOT","FOR","NEW","FUNC",
"FROM","NEXT","LOAD","ELSE","SEE","WHILE","OK","CLASS","RETURN","BUT",
"END","GIVE","BYE","EXIT","TRY","CATCH","DONE","SWITCH","ON","OTHER","OFF",
"IN","LOOP","PACKAGE","IMPORT","PRIVATE","STEP","DO","AGAIN","CALL","ELSEIF",
"PUT","GET","CASE","DEF","ENDFUNC","ENDCLASS","ENDPACKAGE",
"CHANGERINGKEYWORD","CHANGERINGOPERATOR","LOADSYNTAX"]

pState = ring_state_new()
aList = ring_state_filetokens(pState,C_FILENAME)
PrintTokens(aList)
ring_state_delete(pState)

func PrintTokens aList
        for aToken in aList
                switch aToken[1]
                on C_KEYWORD
                        ? Width("Keyword",C_WIDTH) + ": "  + aKeywords[0+aToken[2]]
                on C_OPERATOR
                        ? Width("Operator",C_WIDTH)  + ": " + aToken[2]
                on C_LITERAL
                        ? Width("Literal",C_WIDTH)  + ": " + aToken[2]
                on C_NUMBER
```

(continues on next page)

(continued from previous page)

```
                        ? Width("Number",C_WIDTH)  + ": " + aToken[2]
                on C_IDENTIFIER
                        ? Width("Identifier",C_WIDTH)  + ": " + aToken[2]
                on C_ENDLINE
                        ? "EndLine"
                off
        next

func Width cText,nWidth
        return cText+copy(" ",nWidth-len(cText))
```

Output:

```
EndLine
Keyword     : SEE
Literal     : Hello, World!
EndLine
Operator    : ?
Number      : 3
Operator    : *
Number      : 2
Operator    : +
Number      : 3
EndLine
Identifier  : name
Operator    : =
Literal     : Ring
EndLine
Operator    : ?
Identifier  : name
EndLine
```

CHAPTER

FORTYTHREE

STDLIB FUNCTIONS

In this chapter we are going to learn about functions in the stdlib.ring

Before using the functions in the library, We must load the library first

```
load "stdlib.ring"
```

Instead of using stdlib.ring we can use stdlibcore.ring

Using stdlibcore.ring we can use the StdLib functions (Without Classes)

This is useful when developing standalone console applications

Because using stdlib.ring (functions & classes) will load libraries like RingLibCurl, RingOpenSSL, etc.

43.1 IsAppCompiled() function

check whether the application has been compiled using Ring2EXE

Syntax:

```
IsAppCompiled() ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test IsAppCompiled()")
if IsAppCompiled() see "Application has been compiled using Ring2EXE"
else see "Application is running under Ring interpreter" ok
```

43.2 AppArguments() function

Get the effective arguments passed to the Ring script

Syntax:

```
AppArguments() ---> The arguments as a list of strings
```

Example:

```
Load "stdlib.ring"

# Application Arguments
Puts("Test AppArguments()")
argsList = AppArguments()
argsCount = Len(argsList)
if argsCount = 0 see "No arguments passed to the Ring script" + nl
else see "Ring script arguments = " + nl + list2str(argsList) + nl ok
```

43.3 AppPath() function

Get the path of the application folder

Syntax:

```
AppPath() ---> The path as String
```

Example:

```
Load "stdlib.ring"

# Application Path
Puts("Test AppPath()")
See AppPath() + nl
```

43.4 JustFilePath() function

Get the path of the file, remove the file name.

Syntax:

```
JustFilePath(cFile) ---> The path as String
```

Example:

```
load "stdlib.ring"

see  justfilePath("b:\ring\applications\rnote\rnote.ring")
```

Output:

```
b:\ring\applications\rnote\
```

43.5 JustFileName() function

Get the file, remove the file path.

Syntax:

```
JustFileName(cFile) ---> The file name as String
```

Example:

```
load "stdlib.ring"

see justfileName("b:\ring\applications\rnote\rnote.ring")
```

Output:

```
rnote.ring
```

43.6 Value() function

create a copy from a list or object

Syntax:

```
value(List) ---> new list
```

Example:

```
Load "stdlib.ring"

aList = 1:10
del(value(aList),1) # delete first item
see aList           # print numbers from 1 to 10
```

43.7 Times() function

Execute a Function nCount times

Syntax:

```
Times(nCount,function)
```

Example:

```
Load "stdlib.ring"

Puts("Test Times()")
Times ( 3 , func { see "Hello, World!" + nl } )
```

43.8 Map() function

Execute a Function on each list item

Syntax:

```
Map(alist,function) ---> List
```

Example:

```
Load "stdlib.ring"

Puts("Test Map()")
See Map( 1:10, func x { return x*x } )
```

43.9 Filter() function

Execute a Function on each list item to filter items

Syntax:

```
Filter(alist,function) ---> List
```

Example:

```
Load "stdlib.ring"

Puts("Test Filter()")
See Filter( 1:10 , func x { if x <= 5 return true else return false ok } )
```

43.10 Reduce() function

Apply function cFunc to each result xResult from a list aList, return an accumulated value xResult

The input list aList, the optional intial value xInitial and the output xResult, need to be the same Type

Syntax:

```
Reduce(aList,cFunc,xInitial) ---> final value
```

Example:

```
Load "stdlib.ring"

? Reduce( 1:3, func x, y { return x + y }, 0 )
? Reduce( ["I","Love","Ring"], func x, y { return x + y }, "" )
```

Output:

```
6
ILoveRing
```

43.11 Split() function

Convert string words to list items

Syntax:

```
Split(cstring,delimiter) ---> List
```

Example:

```
Load "stdlib.ring"

Puts("Test Split()")
See Split("one two three four five"," ")
```

43.12 SplitMany() function

Convert string words to list items. Allow many delimiters.

Syntax:

```
SplitMany(cstring,delimiters as string or list) --> List
```

Example:

```
Load "stdlib.ring"

Puts("Test SplitMany()")
See SplitMany("one,two,three,four and five"," ,")
```

43.13 Capitalized() function

Return a copy of a string with the first letter capitalized

Syntax:

```
Capitalized(string) ---> string
```

Example:

```
Load "stdlib.ring"

Puts("Test Capitalized()")
See capitalized("welcome to the Ring Programming Language")
```

43.14 IsSpecial() function

Check whether a character is special or not

Syntax:

```
IsSpecial(char) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Isspecial()")
See "Isspecial  = " + isSpecial("%") + nl
```

43.15 IsVowel() function

Check whether a character is vowel or not

Syntax:

```
IsVowel(char) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Isvowel()")
See "Isvowel = " + isVowel("c") + nl
```

43.16 LineCount() function

Return the lines count in a text file.

Syntax:

```
LineCount(cFileName) ---> Lines Count as number
```

Example:

```
Load "stdlib.ring"

Puts("Test Linecount()")
See "the number of lines = " + lineCount("test.ring")
```

43.17 Factorial() function

Return the factorial of a number

Syntax:

```
Factorial(number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Factorial()")
see "6 factorial is : " + Factorial(6)
```

43.18 Fibonacci() function

Return the fibonacci number

Syntax:

```
Fibonacci(number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Fibonacci()")
see "6 Fibonacci is : " + Fibonacci(6)
```

43.19 IsPrime() function

Check whether a number is prime or not

Syntax:

```
isprime(number) ---> Number
```

Example:

```
Load "stdlib.ring"

Puts("Test Isprime()")
if isPrime(16) see "16 is a prime number"
else see "16 is not a prime number" ok
```

43.20 Sign() function

Returns an integer value indicating the sign of a number.

Syntax:

```
Sign(number) ---> number ( -1 = negative , 0 , 1 (positive) )
```

Example:

```
Load "stdlib.ring"

Puts("Test Sign()")
see "sign of 12 is = " + sign(12) + nl
```

43.21 List2File() function

Write list items to text file (each item in new line).

Syntax:

```
List2File(aList,cFileName)
```

Example:

```
Load "stdlib.ring"

# Test List2File
Puts("Test List2File()")
list2file(1:100,"myfile.txt")
```

43.22 File2List() function

Read text file and convert lines to list items

Syntax:

```
File2List(cFileName) ---> List
```

Example:

```
Load "stdlib.ring"

# Test File2List
Puts("Test File2List()")
see len(file2list("myfile.txt"))
```

43.23 StartsWith() function

Returns true if the given string starts with the specified substring.

Leading white spaces are ignored.

Syntax:

```
StartsWith(string, substring) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Startswith()")
see Startswith("CalmoSoft", "Calmo") + nl
```

43.24 EndsWith() function

Returns true if the given string ends with the specified substring.

Trailing white spaces are ignored.

Syntax:

```
Endswith(string, substring) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Endswith()")
see endsWith("CalmoSoft", "Soft") + nl
```

43.25 GCD() function

Finding of the greatest common divisor of two integers.

Syntax:

```
Gcd(number,number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Gcd()")
see gcd (24, 32) + nl
```

43.26 LCM() function

Compute the least common multiple of two integers.

Syntax:

```
lcm(number,number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Lcm()")
see Lcm(24,36) + nl
```

43.27 SumList() function

Compute the sum of a list of integers.

Syntax:

```
sumlist(list) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Sumlist()")
aList = [1,2,3,4,5]
see Sumlist(aList) + nl
```

43.28 ProdList() function

Compute the product of a list of integers.

Syntax:

```
prodlist(list) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Prodlist()")
aList = [1,2,3,4,5]
see Prodlist(aList) + nl
```

43.29 EvenOrOdd() function

Test whether an integer is even or odd.

Result of test (1=odd 2=even).

Syntax:

```
evenorodd(number) ---> 1 (odd) or 2 (even)
```

Example:

```
Load "stdlib.ring"

Puts("Test Evenorodd()")
nr = 17
see Evenorodd(nr) + nl
```

43.30 Factors() function

Compute the factors of a positive integer.

Syntax:

```
factors(number) ---> list
```

Example:

```
Load "stdlib.ring"

Puts("Test Factors()")
n = 45
aList = factors(n)
see "Factors of " + n + " = "
for i = 1 to len(aList)
    see "" + aList[i] + " "
next
```

43.31 IsPalindrome() function

Check if a sequence of characters is a palindrome or not.

Syntax:

```
IsPalindrome(String) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test IsPalindrome()")
```

(continues on next page)

(continued from previous page)

```
cString = "radar"
see IsPalindrome(cString)
```

43.32 IsLeapYear() function

Check whether a given year is a leap year in the Gregorian calendar.

Syntax:

```
Isleapyear(number) ---> True/False
```

Example:

```
Load "stdlib.ring"

Puts("Test Isleapyear()")
year = 2016
if Isleapyear(year) see "" + year + " is a leap year."
else see "" + year + " is not a leap year." ok
```

43.33 BinaryDigits() function

Compute the sequence of binary digits for a given non-negative integer.

Syntax:

```
binarydigits(number) ---> string
```

Example:

```
Load "stdlib.ring"

Puts("Test Binarydigits()")
b = 35
see "Binary digits of " + b + " = " + Binarydigits(b)
```

43.34 MatrixMulti() function

Multiply two matrices together.

Syntax:

```
Matrixmulti(List,List) ---> List
```

Example:

```
Load "stdlib.ring"

# Multiply two matrices together.
Puts("Test Matrixmulti()")
A = [[1,2,3], [4,5,6], [7,8,9]]
B = [[1,0,0], [0,1,0], [0,0,1]]
see Matrixmulti(A, B)
```

43.35 MatrixTrans() function

Transpose an arbitrarily sized rectangular Matrix.

Syntax:

```
Matrixtrans(List) ---> List
```

Example:

```
Load "stdlib.ring"

# Transpose an arbitrarily sized rectangular Matrix.
Puts("Test Matrixtrans()")
matrix = [[78,19,30,12,36], [49,10,65,42,50], [30,93,24,78,10], [39,68,27,64,29]]
see Matrixtrans(matrix)
```

43.36 DayOfWeek() function

Return the day of the week of given date. (yyyy-mm-dd)

Syntax:

```
dayofweek(string) ---> string
```

Example:

```
Load "stdlib.ring"

# Return the day of the week of given date.
Puts("Test Dayofweek()")
date = "2016-04-24"
see "Data : " + date + " - Day : " + Dayofweek(date) + nl
```

43.37 Permutation() function

Generates all permutations of n different numerals.

Syntax:

```
permutation(list)
```

Example:

```
Load "stdlib.ring"

# Generates all permutations of n different numerals
Puts("Test Permutation()")
list = [1, 2, 3, 4]
for perm = 1 to 24
        for i = 1 to len(list)
                see list[i] + " "
        next
        see nl
        Permutation(list)
next
```

43.38 ReadLine() function

Read line from file

Syntax:

```
readline(fp) ---> string
```

Example:

```
Load "stdlib.ring"

# Read a file line by line.
Puts("Test Readline()")
fp = fopen("test.ring","r")
while not feof(fp)
See Readline(fp) end
fclose(fp)
```

43.39 SubString() function

Return a position of a substring starting from a given position in a string.

Syntax:

```
Substring(str,substr,npos) ---> string
```

Example:

```
Load "stdlib.ring"

# Return a position of a substring starting from a given position in a string.
Puts("Test Substring()")
a = "abcxyzqweabc"
b = "abc"
i = 4
see substring(a,b,i)
```

43.40 ChangeString() function

Change substring from given position to a given position with another substring.

Syntax:

```
Changestring(cString, nPos1, nPos2, cSubstr) ---> cString
```

Example:

```
Load "stdlib.ring"

# Change substring from given position for given position with a substring.
Puts("Test Changestring()")
see Changestring("Rmasdg",2,5,"in")     # Ring
```

43.41 Sleep() function

Sleep for the given amount of time.

Syntax:

```
sleep(nSeconds)
```

Example:

```
Load "stdlib.ring"

Puts("Test Sleep()")
see "Wait 3 Seconds!"
Sleep(3)
see nl
```

43.42 IsMainSourceFile() function

Check if the current file is the main source file

Syntax:

```
IsMainSourceFile() ---> True/False
```

Example:

```
Load "stdlib.ring"

if ismainsourcefile()
        # code
ok
```

43.43 MakeDir() function

Make Directory

Syntax:

```
MakeDir(String)
```

Example:

```
Load "stdlib.ring"

# Create Directory
puts("create Directory : myfolder")
makedir("myfolder")
```

43.44 Fsize() function

The function return the file size in bytes.

Syntax:

```
FSize(File Handle) ---> Number (File Size in Bytes)
```

43.45 TrimAll() function

Remove all spaces and tabs characters from a string

Syntax:

```
TrimAll(cString) ---> cString # Without Spaces and Tabs
```

43.46 TrimLeft() function

Remove all spaces and tabs characters from the left side of a string

Syntax:

```
TrimLeft(cString) ---> cString # Without Spaces and Tabs from the left side
```

43.47 TrimRight() function

Remove all spaces and tabs characters from the right side of a string

Syntax:

```
TrimRight(cString) ---> cString # Without Spaces and Tabs from the right side
```

43.48 EpochTime() function

Return the Epoch Time

Syntax:

```
EpochTime(cDate,cTime) ---> nEpochTime
```

Example:

```
see EpochTime( Date(), Time() )
```

43.49 SystemCmd() Function

We can execute system commands using the SystemCmd() function that outputs to a variable

Syntax:

```
SystemCmd(cCommand)
```

Example:

```
cYou  = SystemCmd("whoami")        # User Name logged in is output a variable
cThem = SystemCmd("dir c:\Users")  # Directory List is output to a variable
```

43.50 ListAllFiles() Function

Using this function we can quickly do a process on a group of files in a folder and it's sub folders.

Syntax:

```
ListAllFiles(cFolder,cExtension) ---> List of Files
```

Example:

```
aList = ListAllFiles("c:/ring/ringlibs","ring") # *.ring only
aList = sort(aList)
see aList
```

Example:

```
see listallfiles("b:/ring/libraries/weblib","") # All Files
```

43.51 SystemSilent() Function

We can execute system commands using the SystemSilent() function to avoid displaying the output!

Syntax:

```
SystemSilent(cCommand)
```

43.52 OSCreateOpenFolder() Function

Create folder then change the current folder to this new folder

Syntax:

```
OSCreateOpenFolder(cCommand)
```

43.53 OSCopyFolder() Function

Copy folder to the current folder

Parameters : The path to the parent folder and the folder name to copy

Syntax:

```
OSCopyFolder(cParentFolder,cFolderName)
```

Example

To copy the folder b:\ring\ringlibs\stdlib to the current folder

```
OSCopyFolder("b:\ring\ringlibs\","stdlib")
```

43.54 OSDeleteFolder() Function

Delete Folder in the current Directory

Syntax:

```
OSDeleteFolder(cFolderName)
```

43.55 OSCopyFile() Function

Copy File to the current directory

Syntax:

```
OSCopyFile(cFileName)
```

43.56 OSDeleteFile() Function

Delete File

Syntax:

```
OSDeleteFile(cFileName)
```

43.57 OSRenameFile() Function

Rename File

Syntax:

```
OSRenameFile(cOldFileName,cNewFileName)
```

43.58 List2Code() Function

This function convert a Ring list during the runtime to Ring source code that we can save to source files.

The list may contains strings, numbers or sub lists.

Example:

```
load "stdlibcore.ring"
aList = 1:10
? list2Code(aList)
```

Output:

```
[
        1,2,3,4,5,6,7,8,9,10
]
```

43.59 Str2ASCIIList()

Convert a string of bytes to a list of numbers where each item represent the ASCII code of one byte in the string.

Syntax:

```
Str2ASCIIList(String) ---> List of numbers
```

43.60 ASCIIList2Str()

Convert a list of numbers where each item represent the ASCII code of one byte to a string of bytes.

Syntax:

```
ASCIIList2Str(List of numbers) ---> String
```

Example:

```
load "stdlibcore.ring"

cStr = "MmMm"

aList = Str2ASCIILIST(cStr)
? aList

cStr2 = ASCIIList2Str(aList)
? cStr2
? len(cStr2)
```

Output:

```
77
109
77
109

MmMm
4
```

43.61 IsListContainsItems()

Syntax:

```
IsListContainsItems(aParent,aChild) ----> True/False
```

Example:

```
load "stdlibcore.ring"
aList1 = "a":"z"
aList2 = [:h,:l,:p,:u]
? IsListContainsItems(aList1,aList2)
```

43.62 IsBetween()

Syntax:

```
IsBetween(nNumber,nMin,nMax) ----> True/False
```

Example:

```
load "stdlibcore.ring"
? isBetween(1,3,4)
? isBetween(1,-3,4)
? isBetween(4,1,6)
? isBetween(4,3,4)
```

43.63 TimeInfo()

Syntax:

```
TimeInfo(cInformation) ----> String
```

The cInformation value could be

```
:hour_24
:hour_12
:minutes
:seconds
:time
:day_short
:day_long
:month_short
:month_long
:date_time
:day
:day_year
:month_year
:am_pm
:week_year
:day_week
:date
:year_century
:year
:time_zone
:percent_sign
```

Example:

```
load "stdlibcore.ring"
? timeInfo(:date)
? timeInfo(:time)
? timeInfo(:hour_12)
```

43.64 RandomList()

Syntax:

```
RandomList(aList) --> List contains the same items using Random order
```

Example:

```
load "stdlib.ring"
aList = 1:5
? RandomList(aList)
```

43.65 RandomItem()

Pick an item from a list (Random Choice)

Syntax:

```
RandomItem(aList) --> Item
```

Example:

```
load "stdlib.ring"
aList = 1:5
? RandomItem(aList)
```

43.66 CheckEquality()

Check if two items are equal. Deep comparison is performed if the two items are lists Return 1 if both items are equal and 0 otherwise

Syntax:

```
CheckEquality(aItem1,aItem2) --> value = 1 if aItem1 = aItem2
                                 value = 0 if aItem1 != aItem2
```

Example:

```
load "stdlib.ring"
aList1 = ["one", 2, [3]]
aList2 = ["one", 2]
aList2 + [3]
? CheckEquality(aList1,aList2)
```

CHAPTER

FORTYFOUR

STDLIB CLASSES

In this chapter we are going to learn about the classes in the stdlib.ring

- StdBase Class
- String Class
- List Class
- Stack Class
- Queue Class
- HashTable Class
- Tree Class
- Math Class
- DateTime Class
- File Class
- System Class
- Debug Class
- DataType Class
- Conversion Class
- ODBC Class
- MySQL Class
- SQLite Class
- PostgreSQL Class
- Security Class
- Internet Class

44.1 StdBase Class

Attributes:

- vValue : Object Value

Methods:

Method	Description/Output
Init(x)	Set vValue Attribute to x value
Print()	Print vValue
PrintLn()	Print vValue then New Line
Size()	return number represent the size of vValue
Value()	return vValue
Set(x)	Call Init(x)

44.2 String Class

Parent Class : StdBase Class

Methods:

Method	Description/Output
Init(String|Number|List)	
Lower()	New String - Lower case characters
Upper()	New String - Upper case characters
Left(x)	New String - contains x characters from the left
Right(x)	New String - contains x characters from the right
Lines()	Number - Lines count
Trim()	New String - Remove Spaces
Copy(x)	New String - repeat string x times
strcmp(cString)	Compare string with cString
tolist()	List (String Lines to String Items)
tofile(cFileName)	Write string to file
mid(nPos1,nPos2)	New String - from nPos1 to nPos2
getfrom(nPos1)	New String - from nPos1 to the end of the string
replace(cStr1,cStr2,lCase)	New String - Replace cStr1 with cStr2 , lCase (True=Match Case)
split()	List - Each Word as list item
startswith(substring)	Return true if the start starts with a substring
endswith(substring)	Return true if the start ends with a substring

Example:

```
Load "stdlib.ring"

See "Testing the String Class" + nl
oString = new string("Hello, World!")
oString.println()
oString.upper().println()
```

(continues on next page)

(continued from previous page)

```
oString.lower().println()
oString.left(5).println()
oString.right(6).println()
oString = new string("Hi" + nl + "Hello" )
See oString.lines() + nl
oString = new string("    Welcome    ")
oString.println()
oString.trim().println()
oString = new string("Hello! ")
oString.copy(3).println()
see oString.strcmp("Hello! ") + nl
see oString.strcmp("Hello ") + nl
see oString.strcmp("Hello!! ") + nl
oString = new string(["one","two","three"])
oString.print()
see oString.lines() + nl
oString = new String(1234)
oString.println()
oString = new String("one"+nl+"two"+nl+"three")
aList = oString.tolist()
see "List Items" + nl See aList
oString = new String( "Welcome to the Ring programming language")
See "the - position : " + oString.pos("the") + nl
oString = oString.getfrom(oString.pos("Ring"))
oString.println()
oString.mid(1,4).println()
oString = oString.replace("Ring","***Ring***",true)
oString.println()
oString = oString.replace("ring","***Ring***",false)
oString.println()
oString1 = new string("First")
oString2 = new string("Second")
oString = oString1 + oString2
oString.println()
oString = oString1 * 3
oString.println()
for t in ostring see t next
oString.tofile("test.txt")
oString = new string("one two three")
see nl
see ostring.split()
oString {
        set("Hello") println()
        set("How are you?") println()
}
```

Output:

```
Testing the String Class
Hello, World!
HELLO, WORLD!
hello, world!
```

(continues on next page)

(continued from previous page)

```
Hello
World!
2
        Welcome
Welcome
Hello! Hello! Hello!
0
1
-1
one
two
three
4
1234
List Items
one
two
three
the - position : 12
Ring programming language
Ring
***Ring*** programming language
******Ring****** programming language
FirstSecond
FirstFirstFirst
FirstFirstFirst
one
two
three
Hello
How are you?
```

44.3 List Class

Parent Class : StdBase Class

Methods:

Method	Description/Output
Init(String|List)	
Add(Value)	Add item to the list
Delete(nIndex)	Delete item from the list
Item(nIndex)	Get item from the list
First()	Get the first item in the list
Last()	Get the last item in the list
Set(nIndex,Value)	Set item value
FindInColumn(nCol,Value)	Find item in a column
Sort()	Sort items - return new list
Reverse()	Reverse items - return new list
Insert(nIndex,Value)	Inset Item after nIndex

example:

```
Load "stdlib.ring"

oList = new list ( [1,2,3] )
oList.Add(4)
oList.print()
see oList.item(1) + nl
oList.delete(4)
oList.print()
see oList.first() + nl
see oList.last() + nl
oList { set(1,"one") set(2,"two") set(3,"three") print() }
see oList.find("two") + nl
oList.sort().print()
oList.reverse().print()
oList.insert(2,"nice")
oList.print()
oList = new list ( [ [1,"one"],[2,"two"],[3,"three"] ] )
see copy("*",10) + nl
oList.print()
see "Search two : " + oList.findincolumn(2,"two") + nl
see "Search 1 : " + oList.findincolumn(1,1) + nl
oList = new list ( [ "Egypt" , "USA" , "KSA" ] )
for x in oList
        see x + nl
next
oList =  new list ( [1,2,3,4] )
oList + [5,6,7]
oList.print()
oList = new list ( ["one","two"] )
oList2 = new list ( ["three","four"] )
oList + oList2
oList.print()
```

output:

```
1
2
3
4
1
1
2
3
1
3
one
two
three
2
one
three
two
```

(continues on next page)

(continued from previous page)

```
three
two
one
one
two
nice
three
**********
1
one
2
two
3
three
Search two : 2
Search 1 : 1
Egypt
USA
KSA
1
2
3
4
5
6
7
one
two
three
four
```

44.4 Stack Class

Parent Class : List Class

Methods:

Method	Description/Output
Init(String|Number|List)	
Push(Value)	Push item to the stack
Pop()	Pop item from the stack
Print()	Print the stack items

example:

```
Load "stdlib.ring"

oStack = new Stack
oStack.push(1)
```

(continues on next page)

(continued from previous page)

```
oStack.push(2)
oStack.push(3)
see oStack.pop() + nl
see oStack.pop() + nl
see oStack.pop() + nl
oStack.push(4)
see oStack.pop() + nl
oStack { push("one") push("two") push("three") }
oStack.print()
```

output:

```
3
2
1
4
three
two
one
```

44.5 Queue Class

Parent Class : List Class

Methods:

Method	Description/Output
Init(String|Number|List)	
Remove()	Remove item from the Queue.

example:

```
Load "stdlib.ring"

oQueue = new Queue
oQueue.add(1)
oQueue.add(2)
oQueue.add(3)
see oQueue.remove() + nl
see oQueue.remove() + nl
see oQueue.remove() + nl
oQueue.add(4)
see oQueue.remove() + nl
oQueue { add("one") add("two") add("three") }
oQueue.print()
```

output:

```
1
2
```

(continues on next page)

(continued from previous page)

```
3
4
one
two
three
```

44.6 HashTable Class

Parent Class : List Class

Methods:

Method	Description/Output
Init(List)	
Add(cKey,Value)	Add item to the HashTable
Set(cKey,Value)	Set item value using the Key
GetValue(cKey)	Get item value using the Key
Contains(cKey)	Check if the HashTable contains item using the Key
Index(cKey)	Get the item index using the Key

example:

```
Load "stdlib.ring"

ohashtable = new hashtable
See "Test the hashtable Class Methods" + nl
ohashtable {
        Add("Egypt","Cairo")
        Add("KSA","Riyadh")
        see self["Egypt"] + nl
        see self["KSA"] + nl
        see contains("Egypt") + nl
        see contains("USA") + nl
        see index("KSA")  + NL
        print()
        delete(index("KSA"))
        see copy("*",60) + nl
        print()
}
```

output:

```
Test the hashtable Class Methods
Cairo
Riyadh
1
0
2
Egypt
```

(continues on next page)

(continued from previous page)

```
Cairo
KSA
Riyadh
*************************************************************
Egypt
Cairo
```

44.7 Tree Class

Data:

Attribute	Description
Data	Node Value
Children	Children List

Methods:

Method	Description/Output
set(value)	Set the node value.
value()	Get the node value.
Add(value)	Add new child.
parent()	Get the parent node.
print()	Print the tree nodes.

example:

```
Load "stdlib.ring"

otree = new tree
See "Test the tree Class Methods" + nl
otree {
        set("The first step")   # set the root node value
        see value() + nl
        Add("one")
        Add("two")
        Add("three") {
                Add("3.1")
                Add("3.2")
                Add("3.3")
                see children
        }
        see children
        oTree.children[2] {
                Add("2.1") Add("2.2") Add("2.3") {
                        Add("2.3.1") Add("2.3.2") Add("test")
                }
        }
```

(continues on next page)

(continued from previous page)

```
        oTree.children[2].children[3].children[3].set("2.3.3")
}
see copy("*",60) + nl
oTree.print()
```

output:

```
Test the tree Class Methods
The first step
data: 3.1
parent: List...
children: List...
data: 3.2
parent: List...
children: List...
data: 3.3
parent: List...
children: List...
data: one
parent: List...
children: List...
data: two
parent: List...
children: List...
data: three
parent: List...
children: List...
************************************************************
one
two
2.1
2.2
2.3
2.3.1
2.3.2
2.3.3
three
3.1
3.2
3.3
```

44.8 Math Class

Methods:

Method	Description
sin(x)	Returns the sine of an angle of x radians
cos(x)	Returns the cosine of an angle of x radians
tan(x)	Returns the tangent of an angle of x radians
asin(x)	Returns the principal value of the arc sine of x, expressed in radians
acos(x)	Returns the principal value of the arc cosine of x, expressed in radians
atan(x)	Returns the principal value of the arc tangent of x, expressed in radians
atan2(y,x)	Returns the principal arc tangent of y/x, in the interval [-pi,+pi] radians
sinh(x)	Returns the hyperbolic sine of x radians
cosh(x)	Returns the hyperbolic cosine of x radians
tanh(x)	Returns the hyperbolic tangent of x radians
exp(x)	Returns the value of e raised to the xth power
log(x)	Returns the natural logarithm of x
log10(x)	Returns the common logarithm (base-10 logarithm) of x
ceil(x)	Returns the smallest integer value greater than or equal to x
floor(x)	Returns the largest integer value less than or equal to x
fabs(x)	Returns the absolute value of x.
pow(x,y)	Returns x raised to the power of y
sqrt(x)	Returns the square root of x
random(x)	Returns a random number in the range [0,x]
unsigned(n,n,c)	Perform operation using unsigned numbers
decimals(n)	Determine the decimals digits after the point in float/double numbers

example:

```
Load "stdlib.ring"

oMath = new Math

See "Test the Math Class Methods" + nl
See "Sin(0) = " + oMath.sin(0) + nl
See "Sin(90) radians = " + oMath.sin(90) + nl
See "Sin(90) degree = " + oMath.sin(90*3.14/180) + nl

See "Cos(0) = " + oMath.cos(0) + nl
See "Cos(90) radians = " + oMath.cos(90) + nl
See "Cos(90) degree = " +oMath. cos(90*3.14/180) + nl

See "Tan(0) = " + oMath.tan(0) + nl
See "Tan(90) radians = " + oMath.tan(90) + nl
See "Tan(90) degree = " + oMath.tan(90*3.14/180) + nl

See "asin(0) = " + oMath.asin(0) + nl
See "acos(0) = " + oMath.acos(0) + nl
See "atan(0) = " + oMath.atan(0) + nl
See "atan2(1,1) = " +oMath. atan2(1,1) + nl
```

(continues on next page)

(continued from previous page)

```
See "sinh(0) = " + oMath.sinh(0) + nl
See "sinh(1) = " + oMath.sinh(1) + nl
See "cosh(0) = " + oMath.cosh(0) + nl
See "cosh(1) = " + oMath.cosh(1) + nl
See "tanh(0) = " + oMath.tanh(0) + nl
See "tanh(1) = " + oMath.tanh(1) + nl

See "exp(0) = " + oMath.exp(0) + nl
See "exp(1) = " + oMath.exp(1) + nl
See "log(1) = " + oMath.log(1) + nl
See "log(2) = " + oMath.log(2) + nl
See "log10(1) = " + oMath.log10(1) + nl
See "log10(2) = " + oMath.log10(2) + nl
See "log10(10) = " + oMath.log10(10) + nl

See "Ceil(1.12) = " + oMath.Ceil(1.12) + nl
See "Ceil(1.72) = " + oMath.Ceil(1.72) + nl

See "Floor(1.12) = " + oMath.floor(1.12) + nl
See "Floor(1.72) = " + oMath.floor(1.72) + nl

See "fabs(1.12) = " + oMath.fabs(1.12) + nl
See "fabs(1.72) = " + oMath.fabs(1.72) + nl

See "pow(2,3) = " + oMath.pow(2,3) + nl

see "sqrt(16) = " + oMath.sqrt(16) + nl

for x = 1 to 20
                see "Random number Max (100) : " + oMath.random(100) + nl
next

x = 1.1234567890123
for d = 0 to 14
        oMath.decimals(d)
        see x + nl
next

cKey = "hello"

h = 0
for x in cKey
        h = oMath.unsigned(h,ascii(x),"+")
        h = oMath.unsigned(h,oMath.unsigned(h,10,"<<"),"+")
        r = oMath.unsigned(h,6,">>")
        h = oMath.unsigned(h, r,"^")
next
h = oMath.unsigned(h,oMath.unsigned(h,3,"<<"),"+")
h = oMath.unsigned(h,oMath.unsigned(h,11,">>"),"^")
h = oMath.unsigned(h,oMath.unsigned(h,15,"<<"),"+")

see "Hash : " + h
```

output:

```
Test the Math Class Methods
Sin(0) = 0
Sin(90) radians = 0.89
Sin(90) degree = 1.00
Cos(0) = 1
Cos(90) radians = -0.45
Cos(90) degree = 0.00
Tan(0) = 0
Tan(90) radians = -2.00
Tan(90) degree = 1255.77
asin(0) = 0
acos(0) = 1.57
atan(0) = 0
atan2(1,1) = 0.79
sinh(0) = 0
sinh(1) = 1.18
cosh(0) = 1
cosh(1) = 1.54
tanh(0) = 0
tanh(1) = 0.76
exp(0) = 1
exp(1) = 2.72
log(1) = 0
log(2) = 0.69
log10(1) = 0
log10(2) = 0.30
log10(10) = 1
Ceil(1.12) = 2
Ceil(1.72) = 2
Floor(1.12) = 1
Floor(1.72) = 1
fabs(1.12) = 1.12
fabs(1.72) = 1.72
pow(2,3) = 8
sqrt(16) = 4
Random number Max (100) : 87
Random number Max (100) : 49
Random number Max (100) : 99
Random number Max (100) : 58
Random number Max (100) : 15
Random number Max (100) : 46
Random number Max (100) : 37
Random number Max (100) : 64
Random number Max (100) : 73
Random number Max (100) : 35
Random number Max (100) : 89
Random number Max (100) : 80
Random number Max (100) : 20
Random number Max (100) : 33
Random number Max (100) : 44
Random number Max (100) : 89
```

(continues on next page)

(continued from previous page)

```
Random number Max (100) : 82
Random number Max (100) : 94
Random number Max (100) : 83
Random number Max (100) : 68
1
1.1
1.12
1.123
1.1235
1.12346
1.123457
1.1234568
1.12345679
1.123456789
1.1234567890
1.12345678901
1.123456789012
1.1234567890123
1.12345678901230
Hash : 3372029979.00000000000000
```

44.9 DateTime Class

Methods:

Method	Description/Output
clock()	The number of clock ticks from program start.
time()	Get the system time.
date()	Get the date.
timelist()	List contains the date and the time information.
adddays(cDate,nDays)	Return Date from cDate and after nDays
diffdays(cDate1,cDate2)	Return the Number of days (cDate1 - cDate2)

example:

```
Load "stdlib.ring"

oDateTime = new datetime

See "Test the datetime Class Methods" + nl

See "Calculate performance" + nl
t1 = oDateTime.clock()
for x = 1 to 1000000 next
see oDateTime.clock() - t1 + nl

See "Time : " + oDateTime.time() + nl

See "Date : " + oDateTime.date() + nl
```

(continues on next page)

(continued from previous page)

```
See oDateTime.TimeList()

See "Month Name : " + oDateTime.TimeList()[4]

cDate = oDateTime.date()
see cDate + nl
cDate = oDateTime.adddays(cDate,10)
see cDate + nl

cDate1 = oDateTime.date()
see cDate1 + nl
cDate2 = oDateTime.adddays(cDate1,10)
see cDate2 + nl
see "DiffDays = " + oDateTime.diffdays(cDate1,cDate2) + nl
see "DiffDays = " + oDateTime.diffdays(cDate2,cDate1) + nl
```

output:

```
Test the datetime Class Methods
Calculate performance
85
Time : 02:53:35
Date : 31/08/2016
Wed
Wednesday
Aug
August
08/31/16 02:53:35
31
02
02
244
08
53
AM
35
35
3
08/31/16
02:53:35
16
2016
Arab Standard Time
%
Month Name : August31/08/2016
10/09/2016
31/08/2016
10/09/2016
DiffDays = -10
DiffDays = 10
```

44.10 File Class

Methods:

Method	Description/Output
read(cFileName)	Read the file content
write(cFileName,cStr)	Write string to file
dir(cFolderPath)	Get the folder contents (files & sub folders)
rename(cOld,cNew)	Rename files using the Rename() function
remove(cFileName)	Delete a file using the Remove() function
open(cFileName,cMode)	Open a file using the Fopen() function
close()	Close file
flush()	Flushes the output buffer of a stream
reopen(cFileName,cMode)	Open another file using the same file handle
tempfile()	Creates a temp. file (binary).
seek(noffset,nwhence)	Set the file position of the stream
tell()	Know the current file position of a stream
rewind()	Set the file position to the beginning of the file
getpos()	Get handle to the current file position
setpos(poshandle)	Set the current file position
clearerr()	Clear the EOF error and the error indicators of a stream
eof()	Test the end-of-file indicator
error()	Test the error indicator
perror(cErrorMessage)	Print error message to the stderr
getc()	Get the next character from the stream
gets(nsize)	Read new line from the stream
putc(cchar)	Write a character to the stream
puts(cStr)	Write a string to the stream
ungetc(cchar)	Push a character to the stream
fread(nsize)	Read data from a stream
fwrite(cString)	Write data to a stream
exists(cFileName)	Check if a file exists

example:

```
Load "stdlib.ring"

ofile = new file

See "Test the file Class Methods" + nl
see ofile.read(filename())

see nl
ofile.open(filename(),"r")
see ofile.gets(100) + nl
ofile.close()
```

44.11 System Class

Methods:

Method	Description/Output
system()	Execute system commands
sysget()	Get environment variables
ismsdos()	Check if the operating system is MSDOS or not
iswindows()	Check if the operating system is Windows or not
iswindows64()	Check if the operating system is Windows 64bit or not
isunix()	Check if the operating system is Unix or not
ismacosx()	Check if the operating system is macOS or not
islinux()	Check if the operating system is Linux or not
isfreebsd()	Check if the operating system is FreeBSD or not
isandroid()	Check if the operating system is Android or not
windowsnl()	Get the windows new line string
sysargv()	Get the command line arguments passed to the ring script
filename()	Get the active source file

example:

```
Load "stdlib.ring"

oSystem = new System

See "Test the System Class Methods" + nl

oSystem.system("dir")
see oSystem.sysget("path") + nl
see oSystem.ismsdos() + nl
see oSystem.iswindows() + nl
see oSystem.iswindows64() + nl
see oSystem.isunix() + nl
see oSystem.ismacosx() + nl
see oSystem.islinux() + nl
see oSystem.isfreebsd() + nl
see oSystem.isandroid() + nl
see oSystem.windowsnl() + nl
see oSystem.sysargv() + nl
see oSystem.filename() + nl
```

44.12 Debug Class

Methods:

Method	Description/Output
eval(cCode)	Execute code during the runtime from string.
raise(cError)	Raise an exception.
assert(cCondition)	Test condition before executing the code.

example:

```
Load "stdlib.ring"

oDebug = new Debug
See "Test the Debug Class Methods" + nl
oDebug.eval("see 'Hello'+nl")
try
        x = 10
        oDebug.assert(x=11)
catch see "assert" + nl done
raise("Error!")
```

44.13 DataType Class

Methods:

Method	Description/Output
isstring(vValue)	We can know if the value is a string or not.
isnumber(vValue)	We can know if the value is a number or not.
islist(vValue)	We can know if the value is a list or not.
type(vValue)	Know the type of a value
isnull(vValue)	Check the value to know if it's null or not.
isalnum(vValue)	1 if the value is digit/letter or 0 if not
isalpha(vValue)	1 if the value is a letter or 0 if not
iscntrl(vValue)	1 if the value is a control character (no printing position)
isdigit(vValue)	1 if the value is a digit or 0 if not
isgraph(vValue)	1 if the value can be printed (Except space) or 0 if not
islower(vValue)	1 if the value is lowercase letter or 0 if not
isprint(vValue)	1 if the value occupies a printing position or 0 if not
ispunct(vValue)	1 if the value is a punctuation character or 0 if not
isspace(vValue)	1 if the value is a white-space or 0 if not
isupper(vValue)	1 if the value is an uppercase alphabetic letter or 0 if not
isxdigit(vValue)	1 if the value is a hexadecimal digit character or 0 if not

example:

```
Load "stdlib.ring"

oDataType = new DataType
See "Test the DataType Class Methods" + nl
see oDataType.isstring("test") + nl
see oDataType.isnumber(1) + nl
see oDataType.islist(1:3) + nl
see oDataType.type("test") + nl
see oDataType.isnull(null) + nl
see oDataType.isalnum("Hello") + nl +     # print 1
oDataType.isalnum("123456") + nl +    # print 1
oDataType.isalnum("ABCabc123") + nl + # print 1
oDataType.isalnum("How are you")  + nl     # print 0 because of spaces
```

(continues on next page)

(continued from previous page)

```
see oDataType.isalpha("Hello") + nl +     # print 1
oDataType.isalpha("123456") + nl +    # print 0
oDataType.isalpha("ABCabc123") + nl + # print 0
oDataType.isalpha("How are you")  + nl    # print 0
See oDataType.iscntrl("hello") + nl +     # print 0
oDataType.iscntrl(nl)                 # print 1
see oDataType.isdigit("0123456789") + nl +        # print 1
oDataType.isdigit("0123a") + nl
see oDataType.isgraph("abcdef") + nl +    # print 1
oDataType.isgraph("abc def")   + nl        # print 0
see oDataType.islower("abcDEF") + nl +    # print 0
oDataType.islower("ghi") + nl          # print 1
see oDataType.isprint("Hello") + nl +             # print 1
oDataType.isprint("Nice to see you") + nl +   # print 1
oDataType.isprint(nl)    + nl                      # print 0
see oDataType.isprint("Hello") + nl              # print 1
see oDataType.isupper("welcome") + nl +    # print 0
oDataType.isupper("WELCOME")  + nl         # print 1
see oDataType.isxdigit("0123456789abcdef") + nl +  # print 1
oDataType.isxdigit("123z")                     # print 0
```

Output:

```
Test the DataType Class Methods
1
1
1
STRING
1
1
1
1
0
1
0
0
0
0
11
0
1
0
0
1
1
1
0
1
0
1
1
0
```

44.14 Conversion Class

Methods:

Method	Description/Output
number(vValue)	Convert strings to numbers.
string(vValue)	Convert numbers to strings.
ascii(vValue)	Get the ASCII code for a letter.
char(vValue)	Convert the ASCII code to character.
hex(vValue)	Convert decimal to hexadecimal.
dec(vValue)	Convert hexadecimal to decimal.
str2hex(vValue)	Convert string characters to hexadecimal characters.
hex2str(vValue)	Convert hexadecimal characters to string.

example:

```
Load "stdlib.ring"

oConversion = new conversion
See "Test the conversion Class Methods" + nl
See oConversion.number("3") + 5  + nl
See oConversion.string(3) + "5" + nl
See oConversion.Ascii("m") + nl
See oConversion.char(77) + nl
see oConversion.hex(162) + nl
see oConversion.dec("a2") + nl
cHex = oConversion.str2hex("Hello")
see cHex + nl
see oConversion.hex2str(cHex) + nl
```

Output:

```
Test the conversion Class Methods
8
35
109
M
a2
162
48656c6c6f
Hello
```

44.15 ODBC Class

Methods:

Method	Description/Output
drivers()	Get a list of ODBC drivers.
datasources()	Get a list of ODBC data sources.
close()	Free resources.
connect(cConString)	Connect to the database.
disconnect()	Close the connection.
execute(cSQL)	Execute SQL Statements
colcount()	Get columns count in the query result
fetch()	Fetch a row from the query result
getdata(nCol)	Get column value from the fetched row
tables()	Get a list of tables inside the database
columns(cTableName)	Get a list of columns inside the table
autocommit(lStatus)	Enable or disable the auto commit feature
commit()	Commit updates to the database
rollback()	Rollback updates to the database

example:

```
Load "stdlib.ring"

oodbc = new odbc
See "Test the odbc Class Methods" + nl
oODBC {
        see drivers()
        see datasources()
        See "Connect to database" + nl
        see connect("DBQ=test.mdb;Driver={Microsoft Access Driver (*.mdb)}") + nl
        See "Select data" + nl
        see execute("select * from person") + nl
        nMax = colcount()
        See "Columns Count : " + nMax + nl
        while fetch()
                See "Row data:" + nl
                for x = 1 to nMax
                        see getdata(x) + " - "
                next
        end
        See "Close database..." + nl
        disconnect()
        close()
}
```

44.16 MySQL Class

Methods:

Method	Description/Output
info()	Return string contains the MySQL Client version.
error()	Get the error message from the MySQL Client.
connect(cServer,cUser,cPass,cDatabase)	Connect to the MySQL database server.
close()	Close the connection to the MySQL database.
query(cQuery)	Execute SQL queries.
insert_id()	Get the inserted row id.
result()	Get the query result (data without column names).
next_result()	Move to the next query result.
columns()	Get a list of columns names.
result2()	Get all of the column names then the query result in one list.
escape_string(cStr)	Before storing binary data and special characters in the database.
autocommit(lStatus)	Enable or disable the auto commit feature.
commit()	Commit updates to the database.
rollback()	Rollback updates to the database.

example:

```
Load "stdlib.ring"

omysql = new mysql
See "Test the MySQL Class Methods" + nl
omysql {
        see info() + nl
        connect("localhost", "root", "root","mahdb")
        see "Execute Query" + nl
        query("SELECT * FROM Employee")
        see "Print Result" + nl
        see result2()
        see "Close database" + nl
        close()
}
```

Output:

```
Test the MySQL Class Methods
5.5.30
Execute Query
Print Result
Id
Name
Salary
1
Mahmoud
15000
2
Samir
16000
```

(continues on next page)

(continued from previous page)

```
3
Fayed
17000
Close database
```

44.17 SQLite Class

Methods:

Method	Description/Output
open(cDatabase)	Open Database.
close()	Close Database.
errormessage()	Get Error Message.
execute(cSQL)	Execute Query.

example:

```
Load "stdlib.ring"

osqlite = new sqlite
See "Test the sqlite Class Methods" + nl
osqlite {
        open("test.db")
        sql = "CREATE TABLE COMPANY("  +
                "ID INT PRIMARY KEY     NOT NULL," +
                 "NAME           TEXT    NOT NULL," +
                 "AGE            INT     NOT NULL," +
                 "ADDRESS        CHAR(50)," +
                 "SALARY         REAL );"

        execute(sql)

        sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "  +
                 "VALUES (1, 'Mahmoud', 29, 'Jeddah', 20000.00 ); " +
                 "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "  +
                 "VALUES (2, 'Ahmed', 27, 'Jeddah', 15000.00 ); "     +
                 "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
                 "VALUES (3, 'Mohammed', 31, 'Egypt', 20000.00 );" +
                 "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
                 "VALUES (4, 'Ibrahim', 24, 'Egypt ', 65000.00 );"

        execute(sql)

        aResult =  execute("select * from COMPANY")
        for x in aResult
                for t in x
                        see t[2] + nl
                next
        next
```

(continues on next page)

(continued from previous page)

```
        see copy("*",50)  + nl
        for x in aResult
                see x["name"] + nl
        next
        close()
}
```

Output:

```
Test the sqlite Class Methods
1
Mahmoud
29
Jeddah
20000.0
2
Ahmed
27
Jeddah
15000.0
3
Mohammed
31
Egypt
20000.0
4
Ibrahim
24
Egypt
65000.0
**************************************************
Mahmoud
Ahmed
Mohammed
Ibrahim
```

44.18 PostgreSQL Class

Methods:

Method	Description/Output
init(cConString)	Open Database.
close()	Close Database.
execute(cSQL)	Execute Query.

example:

```
load "stdlib.ring"
```

(continues on next page)

(continued from previous page)

```
oPostgreSQL = new PostgreSQL("user=postgres password=sa dbname = mahdb")

See "Test the PostgreSQL Class Methods" + nl

oPostgreSQL {

        sql = "CREATE TABLE COMPANY_TEST("  +
                "ID INT PRIMARY KEY     NOT NULL," +
                 "NAME           TEXT    NOT NULL," +
                 "AGE            INT     NOT NULL," +
                 "ADDRESS        CHAR(50)," +
                 "SALARY         REAL );"

        execute(sql)

        sql = "INSERT INTO COMPANY_TEST (ID,NAME,AGE,ADDRESS,SALARY) "  +
                 "VALUES (1, 'Mahmoud', 29, 'Jeddah', 20000.00 ); " +
                 "INSERT INTO COMPANY_TEST (ID,NAME,AGE,ADDRESS,SALARY) "  +
                 "VALUES (2, 'Ahmed', 27, 'Jeddah', 15000.00 ); "     +
                 "INSERT INTO COMPANY_TEST (ID,NAME,AGE,ADDRESS,SALARY)" +
                 "VALUES (3, 'Mohammed', 31, 'Egypt', 20000.00 );" +
                 "INSERT INTO COMPANY_TEST (ID,NAME,AGE,ADDRESS,SALARY)" +
                 "VALUES (4, 'Ibrahim', 24, 'Egypt ', 65000.00 );"

        execute(sql)

        ?  execute("select * from COMPANY_TEST")

        ? copy("*",50)

        close()
}
```

Output:

```
Test the PostgreSQL Class Methods
id
name
age
address
salary
1
Mahmoud
29
Jeddah
20000
2
Ahmed
27
Jeddah
15000
3
```

(continues on next page)

(continued from previous page)

```
Mohammed
31
Egypt
20000
4
Ibrahim
24
Egypt
65000

**************************************************
```

44.19 Security Class

Methods:

Method	Description/Output
md5(cString)	Calculate the MD5 hash.
sha1(cString)	Calculate the SHA1 hash.
sha256(cString)	Calculate the SHA256 hash.
sha512(cString)	Calculate the SHA512 hash.
sha384(cString)	Calculate the SHA384 hash.
sha224(cString)	Calculate the SHA224 hash.
encrypt(cString,cKey,cIV)	Encrypts the data using the Blowfish algorithm in CBC mode.
decrypt(cString,cKey,cIV)	Decrypt the encrypted data using the Blowfish algorithm in CBC mode.
encrypt_ex(cString,cKey,cIV,cCipher)	Encrypts the data using the given cipher algorithm.
decrypt_ex(cString,cKey,cIV,cCipher)	Decrypt the encrypted data using the given cipher algorithm.
supportedciphers()	List all supported cipher algorithms for encryption/decryption.
randbytes(nSize)	Generate a string of cryptographically secure pseudo-random bytes.

example:

```
Load "stdlib.ring"

oSecurity = new security
See "Test the security Class Methods" + nl
oSecurity {
        see md5("hello") + nl +
        sha1("hello") + nl + sha256("hello") + nl +
        sha512("hello") + nl + sha384("hello") + nl +
        sha256("hello") + nl
        list = 0:15  cKey=""   for x in list cKey += char(x) next
        list = 1:8   cIV = ""   for x in list cIV += char(x) next
        cCipher = encrypt("hello",cKey,cIV)
        see cCipher + nl + decrypt(cCipher,cKey,cIV) + nl
}
```

44.20 Internet Class

Methods:

- download(cURL)
- sendemail(cSMTPServer,cEmail,cPassword,cSender,cReceiver,cCC,cTitle,cContent)

example:

```
Load "stdlib.ring"

ointernet = new internet
See "Test the internet Class Methods" + nl
ointernet {
        see download("www.ring-lang.sf.net")
}
```

CHAPTER

FORTYFIVE

DESKTOP, WEBASSEMBLY AND MOBILE DEVELOPMENT USING RINGQT

In this chapter we will learn how to use the Qt framework classes in our Ring applications to create Desktop, WebAssembly and Mobile Applications.

45.1 The First GUI Application

In this example we will create an application to ask the user about his/her name. When the user type the name in the textbox then click on "Say Hello" button, the textbox value will be updated by adding "Hello " to the name.

```
load "guilib.ring"

oApp = new qApp {

        win1 = new qWidget() {

                setwindowtitle("Hello World")
                setGeometry(100,100,370,250)

                label1 = new qLabel(win1) {
                        settext("What is your name ?")
                        setGeometry(10,20,350,30)
                        setalignment(Qt_AlignHCenter)
                }

                btn1 = new qPushButton(win1) {
                        setGeometry(10,200,100,30)
                        settext("Say Hello")
                        setclickevent("pHello()")
                }

                btn2 = new qPushButton(win1) {
                        setGeometry(150,200,100,30)
                        settext("Close")
                        setclickevent("pClose()")
                }

                lineedit1 = new qLineEdit(win1) {
                        setGeometry(10,100,350,30)
```

(continues on next page)

(continued from previous page)

```
                }

                show()
        }

        exec()
}

Func pHello
        lineedit1.settext( "Hello " + lineedit1.text())

Func pClose
        oApp.quit()
```

Program Output:

At first we type the name in the textbox

Then we click on the say hello button

45.2 The Events Loop

Qt uses Event-Driven and the events loop get the control when we call the exec() method from the qApp class.

Once we call exec(), The events loop starts, and the Widgets starts responding to the different events (Mouse, Keyboard, Timers, etc).

You get the control back again when an event is fired and your callback function is called.

Once the execution of your callback function is done, the control go back again to the events loop.

Useful things to remember

(1) We can do most of the work using normal events (Events provided directly by each widget).

(2) We can add more events to any widget using the Events Filter.

(3) Using Timers we can easily get the control back and check for more things to do.

Also when our callback function is busy with doing time consuming operations, we can call the ProcessEvents() method from the qApp class to avoid stopping the GUI.

```
oApp.processevents()
```

We can avoid calling the exec() method, and create our main loop

It's not recommended to do that, It's just an option.

```
# Instead of calling the exec() method
        while true
                oApp.processevents()    # Respond to GUI Events
                # More Thing to do, We have the control!
                # .....
        end
```

45.3 Using Layout

The next example is just an upgrade to the previous application to use the vertical layout.

```
Load "guilib.ring"

MyApp = New qApp {

        win1 = new qWidget() {

                setwindowtitle("Hello World")
                setGeometry(100,100,400,130)
                label1 = new qLabel(win1) {
                        settext("What is your name ?")
                        setGeometry(10,20,350,30)
                        setalignment(Qt_AlignHCenter)
                }
                btn1 = new qpushbutton(win1) {
                        setGeometry(10,200,100,30)
                        settext("Say Hello")
                        setclickevent("pHello()")
                }
                btn2 = new qpushbutton(win1) {
                        setGeometry(150,200,100,30)
                        settext("Close")
                        setclickevent("pClose()")
                }
                lineedit1 = new qlineedit(win1) {
                        setGeometry(10,100,350,30)
                }
                layout1 = new qVBoxLayout() {
                        addwidget(label1)
                        addwidget(lineedit1)
                        addwidget(btn1)
                        addwidget(btn2)
                }
                win1.setlayout(layout1)
                show()
        }

        exec()

}

Func pHello
        lineedit1.settext( "Hello " + lineedit1.text())

Func pClose
        MyApp.quit()
```

The application during the runtime!

45.4 Using the QTextEdit Class

In this example we will use the QTextEdit Class

```
Load "guilib.ring"

New qApp {

        win1 = new qWidget() {

                setwindowtitle("QTextEdit Class")
                setGeometry(100,100,500,500)

                new qtextedit(win1) {
                        setGeometry(10,10,480,480)

                }

                show()
        }

        exec()
}
```

During the runtime we can paste rich text in the qtextedit widget

45.5 Using the QListWidget Class

In this example we will use the QListWidget Class

```
Load "guilib.ring"

New qApp {

        win1 = new qWidget() {

                setGeometry(100,100,400,400)

                list1 = new qlistwidget(win1) {
                        setGeometry(150,100,200,200)
                        alist = ["one","two","three","four","five"]
                        for x in alist additem(x) next
                        setcurrentrow(3,2)
```

(continues on next page)

(continued from previous page)

```
                        win1.setwindowtitle("Items Count : " + count() )
                }

                btn1 = new qpushbutton(win1) {
                        setGeometry(10,200,100,30)
                        settext("selected item")
                        setclickevent("pWork()")
                }

                btn2 = new qpushbutton(win1) {
                        setGeometry(10,240,100,30)
                        settext("Delete item")
                        setclickevent("pWork2()")
                }

                show()
        }

        exec()
}

func pWork
        btn1.settext(string(list1.currentrow()))

func pWork2
        list1 {
               takeitem(currentrow())
        }
```

The application during the runtime

Another Example:

```
Load "guilib.ring"

New qApp {

        win1 = new qWidget() {

                setGeometry(100,100,500,400)

                list1 = new qlistwidget(win1) {
                        setGeometry(150,100,200,200)
                        alist = ["one","two","three","four","five"]
                        for x in alist additem(x) next

                        setcurrentrow(3,2)
                        win1.setwindowtitle("Items Count : " + count() )
                }

                btn1 = new qpushbutton(win1) {
                        setGeometry(10,200,100,30)
                        settext("selected item")
                        setclickevent("pWork()")
                }
```

(continues on next page)

(continued from previous page)

```
                btn2 = new qpushbutton(win1) {
                        setGeometry(10,240,100,30)
                        settext("Delete item")
                        setclickevent("pWork2()")
                }

                show()
        }

        exec()
}

func pWork

        nbrOfItems = list1.count()
        curItemNbr = list1.currentrow()
        curValue   = list1.item(list1.currentrow()).text()

        win1.setwindowtitle( "After Select - NbrOfItems: " + nbrOfItems +
                " CurItemNbr: " + curItemNbr + " CurValue: " + curValue )

        btn1.settext( string(list1.currentrow() ) + " --- " +
                      list1.item(list1.currentrow()).text() )



func pWork2
        list1 {
                takeitem(currentrow())

                nbrOfItems = count()
                curItemNbr = currentrow()
                curValue   = item(currentrow()).text()

                win1.setwindowtitle("After Delete - NbrOfItems: " + nbrOfItems +
                        " CurItemNbr: " + curItemNbr +" CurValue: " + curValue )
        }
```

45.6 Using QTreeView and QFileSystemModel

In this example we will learn how to use the QTreeView widget to represent the File System

```
Load "guilib.ring"

New qApp {

        win1 = New qWidget() {

                setwindowtitle("Using QTreeView and QFileSystemModel")
                setGeometry(100,100,500,400)
```

(continues on next page)

(continued from previous page)

```
                New qtreeview(win1) {
                        setGeometry(00,00,500,400)
                        oDir = new QDir()
                        ofile = new QFileSystemModel()
                        ofile.setrootpath(oDir.currentpath())
                        setmodel(ofile)
                }

                show()
        }

        exec()
}
```

The application during the runtime

45.7 Using QTreeWidget and QTreeWidgetItem

In this example we will learn about using the QTreeWidget and QTreeWidgetItem classes

```
Load "guilib.ring"

New qApp {

        win1 = new qWidget() {

                setwindowtitle("TreeWidget")
                setGeometry(100,100,400,400)

                layout1 = new qvboxlayout()

                tree1 = new qtreewidget(win1) {
                        setGeometry(00,00,400,400)
                        setcolumncount(1)
                        myitem = new qtreewidgetitem()
                        myitem.settext(0,"The First Step")
                        addtoplevelitem(myitem)
                        for  x = 1 to 10
                                myitem2 = new qtreewidgetitem()
                                myitem2.settext(0,"hello"+x)
                                myitem.addchild(myitem2)
                                for  y = 1 to 10
                                        myitem3 = new qtreewidgetitem()
                                        myitem3.settext(0,"hello"+x)
                                        myitem2.addchild(myitem3)
                                next
                        next
                        setheaderlabel("Steps Tree")
                }

                layout1.addwidget(tree1)
                setlayout(layout1)

                show()
        }

        exec()
}
```

The application during the runtime

45.8 Using QComboBox Class

In this example we will learn about using the QComboBox class

```
Load "guilib.ring"

New qApp {

        win1 = new qWidget() {

                setwindowtitle("Using QComboBox")
                setGeometry(100,100,400,400)

                New QComboBox(win1) {
                        setGeometry(150,100,200,30)
                        alist = ["one","two","three","four","five"]
                        for x in aList additem(x,0) next
                }

                show()
        }

        exec()
```

(continues on next page)

(continued from previous page)

```
}
```

The application during the runtime

45.9 Creating Menubar

In this example we will learn about using the QMenuBar class

```
Load "guilib.ring"

MyApp = New qApp {

        win1 = new qWidget() {

                setwindowtitle("Using QMenubar")
                setGeometry(100,100,400,400)

                menu1 = new qmenubar(win1) {
                        sub1 = addmenu("File")
                        sub2 = addmenu("Edit")
                        sub3 = addmenu("Help")
                        sub1 {
```

(continues on next page)

(continued from previous page)

```
                oAction = new qAction(win1) {
                        settext("New")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        settext("Open")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        settext("Save")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        settext("Save As")
                }
                addaction(oAction)
                addseparator()
                oAction = new qaction(win1) {
                        settext("Exit")
                        setclickevent("myapp.quit()")
                }
                addaction(oAction)
        }
        sub2 {
                oAction = new qAction(win1) {
                        settext("Cut")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        settext("Copy")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        settext("Paste")
                }
                addaction(oAction)
                addseparator()
                oAction = new qAction(win1) {
                        settext("Select All")
                }
                addaction(oAction)
        }
        sub3 {
                oAction = new qAction(win1) {
                        settext("Reference")
                }
                addaction(oAction)
                sub4 = addmenu("Sub Menu")
                sub4 {
                        oAction = new qAction(win1) {
                                settext("Website")
                        }
```

(continues on next page)

(continued from previous page)

```
                                        addaction(oAction)
                                        oAction = new qAction(win1) {
                                                settext("Forum")
                                        }
                                        addaction(oAction)
                                        oAction = new qAction(win1) {
                                                settext("Blog")
                                        }
                                        addaction(oAction)
                                }
                                addseparator()
                                        oAction = new qAction(win1) {
                                                settext("About")
                                        }
                                        addaction(oAction)
                        }
                }
                show()
        }
        exec()
}
```

The application during the runtime

45.10 Context Menu

Example:

```
load "guilib.ring"

new qApp {
        win = new qwidget() {
                setwindowtitle("Context Menu")
                resize(400,400)
                myfilter = new qAllEvents(win) {
                        setContextmenuEvent("mymenu()")
                }
                installeventfilter(myfilter)
                show()
        }
        exec()
}


func mymenu

        new qMenu(win) {
                oAction = new qAction(win) {
                        settext("new")
                        setClickEvent("See :New")
                }
                addaction(oAction)
                oAction = new qAction(win) {
                        settext("open")
                        setClickEvent("See :Open")
                }
                addaction(oAction)
                oAction = new qAction(win) {
                        settext("save")
                        setClickEvent("See :Save")
                }
                addaction(oAction)
                oAction = new qAction(win) {
                        settext("close")
                        setClickEvent("See :Close")
                }
                addaction(oAction)
                oCursor  = new qCursor()
                exec(oCursor.pos())
        }
```

45.11 Creating Toolbar

In this example we will learn about using the QToolBar class

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setwindowtitle("Using QToolbar")
                setGeometry(100,100,600,400)

                abtns = [
                                new qpushbutton(win1) { settext("Add") } ,
                                new qpushbutton(win1) { settext("Edit") } ,
                                new qpushbutton(win1) { settext("Find") } ,
                                new qpushbutton(win1) { settext("Delete") } ,
                                new qpushbutton(win1) { settext("Exit")
                                                        setclickevent("win1.close()") }
                        ]

                tool1 = new qtoolbar(win1) {
                        for x in abtns addwidget(x) addseparator() next
                        setmovable(true)
                        setGeometry(0,0,500,30)
                        setFloatable(true)
                }

                show()
        }

        exec()
}
```

The application during the runtime

45.12 Creating StatusBar

In this example we will learn about using the QStatusBar class

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setwindowtitle("Using QStatusbar")
                setGeometry(100,100,400,400)

                status1 = new qstatusbar(win1) {
                        showmessage("Ready!",0)
                }

                setstatusbar(status1)
                show()
        }

        exec()
}
```

The application during the runtime

45.13 Using QDockWidget

In this example we will learn about using the QDockWidget class

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setwindowtitle("QDockWidget")
                setGeometry(100,100,400,400)

                label1 = new qlabel(win1) {
                        settext("Hello")
                        setGeometry(300,300,100,100)
                }

                label2 = new qlabel(win1) {
                        settext("How are you ?")
                        setGeometry(100,100,100,100)
```

(continues on next page)

(continued from previous page)

```
                }

                dock1 = new qdockwidget(win1,0) {
                        setwidget(label1)
                        SetAllowedAreas(1)
                }

                dock2 = new qdockwidget(win1,0) {
                        setwidget(label2)
                        SetAllowedAreas(2)
                }

                adddockwidget(Qt_LeftDockWidgetArea,dock1,Qt_Horizontal)
                adddockwidget(Qt_LeftDockWidgetArea,dock2,Qt_Vertical)

                show()
        }
        exec()
}
```

The application during the runtime

45.14 Using QTabWidget

In this example we will learn about using the QTabWidget class

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setwindowtitle("Using QTabWidget")
                setGeometry(100,100,400,400)

                page1 = new qwidget() {
                        new qpushbutton(page1) {
                                settext("The First Page")
                        }
                }

                page2 = new qwidget() {
                        new qpushbutton(page2) {
                                settext("The Second Page")
                        }
                }

                page3 = new qwidget() {
                        new qpushbutton(page3) {
                                settext("The Third Page")
                        }
                }

                tab1 = new qtabwidget(win1) {
                        inserttab(0,page1,"Page 1")
                        inserttab(1,page2,"Page 2")
                        inserttab(2,page3,"Page 3")
                        setGeometry(100,100,400,400)
                }

                status1 = new qstatusbar(win1) {
                        showmessage("Ready!",0)
                }

                setstatusbar(status1)
                showMaximized()
        }

        exec()
}
```

The application during the runtime

45.15 Using QTableWidget

In this example we will learn about using the QTableWidget class

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setGeometry(100,100,1100,370)
                setwindowtitle("Using QTableWidget")

                Table1 = new qTableWidget(win1) {

                        setrowcount(10) setcolumncount(10)
                        setGeometry(0,0,800,400)
                        setselectionbehavior(QAbstractItemView_SelectRows)

                        for x = 1 to 10
                                for y = 1 to 10
                                        item1 = new qtablewidgetitem("R"+X+"C"+Y)
                                        setitem(x-1,y-1,item1)
                                next
                        next

                }

                setcentralwidget(table1)
                show()
```

(continues on next page)

(continued from previous page)

```
        }

        exec()
}
```

The application during the runtime

45.16 Using QProgressBar

In this example we will learn about using the QProgressBar class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {

                setGeometry(100,100,600,150)
                setwindowtitle("Using QProgressBar")

                for x = 10 to 100 step 10
                        new qprogressbar(win1) {
                                setGeometry(100,x,350,30)
                                setvalue(x)
                        }
                next

                show()
        }
        exec()
}
```

The application during the runtime

45.17 Using QSpinBox

In this example we will learn about using the QSpinBox class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setGeometry(100,100,450,260)
                setwindowtitle("Using QSpinBox")
                        new qspinbox(win1) {
                                setGeometry(50,100,350,30)
                                setvalue(50)
                        }
                show()
        }
        exec()
}
```

The application during the runtime

45.18 Using QSlider

In this example we will learn about using the QSlider class

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setGeometry(100,100,500,400)
                setwindowtitle("Using QSlider")

                new qslider(win1) {
                        setGeometry(100,100,50,130)
                        settickinterval(50)
                }

                new qslider(win1) {
                        setGeometry(100,250,250,30)
                        settickinterval(50)
                        setorientation(Qt_Horizontal)
                }

                show()

        }

        exec()
}
```

The application during the runtime

45.19 Using QDateEdit

In this example we will learn about using the QDateEdit class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setwindowtitle("Using QDateEdit")
                setGeometry(100,100,250,100)
                new qdateedit(win1) {
                        setGeometry(20,40,220,30)
                }
                show()
        }
        exec()
}
```

The application during the runtime

45.20 Using QDial

In this example we will learn about using the QDial class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setGeometry(100,100,450,500)
                setwindowtitle("Using QDial")
                new qdial(win1) {
                        setGeometry(100,100,250,300)
                }
                show()
        }
        exec()
}
```

The application during the runtime

Another Example

```
Load "guilib.ring"

New qApp {
                 win1 = new qMainWindow()
                {
                        setGeometry(100,100,450,500)
                        setwindowtitle("Using QDial")
                        button1 = new QPushButton(win1){
                                setGeometry(100,350,100,30)
                                settext("Increment")
                                setClickEvent("pIncrement()")
                        }

                        button2 = new QPushButton(win1){
                                setGeometry(250,350,100,30)
                                settext("Decrement")
                                setClickEvent("pDecrement()")
```

(continues on next page)

(continued from previous page)

```
                        }
                        pdial = new qdial(win1) {
                                setGeometry(100,50,250,300)
                                setNotchesVisible(true)
                                setValue(50)
                                SetValueChangedEvent("pDialMove()")
                        }
                        lineedit1 = new qlineedit(win1) {
                                setGeometry(200,400,50,30)
                                setalignment(Qt_AlignHCenter)
                                settext(string(pdial.value()))
                                setreturnPressedEvent("pPress()")
                        }
                        show()
                }
        exec()
}

func pIncrement
        pdial{val=value()}
        pdial.setvalue(val+1)
        lineedit1{settext(string(val+1))}

func pDecrement
        pdial{val=value()}
        pdial.setvalue(val-1)
        lineedit1{settext(string(val-1))}

func pPress
        lineedit1{val=text()}
        pdial.setvalue(number(val))

func pDialMove
        lineedit1.settext(""+pdial.value())
```

45.21 Using QWebView

In this example we will learn about using the QWebView class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setwindowtitle("QWebView")
                myweb = new qwebview(win1) {
                        setGeometry(10,10,600,600)
                        loadpage(new qurl("http://google.com"))
                }
                setcentralwidget(myweb)
                showMaximized()
        }
        exec()
}
```

The application during the runtime

45.22 Using QCheckBox

In this example we will learn about using the QCheckBox class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setwindowtitle("Using QCheckBox")
                new qcheckbox(win1) {
                        setGeometry(100,100,100,30)
                        settext("New Customer!")
                }
                showMaximized()
        }
        exec()
}
```

The application during the runtime

Another Example:

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setGeometry(100,100,400,300)
                setwindowtitle("Using QCheckBox")

                ### 0-Unchecked  1-Checked

                CheckBox = new qcheckbox(win1) {
                        setGeometry(100,100,160,30)
                        settext("New Customer!")
                        setclickedEvent("HandleClickEvent()")
                }

                show()
        }
        exec()
}

Func HandleClickEvent

    if CheckBox.isChecked() = 1
        CheckBox.settext("New Customer. Check 1-ON")
    else
        CheckBox.settext("New Customer. Check 0-OFF")
    ok
```

45.23 Using QRadioButton and QButtonGroup

In this example we will learn about using the QRadioButton and QButtonGroup classes

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setwindowtitle("Using QRadioButton")

                new qradiobutton(win1) {
                        setGeometry(100,100,100,30)
                        settext("One")
                }
                new qradiobutton(win1) {
                        setGeometry(100,150,100,30)
                        settext("Two")
                }
                new qradiobutton(win1) {
                        setGeometry(100,200,100,30)
                        settext("Three")
                }


                group2  = new qbuttongroup(win1) {
                        btn4 = new qradiobutton(win1) {
                                setGeometry(200,150,100,30)
                                settext("Four")
                        }
                        btn5 = new qradiobutton(win1) {
                                setGeometry(200,200,100,30)
                                settext("Five")
                        }
                        addbutton(btn4,0)
                        addbutton(btn5,0)

                }

                showMaximized()

        }
        exec()
}
```

The application during the runtime

45.24 Adding Hyperlink to QLabel

In this example we will learn about creating Hyperlink using the QLabel class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setwindowtitle("QLabel - Hyperlink")
                new qlabel(win1) {
                        setGeometry(100,100,100,30)
                        setopenexternallinks(true)
                        settext('<a href="http://google.com">Google</a>')
                }
                showMaximized()
        }
        exec()
}
```

The application during the runtime

45.25 QVideoWidget and QMediaPlayer

In this example we will learn about using the QVideoWidget and QMediaPlayer classes to play a group of movies from different positions at the same time

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setwindowtitle("QVideoWidget")

                btn1 = new qpushbutton(win1)    {
                        setGeometry(0,0,100,30)
                        settext("play")
                        setclickevent("player.play() player2.play()
                                                player3.play() player4.play()")
                }

                videowidget = new qvideowidget(win1) {
                        setGeometry(50,50,600,300)
                        setstylesheet("background-color: black")
                }

                videowidget2 = new qvideowidget(win1) {
                        setGeometry(700,50,600,300)
                        setstylesheet("background-color: black")
                }


                videowidget3 = new qvideowidget(win1) {
                        setGeometry(50,370,600,300)
                        setstylesheet("background-color: black")
```

(continues on next page)

(continued from previous page)

```
                }

                videowidget4 = new qvideowidget(win1) {
                        setGeometry(700,370,600,300)
                        setstylesheet("background-color: black")
                }

                player = new qmediaplayer() {
                        setmedia(new qurl("1.mp4"))
                        setvideooutput(videowidget)
                        setposition(35*60*1000)

                }

                player2 = new qmediaplayer() {
                        setmedia(new qurl("2.mp4"))
                        setvideooutput(videowidget2)
                        setposition(23*60*1000)
                }

                player3 = new qmediaplayer() {
                        setmedia(new qurl("3.mp4"))
                        setvideooutput(videowidget3)
                        setposition(14.22*60*1000)
                }

                player4 = new qmediaplayer() {
                        setmedia(new qurl("4.avi"))
                        setvideooutput(videowidget4)
                        setposition(8*60*1000)
                }

                showfullscreen()

        }

        exec()

}
```

The application during the runtime

45.26 Using QFrame

In this example we will learn about using the QFrame class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setwindowtitle("Using QFrame")
                for x = 0 to 10
                        frame1 = new qframe(win1,0) {
                                setGeometry(100,20+50*x,400,30)
                                setframestyle(QFrame_Raised | QFrame_WinPanel)
                        }
                next
                showMaximized()
        }
        exec()
}
```

The application during the runtime

45.27 Display Image using QLabel

In this example we will learn about displaying an image using the QLabel widget

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setwindowtitle("QLabel - Display image")
                new qlabel(win1) {
                        image = new qpixmap("b:/mahmoud/photo/advice.jpg")
                        setpixmap(image)
```

(continues on next page)

(continued from previous page)

```
                        setGeometry(0,0,image.width(),image.height())
                }
                showMaximized()
        }
        exec()
}
```

The application during the runtime

45.28 Menubar and StyleSheet Example

In this example we will learn about creating menubar and setting the window stylesheet

```
Load "guilib.ring"

New qApp {

        win1 = new qMainWindow() {

                setwindowtitle("Menubar")

                menu1 = new qmenubar(win1) {
                        sub1 = addmenu("File")
                        sub1 {
                                oAction = new qAction(win1) {
                                        settext("New")
                                        setenabled(false)
                                }
                                addaction(oAction)
                                oAction = new qAction(win1) {
                                        settext("Open")
                                        setcheckable(true)
                                        setchecked(true)
                                        setstatustip("open new file")
                                }
                                addaction(oAction)
                                oAction = new qAction(win1) {
                                        settext("Save")
                                }
                                addaction(oAction)
                                oAction = new qAction(win1) {
                                        settext("Save As")
                                }
                                addaction(oAction)

                                addseparator()
                                oAction = new qaction(win1)
                                oAction.settext("Exit")
                                oAction.setclickevent("myapp.quit()")
                                addaction(oAction)
                        }

                }
                status1 = new qstatusbar(win1) {
                        showmessage("Ready!",0)
                }
                setmenubar(menu1)
                setmousetracking(true)
                setstatusbar(status1)
                setStyleSheet("color: black; selection-color: black;
                selection-background-color:white ;
                background: QLinearGradient(x1: 0, y1: 0, x2: 0, y2: 1,
```

(continues on next page)

(continued from previous page)

```
                stop: 0 #eef, stop: 1 #ccf);")
                showmaximized()
        }
        exec()
}
```

The application during the runtime

45.29 QLineEdit Events and QMessageBox

In this example we will learn about using QLineEdit Events and displaying a Messagebox

```
Load "guilib.ring"

MyApp = New qApp {

        win1 = new qWidget() {

                setwindowtitle("Welcome")
                setGeometry(100,100,400,300)


                label1 = new qLabel(win1) {
                        settext("What is your name ?")
                        setGeometry(10,20,350,30)
                        setalignment(Qt_AlignHCenter)
                }

                btn1 = new qpushbutton(win1) {
                        setGeometry(10,200,100,30)
                        settext("Say Hello")
                        setclickevent("pHello()")
                }
```

(continues on next page)

(continued from previous page)

```
                btn1 = new qpushbutton(win1) {
                        setGeometry(150,200,100,30)
                        settext("Close")
                        setclickevent("pClose()")
                }

                lineedit1 = new qlineedit(win1) {
                        setGeometry(10,100,350,30)
                        settextchangedevent("pChange()")
                        setreturnpressedevent("penter()")
                }

                show()
        }

        exec()
}

Func pHello
        lineedit1.settext( "Hello " + lineedit1.text())

Func pClose
        MyApp.quit()

Func pChange
        win1 { setwindowtitle( lineedit1.text() ) }

Func pEnter
        new qmessagebox(win1) {
                setwindowtitle("Thanks")
                settext("Hi " + lineedit1.text() )
                setstylesheet("background-color : white")
                show()
        }
```

The application during the runtime

Welcome
What is your name ?
Say Hello
Close

Somebody
What is your name ?
Thanks
Somebody
Hi Somebody
OK
Say Hello
Close

45.30 Other Widgets Events

Each Qt signal can be used in RingQt, just add Set before the signal name and add event after the signal name to get the method that can be used to determine the event code.

For example the QProgressBar class contains a signal named valueChanged() To use it just use the function setValueChangedEvent()

Example:

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {

                setwindowtitle("QProgressBar valueChanged Event")

                progress1 = new qprogressbar(win1) {
                        setGeometry(100,100,350,30)
                        setvalue(10)
                        setvaluechangedevent("pChange()")
                }

                new qpushbutton(win1) {
                        setGeometry(10,10,100,30)
                        settext("increase")
                        setclickevent("pIncrease()")
                }

                showMaximized()

        }

        exec()
}

func pIncrease
        progress1 { setvalue(value()+1)  }

func pchange
        win1.setwindowtitle("value : " + progress1.value() )
```

The application during the runtime

Another example for the stateChanged event of the QCheckBox class

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setwindowtitle("QCheckBox")
                new qcheckbox(win1) {
                        setGeometry(100,100,100,30)
                        settext("New Customer!")
                        setstatechangedevent("pchange()")
                }
                showMaximized()
        }
        exec()
}

Func pChange

        new qMessageBox(Win1) {
                setWindowTitle("Checkbox")
                settext("State Changed!")
                show()
        }
```

The application during the runtime

45.31 Using the QTimer Class

In this example we will learn about using the QTimer class

```
Load "guilib.ring"

new qApp {
        win1 = new qwidget() {
                setgeometry(100,100,200,70)
                setwindowtitle("Timer")
                label1 = new qlabel(win1) {
                        setgeometry(10,10,200,30)
                        settext(theTime())
                }
                new qtimer(win1) {
                        setinterval(1000)
                        settimeoutevent("pTime()")
                        start()
                }
                show()
        }
        exec()
}

func pTime
        label1.settext(thetime())

func theTime
        return "Time : " + Time()
```

The application during the runtime

45.32 Using QProgressBar and Timer

In this example we will learn about using the "animated" QProgressBar class and Timer

```
###-------------------------------------
### ProgressBar and Timer Example

Load "guilib.ring"

new qApp
{
    win1 = new qwidget()
    {
        setgeometry(100,100,400,100)
        setwindowtitle("Timer and ProgressBar")

        LabelMan = new qlabel(win1)
        {
            setgeometry(10,10,200,30)
            settext(theTime())          ### ==>> func
        }

        TimerMan = new qtimer(win1)
        {
            setinterval(1000)
            settimeoutevent("pTime()")  ### ==>> func
            start()
        }

        BarMan = new qprogressbar(win1)
        {
            setGeometry(100,50,300,10)  ###  Position X y, Length, Thickness
            setvalue(0)                 ###  Percent filled
        }

        show()
    }
    exec()
}

func pTime
        LabelMan.settext(theTime())     ### ==>> func

        Increment = 10
```

(continues on next page)

(continued from previous page)

```
        if BarMan.value() >= 100        ### ProgressBar start over.
            BarMan.setvalue(0)
        ok
        BarMan{ setvalue(value() + Increment) }

Func theTime
        return "Time : " + Time()
```

45.33 Display Scaled Image using QLabel

In this example we will learn about displaying and scaling an image so that it looks "animated" using the QLabel widget

```
Load "guilib.ring"

#-----------------------------------------------------
# REQUIRES: image = "C:\RING\bin\stock.jpg"

# imageStock: start dimensions for growing image

imageW = 200 ;  imageH = 200 ; GrowBy = 4

###-----------------------------------------------------
### Window and Box Size dimensions

WinWidth = 1280 ; WinHeight = 960
BoxWidth = WinWidth -80 ; BoxHeight = WinHeight -80

###-----------------------------------------------------

New qapp {

        win1 = new qwidget() {

                setgeometry(50,50, WinWidth,WinHeight)
                setwindowtitle("Animated Image - Display Image Scaled and Resized")

                imageStock = new qlabel(win1) {

                        image = new qpixmap("C:\RING\bin\stock.jpg")
                        AspectRatio = image.width() / image.height()
```

(continues on next page)

(continued from previous page)

```
                        imageW = 200
                        imageH = imageH / AspectRatio

                        ### Size-H, Size-V, Aspect, Transform
                        setpixmap(image.scaled(imageW , imageH ,0,0))

                        PosLeft = (BoxWidth  - imageW ) / 2
                        PosTop  = (BoxHeight - imageH ) / 2
                        setGeometry(PosLeft,PosTop,imageW,imageH)

                }

                TimerMan = new qtimer(win1) {
                        setinterval(100)            ### interval 100 millisecs.
                        settimeoutevent("pTime()")  ### ==>> func
                        start()
                }

                show()
        }
exec()
}


###-------------------------------------------------------
### Function TimerMan: calling interval 100 milliseconds

func pTime

        ### Stop Timer when image is size of Window area
        if imageW > BoxWidth
                TimerMan.stop()
                imageStock.clear()      ### Will clear the image
        ok

        ### Grow image
        imageW += GrowBy
        imageH  = imageW / AspectRatio

        ### Scaled Image: Size-H, Size-V, Aspect, Transform
        imageStock.setpixmap(image.scaled(imageW , imageH ,0,0))

        ### Center the image
        PosLeft = (WinWidth  - imageW ) / 2
        PosTop  = (WinHeight - imageH ) / 2
        imageStock.setGeometry(PosLeft,PosTop,imageW,imageH)
```

45.34 Using the QFileDialog Class

Example

```
Load "guilib.ring"

New qapp {
        win1 = new qwidget() {
                setwindowtitle("open file")
                setgeometry(100,100,400,400)
                new qpushbutton(win1) {
                        setgeometry(10,10,200,30)
                        settext("open file")
                        setclickevent("pOpen()")
                }
                show()
        }
        exec()
}

Func pOpen
        new qfiledialog(win1) {
                cName = getopenfilename(win1,"open file","c:\","source files(*.ring)")
                win1.setwindowtitle(cName)
        }
```

The application during the runtime

45.35 Drawing using QPainter

In this example we will learn about drawing using the QPainter class

```
Load "guilib.ring"
New qapp {
        win1 = new qwidget() {
                setwindowtitle("Drawing using QPainter")
                setgeometry(100,100,500,500)
                label1 = new qlabel(win1) {
                        setgeometry(10,10,400,400)
                        settext("")
                }
                new qpushbutton(win1) {
                        setgeometry(200,400,100,30)
                        settext("draw")
                        setclickevent("draw()")
                }

                show()
        }
        exec()
}

Func draw
                p1 = new qpicture()
                color = new qcolor() {
                        setrgb(0,0,255,255)
                }
                pen = new qpen() {
                        setcolor(color)
                        setwidth(10)
                }
                new qpainter() {
                        begin(p1)
                        setpen(pen)
                        drawline(500,150,950,450)
                        drawline(950,550,500,150)
                        endpaint()
                }
                label1 { setpicture(p1) show() }
```

The application during the runtime

45.36 Printing using QPrinter

In this example we will learn how to print to PDF file using QPrinter

```
Load "guilib.ring"
new qApp {
        win1 = new qwidget() {
                setwindowtitle("Printer")
                setgeometry(100,100,500,500)
                myweb = new qwebview(win1) {
                        setgeometry(100,100,1000,500)
                        loadpage(new qurl("http://google.com"))
                }
                new qpushbutton(win1) {
                        setGeometry(20,20,100,30)
                        settext("Print")
                        setclickevent("print()")
```

(continues on next page)

(continued from previous page)

```
                }
                showmaximized()
        }
        exec()
}

func print
        printer1 = new qPrinter(0) {
                setoutputformat(1)      # 1 = pdf
                setoutputfilename("test.pdf")
                painter = new qpainter() {
                        begin(printer1)
                        myfont = new qfont("Times",50,-1,0)
                        setfont(myfont)
                        drawtext(100,100,"test")
                        printer1.newpage()
                        drawtext(100,100,"test2")
                        endpaint()
                }
        }

        printer1 = new qPrinter(0) {
                setoutputformat(1)
                setoutputfilename("test2.pdf")
                myweb.print(printer1,' system("test2.pdf") ')
                myweb.show()
        }

        system ("test.pdf")
```

45.37 Using QPrintPreviewDialog

In this example we will learn how to use the QPrintPreviewDialog class.

Example:

```
load "guilib.ring"

new qApp {
        win1 = new qwidget() {
                setwindowtitle("Printer Preview Dialog")
                setgeometry(100,100,800,880)
                printer1 = new qPrinter(0)
                show()
                oPreview = new qPrintPreviewDialog(printer1) {
                        setParent(win1)
                        move(10,10)
                        setPaintrequestedevent("printPreview()")
                        exec()
                }
        }
```

(continues on next page)

(continued from previous page)

```
        exec()
}

func printPreview
        printer1  {
                painter = new qpainter() {
                        begin(printer1)
                        myfont = new qfont("Times",50,-1,0)
                        setfont(myfont)
                        drawtext(100,100,"Test - Page (1)")
                        printer1.newpage()
                        drawtext(100,100,"Test - Page (2)")
                        printer1.newpage()
                        myfont2 = new qfont("Times",14,-1,0)
                        setfont(myfont2)
                        for x = 1 to 30
                                drawtext(100,100+(20*x),"Number : " + x)
                        next
                        endpaint()
                }
        }
```

Screen Shot:

Printer Preview Dialog
56.5%
3 / 3
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
Number : 6
Number : 7
Number : 8
Number : 9
Number : 10
Number : 11
Number : 12
Number : 13
Number : 14
Number : 15
Number : 16
Number : 17
Number : 18
Number : 19
Number : 20
Number : 21
Number : 22
Number : 23
Number : 24
Number : 25
Number : 26
Number : 27
Number : 28
Number : 29
Number : 30

45.38 Creating More than one Window

The next example demonstrates how to create more than one window

```
Load "guilib.ring"
app1 = new qapp {
        win1 = new qwidget() {
                setwindowtitle("First")
                setgeometry(100,100,500,500)

                new qpushbutton(win1) {
                        setgeometry(100,100,100,30)
                        settext("close")
                        setclickevent("app1.quit()")
                }

                new qpushbutton(win1) {
                        setgeometry(250,100,100,30)
                        settext("Second")
                        setclickevent("second()")
                }

                showmaximized()
        }
        exec()
}

func second
        win2 = new qwidget() {
                setwindowtitle("Second")
                setgeometry(100,100,500,500)
                setwindowflags(Qt_dialog)
                show()
        }
```

The application during the runtime

45.39 Playing Sound

Example:

```
Load "guilib.ring"
new qapp {
        win1 = new qwidget() {
                setwindowtitle("play sound!") show()
        }
        new qmediaplayer()      {
                setmedia(new qurl("footstep.wav"))
                setvolume(50) play()
        }
        exec()
}
```

45.40 Using the QColorDialog Class

Example:

```
Load "guilib.ring"

oApp = new myapp { start() }

Class MyApp

        oColor  win1
```

(continues on next page)

(continued from previous page)

```
Func start

        myapp = new qapp

        win1 = new qMainWindow() {
                setwindowtitle("Color Dialog")
                setgeometry(100,100,400,400)
        }

        new qpushbutton(win1) {
                setgeometry(10,10,100,30)
                settext("Get Color")
                setclickevent("oApp.pColor()")
        }

        win1.show()
        myapp.exec()


Func pColor
        myobj = new qcolordialog()
        aColor = myobj.GetColor()
        r=acolor[1] g=acolor[2] b=acolor[3]
        win1.setstylesheet("background-color: rgb("+r+", " + g+ "," + b + ")")
```

The application during the runtime

45.41 Using qLCDNumber Class

In this example we will learn about using the qLCDNumber class

```
Load "guilib.ring"

New qApp
{
        win1 = new qWidget()
        {
                setwindowtitle("LCD Number")
                setgeometry(100,100,250,120)

                new qLCDNumber(win1)
                {
                        setgeometry(10,10,100,40)
                        display(100)

                }

                new qLCDNumber(win1)
                {
                        setgeometry(10,60,100,40)
                        display(80)

                }

                show()
        }

        exec()
}
```

The application during the runtime

45.42 Movable Label Example

```
Load "guilib.ring"

new qApp {

        win1 = new qWidget()
        {

                label1 = new qLabel(win1)
                {
                        setText("Welcome")
                        setgeometry(10,10,200,50)
                        setstylesheet("color: purple ; font-size: 30pt;")
                }

                new qTimer(win1)
                {
                        setInterVal(10)
                        setTimeOutEvent("pMove()")
                        start()
                }

                setWindowTitle("Movable Label")
                setgeometry(100,100,600,80)
                setStyleSheet("background-color: white;")
                show()

        }

        exec()
}

Func pMove
        label1
        {
                move(x()+1,y())
                if x() > 600
                        move(10,y())
                ok
        }
```

The application during the runtime

45.43 QMessagebox Example

In this section we will learn how to check the output of the Message box

```
Load "guilib.ring"

new qApp {
        win1 = new qWidget()
        {
                label1 = new qpushbutton(win1)
                {
                        setText("Test")
                        setgeometry(10,10,200,50)
                        setstylesheet("color: purple ; font-size: 30pt;")
                        setclickevent("pWork()")
                }
                setWindowTitle("Messagebox")
                setgeometry(100,100,600,80)
                setStyleSheet("background-color: white;")
                show()
        }
        exec()
}

func pWork
        new qmessagebox(win1)
        {
                setwindowtitle("messagebox title")
                settext("messagebox text")
                setInformativeText("Do you want to save your changes?")
                setstandardbuttons(QMessageBox_Yes | QMessageBox_No | QMessageBox_Close)
                result = exec()
                win1 {
                        if result = QMessageBox_Yes
                                setwindowtitle("Yes")
                        but result = QMessageBox_No
                                setwindowtitle("No")
                        but result = QMessageBox_Close
                                setwindowtitle("Close")
                        ok
                }
        }
```

The application during the runtime

45.44 Using QInputDialog Class

In the next example we will learn about using the QInputDialog class

```
Load "guilib.ring"

New QApp {

        Win1 = New QWidget () {

                SetGeometry(100,100,400,400)
                SetWindowTitle("Input Dialog")

                New QPushButton(win1)
                {

                        SetText ("Input Dialog")
                        SetGeometry(100,100,100,30)
                        SetClickEvent("pWork()")
                }

                Show()
        }

        exec()
}

Func pWork
        oInput = New QInputDialog(win1)
        {
                setwindowtitle("What is your name?")
                setgeometry(100,100,400,50)
```

(continues on next page)

(continued from previous page)

```
                setlabeltext("User Name")
                settextvalue("Mahmoud")
                lcheck = exec()
                if lCheck win1.setwindowtitle(oInput.textvalue()) ok
        }
```

The application during the runtime

45.45 Dialog Functions

We have the next functions

```
SetDialogIcon(cIconFile)
MsgInfo(cTitle,cMessage)
ConfirmMsg(cTitle,cMessage) --> lResult
InputBox(cTitle,cMessage) --> cValue
InputBoxInt(cTitle,cMessage) --> nValue
InputBoxNum(cTitle,cMessage) --> nValue
InputBoxPass(cTitle,cMessage) --> cValue
```

Example

```
load "guilib.ring"

new qApp
{
        SetDialogIcon("notepad.png")
        msginfo(:Ring,:Welcome)
        see confirmMsg(:Ring,"Are you sure?") + nl
        see InputBoxNum(:Ring,"Enter Number(double) :") + nl
        see InputBox(:Ring,"Enter Value :") + nl
        see InputBoxInt(:Ring,"Enter Number(int)") + nl
        see InputBoxPass(:Ring,"Enter Password") +nl
}
```

45.46 KeyPress and Mouse Move Events

In this example we will learn how to use the Events Filter to know about KeyPress and Mouse Move Events

```
Load "guilib.ring"

new qApp {

        win1 = new qWidget()
        {
                setWindowTitle("Test using Event Filter!")
                setGeometry(100,100,400,400)
                setmousetracking(true)
                myfilter = new qallevents(win1)
                myfilter.setKeyPressEvent("pWork()")
                myfilter.setMouseButtonPressevent("pClick()")
                myfilter.setmousemoveevent("pMove()")

                installeventfilter(myfilter)

                show()
        }

        exec()
}

func pWork
        win1.setwindowtitle('KeyPress! : ' + myfilter.getkeycode())

func pClick
        new qmessagebox(win1) {
                setgeometry(100,100,400,100)
                setwindowtitle("click event!")
                settext("x : " + myfilter.getx() +
                        " y : " + myfilter.gety() + " button : " +
                         myfilter.getbutton() )
                show()
        }
```

(continues on next page)

(continued from previous page)

```
func pMove
        win1.setwindowtitle("Mouse Move , X : " + myfilter.getx() +
                            " Y : " + myfilter.gety() )
```

The application during the runtime

45.47 Moving Objects using the Mouse

In the next example we will learn how to program movable objects where the user can move a label

```
Load "guilib.ring"

lPress = false
nX = 0
nY = 0

new qApp {

        win1 = new qWidget()
        {
```

(continues on next page)

(continued from previous page)

```
                setWindowTitle("Move this label!")
                setGeometry(100,100,400,400)
                setstylesheet("background-color:white;")

                Label1 = new qLabel(Win1){
                        setGeometry(100,100,200,50)
                        setText("Welcome")
                        setstylesheet("font-size: 30pt")
                        myfilter = new qallevents(label1)
                        myfilter.setEnterevent("pEnter()")
                        myfilter.setLeaveevent("pLeave()")
                        myfilter.setMouseButtonPressEvent("pPress()")
                        myfilter.setMouseButtonReleaseEvent("pRelease()")
                        myfilter.setMouseMoveEvent("pMove()")
                        installeventfilter(myfilter)
                }

                show()
        }

        exec()
}

Func pEnter
        Label1.setStyleSheet("background-color: purple; color:white;font-size: 30pt;")

Func pLeave
        Label1.setStyleSheet("background-color: white; color:black;font-size: 30pt;")

Func pPress
        lPress = True
        nX = myfilter.getglobalx()
        ny = myfilter.getglobaly()

Func pRelease
        lPress = False
        pEnter()

Func pMove
        nX2 = myfilter.getglobalx()
        ny2 = myfilter.getglobaly()
        ndiffx = nX2 - nX
        ndiffy = nY2 - nY
        if lPress
                Label1 {
                        move(x()+ndiffx,y()+ndiffy)
                        setStyleSheet("background-color: Green;
                                 color:white;font-size: 30pt;")
                        nX = nX2
                        ny = nY2
                }
```

(continues on next page)

(continued from previous page)

```
        ok
```

The application during the runtime

Move this label!
Welcome

45.48 Inheritance from GUI Classes

Example :

```
Load "guilib.ring"

New MyWindow()

new qApp { exec() }

class mywindow from qwidget
        Func init
                super.init()
                setwindowtitle("First Window")
                setgeometry(100,100,400,400)
                setstylesheet("background-color: purple;")
                settooltip("my first window!")
                show()
```

The application during the runtime

45.49 Using QDesktopWidget Class

In the next example we will learn about using the QDesktopWidget class

```
Load "guilib.ring"

New qApp {
        win1 = New qWidget()
        {
                resize(400,400)
                btn1 = new qPushbutton(win1)
                {
                        setText("Center")
                        move(100,100)
                        resize(100,30)
                        setClickEvent("pCenter()")
                }

                Show()
        }

        exec()
}
```

(continues on next page)

(continued from previous page)

```
Func pCenter
        oDesktop  = new qDesktopWidget()
        win1.move((oDesktop.width()-win1.width()) /2 , (oDesktop.Height()-win1.Height())/
↪2 )
        win1.show()
```

The application during the runtime

45.50 Rotate Text

The next example rotate text using a Timer.

```
Load "guilib.ring"

nAngle  = 0

New qapp {
        win1 = new qwidget() {
                setwindowtitle("Rotate Text")
                resize(800,600)
                label1 = new qlabel(win1) {
```

(continues on next page)

(continued from previous page)

```
                        settext("")
                        myfilter = new qallevents(win1)
                        myfilter.setMouseButtonPressevent("pClick()")
                        installeventfilter(myfilter)
                }
                new qtimer(win1) {
                        setinterval(50)
                        settimeoutevent("pTime()")
                        start()
                }
                pDraw()
                L1 = new qVBoxLayout() { AddWidget(Label1) } SetLayout(L1)
                showMaximized()
        }
        exec()
}

Func pDraw
        p1 = new qpicture()
        color = new qcolor() {
                setrgb(0,0,255,255)
        }
        pen = new qpen() {
                setcolor(color)
                setwidth(50)
        }
        painter = new qpainter() {
                begin(p1)
                        setpen(pen)
                        myfont = font()
                        myfont.setpointsize(50)
                        setfont(myfont)
                        rotate(nAngle)
                        drawtext(350,0*nAngle,"welcome")
                        drawtext(0,0*nAngle,"welcome")
                endpaint()
                  }
        label1 {
                setpicture(p1)
                show()
        }

Func pClick
        win1 { setwindowtitle("Click Event") }

Func pTime
        nAngle++
        if nAngle = 90
                nAngle = 10
        ok
        pDraw()
```

The application during the runtime

45.51 Change Focus

The next example change the focus using the ENTER key.

```
load "guilib.ring"

new qApp {
        win = new qWidget() {
                resize(600,600)
                SetWindowTitle("Change Focus")
                text1 = new qLineEdit(win)
                text2 = new qLineEdit(win)
                text3 = new qLineEdit(win)
                text4 = new qLineEdit(win)
                layout1 = new qVBoxLayout() {
                        AddWidget(text1)
                        AddWidget(text2)
                        AddWidget(text3)
                        AddWidget(text4)
```

(continues on next page)

(continued from previous page)

```
                }
                setLayout(Layout1)
                aList = [text1,text2,text3,text4]
                oFilter = new qallevents(win)
                oFilter.setKeyPressEvent("pWork()")
                installeventfilter(oFilter)
                show()
        }
        exec()
}

func pWork
        nCode =  oFilter.getkeycode()
        if nCode = 16777220     # ENTER Key
                for x=1 to len(aList)
                        if aList[x].HasFocus()
                                t = x+1
                                if t > len(aList) t=1 ok
                                aList[t].SetFocus(0)
                                exit
                        ok
                next
        ok
```

45.52 Regular Expressions

The next example uses the Regular Expressions classes.

```
load "guilib.ring"

new qApp
{
        see "Using Regular Expressions" + nl

        exp = new qregularexpression() {
                setPattern("\d\d \w+")
                see pattern() + nl
                match = match("33 one",0,0,0)
                see match.hasmatch() + nl
                match = match("3 one",0,0,0)
                see match.hasmatch() + nl
                match = match("welcome 11 one",0,0,0)
                see match.hasmatch() + nl
                matched = match.captured(0)
                see matched + nl
        }
        exp = new qregularexpression() {
                setPattern("^(\d\d)/(\d\d)/(\d\d\d\d)$")
                see pattern() + nl
                match = match("08/12/1985",0,0,0)
```

(continues on next page)

(continued from previous page)

```
                see match.hasmatch() + nl
                day = match.captured(1)
                month = match.captured(2)
                year = match.captured(3)
                see day + nl + month + nl + year + nl
                see  "(" + match.capturedStart(1) + "," + match.capturedEnd(1)+ ")" + nl
                see  "(" + match.capturedStart(2) + "," + match.capturedEnd(2)+ ")" + nl
                see  "(" + match.capturedStart(3) + "," + match.capturedEnd(3)+ ")" + nl
        }

}
```

Output

```
Using Regular Expressions
\d\d \w+
1
0
1
11 one
^(\d\d)/(\d\d)/(\d\d\d\d)$
1
08
12
1985
(0,2)
(3,5)
(6,10)
```

45.53 Simple Client and Server Example

In this section we will learn about creating simple Client and Server Application

```
Load "guilib.ring"

new qApp {
        oClient = new Client { client() }
        oServer = new Server { server() }
        exec()
}

Class Client

        win1 lineedit1  cOutput=""
        oTcpSocket

        func client

                win1 = new qwidget()
```

(continues on next page)

(continued from previous page)

```
                new qpushbutton(win1) {
                        setgeometry(50,50,100,30)
                        settext("connect")
                        setclickevent("oClient.Connect()")
                }

                lineedit1 = new qtextedit(win1) {
                        setGeometry(150,50,200,300)
                }

                win1 {
                        setwindowtitle("client")
                        setgeometry(10,100,400,400)
                        show()
                }

        func connect
                cOutput = "Connect to host 127.0.0.1 port 9999" + nl
                lineedit1.settext(cOutput)
                oTcpSocket = new qTcpSocket(win1) {
                        setconnectedevent("oClient.pConnected()")
                        setreadyreadevent("oClient.pRead()")
                        connecttohost("127.0.0.1",9999,3,0)
                        waitforconnected(5000)
                }

        func pConnected

                cOutput += "Connected!" + nl
                lineedit1.settext(cOutput)

        func pRead

                cOutput += "Ready Read!" + nl
                lineedit1.settext(cOutput)
                cOutput += oTcpSocket.readall().data() + nl
                lineedit1.settext(cOutput)

Class Server

        win1 lineedit1
        oTcpServer oTcpClient
        cOutput = ""

        func server

                win1 = new qwidget()

                lineedit1 = new qtextedit(win1) {
                        setGeometry(150,50,200,300)
                }
```

(continues on next page)

(continued from previous page)

```
                win1 {
                        setwindowtitle("Server")
                        setgeometry(450,100,400,400)
                        show()
                }

                oTcpServer = new qTcpServer(win1) {
                        setNewConnectionEvent("oServer.pNewConnection()")
                        oHostAddress = new qHostAddress()
                        oHostAddress.SetAddress("127.0.0.1")
                        listen(oHostAddress,9999)
                }
                cOutput = "Server Started" + nl +
                           "listen to port 9999" + nl

                lineedit1.settext(cOutput)

        Func pNewConnection

                oTcpClient = oTcpServer.nextPendingConnection()
                cOutput += "Accept Connection" + nl
                lineedit1.settext(cOutput)
                oTcpClient {
                        cStr ="Hello from server to client!"+char(13)+char(10)
                        write(cStr,len(cStr))
                        flush()
                        waitforbyteswritten(300000)
                        close()
                }
```

The application during the runtime

45.54 Dynamic Objects

We may create objects in the runtime and add them to windows.

Example:

```
load "guilib.ring"

oFormDesigner = new FormDesigner { start("oFormDesigner") }

Class FormDesigner

        winToolBox  winForm

        aObjects = []

        func start cObjectName

                oApp = new qApp

                winToolBox = new qWidget()
                winToolBox.setWindowTitle("ToolBox")
                winToolBox.move(10,10)
                winToolBox.resize(300,600)

                btn = new qPushButton(winToolBox)
                btn.resize(300,30)
                btn.setText("Create Button")
                btn.setClickEvent(cObjectName+".pCreateButton()")
                btn.show()

                winToolBox.show()

                winForm = new qWidget() {
                        move(400,50)
                        setWindowTitle("Form Designer")
                        resize(600,600)
                        show()
                }

                oApp.exec()


        func pCreateButton

                nCount = len(aObjects)

                aObjects + new MyButton(winForm)
                {
                        nIndex = nCount + 1
                        setText("Button"+ nIndex)
                        Move(30*nIndex,30*nIndex)
                        resize(100,30)
```

(continues on next page)

(continued from previous page)

```
                        show()
                }


Class MyButton from qPushButton
        nIndex = 0
```

45.55 Weight History Application

The next sample help in recording (Date, Time and Weight).

```
Load "guilib.ring"

MyApp = new qApp
{
  $ApplicationObject = "oApp"   # To be used when calling events
  oApp = new App
  exec()
  oApp.CloseDatabase()
}

class App

  cDir = currentdir() + "/"
  oCon
  aIDs = []

  win1 = new qWidget()
  {
        setWindowTitle("Weight History")
        resize(600,600)
        layoutButtons = new qhboxlayout()
        {
          label1 = new qLabel(win1) { setText("Weight") }
          text1 = new qlineedit(win1)
          btnAdd = new qpushbutton(win1) {
                  setText("Add")
                  setClickEvent($ApplicationObject+".AddWeight()")
          }
          btnDelete = new qpushbutton(win1) {
                  setText("Delete")
                  setClickEvent($ApplicationObject+".Deleteweight()")
          }
          addwidget(label1)
          addwidget(text1)
          addwidget(btnAdd)
          addwidget(btnDelete)
        }
        layoutData  = new qhboxlayout()
        {
```

(continues on next page)

(continued from previous page)

```
        Table1 = new qTableWidget(win1) {
              setrowcount(0)
              setcolumncount(3)
              setselectionbehavior(QAbstractItemView_SelectRows)
              setHorizontalHeaderItem(0, new QTableWidgetItem("Date"))
              setHorizontalHeaderItem(1, new QTableWidgetItem("Time"))
              setHorizontalHeaderItem(2, new QTableWidgetItem("Weight"))
              setitemChangedEvent($ApplicationObject+".ItemChanged()")
                                  setAlternatingRowColors(true)
                                  horizontalHeader().setStyleSheet("color: blue")
                                  verticalHeader().setStyleSheet("color: red")
        }
        addWidget(Table1)
      }
      layoutClose = new qhboxlayout()
      {
        btnclose = new qpushbutton(win1) {
          setText("Close")
          setClickEvent("MyApp.Quit()")
        }
        addwidget(btnClose)
      }
      layoutMain = new qvboxlayout()
      {
        addlayout(layoutButtons)
        addLayout(LayoutData)
        addLayout(layoutClose)
      }
      setlayout(layoutMain)
      self.OpenDatabase()
      self.ShowRecords()
      show()
}

Func OpenDatabase
      lCreate = False
      if not fexists(cDir + "weighthistory.db")
        lCreate = True
      ok
      new QSqlDatabase() {
        this.oCon = addDatabase("QSQLITE") {
              setDatabaseName("weighthistory.db")
              Open()
        }
      }
      if lCreate
        new QSqlQuery( ) {
              exec("create table weighthistory (id integer primary key,"+
                   " f_date varchar(10),"+
                   " f_time varchar(8), f_weight varchar(8) );")
              delete()
        }
```

(continues on next page)

(continued from previous page)

```
      ok


Func CloseDatabase
      oCon.Close()

Func AddWeight
      cWeight = text1.text()
      AddRecord(cWeight)

Func DeleteWeight
      Table1 {
         nRow = CurrentRow()
        if nRow >= 0
              nID = this.aIDs[nROW+1]
              new QSqlQuery( ) {
                exec("delete from weighthistory where id = " + nID )
              }
              Del(this.aIDs,nRow+1)
              removerow(nRow)
              selectrow(nRow)
        ok
      }


Func AddRecord cWeight
      new QSqlQuery( ) {
        cStr = "insert into weighthistory (f_date,f_time,f_weight) values"+
        " ('%f1','%f2','%f3')"
        cDate = Date()
        cTime = Time()
        cStr = substr(cStr,"%f1",cDate)
        cStr = substr(cStr,"%f2",cTime)
        cStr = substr(cStr,"%f3",cWeight)
        exec(cStr)
        delete()
      }
      ShowRecords()
      Table1.selectrow(table1.rowcount()-1)


Func ShowRecords
      table1.setitemChangedEvent("")
      aIDs = []
      query = new QSqlQuery() {
        exec("select * from weighthistory")
        nRows = 0
        this.Table1.setrowcount(0)
        while movenext()
              this.table1 {
                insertRow(nRows)
                this.aIDs + query.value(0).tostring()
```

(continues on next page)

(continued from previous page)

```
                for x = 1 to 3
                      cStr = query.value(x).tostring()
                      item = new qTableWidgetItem(cStr)
                      setItem(nRows,x-1,item)
                next
              }
              nRows++
        end
        delete()
      }
      table1.setitemChangedEvent($ApplicationObject+".ItemChanged()")

Func ItemChanged
      nRow =  table1.currentrow()
      if nRow >= 0
        myitem = Table1.item(table1.currentrow(),0)
        cDate = myitem.text()
        myitem = Table1.item(table1.currentrow(),1)
        cTime = myitem.text()
        myitem = Table1.item(table1.currentrow(),2)
        cWeight = myitem.text()
        new QSqlQuery( ) {
              cStr = "update weighthistory set f_date ='%f1' , f_time = '%f2' , "+
              "f_weight ='%f3' where id = " +  this.aIDs[nROW+1]
              cStr = substr(cStr,"%f1",cDate)
              cStr = substr(cStr,"%f2",cTime)
              cStr = substr(cStr,"%f3",cWeight)
              exec(cStr)
              delete()
        }
      ok
```

The next screen shot for the application during the runtime

45.56 Notepad Application

In the next example we will see simple Notepad developed using the RingQt

```
Load "guilib.ring"

cActiveFileName = ""
aTextColor = [0,0,0]
aBackColor = [255,255,255]
cFont = "MS Shell Dlg 2,14,-1,5,50,0,0,0,0,0"
cWebsite = "http://www.google.com"
```

(continues on next page)

(continued from previous page)

```
oSearch = NULL
oSearchValue = NULL
oSearchCase = NULL
oSearchFilter = NULL
oReplaceValue = NULL

lAskToSave = false

MyApp = New qApp {
        win1 = new qMainWindow() {

                setwindowtitle("Ring Notepad")
                setGeometry(100,100,400,400)
                aBtns = [
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/new.png")
                                        setclickevent("pNew()")
                                        settooltip("New File")
                                } ,
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/open.png")
                                        setclickevent("pOpen()")
                                        settooltip("Open File")
                                } ,
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/save.png")
                                        setclickevent("pSave()")
                                        settooltip("Save")
                                 } ,
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/saveas.png")
                                        setclickevent("pSaveAs()")
                                        settooltip("Save As")
                                 } ,
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/cut.png")
                                        setclickevent("pCut()")
                                        settooltip("Cut")
                                 } ,
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/copy.png")
                                        setclickevent("pCopy()")
                                        settooltip("Copy")
                                } ,
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/paste.png")
                                        setclickevent("pPaste()")
                                        settooltip("Paste")
                                } ,
                                new qpushbutton(win1) {
                                        setbtnimage(self,"image/font.png")
```

(continues on next page)

(continued from previous page)

```
                        setclickevent("pFont()")
                        settooltip("Font")
                } ,
                new qpushbutton(win1) {
                        setbtnimage(self,"image/colors.jpg")
                        setclickevent("pColor()")
                        settooltip("Text Color")
                } ,
                new qpushbutton(win1) {
                        setbtnimage(self,"image/search.png")
                        setclickevent("pFind()")
                        settooltip("Find and Replace")
                } ,
                new qpushbutton(win1) {
                        setbtnimage(self,"image/print.png")
                        setclickevent("pPrint()")
                        settooltip("Print")
                } ,
                new qpushbutton(win1) {
                        setbtnimage(self,"image/debug.png")
                        setclickevent("pDebug()")
                        settooltip("Debug (Run then wait!)")
                } ,
                new qpushbutton(win1) {
                        setbtnimage(self,"image/run.png")
                        setclickevent("pRun()")
                        settooltip("Run the program")
                } ,
                new qpushbutton(win1) {
                        setbtnimage(self,"image/close.png")
                        setclickevent("pQuit()")
                        settooltip("Quit")
                }
        ]

tool1 = addtoolbar("files")  {
        for x in aBtns addwidget(x) addseparator() next
}

menu1 = new qmenubar(win1) {
        sub1 = addmenu("File")
        sub2 = addmenu("Edit")
        sub3 = addmenu("View")
        sub4 = addmenu("Help")
        sub1 {
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+n"))
                        setbtnimage(self,"image/new.png")
                        settext("New")
                        setclickevent("pNew()")
                }
                addaction(oAction)
```

(continues on next page)

(continued from previous page)

```
oAction = new qAction(win1) {
        setShortcut(new QKeySequence("Ctrl+o"))
        setbtnimage(self,"image/open.png")
        settext("Open")
        setclickevent("pOpen()")
}
addaction(oAction)
addseparator()
oAction = new qAction(win1) {
        setShortcut(new QKeySequence("Ctrl+s"))
        setbtnimage(self,"image/save.png")
        settext("Save")
        setclickevent("pSave()")
}
addaction(oAction)
addseparator()
oAction = new qAction(win1) {
        setShortcut(new QKeySequence("Ctrl+e"))
        setbtnimage(self,"image/saveas.png")
        settext("Save As")
        setclickevent("pSaveAs()")
}
addaction(oAction)
addseparator()
oAction = new qAction(win1) {
        setShortcut(new QKeySequence("Ctrl+p"))
        setbtnimage(self,"image/print.png")
        settext("Print to PDF")
        setclickevent("pPrint()")
}
addaction(oAction)
addseparator()
oAction = new qAction(win1) {
        setShortcut(new QKeySequence("Ctrl+d"))
        setbtnimage(self,"image/debug.png")
        settext("Debug (Run then wait!)")
        setclickevent("pDebug()")
}
addaction(oAction)
addseparator()
oAction = new qAction(win1) {
        setShortcut(new QKeySequence("Ctrl+r"))
        setbtnimage(self,"image/run.png")
        settext("Run")
        setclickevent("pRun()")
}
addaction(oAction)
addseparator()
oAction = new qAction(win1) {
        setShortcut(new QKeySequence("Ctrl+F5"))
        setbtnimage(self,"image/run.png")
        settext("Run GUI Application (No Console)")
```

(continues on next page)

(continued from previous page)

```
                        setclickevent("pRunNoConsole()")
                }
                addaction(oAction)
                addseparator()
                oAction = new qaction(win1) {
                        setShortcut(new QKeySequence("Ctrl+q"))
                        setbtnimage(self,"image/close.png")
                        settext("Exit")
                        setstatustip("Exit")
                        setclickevent("pQuit()")
                }
                addaction(oAction)
        }
        sub2 {
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+x"))
                        setbtnimage(self,"image/cut.png")
                        settext("Cut")
                        setclickevent("pCut()")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+c"))
                        setbtnimage(self,"image/copy.png")
                        settext("Copy")
                        setclickevent("pCopy()")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+v"))
                        setbtnimage(self,"image/paste.png")
                        settext("Paste")
                        setclickevent("pPaste()")
                }
                addaction(oAction)
                addseparator()
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+i"))
                        setbtnimage(self,"image/font.png")
                        settext("Font")
                        setclickevent("pFont()")
                }
                addseparator()
                addaction(oAction)
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+t"))
                        setbtnimage(self,"image/colors.jpg")
                        settext("Text Color")
                        setclickevent("pColor()")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
```

(continues on next page)

(continued from previous page)

```
                        setShortcut(new QKeySequence("Ctrl+b"))
                        setbtnimage(self,"image/colors.jpg")
                        settext("Back Color")
                        setclickevent("pColor2()")
                }
                addaction(oAction)
                addseparator()
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+g"))
                        settext("Go to line")
                        setclickevent("pGoto()")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+f"))
                        setbtnimage(self,"image/search.png")
                        settext("Find and Replace")
                        setclickevent("pFind()")
                }
                addaction(oAction)
        }
        sub3 {
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+p"))
                        setbtnimage(self,"image/project.png")
                        settext("Project Files")
                        setclickevent("pProject()")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+u"))
                        setbtnimage(self,"image/source.png")
                        setclickevent("pSourceCode()")
                        settext("Source Code")
                }
                addaction(oAction)
                oAction = new qAction(win1) {
                        setShortcut(new QKeySequence("Ctrl+w"))
                        setbtnimage(self,"image/richtext.png")
                        setclickevent("pWebBrowser()")
                        settext("Web Browser")
                }
                addaction(oAction)
        }
        sub4 {
                sub5 = addmenu("Development Tools")
                sub5 {

                        oAction = new qAction(win1) {
                                settext("Programming Language")
                                setclickevent("pLang()")
                        }
```

(continues on next page)

(continued from previous page)

```
                                addaction(oAction)
                                oAction = new qAction(win1) {
                                        settext("GUI Library")
                                        setclickevent("pGUI()")
                                }
                                addaction(oAction)
                        }
                        addseparator()
                                oAction = new qAction(win1) {
                                        settext("About")
                                        setclickevent("pAbout()")
                                }
                                addaction(oAction)
                }
        }

        setmenubar(menu1)

        status1 = new qstatusbar(win1) {
                showmessage("Ready!",0)
        }

        setstatusbar(status1)

        tree1 = new qtreeview(win1) {
                setclickedevent("pChangeFile()")
                setGeometry(00,00,200,400)
                oDir = new QDir()
                ofile = new QFileSystemModel() {
                        setrootpath(oDir.currentpath())
                        myfiles = new qstringlist()
                        myfiles.append("*.ring")
                        myfiles.append("*.rh")
                        setnamefilters(myfiles)
                        setNameFilterDisables(false)
                }
                setmodel(ofile)
                myindex = ofile.index(oDir.currentpath(),0)
                for x = 1 to ofile.columncount()
                        hidecolumn(x)
                next
                setcurrentindex(myindex)
                setexpanded(myindex,true)
                header().hide()
        }

        oDock1 = new qdockwidget(win1,0) {
                setGeometry(00,00,200,200)
                setwindowtitle("Project Files")
                setwidget(tree1)
        }
```

(continues on next page)

(continued from previous page)

```
textedit1 = new qtextedit(win1) {
        setCursorPositionChangedevent("pCursorPositionChanged()")
        setLineWrapMode(QTextEdit_NoWrap)
        setAcceptRichText(false)
        setTextChangedEvent("lAskToSave = true")
}


oDock2 = new qdockwidget(win1,0) {
        setwidget(textedit1)
        setwindowtitle("Source Code")
}

oWebBrowser = new qWidget() {
        setWindowFlags(Qt_SubWindow)
        oWBLabel = new qLabel(win1) {
                setText("Website: ")
        }
        oWBText = new qLineEdit(win1) {
                setText(cWebSite)
                setReturnPressedEvent("pWebGo()")
        }
        oWBGo = new qPushButton(win1) {
                setText("Go")
                setClickEvent("pWebGo()")
        }
        oWBBack = new qPushButton(win1) {
                setText("Back")
                setClickEvent("pWebBack()")
        }
        oWBLayout1 = new qHBoxLayout() {
                addWidget(oWBLabel)
                addWidget(oWBText)
                addWidget(oWBGo)
                addWidget(oWBBack)
        }
        oWebView = new qWebView(win1) {
                loadpage(new qurl(cWebSite))
        }
        oWBlayout2 = new qVBoxLayout() {
                addLayout(oWBLayout1)
                addWidget(oWebView)
        }
        setLayout(oWBLayout2)
}

oDock3 = new qdockwidget(win1,0) {
        setwidget(oWebBrowser)
        setwindowtitle("Web Browser")
        setFeatures(QDockWidget_DocWidgetClosable)
}
```

(continues on next page)

(continued from previous page)

```
                adddockwidget(1,oDock1,1)
                adddockwidget(2,oDock2,2)
                adddockwidget(2,oDock3,1)

                setwinicon(self,"image/notepad.png")

                showmaximized()
        }
        RestoreSettings()
        exec()
}

func pWebGo
        cWebsite = oWBText.text()
        oWebView.LoadPage( new qurl( cWebSite ) )

func pWebBack
        oWebView.Back()

func pProject
        oDock1.Show()

func pSourceCode
        oDock2.Show()

func pWebBrowser
        oDock3.Show()

func pChangeFile
        myitem = tree1.currentindex()
        if ofile.isdir(myitem)
                return
        ok
        cActiveFileName = ofile.filepath(myitem)
        textedit1.settext(read(cActiveFileName))
        textedit1.setfocus(0)
        pCursorPositionChanged()
        pSetActiveFileName()

func pSetActiveFileName
        oDock2.setWindowTitle("Source Code : " + cActiveFileName)

func pCursorPositionChanged
        status1.showmessage(" Line : "+(textedit1.textcursor().blocknumber()+1)+
                        " Column : " +(textedit1.textcursor().columnnumber()+1) +
                        " Total Lines : " + textedit1.document().linecount() ,0)

func pGoto
        oInput = New QInputDialog(win1)
        {
                setwindowtitle("Enter the line number?")
                setgeometry(100,100,400,50)
```

(continues on next page)

(continued from previous page)

```
                setlabeltext("Line")
                settextvalue("1")
                exec()
                nLine = 0 + oInput.textvalue()
                oBlock = textedit1.document().findBlockByLineNumber(nLine-1)
                oCursor = textedit1.textcursor()
                oCursor.setposition(oBlock.position(),0)
                textedit1.settextcursor(oCursor)
        }

func pFind
        if isobject(oSearch)
                oSearch.activatewindow()
                return
        ok
        oSearch = new qWidget()
        {
                new qLabel(oSearch)
                {
                        setText("Find What : ")
                        setgeometry(10,10,50,30)
                }
                oSearchValue = new qlineedit(oSearch)
                {
                        setgeometry(80,10,460,30)
                        setReturnPressedEvent("pFindValue()")
                }
                new qLabel(oSearch)
                {
                        setText("Replace with ")
                        setgeometry(10,45,80,30)
                }
                oReplaceValue = new qlineedit(oSearch)
                {
                        setgeometry(80,45,460,30)
                }
                oSearchCase = new qCheckbox(oSearch)
                {
                        setText("Case Sensitive")
                        setgeometry(80,85,100,30)
                }
                new qPushButton(oSearch)
                {
                        setText("Find/Find Next")
                        setgeometry(80,120,100,30)
                        setclickevent("pFindValue()")
                }
                new qPushButton(oSearch)
                {
                        setText("Replace")
                        setgeometry(200,120,100,30)
                        setclickevent("pReplace()")
```

(continues on next page)

(continued from previous page)

```
                }
                new qPushButton(oSearch)
                {
                        setText("Replace All")
                        setgeometry(320,120,100,30)
                        setclickevent("pReplaceAll()")
                }
                new qPushButton(oSearch)
                {
                        setText("Close")
                        setgeometry(440,120,100,30)
                        setclickevent("pSearchClose()")
                }

                setwinicon(oSearch,"image/notepad.png")
                setWindowTitle("Find/Replace")
                setStyleSheet("background-color:white;")
                setFixedsize(550,160)
                setwindowflags( Qt_CustomizeWindowHint |
                                Qt_WindowTitleHint | Qt_WindowStaysOnTopHint)

                oSearchFilter = new qallevents(oSearch)
                oSearchFilter.setKeyPressEvent("pSearchKeyPress()")
                installeventfilter(oSearchFilter)

                show()
        }

Func pReplace
        oCursor = textedit1.textCursor()
        if oCursor.HasSelection() = false
                new qMessagebox(oSearch)
                {
                        SetWindowTitle("Replace")
                        SetText("No Selection")
                        show()
                }
                return false
        ok
        cValue = oSearchValue.text()
        cSelected = oCursor.SelectedText()
        if oSearchCase.checkState() = Qt_Unchecked
                cValue = lower(cValue)
                cSelected = lower(cSelected)
        ok
        if cSelected != cValue
                new qMessagebox(oSearch)
                {
                        SetWindowTitle("Replace")
                        SetText("No Match")
                        show()
                }
```

(continues on next page)

(continued from previous page)

```
                return false
        ok
        cValue = oReplaceValue.text()
        nStart = oCursor.SelectionStart()
        nEnd = oCursor.SelectionEnd()
        cStr = textedit1.toPlainText()
        cStr = left(cStr,nStart)+cValue+substr(cStr,nEnd+1)
        textedit1.setText(cStr)
        return pFindValue()

Func pReplaceAll
        cStr = textedit1.toPlainText()
        cOldValue = oSearchValue.text()
        cNewValue = oReplaceValue.text()
        if oSearchCase.checkState() = Qt_Unchecked
                # Not Case Sensitive
                cStr = SubStr(cStr,cOldValue,cNewValue,true)
        else
                # Case Sensitive
                cStr = SubStr(cStr,cOldValue,cNewValue)
        ok
        textedit1.setText(cStr)
        new qMessagebox(oSearch)
        {
                SetWindowTitle("Replace All")
                SetText("Operation Done")
                show()
        }

Func pSearchClose
        oSearch.close()
        oSearch = NULL

func pSearchKeyPress
        if oSearchFilter.getKeyCode() = Qt_Key_Escape
                pSearchClose()
        ok

func pFindValue
        oCursor = textedit1.textcursor()
        nPosStart = oCursor.Position() + 1
        cValue = oSearchValue.text()
        cStr = textedit1.toplaintext()
        cStr = substr(cStr,nPosStart)
        if oSearchCase.checkState() = Qt_Unchecked
                cStr = lower(cStr)  cValue = lower(cValue)
        ok
        nPos = substr(cStr,cValue)
        if nPos > 0
                nPos += nPosStart - 2
                oCursor = textedit1.textcursor()
                oCursor.setposition(nPos,0)
```

(continues on next page)

(continued from previous page)

```
                textedit1.settextcursor(oCursor)
                oCursor = textedit1.textcursor()
                oCursor.setposition(nPos+len(cValue),1)
                textedit1.settextcursor(oCursor)
                return true
        else
                new qMessagebox(oSearch)
                {
                        SetWindowTitle("Search")
                        SetText("Cannot find :" + cValue)
                        show()
                }
                return false
        ok

func pNofileopened
        New qMessageBox(win1) {
                setWindowTitle("Sorry")
                setText("Save the file first!")
                show()
        }

func pDebug
        if cActiveFileName = Null return pNofileopened() ok
        cCode = "start run " + cActiveFileName + nl
        system(cCode)

func pRun
        if cActiveFileName = Null return pNofileopened() ok
        cCode = "start ring " + cActiveFileName + nl
        system(cCode)

func pRunNoConsole
        if cActiveFileName = Null return pNofileopened() ok
        cCode = "start /b ring " + cActiveFileName + nl
        system(cCode)

func pSave
        if cActiveFileName = NULL return pSaveAs() ok
        writefile(cActiveFileName,textedit1.toplaintext())
        status1.showmessage("File : " + cActiveFileName + " saved!",0)
        lAskToSave = false

func pSaveAs
        new qfiledialog(win1) {
                cName = getsavefilename(win1,"Save As","","source files(*.ring)")
                if cName != NULL
                        cActiveFileName = cName
                        writefile(cActiveFileName,textedit1.toplaintext())
                        status1.showmessage("File : " + cActiveFileName + " saved!",0)
                        pSetActiveFileName()
                        lAskToSave = false
```

(continues on next page)

(continued from previous page)

```
                ok
        }

func pPrint
        status1.showmessage("Printing to File : RingDoc.pdf",0)
        printer1 = new qPrinter(0) {
                setoutputformat(1)      # 1 = pdf
                setoutputfilename("RingDoc.pdf")
                textedit1.print(printer1)
        }
        status1.showmessage("Done!",0)
        system("RingDoc.pdf")

func pCut
        textedit1.cut()
        status1.showmessage("Cut!",0)

func pCopy
        textedit1.copy()
        status1.showmessage("Copy!",0)

func pPaste
        textedit1.paste()
        status1.showmessage("Paste!",0)

func pFont
        oFontDialog = new qfontdialog() {
                aFont = getfont()
        }
        textedit1.selectall()
        cFont = aFont[1]
        pSetFont()

Func pSetFont
        myfont = new qfont("",0,0,0)
        myfont.fromstring(cFont)
        textedit1.setcurrentfont(myfont)

Func pColor
        new qcolordialog() { aTextColor = GetColor() }
        pSetColors()

Func pColor2
        new qcolordialog() { aBackColor = GetColor() }
        pSetColors()

Func pSetColors
        textedit1.setstylesheet("color: rgb(" + aTextColor[1] + "," + aTextColor[2] +
                                "," + aTextColor[3] + ");" + "background-color: rgb(" +
                                aBackColor[1] + "," + aBackColor[2] + "," +
                                aBackColor[3] + ")")
```

(continues on next page)

(continued from previous page)

```
func pOpen
        new qfiledialog(win1) {
                cName = getopenfilename(win1,"open file","c:\","source files(*.ring)")
                if cName != NULL
                        cActiveFileName = cName
                        textedit1.settext(read(cActiveFileName))
                ok
        }

func pNew
        new qfiledialog(win1) {
                cName = getsavefilename(win1,"New file","","source files(*.ring)")
                if cName != NULL
                        write(cName,"")
                        cActiveFileName = cName
                        textedit1.settext(read(cActiveFileName))

                ok
        }

Func WriteFile cFileName,cCode
        aCode = str2list(cCode)
        fp = fopen(cFileName,"wb")
        for cLine in aCode
                fwrite(fp,cLine+char(13)+char(10))
        next
        fclose(fp)

Func MsgBox cTitle,cMessage
        new qMessagebox(win1) {
                setwindowtitle(cTitle)
                setText(cMessage)
                show()
        }


Func pLang
        MsgBox("Programming Language",
                "This application developed using the Ring programming language")

Func pGUI
        MsgBox("GUI Library",
                "This application uses the Qt GUI Library through RingQt")

Func pAbout
        MsgBox("About",
                "2016, Mahmoud Fayed <msfclipper@yahoo.com>")

Func pSaveSettings
        cSettings = "aTextColor = ["+aTextColor[1]+","+aTextColor[2]+
                        ","+aTextColor[3]+"]" + nl +
                        "aBackColor = ["+aBackColor[1]+","+aBackColor[2]+
```

(continues on next page)

(continued from previous page)

```
                        ","+aBackColor[3]+"]" + nl +
                        "cFont = '" + cFont + "'" + nl +
                        "cWebSite = '" + cWebsite + "'" + nl
        cSettings = substr(cSettings,nl,char(13)+char(10))
        write("ringnotepad.ini",cSettings)
        if lAsktoSave
                new qmessagebox(win1)
                {
                        setwindowtitle("Save Changes?")
                        settext("Some changes are not saved!")
                        setInformativeText("Do you want to save your changes?")
                        setstandardbuttons(QMessageBox_Yes |
                                           QMessageBox_No | QMessageBox_Cancel)
                        result = exec()
                        win1 {
                        if result = QMessageBox_Yes
                                pSave()
                        but result = QMessageBox_Cancel
                                return false
                        ok
                }
        }
        ok
        return true

Func pSetWebsite
        oWebView { loadpage(new qurl(cWebSite)) }
        oWBText  { setText(cWebSite) }

Func RestoreSettings
        eval(read("ringnotepad.ini"))
        pSetColors()
        pSetFont()
        pSetWebsite()

Func pQuit
        if pSaveSettings()
                myapp.quit()
        ok
```

The application during the runtime

The next screen shot demonstrates the “File” menu

The next window for "search and replace"

The next screen shot demonstrates the application main window

Note: the functions pDebug(), pRun() and pRunNoConsole() in the previous sample are not portable! They are written in this sample for MS-Windows and we can update them for other operating systems.

45.57 The Cards Game

In the next example we will see a simple Cards game developed using RingQt

Each player get 5 cards, the cards are unknown to any one. each time one player click on one card to see it. if the card is identical to another card the play get point for each card. if the card value is "5" the player get points for all visible cards.

```
Load "guilib.ring"

nScale = 1

app1 = new qApp

mypic = new QPixmap("cards.jpg")

mypic2 = mypic.copy(0,(124*4)+1,79,124)
Player1EatPic = mypic.copy(80,(124*4)+1,79,124)
Player2EatPic= mypic.copy(160,(124*4)+1,79,124)

aMyCards = []
aMyValues = []
for x1 = 0 to 3
        for y1 = 0 to 12
          temppic = mypic.copy((79*y1)+1,(124*x1)+1,79,124)
                  aMyCards + temppic
```

(continues on next page)

(continued from previous page)

```
                  aMyValues + (y1+1)
        next
next

nPlayer1Score = 0   nPlayer2Score=0

do
        Page1 = new Game
        Page1.Start()
again Page1.lnewgame

mypic.delete()
mypic2.delete()
Player1EatPic.delete()
Player2EatPic.delete()

for t in aMyCards
          t.delete()
next

func gui_setbtnpixmap pBtn,pPixmap
        pBtn {
                setIcon(new qicon(pPixmap.scaled(width(),height(),0,0)))
                setIconSize(new QSize(width(),height()))
        }

Class Game

        nCardsCount = 10
        win1 layout1 label1 label2 layout2 layout3 aBtns aBtns2
        aCards nRole=1 aStatus = list(nCardsCount) aStatus2 = aStatus
        aValues        aStatusValues = aStatus  aStatusValues2 = aStatus
        Player1EatPic   Player2EatPic
        lnewgame = false
        nDelayEat = 0.5
        nDelayNewGame = 1

        func start

                win1 = new qWidget() {
                        setwindowtitle("Five")
                        setstylesheet("background-color: White")
                        showfullscreen()
                }

                layout1 = new qvboxlayout()

                label1 = new qlabel(win1) {
                        settext("Player (1) - Score : " + nPlayer1Score)
                        setalignment(Qt_AlignHCenter | Qt_AlignVCenter)
                        setstylesheet("color: White; background-color: Purple;
                                        font-size:20pt")
```

(continues on next page)

(continued from previous page)

```
                setfixedheight(200)
        }

        closebtn = new qpushbutton(win1)  {
                settext("Close Application")
                setstylesheet("font-size: 18px ; color : white ;
                                background-color: black ;")
                setclickevent("Page1.win1.close()")
        }

        aCards = aMyCards
        aValues = aMyValues

        layout2 = new qhboxlayout()

        aBtns = []

        for x = 1 to nCardsCount
                aBtns + new qpushbutton(win1)
                aBtns[x].setfixedwidth(79*nScale)
                aBtns[x].setfixedheight(124*nScale)
                gui_setbtnpixmap(aBtns[x],mypic2)
                layout2.addwidget(aBtns[x])
                aBtns[x].setclickevent("Page1.Player1click("+x+")")
        next

        layout1.addwidget(label1)
        layout1.addlayout(layout2)

        label2 = new qlabel(win1) {
                settext("Player (2) - Score : " + nPlayer2Score)
                setalignment(Qt_AlignHCenter | Qt_AlignVCenter)
                setstylesheet("color: white; background-color: red;
                                font-size:20pt")
                setfixedheight(200)
        }

        layout3 = new qhboxlayout()

        aBtns2 = []
        for x = 1 to nCardsCount
                aBtns2 + new qpushbutton(win1)
                aBtns2[x].setfixedwidth(79*nScale)
                aBtns2[x].setfixedheight(124*nScale)
                gui_setbtnpixmap(aBtns2[x],mypic2)
                layout3.addwidget(aBtns2[x])
                aBtns2[x].setclickevent("Page1.Player2click("+x+")")
        next

        layout1.addwidget(label2)
        layout1.addlayout(layout3)
        layout1.addwidget(closebtn)
```

(continues on next page)

(continued from previous page)

```
        win1.setlayout(layout1)

        app1.exec()

    Func Player1Click x
        if nRole = 1 and aStatus[x] = 0
            nPos = ((random(100)+clock())%(len(aCards)-1)) + 1
            gui_setbtnpixmap(aBtns[x],aCards[nPos])
            del(aCards,nPos)
            nRole = 2
            aStatus[x] = 1
            aStatusValues[x] = aValues[nPos]
            del(aValues,nPos)
            Player1Eat(x,aStatusValues[x])
            checknewgame()
        ok

    Func Player2Click x
        if nRole = 2 and aStatus2[x] = 0
            nPos = ((random(100)+clock())%(len(aCards)-1)) + 1
            gui_setbtnpixmap(aBtns2[x],aCards[nPos])
            del(aCards,nPos)
            nRole = 1
            aStatus2[x] = 1
            aStatusValues2[x] = aValues[nPos]
            del(aValues,nPos)
            Player2Eat(x,aStatusValues2[x])
            checknewgame()
        ok

    Func Player1Eat nPos,nValue

         app1.processEvents()

         delay(nDelayEat)
         lEat = false
         for x = 1 to nCardsCount
             if aStatus2[x] = 1 and (aStatusValues2[x] = nValue or nValue=5)
                 aStatus2[x] = 2
                 gui_setbtnpixmap(aBtns2[x],Player1EatPic)
                 lEat = True
                 nPlayer1Score++
             ok
             if (x != nPos) and (aStatus[x] = 1) and
                 (aStatusValues[x] = nValue or nValue=5)
                 aStatus[x] = 2
                 gui_setbtnpixmap(aBtns[x],Player1EatPic)
                 lEat = True
                 nPlayer1Score++
             ok
         next
```

(continues on next page)

(continued from previous page)

```
                if lEat
                                nPlayer1Score++
                                gui_setbtnpixmap(aBtns[nPos],Player1EatPic)
                                aStatus[nPos] = 2
                                label1.settext("Player (1) - Score : " + nPlayer1Score)
                ok

        Func Player2Eat nPos,nValue

                app1.processEvents()

                delay(nDelayEat)
                lEat = false
                for x = 1 to  nCardsCount
                        if aStatus[x] = 1 and (aStatusValues[x] = nValue or nValue = 5)
                                aStatus[x] = 2
                                gui_setbtnpixmap(aBtns[x],Player2EatPic)
                                lEat = True
                                nPlayer2Score++
                        ok

                        if (x != nPos) and (aStatus2[x] = 1) and
                                (aStatusValues2[x] = nValue or nValue=5 )
                                aStatus2[x] = 2
                                gui_setbtnpixmap(aBtns2[x],Player2EatPic)
                                lEat = True
                                nPlayer2Score++
                        ok
                next
                if lEat
                                nPlayer2Score++
                                gui_setbtnpixmap(aBtns2[nPos],Player2EatPic)
                                aStatus2[nPos] = 2
                                label2.settext("Player (2) - Score : " + nPlayer2Score)
                ok

        Func checknewgame
                if isnewgame()
                                lnewgame = true

                                if nPlayer1Score > nPlayer2Score
                                        label1.settext("Player (1) Wins!!!")
                                ok
                                if nPlayer2Score > nPlayer1Score
                                        label2.settext("Player (2) Wins!!!")
                                ok

                                app1.processEvents()
                                delay(nDelayNewGame)

                                win1.delete()
                                app1.quit()
```

(continues on next page)

(continued from previous page)

```
                ok

        Func isnewgame
                for t in aStatus
                        if t = 0
                                return false
                        ok
                next
                for t in aStatus2
                        if t = 0
                                return false
                        ok
                next
                return true

        Func delay x
        nTime = x * 1000
        oTest = new qTest
        oTest.qsleep(nTime)
```

The application during the runtime

Note: in the previous screen shot the player get the card number '5' but his score is not increased because he opened this card while no other cards are visible!

The next screen shot while running the game using a Mobile (Android)

Note: using Qt we can run the same application on other Mobile systems

45.58 Classes and their Methods to use the default events

The next table present the class name and the methods that we have to use the default events.

Class Name	Methods to use the default Events
QPushButton	SetClickEvent()
QAction	SetClickEvent()
QLineEdit	SetTextChangedEvent()
	SetCursorPositionChangedEvent()
	SetEditingFinishedEvent()
	SetReturnPressedEvent()
	SetSelectionChangedEvent()
	SetTextEditedEvent()
QTextEdit	SetCopyAvailableEvent()
	SetCurrentCharFormatChangedEvent()

continues on next page

Table 1 – continued from previous page

Class Name	Methods to use the default Events
	SetCursorPositionChangedEvent()
	SetRedoAvailableEvent()
	SetSelectionChangedEvent()
	SetTextChangedEvent()
	SetUndoAvailableEvent()
QListWidget	SetCurrentItemChangedEvent()
	SetCurrentRowChangedEvent()
	SetCurrentTextChangedEvent()
	SetItemActivatedEvent()
	SetItemChangedEvent()
	SetItemClickedEvent()
	SetItemDoubleClickedEvent()
	SetItemEnteredEvent()
	SetItemPressedEvent()
	SetItemSelectionChangedEvent()
QTreeView	SetCollapseEvent()
	SetExpandedEvent()
	SetActivatedEvent()
	SetClickedEvent()
	SetDoubleClickedEvent()
	SetEnteredEvent()
	SetPressedEvent()
	SetViewportEnteredEvent()
QTreeWidget	SetCollapsedEvent()
	SetExpandedEvent()
	SetActivatedEvent()
	SetClickedEvent()
	SetDoubleClickedEvent()
	SetEnteredEvent()
	SetPressedEvent()
	SetViewportEnteredEvent()
	SetCurrentItemChangedEvent()
	SetItemActivatedEvent()
	SetItemChangedEvent()
	SetItemClickedEvent()
	SetItemCollapsedEvent()
	SetItemDoubleClickedEvent()
	SetItemEnteredEvent()
	SetItemExpandedEvent()
	SetItemPressedEvent()
	SetItemSelectionChangedEvent()
QComboBox	SetActivatedEvent()
	SetCurrentIndexChangedEvent()
	SetEditTextChangedEvent()
	SetHighlightedEvent()
QTabWidget	SetCurrentChangedEvent()
	SetTabCloseRequestedEvent()
QTableWidget	SetCellActivatedEvent()
	SetCellChangedEvent()
	SetCellClickedEvent()

continues on next page

Table 1 – continued from previous page

Class Name	Methods to use the default Events
	SetCellDoubleClickedEvent()
	SetCellEnteredEvent()
	SetCellPressedEvent()
	SetCurrentCellChangedEvent()
	SetCurrentItemChangedEvent()
	SetItemActivatedEvent()
	SetItemChangedEvent()
	SetItemClickedEvent()
	SetItemDoubleClickedEvent()
	SetItemEnteredEvent()
	SetItemPressedEvent()
	SetItemSelectionChangedEvent()
QProgressBar	SetValueChangedEvent()
QSpinBox	SetValueChangedEvent()
QSlider	SetActionTriggeredEvent()
	SetRangeChangedEvent()
	SetSliderMovedEvent()
	SetSliderPressedEvent()
	SetSliderReleasedEvent()
	SetValueChangedEvent()
QDial	SetActionTriggeredEvent()
	SetRangeChangedEvent()
	SetSliderMovedEvent()
	SetSliderPressedEvent()
	SetSliderReleasedEvent()
	SetValueChangedEvent()
QWebView	SetLoadFinishedEvent()
	SetLoadProgressEvent()
	SetLoadStartedEvent()
	SetSelectionChangedEvent()
	SetTitleChangedEvent()
	SetUrlChangedEvent()
QCheckBox	SetStateChangedEvent()
	SetClickedEvent()
	SetPressedEvent()
	SetReleasedEvent()
	SetToggledEvent()
QRadioButton	SetClickedEvent()
	SetPressedEvent()
	SetReleasedEvent()
	SetToggledEvent()
QButtonGroup	SetButtonClickedEvent()
	SetButtonPressedEvent()
	SetButtonReleasedEvent()
QVideoWidget	SetBrightnessChangedEvent()
	SetContrastChangedEvent()
	SetFullScreenChangedEvent()
	SetHueChangedEvent()
	SetSaturationChangedEvent()
QTimer	SetTimeoutEvent()

continues on next page

Table 1 – continued from previous page

Class Name	Methods to use the default Events
QTcpServer	SetAcceptErrorEvent()
	SetNewConnectionEvent()
QIODevice	SetAboutToCloseEvent()
	SetBytesWrittenEvent()
	SetReadChannelFinishedEvent()
	SetReadyReadEvent()
QAbstractSocket	SetConnectedEvent()
	SetDisconnectedEvent()
	SetErrorEvent()
	SetHostFoundEvent()
	SetProxyAuthenticationRequiredEvent()
	SetStateChangedEvent()
QTcpSocket	SetConnectedEvent()
	SetDisconnectedEvent()
	SetErrorEvent()
	SetHostFoundEvent()
	SetProxyAuthenticationRequiredEvent()
	SetStateChangedEvent()
	SetAboutToCloseEvent()
	SetBytesWrittenEvent()
	SetReadChannelFinishedEvent()
	SetReadyReadEvent()
QColorDialog	SetColorSelectedEvent()
	SetCurrentColorChangedEvent()
QNetworkAccessManager	SetFinishedEvent()
QThread	SetStartedEvent()
	SetFinishedEvent()

45.59 Methods to use Events with Events Filter

RingQt define a new class called QAllEvents that help you in using Events Filter

The next table presents the methods that we have

Methods to get parameters	Class Name
getKeyCode() –> Number	QAllEvents
getx() –> Number	
gety() –> Number	
getglobalx() –> Number	
getglobaly() –> Number	
getbutton() –> Number	
getbuttons() –> Number	

The next table presents the methods that we have to use events.

Method Name	Class Name
setKeyPressEvent(cEvent)	QAllEvents
setMouseButtonPressEvent(cEvent)	
setMouseButtonReleaseEvent(cEvent)	
setMouseButtonDblClickEvent(cEvent)	
setMouseMoveEvent(cEvent)	
setCloseEvent(cEvent)	
setContextMenuEvent(cEvent)	
setDragEnterEvent(cEvent)	
setDragLeaveEvent(cEvent)	
setDragMoveEvent(cEvent)	
setDropEvent(cEvent)	
setEnterEvent(cEvent)	
setFocusInEvent(cEvent)	
setFocusOutEvent(cEvent)	
setKeyReleaseEvent(cEvent)	
setLeaveEvent(cEvent)	
setNonClientAreaMouseButtonDblClickEvent(cEvent)	
setNonClientAreaMouseButtonPressEvent(cEvent)	
setNonClientAreaMouseButtonReleaseEvent(cEvent)	
setNonClientAreaMouseMoveEvent(cEvent)	
setMoveEvent(cEvent)	
setResizeEvent(cEvent)	
setWindowActivateEvent(cEvent)	
setWindowBlockedEvent(cEvent)	
setWindowDeactivateEvent(cEvent)	
setWindowStateChangeEvent(cEvent)	
setWindowUnblockedEvent(cEvent)	

45.60 The Difference between Qt and RingQt

(1) RingQt use simple methods to set the code that will be executed for events.

Syntax:

```
Set<Event_Name>Event(cEventCode)
```

(2) RingQt change the name of some methods to avoid conflict with Ring Keywords.

The next table present these little changes

Class Name	Qt Method Name	RingQt Method Name
QWebView	load	loadpage
QMediaPlaylist	load	loadfile
QMediaPlaylist	next	movenext
QPainter	end	endpaint
QPicture	load	loadfile
QLineEdit	end	endtext
QDialog	done	donedialog
QTextDocument	end	enddoc
QTextBlock	next	nextblock
QSqlQuery	next	movenext
QImage	load	loadimage
QNetworkAccessManager	get	getvalue
QNetworkAccessManager	put	putvalue
QThread	exit	exitfromthread
QRegularExpressionMatchIterator	next	nextitem
QCamera	load	loadcamera

45.61 RingQt Classes and their Qt Documentation

Qt Documentation : http://doc.qt.io/qt-5/classes.html

See the "RingQt Classes and Methods Reference" chapter for supported classes and methods.

45.62 New Classes names - Index Start from 1

We added new classes to RingQt - another version of classes where the class names doesn't start with the "q" letter Also updated methods so the index start from 1 when we deal with the GUI controls like

- ComboBox
- ListWidget
- TableWidget
- TreeWidget

These classes are inside guilib.ring under the package name : System.GUI

To use it

```
load "guilib.ring"

import System.GUI
```

This doesn't have any effect on our previous code, It's just another choice for better code that is consistent with Ring rules.

Also the form designer is updated to provide us the choice between using classes where (index start from 0) or (index start from 1)

Example (Uses the Form Designer)

(1) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/indexstart/indexstartView.ring

(2) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/indexstart/indexstartController.ring

45.63 TableWidget - AddList() Method

Using this method we can add Ring List to the TableWidget

Tip: TableWidget class is a subclass of QTableWidget class

Note: To use TableWidget class, import system.gui after loading guilib.ring or lightguilib.ring

Example:

Source code: https://github.com/ring-lang/ring/tree/master/samples/UsingQt/TableWidget/AddRingList

```
class addRingListController from windowsControllerParent

        oView = new addRingListView

        aList = [["one","two"],
                 ["three","four"],
                 ["five","six"],
                 [7,8],
                 ["I","Love","Ring","Programming"]]

        oView.tablewidget1.addList(aList)

        aList = [["Number","Square"]]
        for t = 1 to 10
                aList + [ t, t*t]
        next

        oView.tablewidget1.addList(aList)
```

Screen Shot:

45.64 Creating Reports using the WebLib and the GUILib

The WebLib comes with a class called HtmlPage

Using this class we can create reports quickly using WebLib & GUILib together

Example:

```
load "stdlib.ring"
load "weblib.ring"
load "guilib.ring"

import System.Web
import System.GUI
```

(continues on next page)

(continued from previous page)

```
new qApp {
        open_window(:CustomersReportController)
        exec()
}

class CustomersReportController from WindowsControllerParent

        oView = new CustomersReportView

        func Start
                CreateReport()

        func CreateReport
                mypage = new HtmlPage {
                        h1 { text("Customers Report") }
                        Table
                        {
                                style = stylewidth("100%") + stylegradient(4)
                                TR
                                {
                                        TD { WIDTH="10%"
                                                text("Customers Count : " )  }
                                        TD { text (100) }
                                }
                        }
                        Table
                        {
                                style = stylewidth("100%") + stylegradient(26)
                                TR
                                {
                                        style = stylewidth("100%") +
                                                stylegradient(24)
                                        TD { text("Name " )  }
                                        TD { text("Age" ) }
                                        TD { text("Country" ) }
                                        TD { text("Job" ) }
                                        TD { text("Company" ) }
                                }
                                for x =  1 to 100
                                        TR
                                        {
                                                TD { text("Test" )  }
                                                TD { text("30" ) }
                                                TD { text("Egypt" ) }
                                                TD { text("Sales" ) }
                                                TD { text("Future" ) }
                                        }
                                next
                        }
                }
                write("report.html",mypage.output())
```

(continues on next page)

(continued from previous page)

```
        func PrintEvent
                printer1 = new qPrinter(0) {
                        setoutputformat(1)
                        setoutputfilename("report.pdf")
                }
                oView {
                        web.print(printer1, Method( :OpenPDF ) )
                        web.show()
                }

        func OpenPDF
                new QDesktopServices {
                        OpenURL(new qURL("report.pdf") )
                }

class CustomersReportView

                win = new window() {
                                setwindowtitle("Report Window")
                                setgeometry(100,100,500,500)
                                web = new webview(win) {
                                        setgeometry(100,100,1000,500)
                                        loadpage(new qurl("file:///"+
                                        currentdir()+"/report.html"))
                                }
                                new pushbutton(win) {
                                                setGeometry(100,20,100,30)
                                                settext("Print")
                                                setclickevent(Method(:PrintEvent))
                                }
                                showMaximized()
                        }
```

Screen Shot:

Name	Age	Country	Job	Company
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future
Test	30	Egypt	Sales	Future

CHAPTER

FORTYSIX

USING THE FORM DESIGNER

In this chapter we will learn about using the Form Designer.

We can run the From Designer from Ring Notepad

From the Menubar in Ring Notepad - View Menu - We can Show/Hide the Form Designer window.

Also we can run the Form Designer in another window.

From the Ring Notepad - Tools Menu - Select the Form Designer.

46.1 The Designer Windows

- Toolbox : To select controls to be added to the window.
- Properties : To set the properties of the active window or controls.
- Design Region : To select, move and resize the window and the controls.

46.2 The Toolbox

We have many buttons.

- Lock : We can use it to draw many controls of the same type quickly.
- Select : We can use it to select a control in the Design Region
- Controls Buttons : Select a control to be added to the window.

46.3 The Properties

- When we select the window or one control, We will have the selected object properties.
- Also In the properties window we have a combobox to select the active control.
- Some properties provide a button next to the property value. We can click on the button to get more options.
- When we select more than one control, We will have options for multi-selection

46.4 Running Forms

When we save the form file (*.rform), The Form Designer will create two Ring files

- The Controller Class
- The View Class

For example, if the form file is helloworld.rform

The form designer will generate two files

- helloworldcontroller.ring
- helloworldview.ring

To run the program, Open the controller class file then click the Run button (CTRL+F5)

Tip: When you open a form using Ring Notepad, the controller class will be opened automatically, So we can press (CTRL+F5) or click on the Run button while the form designer window is active.

46.5 Events Code

(1) Just type the method name in the event property.

This will add the next code to the Controller Class

```
func CloseWindow
        oView {
        }
```

(2) Then write the method code in the controller class.

Source Code : B:/ring/applications/formdesigner/tests/buttontoclosethewindow/buttontoclosethewindowController.ring

buttontoclosethewindowController.ring

```
# Form/Window Controller - Source Code File

load "buttontoclosethewindowView.ring"

if IsMainSourceFile() {
        new qApp {
                StyleFusion()
                open_window(:buttontoclosethewindowController)
                exec()
        }
}

class buttontoclosethewindowController from windowsControllerParent

        oView = new buttontoclosethewindowView

        func CloseWindow
                oView.win.close()
```

Source Code : B:/ring/applications/formdesigner/tests/buttontoclosethewindow/buttontoclosethewindowController.ring Form Designer Web Browser

In this example we write

```
func CloseWindow
        oView.win.close()
```

Where inside the controller class, We uses the oView object to access the form.

Another Example :

The Event Code

```
func SayHello
        oView {
                LineEdit2.setText("Hello "+ LineEdit1.text() )
        }
```

46.6 Keyboard Shortcuts

After selecting one or group of controls

- Use the Arrows (Up, Down, Left and Right) to move them around.
- Shift + the Arrows (Up, Down, Left and Right) to Resize the controls.
- Del button to delete the controls.
- CTRL+SHIFT+V to Duplicate the controls.

46.7 Menubar Designer

From the Window properties we can open the Menubar Designer

46.8 Window Flags

From the Window properties we can open the Window Flags window.

46.9 Entering Items

For some controls like the List Widget we can enter items separated by comma ‘,’

46.10 Using Layouts

(1) To use layouts, At first add the layout control to the window.

(2) Use the window “Set Layout” property to determine the main layout.

(3) From the layout properties determine the controls and the layout type.

46.11 Objects Order and Layouts

All the objects that are added to a layout must comes first!

If we have a form that contains two buttons and a Layout

Then adding a third button after creating the layout requires changing the objects order

We can open the (Objects Order) window using a button from the (Properties) window

Then we can select the (Button3) and click (Move Up)

Then we close the (Objects Order) window

46.12 More Samples and Tests

Check the folder : ring/samples/UsingFormDesigner

Online : https://github.com/ring-lang/ring/tree/master/samples/UsingFormDesigner

CHAPTER

FORTYSEVEN

GRAPHICS PROGRAMMING USING RINGQT3D

In this chapter we will learn how to use Qt3D through many samples.

47.1 Drawing Cube

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)
```

(continues on next page)

(continued from previous page)

```
        oCube = new  QCuboidMesh(oRootEntity) {
                setXextent(2)
                setYextent(2)
                setZextent(3)
        }

        oCubeTransform = new  QTransform(oCube)
        oCubeTransform.setScale(2)
        oCubeTransform.setTranslation(new QVector3D(3, 3, 3))

        oCubeMaterial = new QPhongMaterial(oCube)
        oCubeMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})

        oCubeEntity = new QEntity(oRootEntity)
        oCubeEntity.addComponent(oCube)
        oCubeEntity.addComponent(oCubeMaterial)
        oCubeEntity.addComponent(oCubeTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Cube")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.2 Drawing Torus

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)
```

(continues on next page)

(continued from previous page)

```
        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)

        oTorus = new  QTorusMesh(oRootEntity)
                oTorus.setRadius(1.0)
                oTorus.setMinorRadius(0.4)
                oTorus.setRings(100)
                oTorus.setSlices(20)

        oTorusTransform = new  QTransform(oTorus)
        oTorusTransform.setScale(2)
        oTorusTransform.setTranslation(new QVector3D(3, 3, 3))

        oTorusMaterial = new QPhongMaterial(oTorus)
        oTorusMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})

        oTorusEntity = new QEntity(oRootEntity)
        oTorusEntity.addComponent(oTorus)
        oTorusEntity.addComponent(oTorusMaterial)
        oTorusEntity.addComponent(oTorusTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Torus")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.3 Drawing Sphere

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
```

(continues on next page)

(continued from previous page)

```
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)

        oSphere = new  QSphereMesh(oRootEntity)
                oSphere.setRadius(1.0)
                oSphere.setRings(100)
                oSphere.setSlices(20)

        oSphereTransform = new  QTransform(oSphere)
        oSphereTransform.setScale(2)
        oSphereTransform.setTranslation(new QVector3D(3, 3, 3))

        oSphereMaterial = new QPhongMaterial(oSphere)
        oSphereMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})

        oSphereEntity = new QEntity(oRootEntity)
        oSphereEntity.addComponent(oSphere)
        oSphereEntity.addComponent(oSphereMaterial)
        oSphereEntity.addComponent(oSphereTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Sphere")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.4 Drawing Cylinder

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
```

(continues on next page)

(continued from previous page)

```
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)


        oCylinder = new  QCylinderMesh(oRootEntity)
                oCylinder.setRadius(1)
                oCylinder.setRings(100)
                oCylinder.setSlices(20)
        oCylinder.setLength(5)

        oCylinderTransform = new  QTransform(oCylinder)
        oCylinderTransform.setScale(2)
        oCylinderTransform.setTranslation(new QVector3D(1, 0, 3))

        oCylinderMaterial = new QPhongMaterial(oCylinder)
        oCylinderMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})

        oCylinderEntity = new QEntity(oRootEntity)
        oCylinderEntity.addComponent(oCylinder)
        oCylinderEntity.addComponent(oCylinderMaterial)
        oCylinderEntity.addComponent(oCylinderTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Cylinder")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.5 Drawing Cone

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLightEntity.addComponent(oLight)
```

(continues on next page)

(continued from previous page)

```
        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)


        oCone = new  QConeMesh(oRootEntity)
                oCone.setRings(100)
                oCone.setSlices(20)
        oCone.setLength(5)

        oConeTransform = new  QTransform(oCone)
        oConeTransform.setScale(2)
        oConeTransform.setTranslation(new QVector3D(1, 0, 3))

        oConeMaterial = new QPhongMaterial(oCone)
        oConeMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})

        oConeEntity = new QEntity(oRootEntity)
        oConeEntity.addComponent(oCone)
        oConeEntity.addComponent(oConeMaterial)
        oConeEntity.addComponent(oConeTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Cone")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.6 Drawing Plane

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)
```

(continues on next page)

(continued from previous page)

```
        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)


        oPlane = new  QPlaneMesh(oRootEntity)
        oPlane.setHeight(5)
        oPlane.setWidth(5)
        oPlane.setmeshresolution(new qSize(10,10))

        oPlaneTransform = new  QTransform(oPlane)
        oPlaneTransform.setScale(2)
        oPlaneTransform.setTranslation(new QVector3D(0, -4, 4))

        oPlaneMaterial = new QPhongMaterial(oPlane)
        oPlaneMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})

        oPlaneEntity = new QEntity(oRootEntity)
        oPlaneEntity.addComponent(oPlane)
        oPlaneEntity.addComponent(oPlaneMaterial)
        oPlaneEntity.addComponent(oPlaneTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Plane")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.7 Texture

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
```

(continues on next page)

(continued from previous page)

```
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)

        oCube = new  QCuboidMesh(oRootEntity) {
                setXextent(2)
                setYextent(2)
                setZextent(3)
        }

        oCubeTransform = new  QTransform(oCube)
        oCubeTransform.setScale(2)
        oCubeTransform.setTranslation(new QVector3D(3, 3, 3))

        oTextureLoader = new  QTextureLoader(oCube);
        oTextureLoader.setSource(
                new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
        oCubeMaterial = new QTextureMaterial(oCube)
        oCubeMaterial.setTexture(oTextureLoader)

        oCubeEntity = new QEntity(oRootEntity)
        oCubeEntity.addComponent(oCube)
        oCubeEntity.addComponent(oCubeMaterial)
        oCubeEntity.addComponent(oCubeTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Adding Texture")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.8 Key Press

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

                oFilter = new qallevents(oView)
                oFilter.setKeyPressEvent("pKeyPress()")
                oView.installeventfilter(oFilter)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
```

(continues on next page)

(continued from previous page)

```
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)
        oCamController.setEnabled(False)

        oCube = new  QCuboidMesh(oRootEntity) {
                setXextent(2)
                setYextent(2)
                setZextent(3)
        }

        oCubeTransform = new  QTransform(oCube)
        oCubeTransform.setScale(2)
        oCubeTransform.setTranslation(new QVector3D(3, 3, 3))

        oTextureLoader = new  QTextureLoader(oCube);
        oTextureLoader.setSource(
                new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
        oCubeMaterial = new QTextureMaterial(oCube)
        oCubeMaterial.setTexture(oTextureLoader)

        oCubeEntity = new QEntity(oRootEntity)
        oCubeEntity.addComponent(oCube)
        oCubeEntity.addComponent(oCubeMaterial)
        oCubeEntity.addComponent(oCubeTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Moving Cube using the Keyboard")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        oContainer.setfocus(0)

        exec()
}

func pKeyPress
        nKey = oFilter.getkeycode()
        oX = oCubeTransform.translation().x()
```

(continues on next page)

(continued from previous page)

```
        oY = oCubeTransform.translation().y()
        oZ = oCubeTransform.translation().z()
        switch nKey
                on Qt_Key_Right
                        oX++
                on Qt_Key_Left
                        oX--
                on Qt_Key_Up
                        oY++
                on Qt_Key_Down
                        oY--
        off
        oCubeTransform.setTranslation(new QVector3D(oX, oY, oZ))
```

47.9 Object Picker

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
```

(continues on next page)

(continued from previous page)

```
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

                oFilter = new qallevents(oView)
                oFilter.setKeyPressEvent("pKeyPress()")
                oView.installeventfilter(oFilter)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)
        oCamController.setEnabled(False)

        oCube = new  QCuboidMesh(oRootEntity) {
                setXextent(2)
                setYextent(2)
                setZextent(3)
        }

        oCubeTransform = new  QTransform(oCube)
        oCubeTransform.setScale(2)
        oCubeTransform.setTranslation(new QVector3D(3, 3, 3))

        oTextureLoader = new  QTextureLoader(oCube);
        oTextureLoader.setSource(
                new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
        oCubeMaterial = new QTextureMaterial(oCube)
        oCubeMaterial.setTexture(oTextureLoader)

        oCubeEntity = new QEntity(oRootEntity)
        oCubeEntity.addComponent(oCube)
        oCubeEntity.addComponent(oCubeMaterial)
        oCubeEntity.addComponent(oCubeTransform)
```

(continues on next page)

(continued from previous page)

```
        oPicker = new qObjectPicker(oCube) {
                setclickedevent("pClick()")
        }
        oCubeEntity.addComponent(oPicker)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Object Picker - Click on the Cube")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        oContainer.setfocus(0)

        exec()
}

func pKeyPress
        nKey = oFilter.getkeycode()
        oX = oCubeTransform.translation().x()
        oY = oCubeTransform.translation().y()
        oZ = oCubeTransform.translation().z()
        switch nKey
                on Qt_Key_Right
                        oX++
                on Qt_Key_Left
                        oX--
                on Qt_Key_Up
                        oY++
                on Qt_Key_Down
                        oY--
        off
        oCubeTransform.setTranslation(new QVector3D(oX, oY, oZ))

func pClick
        msginfo("Event","Click")
        oContainer.setfocus(0)
```

47.10 Frame Action

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
```

(continues on next page)

(continued from previous page)

```
        oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
        oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
        oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

oLightEntity = new QEntity(oRootEntity)
        oLight = new QPointLight(oLightEntity)
oLight.setColor(new QColor() { setRGB(255,255,255,255) })
oLight.setIntensity(1)
oLightEntity.addComponent(oLight)

oLightTransform = new QTransform(oLightEntity)
oLightTransform.setTranslation(oCameraEntity.position())
oLightEntity.addComponent(oLightTransform)

oCamController = new  QFirstPersonCameraController(oRootEntity)
        oCamController.setCamera(oCameraEntity)
oCamController.setEnabled(False)

oCube = new  QCuboidMesh(oRootEntity) {
        setXextent(1)
        setYextent(1)
        setZextent(1)
}

oCubeTransform = new  QTransform(oCube)
oCubeTransform.setScale(2)
oCubeTransform.setTranslation(new QVector3D(-5, -5, -5))

oTextureLoader = new  QTextureLoader(oCube);
oTextureLoader.setSource(
        new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
oCubeMaterial = new QTextureMaterial(oCube)
oCubeMaterial.setTexture(oTextureLoader)

oCubeEntity = new QEntity(oRootEntity)
oCubeEntity.addComponent(oCube)
oCubeEntity.addComponent(oCubeMaterial)
oCubeEntity.addComponent(oCubeTransform)

nAngle=0        nSpeed=0.1
oFrameAction = new qFrameAction(oRootEntity) {
        settriggeredevent("pEvent()")
}

oView.setRootEntity(oRootEntity)

oLayout = new QVBoxLayout()
oLayout.AddWidget(oContainer)

oWidget {
        setwindowtitle("Using Qt3D - Frame Action")
        resize(800,600)
```

(continues on next page)

(continued from previous page)

```
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}

func pEvent

        # Move the Cube
                oX = oCubeTransform.translation().x()
                oY = oCubeTransform.translation().y()
                oZ = oCubeTransform.translation().z()
                if oX >= 4
                        nSpeed=-0.1
                but oX <= -10
                        nSpeed=0.1
                ok
                oCubeTransform.setTranslation(
                        new QVector3D(oX+nSpeed, oY+nSpeed, oZ+nSpeed))

        # Rotate the Cube
                nAngle+=5  if nAngle=360  nAngle=0 ok
                oQ = new QQuaternion(0,0,0,0)
                oCubeTransform.setRotation(
                        oQ.fromAxisAndAngle(new QVector3D(0, 1, 0), nAngle))
```

47.11 Text 2D

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oView.defaultframegraph().setclearcolor(new QColor() {setRGB(100,250,150,255)})

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()
```

(continues on next page)

(continued from previous page)

```
oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
        oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
        oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
        oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

oLightEntity = new QEntity(oRootEntity)
        oLight = new QPointLight(oLightEntity)
oLight.setColor(new QColor() { setRGB(255,255,255,255) })
oLight.setIntensity(1)
oLightEntity.addComponent(oLight)

oLightTransform = new QTransform(oLightEntity)
oLightTransform.setTranslation(oCameraEntity.position())
oLightEntity.addComponent(oLightTransform)

oCamController = new  QFirstPersonCameraController(oRootEntity)
        oCamController.setCamera(oCameraEntity)

oCube = new  QCuboidMesh(oRootEntity) {
        setXextent(2)
        setYextent(2)
        setZextent(3)
}

oCubeTransform = new  QTransform(oCube)
oCubeTransform.setScale(2)
oCubeTransform.setTranslation(new QVector3D(0, 3, 4))

oTextureLoader = new  QTextureLoader(oCube);
oTextureLoader.setSource(
        new QUrl("file:///"+currentdir()+"/assets/texture/ring.bmp") )
oCubeMaterial = new QTextureMaterial(oCube)
oCubeMaterial.setTexture(oTextureLoader)

oCubeEntity = new QEntity(oRootEntity)
oCubeEntity.addComponent(oCube)
oCubeEntity.addComponent(oCubeMaterial)
oCubeEntity.addComponent(oCubeTransform)


oText2DEntity = new  QText2DEntity(oRootEntity) {
        setText("Ring programming language")
        setWidth(400) setHeight(40)
        setColor(new QColor() {setRGB(128,128,128,255)})
}

oText2DTransform = new  QTransform(oText2DEntity)
oText2DTransform.setScale(0.1)
oText2DTransform.setTranslation(new QVector3D(-10.5, -5, 0))

oText2DEntity.addComponent(oText2DTransform)
```

(continues on next page)

(continued from previous page)

```
        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Text2D")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.12 Extruded Text

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oView.defaultframegraph().setclearcolor(new QColor() {setRGB(100,250,150,255)})

        oInput = new QInputAspect(oRootEntity)
```

(continues on next page)

(continued from previous page)

```
        oView.registerAspect(oInput)

oCameraEntity = oView.Camera()

oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
        oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
        oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
        oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

oLightEntity = new QEntity(oRootEntity)
        oLight = new QPointLight(oLightEntity)
oLight.setColor(new QColor() { setRGB(255,255,255,255) })
oLight.setIntensity(1)
oLightEntity.addComponent(oLight)

oLightTransform = new QTransform(oLightEntity)
oLightTransform.setTranslation(oCameraEntity.position())
oLightEntity.addComponent(oLightTransform)

oCamController = new  QFirstPersonCameraController(oRootEntity)
        oCamController.setCamera(oCameraEntity)

oCube = new  QCuboidMesh(oRootEntity) {
        setXextent(2)
        setYextent(2)
        setZextent(3)
}

oCubeTransform = new  QTransform(oCube)
oCubeTransform.setScale(2)
oCubeTransform.setTranslation(new QVector3D(0, 3, 4))

oTextureLoader = new  QTextureLoader(oCube);
oTextureLoader.setSource(
        new QUrl("file:///"+currentdir()+"/assets/texture/ring.bmp") )
oCubeMaterial = new QTextureMaterial(oCube)
oCubeMaterial.setTexture(oTextureLoader)

oCubeEntity = new QEntity(oRootEntity)
oCubeEntity.addComponent(oCube)
oCubeEntity.addComponent(oCubeMaterial)
oCubeEntity.addComponent(oCubeTransform)

oTextEntity = new QEntity(oRootEntity)

oTextMesh = new  QExtrudedTextMesh(oTextEntity) {
        setText("Ring")
}

oTextTransform = new  QTransform(oTextEntity)
oTextTransform.setScale(3)
oTextTransform.setTranslation(new QVector3D(-5.5, -4, 3))
```

(continues on next page)

(continued from previous page)

```
        oTextMaterial = new QPhongMaterial(oTextEntity);
        oTextMaterial.setDiffuse(new QColor() {setRGB(0,0,255,255)})

        oTextEntity.addComponent(oTextMesh)
        oTextEntity.addComponent(oTextTransform)
        oTextEntity.addComponent(oTextMaterial)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Extruded Text")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.13 Model

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)

        oModel = new qmesh(oRootEntity)

        oModel.setsource(
                new qURL("file:///"+currentdir()+"/assets/model/lucky_cat.obj") )

        oModelTransform = new  QTransform(oModel)
        oModelTransform.setScale(0.1)
        oModelTransform.setTranslation(new QVector3D(0, 0, 0))
        oQ = new QQuaternion(0,0,0,0)
        oModelTransform.setRotation(oQ.fromAxisAndAngle(new QVector3D(0, 1, 0), 180))

        oModelMaterial = new QPhongMaterial(oModel)
        oModelMaterial.setDiffuse(new QColor() {setRGB(0,255,128,255)})

        oModelEntity = new QEntity(oRootEntity)
        oModelEntity.addComponent(oModel)
```

(continues on next page)

(continued from previous page)

```
        oModelEntity.addComponent(oModelmaterial)
        oModelEntity.addComponent(oModelTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Model (Obj File)")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.14 Model Texture

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)
```

(continues on next page)

(continued from previous page)

```
        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 25, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)

        oModel = new qmesh(oRootEntity)

        oModel.setsource(
                new qURL("file:///"+currentdir()+"/assets/model/Robot.obj") )

        oModelTransform = new  QTransform(oModel)
        oModelTransform.setScale(0.5)
        oModelTransform.setTranslation(new QVector3D(0, 12, 4))

        oLoader = new  QTextureLoader(oModel)
        oModelMaterial = new QTextureMaterial(oModel)
        oLoader.setSource(
                new QUrl("file:///"+currentdir()+"/assets/texture/Robot.jpg") )
        oModelMaterial.setTexture(oLoader)

        oModelEntity = new QEntity(oRootEntity)
        oModelEntity.addComponent(oModel)
        oModelEntity.addComponent(oModelMaterial)
        oModelEntity.addComponent(oModelTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Model Texture")
```

(continues on next page)

(continued from previous page)

```
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.15 Draw Office

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
```

(continues on next page)

(continued from previous page)

```
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 0, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)

        oModel = new qmesh(oRootEntity)

        oModel.setsource(
                new qURL("file:///"+currentdir()+"/assets/model/Reception_Table.obj") )

        oModelTransform = new  QTransform(oModel)
        oModelTransform.setScale(1)
        oModelTransform.setTranslation(new QVector3D(0, -2.5, 16))

        oModelMaterial = new QPhongMaterial(oModel)
        oModelMaterial.setDiffuse(new QColor() {setRGB(0,255,128,255)})

        oModelEntity = new QEntity(oRootEntity)
        oModelEntity.addComponent(oModel)
        oModelEntity.addComponent(oModelmaterial)
        oModelEntity.addComponent(oModelTransform)

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Model (Obj File) - Office")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.16 Many Objects

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
```

(continues on next page)

(continued from previous page)

```
        oCameraEntity.setViewCenter(new QVector3D(0, 20, 0))

oLightEntity = new QEntity(oRootEntity)
        oLight = new QPointLight(oLightEntity)
oLight.setColor(new QColor() { setRGB(255,255,255,255) })
oLight.setIntensity(1)
oLightEntity.addComponent(oLight)

oLightTransform = new QTransform(oLightEntity)
oLightTransform.setTranslation(oCameraEntity.position())
oLightEntity.addComponent(oLightTransform)

oCamController = new  QFirstPersonCameraController(oRootEntity)
        oCamController.setCamera(oCameraEntity)

oModel = new qmesh(oRootEntity)
oModel.setsource(
        new qURL("file:///"+currentdir()+"/assets/model/Robot.obj") )

oModelTransform = new  QTransform(oModel)
oModelTransform.setScale(0.2)
oModelTransform.setTranslation(new QVector3D(0, 10, 10))

oModelMaterial = new QPhongMaterial(oModel)
oModelMaterial.setDiffuse(new QColor() {setRGB(0,100,0,0)})

oModelEntity = new QEntity(oRootEntity)
oModelEntity.addComponent(oModel)

oLoader = new  QTextureLoader(oModel);
oModelMaterial = new QTextureMaterial(oModel)
oLoader.setSource(
        new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
oModelMaterial.setTexture(oLoader)

oModelEntity.addComponent(oModelMaterial)
oModelEntity.addComponent(oModelTransform)

for n = 1 to 10

        oTorus = new  QTorusMesh(oRootEntity)
                oTorus.setRadius(1.0*n)
                oTorus.setMinorRadius(0.4*n)
                oTorus.setRings(100)
                oTorus.setSlices(20)

        oTorusTransform = new  QTransform(null)
        oTorusTransform.setScale(2)
        oTorusTransform.setTranslation(new QVector3D(5.0*n, 4.0*n, 0.0))

        oTorusMaterial = new QPhongMaterial(null);
        oTorusMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})
```

(continues on next page)

(continued from previous page)

```
                oTorusEntity = new QEntity(oRootEntity)
                oTorusEntity.addComponent(oTorus)

                oLoader = new  QTextureLoader(oTorus);
                oTorusMaterial = new QTextureMaterial(oTorus)
                oLoader.setSource(
                        new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
                oTorusMaterial.setTexture(oLoader)

                oTorusEntity.addComponent(oTorusMaterial)
                oTorusEntity.addComponent(oTorusTransform)
        next

        oView.setRootEntity(oRootEntity)

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)

        oWidget {
                setwindowtitle("Using Qt3D - Many Objects")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}
```

47.17 Camera

```
load "guilib.ring"

new qApp {

        oView = new Qt3dwindow()

        oWidget = new QWidget()
        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000.0)
                oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(0, 20, 0))

        oLightEntity = new QEntity(oRootEntity)
                oLight = new QPointLight(oLightEntity)
```

(continues on next page)

(continued from previous page)

```
        oLight.setColor(new QColor() { setRGB(255,255,255,255) })
        oLight.setIntensity(1)
        oLightEntity.addComponent(oLight)

        oLightTransform = new QTransform(oLightEntity)
        oLightTransform.setTranslation(oCameraEntity.position())
        oLightEntity.addComponent(oLightTransform)

        oCamController = new  QFirstPersonCameraController(oRootEntity)
                oCamController.setCamera(oCameraEntity)

        oModel = new qmesh(oRootEntity)
        oModel.setsource(
                new qURL("file:///"+currentdir()+"/assets/model/Robot.obj") )

        oModelTransform = new  QTransform(oModel)
        oModelTransform.setScale(0.2)
        oModelTransform.setTranslation(new QVector3D(0, 10, 10))

        oModelMaterial = new QPhongMaterial(oModel)
        oModelMaterial.setDiffuse(new QColor() {setRGB(0,100,0,0)})

        oModelEntity = new QEntity(oRootEntity)
        oModelEntity.addComponent(oModel)

        oLoader = new  QTextureLoader(oModel);
        oModelMaterial = new QTextureMaterial(oModel)
        oLoader.setSource(
                new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
        oModelMaterial.setTexture(oLoader)

        oModelEntity.addComponent(oModelMaterial)
        oModelEntity.addComponent(oModelTransform)

        for n = 1 to 10

                oTorus = new  QTorusMesh(oRootEntity)
                        oTorus.setRadius(1.0*n)
                        oTorus.setMinorRadius(0.4*n)
                        oTorus.setRings(100)
                        oTorus.setSlices(20)

                oTorusTransform = new  QTransform(null)
                oTorusTransform.setScale(2)
                oTorusTransform.setTranslation(new QVector3D(5.0*n, 4.0*n, 0.0))

                oTorusMaterial = new QPhongMaterial(null);
                oTorusMaterial.setDiffuse(new QColor() {setRGB(200,100,100,100)})

                oTorusEntity = new QEntity(oRootEntity)
                oTorusEntity.addComponent(oTorus)
```

(continues on next page)

(continued from previous page)

```
                oLoader = new  QTextureLoader(oTorus);
                oTorusMaterial = new QTextureMaterial(oTorus)
                oLoader.setSource(
                        new QUrl("file:///"+currentdir()+"/assets/texture/gold.jpg") )
                oTorusMaterial.setTexture(oLoader)

                oTorusEntity.addComponent(oTorusMaterial)
                oTorusEntity.addComponent(oTorusTransform)
        next

        oView.setRootEntity(oRootEntity)


        btn1 = new qPushButton(oWidget) {
                setText("Move the Camera and the Robot") setClickEvent("pMove()")
        }

        oLayout = new QVBoxLayout()
        oLayout.AddWidget(oContainer)
        oLayout.AddWidget(btn1)

        oWidget {
                setwindowtitle("Using Qt3D - Camera")
                resize(800,600)
                setLayout(oLayout)
                showMaximized()
        }

        exec()
}

func pMove

        oCameraEntity.setPosition(new QVector3D(0, 0, 20.0))
                oCameraEntity.setUpVector(new QVector3D(0, 1, 0))
                oCameraEntity.setViewCenter(new QVector3D(20, 15, 20))

        oModelTransform.setTranslation(new QVector3D(20, 15, 20))
        oQ = new QQuaternion(0,0,0,0)
        oModelTransform.setRotation(oQ.fromAxisAndAngle(new QVector3D(1, 1, 0), 270))
```

47.18 Scence

```
load "guilib.ring"

new qApp {

        oWidget = new QWidget()

        oView = new Qt3DWindow()
        oView.defaultFrameGraph().setClearColor(new QColor() {setRGB(0,0,0,255)})

        oContainer = oWidget.createWindowContainer(oView,oWidget,0)

        oRootEntity = new QEntity(oContainer)

        oInput = new QInputAspect(oRootEntity)
                oView.registerAspect(oInput)

        oCameraEntity = oView.Camera()

        oCameraEntity.lens().setPerspectiveProjection(45.0, 16.0/9.0, 0.1, 1000)
        oCameraEntity.setPosition(new QVector3D(-13.34, -6.43, 16.47))
                oCameraEntity.setUpVector(new QVector3D(0.02, 0, 1))
                oCameraEntity.setViewCenter(new QVector3D(-13.34, 17.05, 15.42))

        oCameraController = new  QFirstPersonCameraController(oRootEntity)
        oCameraController.setCamera(oCameraEntity)
        oCameraController.setEnabled(False)
```

(continues on next page)

(continued from previous page)

```
oLongRoomEntity = new QEntity(oRootEntity)

oLongRoomModel = new QMesh(oLongRoomEntity)

oLongRoomModel.setSource(
        new qURL("file:///"+CurrentDir()+"/assets/model/Long_Room.obj") )

oLongRoomTransform = new  QTransform(oLongRoomEntity)
oLongRoomTransform.setScale(1)
oLongRoomTransform.setTranslation(new QVector3D(5, 0, 15))

oLongRoomLoader = new  QTextureLoader(oLongRoomModel)
oLongRoomMaterial = new QTextureMaterial(oLongRoomModel)
oLongRoomLoader.setSource(
        new QUrl("file:///"+currentdir()+"/assets/texture/croc.jpg") )
oLongRoomMaterial.setTexture(oLongRoomLoader)

oLongRoomEntity.addComponent(oLongRoomModel)
oLongRoomEntity.addComponent(oLongRoomMaterial)
oLongRoomEntity.addComponent(oLongRoomTransform)

oTableEntity = new QEntity(oRootEntity)

oTableModel = new QMesh(oTableEntity)
oTableModel.setSource(
        new qURL("file:///"+CurrentDir()+"/assets/model/Reception_Table.obj") )

oTableTransform = new  QTransform(oTableEntity)
oTableTransform.setScale(0.3)
oTableTransform.setTranslation(new QVector3D(5, 0, 15))
oQ = new QQuaternion(0,0,0,0)
oTableTransform.setRotation(oQ.fromAxisAndAngle(new QVector3D(0, 1, 1), -180))

oTableModelMaterial = new QPhongMaterial(oTableEntity)
oTableModelMaterial.setDiffuse(new QColor() {setRGB(255,255,255,255)})

oTableEntity.addComponent(oTableModel)
oTableEntity.addComponent(oTableModelmaterial)
oTableEntity.addComponent(oTableTransform)

oFirstLightEntity = new QEntity(oRootEntity)

        oFirstLight = new QPointLight(oFirstLightEntity)
oFirstLight.setColor(new qColor() { setRGB(128,128,128,128) })
oFirstLight.setIntensity(1)

oFirstLightTransform = new QTransform(oFirstLightEntity)
oFirstLightTransform.setTranslation(new QVector3D(5, 0, 20))

oFirstLightEntity.addComponent(oFirstLight)
oFirstLightEntity.addComponent(oFirstLightTransform)
```

(continues on next page)

(continued from previous page)

```
aCats = list(5)
for n = 1 to 5
v = n * 0.1
aCats[n] = []
aCats[n][:oCatModelEntity] = new QEntity(oRootEntity)
aCats[n][:oCatModel] = new QMesh(aCats[n][:oCatModelEntity])
aCats[n][:oCatModel].setSource(
        new qURL("file:///"+CurrentDir()+"/assets/model/Lucky_Cat.obj") )
aCats[n][:oCatModelMaterial] = new QPhongMaterial(aCats[n][:oCatModel])
aCats[n][:oCatModelMaterial].setDiffuse(
        new QColor() {setRGB(255,255,255,255)})
aCats[n][:oCatModelTransform] = new  QTransform(aCats[n][:oCatModelEntity])
aCats[n][:oCatModelTransform].setScale(0.01)
aCats[n][:oCatModelTransform].setTranslation(
        new QVector3D(-5*(v+v), 1, 15.2))
oQ = new QQuaternion(0,0,0,0)
aCats[n][:oCatModelTransform].setRotation(
        oQ.fromAxisAndAngle(
                new QVector3D(0, 1, 1), 180))
aCats[n][:oCatModelEntity].addComponent(aCats[n][:oCatModel])
aCats[n][:oCatModelEntity].addComponent(aCats[n][:oCatModelmaterial])
aCats[n][:oCatModelEntity].addComponent(aCats[n][:oCatModelTransform])
next

oSecondLightEntity = new QEntity(oRootEntity)

oSecondLight = new QPointLight(oSecondLightEntity)
oSecondLight.setColor(new qColor() { setRGB(255,255,255,255) })
oSecondLight.setIntensity(1)

oSecondLightTransform = new QTransform(oSecondLightEntity)
oSecondLightTransform.setTranslation(new QVector3D(-5, 1, 15.5))

oSecondLightEntity.addComponent(oSecondLight)
oSecondLightEntity.addComponent(oSecondLightTransform)

oRobotEntity = new QEntity(oRootEntity)

oRobotModel = new QMesh(oRobotEntity)
oRobotModel.setSource(
        new qURL("file:///"+CurrentDir()+"/assets/model/Fat_Robot.obj") )
oRobotTransform = new  QTransform(oRobotEntity)
oRobotTransform.setScale(0.006)

robotX = -15
robotY = -2
robotZ = 15

oRobotTransform.setTranslation(new QVector3D(-15, -2, 15))
oQ = new QQuaternion(0,0,0,0)
oRobotTransform.setRotation(
```

(continues on next page)

(continued from previous page)

```
                oQ.fromAxisAndAngle(new QVector3D(0, 1, 1), 170))

        oRobotMaterial = new QPhongMaterial(oRobotEntity)
        oRobotMaterial.setDiffuse(new QColor() {setRGB(128,128,128,255)})

        oRobotEntity.addComponent(oRobotModel)
        oRobotEntity.addComponent(oRobotTransform)
        oRobotEntity.addComponent(oRobotMaterial)

        oView.setRootEntity(oRootEntity)

        oWidget {
                setwindowtitle("Using Qt3D - Scene")
                showfullscreen()
        }

        oContainer.resize(oWidget.width(),oWidget.height())

                oFilter = new QAllEvents(oView)
                oFilter.setKeyPressEvent("pKeyPress()")
                oView.installeventfilter(oFilter)
        oContainer.setfocus(0)

        exec()

}

func pKeyPress

        nKey    = oFilter.getKeyCode()
        nSpeed  = 0.1
        cX      = oCameraEntity.position().x()
        CY      = oCameraEntity.position().y()
        cZ      = oCameraEntity.position().z()
        cVCx    = oCameraEntity.viewCenter().x()
        cVCy    = oCameraEntity.viewCenter().y()
        cVCz    = oCameraEntity.viewCenter().z()

        switch nKey
                on Qt_Key_Right
                        if cX < 4.8
                                robotX+= nSpeed
                                oCameraEntity.setPosition(
                                        new QVector3D(cX+0.1, cY, cZ))
                                oCameraEntity.setViewCenter(
                                        new QVector3D(cVCx+nSpeed, cVCy, cVCz))
                                oRobotTransform.setRotation(
                                        oQ.fromAxisAndAngle(new QVector3D(0, 1, 1), 170))
                        ok
                on Qt_Key_Left
                        if cX > - 13.8
                                robotX-= nSpeed
```

(continues on next page)

(continued from previous page)

```
                                oCameraEntity.setPosition(
                                  new QVector3D(cX-0.1, cY, cZ))
                                oCameraEntity.setViewCenter(
                                  new QVector3D(cVCx-nSpeed, cVCy, cVCz))
                                oRobotTransform.setRotation(
                                  oQ.fromAxisAndAngle(new QVector3D(0, 1, 1), 160))
                        ok
                on Qt_Key_Down
                        if robotY > -3.5
                                robotY-= nSpeed
                                oCameraEntity.setPosition(
                                  new QVector3D(cX, cY, cZ))
                                oRobotTransform.setRotation(
                                  oQ.fromAxisAndAngle(new QVector3D(0, 1, 1), 190))
                        ok
                on Qt_Key_Up
                        if robotY < 2
                                robotY+= nSpeed
                                oCameraEntity.setPosition(
                                  new QVector3D(cX, cY, cZ))
                                oRobotTransform.setRotation(
                                  oQ.fromAxisAndAngle(new QVector3D(0, 1, 1), 180))
                        ok
                on Qt_Key_Escape
                        oWidget.close()
        off

        oRobotTransform.setTranslation(new QVector3D(robotX, robotY, robotZ))
```

CHAPTER

FORTYEIGHT

OBJECTS LIBRARY FOR RINGQT APPLICATION

In this chapter we will learn about the objects library and using it in GUI applications.

Instead of using global variables for windows objects and connecting events to objects using the object name, the Objects Library will manage a list of the GUI objects and will provide a more natural API to quickly create one or many windows from the same class.

Also the Objects Library provide a way to quickly set methods to be executed when an event is fired. Also the library provide a natural interface to quickly use the parent or the caller windows from the child or sub windows and the other way around.

The Objects Library is designed to be used with the MVC Design Pattern.

The Objects Library uses reflection and meta-programming to add new methods to Controller classes to provide an easy way for communication between these classes.

The Objects Library is called automatically by the GUILib

Also we can call it alone by using the next command

```
load "objectslib.ring"
```

The ObjectsLib provide functions like openObject(), lastObject() and a class called ObjectsParent

The GUILib provide a new specific API for GUI applications like openWindow(), lastWindow(), etc.

48.1 Library Usage with GUI Applications

- Use the openWindow(cWindowControllerClassName) function to open new Windows
- Create at least Two Classes for each window, The Controller Class and the View Class
- Create each controller class from the WindowsControllerParent Class
- Create each view class from the WindowsViewParent Class
- Use the lastWindow() function to get the object of the last window created (The Controller object).
- When you call a sub window, use the SetParentObject() method and pass the self object.
- In the View Class, To determine the event method use the Method(cMethodName) function.
- The Method(cMethodName) function determine the method in the controller class that will be executed.
- Each controller class contains by default the CloseAction() method that you can call to close the window.
- You don't need to call the Show() Method for each window, When you use openWindow() It will be called.
- In the view class, Define the GUI window object as an attribute called win.

- You can use openWindowNoShow() to avoid displaying the window.
- You can use openWindowAndLink() to quickly get methods to access the windows.

48.2 Example

In the next example we will create two types of windows.

- Main Window contains a button. When the user click on the button a sub window will be opened.
- The User Can click on the button many times to open many sub windows.
- Each Sub Window contains Two buttons.
- The first button in the sub window change the Main and the Sub Windows Titles.
- The second button in the sub window close the Sub Window.

```
load "guilib.ring"

new qApp {
        openWindow( :MainWindowController )
        exec()
}

class MainWindowController from WindowsControllerParent
        oView = new MainWindowView
        func SubWindowAction
                openWindow( :SubWindowController )
                lastWindow().SetParentObject(self)

class MainWindowView from WindowsViewParent
        win = new qWidget() {
                SetWindowTitle("Main Window")
                btnSub = new qPushButton(win) {
                        setText("Sub Window")
                        setClickEvent( Method( :SubWindowAction ) )
                }
                resize(400,400)
        }

class SubWindowController from WindowsControllerParent
        oView = new SubWindowView
        func SetMainWindowTitleAction
                Parent().oView.win.SetWindowTitle("Message from the Sub Window")
                oView.win.SetWindowTitle("Click Event Done!")

class SubWindowView from WindowsViewParent
        win = new qWidget() {
                SetWindowTitle("Sub Window")
                btnMsg = new qPushButton(win) {
                        setText("Set Main Window Title")
                        setClickEvent( Method( :SetMainWindowTitleAction ) )
                }
                btnClose = new qPushButton(win) {
```

(continues on next page)

(continued from previous page)

```
                        Move(200,0)
                        setText("Close")
                        setClickEvent( Method( :CloseAction ) )
                }
                resize(400,400)
        }
```

The next screen shot after creating three sub windows.

The next screen shot after clicking on the button in each sub window.

48.3 openWindowAndLink() Function

We can use the openWindowAndLink() function to connect between the application windows, pass messages (call methods) between the objects.

This function uses Meta-programming to define dynamic methods in the Caller Class to use the dynamic objects of other windows that we create.

Example : (Uses the Form Designer)

First Window

(1) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/twowindowspart5/firstwindowView.ring

(2) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/twowindowspart5/firstwindowController.ring

Second Window

(1) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/twowindowspart5/secondwindowView.ring

(2) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/twowindowspart5/secondwindowController.ring

In the next code for example (from FirstWindowController.ring)

The openWindowAndLink() will create an object from the SecondWindowController Class

Then will add the SecondWindow() and IsSecondWindow() Methods to the FirstWindowController Class

Also will add the FirstWindow() and IsFirstWindow() Methods to the SecondWindowController Class

So the SendMessage() method in FirstWindowController class can use the SecondWindow() method to access the object.

This is more simple than using lastWindow(), Parent() and SetParentObject() methods.

```
class firstwindowController from windowsControllerParent

    oView = new firstwindowView

    func OpenSecondWindow
        openWindowAndLink(:SecondWindowController,self)

    func SendMessage
        if IsSecondWindow()
            SecondWindow().setMessage("Message from the first window")
        ok

    func setMessage cMessage
        oView.Label1.setText(cMessage)
```

48.4 openWindowInPackages() Function

The openWindowInPackages() function is the same as openWindow() but takes an extra list that determine the packages to import before opening the window.

Syntax:

```
openWindowInPackages(cClassName,aPackagesList)
```

Example:

The next example from the Form Designer source code, Open the Window Flags window using the openWindowInPackages() function.

We determine the class name "WindowFlagsController" and the packages name.

The Window Flags window uses the FormDesigner and System.GUI packages.

```
openWindowInPackages(:WindowFlagsController,[
        "formdesigner",
        "System.GUI"
])
```

48.5 Using ObjectID() and GetWindowByID()

Each window created using the Objects Library have a unique ID

We can get the window ID using the ObjectID() Method in the Controller Class

The Objects Library comes with the GetWindowByID() function

Using this function we can get the controller object of a window using the ID

Example:

```
load "guilib.ring"

new qApp {
        openWindow( :MainWindowController )
        exec()
}

class MainWindowController from WindowsControllerParent

        oView = new MainWindowView

        nFirstWindowID  nSecondWindowID  nThirdWindowID

                func CreateThreeWindowsAction

                        openWindow( :SubWindowController )
                        nFirstWindowID = lastWindow().ObjectID()
                        openWindow( :SubWindowController )
                        nSecondWindowID = lastWindow().ObjectID()
                        openWindow( :SubWindowController )
                        nThirdWindowID = lastWindow().ObjectID()

                        FirstWindow().oView.win {
                                setWindowTitle("One")
                                move(100,100)
                        }
                        SecondWindow().oView.win {
                                setWindowTitle("Two")
                                move(200,200)
                        }
                        ThirdWindow().oView.win {
                                setWindowTitle("Three")
                                move(300,300)
                        }

        func FirstWindow
                return GetWindowByID(nFirstWindowID)

        func SecondWindow
                return GetWindowByID(nSecondWindowID)

        func ThirdWindow
                return GetWindowByID(nThirdWindowID)
```

(continues on next page)

(continued from previous page)

```
class MainWindowView from WindowsViewParent

        win = new qWidget() {
                setWindowTitle("Main Window")
                move(500,100)
                btnSub = new qPushButton(win) {
                        setText("Create Three Windows")
                        setClickEvent( Method( :CreateThreeWindowsAction ) )
                }
                resize(400,400)
        }

class SubWindowController from WindowsControllerParent

        oView = new SubWindowView

class SubWindowView from WindowsViewParent

        win = new qWidget() {
                setWindowTitle("Sub Window")
                btnClose = new qPushButton(win) {
                        move(200,0)
                        setText("Close")
                        setClickEvent( Method( :CloseAction ) )
                }
                resize(400,400)
        }
```

Screen Shot:

48.6 Using ObjectID() and OpenWindowAndLink()

The next example demonstrates using OpenWindowAndLink() instead of GetWindowByID()

This introduce an interesting question: When to use OpenWindowAndLink()?

If the parent window will call one object of the sub window then use OpenWindowAndLink()

i.e. the relationship between the parent window and the sub window is one-to-one.

If the parent window will call many objects of the sub window at different times

Then use GetWindowByID() to determine which object to use

i.e. the relationship between the parent window and the sub window is one-to-many.

Example:

```
load "guilib.ring"

new qApp {
                openWindow( :MainWindowController )
                exec()
```

(continues on next page)

(continued from previous page)

```
}

class MainWindowController from WindowsControllerParent
                oView = new MainWindowView
                func SubWindowAction
                        openWindowAndLink( :SubWindowController,self)
                        subWindow().oView.win {
                                move(50,100)
                                setStyleSheet("background-color:yellow;")
                        }

class MainWindowView from WindowsViewParent
                win = new qWidget() {
                                setWindowTitle("Main Window")
                                resize(500,400)
                                btnSub = new qPushButton(win) {
                                                setText("Sub Window")
                                                setClickEvent( Method( :SubWindowAction␣
↪) )
                                }
                }

class SubWindowController from WindowsControllerParent
                oView = new SubWindowView
                func SetMainWindowTitleAction
                        MainWindow().oView.win.SetWindowTitle(
                                "Message from the Sub Window - Window ID = " + ObjectID()
                        )
                                oView.win.SetWindowTitle("Click Event Done!")

class SubWindowView from WindowsViewParent
                win = new qWidget() {
                                setWindowTitle("Sub Window")
                                resize(300,400)
                                btnMsg = new qPushButton(win) {
                                                setText("Set Main Window Title")
                                                setClickEvent( Method(␣
↪:SetMainWindowTitleAction ) )
                                }
                                btnClose = new qPushButton(win) {
                                                Move(200,0)
                                                setText("Close")
                                                setClickEvent( Method( :CloseAction ) )
                                }
                }
```

Screen Shot:

48.7 Objects Library Source Code

The library source code is very simple, You can check the source code files

The source code for the Objects Library (can be used without GUILib)

- https://github.com/ring-lang/ring/blob/master/libraries/objectslib/objects.ring
- https://github.com/ring-lang/ring/blob/master/libraries/objectslib/objectslib.ring

The source code for the MVC classes in GUILib

- https://github.com/ring-lang/ring/blob/master/libraries/guilib/mvc/controllerparent.ring
- https://github.com/ring-lang/ring/blob/master/libraries/guilib/mvc/viewparent.ring

CHAPTER

FORTYNINE

MULTI-LANGUAGE APPLICATIONS

There are many ways to create multi-language Ring application!

In this chapter we will learn about using the String2Constant tool

49.1 Using String2Constant

Starting from Ring 1.8 we have the String2Constant application

You will find this tool in the ring/tools/string2constant folder

Using this tool we can convert the source code to be based on constants instead of string literals

Then we can store constants in separate source code files that we can translate to different languages

Where we can have special file for each language, like (English.ring, Arabic.ring and so on)

Using this simple tool, the Form Designer is translated to Arabic language too just as an example.

49.2 Form Designer Translation

You will find the form designer application in the ring/applications/formdesigner folder

The files used for translation are stored in the ring/applications/formdesigner/translation folder

You will find two files

- Arabic.ring
- English.ring

You can check these files to get an idea about constants definition.

The next section from the English.ring file

```
T_LANGUAGE = "english"
T_LAYOUTDIRECTION = 0                    # Left to Right

T_FORMDESIGNER_FORMDESIGNER              = "Form Designer"
T_FORMDESIGNER_FORMTITLE                 = "Form1"

T_FORMDESIGNER_FILE                      = "File"
T_FORMDESIGNER_NEW                       = "New"
T_FORMDESIGNER_OPEN                      = "Open"
T_FORMDESIGNER_SAVE                      = "Save"
T_FORMDESIGNER_SAVEAS                    = "Save As"
T_FORMDESIGNER_CLOSE                     = "Close"
```

The form designer source code files will use these constants instead of typing the string literals

the next section from the formdesigner/mainwindow/formdesignerview.ring

```
# Create the Main Window and use the Mdi Area
        win = new qMainwindow() {
                setWindowTitle(T_FORMDESIGNER_FORMDESIGNER) # "Form Designer"
                setcentralWidget(this.oArea)
                setLayoutDirection(T_LAYOUTDIRECTION)
        }
```

- Using comments we can write the string literal to get more readable code.
- Using setLayoutDirection() method we can set the window direction to be Right To Left.
- Using the Load command, We can determine which translation file to use.

49.3 Forms Translation

After creating the form using the Form Designer, the View class will be generated.

We don't modify the view class, We just add the translation through the Controller class.

For example, we have the form file : ring/formdesigner/selobjects/selobjects.rform

And we add the translation through the Controller class using the next code

And we define the constants in English.ring and Arabic.ring

```
class selobjectsController from windowsControllerParent

        oView = new selobjectsView  {
                ListObjects.setselectionmode(QAbstractItemView_MultiSelection)
                win.setwindowmodality(2)
                # Translation
                        win.setWindowTitle(T_FORMDESIGNER_SELOBJECTS_TITLE)
                        win.setLayoutDirection(T_LAYOUTDIRECTION)
                        labelobjects.setText(T_FORMDESIGNER_SELOBJECTS_OBJECTS)
                        btnSelect.setText(T_FORMDESIGNER_SELOBJECTS_SELECT)
                        btnClose.setText(T_FORMDESIGNER_SELOBJECTS_CLOSE)
        }
```

CHAPTER

FIFTY

BUILDING RINGQT APPLICATIONS FOR MOBILE

In this chapter we will learn about Building RingQt Applications for Mobile.

50.1 Download Requirements

- Install Qt 5.15.2
- Install Qt Creator (Tested using Qt Creator 6.0.1)

50.2 Install Qt for Android

(1) Check the next link : http://doc.qt.io/qt-5/androidgs.html

- Run Qt Creator
- Select Tools > Options > Devices > Android
- Set the path for JDK, Android NDK and Android SDK Tools

(2) Using Qt Creator, We can download these requirements

- The Android SDK Tools (Tested using version 2.1)
- The Android NDK (Tested using version 21.3)
- Java SE Development Kit (JDK) v6 or later

(3) Update the Android SDK to get the API and tools packages required for development

- Tested using Android 7.1 (API 25)

Screen Shot:

50.3 Using Ring2EXE

We can use Ring2EXE to quickly prepare Qt project for our application

Example:

```
ring2exe myapp.ring -dist -mobileqt
```

Note: We can use the Distribute Menu in Ring Notepad

Tip: The option (Prepare Qt project for Mobile devices) in the Distribute Menu

50.4 The Qt project for your Ring application

After using Ring2EXE or the Distribute Menu in Ring Notepad

- Using the Qt Creator Open the generated Qt project

  Folder : target/mobile/qtproject

  Project file : project.pro

- Using Qt Creator, You will find the compiled Ring application in the resources (YourAppName.ringo)

  This file (Ring Object File) is generated by the Ring compiler using

```
ring YourAppName.ring -go -norun
```

- You can build your application using Qt Creator

When we open the project file, We can select the Kit

After selecting the Kit, Click Configure Project

Now We can build & Run the application

The next screen shot for the application during the runtime

(1) You can add your application images to the resources

Or You can use any text editor (Notepad) and modify : project.qrc

(2) To find images from your Ring application, You need to use the file name in resources

Example

```
if isandroid()
        mypic = new QPixmap(":/cards.jpg")
else
        mypic = new QPixmap("cards.jpg")
ok
```

50.5 Comments about developing for Android using RingQt

(1) The main project file is main.cpp

This file load Ring Compiler/Virtual Machine and RingQt

Then get the Ring Object File during the runtime from the resources

Then run the Ring Object File (ringapp.ringo) using the Ring VM

Through main.cpp you can extract more files from the resources to temp. folder once you add them (create projects with many files).

(2) The next functions are missing from this Ring edition

- Database (ODBC, SQLite & MySQL)
- Security and Internet functions (LibCurl & OpenSSL)

- RingAllegro (Allegro Library)
- RingLibSDL (LibSDL Library)

Just use Qt Classes through RingQt.

For database access use the QSqlDatabase Class

Note: All of the missing libraries (LibCurl, OpenSSL & Allegro) can be compiled for Android, but they are not included in this Qt project.

(3) use if isandroid() when you want to modify the code just for android

Example:

```
if isandroid()
        // Android code
else
        // other platforms
ok
```

(4) Sometimes you will find that the button text/image is repeated in drawing ! it's Qt problem that you can avoid using the next code.

```
if isandroid()
        setStyleSheet("
                border-style: outset;
                border-width: 2px;
                border-radius: 4px;
                border-color: black;
                padding: 6px;")
 ok
```

(5) Always use Layouts instead of manual setting of controls position and size.

This is the best way to get the expected user interface to avoid problems like (controls with small/extra size)

(6) When you deal with Qt Classes you can determine the images from resources (you don't need to copy them using main.cpp)

Example:

```
if isandroid()
    mypic = new QPixmap(":/cards.jpg")
else
    mypic = new QPixmap("cards.jpg")
ok
```

Now RingQt comes with the AppFile() function to determine the file name

Example:

```
mypic = new QPixmap(AppFile("cards.jpg"))  # Desktop or Android
```

(7) When you update your project code, You don't have to use Ring2EXE to generate the Qt project again

Just use the Distribute Menu in Ring Notepad and select (Generate Ring Object File)

Then copy the YourAppName.ringo file to target/mobile/qtproject folder and accept replacing files.

(8) If your application folder contains a Qt resource file (project.qrc)

Then when you use Ring2EXE or Ring Notepad (Distribute - Prepare Qt project for Mobile devices) the resource file will be used

See ring/applications/cards game as an example.

CHAPTER

FIFTYONE

BUILDING RINGQT APPLICATIONS FOR WEBASSEMBLY

In this chapter we will learn about Building RingQt Applications for WebAssembly.

51.1 Download Requirements

Check the next link : https://doc.qt.io/qt-5/wasm.html

Tested using

- Qt (5.15.2)
- Qt Creator (11.0.2)
- Emscripten (1.39.8) : https://emscripten.org/docs/getting_started/index.html

Use Git to have emsdk

```
# Get the emsdk repo
git clone https://github.com/emscripten-core/emsdk.git

# Enter that directory
cd emsdk
```

Use emsdk to install and activate the required version for Qt 5.15

```
emsdk install 1.39.8
emsdk activate --embedded 1.39.8
```

Check Emscripten installation

```
em++ --version
```

Output

```
emcc (Emscripten gcc/clang-like replacement) 1.39.8
(commit 24d88487f47629fac9d4acd231497a3a412bdee8)
Copyright (C) 2014 the Emscripten authors (see AUTHORS.txt)
This is free and open source software under the MIT license.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.
```

- Run Qt Creator
- Select Tools > Options > Devices > WebAssembly

51.2 Using Ring2EXE

We can use Ring2EXE to quickly prepare Qt project for our application

Example:

```
ring2exe myapp.ring -dist -webassemblyqt
```

Note: We can use the Distribute Menu in Ring Notepad

Tip: The option (Prepare Qt project for WebAssembly) in the Distribute Menu

51.3 The Qt project for your Ring application

After using Ring2EXE or the Distribute Menu in Ring Notepad

- Using the Qt Creator Open the generated Qt project

  Folder : target/webassembly/qtproject

  Project file : project.pro

- Using Qt Creator, You will find the compiled Ring application in the resources (YourAppName.ringo)

  This file (Ring Object File) is generated by the Ring compiler using

```
ring YourAppName.ring -go -norun
```

- You can build your application using Qt Creator

The next screen shot for the application during the runtime

(1) You can add your application images to the resources

Or You can use any text editor (Notepad) and modify : project.qrc

(2) To find images from your Ring application, You need to use the file name in resources

Example

```
if isWebAssembly()
        mypic = new QPixmap(":/cards.jpg")
else
        mypic = new QPixmap("cards.jpg")
ok
```

51.4 Comments about developing for WebAssembly using RingQt

(1) The main project file is main.cpp

This file load Ring Compiler/Virtual Machine and RingQt

Then get the Ring Object File during the runtime from the resources

Then run the Ring Object File (ringapp.ringo) using the Ring VM

Through main.cpp you can extract more files from the resources to temp. folder once you add them (create projects with many files).

(2) use if isWebAssembly() when you want to modify the code just for WebAssembly

Example:

```
if isWebAssembly()
        // WebAssembly code
else
        // other platforms
ok
```

(3) When you deal with Qt Classes you can determine the images from resources (you don't need to copy them using main.cpp)

Example:

```
if isWebAssembly()
    mypic = new QPixmap(":/cards.jpg")
else
    mypic = new QPixmap("cards.jpg")
ok
```

Now RingQt comes with the AppFile() function to determine the file name

Example:

```
mypic = new QPixmap(AppFile("cards.jpg"))  # Desktop, Android or WebAssembly
```

(4) When you update your project code, You don't have to use Ring2EXE to generate the Qt project again

Just use the Distribute Menu in Ring Notepad and select (Generate Ring Object File)

Then copy the YourAppName.ringo file to target/webassembly/qtproject folder and accept replacing files.

(5) If your application folder contains a Qt resource file (project.qrc)

Then when you use Ring2EXE or Ring Notepad (Distribute - Prepare Qt project for WebAssembly) the resource file will be used

See ring/applications/cards game as an example.

(6) Use stdlibcore.ring instead of stdlib.ring when using StdLib functions

(7) Use ClocksPerSecond() function instead of typing the value (1000)

(8) Nested events loops are not supported, use events for dialogs instead of calling the exec() method

(9) Using Sleep() or ProcessEvents() doesn't provide the expected results, use Qt Timers.

(10) We don't have a direct access to the File System because the applications are executed in a secure environment

Tip: We can use special functions for Uploading/Downloading files (See FileContent sample)

51.5 Dialogs

See the folder: ring/samples/UsingQtWASM

Folders:

- ColorDialog
- FontDialog
- FileDialog
- FileContent

51.6 Online Applications

- Hello World : https://ring-lang.github.io/web/helloworld/project.html
- Matching Game : https://ring-lang.github.io/web/matching/project.html
- Pairs Game : https://ring-lang.github.io/web/pairs/project.html
- Othello Game : https://ring-lang.github.io/web/othello/project.html
- Game of Life : https://ring-lang.github.io/web/gameoflife/project.html
- Form Designer : https://ring-lang.github.io/web/formdesigner/project.html

CHAPTER

FIFTYTWO

WEB DEVELOPMENT (CGI LIBRARY)

In this chapter we will learn about developing Web applications using a CGI Library written in the Ring language.

52.1 Configure the Apache web server

In this section we will learn about using Ring with the Apache HTTP Server. We can also use Ring with any web server that supports CGI.

Note: If you are developing your application using Ring Notepad then there's no need to configure Apache manually.

52.1.1 Using XAMPP Apache web server

Here we will use the XAMPP application that has Apache web server already been included. You can download XAMPP package from this link

XAMPP : https://www.apachefriends.org/download.html

Install then open the configuration file from the XAMPP application or search for it in the following locations based on your operating system.

For Windows:

```
xampp\apache\conf\httpd.conf
```

For Linux:

```
/opt/lampp/etc/httpd.conf
```

For macOS:

```
/Applications/XAMPP/xamppfiles/etc/httpd.conf
```

Search for the next line and make sure that it's not commented

```
LoadModule cgi_module modules/mod_cgi.so
```

Search for : AddHandler cgi-script

Then add ".ring" to the supported cgi extensions

Example

```
AddHandler cgi-script .cgi .ring
```

Example

```
AddHandler cgi-script .cgi .pl .asp .ring
```

Restart/Start the server using XAMPP "Manage Servers" or "Services" section

Create your web applications in a directory supported by the web server.

For Windows:

```
xampp\htdocs\mywebapplicationfolder
```

For Linux:

```
/opt/lampp/htdocs/mywebapplicationfolder
```

For macOS:

```
Open XAMPP application then go to "Volumes" section

Mount the "/opt/lampp" volume

Open Finder then you will find it mounted as an IP address under "Locations" on the left␣
↪menu
```

52.2 Setting the shebang line

For Ring files to run properly under CGI we need to set the shebang line to locate the executable file of Ring. So, Inside the source code file (*.ring), Add next line as the very first line in the file:

For Windows:

```
#!ring -cgi
```

For Linux:

```
#!/usr/bin/ring -cgi
```

For macOS:

```
#!/usr/local/bin/ring -cgi
```

Note: Change the previous line based on the path to ring executable in your machine

52.3 Grant Ring files execution permission

In Linux and Mac operating systems ring files cannot be run properly using CGI until we grant them executable permission. You can do that using the properties section of your preferred File manager or you can do it using Terminal as follow:

```
sudo chmod +x ringAppFile.ring
```

Note: If you are developing your application using Ring Notepad then this step will be done automatically.

52.4 Ring CGI Hello World Program

The next program is the Hello World program

For Windows:

```
#!ring -cgi

See "content-type: text/html" +nl+nl+
    "Hello World!" + nl
```

For Linux:

```
#!/usr/bin/ring -cgi

See "content-type: text/html" +nl+nl+
    "Hello World!" + nl
```

For macOS:

```
#!/usr/local/bin/ring -cgi

See "content-type: text/html" +nl+nl+
    "Hello World!" + nl
```

52.5 Hello World Program using the Web Library

We can use the web library to write CGI Web applications quickly

Example (1) :

```
#!ring -cgi

Load "weblib.ring"

Import System.Web

New Page
{
```

(continues on next page)

(continued from previous page)

```
        Text("Hello World!")
}
```

Example (2) :

```
#!ring -cgi

Load "weblib.ring"

Import System.Web

WebPage()
{
        Text("Hello World!")
}
```

Tip: the difference between ex. 1 and ex. 2 is using WebPage() function to return the page object instead of creating the object using new statement.

52.6 Web Library Features

The next features are provided by the Web library to quickly create web applications.

- Generate HTML pages using functions
- Generate HTML pages using objects
- HTTP Get
- HTTP Post
- Files Upload
- URL Encode
- Templates
- CRUD MVC Sample
- Users Logic & Registration Sample

52.7 HTTP Get Example

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web
New Page
{
  Title = "Test HTTP Get"
```

(continues on next page)

(continued from previous page)

```
divstart([ :style = StyleSizeFull() ] )
boxstart()
      text( "Test HTTP GET" )
      newline()
boxend()
divstart([ :style = Styledivcenter("600px","550px") +
                    StyleGradient(21) ])
divstart([:style = stylefloatleft() + stylesize("100px","100%") +
                   stylecolor("black") + stylegradient(58)])
formstart("ex5.ring")
      tablestart([ :style = stylesize("65%","90%") +
                            stylemarginleft("35%") +
                            stylemargintop("30%") ])
        rowstart([])
              cellstart([])
                text ( "Name : "  )
              cellend()
              cellstart([])
                cTextboxStyle = StyleMarginLeft("5%") +
                                StyleWidth("250px") +
                                StyleColor("black") +
                                StyleBackColor("white")
                textbox([ :name = "Name", :style = cTextboxStyle ] )
              cellend()
        rowend()
        rowstart([])
              cellstart([])
                text ( "Address : " )
              cellend()
              cellstart([])
                textbox([ :name = "Address", :style = cTextboxStyle] )
              cellend()
        rowend()
        rowstart([])
              cellstart([])
                text ( "Phone : " )
              cellend()
              cellstart([])
                textbox([ :name = "Phone", :style = cTextboxStyle ])
              cellend()
        rowend()
        rowstart([])
              cellstart([])
                text ( "Age : "  )
              cellend()
              cellstart([])
                textbox([ :name = "Age", :style = cTextboxStyle ])
              cellend()
        rowend()
        rowstart([])
              cellstart([])
                text ( "City: " )
```

(continues on next page)

(continued from previous page)

```
                cellend()
                cellstart([])
                  listbox([ :name = "City", :items = ["Cairo","Riyadh","Jeddah"],
                          :style = stylemarginleft("5%") + stylewidth("400px") ] )
                cellend()
          rowend()
          rowstart([])
                cellstart([])
                  text ( "Country : " )
                cellend()
                cellstart([])
                  combobox([ :name = "Country",
                          :items = ["Egypt","Saudi Arabia","USA"],
                          :style = stylemarginleft("5%") +
                                   stylewidth("400px")+
                                   stylecolor("black")+
                                   stylebackcolor("white")+
                                   stylefontsize("14px") ])
                cellend()
          rowend()
          rowstart([])
                cellstart([])
                  text ( "Note : " )
                cellend()
                cellstart([])
                  editbox([ :name = "Notes",
                        :style = stylemarginleft("5%") +
                                 stylesize("400px","100px")+
                                 stylecolor("black")+
                                 stylebackcolor("white") ,
                        :value = "write comments here..." ] )
                cellend()
          rowend()
          rowstart([])
                cellstart([])
                cellend()
                cellstart([])
                  submit([ :value = "Send" , :Style = stylemarginleft("5%") ])
                cellend()
          rowend()
        tableend()
  formend()
  divend()
  divend()
  divend()
}
```

Screen Shot:

The Response

```
#!ring -cgi
Load "weblib.ring"
Import System.Web
New Page
{
 divstart([ :style = styledivcenter("800px","500px") ])
  boxstart()
        text ( "HTTP GET Response" )  newline()
```

(continues on next page)

(continued from previous page)

```
  boxend()
  divstart([ :style = stylefloatleft()+stylewidth("10%")+
                      stylecolor("black")+stylegradient(58) ])
        newline()
        text ( "Name : "   )
        newline() newline()
        text ( "Address : "  )
        newline() newline()
        text ( "Phone : "   )
        newline() newline()
        text ( "Age : "   )
        newline() newline()
        text ( "City : "   )
        newline() newline()
        text ( "Country : "   )
        newline() newline()
        text ( "Note : "   )
        newline() newline()
  divend()
  divstart([ :style = stylefloatleft()+stylewidth("90%")+
                      stylecolor("black")+stylegradient(47) ])
        divstart([ :style = stylefloatleft() + stylewidth("1%") ])
          newline()
        divend()
        divstart([ :style = stylefloatleft() + stylewidth("95%") ])
          newline()
          text ( aPageVars["Name"] )
          newline() newline()
          text ( aPageVars["Address"] )
          newline() newline()
          text ( aPageVars["Phone"] )
          newline() newline()
          text (  aPageVars["Age"] )
          newline() newline()
          text ( aPageVars["City"] )
          newline() newline()
          text (aPageVars["Country"] )
          newline() newline()
          text ( aPageVars["Notes"] )
          newline() newline()
        divend()
  divend()
 divend()
}
```

Screen Shot:

52.8 HTTP POST Example

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web
New Page
{
        boxstart()
                text( "Post Test")
                newline()
        boxend()
        divstart([ :style=StyleFloatLeft()+StyleWidth("100px") ])
                newline()
                text( "Number1 : " )    newline() newline()
                text( "Number2 : " )    newline() newline()
        divend()
        formpost("ex7.ring")
                divstart([ :style = styleFloatLeft()+StyleWidth("200px") ])
                        newline()
                        textbox([ :name = "Number1" ])  newline() newline()
                        textbox([ :name = "Number2" ])  newline() newline()
                        submit([ :value = "Send" ] )
                divend()
        formend()
}
```

Screen Shot:

The Response

```
#!ring -cgi
Load "weblib.ring"
Import System.Web
New Page
{
        boxstart()
                text( "Post Result" )
                newline()
        boxend()
        divstart([ :style = styleFloatLeft()+styleWidth("200px") ])
                newline()
                text( "Number1 : " + aPageVars["Number1"] )
                newline() newline()
                text( "Number2 : " + aPageVars["Number2"] )
                newline() newline()
                text( "Sum : " + (0 + aPageVars["Number1"] + aPageVars["Number2"] ) )
                newline()
        divend()
}
```

Screen Shot:

52.9 Upload Files

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web
New page
{
        boxstart()
                text( "Upload File" )
                newline()
        boxend()
        for x = 1 to 3 newline() next
        formupload("ex9.ring")
                text( "Customer Name : " )
                textbox([ :name = "custname" ])
                newline() newline()
                divstart([ :style = styleFloatLeft() + styleWidth("90px") ])
                        uploadfile("file")  newline() newline()
                        uploadfile("file2") newline() newline()
                        submit([ :value = "Send" ])
                divend()
        formend()
}
```

Screen Shot:

The Response

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

cUploadPath = "C:/Apache2.2/htdocs/ringapp/upload/"
cUploadFolder = "/ringapp/upload/"

New page
{
        boxstart()
                text( "Upload Result" )
                newline()
        boxend()
        newline()
        divstart([ :style=  styleFloatLeft() + styleWidth("100px") ])
                text( "Name : " + aPageVars["custname"] )
                newline()
        divend()
        if aPageVars["file"] != char(13)
                getuploadedfile(self,"file")
        ok
        if aPageVars["file2"] != char(13)
                getuploadedfile(self,"file2")
        ok
}

Func getuploadedfile oObj,cFile
```

(continues on next page)

(continued from previous page)

```
        # here we use object.property
        # instead of object { } to avoid executing braceend method
        cFileName = cUploadPath + oObj.getfilename(aPageVars,cFile)
        write(cFileName,aPageVars[cFile])
        system("chmod a+x "+cFileName)
        oObj.newline()
        oObj.text( "File "+cFileName+ " Uploaded ..." )
        oObj.newline()
        imageURL = cUploadFolder + oObj.getfilename(aPageVars,cFile)
        oObj.link([ :url = imageURL, :title = "Download" ])
        oObj.newline()
        oObj.image( [ :url = imageURL , :alt = :image  ] )
        oObj.newline()
```

Screen Shot:

Test
localhost/ringapp/ex9.ring
Search
Upload Result
Name : Mahmoud
File C:/Apache2.2/htdocs/ringapp/upload/Nice.jpg Uploaded ...
Download
File C:/Apache2.2/htdocs/ringapp/upload/nice3.jpg Uploaded ...
Download

52.10 Cookies

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New page
{
        boxstart()
                text( "Cookie Test" )
                newline()
        boxend()
        newline()
        link([ :url = "ex11.ring", :title = "Use Cookies" ])
        cookie("custname","Mahmoud Fayed")
        cookie("custage",28)
}
```

Screen Shot:

The Response

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New Page
{
        boxstart()
                text( "Cookies Values" )
                newline()
        boxend()
        link([ :url = "ex10.ring", :title = "back" ])
        newline()
        divstart([:style="float:left;width:200px"])
                text( "Name : " + aPageVars["custname"] )
                newline()
                text( "Age : " + aPageVars["custage"] )
                newline()
```

(continues on next page)

(continued from previous page)

```
        divend()
}
```

Screen Shot:

52.11 URL Encode

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New Page
{
        boxstart()
                text( "URLEncode" )
                newline()
        boxend()
        link([ :url = "ex5.ring?Name="+URLEncode("-*{Mahmoud}*-")+
                      "&Address=Egypt&Phone=123456&Age=28&Notes=Programmer",
                      :title = "Test URL Encode" ])
}
```

Screen Shot:

Screen Shot:

52.12 Templates

Using Templates we can write Ring code inside HTML files

Syntax:

```
<%= Ring Expression %>
<%  Ring Statements %>
```

The HTML Code

```
<h1>Listing Numbers</h1>
<table>
  <tr>
    <th> <%= myheader.cColumn1 %> </th>
    <th> <%= myheader.cColumn2 %> </th>
    <th></th>
    <th></th>
    <th></th>
  </tr>
<% for x in aNumbers %>
  <tr>
    <td> <%= x.nValue  %> </td>
    <td> <%= x.nSquare %> </td>
  </tr>
<% next %>
</table>
```

The Ring Code

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New NumbersController { start() }

Class NumbersController

        MyHeader aNumbers

        Func Start

                MyHeader = New Header
                {
                        cColumn1 = "Number" cColumn2 = "Square"
                }

                aNumbers = list(20)

                for x = 1 to len(aNumbers)
                        aNumbers[x] = new number
                        {
                                nValue  = x   nSquare = x*x
                        }
                next

                cTemp = Template("mynumbers.html",self)

                New Page
                {
                        boxstart()
                                text( "Test Templates" )
                                newline()
                        boxend()
                        html(cTemp)
                }

Class Header cColumn1 cColumn2
Class Number nValue   nSquare
```

Screen Shot:

Test
localhost/ringapp/ex13.ring
Search
Test Templates
Listing Numbers
Number Square
1 1
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100
11 121
12 144
13 169
14 196
15 225
16 256
17 289
18 324
19 361
20 400

52.13 HTML Special Characters

The text() function display HTML special characters.

If you want to write html code, use the html() function.

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New Page
{
        boxstart()
                text("HTML Special Characters")
                newline()
        boxend()
        text('
                <html>
                        <body>
                                <p> "hello world" </p>
                        </body>
                </html>
        ')
}
```

Screen Shot:

52.14 Hash Functions

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New Page
{
        boxstart()
                text( "Hash Test")
                newline()
```

(continues on next page)

(continued from previous page)

```
        boxend()
        divstart([ :style = StyleFloatLeft() + StyleWidth("100px") ])
                newline()
                text( "Value : " )
                newline() newline()
        divend()
        formpost("ex16.ring")
                divstart([ :style = StyleFloatLeft() + StyleWidth("300px") ])
                        newline()
                        textbox([ :name = "Value" ])
                        newline() newline()
                        submit([ :value = "Send" ])
                divend()
        formend()
}
```

Screen Shot:

The Response

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New Page
{
        boxstart()
                text( "Hash Result" )
                newline()
        boxend()
        divstart([ :style = styleFloatLeft() + styleWidth("100%") ])
                newline()
                text( "Value : "  + aPageVars["Value"] )
                newline()
                text( "MD5 : "    + MD5(aPageVars["Value"]) )
                newline()
                text( "SHA1 : "   + SHA1(aPageVars["Value"]) )
```

(continues on next page)

(continued from previous page)

```
                newline()
                text( "SHA256 : " + SHA256(aPageVars["Value"]) )
                newline()
                text( "SHA224 : " + SHA224(aPageVars["Value"]) )
                newline()
                text( "SHA384 : " + SHA384(aPageVars["Value"]) )
                newline()
                text( "SHA512 : " + SHA512(aPageVars["Value"]) )
                newline()
        divend()
}
```

Screen Shot:

52.15 Random Image

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

cUploadPath = "C:/Apache2.2/htdocs/ringapp/upload/"

New Page
{
        boxstart()
                text( "Random Test")
                newline()
        boxend()
        divstart([ :style = styleFloatLeft() + styleWidth("400px") ])
                newline()
                aList = dir(cUploadPath)
                if len(aList) > 0
                        nIndex = random(len(aList))
                        if nindex = 0 nIndex = 1 ok
                        cItem = "upload/" + aList[nIndex][1]
                        newline()
                        image( [ :url = cItem , :alt = :image  ] )
```

(continues on next page)

(continued from previous page)

```
                else
                        text("No images!") newline()
                ok
        divend()
}
```

Screen Shot:

52.16 HTML Lists

The next example print a list contains numbers from 1 to 10

Then print a list from Ring List.

Finally we have a list of buttons and when we press on a button we get a message contains the clicked button number.

To start the list we uses the ulstart() function.

To end the list we uses the ulend() function.

We uses listart() and liend() to determine the list item.

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

Func Main
        New Page
        {
```

(continues on next page)

(continued from previous page)

```
                ulstart([])
                        for x = 1 to 10
                                listart([])
                                        text(x)
                                liend()
                        next
                ulend()
                list2ul(["one","two","three","four","five"])
                ulstart([])
                        for x = 1 to 10
                                listart([])
                                        cFuncName = "btn"+x+"()"
                                        button([ :onclick = cFuncName , :value = x])
                                        script(scriptfuncalert(cFuncName,string(x)))
                                liend()
                        next
                ulend()
        }
```

Screen Shot:

Test
localhost/ringapp/ex18.
Search
1
2
3
4
5
6
7
8
9
10
one
two
three
four
five
1
2
3
4
5
6
7
8
9
10
5
OK

52.17 HTML Tables

In this example we will learn how to generate HTML tables using the tablestart(), tableend(), rowstart(), rowend() ,headerstart(), headerend(), cellstart() and cellend() functions.

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

Func Main
        New Page
        {
                divstart([ :style = styledivcenter("400px","500px") ] )
                        style(styletable() + styletablerows("t01"))
                        tablestart([ :id = :t01 , :style = stylewidth("100%") ])
                                rowstart([])
                                        headerstart([]) text("Number") headerend()
                                        headerstart([]) text("square") headerend()
                                rowend()
                                for x = 1 to 10
                                        rowstart([])
                                                cellstart([]) text(x)   cellend()
                                                cellstart([]) text(x*x) cellend()
                                        rowend()
                                next
                        tableend()
                divend()
        }
```

Screen Shot:

Number	square
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

52.18 Gradient

In this example we will learn how to use the StyleGradient() function.

The function takes the style number as input (range from 1 to 60).

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

Func Main
        New Page
        {
                boxstart()
                        text("StyleGradient() Function")
                boxend()
                for x = 1 to 60
                        divstart([ :id = x , :align = "center" ,
                                   :style = stylefloatleft() +
                                            stylesize(string(100/60*6)+"%","50px") +
                                            stylegradient(x) ])
                                h3(x)
                        divend()
                next
        }
```

Screen Shot:

52.19 Generating Pages using Objects

Instead of using functions/methods to generate HTML pages, we can use an object for each element in the page.

This choice means more beautiful code but slower.

The fastest method is to print HTML code directly, then using functions then using templates then using objects (slower).

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

Func Main

  WebPage()
  {
        Title = "Using objects to create the Web Page content"
        h1 { text("welcome") }
        link
        {
          Title = "Google"
          Link  = "http://www.google.com"
        }
        div
        {
          id = "div1"
          style = stylegradient(30) + stylesize("50%","50%")
          text("Outer Div")
          div
          {
                id = "div2"
                color = "white"
                backgroundcolor = "green"
                width = "50%"
                height = "50%"
                marginleft = "5%"
                margintop = "5%"
                text("Inner Div")
          }
        }
        div
        {
          id = "div3"
          color = "black"
          backgroundcolor = "silver"
          width = "100%"
          height = "100%"
          text("Form")
          form
          {
                method = "POST"
                Action = "helloworld.ring"
                Table
                {
```

(continues on next page)

(continued from previous page)

```
                style = stylewidth("100%") + stylegradient(24)
                TR
                {
                      TD { WIDTH="10%" text("Name : " )  }
                      TD { Input { type = "text" } }
                }
                TR
                {
                      TD { WIDTH="10%" text("Email : " ) }
                      TD { Input { type = "text" } }
                }
                TR
                {
                      TD { WIDTH="10%" text("Password : " ) }
                      TD { Input { type = "password" } }
                }
                TR
                {

                      TD { WIDTH="10%" text("Notes") }
                      TD { TextArea { width="100%" rows = 10  cols = 10
                                      text("type text here...") } }
                }
                TR
                {
                      TD { WIDTH="10%" text("Gender") }
                      TD {
                        select
                         {
                               width = "100%"
                               option { text("Male") }
                               option { text("Female") }
                         }
                      }
                }
                TR
                {
                      TD {   WIDTH="10%" text("Role") }
                      TD
                      {
                        select
                        {
                               multiple = "multiple"
                               width    = "100%"
                               option { text("student") }
                               option { text("admin") }
                         }
                      }
                }
              }
              Input { type = "submit" value = "send" }
              Image { src="upload/profile1.jpg" alt="profile"}
```

(continues on next page)

(continued from previous page)

```
                Input { type = "checkbox" value = "Old Member"} text("old member")
                Input { type = "range" min=1 max=100}
                Input { type = "number" min=1 max=100}
                Input { type = "radio" color="black" name="one"
                        value = "one"} text("one")
        }
      }
      div
      {
        color = "white"
        backgroundcolor = "blue"
        width = "100%"
        UL
        {
              LI { TEXT("ONE") }
              LI { TEXT("TWO") }
              LI { TEXT("THREE") }
        }
      }
      div
      {
        audio
        {
              src = "horse.ogg"
              type = "audio/ogg"
        }

        video
        {
               width = 320
               height = 240
               src = "movie.mp4"
               type = "video/mp4"
        }

        Input
        {
              type = "color"
              value = "#ff0000"
              onchange = "clickColor(0, -1, -1, 5)"
        }
      }
}
```

Screen Shot:

Using objects to create the We...
localhost/ringapp/ex21.ring
Search
welcome
Google
Outer Div
Inner Div
Form
Name :
Email :
Password :
type text here...
Notes

52.20 HtmlPage Class

Using this class we can create HTML documents without printing the output to the standard output

So instead of using the WebLib in Web Applications only

We can use it in Console/GUI/Mobile Applications too

Example:

```
load "stdlib.ring"
load "weblib.ring"

import System.Web
```

(continues on next page)

(continued from previous page)

```
func main

        mypage = new HtmlPage {
                h1 { text("Customers Report") }
                Table
                {
                        style = stylewidth("100%") + stylegradient(4)
                        TR
                        {
                              TD { WIDTH="10%" text("Customers Count : " )  }
                              TD { text (100) }
                        }
                }

                Table
                {
                        style = stylewidth("100%") + stylegradient(26)
                        TR
                        {
                              style = stylewidth("100%") + stylegradient(24)
                              TD { text("Name " )  }
                              TD { text("Age" ) }
                              TD { text("Country" ) }
                              TD { text("Job" ) }
                              TD { text("Company" ) }
                        }
                        for x =  1 to 100
                              TR
                              {
                                      TD { text("Test" )  }
                                      TD { text("30" ) }
                                      TD { text("Egypt" ) }
                                      TD { text("Sales" ) }
                                      TD { text("Future" ) }
                              }
                        next
                }

        }

        write("report.html",mypage.output())
```

52.21 Using Bootstrap Library using Functions

The next example uses the Bootstrap JavaScript Library when generating the HTML page.

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

Func Main
  new BootstrapPage {
        divstart([ :class = "container" ])
           divstart([ :class = "jumbotron"  ])
                h1("Bootstrap Page")
          divend()
          divstart([ :class = :row ])
                divstart([ :class = "col-sm-4" ])
                  h3("Welcome to the Ring programming language")
                  p([ :text = "Using a scripting language is very fun!" ])
                divend()
                divstart([ :class = "col-sm-4" ])
                  h3("Welcome to the Ring programming language")
                  p([ :text = "using a scripting language is very fun!" ])
                divend()
                divstart([ :class = "col-sm-4" ])
                  h3("Welcome to the Ring programming language")
                  p([ :text = "using a scripting language is very fun!" ])
                divend()
          divend()
        divend()
  }
```

Screen Shot:

52.22 Using Bootstrap Library using Objects

The next example uses the Bootstrap JavaScript Library when generating the HTML page.

Instead of using functions to generate the HTML elements, we will use objects.

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

Func Main
  BootStrapWebPage()
  {
        div
        {
          classname = :container
          div
          {
                classname = :jumbotron
                H1 {   text("Bootstrap Page")   }
          }
          div
          {
                classname = :row
                for x = 1 to 3
                  div
```

(continues on next page)

(continued from previous page)

```
                {
                        classname = "col-sm-4"
                        H3 { html("Welcome to the Ring programming language") }
                        P  { html("Using a scripting language is very fun!") }
                }
                next
        }
        div
        {
                classname = :row
                div
                {
                  classname = "col-sm-4"
                  Button
                  {
                        classname = "btn btn-info btn-lg"
                        datatoggle= "modal"
                        datatarget = "#myModal"
                        text("Open Large Modal")
                  }
                }
                div
                {
                  classname = "col-sm-4"
                  Button {  classname = "btn btn-default btn-lg" text("default") }
                  Button {  classname = "btn btn-primary btn-md" text("primary") }
                  Button {  classname = "btn btn-success btn-sm" text("success") }
                  Button {  classname = "btn btn-info btn-xs"    text("info") }
                  Button {  classname = "btn btn-warning"        text("warning") }
                  Button {  classname = "btn btn-danger"         text("danger") }
                  Button {  classname = "btn btn-link"           text("link") }
                }
                div
                {
                  classname = "col-sm-4"
                  Button {  classname = "btn btn-default btn-block" text("default") }
                  Button {  classname = "btn btn-primary btn-block" text("primary") }
                  Button {  classname = "btn btn-success btn-block" text("success") }
                  Button {  classname = "btn btn-info btn-block"    text("info") }
                  Button {  classname = "btn btn-warning btn-block" text("warning") }
                  Button {  classname = "btn btn-danger btn-block"  text("danger") }
                  Button {  classname = "btn btn-link btn-block"    text("link") }
                }
                div
                {
                  classname = "col-sm-4"
                  div { classname = "btn-group"
                        button {  classname="btn btn-primary" text("one") }
                        button {  classname="btn btn-primary" text("two") }
                        button {  classname="btn btn-primary" text("three") }
                  }
                }
```

(continues on next page)

(continued from previous page)

```
                div
                {
                  classname = "col-sm-4"
                  div { classname = "btn-group btn-group-lg"
                        button {  classname="btn btn-primary" text("one") }
                        button {  classname="btn btn-primary" text("two") }
                        button {  classname="btn btn-primary" text("three") }
                  }
                }
                div
                {
                  classname = "col-sm-4"
                  div {
                        classname = "btn-group-vertical btn-group-lg"
                        button {  classname="btn btn-primary" text("one") }
                        button {  classname="btn btn-primary" text("two") }
                        button {  classname="btn btn-primary" text("three") }
                  }
                }
          }
          div { classname="modal fade" id="myModal" role="dialog"
                div { classname = "modal-dialog modal-lg"
                  div { classname="modal-content"
                        div { classname="modal-header"
                          button {  classname="close" datadismiss="modal"
                                html("&times")
                          }
                          h4 {  classname="modal-title"
                                text("Modal Header")
                          }
                        }
                        div { classname = "modal-body"
                          p { text("This is a large model.") }
                        }
                        div { classname="modal-footer"
                          button { classname = "btn btn-default" datadismiss="modal"
                                text("close")
                          }
                        }
                  }
                }
          }
        }
  }
```

Screen Shot:

52.23 CRUD Example using MVC

The next example uses the weblib.ring & datalib.ring.

The datalib.ring contains classes for creating database applications using MVC pattern.

In this example we create an object from the SalaryController class then call the Routing method.

We define the website variable to contains the basic url of the page.

When we create the SalaryModel class from the ModelBase class, the salary table will be opened and the columns data will be defined as attributes in the model class.

The SalaryView class create an object from the SalaryLanguageEnglish class to be used for translation.

The method AddFuncScript is used to call the form for adding/modifying record data.

The method FormViewContent is used to determine the controls in the form when we add or modify a record.

```
#!ring -cgi
Load "weblib.ring"
Load "datalib.ring"
Import System.Web

website = "ex24.ring"

New SalaryController { Routing() }

Class SalaryModel from ModelBase
```

(continues on next page)

(continued from previous page)

```
Class SalaryController From ControllerBase

Class SalaryView From ViewBase

  oLanguage = new SalaryLanguageEnglish

  Func AddFuncScript oPage,oController
        return   oPage.scriptfuncajax("myadd",oController.cMainURL+
                 oController.cOperation+"=add","mysubpage")

  Func FormViewContent oController,oTranslation,oPage
        return [
                         [ oTranslation.aColumnsTitles[2], "textbox", "name",
                           oController.oModel.Name, oPage.stylewidth("100%")    ],
                         [ oTranslation.aColumnsTitles[3], "textbox", "salary",
                           oController.oModel.Salary, oPage.stylewidth("50%") ]
                   ]

Class SalaryLanguageEnglish
  cTitle = "Salary Table"
  cBack = "back"
  aColumnsTitles = ["ID","Name","Salary"]
  cOptions = "Options"
  cSearch = "Search"
  comboitems = ["Select Option...","Edit","Delete"]
  cAddRecord = "Add Record"
  cEditRecord = "Edit Record"
  cRecordDeleted = "Record Deleted!"
  aMovePages = ["First","Prev","Next","Last"]
  cPage = "Page"
  cOf = "of"
  cRecordsCount = "Records Count"
  cSave = "Save"
  temp = new page
  cTextAlign = temp.StyleTextRight()
  cNoRecords = "No records!"
```

Screen Shot:

Test
localhost/ringapp/ex24.ring
Search
Salary Table
back
Name :
Search
ID
Name
Salary
Options
1
Mahmoud
15000
Select Option...
2
Samir
16000
Select Option...
4
Ahmed
50000
Select Option...
5
Ibrahim
50000
Select Option...
12
Mohammed
56786
Select Option...
First Prev Next Last
Records Count (15) : Page 1 of 3
Add Record

Test
localhost/ringapp/ex24.ring?part=1&searchname=m
Search
Salary Table
back
Name : m
Search
ID
Name
Salary
Options
1
Mahmoud
15000
Select Option...
12
Mohammed
56786
Select Option...
Edit
131
Mageed
23623
Delete
First Prev Next Last
Records Count (3) : Page 1 of 1
Add Record
Salary Table - Edit Record
x
Name : Mahmoud
Salary : 15000
Save

52.24 Users registration and Login

We have the users classes (Model, View & Controller) to deal with the users data like username & email.

The next code is stored in ex25_users.ring

```
Class UsersModel from ModelBase
  cSearchColumn = "username"

Class UsersController From ControllerBase
  aColumnsNames = ["id","username","email"]

  Func UpdateRecord
        oModel.id = aPageVars[cRecID]
        oModel.updatecolumn("username", aPageVars[:username] )
        oModel.updatecolumn("email", aPageVars[:email] )
        oView.UpdateView(self)

Class UsersView from ViewBase

  oLanguage = new UsersLanguageEnglish

  Func AddFuncScript oPage,oController
        return   oPage.scriptfunc("myadd",oPage.scriptredirection("ex26.ring"))

  Func FormViewContent oController,oTranslation,oPage
        return [
                        [oTranslation.aColumnsTitles[2],"textbox","username",
                        oController.oModel.UserName,oPage.stylewidth("100%")],
                        [oTranslation.aColumnsTitles[3],"textbox","email",
                        oController.oModel.Email,oPage.stylewidth("50%")]
                    ]

Class UsersLanguageEnglish
  cTitle = "Users Table"
  cBack = "back"
  aColumnsTitles = ["ID","User Name","Email"]
  cOptions = "Options"
  cSearch = "Search"
  comboitems = ["Select Option...","Edit","Delete"]
  cAddRecord = "Add Record"
  cEditRecord = "Edit Record"
  cRecordDeleted = "Record Deleted!"
  aMovePages = ["First","Prev","Next","Last"]
  cPage = "Page"
  cOf = "of"
  cRecordsCount = "Records Count"
  cSave = "Save"
  temp = new page
  cTextAlign = temp.StyleTextRight()
  cNoRecords = "No records!"
```

In the file ex25.ring we load ex25_users.ring then create an object from UsersController class.

Using the created object, we call the routing method.

```
#!ring -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

Import System.Web
website = "ex25.ring"
New UsersController { Routing() }
```

Screen Shot:

See the next code for the registration page

```
#!ring -cgi
Load "weblib.ring"
Load "datalib.ring"
Import System.Web

website = "ex26.ring"

new page {
  boxstart()
        text( "Register")
        newline()
  boxend()
  divstart([:style = stylegradient(6) + stylesize("100%","95%") ])
  link([ :url = website, :title = "back" , :style = stylecolor("white")])
  newline()
  divstart([ :style= styledivcenter("500","160") + stylegradient(52) ])
  formpost("ex27.ring")
        tablestart([ :Style =    stylemarginleft("2%") + stylemargintop("2%") +
```

(continues on next page)

(continued from previous page)

```
                                stylewidth("90%") ])
          rowstart([])
                cellstart([:style = stylewidth("20%") + styleheight(30)])
                  text("User Name")
                cellend()
                cellstart([ :style = stylewidth("80%")  ])
                  textbox([:name = "username", :style = stylewidth("100%")])
                cellend()
          rowend()
          rowstart([])
                cellstart([ :Style = styleheight(30)])
                  text("Password")
                cellend()
                cellstart([])
                  textbox([:name = "password" , :type = "password"])
                cellend()
          rowend()
          rowstart([])
                cellstart([ :style = styleheight(30)])
                  text("Email")
                cellend()
                cellstart([])
                  textbox([:name = "email" , :style = stylewidth("100%")])
                cellend()
          rowend()
          rowstart([])
                cellstart([ :style = styleheight(30)])
                cellend()
                cellstart([ :style = styleheight(30)])
                  submit([:value = "Register"  ])
                cellend()
          rowend()
        tableend()
  formend()
  divend()
  divend()
}
```

Screen Shot:

The Registration response

```
#!ring -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

Import System.Web

oUser = new UsersModel
oUser.Connect()
if oUser.findwith("username",aPageVars["username"])
        new page {
                text("The user name is already registered")
        }
        return
ok
if oUser.findwith("email",aPageVars["email"])
        new page {
                text("This email is already registered")
        }
        return
ok

aPageVars["salt"] = str2hex(RandBytes(32))
aPageVars["pwhash"] = sha256(aPagevars["password"]+aPageVars["salt"])
aPageVars["sessionid"] = str2hex(randbytes(32))
oUser.Insert()
```

(continues on next page)

(continued from previous page)

```
new page {
        cookie("sessionid",aPageVars["sessionid"])
        text("New User Created!")
        newline()
        text("User Name : " + aPageVars["username"])
        newline()
}
oUser.Disconnect()
```

See the next code for the Login page

```
#!ring -cgi
Load "weblib.ring"
Load "datalib.ring"

Import System.Web

website = "ex28.ring"

new page {
  boxstart()
        text( "Login")
        newline()
  boxend()
  divstart([:style = stylegradient(6) + stylesize("100%","95%") ])
  link([ :url = website, :title = "back" , :style = stylecolor("white")])
  newline()
  divstart([ :style= styledivcenter("500","130") + stylegradient(52) ])
  formpost("ex29.ring")
        tablestart([ :Style =   stylemarginleft("2%") + stylemargintop("2%") +
                                stylewidth("90%") ])
          rowstart([])
                cellstart([:style = stylewidth("20%") + styleheight(30)])
                  text("User Name")
                cellend()
                cellstart([ :style = stylewidth("80%") ])
                  textbox([:name = "username", :style = stylewidth("100%")])
                cellend()
          rowend()
          rowstart([])
                cellstart([ :style = styleheight(30)])
                  text("Password")
                cellend()
                cellstart([])
                  textbox([:name = "password" , :type = "password"])
                cellend()
          rowend()
          rowstart([])
                cellstart([ :style = styleheight(30) ])
                cellend()
                cellstart([])
                  submit([:value = "Login"  ])
```

(continues on next page)

(continued from previous page)

```
                cellend()
            rowend()
        tableend()
  formend()
  divend()
  divend()
}
```

Screen Shot:

The response page

```
#!ring -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

Import System.Web

oUser = new UsersModel
oUser.Connect()
lResult = oUser.FindWith("username",aPageVars["username"])
new page {
        if lResult
                if sha256(aPagevars["password"]+oUser.Salt) = oUser.pwhash
                        text ("Correct Password!")
                        aPageVars["sessionid"] = str2hex(randbytes(32))
```

(continues on next page)

(continued from previous page)

```
                        oUser.UpdateColumn("sessionid",aPageVars["sessionid"])
                        cookie("sessionid",aPageVars["sessionid"])
                else
                        text ("Bad password!")
                ok
        else
                text("Bad User Name!")
        ok
}
oUser.Disconnect()
```

The next code for checking if the user needs to login or not

```
#!ring -cgi
Load "weblib.ring"
Load "datalib.ring"
Load "ex25_users.ring"

Import System.Web

oUser = new UsersModel
oUser.Connect()
lResult = oUser.FindWith("sessionid",aPageVars["sessionid"])
new page {
        if lResult
                text("User Name : " + oUser.username )
        else
                text("Please Login First!")
        ok
}
oUser.Disconnect()
```

52.25 Database, ModelBase & ControllerBase classes

In this section we will see some code from datalib.ring

The next code presents the Database, ModelBase & ControllerBase classes

```
Import System.Web

Class Database

  cServer = "localhost"
  cUserName = "root"
  cPassword = "root"
  cDatabase = "mahdb"

  Func Connect

        con = mysql_init()
        mysql_connect(con, cServer, cUserName, cPassWord,cDatabase)
```

(continues on next page)

(continued from previous page)

```
  Func Disconnect

        mysql_close(con)

  Func Query cQuery

        mysql_query(con,cQuery)

  Func QueryResult

        return mysql_result(con)

  Func QueryResultWithColumns
        # return columns names + query result
        return mysql_result2(con)

  Func QueryValue
        aResult = mysql_result(con)
        if islist(aResult) and len(aResult) >= 1
          aResult = aResult[1]
          if len(aResult) >= 1
                return aResult[1]
          ok
        ok
        return 0

  Func EscapeString x
        if isstring(x)
          return MySQL_Escape_String(con,x)
        else
          return MySQL_Escape_String(con,string(x))
        ok

  Private
        con = NULL

Class ModelBase from Database

  cTableName = ""
  cSearchColumn = "name"
  aColumns = []
  aQueryResult = []
  ID = 0

  # set table name from class name
  classname = lower(classname(self))
  if right(classname,5) = :model
        cTablename = left(classname,len(classname)-5)
  ok

  Func Insert
```

(continues on next page)

(continued from previous page)

```
        cValues = ""
        for x in aColumns
          cValues += "'" + EscapeString(aPageVars[x]) + "',"
        Next
        cValues = left(cValues,len(cValues)-1)  # remove last comma
        cColumns = ""
        for x in aColumns
          cColumns += x + ","
        next
        cColumns = left(cColumns,len(cColumns)-1)
        query("insert into " + cTableName + "("+cColumns+") values (" +
                 cValues + ")" )

  Func Update nID

        cStr = ""
        for x in aColumns
          cStr += x + " = '" + EscapeString(aPageVars[x]) + "' , "
        # the space after comma is necessary
        Next
        cStr = left(cStr,len(cStr)-2)
        query("update " + cTableName + " set " + cStr + " where id = " + nID )

  Func UpdateColumn cColumn,cValue
        query("update " + cTableName + " set " + cColumn + " = '" +
        EscapeString(cValue) + "' where id = " + self.ID )


  Func Count cValue

        query("SELECT count(*) FROM " + cTableName +
                 " where "+cSearchColumn+" like '" + EscapeString(cValue) + "%'")
        return queryValue()

  Func Read nStart,nRecordsPerPage

        query("SELECT * FROM "+ cTableName+" limit " + EscapeString(nStart) + "," +
        EscapeString(nRecordsPerPage) )
        aQueryResult = queryResult()

  Func Search cValue,nStart,nRecordsPerPage

        query("SELECT * FROM "+ cTableName+" where "+cSearchColumn+" like '" +
        EscapeString(cValue) + "%'" +
          " limit " + EscapeString(nStart) + "," + EscapeString(nRecordsPerPage) )
        aQueryResult = queryResult()

  Func Find nID

        query("select * from " + cTableName + " where id = " + EscapeString(nID) )
        aResult = queryResult()[1]
```

(continues on next page)

(continued from previous page)

```
        # move the result from the array to the object attributes
        ID = nID
        cCode = ""
        for x = 2 to len(aResult)
          cCode += aColumns[x-1] + " = hex2str('" + str2hex(aResult[x]) + "')" + nl
        next
        eval(cCode)

  Func FindWith cColumn,cValue

        query("select * from " + cTableName + " where "+cColumn+" = '" +
        EscapeString(cValue) + "'" )
        aResult = queryResult()
        if len(aResult) > 0
          aResult = aResult[1]
        else
          return 0
        ok
        # move the result from the array to the object attributes
        ID = aResult[1]
        cCode = ""
        for x = 2 to len(aResult)
          cCode += aColumns[x-1] + " = hex2str('" + str2hex(aResult[x]) + "')" + nl
        next
        eval(cCode)
        return 1

  Func Delete ID

        query("delete from " + cTableName + " where id = " + EscapeString(ID) )

  Func Clear

        cCode = ""
        for x in aColumns
          cCode += x + ' = """' + nl
        next
        eval(cCode)

  Func LoadModel

        # create the columns array
        query("SELECT * FROM "+ cTableName + " limit 0,1")
        aQueryResult = QueryResultWithColumns()[1]
        for x = 2 to len(aQueryResult)
          aColumns + lower(trim(aQueryResult[x]))
        next

        # create attribute for each column
        for x in aColumns
          addattribute(self,x)
        next
```

(continues on next page)

(continued from previous page)

```
  Func Connect

        Super.Connect()
        if nLoadModel = 0
          nLoadModel = 1
          LoadModel()
        ok

  private

        nLoadModel = 0


Class ControllerBase

  nRecordsPerPage = 5
  nRecordsCount = 0
  nPagesCount = 0
  nActivePage = 0

  # Dynamic creation of oView = new tablenameView and oModel = new tablename.Model
  classname = lower(classname(self))
  if right(classname,10)  = :controller
        tablename = left(classname,len(classname)-10)
        cCode = "oView = new " + tablename+"View" + nl
        cCode += "oModel = new " + tablename+"Model" + nl
        eval(cCode)
        oModel.connect()
  ok

  cSearchName = "searchname"
  cPart = "part"
  cPageError = "The page number is not correct"
  cLast = "last"
  cOperation = "operation"
  cRecID = "recid"

  aColumnsNames = ["id"]
  for t in oModel.aColumns
        aColumnsNames + t
  next

  cMainURL = website + "?"

  func Routing

        switch      aPageVars[cOperation]
        on NULL     showtable()
        on :add     addrecord()
        on :save    saverecord()
        on :delete  deleterecord()
```

(continues on next page)

(continued from previous page)

```
        on :edit    editrecord()
        on :update  updaterecord()
        off

  func ShowTable

        nRecordsCount = oModel.Count( aPageVars[cSearchName] )

        nPagesCount = ceil(nRecordsCount / nRecordsPerPage)

        if aPageVars[cPart] = cLast
          aPageVars[cPart] = string(nPagesCount)
        ok

        nActivePage = number(aPageVars[cPart])
        if nActivePage = 0 nActivePage = 1 ok

        if ( nActivePage > nPagesCount ) and nRecordsCount > 0
          ErrorMsg(cPageError)
          return
        ok

        nStart = (nActivePage-1)*nRecordsPerPage

        if aPageVars[cSearchName] = NULL
          oModel.Read( nStart,nRecordsPerPage )
        else
          oModel.Search( aPageVars[cSearchName],nStart,nRecordsPerPage )
        ok

        oView.GridView(self)

  func AddRecord

        oModel.clear()
        oView.FormViewAdd(Self,:save,false) # false mean don't include record id

  func SaveRecord

        oModel.Insert()
        oView.SaveView(self)

  func EditRecord

        oModel.Find( aPageVars[cRecID] )
        oView.FormViewEdit(Self,:update,true) # true mean include record id

  func UpdateRecord

        oModel.update( aPageVars[cRecID] )
        oView.UpdateView(self)
```

(continues on next page)

(continued from previous page)

```
func DeleteRecord

      oModel.Delete( aPageVars[cRecID] )
      oView.DeleteView()

func braceend

      oModel.Disconnect()
```

52.26 WebLib API

In this section we will see the web library functions, classes and methods.

Function	Parameters	Description
LoadVars	None	Save the request parameters and cookies to aPageVars List
WebPage	None	Create new object from the WebPage Class
BootStrapWebPage	None	Create new object from the BootStrapWebPage Class
HTMLSpecialChars	cString	Encode Special characters to HTML equivalent
Template	cFile,oObject	Execute Ring Code in cFile after accessing oObject using {}
Alert	cMessage	Generate HTML Web Page that display cMessage using JavaScript Alert()
HTML2PDF	cString	Generate and Display PDF File from HTML String (cString)

The Package System.Web contains the next classes

Class Name	Description
Application	Contains methods for Encoding, Decoding, Cookies & More.
Page	Contains methods to generate HTML pages.
ScriptFunctions	Contains methods to generate some JavaScript Functions.
StyleFunctions	Contains methods to generate CSS.
PageBuffer	Generate HTML Page in memory (don't print the output).
HTML2PDF	Generate PDF File from HTML code.
BootStrapPage	Using BootStrap Library.
WebPage	Generate page using objects for each element.
HtmlPage	Like WebPage but doesn't print the output to stdout.
BootStrapWebPage	Generate page using objects, using BootStrap Library.
ObjsBase	Parent Class for page objects.
NewObjectsFunctions	Methods to create new objects in the page or element.
H1	Wraps HTML H1.
H2	Wraps HTML H2.
H3	Wraps HTML H3.
H4	Wraps HTML H4.
H5	Wraps HTML H5.
H6	Wraps HTML H6.
P	Wraps HTML P.
Link	Wraps HTML link.
NewLine	Wraps HTML NewLine.
Div	Wraps HTML Div.

continues on next page

Table 1 – continued from previous page

Class Name	Description
Form	Wraps HTML Form.
Input	Wraps HTML Input.
TextArea	Wraps HTML TextArea.
Select	Wraps HTML Select.
Option	Wraps HTML Option.
Image	Wraps HTML Image.
UL	Wraps HTML UL.
LI	Wraps HTML LI.
Table	Wraps HTML Table.
TR	Wraps HTML TR.
TD	Wraps HTML TD.
TH	Wraps HTML TH.
Audio	Wraps HTML Audio.
Video	Wraps HTML Video.
Nav	Wraps HTML Nav.
Span	Wraps HTML Span.
Button	Wraps HTML Button.

52.27 Application Class

Method	Parameters	Description
DecodeString	cString	Decode request parameters
Decode	cString	Decode multipart/form-data
GetFileName	aArray,cVar	Get File Name in aArray using cVar
SetCookie	name,value,expires,path,domain,secure	Set Cookie
Cookie	name,value	Set Cookie using name and value only
GetCookies	None	Get Cookies
URLEncode	cString	URL Encode
ScriptLibs	None	Add JavaScript Libraries like BootStrap
Print	None	Print Page Content
Style	cStyle	Add cStyle to page CSS content
StartHTML	None	Add HTTP Header to page content
Redirect	cLocation	Will redirect the webpage
NoJavaScript	None	Avoid JavaScript links

The method DecodeString is used to get HTTP request parameters.

The methods Decode and GetFileName are used for uploading files.

The methods SetCookie, Cookie & GetCookies are used for adding and reading cookies.

The methods StartHTML, ScriptsLibs, Style & Print are used for page structure and JS/CSS support.

The method URLEncode is used to encode a URL to be used in HTML pages.

52.28 Page Class

Method	Parameters	Description
text	x	add HTMLSpecialChars(x) to page content (accept strings and numbers)
html	cString	add html code to page content
h1	x	add x to page content between <h1> and </h1>
h2	x	add x to page content between <h2> and </h2>
h3	x	add x to page content between <h3> and </h3>
h4	x	add x to page content between <h4> and </h4>
h5	x	add x to page content between <h5> and </h5>
h6	x	add x to page content between <h6> and </h6>
p	aPara	HTML , uses aPara List as Hash to get attributes
NewLine	None	add to page content
AddAttributes	aPara	Convert aPara list as hash to HTML element attributes
Link	aPara	HTML <a href> and </a>, uses aPara List as Hash to get attributes
Image	aPara	HTML <img>, uses aPara List as Hash to get attributes
Button	aPara	HTML <input type="button">, uses aPara List as Hash to get attributes
ButtonLink	aPara	HTML <input type="button">, uses link attribute to navigate to link
Textbox	aPara	HTML <input type="text">, uses aPara List as Hash to get attributes
Editbox	aPara	HTML <textarea> and </textarea>, uses aPara to get attributes
Combobox	aPara	HTML <select>, uses items attribute as list for <option>
Listbox	aPara	HTML <select multiple='multiple'>, uses items attribute for <option>
ulstart	aPara	HTML <ul>
ulend	aPara	HTML </ul>
listart	aPara	HTML <li>
liend	aPara	HTML </li>
List2UL	aList	Generate HTML <ul> including items from Ring List items
DivStart	aPara	HTML , uses aPara List as Hash to get attributes
NavStart	aPara	HTML <nav>, uses aPara List as Hash to get attributes
SpanStart	aPara	HTML , uses aPara List as Hash to get attributes
BoxStart	None	Generate Div with black background to be used as page header
DivEnd	None	HTML
NavEnd	None	HTML </nav>
SpanEnd	None	HTML
BoxEnd	None	HTML , the same as divend()
FormStart	cAction	HTML <form>, with cAction as the action attribute or an empty value
FormPost	cAction	HTML <form method="post"> , with cAction as the action attribute
FormEnd	None	HTML </form>
Submit	aPara	HTML <input type="submit">
Hidden	cName,cValue	HTML <input type="hidden">
FormUpload	x	HTML Form, method="post" enctype="multipart/form-data" and x = action
UploadFile	x	HTML <input type="file"> and name = x
Video	aPara	HTML <video>
Audio	aPara	HTML <audio>
GetColor	aPara	Select Color
Radio	aPara	HTML <input type="radio">
Checkbox	aPara	HTML <input type="checkbox">
Spinner	aPara	HTML <input type="number">
Slider	aPara	HTML <input type="range">
TableStart	aPara	HTML <table>
TableEnd	None	HTML </table>

continues on next page

Table 2 – continued from previous page

Method	Parameters	Description
RowStart	aPara	HTML <tr>
RowEnd	None	HTML </tr>
CellStart	aPara	HTML <td>
CellEnd	None	HTML </td>
HeaderStart	aPara	HTML <th>
HeaderEnd	None	HTML </th>
theadStart	aPara	HTML <thead>
theadEnd	None	HTML </thead>
tbodyStart	aPara	HTML <tbody>
tbodyEnd	None	HTML </tbody>
tfootStart	aPara	HTML <tfoot>
tfootEnd	None	HTML </tfoot>

aPara in the page methods is a list contains attributes and values. Using aPara we can set values for the next attributes

```
classname id name align style dir value onclick oncontextmenu ondblclick
onmousedown onmouseenter onmouseleave onmousemove onmouseover onmouseout
onmouseup onkeydown onkeypress onkeyup onabort onbeforeunload onerror
onhashchange onload onpageshow onpagehide onresize onscroll onunload
onblur onchange onfocus onfocusin onfocusout oninput oninvalid onreset
onsearch onselect onsubmit ondrag ondragend ondragenter ondragleave
ondragover ondragstart ondrop oncopy oncut onpaste onafterprint
onbeforeprint oncanplay oncanplaythrough ondurationchange onemptied
onended onloadeddata onloadedmetadata onloadstart onpause onplay
onplaying onprogress onratechange onseeked onseeking onstalled onsuspend
ontimeupdate onvolumechange onwaiting animationend animationiteration
animationstart transitionend onmessage onopen onmousewheel ononline
onoffline onpostate onshow onstorage ontoggle onwheel ontouchcancel
ontouchend ontouchmove ontouchstart color opacity background backgroundattachment
backgroundcolor backgroundimage backgroundposition backgroundrepeat backgroundclip
backgroundorigin backgroundsize border borderbottom borderbottomcolor
borderbottomleftradius borderbottomrightradius borderbottomstyle borderbottomwidth
bordercolor borderimage borderimageoutset borderimagerepeat borderimageslice
borderimagesource borderimagewidth borderleft borderleftcolor borderleftstyle
borderleftwidth borderradius borderright  borderrightcolor borderrightstyle
borderrightwidth borderstyle bordertop bordertopcolor bordertopleftradius
bordertoprightradius bordertopstyle bordertopwidth borderwidth boxdecorationbreak
boxshadow bottom clear clip display float height left margin marginbottom marginleft
marginright margintop maxheight maxwidth minheight minwidth overflow overflowx
overflowy padding paddingbottom paddingleft paddingright paddingtop position
right top visibility width verticalalign zindex aligncontent alignitems alignself
flex flexbasis flexdirection flexflow flexgrow flexshrink flexwrap justifycontent
order hangingpunctuation hyphens letterspacing linebreak lineheight overflowwrap
tabsize textalign textalignlast textcombineupright textindent textjustify
texttransform whitespace wordbreak wordspacing wordwrap textdecoration
textdecorationcolor textdecorationline textdecorationstyle textshadow
textunderlineposition @fontface @fontfeaturevalues font fontfamily fontfeaturesettings
fontkerning fontlanguageoverride fontsize fontsizeadjust fontstretch fontstyle
fontsynthesis fontvariant fontvariantalternates fontvariantcaps fontvarianteastasian
fontvariantligatures fontvariantnumeric fontvariantposition fontweight direction
```

(continues on next page)

(continued from previous page)

```
textorientation unicodebidi writingmode bordercollapse borderspacing captionside
emptycells tablelayout counterincrement counterreset liststyle liststyleimage
liststyleposition liststyletype @keyframes animation animationdelay animationdirection
animationduration animationfillmode animationiterationcount animationname
animationplaystate animationtimingfunction backfacevisibility perspective
perspectiveorigin transform transformorigin transformstyle transition
transitionproperty transitionduration transitiontimingfunction transitiondelay
boxsizing content cursor imemode navdown navindex navleft navright navup
outline outlinecolor outlineoffset outlinestyle outlinewidth resize textoverflow
breakafter breakbefore breakinside columncount columnfill columngap columnrule
columnrulecolor columnrulestyle columnrulewidth columnspan columnwidth columns
widows orphans pagebreakafter pagebreakbefore pagebreakinside marks quotes
filter imageorientation imagerendering imageresolution objectfit objectposition
mask masktype mark markafter markbefore phonemes rest restafter restbefore
voicebalance voiceduration voicepitch voicepitchrange voicerate voicestress
voicevolume marqueedirection marqueeplaycount marqueespeed marqueestyle datatoggle
dataride datatarget dataslideto dataslide datadismiss dataplacement datacontent
datatrigger dataspy dataoffset dataoffsettop
```

52.29 ScriptFunctions Class

This class contains methods for adding JavaScript code to the generated web page.

The class methods are merged to the Page class, so we can use the next methods with page objects directly.

Method	Parameters	Description
Script	cCode	Add cCode string between <script> and </script>
ScriptRedirection	cURL	set window.location to cURL
ScriptFunc	cFuncName,cCode	Define function cFuncName that contains cCode
ScriptFuncAlert	cFuncName,cMsg	Define function cFuncName that uses alert() to print cMsg
ScriptFuncAjax	cFuncName,cLink,cDiv	Define function cFuncName that load cLink in cDiv
ScriptFuncClean	cFuncName,cDiv	Define function cFuncName that clear the cDiv
ScriptFuncSelect	cF,aL,cD,cR,cGR,cFC,nTO,cL1,cL2	Used to Edit/Delete Grid Record
ScriptScrollFixed	cDiv,nSize	Set cDiv as Fixed Div with Size = nSize

52.30 StyleFunctions Class

This class contains methods for adding CSS to the generated web page.

Like ScriptFunctions Class, The StyleFunctions class methods are merged to the Page class, so we can use the next methods with page objects directly.

Method	Parameters	Description
StyleFloatLeft	None	Return float: left ;
StyleFloatRight	None	Return float: right ;
StyleSizeFull	None	Return width: 100% ; height: 100% ;
Stylecolor	x	Return " color: " + x + " ; "
Stylebackcolor	x	Return " background-color: " + x + " ;"
StyleTextCenter	None	Return "text-align: center ;"
StyleTextRight	None	Return "text-align: right ;"
StyleTextLeft	None	Return "text-align: left ;"
StyleSize	x,y	Return " width: " + x + " ; height: " + y + " ;"
StyleWidth	x	Return " width: " + x + " ;"
StyleHeight	x	Return " height: " + x + " ;"
StyleTop	x	Return " top: " + x + " ;"
StyleLeft	x	Return " Left: " + x + " ;"
StylePos	x,y	Return " top: " + x + " ;" + " Left: " + y + " ;"
StyleHorizontalCenter	None	Return " margin-right:auto ; margin-left:auto; "
StyleMarginTop	x	Return " margin-top: " + x + " ;"
StyleMarginRight	x	Return " margin-right: " + x + " ;"
StyleMarginLeft	x	Return " margin-left: " + x + " ;"
StyleDivCenter	nWidth,nHeight	Create Div in the center of the page
StyleAbsolute	None	Return " position:absolute ;"
StyleFixed	None	Return " position:fixed ;"
StyleZIndex	x	Return " z-index: " + x + " ;"
StyleFontSize	x	Return " font-size: " + x + " ;"
StyleGradient	x	Generate Gradient (x values from 1 to 60)
StyleTable	None	Set table properties
StyleTableRows	id	Set different color to even and odd rows in the table
StyleTableNoBorder	None	Return " border-style: none;"

52.31 WebPage Class

We use braces to access the active WebPage object attributes

Each one of these attribute will return a new object to access again using braces.

Attribute	Description
H1	Wraps HTML H1.
H2	Wraps HTML H2.
H3	Wraps HTML H3.
H4	Wraps HTML H4.
H5	Wraps HTML H5.
H6	Wraps HTML H6.

continues on next page

Table 3 – continued from previous page

Attribute	Description
P	Wraps HTML P.
Link	Wraps HTML link.
NewLine	Wraps HTML NewLine.
Div	Wraps HTML Div.
Form	Wraps HTML Form.
Input	Wraps HTML Input.
TextArea	Wraps HTML TextArea.
Select	Wraps HTML Select.
Option	Wraps HTML Option.
Image	Wraps HTML Image.
UL	Wraps HTML UL.
LI	Wraps HTML LI.
Table	Wraps HTML Table.
TR	Wraps HTML TR.
TD	Wraps HTML TD.
TH	Wraps HTML TH.
Audio	Wraps HTML Audio.
Video	Wraps HTML Video.
Nav	Wraps HTML Nav.
Span	Wraps HTML Span.
Button	Wraps HTML Button.
THead	Wraps HTML THEAD.
TBody	Wraps HTML TBODY.
TFoot	Wraps HTML TFOOT.

52.32 HtmlPage Class

The same as the WebPage class with the next changes

(1) No output to the stdout

(2) Provide the Output Method to get the output

Syntax:

```
output() ---> The output as string
```

CHAPTER

FIFTYTHREE

DEPLOYING WEB APPLICATIONS IN THE CLOUD

In this chapter we will learn about deploying Ring Web Applications in the Cloud using Heroku

53.1 Introduction

We created a new project and tutorial to explain how to deploy Ring web applications in the Cloud using Heroku

Project : https://github.com/ring-lang/RingWebAppOnHeroku

Heroku Website : https://www.heroku.com/

53.2 Usage

To use this project and deploy it on Heroku

(1) Create Heroku account

(2) Open your Heroku account and create new application

Example : testring

Note (You have to select a unique name for your application)

(3) Open the command prompt, Create new folder : MyApp

```
md MyApp
```

(4) Open the application folder

```
cd MyApp
```

(5) Clone this project using Git (Don't forget the dot in the end to clone in the current directory)

```
git clone https://github.com/ring-lang/RingWebAppOnHeroku .
```

(6) Login to Heroku (Enter your Email and Password)

```
heroku login
```

(7) Add heroku (remote) to your Git project

change testring to your application name

```
heroku git:remote -a testring
```

(8) Set the buildpacks (So Heroku can know how to support your project)

```
heroku buildpacks:add --index 1 https://github.com/ring-lang/heroku-buildpack-apt
heroku buildpacks:add --index 2 https://github.com/ring-lang/heroku-buildpack-ring
```

(9) Now build your project and deploy it

```
git push heroku master
```

(10) Test your project (In the browser)

```
heroku open
```

53.3 Ring source code files and permissions

To be able to run your new Ring scripts, Set the permission of the file to be executable using Git

For example, if you created a file : myscript.ring

```
git update-index --chmod=+x myscript.ring
git commit -m "Update file permission"
```

If you are using TortoiseGit, From windows explorer, select the file

Right click —> Properties —> Git —> Executable (+x)

Then commit and deploy!

53.4 Hello World program

file : ringapp/helloworld.ring

```
#!/app/runring.sh -cgi

see "content-type: text/html" +nl+nl
see "Hello, World!" + nl
```

file : ringapp/helloworld2.ring

```
#!/app/runring.sh -cgi
load "weblib.ring"
import System.Web
new page {
        text("Hello, World!")
}
```

53.5 Application Database

When you deploy the application, Everything will works directly!

No change is required, but in practice, You will need to update the next files to use your database

There are two scripts to interact with the database (We are using PostgreSQL in the cloud)

You will need to update the connection string in these files if you will use another database

- file: ringapp/database/newdb.ring (We run it using the browser for one time to create the tables)
- file: ringapp/datalib.ring (Class: Database)

In your practical projects, You can write better code (To be able to change the database)

Also you can create configuration file (To write the connection string in one place)

Database service : https://www.heroku.com/postgres

53.6 Deploying after updates

Just use Git and commit then push to heroku

file: build.bat contains the next commands for quick tests

```
git add .
git commit -m "Update RingWebAppOnHeroku"
git push heroku master
heroku open
```

53.7 Local Tests

Local tests using Ring Notepad on Windows (Using local Apache Web Server)

Replace the first line in the file : ringapp/index.ring with

```
#!ring -cgi
```

Then run it from Ring Notepad (Ctrl+F6)

CHAPTER

FIFTYFOUR

GRAPHICS AND 2D GAMES PROGRAMMING USING RINGALLEGRO

In this chapter we will learn how to use the allegro game programming library in our Ring applications.

We have the file gamelib.ring that load the DLL library that contains wrappers for the Allegro functions

```
Load "allegro.rh"
if iswindows()
        LoadLib("ring_allegro.dll")
but ismacosx()
        LoadLib("libringallegro.dylib")
else
        LoadLib("libringallegro.so")
ok
```

The file gamelib.ring uses the Load instruction to execute the file allegro.rh which is a ring source code file contains constants to be used in our programs. Then using the function LoadLib() we can load the DLL library "ring_allegro.dll".

To write portable code we can change the gamelib.ring to check the platform before loading the DLL/So file.

54.1 Drawing, Animation and Input

The next example uses the Allegro library for drawing, moving objects on the screen and getting input from the keyboard and the mouse.

```
Load "gamelib.ring"

al_init()
al_init_image_addon()

display = al_create_display(640,480)

al_show_native_message_box(display, "Hello", "Welcome",
                        "Using Allegro from the Ring programming language",
                        "", 0);

al_clear_to_color(al_map_rgb(0,0,255))

BOUNCER_SIZE = 40
bouncer_x = 10
bouncer_y = 20
bouncer = al_create_bitmap(BOUNCER_SIZE, BOUNCER_SIZE)
```

(continues on next page)

(continued from previous page)

```
al_set_target_bitmap(bouncer)
al_clear_to_color(al_map_rgb(255,0,255))

for x = 1 to 30
        bouncer_x += x
        bouncer_y += x
        al_set_target_bitmap(al_get_backbuffer(display))
        al_clear_to_color(al_map_rgb(0,0,0))
        al_draw_bitmap(bouncer, bouncer_x, bouncer_y, 0)
        al_draw_bitmap(bouncer, 200+bouncer_x,200+ bouncer_y, 0)
        al_flip_display()
        al_rest(0.1)
next

al_clear_to_color(al_map_rgb(255,255,255))
image = al_load_bitmap("man2.jpg")
al_draw_bitmap(image,200,200,0)
al_flip_display()
al_rest(2)

event_queue = al_create_event_queue()
al_register_event_source(event_queue, al_get_display_event_source(display))

ev = al_new_allegro_event()
timeout = al_new_allegro_timeout()
al_init_timeout(timeout, 0.06)

FPS = 60
timer = al_create_timer(1.0 / FPS)
al_register_event_source(event_queue, al_get_timer_event_source(timer))
al_start_timer(timer)
redraw = true

SCREEN_W = 640
SCREEN_H = 480
BOUNCER_SIZE = 32
bouncer_x = SCREEN_W / 2.0 - BOUNCER_SIZE / 2.0
bouncer_y = SCREEN_H / 2.0 - BOUNCER_SIZE / 2.0
bouncer_dx = -4.0
bouncer_dy = 4.0

al_install_mouse()
al_register_event_source(event_queue, al_get_mouse_event_source())

al_install_keyboard()
al_register_event_source(event_queue, al_get_keyboard_event_source())

KEY_UP = 1
KEY_DOWN = 2
KEY_LEFT = 3
KEY_RIGHT = 4
Key = [false,false,false,false]
```

(continues on next page)

(continued from previous page)

```
while true
        al_init_timeout(timeout, 0.06)
        al_wait_for_event_until(event_queue, ev, timeout)
        switch al_get_allegro_event_type(ev)
        on ALLEGRO_EVENT_DISPLAY_CLOSE
                exit
        on ALLEGRO_EVENT_TIMER

                # Animation
                if bouncer_x < 0 or bouncer_x > SCREEN_W - BOUNCER_SIZE
                        bouncer_dx = -bouncer_dx
                ok

                if bouncer_y < 0 or bouncer_y > SCREEN_H - BOUNCER_SIZE
                        bouncer_dy = -bouncer_dy
                ok

                bouncer_x += bouncer_dx
                bouncer_y += bouncer_dy

                # Keyboard
                if key[KEY_UP] and bouncer_y >= 4.0
                        bouncer_y -= 4.0
                ok
                if key[KEY_DOWN] and bouncer_y <= SCREEN_H - BOUNCER_SIZE - 4.0
                        bouncer_y += 4.0
                ok
                if key[KEY_LEFT] and bouncer_x >= 4.0
                        bouncer_x -= 4.0
                ok
                if key[KEY_RIGHT] and bouncer_x <= SCREEN_W - BOUNCER_SIZE - 4.0
                        bouncer_x += 4.0
                ok

                redraw = true

        on ALLEGRO_EVENT_MOUSE_AXES
                bouncer_x = al_get_allegro_event_mouse_x(ev)
                bouncer_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
                bouncer_x = al_get_allegro_event_mouse_x(ev)
                bouncer_y = al_get_allegro_event_mouse_y(ev)
        on ALLEGRO_EVENT_MOUSE_BUTTON_UP
                exit
        on ALLEGRO_EVENT_KEY_DOWN
                switch al_get_allegro_event_keyboard_keycode(ev)
                        on ALLEGRO_KEY_UP
                                key[KEY_UP] = true
                        on ALLEGRO_KEY_DOWN
                                key[KEY_DOWN] = true
                        on ALLEGRO_KEY_LEFT
```

(continues on next page)

(continued from previous page)

```
                                key[KEY_LEFT] = true
                        on ALLEGRO_KEY_RIGHT
                                key[KEY_RIGHT] = true
                off
        on ALLEGRO_EVENT_KEY_UP
                switch al_get_allegro_event_keyboard_keycode(ev)
                        on ALLEGRO_KEY_UP
                                key[KEY_UP] = false
                        on ALLEGRO_KEY_DOWN
                                key[KEY_DOWN] = false
                        on ALLEGRO_KEY_LEFT
                                key[KEY_LEFT] = false
                        on ALLEGRO_KEY_RIGHT
                                key[KEY_RIGHT] = false
                        on ALLEGRO_KEY_ESCAPE
                                exit
                off
        off
        if redraw and al_is_event_queue_empty(event_queue)
                redraw = false
                al_clear_to_color(al_map_rgb(0,0,0))
                al_draw_bitmap(bouncer, bouncer_x, bouncer_y, 0)
                al_flip_display()
        ok
        callgc()
end

al_destroy_timer(timer)
al_destroy_allegro_event(ev)
al_destroy_allegro_timeout(timeout)
al_destroy_event_queue(event_queue)
al_destroy_bitmap(bouncer)
al_destroy_bitmap(image)
al_destroy_display(display)
```

Note: In the previous example we used the function callgc() which is a Ring function to force calling the Garbage collector inside the While/End loop.

Program Output:

At first the program display a messagebox

Then we see two rectangles are moving on the screen

Then we see an image displayed on the screen

Finally we have one rectangle, and we see it moving all of the time on the screen but we can control it using the Mouse and/or the Keyboard

54.2 Using TrueType Fonts

In this example we will see how to use TrueType Fonts *.ttf in our Games using Allegro

```
Load "gamelib.ring"

al_init()
al_init_font_addon()
al_init_ttf_addon()

display = al_create_display(800,600)

al_clear_to_color(al_map_rgb(0,0,255))
font = al_load_ttf_font("pirulen.ttf",14,0 )
al_draw_text(font, al_map_rgb(255,255,255), 10, 10,ALLEGRO_ALIGN_LEFT,
         "Welcome to the Ring programming language")
al_flip_display()
al_rest(2)

al_destroy_display(display)
```

Screen Shot:

54.3 Playing Sound Files

The next example play a sound file

```
Load "gamelib.ring"

al_init()
al_install_audio()
al_init_acodec_addon()
al_reserve_samples(1)

sample = al_load_sample( "footstep.wav" )

sampleid = al_new_allegro_sample_id()
al_play_sample(sample, 1.0, 0.0,1.0,ALLEGRO_PLAYMODE_LOOP,sampleid)

display = al_create_display(640,480)
al_clear_to_color(al_map_rgb(0,0,255))
al_flip_display()
```

(continues on next page)

(continued from previous page)

```
al_rest(10)

al_destroy_allegro_sample_id(sampleid)
al_destroy_sample(sample)
al_destroy_display(display)

al_exit()
```

54.4 Scaling and Rotating Images

The next example display and rotate an image

```
Load "gamelib.ring"

al_init()
al_init_image_addon()

display = al_create_display(640,480)
al_set_target_bitmap(al_get_backbuffer(display))
al_clear_to_color(al_map_rgb(255,255,255))

image = al_load_bitmap("man2.jpg")
al_draw_rotated_bitmap(image,0,0,250,250,150,0)
al_draw_scaled_bitmap(image,0,0,250,250,20,20,400,400,0)

al_flip_display()
al_rest(2)

al_destroy_bitmap(image)
al_destroy_display(display)
```

Screen Shot:

54.5 Display Transparent Image

The next example display image with white background on another image

```
Load "gamelib.ring"

al_init()
al_init_image_addon()

display = al_create_display(640,480)
imageback = al_load_bitmap("palace.jpg")
al_draw_bitmap(imageback,0,0,0)

image = al_load_bitmap("man4.png")
al_convert_mask_to_alpha(image,al_map_rgb(255,255,255))
al_draw_bitmap(image,0,0,0)
al_flip_display()
al_rest(10)

al_destroy_bitmap(image)
```

(continues on next page)

(continued from previous page)

```
al_destroy_display(display)
```

Screen Shot:

54.6 Using Threads

In this example we will learn how to use threads from the Allegro library

```
Load "gamelib.ring"

o1 = new mythreads

Func Main
        al_init()
        for k = 1 to 5
                al_create_thread("o1.thread1()")
                al_create_thread("o1.thread2()")
                al_create_thread("o1.thread3()")
        next
```

(continues on next page)

(continued from previous page)

```
        al_rest(2)

Class Mythreads

        cAppName = "Threads Application"

        Func Thread1
                for x = 1 to 5
                        see x + nl
                next
                See 'Thread(1) : Application Name : '  + cAppName + nl

        Func Thread2
                for x = 1 to 5
                        see '*****' + x + nl
                next
                See 'Thread(2) : Application Name : '  + cAppName + nl

        Func Thread3
                for x = 1 to 5
                        see '!!!!' + x + nl
                next
                See 'Thread(3) : Application Name : '  + cAppName + nl
```

Output:

```
1
2
3
4
5
Thread(1) : Application Name : Threads Application
*****1
*****2
*****3
*****4
*****5
Thread(2) : Application Name : Threads Application
!!!!1
!!!!2
!!!!3
!!!!4
!!!!5
Thread(3) : Application Name : Threads Application
1
2
3
4
5
Thread(1) : Application Name : Threads Application
!!!!1
!!!!2
```

(continues on next page)

(continued from previous page)

```
!!!!3
!!!!4
!!!!5
Thread(3) : Application Name : Threads Application
*****1
*****2
*****3
*****4
*****5
Thread(2) : Application Name : Threads Application
*****1
*****2
*****3
*****4
*****5
Thread(2) : Application Name : Threads Application
!!!!1
!!!!2
!!!!3
!!!!4
!!!!5
Thread(3) : Application Name : Threads Application
1
2
3
4
5
Thread(1) : Application Name : Threads Application
*****1
*****2
*****3
*****1
*****4
*****2
!!!!1
*****5
*****3
1
!!!!2
Thread(2) : Application Name : Threads Application
1
*****4
!!!!1
2
!!!!3
!!!!4
*****5
!!!!2
3
2
!!!!5
Thread(2) : Application Name : Threads Application
```

(continues on next page)

(continued from previous page)

```
!!!!3
4
3
Thread(3) : Application Name : Threads Application
!!!!4
5
4
!!!!5
Thread(1) : Application Name : Threads Application
5
Thread(3) : Application Name : Threads Application
Thread(1) : Application Name : Threads Application
```

CHAPTER

FIFTYFIVE

DEMO PROJECT - GAME ENGINE FOR 2D GAMES

In this chapter we will learn about using the different programming paradigms in the same project.

We will create a simple Game Engine for 2D Games.

You can use the Engine directly to create 2D Games for Desktop or Mobile.

55.1 Project Layers

The project contains the next layers

- Games Layer (Here we will use declarative programming)
- Game Engine Classes (Here we will use the Object-Oriented Programming paradigm)
- Interface to graphics library (Here we will use procedural programming)
- Graphics Library bindings (Here we have RingAllegro and RingLibSDL)

55.2 Graphics Library bindings

We already have RingAllegro to use the Allegro game programming library and we have RingLibSDL to use the LibSDL game programming library.

Both of RingAllegro and RingLibSDL are created using the C language with the help of the Ring code generator for extensions.

Each of them is over 10,000 lines of C code which is generated after writing simple configuration files (That are processed by the code generator).

Each configuration file determines the functions names, structures information and constants then the generator process this configuration file to produce the C code and the library that can be loaded from Ring code.

Using RingAllegro and RingLibSDL is very similar to using Allegro and LibSDL from C code where you have the same functions but we can build on that using the Ring language features

- RingAllegro Source Code : https://github.com/ring-lang/ring/tree/master/extensions/ringallegro
- RingLibSDL Source Code : https://github.com/ring-lang/ring/tree/master/extensions/ringsdl

55.3 Interface to graphics library

In this layer we have gl_allegro.ring and gl_libsdl.ring

Each library provides the same functions to be used with interacting with the Graphics Library.

This layer hides the details and the difference between RingAllegro and RingLibSDL.

You have the same functions, Just use it and you can switch between Allegro and LibSDL at anytime.

Why ?

Allegro is very simple, we can use it to quickly create 2D games for Windows, Linux and MacOS X.

In Ring 1.0 we started by supporting Allegro.

Also LibSDL is very powerful and popular, very easy to use for Mobile Development.

Ring 1.1 comes with support for LibSDL so we can quickly create games for Mobile.

Note: We can use just one library for Desktop and Mobile development.

- gl_allegro.ring source code : https://github.com/ring-lang/ring/blob/master/libraries/gameengine/gl_allegro.ring
- gl_libsdl.ring source code : https://github.com/ring-lang/ring/blob/master/libraries/gameengine/gl_libsdl.ring

55.4 Game Engine Classes

The Engine comes with the next classes

- GameBase class
- Resources class
- Game class
- GameObject class
- Sprite class
- Text class
- Animate class
- Sound class
- Map class
- Source Code : https://github.com/ring-lang/ring/blob/master/libraries/gameengine/gameengine.ring

55.5 Games Layer

In this layer we create our games using the Game Engine classes

The classes are designed to be used through Declarative Programming.

In our games we will use the next classes

- Game class
- Sprite class
- Text class
- Animate class
- Sound class
- Map class

Note: Other classes in the engine are for internal use by the engine.

We will introduce some examples and three simple games :-

- Stars Fighter Game
- Flappy Bird 3000 Game
- Super Man 2016 Game

55.6 Game Class

The next table present the class attributes.

Attributes	Description
FPS	Number determines how many times the draw() method will be called per second.
FixedFPS	Number determines how many times the animate() method will be called per second.
Title	String determines the window title of the game.
Icon	String determines the window icon (file name)
aObjects	List contains all objects in the game
shutdown	True/False value to end the game loop

The next table present the class methods.

Method	Description
refresh()	Delete objects.
settitle(cTitle)	Set the window title using a string parameter.
shutdown()	Close the application.
find(cName)	Find an object using the object name
remove(nID)	Remove an object using the object ID

The next table present a group of keywords defined by the class.

Keyword	Description
sprite	Create new Sprite object and add it to the game objects.
text	Create new Text object and add it to the game objects.
animate	Create new Animate object and add it to the game objects.
sound	Create new Sound object and add it to the game objects.
map	Create new Map object and add it to the game objects.

55.7 GameObject Class

The next table present the class attributes.

Attributes	Description
enabled	True/False determine the state of the object (Active/Not Active)
x	Number determine the x position of the object.
y	Number determine the y position of the object.
width	Number determine the width of the object.
height	Number determine the height of the object.
nIndex	Number determine the ID of the object.
name	String represent the object name.
animate	True/False to animate the object or not.
move	True/False to move the object using the keyboard or not.
Scaled	True/False to scale the object image or not.
draw	Function to be called when drawing the object.
state	Function to be called for object animation.
keypress	Function to be called when a key is pressed.
mouse	Function to be called when a mouse event happens.

The next table present the class methods.

Method	Description
keyboard(oGame,nkey)	Check Keyboard Events
mouse(oGame,nType,aMouseList)	Check Mouse Events
rgb(r,g,b)	Return new color using the RGB (Red, Green and Blue) Values.

55.8 Sprite Class

Parent Class : GameObject Class

The next table present the class attributes.

Attributes	Description
image	String determine the image file name.
point	Number determine the limit of automatic movement of the object.
direction	Number determine the direction of movement.
nstep	Number determine the increment/decrement during movement.
type	Number determine the object type in the game (Optional).
transparent	True/False value determine if the image is transparent.

The next table present the class methods.

Method	Description
Draw(oGame)	Draw the object

55.9 Text Class

Parent Class : Sprite Class

The next table present the class attributes.

Attributes	Description
size	Number determine the font size
font	String determine the font file name
text	String determine the text to be displayed
color	Number determine the color

The next table present the class methods.

Method	Description
Draw(oGame)	Draw the object

55.10 Animate Class

Parent Class : Sprite Class

The next table present the class attributes.

Attributes	Description
frames	Number determine the number of frames
frame	Number determine the active frame
framewidth	Number determine the frame width.
animate	True/False determine using animate or not.
scaled	True/False determine scaling image or not.

The next table present the class methods.

Method	Description
Draw(oGame)	Draw the object

55.11 Sound Class

Parent Class : GameObject Class

The next table present the class attributes.

Attributes	Description
file	String determine the sound file name.
once	True/False determine to play the file one time or not (loop).

The next table present the class methods.

Method	Description
playsound()	Play the sound file

55.12 Map Class

Parent Class : Sprite Class

The next table present the class attributes.

Attributes	Description
aMap	List determine the map content using numbers.
aImages	List determine the image used for each number in the map.
BlockWidth	Number determine the block width (default = 32).
BlockHeight	Number determine the block height (default = 32).
Animate	True/False determine the animation status.

The next table present the class methods.

Method	Description
getvalue(x,y)	Return the item value in the Map according to the visible part

55.13 Using the Game Engine - Creating the Game Window

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

        oGame = New Game        # Create the Game Object
        {
                title = "My First Game"
        }                       # Start the Events Loop
```

Note: if you want to define global variables, this must be before load "gameengine.ring" because this instruction will give the control to the game engine.

Screen Shot:

55.14 Using the Game Engine - Drawing Text

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

        oGame = New Game        # Create the Game Object
        {
                title = "My First Game"
                text {
                        x = 10  y=50
                        animate = false
                        size = 20
                        file = "fonts/pirulen.ttf"
                        text = "game development using ring is very fun!"
                        color = rgb(0,0,0)
                }
        }               # Start the Events Loop
```

Screen Shot:

55.15 Using the Game Engine - Moving Text

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

        oGame = New Game        # Create the Game Object
        {
                title = "My First Game"
                text {
                        x = 10  y=50
                        animate = false
                        size = 20
                        file = "fonts/pirulen.ttf"
                        text = "game development using ring is very fun!"
                        color = rgb(0,0,0)      # Color = black
                }
                text {
                        x = 10  y=150
                        # Animation Part ====================================
                        animate = true                  # Use Animation
                        direction = GE_DIRECTION_INCVERTICAL    # Increase y
                        point = 400                     # Continue until y=400
                        nStep = 3                       # Each time y+= 3
                        #=====================================================
                        size = 20
                        file = "fonts/pirulen.ttf"
                        text = "welcome to the real world!"
                        color = rgb(0,0,255)            # Color = Blue
                }
        }                                       # Start the Events Loop
```

Screen Shot:

55.16 Using the Game Engine - Playing Sound

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

        oGame = New Game        # Create the Game Object
        {
                title = "My First Game"
                text {
                        x = 10  y=50
                        animate = false
                        size = 20
                        file = "fonts/pirulen.ttf"
                        text = "game development using ring is very fun!"
                        color = rgb(0,0,0)      # Color = black
                }
                text {
                        x = 10  y=150
                        # Animation Part ======================================
```

(continues on next page)

(continued from previous page)

```
                        animate = true                          # Use Animation
                        direction = GE_DIRECTION_INCVERTICAL    # Increase y
                        point = 400             # Continue until y=400
                        nStep = 3               # Each time y+= 3
                        #=====================================================
                        size = 20
                        file = "fonts/pirulen.ttf"
                        text = "welcome to the real world!"
                        color = rgb(0,0,255)    # Color = Blue
                }
                Sound {                                 # Play Sound
                        file = "sound/music1.wav"       # Sound File Name
                }
        }                                       # Start the Events Loop
```

55.17 Using the Game Engine - Animation

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

  oGame = New Game      # Create the Game Object
  {
        title = "My First Game"

        animate {
          file = "images/fire.png"
          x = 100
          y = 200
          framewidth = 40
          height = 42
          nStep = 3      # Used for delay
          transparent = true
          state = func oGame,oSelf {  # Called by engine each frame
                oSelf {
                  nStep--
                  if nStep = 0
                        nStep = 3
                        if frame < 13      # we have 13 frames in animation
                          frame++   # move to next frame
                        else
                          oGame.remove(oself.nIndex)   # remove object
                        ok
                  ok
                }
          }
         }


  }          # Start the Events Loop
```

55.18 Using the Game Engine - Animation and Functions

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

  oGame = New Game      # Create the Game Object
  {
        title = "My First Game"
        for x = 70 to 700 step 50
          for y = 70 to 500 step 50
                showfire(oGame,x,y)
          next
        next

  }          # Start the Events Loop

func showfire oGame,nX,nY
  oGame {
        animate {
```

(continues on next page)

(continued from previous page)

```
        file = "images/fire.png"
        x = nX
        y = nY
        framewidth = 40
        height = 42
        nStep = 3      # Used for delay
        transparent = true
        state = func oGame,oSelf {  # Called by engine each frame
              oSelf {
                nStep--
                if nStep = 0
                      nStep = 3
                      if frame < 13      # we have 13 frames in animation
                        frame++   # move to next frame
                      else
                        frame=1
                      ok
                ok
              }
        }
       }
}
```

55.19 Using the Game Engine - Sprite - Automatic Movement using Keyboard

```
Load "gameengine.ring"                          # Give control to the game engine

func main                                       # Called by the Game Engine

        oGame = New Game                        # Create the Game Object
        {
                title = "My First Game"
                sprite
                {
                        type = GE_TYPE_PLAYER           # Just for our usage
                        x=400 y=400 width=100 height=100
                        file = "images/player.png"
                        transparent = true
                        Animate=false
                        Move=true       # we can move it using keyboard arrows
                        Scaled=true
```

(continues on next page)

(continued from previous page)

```
                }
        }                                       # Start the Events Loop
```

55.20 Using the Game Engine - Sprite - Keypress event

```
Load "gameengine.ring"                          # Give control to the game engine

func main                                       # Called by the Game Engine

        oGame = New Game                        # Create the Game Object
        {
                title = "My First Game"
                sprite
                {
                        type = GE_TYPE_PLAYER           # Just for our usage
                        x=400 y=400 width=100 height=100
                        file = "images/player.png"
                        transparent = true
```

(continues on next page)

(continued from previous page)

```
                        Animate=false
                        Move=false                      # Custom Movement
                        Scaled=true
                        keypress = func oGame,oSelf,nKey {
                                oSelf {
                                        Switch nKey
                                        on KEY_LEFT
                                                x -= 10
                                        on KEY_RIGHT
                                                x += 10
                                        on KEY_UP
                                                y -= 10
                                        on KEY_DOWN
                                                y += 10
                                        off
                                }
                        }
                }
        }                                       # Start the Events Loop
```

55.21 Using the Game Engine - Sprite - Mouse event

```
Load "gameengine.ring"                          # Give control to the game engine

func main                                       # Called by the Game Engine

        oGame = New Game                        # Create the Game Object
        {
                title = "My First Game"
                sprite
                {
                        type = GE_TYPE_PLAYER           # Just for our usage
                        x=400 y=400 width=100 height=100
                        file = "images/player.png"
                        transparent = true
                        Animate=false
                        Move=false                      # Custom Movement
                        Scaled=true
                        keypress = func oGame,oSelf,nKey {
                                oSelf {
                                        Switch nKey
                                        on KEY_LEFT
                                                x -= 10
                                        on KEY_RIGHT
                                                x += 10
                                        on KEY_UP
                                                y -= 10
                                        on KEY_DOWN
                                                y += 10
                                        off
```

(continues on next page)

(continued from previous page)

```
                                }
                        }
                        mouse = func oGame,oSelf,nType,aMouseList {
                                if nType = GE_MOUSE_UP
                                        oSelf {
                                                x = aMouseList[GE_MOUSE_X]
                                                y = aMouseList[GE_MOUSE_Y]
                                        }
                                ok
                        }
                }
        }                                       # Start the Events Loop
```

55.22 Using the Game Engine - Sprite - State event

```
Load "gameengine.ring"                          # Give control to the game engine

func main                                       # Called by the Game Engine

        oGame = New Game                        # Create the Game Object
        {
                title = "My First Game"
                sprite
                {
                        type = GE_TYPE_PLAYER           # Just for our usage
                        x=400 y=400 width=100 height=100
                        file = "images/player.png"
                        transparent = true
                        Animate=false
                        Move=false                      # Custom Movement
                        Scaled=true
                        keypress = func oGame,oSelf,nKey {
                                oSelf {
                                        Switch nKey
                                        on KEY_LEFT
                                                x -= 10
                                        on KEY_RIGHT
                                                x += 10
                                        on KEY_UP
                                                y -= 10
                                        on KEY_DOWN
                                                y += 10
                                        off
                                }
                        }
                        mouse = func oGame,oSelf,nType,aMouseList {
                                if nType = GE_MOUSE_UP
                                        oSelf {
                                                x = aMouseList[GE_MOUSE_X]
                                                y = aMouseList[GE_MOUSE_Y]
```

(continues on next page)

(continued from previous page)

```
                                        }
                                ok
                        }
                        state = func oGame,oSelf {
                                oself {
                                        if x < 0 x = 0 ok
                                        if y < 0 y = 0 ok
                                        if x > ogame.width-width
                                                x= ogame.width - width ok
                                        if y > ogame.height-height
                                                y=ogame.height - height ok
                                }
                        }
                }
        }                                       # Start the Events Loop
```

55.23 Using the Game Engine - Animate - Events

```
Load "gameengine.ring"                          # Give control to the game engine

func main                                       # Called by the Game Engine

        oGame = New Game                        # Create the Game Object
        {
                title = "My First Game"

                animate {

                        file = "images/fbbird.png"
                        x = 10
                        y = 10
                        framewidth = 20
                        scaled = true
                        height = 50
                        width = 50
                        nStep = 3
                        transparent = true

                        state = func oGame,oSelf {
                                oSelf {

                                        # Animation
                                                nStep--
                                                if nStep = 0
                                                        nStep = 3
                                                        if frame < 3
                                                                frame++
                                                        else
                                                                frame=1
                                                        ok
```

(continues on next page)

(continued from previous page)

```
                                        ok

                                # Move Down
                                        y += 3
                                        if y > 550 y=550 ok

                        }

                }

                keypress = func ogame,oself,nKey {
                        oself {
                                if nkey = key_space
                                        y -= 55
                                        if y<=0 y=0 ok
                                ok
                        }
                }

                mouse = func ogame,oself,nType,aMouseList {
                        if nType = GE_MOUSE_UP
                                cFunc = oself.keypress
                                call cFunc(oGame,oSelf,Key_Space)
                        ok
                }
        }
}                                       # Start the Events Loop
```

Screen Shot:

55.24 Using the Game Engine - Map

```
Load "gameengine.ring"        # Give control to the game engine

func main          # Called by the Game Engine

oGame = New Game      # Create the Game Object
{
  title = "My First Game"

  Map {

        blockwidth = 80
        blockheight = 80

        aMap = [
                [0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
                [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
                [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,1,0,0,0],
                [0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,3,0,0,0,1,0,0,0,1,0,0,0],
```

(continues on next page)

(continued from previous page)

```
                [0,0,0,0,0,0,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0],
                [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
                [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
                [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
        ]

        aImages = ["images/fbwall.png",
                   "images/fbwallup.png",
                   "images/fbwalldown.png"]

        state = func oGame,oSelf {
          oSelf {
                x -= 3
                if x < - 2100  x = 0  ok
          }
        }

  }
}          # Start the Events Loop
```

Screen Shot:

55.25 Using the Game Engine - Map Events

```
Load "gameengine.ring"        # Give control to the game engine

func main          # Called by the Game Engine

  oGame = New Game      # Create the Game Object
  {
        title = "My First Game"

        Map {

          blockwidth = 80
          blockheight = 80

          aMap = [
                   [0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
                  [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
                  [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,1,0,0,0],
                  [0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,3,0,0,0,1,0,0,0,1,0,0,0],
                  [0,0,0,0,0,0,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0],
                  [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
          ]

          aImages = ["images/fbwall.png",
                         "images/fbwallup.png",
                         "images/fbwalldown.png"]

          state = func oGame,oSelf {
                oSelf {
                  x -= 3
                  if x < - 2100  x = 0  ok
                }
          }

          mouse = func ogame,oself,nType,aMouseList {
                if nType = GE_MOUSE_UP
                  oSelf {
                        mX = aMouseList[GE_MOUSE_X]
                        mY = aMouseList[GE_MOUSE_Y]
                        nValue = GetValue(mX,mY)
                        nRow = GetRow(mX,mY)
                        nCol = GetCol(mX,mY)
                        Switch nValue
                        On 1  aMap[nRow][nCol] = 0
                        On 2  aMap[nRow][nCol] = 0
                        On 3  aMap[nRow][nCol] = 0
                        On 0  aMap[nRow][nCol] = 1
                        Off
                  }
```

(continues on next page)

(continued from previous page)

```
                ok
            }

        }
    }           # Start the Events Loop
```

Screen Shot:

55.26 Using the Game Engine - Object and Drawing

We can use the Object keyword (defined by the game engine) to create objects from the GameObject class.

Example:

```
Load "gameengine.ring"                          # Give control to the game engine

func main                                       # Called by the Game Engine

        oGame = New Game                        # Create the Game Object
        {
```

(continues on next page)

(continued from previous page)

```
                title = "My First Game"
                Object {
                        x = 0 y=300 width = 200 height=200
                        draw = func oGame,oSelf {
                                oSelf {
                                        for t = 1 to 210
                                                gl_draw_circle(x,y,t,
                                                gl_map_rgb(t*random(255),
                                                t*2,t*3),1)
                                        next
                                }
                        }
                        state = func oGame,oSelf {
                                oSelf {
                                        if x <= 800
                                                x+= 3
                                        else
                                                x=0
                                        ok
                                }
                        }
                        keypress = func oGame,oSelf,nKey {
                                oSelf {
                                        Switch nKey
                                        on KEY_LEFT
                                                x -= 10
                                        on KEY_RIGHT
                                                x += 10
                                        on KEY_UP
                                                y -= 10
                                        on KEY_DOWN
                                                y += 10
                                        off
                                }
                        }
                }

        }                                       # Start the Events Loop
```

Screen Shot:

Example:

```
Load "gameengine.ring"                          # Give control to the game engine

func main                                       # Called by the Game Engine

        oGame = New Game                        # Create the Game Object
        {
                title = "My First Game"
                Object {
                        x = 400 y=300 width = 200 height=200
                        draw = func oGame,oSelf {
                                oSelf {
                                        for t = 1 to 210
                                                gl_draw_rectangle(x+t,y+t,
                                                x+t*2,y+t*2,
                                                gl_map_rgb(t*random(255),
                                                t*2,t*3),1)
                                                gl_draw_rectangle(x+t*2,y+t*2,
                                                x-t*2,y-t*2,
                                                gl_map_rgb(t*random(255),
                                                t*2,t*3),1)
```

(continues on next page)

(continued from previous page)

```
                                        next
                                }
                        }
                        keypress = func oGame,oSelf,nKey {
                                oSelf {
                                        Switch nKey
                                        on KEY_LEFT
                                                x -= 10
                                        on KEY_RIGHT
                                                x += 10
                                        on KEY_UP
                                                y -= 10
                                        on KEY_DOWN
                                                y += 10
                                        off
                                }
                        }
                }

        }                                       # Start the Events Loop
```

Screen Shot:

55.27 Stars Fighter Game

The Stars Fighter source code

```
# The Ring Standard Library
# Game Engine for 2D Games
# 2016, Mahmoud Fayed <msfclipper@yahoo.com>

oGameState = NULL

load "gameengine.ring"

func main

  oGame = New Game

  while true

          oGameState = new GameState
```

(continues on next page)

(continued from previous page)

```
        oGame {
              title = "Stars Fighter!"
              sprite
              {
                file = "images/menu1.jpg"
                x = 0 y=0 width=800 height = 600 scaled = true animate = false
                keypress = func ogame,oself,nKey {
                      if nkey = key_esc or nKey = GE_AC_BACK
                        ogame.shutdown()
                      but nKey = key_space
                        oGameState.startplay=true
                        ogame.shutdown=true
                      ok
                }
                mouse = func ogame,oself,nType,aMouseList {
                      if nType = GE_MOUSE_UP
                        oGameState.startplay=true
                        ogame.shutdown=true
                      ok
                }
              }
              text {
                animate = false
                size = 35
                file = "fonts/pirulen.ttf"
                text = "Stars Fighter"
                x = 10  y=50
              }
              text {
                animate = false
                size = 25
                file = "fonts/pirulen.ttf"
                text = "Version 1.0"
                x = 80  y=100
              }
              text {
                animate = false
                size = 16
                file = "fonts/pirulen.ttf"
                text = "(C) 2016, Mahmoud Fayed"
                x = 45  y=140
              }

              text {
                animate = false
                size = 25
                file = "fonts/pirulen.ttf"
                text = "Press Space to start"
                x = 190  y=470
              }
              text {
                animate = false
```

(continues on next page)

(continued from previous page)

```
                  size = 20
                  file = "fonts/pirulen.ttf"
                  text = "Press Esc to Exit"
                  x = 260  y=510
                }
                Sound {
                  file = "sound/music1.wav"
                }
          }

          if oGameState.startplay
                oGame.refresh()
                playstart(oGame)
                oGame.refresh()
          ok

  end

func playstart oGame

  oSound = New Sound {
        file = "sound/music2.wav"
  }

  while true
        play(oGame)
        if ogame.shutdown = true and oGameState.value = 0
          exit
        ok
        ogame.refresh()
  end

  oSound.Delete()

func play oGame

  oGame
  {
        FPS = 60
        FixedFPS = 120
        title = "Stars Fighter!"
        sprite
        {
          file = "images/stars.jpg"
          x = 0
          y = 0
          point = -370
          direction = ge_direction_dec
          type = ge_type_background
          state = func ogame,oself {
                  oself {
                        if x < -350
```

(continues on next page)

(continued from previous page)

```
                        direction = ge_direction_inc
                        point = 370
                      but x = 0 and direction = ge_direction_inc
                        direction = ge_direction_dec
                        point = -370
                      ok
                  }
                }
        }
        sprite
        {
          file = "images/player.png"
          transparent = true
          type = ge_type_player
          x = 400 y =400 width=100 height=100
          animate=false move=true Scaled=true
          mouse = func ogame,oself,nType,aMouseList {

                if not ( aMouseList[GE_MOUSE_X] >= oSelf.x and
                                 aMouseList[GE_MOUSE_X] <= oSelf.x+oSelf.width and
                                 aMouseList[GE_MOUSE_Y] >= oself.y and
                                 aMouseList[GE_MOUSE_Y] <= oSelf.y+oSelf.height )

                  if nType = GE_MOUSE_DOWN
                      if aMouseList[1] < oSelf.X  # left
                        oSelf.X -= 100
                      else
                        oSelf.X += 100
                      ok
                      if aMouseList[2] < oSelf.Y  # up
                        oSelf.Y -= 100
                      else
                        oSelf.Y += 100
                      ok
                  ok

                else
                  if nType = GE_MOUSE_UP
                      cFunc = oself.keypress
                      call cFunc(oGame,oSelf,Key_Space)
                  ok
                ok
          }
          keypress = func oGame,oself,nkey {
                if nkey = key_space
                  ogame {
                      sprite {
                        type = ge_type_fire
                        file  = "images/rocket.png"
                        transparent = true
                              x = oself.x + 30
                        y = oself.y - 30
```

(continues on next page)

(continued from previous page)

```
                        width = 30
                        height = 30
                        point = -30
                        nstep = 20
                        direction = ge_direction_decvertical
                        state = func oGame,oSelf {
                              for x in oGame.aObjects
                                if x.type = ge_type_enemy
                                      if oself.x >= x.x and oself.y >= x.y and
                                        oself.x <= x.x + x.width and
                                        oself.y <= x.y + x.height
                                        showfire(oGame,x.x+40,x.y+40)
                                        ogame.remove(x.nindex)
                                        oGameState.score+=10
                                        oGameState.enemies--
                                        checkwin(oGame)
                                        exit
                                      ok
                                ok
                              next
                        }
                      }
                }
              but nkey = key_esc or nKey = GE_AC_BACK ogame.shutdown()
              ok
        }
        state = func oGame,oSelf {
              oself {
                if x < 0 x = 0 ok
                if y < 0 y = 0 ok
                if x > ogame.screen_w-width  x= ogame.screen_w - width ok
                if y > ogame.screen_h-height y=ogame.screen_h-height ok
              }
        }
      }
      for g = 1 to oGameState.enemies
        sprite
        {
              type = ge_type_enemy
              file = "images/enemy.png"
              transparent = true
              x = g*random(50) y =g width=100 height=100
              animate=true Scaled=true
              direction = ge_direction_random
              state = func oGame,oSelf {
                oself {
                      if x < 0 x = 0 ok
                      if y < 0 y = 0 ok
                      if x > ogame.screen_w-width  x= ogame.screen_w - width ok
                      if y > ogame.screen_h-height y=ogame.screen_h-height ok
                }
                if random(100) = 1
```

(continues on next page)

(continued from previous page)

```
                        ogame {
                          sprite {
                                type = ge_type_fire
                                file  = "images/rocket2.png"
                                transparent = true
                                x = oself.x + 30
                                y = oself.y + oself.height+ 30
                                width = 30
                                height = 30
                                point = ogame.screen_h+30
                                nstep = 10
                                direction = ge_direction_incvertical
                                state = func oGame,oSelf {
                                  x =  oGame.aObjects[oGameState.playerindex]
                                  if oself.x >= x.x and oself.y >= x.y and
                                         oself.x <= x.x + x.width and
                                         oself.y <= x.y + x.height
                                         if oGameState.value > 0
                                           oGameState.value-=10
                                         ok
                                         ogame.remove(oself.nindex)
                                         checkgameover(oGame)
                                  ok
                                }
                          }
                        }
                  ok
                }
          }
        next
        text {
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Destroy All Enemies!"
          nstep = 3
          color = GE_COLOR_GREEN
          x = 100  y=50
          direction = ge_direction_incvertical
          point = 500
        }
        text {
          animate = false
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Score : " + oGameState.score
          x = 500  y=10
          state = func oGame,oSelf { oSelf { text = "Score : " + oGameState.score } }
        }
        text {
          animate = false
          point = 400
```

(continues on next page)

(continued from previous page)

```
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Energy : " + oGameState.value
          x = 500  y=50
          state = func oGame,oSelf { oSelf { text = "Energy : " + oGameState.value } }
        }
        text {
          animate = false
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Level : " + oGameState.level
          x = 500  y=90
        }
  }


func checkwin ogame
  if oGameState.gameresult  return ok
  if oGameState.enemies = 0
        oGameState.gameresult = true
        oGame {
          if oGameState.level < 30
          text {
                point = 400
                size = 30
                file = "fonts/pirulen.ttf"
                text = "Level Completed!"
                nStep = 3
                x = 500  y=10
                state = func ogame,oself {
                  if oself.y >= 400
                        ogame.shutdown = true
                        oGameState.level++
                        oGameState.enemies = oGameState.level
                        oGameState.gameresult = false
                  ok
                }
          }
          else
          text {
                point = 400
                size = 30
                nStep = 3
                file = "fonts/pirulen.ttf"
                text = "You Win !!!"
                x = 500  y=10
                state = func ogame,oself {
                  if oself.y >= 400
                        ogame.shutdown = true
                        oGameState.value = 0
                  ok
```

(continues on next page)

(continued from previous page)

```
                }
            }
            ok
        }
    ok

func checkgameover ogame
    if oGameState.gameresult  return ok
    if oGameState.value <= 0
        oGameState.gameresult = true
        oGame {
            text {
                point = 400
                size = 30
                nStep = 3
                file = "fonts/pirulen.ttf"
                text = "Game Over !!!"
                x = 500  y=10
                state = func ogame,oself {
                    if oself.y >= 400
                        ogame.shutdown = true
                    ok
                }
            }
        }
        showfire(oGame,oGame.aObjects[oGameState.PlayerIndex].x+40,
                         oGame.aObjects[oGameState.PlayerIndex].y+40)
        oGame.aObjects[oGameState.PlayerIndex].enabled = false
        oGame.remove(oGameState.PlayerIndex)
    ok


func showfire oGame,nX,nY
    oGame {
        animate {
            file = "images/fire.png"
            x = nX
            y = nY
            framewidth = 40
            height = 42
            nStep = 3
            transparent = true
            state = func oGame,oSelf {
                oSelf {
                    nStep--
                    if nStep = 0
                        nStep = 3
                        if frame < 13
                            frame++
                        else
                            frame=1
                            oGame.remove(oself.nIndex)
```

(continues on next page)

(continued from previous page)

```
                        ok
                    ok
                }
            }
        }
    }


class gamestate
    score = 0
    level = 1
    enemies = 1
    value = 100
    playerindex = 2
    gameresult = false
    startplay=false
```

Screen Shot:

55.28 Flappy Bird 3000 Game

The Flappy Bird 3000 Game source code

```
# The Ring Standard Library
# Game Engine for 2D Games
# 2016, Mahmoud Fayed <msfclipper@yahoo.com>

oGameState = NULL

Load "gameengine.ring"

func main

  oGame = New Game


  while true

        oGameState = New GameState

        oGame {
          title = "Flappy Bird 3000"
          sprite
          {
                file = "images/fbback.png"
                x = 0 y=0 width=800 height = 600 scaled = true animate = false
                keypress = func ogame,oself,nKey {
                  if nkey = key_esc or nKey = GE_AC_BACK
                        ogame.shutdown()
                  but nKey = key_space
                        oGameState.startplay=true
                        ogame.shutdown=true
                  ok
                }
                mouse = func ogame,oself,nType,aMouseList {
                  if nType = GE_MOUSE_UP
                        cFunc = oself.keypress
                        call cFunc(oGame,oSelf,Key_Space)
                  ok
                }
          }
          text {
                animate = false
                size = 35
                file = "fonts/pirulen.ttf"
                text = "Flappy Bird 3000"
                x = 150  y=50
          }
          text {
                animate = false
                size = 25
                file = "fonts/pirulen.ttf"
```

(continues on next page)

(continued from previous page)

```
            text = "Version 1.0"
            x = 280  y=100
        }
        text {
            animate = false
            size = 16
            file = "fonts/pirulen.ttf"
            text = "(C) 2016, Mahmoud Fayed"
            x = 245  y=140
        }

        text {
            animate = false
            size = 25
            file = "fonts/pirulen.ttf"
            text = "To Win Get Score = 3000"
            x = 150  y=270
        }

        text {
            animate = false
            size = 25
            file = "fonts/pirulen.ttf"
            text = "Press Space to start"
            x = 190  y=470
        }
        text {
            animate = false
            size = 20
            file = "fonts/pirulen.ttf"
            text = "Press Esc to Exit"
            x = 260  y=510
        }

        animate {
            file = "images/fbbird.png"
            x = 200
            y = 200
            framewidth = 20
            scaled = true
            height = 50
            width = 50
            nStep = 3
            transparent = true
            animate = true
            direction = ge_direction_random
            state = func oGame,oSelf {
              oSelf {
                  nStep--
                  if nStep = 0
                    nStep = 3
                    if frame < 3
```

(continues on next page)

(continued from previous page)

```
                                frame++
                          else
                                frame=1
                          ok
                        ok
                        if x <= 0 x=0 ok
                        if y <= 0 y=0 ok
                        if x >= 750 x= 750 ok
                        if y > 550 y=550 ok
                    }
                  }
            }

            Sound {
                  file = "sound/music2.wav"
            }
        }
        if oGameState.startplay
            oGame.refresh()
            playstart(oGame)
            oGame.refresh()
        ok

    end


func playstart oGame

    oGame {
        FPS = 60
        FixedFPS = 120
        Title = "Flappy Bird 3000"
        Sprite {
            file = "images/fbback.png"
            x = 0 y=0 width=800 height = 600 scaled = true animate = false
            keypress = func ogame,oself,nKey {
                  if nkey = key_esc or nKey = GE_AC_BACK
                      ogame.shutdown()
                  ok
            }
        }

        Map {
            blockwidth = 80
            blockheight = 80
            aMap = [
                      [0,0,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
                    [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0],
                    [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,1,0,0,0],
                    [0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,3,0,0,0,1,0,0,0,1,0,0,0],
                    [0,0,0,0,0,0,0,0,0,3,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,3,0,0,0],
                    [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
```

(continues on next page)

(continued from previous page)

```
                [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
                [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
              ]
        newmap(aMap)
        aImages = ["images/fbwall.png","images/fbwallup.png",
                "images/fbwalldown.png"]
        state = func oGame,oSelf {
              if oGameState.gameresult = false
                px = oGame.aObjects[3].x
                py = oGame.aObjects[3].y
                oSelf {
                      x -=  3
                      if x < - 2100
                        x = 0
                        newmap(aMap)
                      ok
                      nCol =  getcol(px,0)
                      if nCol=11 or nCol=15 or nCol=19 or nCol=23 or nCol=27
                        if nCol != oGameState.lastcol
                              oGameState.lastcol = nCol
                              oGameState.Score += 100
                              oGame { Sound {
                                once = true
                                file = "sound/sfx_point.wav"
                              } }
                              checkwin(oGame)
                        ok
                      ok
                }
                if  oSelf.getvalue(px+40,py) != 0 or
                        oSelf.getvalue(px+40,py+40) != 0 or
                        oSelf.getvalue(px,py) != 0 or
                        oSelf.getvalue(px,py+40) != 0
                      oGameState.gameresult = true
                      oGame {
                        text {
                              point = 550
                              size = 30
                              nStep = 3
                              file = "fonts/pirulen.ttf"
                              text = "Game Over !!!"
                              x = 500  y=10
                              state = func ogame,oself {
                                if oself.y >= 550
                                        ogame.shutdown = true
                                ok
                                      if oself.y = 90
                                      ogame {
                                        Sound {
                                              once = true
                                              file = "sound/sfx_die.wav"
                                        }
```

(continues on next page)

(continued from previous page)

```
                                        }
                                ok
                              }
                        }
                        Sound {
                              once = true
                              file = "sound/sfx_hit.wav"
                        }
                      }
                ok
              ok
        }
      }

      animate {
        file = "images/fbbird.png"
        x = 10
        y = 10
        framewidth = 20
        scaled = true
        height = 50
        width = 50
        nStep = 3
        transparent = true
        state = func oGame,oSelf {
              oSelf {
                nStep--
                if nStep = 0
                      nStep = 3
                      if frame < 3
                        frame++
                      else
                        frame=1
                      ok
                ok
              }

              if not oGameState.playerwin
                oGameState.down --
                if oGameState.down = 0
                      oGameState.down = 3
                      oself {
                        y += 25
                        if y > 550 y=550 ok
                      }
                ok
              ok

        }
        keypress = func ogame,oself,nKey {
              if oGameState.gameresult = false
                oself {
```

(continues on next page)

(continued from previous page)

```
                        if nkey = key_space
                          y -= 55
                          oGameState.down = 60
                          if y<=0 y=0 ok
                        ok
                  }
                ok
          }
          mouse = func ogame,oself,nType,aMouseList {
                if nType = GE_MOUSE_UP
                  cFunc = oself.keypress
                  call cFunc(oGame,oSelf,Key_Space)
                ok
          }
        }

        text {
          animate = false
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Score : " + oGameState.score
          x = 500  y=10
          state = func oGame,oSelf {
                oSelf { text = "Score : " + oGameState.score }
          }
        }

  }

func newmap aMap
  aV = [
  [1,1,3,0,0,2,1,1],
  [1,3,0,0,0,2,1,1],
  [1,1,1,3,0,2,1,1],
  [1,1,1,3,0,0,0,0],
  [0,0,0,0,2,1,1,1],
  [0,0,2,1,1,1,1,1],
  [0,0,0,2,1,1,1,1],
  [1,1,1,3,0,2,1,1],
  [1,1,1,1,1,3,0,0],
  [3,0,0,2,1,1,1,1],
  [3,0,0,2,3,0,0,2]
  ]
  for x = 10 to 24 step 4
        aVar = aV[ (random(10)+1) ]
        for y = 1 to 8
          aMap[y][x] = aVar[y]
        next
  next

func checkwin ogame
```

(continues on next page)

(continued from previous page)

```
  if oGameState.score = 3000
        oGameState.gameresult = true
        oGameState.playerwin = true
        oGame {
          text {
                point = 400
                size = 30
                nStep = 3
                file = "fonts/pirulen.ttf"
                text = "You Win !!!"
                x = 500  y=10
                state = func ogame,oself {
                  if oself.y >= 400
                        ogame.shutdown = true
                        oGameState.value = 0
                  ok
                }
          }
        }
  ok

Class GameState
  down = 3
  gameresult = false
  Score = 0
  startplay=false
  lastcol = 0
  playerwin = false
```

Screen Shot:

55.29 Super Man 2016 Game

The Super Man 2016 Game source code

```
# The Ring Standard Library
# Game Engine for 2D Games
# 2016, Mahmoud Fayed <msfclipper@yahoo.com>

oGameState = NULL

Load "gameengine.ring"

func main

  oGame = New Game

  while true

        oGameState = new GameState

        oGame {
```

(continues on next page)

(continued from previous page)

```
        title = "Super Man 2016"
        sprite
        {
              file = "images/superman.jpg"
              x = 0 y=0 width=800 height = 600 scaled = true animate = false
              keypress = func ogame,oself,nKey {
                if nkey = key_esc or nKey = GE_AC_BACK
                      ogame.shutdown()
                but nKey = key_space
                      oGameState.startplay=true
                      ogame.shutdown=true
                ok
              }
              mouse = func ogame,oself,nType,aMouseList {
                if nType = GE_MOUSE_UP
                      oGameState.startplay=true
                      ogame.shutdown=true
                ok
              }
              state = func ogame,oself {
                oself {
                      if x > -500
                        x-=1
                        y-=1
                        width +=1
                        height +=4
                      ok
                }
              }
        }
        text {
              animate = false
              size = 35
              file = "fonts/pirulen.ttf"
              text = "Super Man 2016"
              x = 20  y=30
        }
        text {
              animate = false
              size = 25
              file = "fonts/pirulen.ttf"
              text = "Version 1.0"
              x = 20  y=80
        }
        text {
              animate = false
              size = 16
              file = "fonts/pirulen.ttf"
              text = "(C) 2016, Mahmoud Fayed"
              x = 20  y=120
        }
```

(continues on next page)

(continued from previous page)

```
        text {
              animate = false
              size = 25
              file = "fonts/pirulen.ttf"
              text = "Press Space to start"
              x = 190  y=470
        }
        text {
              animate = false
              size = 20
              file = "fonts/pirulen.ttf"
              text = "Press Esc to Exit"
              x = 260  y=510
        }

        animate {
              file = "images/superman.png"
              x = 200
              y = 200
              framewidth = 68
              scaled = true
              height = 86
              width = 60
              nStep = 10
              transparent = true
              animate = true
              direction = ge_direction_random
              state = func oGame,oSelf {
                oSelf {
                      nStep--
                      if nStep = 0
                        nStep = 10
                        if frame < 1
                              frame++
                        else
                              frame=1
                        ok
                      ok
                      if x <= 0 x=0 ok
                      if y <= 0 y=0 ok
                      if x >= 750 x= 750 ok
                      if y > 550 y=550 ok
                }
              }
        }

        Sound {
              file = "sound/music2.wav"
        }
      }
      if oGameState.startplay
        oGame.refresh()
```

(continues on next page)

(continued from previous page)

```
          playstart(oGame)
          oGame.refresh()
        ok

  end


func playstart oGame

  oGame {
        FPS = 60
        FixedFPS = 15
        Title = "Super Man 2016"
        Sprite {
          file = "images/supermancity.jpg"
          x = 0 y=0 width=800 height = 600 scaled = true animate = false
        }
        Map {
          blockwidth = 80
          blockheight = 80
          aMap = [
                  [0,0,0,4,4,4,0,0,0,1,0,0,0,1,4,4,0,1,0,0,0,0,4,4,0,1,4,
4,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,1,0,0,0,1,0,3,3,3,5,3,3,3,3,0],
                  [0,0,4,0,4,0,4,0,0,1,0,0,0,3,4,4,4,1,0,0,0,0,4,4,0,1,4,
4,4,0,0,4,4,4,4,4,4,4,4,4,4,4,4,4,1,4,1,0,0,0,1,0,0,0,1,0,4,4,4,4,4,4,4,4,0],
                  [0,0,0,4,4,4,0,0,0,1,0,0,0,4,4,4,4,1,0,0,0,0,0,0,0,3,4,
4,4,0,0,4,0,0,0,0,0,0,4,2,0,0,4,1,4,1,4,2,4,1,0,2,0,1,0,4,4,4,4,4,4,4,4,4,0],
                  [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,
0,0,0,0,4,4,4,4,4,4,4,4,1,0,0,4,1,4,1,4,1,4,1,0,1,0,1,0,2,2,2,2,2,2,2,2,0],
                  [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,
0,0,0,0,2,0,0,0,0,0,2,0,3,0,0,0,1,4,1,4,1,4,1,0,1,0,1,0,1,0,0,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,1,0,0,0,2,0,0,0,0,0,
0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,4,3,4,1,4,3,0,1,0,3,0,1,0,0,0,0,0,0,0,0,0],
                  [0,0,2,0,0,2,0,0,2,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,
0,0,0,0,1,0,0,0,0,0,3,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0],
                  [0,0,1,0,0,1,0,0,1,3,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0]
                 ]
          aImages = ["images/smwall.png","images/smwallup.png",
                  "images/smwalldown.png","images/smstar.png",
                  "images/smkey.png","images/smstar2.png"]
        }

        sprite {
          type = ge_type_enemy
          animate = false
          file  = "images/smhome.png"
          x = 5000
          y = 400
          width = 290
          height = 200
          transparent = true
```

(continues on next page)

(continued from previous page)

```
        state = func oGame,oSelf {
              oself {
                x = 5000 +  oGame.aObjects[2].x
                if x < 0 or x > SCREEN_W return ok
              }
              if oGameState.gameresult or oGameState.DoorKey = false  return ok
              if oGame.aObjects[oGameState.playerindex].x > oself.x + 100 and
                oGame.aObjects[oGameState.playerindex].y > oself.y + 50
                oGameState.gameresult = true
                oGame {
                      sprite {
                        file = "images/smwin.jpg"
                        x=0 y=0 width=800 height=600
                        scaled = true animate=false
                        state = func ogame,oself {
                              oself {
                                x-=5
                                y-=5
                                width +=10
                                height +=10
                                if x = -300
                                      ogame.shutdown = true
                                ok
                              }
                        }
                      }
                }
              ok
        }

      }

      animate {
        file = "images/superman.png"
        x = 0
        y = 0
        framewidth = 60
        scaled = true
        height = 86
        width = 60
        nStep = 3
        transparent = true
        state = func oGame,oSelf {

              checkstarskeycol(oGame,oSelf)

              if not oGameState.playerwin
                      oself {
                        file = "images/superman.png"
                        height = 86
                        width = 60
```

(continues on next page)

(continued from previous page)

```
                        for t=1 to 8
                              if checkwall2(oGame,oSelf,0,5,[2,1])
                                y += 5
                              else
                                exit
                              ok
                        next
                        if y > 500 y=500 ok
                      }
                ok

        }
        keypress = func ogame,oself,nKey {
                if oGameState.gameresult = false

                  oself {
                      if nkey = key_up  and checkwall(oGame,oSelf,0,-40)
                        oGameState.value -= 1
                        checkgameover(oGame)
                        file = "images/supermanup.png"
                        height = 123
                        dotransparent()
                        y -= 40
                        oGameState.down = 10
                        if y<=0 y=0 ok
                      but nkey = key_down and checkwall(oGame,oSelf,0,40)
                        file = "images/supermandown.png"
                        dotransparent()
                        y += 40
                        if y>=500 y=500 ok
                      but nKey = key_right and checkwall(oGame,oSelf,10,0)
                        file = "images/supermanright.png"
                        dotransparent()
                        x += 10
                        if x >= 440
                              if oGame.aObjects[2].x > -4500
                                oGame.aObjects[2].x -= 50
                                callenemystate(oGame)
                              else
                                if x <= 750
                                      if  checkwall(oGame,oSelf,10,0)
                                        x += 10
                                      ok
                                else
                                      if  checkwall(oGame,oSelf,-10,0)
                                        x -= 10
                                      ok
                                ok
                                return
                              ok
                              x=400
                        ok
```

(continues on next page)

(continued from previous page)

```
                    but nKey = key_left and checkwall(oGame,oSelf,-10,0)
                      file = "images/supermanleft.png"
                      dotransparent()
                      x -= 10
                      if x <= 0
                            x += 10
                            if oGame.aObjects[2].x != 0
                              oGame.aObjects[2].x += 50
                              callenemystate(oGame)
                              x += 50
                            ok
                      ok
                    but nkey = key_esc or nKey = GE_AC_BACK
                      ogame.shutdown()
                    ok
                }
              ok
        }
        mouse = func ogame,oself,nType,aMouseList {
              if nType = GE_MOUSE_DOWN
                oGameState.moveplayer = TRUE
              But nType = GE_MOUSE_UP
                oGameState.moveplayer = FALSE
              ok
              if oGameState.moveplayer = TRUE
                if aMouseList[GE_MOUSE_X] < oSelf.X  # left
                      cFunc = oself.keypress
                      call cFunc(oGame,oSelf,Key_left)
                else
                      cFunc = oself.keypress
                      call cFunc(oGame,oSelf,Key_right)
                ok
                if aMouseList[GE_MOUSE_Y] < oSelf.Y  # up
                      cFunc = oself.keypress
                      call cFunc(oGame,oSelf,Key_up)
                else
                      cFunc = oself.keypress
                      call cFunc(oGame,oSelf,Key_down)
                ok
              ok
        }
      }

      addenemy(oGame,600)
      addenemy(oGame,900)
      addenemy(oGame,1550)
      addenemy(oGame,2350)
      addenemy(oGame,3350)
      addenemy(oGame,3500)
      addenemy(oGame,3670)
      addenemy(oGame,3840)
```

(continues on next page)

(continued from previous page)

```
        text {
          animate = false
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Score : " + oGameState.score
          x = 500  y=0
          state = func oGame,oSelf {
                oSelf { text = "Score : " + oGameState.score }
          }
        }

        text {
          animate = false
          point = 400
          size = 30
          file = "fonts/pirulen.ttf"
          text = "Energy : " + oGameState.value
          x = 10  y=0
          state = func oGame,oSelf { oSelf { text = "Energy : " + oGameState.value } }
        }
  }


func inlist nValue,aList
  for x in aList
        if x = nValue
          return true
        ok
  next
  return false

func checkwall oGame,oself,diffx,diffy
  alist = [1,2,3]
  return checkwall2(oGame,oself,diffx,diffy,aList)

func checkwall2 oGame,oself,diffx,diffy,aList
  xPos = oSelf.x + diffx
  yPos = oSelf.y + diffy
  nValue = oGame.aObjects[2].getvalue(xPos,yPos)
  nValue = inlist(nValue,aList)
  nValue = not nValue
  if nValue = 0 return nValue ok

  xPos = oSelf.x + diffx
  yPos = oSelf.y + diffy + oSelf.height
  nValue = oGame.aObjects[2].getvalue(xPos,yPos)
  nValue = inlist(nValue,aList)
  nValue = not nValue
  if nValue = 0 return nValue ok

  xPos = oSelf.x + diffx + oSelf.width
```

(continues on next page)

(continued from previous page)

```
  yPos = oSelf.y + diffy
  nValue = oGame.aObjects[2].getvalue(xPos,yPos)
  nValue = inlist(nValue,aList)
  nValue = not nValue
  if nValue = 0 return nValue ok

  xPos = oSelf.x + diffx + oSelf.width
  yPos = oSelf.y + diffy + oSelf.height
  nValue = oGame.aObjects[2].getvalue(xPos,yPos)
  nValue = inlist(nValue,aList)
  nValue = not nValue
  if nValue = 0 return nValue ok

  return nValue

func checkopenwall oGame
  if oGameState.score = 900
        oGame.aObjects[2].aMap[3][10] = 3
        oGame.aObjects[2].aMap[4][10] = 0
        oGame.aObjects[2].aMap[5][10] = 0
        oGame.aObjects[2].aMap[6][10] = 0
        oGame.aObjects[2].aMap[7][10] = 0
        oGame.aObjects[2].aMap[8][10] = 0
  but oGameState.score = 1800
        oGame.aObjects[2].aMap[3][18] = 3
        oGame.aObjects[2].aMap[4][18] = 0
        oGame.aObjects[2].aMap[5][18] = 0
        oGame.aObjects[2].aMap[6][18] = 0
        oGame.aObjects[2].aMap[7][18] = 0
        oGame.aObjects[2].aMap[8][18] = 0
  but oGameState.score = 5500
        oGame.aObjects[2].aMap[1][44] = 0
        oGame.aObjects[2].aMap[2][44] = 0
        oGame.aObjects[2].aMap[3][44] = 2
  ok


func checkgameover ogame
  if oGameState.gameresult  return ok
  if oGameState.value <= 0
        oGameState.value = 0
        oGameState.gameresult = true
        oGame {
          text {
                point = 400
                size = 30
                nStep = 9
                file = "fonts/pirulen.ttf"
                text = "Game Over !!!"
                x = 500  y=10
                state = func ogame,oself {
                  if oself.y >= 400
```

(continues on next page)

(continued from previous page)

```
                        ogame.shutdown = true
                  ok
                }
          }
        }
        showfire(oGame,oGame.aObjects[oGameState.PlayerIndex].x+40,
                 oGame.aObjects[oGameState.PlayerIndex].y+40)
        oGame.aObjects[oGameState.PlayerIndex].enabled = false
        oGame.remove(oGameState.PlayerIndex)
  ok


func showfire oGame,nX,nY
  oGame {
        animate {
          file = "images/fire.png"
          x = nX
          y = nY
          framewidth = 40
          height = 42
          nStep = 3
          transparent = true
          state = func oGame,oSelf {
                oSelf {
                  nStep--
                  if nStep = 0
                        nStep = 3
                        if frame < 13
                          frame++
                        else
                          frame=1
                          oGame.remove(oself.nIndex)
                        ok
                  ok
                }
          }
        }
  }

func addenemy oGame,xPos
  oGame {
        lbraceend = false
        sprite {
                type = ge_type_enemy
                file = "images/smenemy.png"
                transparent = true
                x = xPos y =10 width=100 height=100
                animate=true Scaled=true
                direction = GE_DIRECTION_NOMOVE
                temp = xPos
                state = func oGame,oSelf {
                  oself {
```

(continues on next page)

(continued from previous page)

```
                        x = oSelf.temp +  oGame.aObjects[2].x
                        if y < 0 y = 0 ok
                        if y > 100 y=100 ok
                        if x > SCREEN_W or x < 0 return ok
                  }

                  if random(10) = 1
                        if oGameState.gameresult return ok
                        ogame {
                          sprite {
                                type = ge_type_fire
                                file  = "images/smrocket.png"
                                scaled  = true
                                transparent = true
                                x = oself.x + 30
                                y = oself.y + oself.height+ 30
                                width = 30
                                height = 30
                                point = ogame.screen_h+30
                                nstep = 30
                                direction = ge_direction_incvertical
                                xvalue =  oGame.aObjects[2].x
                                temp = oself.x + 30 - xvalue
                                state = func oGame,oSelf {
                                  oself { x = oSelf.temp +  oGame.aObjects[2].x  }
                                  x =  oGame.aObjects[oGameState.playerindex]
                                  if oself.x >= x.x and oself.y >= x.y and
                                         oself.x <= x.x + x.width and
                                         oself.y <= x.y + x.height
                                         if oGameState.value > 0
                                           oGameState.value-=1000
                                         ok
                                         ogame.remove(oself.nindex)
                                         checkgameover(oGame)
                                  ok
                                }
                          }
                        }
                  ok
                }
          }
  }
  ogame.lbraceend = true


func checkstarskey oGame,oSelf,nValue,nRow,nCol
  switch nValue
        on 4
          oGame.aObjects[2].aMap[nRow][nCol] = 6
          oGameState.Score += 100
          checkopenwall(oGame)
          oGame { Sound {
```

(continues on next page)

(continued from previous page)

```
                once = true
                file = "sound/sfx_point.wav"
          } }
        on 5
          oGame.aObjects[2].aMap[nRow][nCol] = 0
          oGameState.DoorKey = true
          oGameState.Score += 500
          checkopenwall(oGame)
          oGame { Sound {
                once = true
                file = "sound/sfx_point.wav"
          } }
  off

func checkstarskeycol oGame,oSelf
  nValue = oGame.aObjects[2].getvalue(oSelf.x,oSelf.y)
  nRow = oGame.aObjects[2].getrow(oSelf.x,oSelf.y)
  nCol = oGame.aObjects[2].getcol(oSelf.x,oSelf.y)
  checkstarskey(oGame,oSelf,nValue,nRow,nCol)

  nValue = oGame.aObjects[2].getvalue(oSelf.x+oSelf.width,oSelf.y+oSelf.height)
  nRow = oGame.aObjects[2].getrow(oSelf.x+oSelf.width,oSelf.y+oSelf.height)
  nCol = oGame.aObjects[2].getcol(oSelf.x+oSelf.width,oSelf.y+oSelf.height)
  checkstarskey(oGame,oSelf,nValue,nRow,nCol)

  nValue = oGame.aObjects[2].getvalue(oSelf.x+oSelf.width,oSelf.y)
  nRow = oGame.aObjects[2].getrow(oSelf.x+oSelf.width,oSelf.y)
  nCol = oGame.aObjects[2].getcol(oSelf.x+oSelf.width,oSelf.y)
  checkstarskey(oGame,oSelf,nValue,nRow,nCol)

  nValue = oGame.aObjects[2].getvalue(oSelf.x,oSelf.y+oSelf.height)
  nRow = oGame.aObjects[2].getrow(oSelf.x,oSelf.y+oSelf.height)
  nCol = oGame.aObjects[2].getcol(oSelf.x,oSelf.y+oSelf.height)
  checkstarskey(oGame,oSelf,nValue,nRow,nCol)

func callenemystate oGame
  for t in oGame.aObjects
        t {
          if type = GE_TYPE_ENEMY
                call state(oGame,t)
          ok
        }
  next

Class GameState

  down = 3
  gameresult = false
  Score = 0
  startplay=false
  lastcol = 0
  playerwin = false
```

(continues on next page)

(continued from previous page)

```
DoorKey = false
playerindex = 4
value = 1000
moveplayer = false
```

Screen Shot:

CHAPTER

FIFTYSIX

BUILDING GAMES FOR ANDROID

In this chapter we will learn about Building RingLibSDL Games for Mobile.

So we can create packages (*.apk) for the applications that are developed using Ring Game Engine for 2D Games.

56.1 Download Requirements and Update the Android SDK

- The Android SDK Tools

  https://developer.android.com/studio/index.html

- The Android NDK (Tested using android-ndk-r10c)

  https://developer.android.com/ndk/index.html

- Apache Ant v1.8 or later

  http://ant.apache.org/bindownload.cgi

- Java SE Development Kit (JDK) v6 or later

  http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

- Update the Android SDK to get the API and tools packages required for development

  Tested using Android 4.4.2 (API 19)

- In Windows - Define the next Environment Variables based on your system.

(1) JAVA_HOME

```
For Example : C:\Program Files (x86)\Java\jdk1.8.0_05
```

(2) ANDROID_HOME

```
For Example : B:\mahmoud\Tools\Java-Android\adt-bundle-windows-x86-20140702\sdk
```

56.2 Project Folder

Open the project folder : ring/extensions/android/ringlibsdl/project

You can add the source code (*.ring) and Images/Sound Files to the assets folder.

You will find the Flappy Bird 3000 Game ready for building.

The execution starts from the start.ring file

```
load "game2.ring"
```

56.3 Building the project

Move to the ring/extensions/android/ringlibsdl/project folder

We can build using the next command (We need to do this for one time only).

```
ndk-build
```

Then we can create the package (*.apk) using the next command.

```
ant debug
```

We can write a batch file for building the project (file: build.bat)

```
rem You will need to modify this batch file based on your environment

set JAVA_HOME=C:\Program Files (x86)\Java\jdk1.8.0_05
set ANDROID_HOME=B:\mahmoud\Tools\JavaAndroid\adt-bundle-windows-x86-20140702\sdk
set NDK_ROOT=B:\mahmoud\Tools\JavaAndroid\android-ndk-r10c

set path=%path%;B:\mahmoud\Tools\JavaAndroid\android-ndk-r10c
set path=%path%;B:\mahmoud\Tools\JavaAndroid\apache-ant-1.9.4\bin

ndk-build
```

CHAPTER

FIFTYSEVEN

DEVELOPING GAMES USING RINGRAYLIB

In this chapter we will learn how to use the RingRayLib extension.

57.1 Introduction

RingRayLib is an extension for the RayLib game programming library.

Also RayGUI functions are supported by this extension.

57.2 Basic Window

```
load "raylib.ring"

screenWidth     = 800
screenHeight    = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - basic window")

SetTargetFPS(60)

while !WindowShouldClose()
        BeginDrawing()
        ClearBackground(RED)
        DrawText("Congrats! You created your first window!", 190, 200, 20, WHITE)
        EndDrawing()
end

CloseWindow()
```

Screen Shot:

57.3 Input Keys

```
load "raylib.ring"

screenWidth  = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - keyboard input")
ballPosition = Vector2(screenWidth/2, screenHeight/2)
SetTargetFPS(60)

while !WindowShouldClose()
                if IsKeyDown(KEY_RIGHT) ballPosition.x += 2 ok
                if IsKeyDown(KEY_LEFT)  ballPosition.x -= 2 ok
                if IsKeyDown(KEY_UP)    ballPosition.y -= 2 ok
                if IsKeyDown(KEY_DOWN)  ballPosition.y += 2 ok
                BeginDrawing()
                ClearBackground(RAYWHITE)
                DrawText("move the ball with arrow keys", 10, 10, 20, DARKGRAY)
                DrawCircleV(ballPosition, 50, MAROON)
                EndDrawing()
end

CloseWindow()
```

Screen Shot:

57.4 Input Mouse

```
load "raylib.ring"

screenWidth  = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - mouse input")

ballPosition    = Vector2(100, 100)
ballColor       = DARKBLUE

SetTargetFPS(60)

while ! WindowShouldClose()

                ballPosition = GetMousePosition()

                if IsMouseButtonPressed(MOUSE_LEFT_BUTTON)
                        ballColor = MAROON
                but IsMouseButtonPressed(MOUSE_MIDDLE_BUTTON)
                        ballColor = LIME
                but IsMouseButtonPressed(MOUSE_RIGHT_BUTTON)
                        ballColor = DARKBLUE
                ok

                BeginDrawing()
```

(continues on next page)

(continued from previous page)

```
                ClearBackground(BLACK)
                DrawCircleV(ballPosition, 40, ballColor)
                DrawText("move ball with mouse and click mouse button to change color",
                         10, 10, 20, YELLOW)
                EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.5 3D Camera

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - 3d camera mode")

camera = Camera3D(
        0, 10, 10,              // Camera position
        0, 0, 0 ,               // Camera looking at point
        0, 1, 0,                // Camera up vector (rotation towards target)
        45,                     // Camera field-of-view Y
```

(continues on next page)

(continued from previous page)

```
        CAMERA_PERSPECTIVE)     // Camera mode type

cubePosition = Vector3(0, 0, 0)

SetTargetFPS(60)

while !WindowShouldClose()

                BeginDrawing()

                ClearBackground(RAYWHITE)

                BeginMode3D(camera)

                DrawCube(cubePosition, 2, 2, 2, RED)
                DrawCubeWires(cubePosition, 2, 2, 2, MAROON)

                DrawGrid(10, 1)

                EndMode3D()

                DrawText("Welcome to the third dimension!", 10, 40, 20, DARKGRAY)

                DrawFPS(10, 10)

                EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.6 3D Camera Free

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - 3d camera free")

camera = Camera3D(
        10, 10, 10,             // Camera position
        0, 0, 0 ,               // Camera looking at point
        0, 1, 0,                // Camera up vector (rotation towards target)
        45,                     // Camera field-of-view Y
        CAMERA_PERSPECTIVE)     // Camera mode type

cubePosition = Vector3(0, 0, 0)

SetCameraMode(camera, CAMERA_FREE)              // Set a free camera mode

SetTargetFPS(60)

while !WindowShouldClose()

                UpdateCamera(camera)

                if IsKeyDown("Z") camera.target = Vector3( 0, 0, 0) ok
```

(continues on next page)

(continued from previous page)

```
                BeginDrawing()

                ClearBackground(RAYWHITE)

                BeginMode3D(camera)

                DrawCube(cubePosition, 2, 2, 2, RED)
                DrawCubeWires(cubePosition, 2, 2, 2, MAROON)

                DrawGrid(10, 1)

                EndMode3D()

                DrawRectangle( 10, 10, 320, 133, Fade(SKYBLUE, 0.5))
                DrawRectangleLines( 10, 10, 320, 133, BLUE)

                DrawText("Free camera default controls:", 20, 20, 10, BLACK)
                DrawText("- Mouse Wheel to Zoom in-out", 40, 40, 10, DARKGRAY)
                DrawText("- Mouse Wheel Pressed to Pan", 40, 60, 10, DARKGRAY)
                DrawText("- Alt + Mouse Wheel Pressed to Rotate", 40, 80, 10, DARKGRAY)
                DrawText("- Alt + Ctrl + Mouse Wheel Pressed for Smooth Zoom",
                         40, 100, 10, DARKGRAY)
                DrawText("- Z to zoom to (0, 0, 0)", 40, 120, 10, DARKGRAY)

                EndDrawing()
end

CloseWindow()
```

Screen Shot:

57.7 Mouse Wheel

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - input mouse wheel")

boxPositionY = screenHeight/2 - 40
scrollSpeed = 4

SetTargetFPS(60)

while !WindowShouldClose()
        boxPositionY -= (GetMouseWheelMove()*scrollSpeed)

        BeginDrawing()

        ClearBackground(RAYWHITE)

        DrawRectangle(screenWidth/2 - 40, boxPositionY, 80, 80, MAROON)

        DrawText("Use mouse wheel to move the cube up and down!", 10, 10, 20, GRAY)
        DrawText("Box position Y: "+boxPositionY, 10, 40, 20, LIGHTGRAY)

        EndDrawing()
```

(continues on next page)

(continued from previous page)

```
end

CloseWindow()
```

Screen Shot:

57.8 Input Multi-touch

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - input multitouch")

ballPosition = Vector2(-100, -100)
ballColor = BEIGE

touchCounter = 0
touchPosition = vector2(0,0)

MAX_TOUCH_POINTS = 5

SetTargetFPS(60)

while !WindowShouldClose()
```

(continues on next page)

(continued from previous page)

```
        ballPosition = GetMousePosition()

        ballColor = BEIGE

        if IsMouseButtonDown(MOUSE_LEFT_BUTTON) ballColor = MAROON ok
        if IsMouseButtonDown(MOUSE_MIDDLE_BUTTON) ballColor = LIME ok
        if IsMouseButtonDown(MOUSE_RIGHT_BUTTON) ballColor = DARKBLUE ok

        if IsMouseButtonPressed(MOUSE_LEFT_BUTTON) touchCounter = 10 ok
        if IsMouseButtonPressed(MOUSE_MIDDLE_BUTTON) touchCounter = 10 ok
        if IsMouseButtonPressed(MOUSE_RIGHT_BUTTON) touchCounter = 10 ok

        if touchCounter > 0 touchCounter-- ok

        BeginDrawing()

        ClearBackground(RAYWHITE)

        for i = 0 to MAX_TOUCH_POINTS-1
                touchPosition = GetTouchPosition(i)

                if touchPosition.x >= 0 && touchPosition.y >= 0
                        DrawCircleV(touchPosition, 34, ORANGE)
                        DrawText(""+ i, touchPosition.x - 10,
                                touchPosition.y - 70, 40, BLACK)
                ok
        next

        DrawCircleV(ballPosition, 30 + (touchCounter*3), ballColor)

        DrawText("move ball with mouse and click mouse button to change color",
                         10, 10, 20, DARKGRAY)
        DrawText("touch the screen at multiple locations to get multiple balls",
                         10, 30, 20, DARKGRAY)

        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.9 Camera First Person

```
load "raylib.ring"

MAX_COLUMNS = 20

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - 3d camera first person")

camera = Camera3d(
        4, 2, 4,
        0, 1, 0,
        0, 1, 0,
        60,
        CAMERA_PERSPECTIVE
)

heights = list(MAX_COLUMNS)
positions = list(MAX_COLUMNS)
for item in positions item = vector3(0,0,0) next
colors = list(MAX_COLUMNS)
for item in colors item = BLACK next

for i = 1 to  MAX_COLUMNS
        heights[i] = GetRandomValue(1, 12)
```

(continues on next page)

(continued from previous page)

```
        positions[i] = Vector3(GetRandomValue(-15, 15),
                        heights[i]/2, GetRandomValue(-15, 15) )
        colors[i] = RAYLibColor(GetRandomValue(20, 255),
                        GetRandomValue(10, 55), 30, 255 )
next

SetCameraMode(camera, CAMERA_FIRST_PERSON)

SetTargetFPS(60)

while !WindowShouldClose()

        UpdateCamera(camera)
        BeginDrawing()

        ClearBackground(RAYWHITE)

        BeginMode3D(camera)

        DrawPlane(Vector3( 0, 0, 0 ), Vector2(32, 32 ), LIGHTGRAY) // Draw ground
        DrawCube(Vector3( -16, 2.5, 0 ), 1, 5, 32, BLUE)     // Draw a blue wall
        DrawCube(Vector3( 16, 2.5, 0 ), 1, 5, 32, LIME)      // Draw a green wall
        DrawCube(Vector3( 0, 2.5, 16 ), 32, 5, 1, GOLD)      // Draw a yellow wall

        for i = 1 to  MAX_COLUMNS
                DrawCube(positions[i], 2, heights[i], 2, colors[i])
                DrawCubeWires(positions[i], 2, heights[i], 2, MAROON)
        next

        EndMode3D()

        DrawRectangle( 10, 10, 220, 70, Fade(SKYBLUE, 0.5f))
        DrawRectangleLines( 10, 10, 220, 70, BLUE)

        DrawText("First person camera default controls:", 20, 20, 10, BLACK)
        DrawText("- Move with keys: W, A, S, D", 40, 40, 10, DARKGRAY)
        DrawText("- Mouse move to look around", 40, 60, 10, DARKGRAY)

        EndDrawing()
end

CloseWindow()
```

Screen Shot:

57.10 3D Picking

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - 3d picking")

camera = Camera3D(
        10, 10, 10,
        0, 0, 0 ,
        0, 1, 0 ,
        45,
        CAMERA_PERSPECTIVE
)

cubePosition = Vector3( 0, 1, 0 )
cubeSize = Vector3( 2, 2, 2 )

ray = Ray(0,0,0,0,0,0)

collision = false

SetCameraMode(camera, CAMERA_FREE)

SetTargetFPS(60)
```

(continues on next page)

(continued from previous page)

```
while !WindowShouldClose()

        UpdateCamera(camera)

        if IsMouseButtonPressed(MOUSE_LEFT_BUTTON)
                if !collision
                        ray = GetMouseRay(GetMousePosition(), camera)
                        collision = CheckCollisionRayBox(ray,
                        BoundingBox( cubePosition.x - cubeSize.x/2,
                        cubePosition.y - cubeSize.y/2, cubePosition.z - cubeSize.z/2,
                        cubePosition.x + cubeSize.x/2, cubePosition.y + cubeSize.y/2,
                        cubePosition.z + cubeSize.z/2 ) )
                else collision = false
                ok
        ok

        BeginDrawing()

        ClearBackground(RAYWHITE)

        BeginMode3D(camera)

        if collision
                DrawCube(cubePosition, cubeSize.x, cubeSize.y, cubeSize.z, RED)
                DrawCubeWires(cubePosition, cubeSize.x, cubeSize.y, cubeSize.z, MAROON)

                DrawCubeWires(cubePosition, cubeSize.x + 0.2f,
                                cubeSize.y + 0.2f, cubeSize.z + 0.2f, GREEN)
        else
                DrawCube(cubePosition, cubeSize.x, cubeSize.y, cubeSize.z, GRAY)
                DrawCubeWires(cubePosition, cubeSize.x, cubeSize.y, cubeSize.z, DARKGRAY)
        ok

        DrawRay(ray, MAROON)
        DrawGrid(10, 1)

        EndMode3D()

        DrawText("Try selecting the box with mouse!", 240, 10, 20, DARKGRAY)

        if collision  DrawText("BOX SELECTED",
                (screenWidth - MeasureText("BOX SELECTED", 30)) / 2,
                screenHeight * 0.1f, 30, GREEN) ok

        DrawFPS(10, 10)

        EndDrawing()
end

CloseWindow()
```

Screen Shot:

57.11 Full Screen

```
load "raylib.ring"

screenWidth     = 1024
screenHeight    = 768

InitWindow(screenWidth, screenHeight, "Full Screen")
ToggleFullScreen()

SetTargetFPS(60)

while !WindowShouldClose()
        BeginDrawing()
                ClearBackground(DARKBLUE)
                DrawText("Count from 1 to 10", 190, 200, 20, Yellow)
                for t = 1 to 10
                        DrawText("Number: " + t, 190, 200+(30*t), 20, WHITE)
                next
        EndDrawing()
end

CloseWindow()
```

Screen Shot:

57.12 Two Cubes

```
load "raylib.ring"

screenWidth  = 800
screenHeight = 450
InitWindow(screenWidth, screenHeight, "raylib [core] example - Two Cubes")

camera = Camera3D(
        10, 10, 10,
        0, 0, 0 ,
        0, 1, 0 ,
        45,
        CAMERA_PERSPECTIVE
)

cubePosition1 = Vector3( 0, 1, 4 )
cubePosition2 = Vector3( 0, 1, -4 )
cubeSize = Vector3( 2, 2, 2 )

ray = Ray(0,0,0,0,0,0)
```

(continues on next page)

(continued from previous page)

```
collision1 = false
collision2 = false

SetCameraMode(camera, CAMERA_FREE)

SetTargetFPS(60)

while !WindowShouldClose()

                UpdateCamera(camera)

                if IsMouseButtonPressed(MOUSE_LEFT_BUTTON)
                        if !collision1
                                ray = GetMouseRay(GetMousePosition(), camera)

                                collision1 = CheckCollisionRayBox(ray,
                                BoundingBox( cubePosition1.x -
                                cubeSize.x/2, cubePosition1.y - cubeSize.y/2,
                                cubePosition1.z - cubeSize.z/2,
                                cubePosition1.x + cubeSize.x/2,
                                cubePosition1.y + cubeSize.y/2,
                                cubePosition1.z + cubeSize.z/2 ) )
                        else
                                collision1 = false
                        ok
                        if  !collision2
                                ray = GetMouseRay(GetMousePosition(), camera)

                                collision2 = CheckCollisionRayBox(ray,
                                BoundingBox( cubePosition2.x -
                                cubeSize.x/2, cubePosition2.y - cubeSize.y/2,
                                cubePosition2.z - cubeSize.z/2,
                                cubePosition2.x + cubeSize.x/2,
                                cubePosition2.y + cubeSize.y/2,
                                cubePosition2.z + cubeSize.z/2 ) )
                        else
                                collision2 = false
                ok
        ok

                BeginDrawing()

                        ClearBackground(RAYWHITE)

                        BeginMode3D(camera)

                                 if collision1
                                        DrawCube(cubePosition1, cubeSize.x,
                                        cubeSize.y, cubeSize.z, RED)
                                        DrawCubeWires(cubePosition1, cubeSize.x,
                                        cubeSize.y, cubeSize.z, MAROON)
```

(continues on next page)

(continued from previous page)

```
                                        DrawCubeWires(cubePosition1, cubeSize.x +
                                        0.2f, cubeSize.y + 0.2f,
                                        cubeSize.z + 0.2f, GREEN)
                                        collision1 = true
                                else
                                        DrawCube(cubePosition1, cubeSize.x,
                                        cubeSize.y, cubeSize.z, GRAY)
                                        DrawCubeWires(cubePosition1, cubeSize.x,
                                        cubeSize.y, cubeSize.z, DARKGRAY)
                                        collision1 = false
                                ok

                                if collision2
                                        DrawCube(cubePosition2, cubeSize.x,
                                        cubeSize.y, cubeSize.z, RED)
                                        DrawCubeWires(cubePosition2, cubeSize.x,
                                        cubeSize.y, cubeSize.z, MAROON)

                                        DrawCubeWires(cubePosition2, cubeSize.x +
                                        0.2f, cubeSize.y + 0.2f,
                                        cubeSize.z + 0.2f, GREEN)
                                        collision2 = true
                                else
                                        DrawCube(cubePosition2, cubeSize.x,
                                        cubeSize.y, cubeSize.z, GRAY)
                                        DrawCubeWires(cubePosition2, cubeSize.x,
                                        cubeSize.y, cubeSize.z, DARKGRAY)
                                        collision2 = false
                                ok


                                DrawRay(ray, MAROON)
                                DrawGrid(10, 1)

                        EndMode3D()

                        DrawText("Try selecting the box with mouse!", 240, 10,
                                        20, DARKGRAY)

                        if collision1 or collision2
                                DrawText("BOX SELECTED",
                                (screenWidth - MeasureText("BOX SELECTED", 30)) / 2,
                                screenHeight * 0.1f, 30, GREEN)
                        ok

                        DrawFPS(10, 10)

                EndDrawing()
end

CloseWindow()
```

Screen Shot:

57.13 Basic Shapes

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [shapes] example - basic shapes drawing")

SetTargetFPS(60)

while !WindowShouldClose()

        BeginDrawing()

        ClearBackground(RAYWHITE)

        DrawText("some basic shapes available on raylib", 20, 20, 20, DARKGRAY)

        DrawCircle(screenWidth/4, 120, 35, DARKBLUE)

        DrawRectangle(screenWidth/4*2 - 60, 100, 120, 60, RED)
        DrawRectangleLines(screenWidth/4*2 - 40, 320, 80, 60, ORANGE)
        DrawRectangleGradientH(screenWidth/4*2 - 90, 170, 180, 130, MAROON, GOLD)
```

(continues on next page)

(continued from previous page)

```
        DrawTriangle(Vector2(screenWidth/4*3, 80),
                                Vector2(screenWidth/4*3 - 60, 150),
                                Vector2(screenWidth/4*3 + 60, 150), VIOLET)

        DrawPoly(Vector2(screenWidth/4*3, 320), 6, 80, 0, BROWN)

        DrawCircleGradient(screenWidth/4, 220, 60, GREEN, SKYBLUE)

        DrawLine(18, 42, screenWidth - 18, 42, BLACK)
        DrawCircleLines(screenWidth/4, 340, 80, DARKBLUE)
        DrawTriangleLines(Vector2(screenWidth/4*3, 160),
                        Vector2(screenWidth/4*3 - 20, 230),
                        Vector2(screenWidth/4*3 + 20, 230), DARKBLUE)
        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.14 Draw Ring

```
load "raylib.ring"

screenWidth = 800       screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [shapes] example - draw ring")

center = Vector2((GetScreenWidth() - 300)/2, GetScreenHeight()/2 )

innerRadius = 80        outerRadius = 190
startAngle = 0          endAngle = 360          segments = 0
drawRing = true         drawRingLines = false   drawCircleLines = false

SetTargetFPS(60)

while !WindowShouldClose()

        BeginDrawing()

        ClearBackground(RAYWHITE)

        DrawLine(500, 0, 500, GetScreenHeight(), Fade(LIGHTGRAY, 0.6))
        DrawRectangle(500, 0, GetScreenWidth() - 500,
                         GetScreenHeight(), Fade(LIGHTGRAY, 0.3))

        if drawRing DrawRing(center, innerRadius, outerRadius,
                startAngle, endAngle, segments, Fade(MAROON, 0.3)) ok
        if drawRingLines DrawRingLines(center, innerRadius,
                outerRadius, startAngle, endAngle,
                 segments, Fade(BLACK, 0.4)) ok
        if drawCircleLines DrawCircleSectorLines(center, outerRadius,
                startAngle, endAngle, segments, Fade(BLACK, 0.4)) ok

        startAngle = GuiSliderBar(Rectangle( 600, 40, 120, 20 ),
                        "StartAngle", startAngle, -450, 450, true)
        endAngle = GuiSliderBar(Rectangle( 600, 70, 120, 20 ),
                        "EndAngle", endAngle, -450, 450, true)

        innerRadius = GuiSliderBar(Rectangle( 600, 140, 120, 20 ),
                        "InnerRadius", innerRadius, 0, 100, true)
        outerRadius = GuiSliderBar(Rectangle( 600, 170, 120, 20 ),
                        "OuterRadius", outerRadius, 0, 200, true)

        segments = GuiSliderBar(Rectangle( 600, 240, 120, 20 ),
                        "Segments", segments, 0, 100, true)

        drawRing = GuiCheckBox(Rectangle( 600, 320, 20, 20 ),
                        "Draw Ring", drawRing)
        drawRingLines = GuiCheckBox(Rectangle( 600, 350, 20, 20 ),
                        "Draw RingLines", drawRingLines)
        drawCircleLines = GuiCheckBox(Rectangle( 600, 380, 20, 20 ),
```

(continues on next page)

(continued from previous page)

```
                        "Draw CircleLines", drawCircleLines)

        if segments >= 4
                DrawText("MODE: MANUAL", 600, 270, 10, MAROON)
        else
                DrawText("MODE: AUTO", 600, 270, 10, DARKGRAY)
        ok

        DrawFPS(10, 10)

        EndDrawing()
end

CloseWindow()
```

Screen Shot:

Screen Shot (2):

57.15 Bezier Lines

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

SetConfigFlags(FLAG_MSAA_4X_HINT)
InitWindow(screenWidth, screenHeight, "raylib [shapes] example - cubic-bezier lines")

start = Vector2(0,0)
endvec = Vector2(screenWidth,screenHeight)

SetTargetFPS(60)

while (!WindowShouldClose())

        if (IsMouseButtonDown(MOUSE_LEFT_BUTTON))
                start = GetMousePosition()
        else (IsMouseButtonDown(MOUSE_RIGHT_BUTTON))
                  endvec = GetMousePosition()
        ok


        BeginDrawing()

        ClearBackground(RAYWHITE)
```

(continues on next page)

(continued from previous page)

```
        DrawText("USE MOUSE LEFT-RIGHT CLICK to DEFINE LINE START and END POINTS",
                         15, 20, 20, GRAY)
        DrawLineBezier(start, endvec, 2.0, RED)

        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.16 Collision Area

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [shapes] example - collision area")

// Box A: Moving box
boxA = Rectangle( 10, GetScreenHeight()/2 - 50, 200, 100 )
boxASpeedX = 4

// Box B: Mouse moved box
```

(continues on next page)

(continued from previous page)

```
boxB = Rectangle( GetScreenWidth()/2 - 30, GetScreenHeight()/2 - 30, 60, 60 )

boxCollision = GetCollisionRec(boxA, boxB)

boxCollision = Rectangle( 0,0,0,0 ) // Collision rectangle

screenUpperLimit = 40      // Top menu limits

pause = false             // Movement pause
collision = false         // Collision detection

SetTargetFPS(60)

while !WindowShouldClose()

        // Move box if not paused
        if (not pause) boxA.x += boxASpeedX ok

        // Bounce box on x screen limits
        if (((boxA.x + boxA.width) >= GetScreenWidth()) or (boxA.x <= 0))
                boxASpeedX = boxASpeedX*(-1) ok

        // Update player-controlled-box (box02)
        boxB.x = GetMouseX() - boxB.width/2
        boxB.y = GetMouseY() - boxB.height/2

        // Make sure Box B does not go out of move area limits
        if ((boxB.x + boxB.width) >= GetScreenWidth())
                boxB.x = GetScreenWidth() - boxB.width
        else (boxB.x <= 0) boxB.x = 0 ok

        if ((boxB.y + boxB.height) >= GetScreenHeight())
                boxB.y = GetScreenHeight() - boxB.height
        else (boxB.y <= screenUpperLimit) boxB.y = screenUpperLimit ok

        // Check boxes collision
        collision = CheckCollisionRecs(boxA, boxB)

        // Get collision rectangle (only on collision)
        if (collision) boxCollision = GetCollisionRec(boxA, boxB) ok

        // Pause Box A movement
        if (IsKeyPressed(KEY_SPACE)) pause = not pause ok

        BeginDrawing()

                ClearBackground(RAYWHITE)

                if collision = true
                   color = RED
                else
                   color = BLACK
```

(continues on next page)

(continued from previous page)

```
                ok
                DrawRectangle(0, 0, screenWidth, screenUpperLimit, color)
                DrawRectangleRec(boxA, GOLD)

                boxB.x = GetMouseX() - boxB.width/2
                boxB.y = GetMouseY() - boxB.height/2
                collision = CheckCollisionRecs(boxA, boxB)
                DrawRectangleRec(boxB, BLUE)
                boxCollision = GetCollisionRec(boxA, boxB)

                if (collision) = true

                        // Draw collision area
                        DrawRectangleRec(boxCollision, LIME)

                        // Draw collision message
                        DrawText("COLLISION!", GetScreenWidth()/2 -
                                         MeasureText("COLLISION!", 20)/2,
                                        screenUpperLimit/2 - 10, 20, BLACK)

                        // Draw collision area
                        DrawText("Collision Area: " +
                                string(boxCollision.width*boxCollision.height),
                                GetScreenWidth()/2 - 100,
                                screenUpperLimit + 10, 20, BLACK)

                ok

                DrawFPS(10, 10)

        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.17 Following Eyes

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [shapes] example - following eyes")

scleraLeftPosition = Vector2( GetScreenWidth()/2 - 100, GetScreenHeight()/2 )
scleraRightPosition = Vector2( GetScreenWidth()/2 + 100, GetScreenHeight()/2 )
scleraRadius = 80

irisLeftPosition = Vector2( GetScreenWidth()/2 - 100, GetScreenHeight()/2 )
irisRightPosition = Vector2( GetScreenWidth()/2 + 100, GetScreenHeight()/2 )
irisRadius = 24

angle = 0.0
dx = 0.0 dy = 0.0 dxx = 0.0 dyy = 0.0

SetTargetFPS(60)

while !WindowShouldClose()

        irisLeftPosition = GetMousePosition()
        irisRightPosition = GetMousePosition()
```

(continues on next page)

(continued from previous page)

```
        // Check not inside the left eye sclera
        if !CheckCollisionPointCircle(irisLeftPosition,
                scleraLeftPosition, scleraRadius - 20)
                dx = irisLeftPosition.x - scleraLeftPosition.x
                dy = irisLeftPosition.y - scleraLeftPosition.y

                angle = atan2(dy, dx)

                dxx = (scleraRadius - irisRadius)*cos(angle)
                dyy = (scleraRadius - irisRadius)*sin(angle)

                irisLeftPosition.x = scleraLeftPosition.x + dxx
                irisLeftPosition.y = scleraLeftPosition.y + dyy
        ok

        // Check not inside the right eye sclera
        if !CheckCollisionPointCircle(irisRightPosition,
                 scleraRightPosition, scleraRadius - 20)
                dx = irisRightPosition.x - scleraRightPosition.x
                dy = irisRightPosition.y - scleraRightPosition.y

                angle = atan2(dy, dx)

                dxx = (scleraRadius - irisRadius)*cos(angle)
                dyy = (scleraRadius - irisRadius)*sin(angle)

                irisRightPosition.x = scleraRightPosition.x + dxx
                irisRightPosition.y = scleraRightPosition.y + dyy
        ok

        BeginDrawing()

                ClearBackground(RAYWHITE)

                DrawCircleV(scleraLeftPosition, scleraRadius, LIGHTGRAY)
                DrawCircleV(irisLeftPosition, irisRadius, BROWN)
                DrawCircleV(irisLeftPosition, 10, BLACK)

                DrawCircleV(scleraRightPosition, scleraRadius, LIGHTGRAY)
                DrawCircleV(irisRightPosition, irisRadius, DARKGREEN)
                DrawCircleV(irisRightPosition, 10, BLACK)

                DrawFPS(10, 10)

        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.18 Colors Palette

```
load "raylib.ring"

MAX_COLORS_COUNT = 21          // Number of colors available

screenWidth = 800
screenHeight = 450
colors = list(MAX_COLORS_COUNT)
colorNames = list(MAX_COLORS_COUNT)
colorsRecs = list(MAX_COLORS_COUNT)
colorState = list(MAX_COLORS_COUNT)

InitWindow(screenWidth, screenHeight, "raylib [shapes] example - colors palette")

colors = [
        DARKGRAY, MAROON, ORANGE, DARKGREEN, DARKBLUE, DARKPURPLE, DARKBROWN,
        GRAY, RED, GOLD, LIME, BLUE, VIOLET, BROWN, LIGHTGRAY, PINK, YELLOW,
        GREEN, SKYBLUE, PURPLE, BEIGE ]

colorNames = [
        "DARKGRAY", "MAROON", "ORANGE", "DARKGREEN", "DARKBLUE", "DARKPURPLE",
        "DARKBROWN", "GRAY", "RED", "GOLD", "LIME", "BLUE", "VIOLET", "BROWN",
        "LIGHTGRAY", "PINK", "YELLOW", "GREEN", "SKYBLUE", "PURPLE", "BEIGE" ]

for i = 1 to MAX_COLORS_COUNT
        colorsRecs[i] = new Rectangle(0,0,0,0)
```

(continues on next page)

(continued from previous page)

```
next

for i = 1 to MAX_COLORS_COUNT
        colorState[i] = 0
next


// Fills colorsRecs data (for every rectangle)
for i = 1 to MAX_COLORS_COUNT
        colorsRecs[i].x = 20 + 100*((i-1)%7) + 10*((i-1)%7)
        colorsRecs[i].y = 80 + 100*floor((i-1)/7) + 10*floor((i-1)/7)
        colorsRecs[i].width = 100
        colorsRecs[i].height = 100
next

mousePoint = Vector2( 0.0, 0.0 )

SetTargetFPS(60)

// Main game loop
while !WindowShouldClose()

        mousePoint = GetMousePosition()

        for i = 1 to MAX_COLORS_COUNT
                if (CheckCollisionPointRec(mousePoint,
                         colorsRecs[i])) colorState[i] = 1
                else colorState[i] = 0 ok
        next

        BeginDrawing()

                ClearBackground(RAYWHITE)

                DrawText("raylib colors palette", 28, 42, 20, BLACK)
                DrawText("press SPACE to see all colors",
                                 GetScreenWidth() - 180,
                                 GetScreenHeight() - 40, 10, GRAY)

                for i = 1 to MAX_COLORS_COUNT    // Draw all rectangles
                        if colorState[i]
                           cstate = 0.6
                        else
                           cstate = 1.0
                        ok

                        DrawRectangleRec(colorsRecs[i], Fade(colors[i], cstate))

                        if (IsKeyDown(KEY_SPACE) || colorState[i])
                                DrawRectangle(colorsRecs[i].x,
                                colorsRecs[i].y + colorsRecs[i].height - 26,
                                colorsRecs[i].width, 20, BLACK)
```

(continues on next page)

(continued from previous page)

```
                                DrawRectangleLinesEx(colorsRecs[i], 6,
                                Fade(BLACK, 0.3f))
                                DrawText(colorNames[i], colorsRecs[i].x +
                                colorsRecs[i].width - MeasureText(colorNames[i],
                                10) - 12, colorsRecs[i].y +
                                colorsRecs[i].height - 20, 10, colors[i])
                        ok
                next

        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.19 Rectangle Scaling

```
load "raylib.ring"

MOUSE_SCALE_MARK_SIZE = 12

screenWidth = 800
screenHeight = 450
```

(continues on next page)

(continued from previous page)

```
InitWindow(screenWidth, screenHeight,
                 "raylib [shapes] example - rectangle scaling mouse")

rec = Rectangle( 100, 100, 200, 80 )

mousePosition = Vector2( 0,0 )

mouseScaleReady = false
mouseScaleMode = false

SetTargetFPS(60)

while !WindowShouldClose()

        mousePosition = GetMousePosition()

        if (CheckCollisionPointRec(mousePosition, rec) and
                CheckCollisionPointRec(mousePosition,
                 Rectangle(rec.x + rec.width - MOUSE_SCALE_MARK_SIZE,
                        rec.y + rec.height - MOUSE_SCALE_MARK_SIZE,
                         MOUSE_SCALE_MARK_SIZE, MOUSE_SCALE_MARK_SIZE )))
                mouseScaleReady = true
                if (IsMouseButtonPressed(MOUSE_LEFT_BUTTON))
                        mouseScaleMode = true ok
        else mouseScaleReady = false ok

        if (mouseScaleMode)

                mouseScaleReady = true

                rec.width = (mousePosition.x - rec.x)
                rec.height = (mousePosition.y - rec.y)

                if (rec.width < MOUSE_SCALE_MARK_SIZE)
                        rec.width = MOUSE_SCALE_MARK_SIZE ok
                if (rec.height < MOUSE_SCALE_MARK_SIZE)
                        rec.height = MOUSE_SCALE_MARK_SIZE ok

                if (IsMouseButtonReleased(MOUSE_LEFT_BUTTON))
                        mouseScaleMode = false ok
        ok

        BeginDrawing()

                ClearBackground(RAYWHITE)

                DrawText("Scale rectangle dragging from bottom-right corner!",
                                 10, 10, 20, GRAY)

                DrawRectangleRec(rec, Fade(GREEN, 0.5f))

                if (mouseScaleReady)
```

(continues on next page)

(continued from previous page)

```
                        DrawRectangleLinesEx(rec, 1, RED)
                        DrawTriangle(Vector2( rec.x + rec.width -
                        MOUSE_SCALE_MARK_SIZE, rec.y + rec.height ),
                        Vector2( rec.x + rec.width, rec.y + rec.height ),
                        Vector2( rec.x + rec.width, rec.y + rec.height -
                                 MOUSE_SCALE_MARK_SIZE ), RED)
                ok

        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.20 Music Playing

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight,
         "raylib [audio] example - music playing (streaming)")
```

(continues on next page)

(continued from previous page)

```
InitAudioDevice()

music = LoadMusicStream("guitar_noodling.ogg")

PlayMusicStream(music)

timePlayed = 0.0
pause = false

SetTargetFPS(60)


while !WindowShouldClose()

        UpdateMusicStream(music)

        if IsKeyPressed(KEY_SPACE)
                StopMusicStream(music)
                PlayMusicStream(music)
        ok

        if IsKeyPressed(KEY_P)
                pause = !pause

                if pause
                        PauseMusicStream(music)
                else
                        ResumeMusicStream(music)
                ok
        ok

        timePlayed = GetMusicTimePlayed(music) / GetMusicTimeLength(music) *400

        if timePlayed > 400
                StopMusicStream(music)
        ok

        BeginDrawing()

                ClearBackground(RAYWHITE)

                DrawText("MUSIC SHOULD BE PLAYING!", 255, 150, 20, LIGHTGRAY)

                         DrawRectangle(200, 200, 400, 12, LIGHTGRAY)
                         DrawRectangle(200, 200, timePlayed, 12, MAROON)
                DrawRectangleLines(200, 200, 400, 12, GRAY)

                DrawText("PRESS SPACE  TO RESTART MUSIC", 215, 250, 20, LIGHTGRAY)
                DrawText("PRESS P TO PAUSE/RESUME MUSIC", 208, 280, 20, LIGHTGRAY)

        EndDrawing()
```

(continues on next page)

(continued from previous page)

```
end

UnloadMusicStream(music)

CloseAudioDevice()

CloseWindow()
```

Screen Shot:

57.21 Sound Loading

```
load "raylib.ring"

screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight,
                "raylib [audio] example - sound loading and playing")

InitAudioDevice()

fxWav = LoadSound("sound.wav")
fxOgg = LoadSound("tanatana.ogg")
```

(continues on next page)

(continued from previous page)

```
SetTargetFPS(60)

while !WindowShouldClose()

        if IsKeyPressed(KEY_SPACE) PlaySound(fxWav)  ok
        if IsKeyPressed(KEY_ENTER) PlaySound(fxOgg)  ok

        BeginDrawing()

                ClearBackground(RAYWHITE)

                DrawText("Press SPACE to PLAY the WAV sound!", 200, 180, 20, LIGHTGRAY)
                DrawText("Press ENTER to PLAY the OGG sound!", 200, 220, 20, LIGHTGRAY)

        EndDrawing()

end

UnloadSound(fxWav)
UnloadSound(fxOgg)

CloseAudioDevice()

CloseWindow()
```

Screen Shot:

57.22 Image Drawing

```
load "raylib.ring"

screenWidth  = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight,
                 "raylib [textures] example - image drawing")

cat = LoadImage("cat.png")

ImageCrop( cat,  Rectangle( 100, 10, 280, 380 ))
ImageFlipHorizontal( cat)
ImageResize( cat, 150, 200)

parrots = LoadImage("parrots.png")

ImageDraw( parrots, cat, Rectangle( 0, 0, cat.width, cat.height ),
                        Rectangle( 30, 40, cat.width*1.5, cat.height*1.5 ))
ImageCrop( parrots, Rectangle( 0, 50, parrots.width, parrots.height - 100 ))

UnloadImage(cat)

font = LoadFont("custom_jupiter_crash.png")

ImageDrawTextEx(parrots, Vector2( 300, 230 ), font,
                 "PARROTS & CAT", font.baseSize, -2, WHITE)
UnloadFont(font);

texture = LoadTextureFromImage(parrots)

UnloadImage(parrots)

SetTargetFPS(60)

while !WindowShouldClose()
        BeginDrawing()

        ClearBackground(RAYWHITE)

        DrawTexture(texture, screenWidth/2 - texture.width/2,
                screenHeight/2 - texture.height/2 - 40, WHITE)
        DrawRectangleLines(screenWidth/2 - texture.width/2,
                screenHeight/2 - texture.height/2 - 40,
                 texture.width, texture.height, DARKGRAY)

        DrawText("We are drawing only one texture from various images composed!",
                                240, 350, 10, DARKGRAY)
        DrawText("Source images have been cropped, scaled,"+
                        " flipped and copied one over the other.",
                                190, 370, 10, DARKGRAY)
```

(continues on next page)

(continued from previous page)

```
        EndDrawing()
end

UnloadTexture(texture)

CloseWindow()
```

Screen Shot:

57.23 Image Generation

```
load "raylib.ring"

NUM_TEXTURES = 7
textures     = list(7)

screenWidth  = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight,
         "raylib [textures] example - procedural images generation")

verticalGradient   = GenImageGradientV(screenWidth, screenHeight, RED, BLUE)
horizontalGradient = GenImageGradientH(screenWidth, screenHeight, RED, BLUE)
radialGradient     = GenImageGradientRadial(screenWidth,
```

(continues on next page)

(continued from previous page)

```
                        screenHeight, 0.0, WHITE, BLACK)
checked             = GenImageChecked(screenWidth, screenHeight,
                        32, 32, RED, BLUE)
whiteNoise          = GenImageWhiteNoise(screenWidth, screenHeight, 0.5)
perlinNoise         = GenImagePerlinNoise(screenWidth, screenHeight, 50, 50, 4.0)
cellular            = GenImageCellular(screenWidth, screenHeight, 32)

textures[NUM_TEXTURES] =  0

textures[1] = LoadTextureFromImage(verticalGradient)
textures[2] = LoadTextureFromImage(horizontalGradient)
textures[3] = LoadTextureFromImage(radialGradient)
textures[4] = LoadTextureFromImage(checked)
textures[5] = LoadTextureFromImage(whiteNoise)
textures[6] = LoadTextureFromImage(perlinNoise)
textures[7] = LoadTextureFromImage(cellular)

UnloadImage(verticalGradient)
UnloadImage(horizontalGradient)
UnloadImage(radialGradient)
UnloadImage(checked)
UnloadImage(whiteNoise)
UnloadImage(perlinNoise)
UnloadImage(cellular)

currentTexture = 1

SetTargetFPS(10)

while !WindowShouldClose()

        if IsMouseButtonPressed(MOUSE_LEFT_BUTTON) || IsKeyPressed(KEY_RIGHT)
                currentTexture++
                if currentTexture > NUM_TEXTURES  currentTexture = 1 ok
        ok

        BeginDrawing()

                ClearBackground(RAYWHITE)

                DrawTexture(textures[currentTexture], 0, 0, WHITE)

                DrawRectangle(30, 400, 325, 30, Fade(SKYBLUE, 0.5))
                DrawRectangleLines(30, 400, 325, 30, Fade(WHITE, 0.5))
                DrawText("MOUSE LEFT BUTTON to CYCLE PROCEDURAL TEXTURES",
                         40, 410, 10, WHITE)

                switch(currentTexture)
                        on 1  DrawText("VERTICAL GRADIENT", 560, 10, 20, RAYWHITE)
                        on 2  DrawText("HORIZONTAL GRADIENT", 540, 10, 20, RAYWHITE)
                        on 3  DrawText("RADIAL GRADIENT", 580, 10, 20, LIGHTGRAY)
                        on 4  DrawText("CHECKED", 680, 10, 20, RAYWHITE)
```

(continues on next page)

(continued from previous page)

```
                        on 5  DrawText("WHITE NOISE", 640, 10, 20, RED)
                        on 6  DrawText("PERLIN NOISE", 630, 10, 20, RAYWHITE)
                        on 7  DrawText("CELLULAR", 670, 10, 20, RAYWHITE)
                off

        EndDrawing()

end

for i = 1 to  NUM_TEXTURES
        UnloadTexture( textures[i] )
next

CloseWindow()
```

Screen Shot:

57.24 Texture Source

```
load "raylib.ring"

screenWidth = 800
screenHeight = 800

InitWindow(screenWidth, screenHeight,
        "raylib [textures] examples - texture source and destination rectangles")
```

(continues on next page)

(continued from previous page)

```
// NOTE: Textures MUST be loaded after Window initialization
// (OpenGL context is required)

scarfy = LoadTexture("RingLogo.png")        // Texture loading

frameWidth = scarfy.width
frameHeight = scarfy.height

// Source rectangle (part of the texture to use for drawing)
sourceRec = Rectangle( 0.0, 0.0, frameWidth, frameHeight )

// Destination rectangle (screen rectangle where drawing part of texture)
destRec = Rectangle( screenWidth/2, screenHeight/2, frameWidth*2, frameHeight*2 )

// Origin of the texture (rotation/scale point)
// it's relative to destination rectangle size
origin = Vector2( frameWidth, frameHeight )

rotation = 0

SetTargetFPS(60)

while !WindowShouldClose()

        rotation = rotation+1

        BeginDrawing()

                ClearBackground(RAYWHITE)

                DrawTexturePro(scarfy, sourceRec, destRec,
                                origin, rotation, WHITE)

                DrawLine(destRec.x, 0, destRec.x, screenHeight, GRAY)
                DrawLine(0, destRec.y, screenWidth, destRec.y, GRAY)

                DrawText("(c) Scarfy sprite by Eiden Marsal",
                         screenWidth - 200, screenHeight - 20, 10, GRAY)

        EndDrawing()

end

UnloadTexture(scarfy)        // Texture unloading

CloseWindow()
```

Screen Shot:

57.25 Geometric Shapes

```
load "raylib.ring"

FOVY_PERSPECTIVE   =  45.0
WIDTH_ORTHOGRAPHIC =  10.0

screenWidth  = 800
screenHeight = 450
```

(continues on next page)

(continued from previous page)

```
InitWindow(screenWidth, screenHeight,
         "raylib [models] example - geometric shapes")

camera = Camera3D(  0.0, 10.0, 10.0,
                                0.0, 0.0, 0.0,
                                0.0, 1.0, 0.0,
                                FOVY_PERSPECTIVE, CAMERA_PERSPECTIVE
                        )

SetTargetFPS(60)

while !WindowShouldClose()

        if IsKeyPressed(KEY_SPACE)

                if camera.type = CAMERA_PERSPECTIVE

                        camera.fovy = WIDTH_ORTHOGRAPHIC
                        camera.type = CAMERA_ORTHOGRAPHIC

                else

                        camera.fovy = FOVY_PERSPECTIVE
                        camera.type = CAMERA_PERSPECTIVE
                ok
        ok

        BeginDrawing()

        ClearBackground(RAYWHITE)

        BeginMode3D(camera)

        DrawCube(Vector3(-4.0, 0.0,  2.0), 2.0, 5.0, 2.0, RED)
        DrawCubeWires(Vector3(-4.0, 0.0,  2.0), 2.0, 5.0, 2.0, GOLD)
        DrawCubeWires(Vector3(-4.0, 0.0, -2.0), 3.0, 6.0, 2.0, MAROON)

        DrawSphere(Vector3(-1.0, 0.0, -2.0), 1.0, GREEN)
        DrawSphereWires(Vector3( 1.0, 0.0,  2.0), 2.0, 16, 16, LIME)

        DrawCylinder(Vector3(4.0, 0.0, -2.0), 1.0, 2.0, 3.0, 4, SKYBLUE)
        DrawCylinderWires(Vector3(4.0, 0.0, -2.0), 1.0, 2.0, 3.0, 4, DARKBLUE)
        DrawCylinderWires(Vector3(4.5, -1.0, 2.0), 1.0, 1.0, 2.0, 6, BROWN)

        DrawCylinder(Vector3(1.0, 0.0, -4.0), 0.0, 1.5, 3.0, 8, GOLD)
        DrawCylinderWires(Vector3(1.0, 0.0, -4.0), 0.0, 1.5, 3.0, 8, PINK)

        DrawGrid(10, 1.0)

        EndMode3D()
```

(continues on next page)

(continued from previous page)

```
        DrawText("Press Spacebar to switch camera type", 10,
                         GetScreenHeight() - 30, 20, DARKGRAY)

        if camera.type = CAMERA_ORTHOGRAPHIC
                DrawText("ORTHOGRAPHIC", 10, 40, 20, BLACK)
        else
                if camera.type = CAMERA_PERSPECTIVE
                        DrawText("PERSPECTIVE", 10, 40, 20, BLACK)
                ok
        ok

        DrawFPS(10, 10)

        EndDrawing()

end

CloseWindow()
```

Screen Shot:

57.26 Cubic Map

```
load "raylib.ring"

screenWidth  = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight,
         "raylib [models] example - cubesmap loading and drawing")

camera = Camera3D( 16.0, 14.0, 16.0,
                                 0.0, 0.0, 0.0,
                                 0.0, 1.0, 0.0,
                                 45.0, 0 )

image    = LoadImage("cubicmap.png")
cubicmap = LoadTextureFromImage(image)

mesh     = GenMeshCubicmap(image, Vector3( 1.0, 1.0, 1.0 ))
model    = LoadModelFromMesh(mesh)

texture  = LoadTexture("cubicmap_atlas.png")

setmodelmaterialtexture(model,0,MAP_DIFFUSE,texture)

mapPosition = Vector3( -16.0, 0.0, -8.0 )

UnloadImage(image)
SetCameraMode(camera, CAMERA_ORBITAL)
SetTargetFPS(60)

while !WindowShouldClose()

        UpdateCamera(camera)

        BeginDrawing()

                ClearBackground(RAYWHITE)
                BeginMode3D(camera)
                        DrawModel(model, mapPosition, 1.0, WHITE)
                EndMode3D()

                DrawTextureEx(cubicmap, Vector2( screenWidth -
                                 cubicmap.width*4 - 20, 20 ),
                                               0.0, 4.0, WHITE)
                DrawRectangleLines(screenWidth - cubicmap.width*4 -
                                 20, 20, cubicmap.width*4,
                                cubicmap.height*4, GREEN)

                DrawText("cubicmap image used to", 658,  90, 10, GRAY)
                DrawText("generate map 3d model",  658, 104, 10, GRAY)
                DrawFPS(10, 10)
```

(continues on next page)

(continued from previous page)

```
        EndDrawing()

end

UnloadTexture(cubicmap)
UnloadTexture(texture)
UnloadModel(model)

CloseWindow()
```

Screen Shot:

57.27 Functions

```
void InitWindow(int width, int height, const char *title)
bool WindowShouldClose(void)
void CloseWindow(void)
bool IsWindowReady(void)
bool IsWindowMinimized(void)
bool IsWindowResized(void)
bool IsWindowHidden(void)
void ToggleFullscreen(void)
void UnhideWindow(void)
void HideWindow(void)
void SetWindowIcon(Image image)
```

(continues on next page)

(continued from previous page)

```
void SetWindowTitle(const char *title)
void SetWindowPosition(int x, int y)
void SetWindowMonitor(int monitor)
void SetWindowMinSize(int width, int height)
void SetWindowSize(int width, int height)
void *GetWindowHandle(void)
int GetScreenWidth(void)
int GetScreenHeight(void)
int GetMonitorCount(void)
int GetMonitorWidth(int monitor)
int GetMonitorHeight(int monitor)
int GetMonitorPhysicalWidth(int monitor)
int GetMonitorPhysicalHeight(int monitor)
const char *GetMonitorName(int monitor)
const char *GetClipboardText(void)
void SetClipboardText(const char *text)
void ShowCursor(void)
void HideCursor(void)
bool IsCursorHidden(void)
void EnableCursor(void)
void DisableCursor(void)
void ClearBackground(Color color)
void BeginDrawing(void)
void EndDrawing(void)
void BeginMode2D(Camera2D camera)
void EndMode2D(void)
void BeginMode3D(Camera3D camera)
void EndMode3D(void)
void BeginTextureMode(RenderTexture2D target)
void EndTextureMode(void)
Ray GetMouseRay(Vector2 mousePosition, Camera3D camera)
Vector2 GetWorldToScreen(Vector3 position, Camera3D camera)
Matrix GetCameraMatrix(Camera3D camera)
void SetTargetFPS(int fps)
int GetFPS(void)
float GetFrameTime(void)
double GetTime(void)
int ColorToInt(Color color)
Vector4 ColorNormalize(Color color)
Vector3 ColorToHSV(Color color)
Color ColorFromHSV(Vector3 hsv)
Color GetColor(int hexValue)
Color Fade(Color color, float alpha)
void SetConfigFlags(unsigned char flags)
void SetTraceLogLevel(int logType)
void SetTraceLogExit(int logType)
void SetTraceLogCallback(TraceLogCallback callback)
void TraceLog(int logType, const char *text)
void TakeScreenshot(const char *fileName)
int GetRandomValue(int min, int max)
bool FileExists(const char *fileName)
bool IsFileExtension(const char *fileName, const char *ext)
```

(continues on next page)

(continued from previous page)

```
const char *GetExtension(const char *fileName)
const char *GetFileName(const char *filePath)
const char *GetFileNameWithoutExt(const char *filePath)
const char *GetDirectoryPath(const char *fileName)
const char *GetWorkingDirectory(void)
char **GetDirectoryFiles(const char *dirPath, int *count)
void ClearDirectoryFiles(void)
bool ChangeDirectory(const char *dir)
bool IsFileDropped(void)
char **GetDroppedFiles(int *count)
void ClearDroppedFiles(void)
long GetFileModTime(const char *fileName)
void StorageSaveValue(int position, int value)
int StorageLoadValue(int position)
void OpenURL(const char *url)
bool IsKeyPressed(int key)
bool IsKeyDown(int key)
bool IsKeyReleased(int key)
bool IsKeyUp(int key)
int GetKeyPressed(void)
void SetExitKey(int key)
bool IsGamepadAvailable(int gamepad)
bool IsGamepadName(int gamepad, const char *name)
const char *GetGamepadName(int gamepad)
bool IsGamepadButtonPressed(int gamepad, int button)
bool IsGamepadButtonDown(int gamepad, int button)
bool IsGamepadButtonReleased(int gamepad, int button)
bool IsGamepadButtonUp(int gamepad, int button)
int GetGamepadButtonPressed(void)
int GetGamepadAxisCount(int gamepad)
float GetGamepadAxisMovement(int gamepad, int axis)
bool IsMouseButtonPressed(int button)
bool IsMouseButtonDown(int button)
bool IsMouseButtonReleased(int button)
bool IsMouseButtonUp(int button)
int GetMouseX(void)
int GetMouseY(void)
Vector2 GetMousePosition(void)
void SetMousePosition(int x, int y)
void SetMouseOffset(int offsetX, int offsetY)
void SetMouseScale(float scaleX, float scaleY)
int GetMouseWheelMove(void)
int GetTouchX(void)
int GetTouchY(void)
Vector2 GetTouchPosition(int index)
void SetGesturesEnabled(unsigned int gestureFlags)
bool IsGestureDetected(int gesture)
int GetGestureDetected(void)
int GetTouchPointsCount(void)
float GetGestureHoldDuration(void)
Vector2 GetGestureDragVector(void)
float GetGestureDragAngle(void)
```

(continues on next page)

(continued from previous page)

```
Vector2 GetGesturePinchVector(void)
float GetGesturePinchAngle(void)
void SetCameraMode(Camera3D camera, int mode)
void UpdateCamera(Camera3D *camera)
void SetCameraPanControl(int panKey)
void SetCameraAltControl(int altKey)
void SetCameraSmoothZoomControl(int szKey)
void SetCameraMoveControls(int frontKey, int backKey, int rightKey, int leftKey,
         int upKey, int downKey)
void DrawPixel(int posX, int posY, Color color)
void DrawPixelV(Vector2 position, Color color)
void DrawLine(int startPosX, int startPosY, int endPosX, int endPosY, Color color)
void DrawLineV(Vector2 startPos, Vector2 endPos, Color color)
void DrawLineEx(Vector2 startPos, Vector2 endPos, float thick, Color color)
void DrawLineBezier(Vector2 startPos, Vector2 endPos, float thick, Color color)
void DrawLineStrip(Vector2 *points, int numPoints, Color color)
void DrawCircle(int centerX, int centerY, float radius, Color color)
void DrawCircleSector(Vector2 center, float radius, int startAngle, int endAngle,
         int segments, Color color)
void DrawCircleSectorLines(Vector2 center, float radius, int startAngle,
        int endAngle, int segments, Color color)
void DrawCircleGradient(int centerX, int centerY, float radius,
        Color color1, Color color2)
void DrawCircleV(Vector2 center, float radius, Color color)
void DrawCircleLines(int centerX, int centerY, float radius, Color color)
void DrawRing(Vector2 center, float innerRadius, float outerRadius,
         int startAngle, int endAngle, int segments, Color color)
void DrawRingLines(Vector2 center, float innerRadius, float outerRadius,
         int startAngle, int endAngle, int segments, Color color)
void DrawRectangle(int posX, int posY, int width, int height, Color color)
void DrawRectangleV(Vector2 position, Vector2 size, Color color)
void DrawRectangleRec(Rectangle rec, Color color)
void DrawRectanglePro(Rectangle rec, Vector2 origin,
        float rotation, Color color)
void DrawRectangleGradientV(int posX, int posY, int width,
        int height, Color color1, Color color2)
void DrawRectangleGradientH(int posX, int posY, int width,
        int height, Color color1, Color color2)
void DrawRectangleGradientEx(Rectangle rec, Color col1,
        Color col2, Color col3, Color col4)
void DrawRectangleLines(int posX, int posY, int width,
        int height, Color color)
void DrawRectangleLinesEx(Rectangle rec, int lineThick, Color color)
void DrawRectangleRounded(Rectangle rec, float roundness,
        int segments, Color color)
void DrawRectangleRoundedLines(Rectangle rec, float roundness,
        int segments, int lineThick, Color color)
void DrawTriangle(Vector2 v1, Vector2 v2, Vector2 v3, Color color)
void DrawTriangleLines(Vector2 v1, Vector2 v2, Vector2 v3, Color color)
void DrawTriangleFan(Vector2 *points, int numPoints, Color color)
void DrawPoly(Vector2 center, int sides, float radius,
        float rotation, Color color)
```

(continues on next page)

(continued from previous page)

```
void SetShapesTexture(Texture2D texture, Rectangle source)
bool CheckCollisionRecs(Rectangle rec1, Rectangle rec2)
bool CheckCollisionCircles(Vector2 center1, float radius1,
        Vector2 center2, float radius2)
bool CheckCollisionCircleRec(Vector2 center, float radius,
        Rectangle rec)
Rectangle GetCollisionRec(Rectangle rec1, Rectangle rec2)
bool CheckCollisionPointRec(Vector2 point, Rectangle rec)
bool CheckCollisionPointCircle(Vector2 point, Vector2 center, float radius)
bool CheckCollisionPointTriangle(Vector2 point, Vector2 p1,
        Vector2 p2, Vector2 p3)
Image LoadImage(const char *fileName)
Image LoadImageEx(Color *pixels, int width, int height)
Image LoadImagePro(void *data, int width, int height, int format)
Image LoadImageRaw(const char *fileName, int width, int height,
        int format, int headerSize)
void ExportImage(Image image, const char *fileName)
void ExportImageAsCode(Image image, const char *fileName)
Texture2D LoadTexture(const char *fileName)
Texture2D LoadTextureFromImage(Image image)
TextureCubemap LoadTextureCubemap(Image image, int layoutType)
RenderTexture2D LoadRenderTexture(int width, int height)
void UnloadImage(Image image)
void UnloadTexture(Texture2D texture)
void UnloadRenderTexture(RenderTexture2D target)
Color *GetImageData(Image image)
Vector4 *GetImageDataNormalized(Image image)
int GetPixelDataSize(int width, int height, int format)
Image GetTextureData(Texture2D texture)
Image GetScreenData(void)
void UpdateTexture(Texture2D texture, const void *pixels)
Image ImageCopy(Image image)
void ImageToPOT(Image *image, Color fillColor)
void ImageFormat(Image *image, int newFormat)
void ImageAlphaMask(Image *image, Image alphaMask)
void ImageAlphaClear(Image *image, Color color, float threshold)
void ImageAlphaCrop(Image *image, float threshold)
void ImageAlphaPremultiply(Image *image)
void ImageCrop(Image *image, Rectangle crop)
void ImageResize(Image *image, int newWidth, int newHeight)
void ImageResizeNN(Image *image, int newWidth,int newHeight)
void ImageResizeCanvas(Image *image, int newWidth, int newHeight,
        int offsetX, int offsetY, Color color)
void ImageMipmaps(Image *image)
void ImageDither(Image *image, int rBpp, int gBpp, int bBpp, int aBpp)
Color *ImageExtractPalette(Image image, int maxPaletteSize, int *extractCount)
Image ImageText(const char *text, int fontSize, Color color)
Image ImageTextEx(Font font, const char *text, float fontSize,
        float spacing, Color tint)
void ImageDraw(Image *dst, Image src, Rectangle srcRec, Rectangle dstRec)
void ImageDrawRectangle(Image *dst, Rectangle rec, Color color)
void ImageDrawRectangleLines(Image *dst, Rectangle rec, int thick, Color color)
```

(continues on next page)

(continued from previous page)

```
void ImageDrawText(Image *dst, Vector2 position, const char *text,
        int fontSize, Color color)
void ImageDrawTextEx(Image *dst, Vector2 position, Font font,
        const char *text, float fontSize, float spacing, Color color)
void ImageFlipVertical(Image *image)
void ImageFlipHorizontal(Image *image)
void ImageRotateCW(Image *image)
void ImageRotateCCW(Image *image)
void ImageColorTint(Image *image, Color color)
void ImageColorInvert(Image *image)
void ImageColorGrayscale(Image *image)
void ImageColorContrast(Image *image, float contrast)
void ImageColorBrightness(Image *image, int brightness)
void ImageColorReplace(Image *image, Color color, Color replace)
Image GenImageColor(int width, int height, Color color)
Image GenImageGradientV(int width, int height, Color top, Color bottom)
Image GenImageGradientH(int width, int height, Color left, Color right)
Image GenImageGradientRadial(int width, int height, float density,
        Color inner, Color outer)
Image GenImageChecked(int width, int height, int checksX, int checksY,
        Color col1, Color col2)
Image GenImageWhiteNoise(int width, int height, float factor)
Image GenImagePerlinNoise(int width, int height, int offsetX,
        int offsetY, float scale)
Image GenImageCellular(int width, int height, int tileSize)
void GenTextureMipmaps(Texture2D *texture)
void SetTextureFilter(Texture2D texture, int filterMode)
void SetTextureWrap(Texture2D texture, int wrapMode)
void DrawTexture(Texture2D texture, int posX, int posY, Color tint)
void DrawTextureV(Texture2D texture, Vector2 position, Color tint)
void DrawTextureEx(Texture2D texture, Vector2 position,
        float rotation, float scale, Color tint)
void DrawTextureRec(Texture2D texture, Rectangle sourceRec,
        Vector2 position, Color tint)
void DrawTextureQuad(Texture2D texture, Vector2 tiling, Vector2 offset,
        Rectangle quad, Color tint)
void DrawTexturePro(Texture2D texture, Rectangle sourceRec,
        Rectangle destRec, Vector2 origin, float rotation, Color tint)
void DrawTextureNPatch(Texture2D texture, NPatchInfo nPatchInfo,
        Rectangle destRec, Vector2 origin, float rotation, Color tint)
Font GetFontDefault(void)
Font LoadFont(const char *fileName)
Font LoadFontEx(const char *fileName, int fontSize,
        int *fontChars, int charsCount)
Font LoadFontFromImage(Image image, Color key, int firstChar)
CharInfo *LoadFontData(const char *fileName, int fontSize,
        int *fontChars, int charsCount, int type)
Image GenImageFontAtlas(CharInfo *chars, int charsCount,
        int fontSize, int padding, int packMethod)
void UnloadFont(Font font)
void DrawFPS(int posX, int posY)
void DrawText(const char *text, int posX, int posY, int fontSize, Color color)
```

(continues on next page)

(continued from previous page)

```
void DrawTextEx(Font font, const char *text, Vector2 position,
        float fontSize, float spacing, Color tint)
void DrawTextRec(Font font, const char *text, Rectangle rec,
        float fontSize, float spacing, bool wordWrap, Color tint)
void DrawTextRecEx(Font font, const char *text, Rectangle rec,
        float fontSize, float spacing, bool wordWrap, Color tint,
        int selectStart, int selectLength, Color selectText, Color selectBack)
int MeasureText(const char *text, int fontSize)
Vector2 MeasureTextEx(Font font, const char *text,
        float fontSize, float spacing)
int GetGlyphIndex(Font font, int character)
int GetNextCodepoint(const char *text, int *count)
bool TextIsEqual(const char *text1, const char *text2)
unsigned int TextLength(const char *text)
unsigned int TextCountCodepoints(const char *text)
const char *TextFormat(const char *text)
const char *TextSubtext(const char *text, int position, int length)
const char *TextReplace(char *text, const char *replace, const char *by)
const char *TextInsert(const char *text, const char *insert, int position)
const char *TextJoin(const char **textList, int count, const char *delimiter)
const char **TextSplit(const char *text, char delimiter, int *count)
void TextAppend(char *text, const char *append, int *position)
int TextFindIndex(const char *text, const char *find)
const char *TextToUpper(const char *text)
const char *TextToLower(const char *text)
const char *TextToPascal(const char *text)
int TextToInteger(const char *text)
void DrawLine3D(Vector3 startPos, Vector3 endPos, Color color)
void DrawCircle3D(Vector3 center, float radius, Vector3 rotationAxis,
float rotationAngle, Color color)
void DrawCube(Vector3 position, float width, float height, float length, Color color)
void DrawCubeV(Vector3 position, Vector3 size, Color color)
void DrawCubeWires(Vector3 position, float width, float height,
        float length, Color color)
void DrawCubeWiresV(Vector3 position, Vector3 size, Color color)
void DrawCubeTexture(Texture2D texture, Vector3 position,
        float width, float height, float length, Color color)
void DrawSphere(Vector3 centerPos, float radius, Color color)
void DrawSphereEx(Vector3 centerPos, float radius, int rings,
        int slices, Color color)
void DrawSphereWires(Vector3 centerPos, float radius, int rings,
        int slices, Color color)
void DrawCylinder(Vector3 position, float radiusTop, float radiusBottom,
        float height, int slices, Color color)
void DrawCylinderWires(Vector3 position, float radiusTop,
        float radiusBottom, float height, int slices, Color color)
void DrawPlane(Vector3 centerPos, Vector2 size, Color color)
void DrawRay(Ray ray, Color color)
void DrawGrid(int slices, float spacing)
void DrawGizmo(Vector3 position)
Model LoadModel(const char *fileName)
Model LoadModelFromMesh(Mesh mesh)
```

(continues on next page)

(continued from previous page)

```
void UnloadModel(Model model)
Mesh *LoadMeshes(const char *fileName, int *meshCount)
void ExportMesh(Mesh mesh, const char *fileName)
void UnloadMesh(Mesh *mesh)
Material *LoadMaterials(const char *fileName, int *materialCount)
Material LoadMaterialDefault(void)
void UnloadMaterial(Material material)
void SetMaterialTexture(Material *material, int mapType, Texture2D texture)
void SetModelMeshMaterial(Model *model, int meshId, int materialId)
ModelAnimation *LoadModelAnimations(const char *fileName, int *animsCount)
void UpdateModelAnimation(Model model, ModelAnimation anim, int frame)
void UnloadModelAnimation(ModelAnimation anim)
bool IsModelAnimationValid(Model model, ModelAnimation anim)
Mesh GenMeshPoly(int sides, float radius)
Mesh GenMeshPlane(float width, float length, int resX, int resZ)
Mesh GenMeshCube(float width, float height, float length)
Mesh GenMeshSphere(float radius, int rings, int slices)
Mesh GenMeshHemiSphere(float radius, int rings, int slices)
Mesh GenMeshCylinder(float radius, float height, int slices)
Mesh GenMeshTorus(float radius, float size, int radSeg, int sides)
Mesh GenMeshKnot(float radius, float size, int radSeg, int sides)
Mesh GenMeshHeightmap(Image heightmap, Vector3 size)
Mesh GenMeshCubicmap(Image cubicmap, Vector3 cubeSize)
BoundingBox MeshBoundingBox(Mesh mesh)
void MeshTangents(Mesh *mesh)
void MeshBinormals(Mesh *mesh)
void DrawModel(Model model, Vector3 position, float scale, Color tint)
void DrawModelEx(Model model, Vector3 position, Vector3 rotationAxis,
        float rotationAngle, Vector3 scale, Color tint)
void DrawModelWires(Model model, Vector3 position, float scale, Color tint)
void DrawModelWiresEx(Model model, Vector3 position,
        Vector3 rotationAxis, float rotationAngle, Vector3 scale, Color tint)
void DrawBoundingBox(BoundingBox box, Color color)
void DrawBillboard(Camera3D camera, Texture2D texture,
        Vector3 center, float size, Color tint)
void DrawBillboardRec(Camera3D camera, Texture2D texture,
        Rectangle sourceRec, Vector3 center, float size, Color tint)
bool CheckCollisionSpheres(Vector3 centerA, float radiusA,
        Vector3 centerB, float radiusB)
bool CheckCollisionBoxes(BoundingBox box1, BoundingBox box2)
bool CheckCollisionBoxSphere(BoundingBox box, Vector3 centerSphere, float radiusSphere)
bool CheckCollisionRaySphere(Ray ray, Vector3 spherePosition, float sphereRadius)
bool CheckCollisionRaySphereEx(Ray ray, Vector3 spherePosition,
        float sphereRadius, Vector3 *collisionPoint)
bool CheckCollisionRayBox(Ray ray, BoundingBox box)
RayHitInfo GetCollisionRayModel(Ray ray, Model *model)
RayHitInfo GetCollisionRayTriangle(Ray ray, Vector3 p1, Vector3 p2, Vector3 p3)
RayHitInfo GetCollisionRayGround(Ray ray, float groundHeight)
char *LoadText(const char *fileName)
Shader LoadShader(const char *vsFileName, const char *fsFileName)
Shader LoadShaderCode(char *vsCode, char *fsCode)
void UnloadShader(Shader shader)
```

(continues on next page)

(continued from previous page)

```
Shader GetShaderDefault(void)
Texture2D GetTextureDefault(void)
int GetShaderLocation(Shader shader, const char *uniformName)
void SetShaderValue(Shader shader, int uniformLoc,
        const void *value, int uniformType)
void SetShaderValueV(Shader shader, int uniformLoc,
        const void *value, int uniformType, int count)
void SetShaderValueMatrix(Shader shader, int uniformLoc, Matrix mat)
void SetShaderValueTexture(Shader shader, int uniformLoc, Texture2D texture)
void SetMatrixProjection(Matrix proj)
void SetMatrixModelview(Matrix view)
Matrix GetMatrixModelview(void)
Texture2D GenTextureCubemap(Shader shader, Texture2D skyHDR, int size)
Texture2D GenTextureIrradiance(Shader shader, Texture2D cubemap, int size)
Texture2D GenTexturePrefilter(Shader shader, Texture2D cubemap, int size)
Texture2D GenTextureBRDF(Shader shader, int size)
void BeginShaderMode(Shader shader)
void EndShaderMode(void)
void BeginBlendMode(int mode)
void EndBlendMode(void)
void BeginScissorMode(int x, int y, int width, int height)
void EndScissorMode(void)
void InitVrSimulator(void)
void CloseVrSimulator(void)
void UpdateVrTracking(Camera3D *camera)
void SetVrConfiguration(VrDeviceInfo info, Shader distortion)
bool IsVrSimulatorReady(void)
void ToggleVrMode(void)
void BeginVrDrawing(void)
void EndVrDrawing(void)
void InitAudioDevice(void)
void CloseAudioDevice(void)
bool IsAudioDeviceReady(void)
void SetMasterVolume(float volume)
Wave LoadWave(const char *fileName)
Wave LoadWaveEx(void *data, int sampleCount, int sampleRate,
        int sampleSize, int channels)
Sound LoadSound(const char *fileName)
Sound LoadSoundFromWave(Wave wave)
void UpdateSound(Sound sound, const void *data, int samplesCount)
void UnloadWave(Wave wave)
void UnloadSound(Sound sound)
void ExportWave(Wave wave, const char *fileName)
void ExportWaveAsCode(Wave wave, const char *fileName)
void PlaySound(Sound sound)
void PauseSound(Sound sound)
void ResumeSound(Sound sound)
void StopSound(Sound sound)
bool IsSoundPlaying(Sound sound)
void SetSoundVolume(Sound sound, float volume)
void SetSoundPitch(Sound sound, float pitch)
void WaveFormat(Wave *wave, int sampleRate, int sampleSize, int channels)
```

(continues on next page)

(continued from previous page)

```
Wave WaveCopy(Wave wave)
void WaveCrop(Wave *wave, int initSample, int finalSample)
float *GetWaveData(Wave wave)
Music LoadMusicStream(const char *fileName)
void UnloadMusicStream(Music music)
void PlayMusicStream(Music music)
void UpdateMusicStream(Music music)
void StopMusicStream(Music music)
void PauseMusicStream(Music music)
void ResumeMusicStream(Music music)
bool IsMusicPlaying(Music music)
void SetMusicVolume(Music music, float volume)
void SetMusicPitch(Music music, float pitch)
void SetMusicLoopCount(Music music, int count)
float GetMusicTimeLength(Music music)
float GetMusicTimePlayed(Music music)
AudioStream InitAudioStream(unsigned int sampleRate,
        unsigned int sampleSize, unsigned int channels)
void UpdateAudioStream(AudioStream stream, const void *data, int samplesCount)
void CloseAudioStream(AudioStream stream)
bool IsAudioBufferProcessed(AudioStream stream)
void PlayAudioStream(AudioStream stream)
void PauseAudioStream(AudioStream stream)
void ResumeAudioStream(AudioStream stream)
bool IsAudioStreamPlaying(AudioStream stream)
void StopAudioStream(AudioStream stream)
void SetAudioStreamVolume(AudioStream stream, float volume)
void SetAudioStreamPitch(AudioStream stream, float pitch)
void GuiEnable(void)
void GuiDisable(void)
void GuiLock(void)
void GuiUnlock(void)
void GuiState(int state)
void GuiFont(Font font)
void GuiFade(float alpha)
void GuiSetStyle(int control, int property, int value)
int GuiGetStyle(int control, int property)
bool GuiWindowBox(Rectangle bounds, const char *text)
void GuiGroupBox(Rectangle bounds, const char *text)
void GuiLine(Rectangle bounds, const char *text)
void GuiPanel(Rectangle bounds)
Rectangle GuiScrollPanel(Rectangle bounds, Rectangle content, Vector2 *scroll)
void GuiLabel(Rectangle bounds, const char *text)
bool GuiButton(Rectangle bounds, const char *text)
bool GuiLabelButton(Rectangle bounds, const char *text)
bool GuiImageButton(Rectangle bounds, Texture2D texture)
bool GuiImageButtonEx(Rectangle bounds, Texture2D texture,
        Rectangle texSource, const char *text)
bool GuiToggle(Rectangle bounds, const char *text, bool active)
int GuiToggleGroup(Rectangle bounds, const char *text, int active)
bool GuiCheckBox(Rectangle bounds, const char *text, bool checked)
int GuiComboBox(Rectangle bounds, const char *text, int active)=
```

(continues on next page)

(continued from previous page)

```
bool GuiDropdownBox(Rectangle bounds, const char *text,
        int *active, bool editMode)
bool GuiSpinner(Rectangle bounds, int *value, int minValue,
        int maxValue, bool editMode)
bool GuiValueBox(Rectangle bounds, int *value, int minValue,
        int maxValue, bool editMode)
bool GuiTextBox(Rectangle bounds, char *text, int textSize, bool editMode)
bool GuiTextBoxMulti(Rectangle bounds, char *text, int textSize, bool editMode)
float GuiSlider(Rectangle bounds, const char *text, float value,
        float minValue, float maxValue, bool showValue)
float GuiSliderBar(Rectangle bounds, const char *text, float value,
        float minValue, float maxValue, bool showValue)
float GuiProgressBar(Rectangle bounds, const char *text, float value,
        float minValue, float maxValue, bool showValue)
void GuiStatusBar(Rectangle bounds, const char *text)
void GuiDummyRec(Rectangle bounds, const char *text)
int GuiScrollBar(Rectangle bounds, int value, int minValue, int maxValue)
Vector2 GuiGrid(Rectangle bounds, float spacing, int subdivs)
bool GuiListView(Rectangle bounds, const char *text, int *active,
        int *scrollIndex, bool editMode)
bool GuiListViewEx(Rectangle bounds, const char **text, int count,
        int *enabled, int *active, int *focus, int *scrollIndex, bool editMode)
int GuiMessageBox(Rectangle bounds, const char *windowTitle,
        const char *message, const char *buttons)
int GuiTextInputBox(Rectangle bounds, const char *windowTitle,
        const char *message, char *text, const char *buttons)
Color GuiColorPicker(Rectangle bounds, Color color)
void GuiLoadStyle(const char *fileName)
void GuiLoadStyleProps(const int *props, int count)
void GuiLoadStyleDefault(void)
void GuiUpdateStyleComplete(void)
const char *GuiIconText(int iconId, const char *text)
```

57.28 More Functions

The next functions are added by the RingRayLib extension

```
* SetShaderInt(shader, Location, nValue)
* SetShaderFloat(shader, Location, nValue)
* SetShaderVec2(shader, Location, aValue)
* SetShaderVec3(shader, Location, aValue)
* SetShaderVec4(shader, Location, aValue)
* SetShaderIVec2(shader, Location, aValue)
* SetShaderIVec3(shader, Location, aValue)
* SetShaderIVec4(shader, Location, aValue)
* SetModelMaterialShader(model, nID, shader)
* vec2() --> Vector2
* vec2getx() --> double
* vec2setx(double)
* vec2gety() --> double
```

(continues on next page)

(continued from previous page)

```
* vec2sety(double)
* vec3() --> Vector3
* vec3getx() --> double
* vec3setx(double)
* vec3gety() --> double
* vec3sety(double)
* vec3getz() --> double
* vec3setz(double)
* vec4() --> Vector4
* vec4getx() --> double
* vec4setx(double)
* vec4gety() --> double
* vec4sety(double)
* vec4getz() --> double
* vec4setz(double)
* vec4getw() --> double
* vec4setw(double)
* getcamera3dposx() --> double
* setcamera3dposx(double)
* getcamera3dposy() --> double
* setcamera3dposy(double)
* getcamera3dposz() --> double
* setcamera3dposz(double)
* getcamera3dtarx() --> double
* setcamera3dtarx(double)
* getcamera3dtary() --> double
* setcamera3dtary(double)
* getcamera3dtarz() --> double
* setcamera3dtarz(double)
* getcamera3dupx() --> double
* setcamera3dupx(double)
* getcamera3dupy() --> double
* setcamera3dupy(double)
* getcamera3dupz() --> double
* setcamera3dupz(double)
* getcamera3fovy() --> double
* setcamera3fovy(double)
* getcamera3type() --> double
* setcamera3type(double)
```

CHAPTER

FIFTYEIGHT

USING RINGOPENGL AND RINGFREEGLUT FOR 3D GRAPHICS

In this chapter we will learn about using RingOpenGL

58.1 Samples Source (Authors)

The samples in this chapter are based on C Tutorials

from

(1) http://www.lighthouse3d.com/tutorials/glut-tutorial/

(2) http://www.wikihow.com/Make-a-Cube-in-OpenGL

58.2 What is RingOpenGL?

RingOpenGL contains the Ring binding to the OpenGL library

You can learn about OpenGL from : https://www.opengl.org/

RingOpenGL comes with support for the next versions

- OpenGL 1.1
- OpenGL 1.2
- OpenGL 1.3
- OpenGL 1.4
- OpenGL 1.5
- OpenGL 2.0
- OpenGL 2.1
- OpenGL 3.0
- OpenGL 3.2
- OpenGL 3.3
- OpenGL 4.0
- OpenGL 4.1
- OpenGL 4.2
- OpenGL 4.3

- OpenGL 4.4
- OpenGL 4.5
- OpenGL 4.6

For example, if you want to use OpenGL 2.1 then load RingOpenGL 2.1 library

```
load "opengl21lib.ring"
```

58.3 What is RingFreeGLUT?

RingFreeGLUT contains the Ring binding to the FreeGLUT library

You can learn about FreeGLUT from : http://freeglut.sourceforge.net/

To use the RingFreeGLUT library, Just load the library

```
load "freeglut.ring"
```

58.4 The First Window using RingFreeGLUT

Example:

```
load "freeglut.ring"

func main
        glutInit()
        glutInitDisplayMode(GLUT_SINGLE)
        glutInitWindowSize(800, 600)
        glutInitWindowPosition(100, 10)
        glutCreateWindow("RingFreeGLUT - Test 1")
        glutDisplayFunc(:displayCode)
        glutMainLoop()

func displaycode
```

Screen Shot

58.5 Drawing using RingOpenGL

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"


func main
        glutInit()
        glutInitDisplayMode(GLUT_SINGLE)
        glutInitWindowSize(800, 600)
        glutInitWindowPosition(100, 10)
        glutCreateWindow("RingFreeGLUT - Test 2")
        glutDisplayFunc(:displayCode)
        glutMainLoop()

func displaycode
        glClear(GL_COLOR_BUFFER_BIT)
        glColor3f(0,255,0)
```

(continues on next page)

(continued from previous page)

```
        glBegin(GL_POLYGON)
                glVertex3f(0.0, 0.0, 0.0)
                glVertex3f(0.5, 0.0, 0.0)
                glVertex3f(0.5, 0.5, 0.0)
                glVertex3f(0.0, 0.5, 0.0)
        glEnd()
        glColor3f(255,0,0)
        glBegin(GL_POLYGON)
                glVertex3f(0.0, 0.0, 0.0)
                glVertex3f(0.5, 0.0, 0.0)
                glVertex3f(-0.5,- 1, 0.0)
                glVertex3f(0.0, -1, 0.0)
        glEnd()
        glColor3f(0,0,255)
        glBegin(GL_POLYGON)
               glVertex3f(0.0, 0.0, 0.0)
                glVertex3f(-0.5, 0.0, 0.0)
                glVertex3f(-0.5,- 0.5, 0.0)
                glVertex3f(0.0, -0.5, 0.0)
        glEnd()

        glFlush()
```

Screen Shot

58.6 The First Triangle

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

func main
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowSize(320,320)
        glutInitWindowPosition(100, 10)
        glutCreateWindow("RingFreeGLUT - Test 3")
        glutDisplayFunc(:renderScene)
        glutMainLoop()

func renderScene

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
```

(continues on next page)

(continued from previous page)

```
        glBegin(GL_TRIANGLES)
                glVertex3f(-0.5,-0.5,0.0)
                glVertex3f(0.5,0.0,0.0)
                glVertex3f(0.0,0.5,0.0)
        glEnd()

        glutSwapBuffers()
```

Screen Shot

58.7 Window Resize Event

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 4")

        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
```

(continues on next page)

(continued from previous page)

```
        glutMainLoop()

func renderScene

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        glBegin(GL_TRIANGLES)
                glVertex3f(-2,-2,-5.0)
                glVertex3f(2,0.0,-5.0)
                glVertex3f(0.0,2,-5.0)
        glEnd()

        glutSwapBuffers()

func changesize

        h = glutEventHeight()
        w = glutEventWidth()

                // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if (h = 0)
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45,ratio,1,100)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)
```

58.8 Triangle Rotation

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

angle = 0

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 5")

        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        glutMainLoop()

func renderScene

        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()
        // Set the camera
        gluLookAt(      0.0, 0.0, 10.0,
                                0.0, 0.0,  0.0,
                                0.0, 1.0,  0.0)

        glRotatef(angle, 0.0, 1.0, 0.0)

        glBegin(GL_TRIANGLES)
                glVertex3f(-2.0,-2.0, 0.0)
                glVertex3f( 2.0, 0.0, 0.0)
                glVertex3f( 0.0, 2.0, 0.0)
        glEnd()

        angle+=0.1

        glutSwapBuffers();

func changesize

        h = glutEventHeight()
        w = glutEventWidth()
```

(continues on next page)

(continued from previous page)

```
                // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if (h = 0)
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45,ratio,1,100)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)
```

Screen Shot

58.9 Keyboard Events and Colors

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

angle = 0

red=1.0
blue=1.0
green=1.0

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 6")

        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        // here are the new entries
        glutKeyboardFunc(:processNormalKeys)
        glutSpecialFunc(:processSpecialKeys)

        glutMainLoop()

func renderScene

        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()
        // Set the camera
        gluLookAt(      0.0, 0.0, 10.0,
                                0.0, 0.0,  0.0,
                                0.0, 1.0,  0.0)

        glRotatef(angle, 0.0, 1.0, 0.0)


        glColor3f(red,green,blue);

        glBegin(GL_TRIANGLES)
                glVertex3f(-2.0,-2.0, 0.0)
                glVertex3f( 2.0, 0.0, 0.0)
                glVertex3f( 0.0, 2.0, 0.0)
```

(continues on next page)

(continued from previous page)

```
        glEnd()

        angle+=0.1

        glutSwapBuffers();

func changesize

        h = glutEventHeight()
        w = glutEventWidth()

                // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if (h = 0)
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45,ratio,1,100)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)

func processNormalKeys
        key = GLUTEventKey()
        if key = 27
                shutdown()
        ok

func processSpecialKeys

        key = GLUTEventKey()

        switch key
                on GLUT_KEY_F1
                                red = 1.0
                                green = 0.0
                                blue = 0.0
                on GLUT_KEY_F2
                                red = 0.0
                                green = 1.0
```

(continues on next page)

(continued from previous page)

```
                                blue = 0.0
                on GLUT_KEY_F3
                                red = 0.0
                                green = 0.0
                                blue = 1.0
        off
```

Screen Shot

58.10 The Camera

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

// angle of rotation for the camera direction
angle=0.0
// actual vector representing the camera's direction
lx=0.0
lz=-1.0
// XZ position of the camera
x=0.0
z=5.0

func drawSnowMan

        glColor3f(1.0, 1.0, 1.0)
```

(continues on next page)

(continued from previous page)

```
// Draw Body
        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

// Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

// Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)

        glPopMatrix()

// Draw Nose
        glColor3f(1.0, 0.5 , 0.5)
        glutSolidCone(0.08,0.5,10,2)


func changeSize
        w = glutEventWidth()
        h = glutEventHeight()

        // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if h = 0
                h = 1
        ok

        ratio =  w * 1.0  / h

                // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

                // Reset Matrix
        glLoadIdentity()


        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45.0, ratio, 0.1, 100.0);

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)
```

(continues on next page)

(continued from previous page)

```
func processNormalKeys
        key = glutEventKey()

        if key = 27
                shutdown()
        ok



func renderScene

        // Clear Color and Depth Buffers

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()
        // Set the camera
        gluLookAt(      x, 1.0, z,
                        x+lx, 1.0,  z+lz,
                        0.0, 1.0,  0.0)

                // Draw ground

        glColor3f(0.9, 0.9, 0.9)
        glBegin(GL_QUADS)
                glVertex3f(-100.0, 0.0, -100.0)
                glVertex3f(-100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0, -100.0)
        glEnd()

                // Draw 36 SnowMen
        for i = -3 to 2
                for  j=-3 to 2
                        glPushMatrix()
                        glTranslatef(i*10.0,0,j * 10.0)
                        drawSnowMan()
                        glPopMatrix()
                next
        next
        glutSwapBuffers()




func processSpecialKeys

        key = glutEventKey()

        fraction = 0.1
```

(continues on next page)

(continued from previous page)

```
        switch key
                on GLUT_KEY_LEFT
                        angle -= 0.01
                        lx = sin(angle)
                        lz = -cos(angle)
                on GLUT_KEY_RIGHT
                        angle += 0.01
                        lx = sin(angle)
                        lz = -cos(angle)
                on GLUT_KEY_UP
                        x += lx * fraction
                        z += lz * fraction
                on GLUT_KEY_DOWN
                        x -= lx * fraction
                        z -= lz * fraction
        off


func main

        // init GLUT and create window

        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)

        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 7")

        // register callbacks
        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)
        glutKeyboardFunc(:processNormalKeys)
        glutSpecialFunc(:processSpecialKeys)

        // OpenGL init
        glEnable(GL_DEPTH_TEST)

        // enter GLUT event processing cycle
        glutMainLoop()
```

Screen Shot

Another Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0 z=5.0
// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0

func changeSize
        w = glutEventWidth()
        h = glutEventHeight()

        // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if h = 0
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)
```

(continues on next page)

(continued from previous page)

```
        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45.0, ratio, 0.1, 100.0)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)


func drawSnowMan

        glColor3f(1.0, 1.0, 1.0)

// Draw Body

        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

// Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

// Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)
        glPopMatrix()

// Draw Nose
        glColor3f(1.0, 0.5 , 0.5)
        glRotatef(0.0,1.0, 0.0, 0.0)
        glutSolidCone(0.08,0.5,10,2)


func computePos deltaMove

        x += deltaMove * lx * 0.1
        z += deltaMove * lz * 0.1


func computeDir deltaAngle

        angle += deltaAngle
        lx = sin(angle)
```

(continues on next page)

(continued from previous page)

```
        lz = -cos(angle)

func renderScene

        if deltaMove
                computePos(deltaMove)
        ok

        if deltaAngle
                computeDir(deltaAngle)
        ok

        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()
        // Set the camera
        gluLookAt(      x, 1.0, z,
                                x+lx, 1.0,  z+lz,
                                0.0, 1.0,  0.0)

// Draw ground

        glColor3f(0.9, 0.9, 0.9)
        glBegin(GL_QUADS)
                glVertex3f(-100.0, 0.0, -100.0)
                glVertex3f(-100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0, -100.0)
        glEnd()

// Draw 36 SnowMen

        for i = -3 to 2
                for j=-3 to 2
                        glPushMatrix()
                        glTranslatef(i*10.0,0,j * 10.0)
                        drawSnowMan()
                        glPopMatrix()
                next
        next
        glutSwapBuffers()


func pressKey
        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on GLUT_KEY_LEFT
```

(continues on next page)

(continued from previous page)

```
                        deltaAngle = -0.01
                on GLUT_KEY_RIGHT
                        deltaAngle = 0.01
                on GLUT_KEY_UP
                        deltaMove = 0.5
                on GLUT_KEY_DOWN
                        deltaMove = -0.5
        off

func releaseKey

        key = glutEventKey()

        switch key
                on GLUT_KEY_LEFT
                        deltaAngle = 0.0
                on GLUT_KEY_RIGHT
                        deltaAngle = 0.0
                on GLUT_KEY_UP
                        deltaMove = 0
                on GLUT_KEY_DOWN
                        deltaMove = 0
        off

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 8")

        // register callbacks
        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        glutSpecialFunc(:pressKey)

        // here are the new entries
        glutIgnoreKeyRepeat(1)
        glutSpecialUpFunc(:releaseKey)

        // OpenGL init
        glEnable(GL_DEPTH_TEST)

        // enter GLUT event processing cycle
        glutMainLoop()
```

58.11 Mouse Events

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0  z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0.0
xOrigin = -1

func changeSize
        w = glutEventWidth()
        h = glutEventHeight()

        // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if h = 0
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45.0, ratio, 0.1, 100.0)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)


func drawSnowMan

        glColor3f(1.0, 1.0, 1.0)
```

(continues on next page)

(continued from previous page)

```
        // Draw Body
        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

        // Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

        // Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)
        glPopMatrix()

        // Draw Nose
        glColor3f(1.0, 0.5 , 0.5)
        glRotatef(0.0,1.0, 0.0, 0.0)
        glutSolidCone(0.08,0.5,10,2)


func computePos deltaMove

        x += deltaMove * lx * 0.1
        z += deltaMove * lz * 0.1

func renderScene

        if deltaMove
                computePos(deltaMove)
        ok

        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()
        // Set the camera
        gluLookAt(      x, 1.0, z,
                        x+lx, 1.0,  z+lz,
                        0.0, 1.0,  0.0)

        // Draw ground

        glColor3f(0.9, 0.9, 0.9)
        glBegin(GL_QUADS)
                glVertex3f(-100.0, 0.0, -100.0)
                glVertex3f(-100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0,  100.0)
```

(continues on next page)

(continued from previous page)

```
                glVertex3f( 100.0, 0.0, -100.0)
        glEnd()

        // Draw 36 SnowMen

        for i = -3 to 2
                for  j=-3 to 2
                                        glPushMatrix()
                                        glTranslatef(i*10.0,0,j * 10.0)
                                        drawSnowMan()
                                        glPopMatrix()
                 next
        next
        glutSwapBuffers()

func processNormalKeys

        key  = glutEventKey()

                if key = 27
                        shutdown()
                ok


func pressKey
        key  = glutEventKey()

           switch key
                        on GLUT_KEY_UP
                               deltaMove = 0.5
                        on GLUT_KEY_DOWN
                                deltaMove = -0.5
           off


func releaseKey
        key  = glutEventKey()
                switch key
                        on GLUT_KEY_UP
                               deltaMove = 0
                        on GLUT_KEY_DOWN
                               deltaMove = 0
                off


func mouseMove
        xx = glutEventX()
        yy = glutEventY()
                // this will only be true when the left button is down
                if xOrigin >= 0

                // update deltaAngle
```

(continues on next page)

(continued from previous page)

```
                deltaAngle = (xx - xOrigin) * 0.001

                // update camera's direction
                lx = sin(angle + deltaAngle)
                lz = -cos(angle + deltaAngle)
        ok



func mouseButton

        button  = glutEventButton()
        state = glutEventState()
        xx = glutEventX()
        yy = glutEventY()

        // only start motion if the left button is pressed
        if button = GLUT_LEFT_BUTTON
                // when the button is released
                if state = GLUT_UP
                        angle += deltaAngle
                        xOrigin = -1
                else
                        // state = GLUT_DOWN
                        xOrigin = xx
                ok
                fflush(stdout)
        ok


func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 9")

        // register callbacks
        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        glutIgnoreKeyRepeat(1)
        glutKeyboardFunc(:processNormalKeys)
        glutSpecialFunc(:pressKey)
        glutSpecialUpFunc(:releaseKey)

        // here are the two new functions
        glutMouseFunc(:mouseButton)
        glutMotionFunc(:mouseMove)
```

(continues on next page)

(continued from previous page)

```
        // OpenGL init
        glEnable(GL_DEPTH_TEST)

        // enter GLUT event processing cycle
        glutMainLoop()
```

58.12 Menu Events

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0  z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1

// Constant definitions for Menus


// for RingFreeGLUT - We must have different ID for each menu item
C_RED = 1
C_GREEN = 2
C_BLUE  = 3
C_ORANGE = 4

C_FILL = 5
C_LINE = 6

C_SHRINK = 7
C_NORMAL = 8

// Pop up menu identifiers
fillMenu= 0
shrinkMenu= 0
mainMenu=0
colorMenu=0
```

(continues on next page)

(continued from previous page)

```
// color for the nose
red = 1.0  blue=0.5  green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

func changeSize
        w = glutEventWidth()
        h = glutEventHeight()

        // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if h = 0
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45.0, ratio, 0.1, 100.0)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)

func drawSnowMan

        glScalef(scale, scale, scale)
        glColor3f(1.0, 1.0, 1.0)

// Draw Body
        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

// Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

// Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
```

(continues on next page)

(continued from previous page)

```
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)
        glPopMatrix()

// Draw Nose
        glColor3f(red, green, blue)
        glRotatef(0.0,1.0, 0.0, 0.0)
        glutSolidCone(0.08,0.5,10,2)

        glColor3f(1.0, 1.0, 1.0)


func computePos deltaMove

        x += deltaMove * lx * 0.1
        z += deltaMove * lz * 0.1


func renderScene

        if deltaMove
                computePos(deltaMove)
        ok

        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()
        // Set the camera
        gluLookAt(      x, 1.0, z,
                        x+lx, 1.0,  z+lz,
                        0.0, 1.0,  0.0)

// Draw ground

        glColor3f(0.9, 0.9, 0.9)
        glBegin(GL_QUADS)
                glVertex3f(-100.0, 0.0, -100.0)
                glVertex3f(-100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0, -100.0)
        glEnd()

// Draw 36 SnowMen

        for  i = -3 to 2
                for  j = -3 to 2
                        glPushMatrix()
                        glTranslatef(i*10.0, 0.0, j * 10.0)
```

(continues on next page)

(continued from previous page)

```
                        drawSnowMan()
                        glPopMatrix()
                next
        next
        glutSwapBuffers()


// ----------------------------------
//             KEYBOARD
// ----------------------------------

func processNormalKeys
        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        glutSetMenu(mainMenu)
        switch key
                on 27
                        glutDestroyMenu(mainMenu)
                        glutDestroyMenu(fillMenu)
                        glutDestroyMenu(colorMenu)
                        glutDestroyMenu(shrinkMenu)
                        shutdown()

                on 's'
                        if not menuFlag
                          glutChangeToSubMenu(2,"Shrink",shrinkMenu)
                        ok
                on 'c'
                        if not menuFlag
                                glutChangeToSubMenu(2,"Color",colorMenu)
                        ok
        off
        if key = 27
                shutdown()
        ok


func pressKey

        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on GLUT_KEY_UP
                         deltaMove = 0.5
                on GLUT_KEY_DOWN
                        deltaMove = -0.5
        off
```

(continues on next page)

(continued from previous page)

```
func releaseKey

        key = glutEventKey()

        switch key
                on GLUT_KEY_UP
                        deltaMove = 0
                on GLUT_KEY_DOWN
                        deltaMove = 0
        off


// -----------------------------------
//              MOUSE
// -----------------------------------

func mouseMove
        xx = glutEventX()
        yy = glutEventY()

        // this will only be true when the left button is down
        if xOrigin >= 0

                // update deltaAngle
                deltaAngle = (xx - xOrigin) * 0.001

                // update camera's direction
                lx = sin(angle + deltaAngle)
                lz = -cos(angle + deltaAngle)
        ok


func mouseButton

        button = glutEventButton()
        state = glutEventState()
        xx = glutEventX()
        yy = glutEventY()

        // only start motion if the left button is pressed
        if button = GLUT_LEFT_BUTTON
                // when the button is released
                if state = GLUT_UP
                        angle += deltaAngle
                        xOrigin = -1
                else
                        // state = GLUT_DOWN
                        xOrigin = xx
                ok
        ok
```

(continues on next page)

(continued from previous page)

```
// ----------------------------------
//             MENUS
// ----------------------------------

func processMenuStatus

        status = glutEventStatus()
        xx = glutEventX()
        yy = glutEventY()

        if  status = GLUT_MENU_IN_USE
                menuFlag = 1
        else
                menuFlag = 0
        ok


func processMainMenu

        // nothing to do in here
        // all actions are for submenus


func processFillMenu

        option = glutEventValue()

        switch option

                on C_FILL
                        glPolygonMode(GL_FRONT, GL_FILL)
                on C_LINE
                        glPolygonMode(GL_FRONT, GL_LINE)
        off


func processShrinkMenu

        option = glutEventValue()

        switch option

                on C_SHRINK
                        scale = 0.5
                on C_NORMAL
                        scale = 1.0
        off


func processColorMenu
```

(continues on next page)

(continued from previous page)

```
        option = glutEventValue()

        switch option
                on C_RED
                        red = 1.0
                        green = 0.0
                        blue = 0.0
                on C_GREEN
                        red = 0.0
                        green = 1.0
                        blue = 0.0
                on C_BLUE
                        red = 0.0
                        green = 0.0
                        blue = 1.0
                on C_ORANGE
                        red = 1.0
                        green = 0.5
                        blue = 0.5
        off


func createPopupMenus

        shrinkMenu = glutCreateMenu(:processShrinkMenu)

        glutAddMenuEntry("Shrink",C_SHRINK)
        glutAddMenuEntry("NORMAL",C_NORMAL)

        fillMenu = glutCreateMenu(:processFillMenu)

        glutAddMenuEntry("Fill",C_FILL)
        glutAddMenuEntry("Line",C_LINE)

        colorMenu = glutCreateMenu(:processColorMenu)
        glutAddMenuEntry("Red",C_RED)
        glutAddMenuEntry("Blue",C_BLUE)
        glutAddMenuEntry("Green",C_GREEN)
        glutAddMenuEntry("Orange",C_ORANGE)

        mainMenu = glutCreateMenu(:processMainMenu)

        glutAddSubMenu("Polygon Mode", fillMenu)
        glutAddSubMenu("Color", colorMenu)
        // attach the menu to the right button
        glutAttachMenu(GLUT_RIGHT_BUTTON)

        // this will allow us to know if the menu is active
        glutMenuStatusFunc(:processMenuStatus)


// -----------------------------------
```

(continues on next page)

(continued from previous page)

```
//              MAIN
// -----------------------------------

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 10")

        // register callbacks
        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        glutIgnoreKeyRepeat(1)
        glutKeyboardFunc(:processNormalKeys)
        glutSpecialFunc(:pressKey)
        glutSpecialUpFunc(:releaseKey)

        // here are the two new functions
        glutMouseFunc(:mouseButton)
        glutMotionFunc(:mouseMove)

        // OpenGL init
        glEnable(GL_DEPTH_TEST)
        glEnable(GL_CULL_FACE)

        // init Menus
        createPopupMenus()

        // enter GLUT event processing cycle
        glutMainLoop()
```

Screen Shot

58.13 Using Fonts

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"


// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0  z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1

// Constant definitions for Menus
C_RED  = 1
C_GREEN = 2
C_BLUE = 3
C_ORANGE = 4

C_FILL = 5
```

(continues on next page)

(continued from previous page)

```
C_LINE = 6

// Pop up menu identifiers
fillMenu=NULL
fontMenu=NULL
mainMenu=NULL
colorMenu=NULL

// color for the nose
red = 1.0
blue=0.5
green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

// default font
font = GLUT_BITMAP_TIMES_ROMAN_24

C_INT_GLUT_BITMAP_8_BY_13 = 7
C_INT_GLUT_BITMAP_9_BY_15 = 8
C_INT_GLUT_BITMAP_TIMES_ROMAN_10  = 9
C_INT_GLUT_BITMAP_TIMES_ROMAN_24  = 10
C_INT_GLUT_BITMAP_HELVETICA_10  = 11
C_INT_GLUT_BITMAP_HELVETICA_12  = 12
C_INT_GLUT_BITMAP_HELVETICA_18  = 13

func changeSize
        w = glutEventWidth()
        h = glutEventHeight()

        // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if h = 0
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
```

(continues on next page)

(continued from previous page)

```
        gluPerspective(45.0, ratio, 0.1, 100.0)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)

func drawSnowMan

        glScalef(scale, scale, scale)
        glColor3f(1.0, 1.0, 1.0)

// Draw Body
        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

// Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

// Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)
        glPopMatrix()

// Draw Nose
        glColor3f(red, green, blue)
        glRotatef(0.0,1.0, 0.0, 0.0)
        glutSolidCone(0.08,0.5,10,2)

        glColor3f(1.0, 1.0, 1.0)

func renderBitmapString x,y,z,font,string
        glRasterPos3f(x, y,z)
        for c in string
                glutBitmapCharacter(font,ascii(c))
        next


func computePos deltaMove

        x += deltaMove * lx * 0.1
        z += deltaMove * lz * 0.1


func renderScene

        if  deltaMove
                computePos(deltaMove)
        ok
```

(continues on next page)

(continued from previous page)

```
        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()

        // Set the camera
        gluLookAt(      x, 1.0, z,
                        x+lx, 1.0,  z+lz,
                        0.0, 1.0,  0.0)

        // Draw ground

        glColor3f(0.9, 0.9, 0.9)
        glBegin(GL_QUADS)
                glVertex3f(-100.0, 0.0, -100.0)
                glVertex3f(-100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0, -100.0)
        glEnd()

// Draw 36 SnowMen
        for i = -3 to 2
                for j = -3 to 2
                        glPushMatrix()
                        glTranslatef(i*10.0, 0.0, j * 10.0)
                        drawSnowMan()
                        number = (i+3)*6+(j+3)
                        renderBitmapString(0.0, 0.5, 0.0,font ,""+number)
                        glPopMatrix()
                next
        next

        glutSwapBuffers()


// -----------------------------------
//             KEYBOARD
// -----------------------------------

func processNormalKeys
        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on 27
                        glutDestroyMenu(mainMenu)
                        glutDestroyMenu(fillMenu)
                        glutDestroyMenu(colorMenu)
                        glutDestroyMenu(fontMenu)
```

(continues on next page)

(continued from previous page)

```
                        Shutdown()
        off


func pressKey

        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on GLUT_KEY_UP
                         deltaMove = 0.5
                on GLUT_KEY_DOWN
                        deltaMove = -0.5
        off


func releaseKey

        key = glutEventKey()

        switch key
                on GLUT_KEY_UP
                        deltaMove = 0
                on GLUT_KEY_DOWN
                        deltaMove = 0
        off


// -----------------------------------
//             MOUSE
// -----------------------------------

func mouseMove
        xx = glutEventX()
        yy = glutEventY()

        // this will only be true when the left button is down
        if xOrigin >= 0

                // update deltaAngle
                deltaAngle = (xx - xOrigin) * 0.001

                // update camera's direction
                lx = sin(angle + deltaAngle)
                lz = -cos(angle + deltaAngle)
        ok


func mouseButton
```

(continues on next page)

(continued from previous page)

```
        button = glutEventButton()
        state = glutEventState()
        xx = glutEventX()
        yy = glutEventY()

        // only start motion if the left button is pressed
        if button = GLUT_LEFT_BUTTON
                // when the button is released
                if state = GLUT_UP
                        angle += deltaAngle
                        xOrigin = -1
                else
                        // state = GLUT_DOWN
                        xOrigin = xx
                ok
        ok


// ----------------------------------
//             MENUS
// ----------------------------------

func processMenuStatus

        status = glutEventStatus()

        if status = GLUT_MENU_IN_USE
                menuFlag = 1
        else
                menuFlag = 0
        ok


func processMainMenu

        // nothing to do in here
        // all actions are for submenus


func processFillMenu

        option = glutEventValue()

        switch option

                on C_FILL
                        glPolygonMode(GL_FRONT, GL_FILL)
                on C_LINE
                         glPolygonMode(GL_FRONT, GL_LINE)
        off
```

(continues on next page)

(continued from previous page)

```
func processFontMenu

        option = glutEventValue()

        switch (option) {
                on C_INT_GLUT_BITMAP_8_BY_13
                        font = GLUT_BITMAP_8_BY_13
                on C_INT_GLUT_BITMAP_9_BY_15
                        font = GLUT_BITMAP_9_BY_15
                on C_INT_GLUT_BITMAP_TIMES_ROMAN_10
                        font = GLUT_BITMAP_TIMES_ROMAN_10
                on C_INT_GLUT_BITMAP_TIMES_ROMAN_24
                        font = GLUT_BITMAP_TIMES_ROMAN_24
                on C_INT_GLUT_BITMAP_HELVETICA_10
                        font = GLUT_BITMAP_HELVETICA_10
                on C_INT_GLUT_BITMAP_HELVETICA_12
                        font = GLUT_BITMAP_HELVETICA_12
                on C_INT_GLUT_BITMAP_HELVETICA_18
                        font = GLUT_BITMAP_HELVETICA_18
        off

func processColorMenu

        option = glutEventValue()

        switch option
                on C_RED
                        red = 1.0
                        green = 0.0
                        blue = 0.0
                on C_GREEN
                        red = 0.0
                        green = 1.0
                        blue = 0.0
                on C_BLUE
                        red = 0.0
                        green = 0.0
                        blue = 1.0
                on C_ORANGE
                        red = 1.0
                        green = 0.5
                        blue = 0.5
        off


func createPopupMenus

        fontMenu = glutCreateMenu(:processFontMenu)

        glutAddMenuEntry("BITMAP_8_BY_13 ",C_INT_GLUT_BITMAP_8_BY_13 )
        glutAddMenuEntry("BITMAP_9_BY_15",C_INT_GLUT_BITMAP_9_BY_15 )
        glutAddMenuEntry("BITMAP_TIMES_ROMAN_10 ",C_INT_GLUT_BITMAP_TIMES_ROMAN_10  )
```

(continues on next page)

(continued from previous page)

```
        glutAddMenuEntry("BITMAP_TIMES_ROMAN_24",C_INT_GLUT_BITMAP_TIMES_ROMAN_24  )
        glutAddMenuEntry("BITMAP_HELVETICA_10 ",C_INT_GLUT_BITMAP_HELVETICA_10  )
        glutAddMenuEntry("BITMAP_HELVETICA_12",C_INT_GLUT_BITMAP_HELVETICA_12  )
        glutAddMenuEntry("BITMAP_HELVETICA_18",C_INT_GLUT_BITMAP_HELVETICA_18  )

        fillMenu = glutCreateMenu(:processFillMenu)

        glutAddMenuEntry("Fill",C_FILL)
        glutAddMenuEntry("Line",C_LINE)

        colorMenu = glutCreateMenu(:processColorMenu)
        glutAddMenuEntry("Red",C_RED);
        glutAddMenuEntry("Blue",C_BLUE);
        glutAddMenuEntry("Green",C_GREEN);
        glutAddMenuEntry("Orange",C_ORANGE);

        mainMenu = glutCreateMenu(:processMainMenu)

        glutAddSubMenu("Polygon Mode", fillMenu)
        glutAddSubMenu("Color", colorMenu)
        glutAddSubMenu("Font",fontMenu)
        // attach the menu to the right button
        glutAttachMenu(GLUT_RIGHT_BUTTON)

        // this will allow us to know if the menu is active
        glutMenuStatusFunc(:processMenuStatus)


// -----------------------------------
//             MAIN
// -----------------------------------

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test 11")

        // register callbacks
        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        glutIgnoreKeyRepeat(1)
        glutKeyboardFunc(:processNormalKeys)
        glutSpecialFunc(:pressKey)
        glutSpecialUpFunc(:releaseKey)

        // here are the two new functions
```

(continues on next page)

(continued from previous page)

```
        glutMouseFunc(:mouseButton)
        glutMotionFunc(:mouseMove)

        // OpenGL init
        glEnable(GL_DEPTH_TEST)
        glEnable(GL_CULL_FACE)

        // init Menus
        createPopupMenus()

        // enter GLUT event processing cycle
        glutMainLoop()
```

Screen Shot

58.14 Frames Per Second

Example

```
load "freeglut.ring"
load "opengl21lib.ring"

// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
```

(continues on next page)

(continued from previous page)

```
x=0.0  z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1

// Constant definitions for Menus
C_RED  = 1
C_GREEN = 2
C_BLUE = 3
C_ORANGE = 4

C_FILL = 5
C_LINE = 6

// Pop up menu identifiers
fillMenu=NULL
fontMenu=NULL
mainMenu=NULL
colorMenu=NULL

// color for the nose
red = 1.0
blue=0.5
green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

// default font
font = GLUT_BITMAP_TIMES_ROMAN_24

C_INT_GLUT_BITMAP_8_BY_13 = 7
C_INT_GLUT_BITMAP_9_BY_15 = 8
C_INT_GLUT_BITMAP_TIMES_ROMAN_10  = 9
C_INT_GLUT_BITMAP_TIMES_ROMAN_24  = 10
C_INT_GLUT_BITMAP_HELVETICA_10  = 11
C_INT_GLUT_BITMAP_HELVETICA_12  = 12
C_INT_GLUT_BITMAP_HELVETICA_18  = 13

// width and height of the window
h = 0
w = 0

// variables to compute frames per second
frame=0
time=0
```

(continues on next page)

(continued from previous page)

```
timebase=0
s = ""

func changeSize
        w = glutEventWidth()
        h = glutEventHeight()

        // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if h = 0
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45.0, ratio, 0.1, 100.0)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)

func drawSnowMan

        glScalef(scale, scale, scale)
        glColor3f(1.0, 1.0, 1.0)

// Draw Body
        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

// Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

// Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)
        glPopMatrix()
```

(continues on next page)

(continued from previous page)

```
// Draw Nose
        glColor3f(red, green, blue)
        glRotatef(0.0,1.0, 0.0, 0.0)
        glutSolidCone(0.08,0.5,10,2)

        glColor3f(1.0, 1.0, 1.0)

func renderBitmapString x,y,z,font,string
        glRasterPos3f(x, y,z)
        for c in string
                glutBitmapCharacter(font,ascii(c))
        next

func renderStrokeFontString x,y,z,font,string
        glPushMatrix()
        glTranslatef(x, y,z)
        glScalef(0.002, 0.002, 0.002)
        for c in string
                glutStrokeCharacter(font, Ascii(c));
        next
        glPopMatrix()


func restorePerspectiveProjection

        glMatrixMode(GL_PROJECTION)
        // restore previous projection matrix
        glPopMatrix()

        // get back to modelview mode
        glMatrixMode(GL_MODELVIEW)


func setOrthographicProjection

        // switch to projection mode
        glMatrixMode(GL_PROJECTION)

        // save previous matrix which contains the
        //settings for the perspective projection
        glPushMatrix()

        // reset matrix
        glLoadIdentity()

        // set a 2D orthographic projection
        gluOrtho2D(0, w, h, 0)

        // switch back to modelview mode
        glMatrixMode(GL_MODELVIEW)
```

(continues on next page)

(continued from previous page)

```
func computePos deltaMove

        x += deltaMove * lx * 0.1
        z += deltaMove * lz * 0.1


func renderScene

        if  deltaMove
                computePos(deltaMove)
        ok

        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()

        // Set the camera
        gluLookAt(      x, 1.0, z,
                        x+lx, 1.0,  z+lz,
                        0.0, 1.0,  0.0)

        // Draw ground

        glColor3f(0.9, 0.9, 0.9)
        glBegin(GL_QUADS)
                glVertex3f(-100.0, 0.0, -100.0)
                glVertex3f(-100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0, -100.0)
        glEnd()

// Draw 9 SnowMen
        for i = -3 to -1
                for j = -3 to -1
                        glPushMatrix()
                        glTranslatef(i*10.0, 0.0, j * 10.0)
                        drawSnowMan()
                        number = (i+3)*3+(j+3)
                        renderBitmapString(0.0, 0.5, 0.0,font ,""+number)
                        glPopMatrix()
                next
        next

        // Code to compute frames per second
        frame++

        time=glutGet(GLUT_ELAPSED_TIME)
        if time - timebase > 1000
                s = "RingFreeGLUT - FPS: " + (frame*1000.0/(time-timebase))
```

(continues on next page)

(continued from previous page)

```
                timebase = time
                frame = 0
        ok

        // Code to display a string (fps) with bitmap fonts
        setOrthographicProjection()

        glPushMatrix()
        glLoadIdentity()
        renderBitmapString(5,30,0,GLUT_BITMAP_HELVETICA_18,s)
        glPopMatrix()

        restorePerspectiveProjection()

        glutSwapBuffers()


// -----------------------------------
//             KEYBOARD
// -----------------------------------

func processNormalKeys
        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on 27
                        glutDestroyMenu(mainMenu)
                        glutDestroyMenu(fillMenu)
                        glutDestroyMenu(colorMenu)
                        glutDestroyMenu(fontMenu)
                        Shutdown()
        off


func pressKey

        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on GLUT_KEY_UP
                         deltaMove = 0.5
                on GLUT_KEY_DOWN
                        deltaMove = -0.5
        off


func releaseKey
```

(continues on next page)

(continued from previous page)

```
        key = glutEventKey()

        switch key
                on GLUT_KEY_UP
                        deltaMove = 0
                on GLUT_KEY_DOWN
                        deltaMove = 0
        off


// -----------------------------------
//             MOUSE
// -----------------------------------

func mouseMove
        xx = glutEventX()
        yy = glutEventY()

        // this will only be true when the left button is down
        if xOrigin >= 0

                // update deltaAngle
                deltaAngle = (xx - xOrigin) * 0.001

                // update camera's direction
                lx = sin(angle + deltaAngle)
                lz = -cos(angle + deltaAngle)
        ok


func mouseButton

        button = glutEventButton()
        state = glutEventState()
        xx = glutEventX()
        yy = glutEventY()

        // only start motion if the left button is pressed
        if button = GLUT_LEFT_BUTTON
                // when the button is released
                if state = GLUT_UP
                        angle += deltaAngle
                        xOrigin = -1
                else
                        // state = GLUT_DOWN
                        xOrigin = xx
                ok
        ok


// -----------------------------------
//             MENUS
```

(continues on next page)

(continued from previous page)

```
// -----------------------------------

func processMenuStatus

        status = glutEventStatus()

        if status = GLUT_MENU_IN_USE
                menuFlag = 1
        else
                menuFlag = 0
        ok


func processMainMenu

        // nothing to do in here
        // all actions are for submenus


func processFillMenu

        option = glutEventValue()

        switch option

                on C_FILL
                        glPolygonMode(GL_FRONT, GL_FILL)
                on C_LINE
                         glPolygonMode(GL_FRONT, GL_LINE)
        off


func processFontMenu

        option = glutEventValue()

        switch (option) {
                on C_INT_GLUT_BITMAP_8_BY_13
                        font = GLUT_BITMAP_8_BY_13
                on C_INT_GLUT_BITMAP_9_BY_15
                        font = GLUT_BITMAP_9_BY_15
                on C_INT_GLUT_BITMAP_TIMES_ROMAN_10
                        font = GLUT_BITMAP_TIMES_ROMAN_10
                on C_INT_GLUT_BITMAP_TIMES_ROMAN_24
                        font = GLUT_BITMAP_TIMES_ROMAN_24
                on C_INT_GLUT_BITMAP_HELVETICA_10
                        font = GLUT_BITMAP_HELVETICA_10
                on C_INT_GLUT_BITMAP_HELVETICA_12
                        font = GLUT_BITMAP_HELVETICA_12
                on C_INT_GLUT_BITMAP_HELVETICA_18
                        font = GLUT_BITMAP_HELVETICA_18
        off
```

(continues on next page)

(continued from previous page)

```
func processColorMenu

        option = glutEventValue()

        switch option
                on C_RED
                        red = 1.0
                        green = 0.0
                        blue = 0.0
                on C_GREEN
                        red = 0.0
                        green = 1.0
                        blue = 0.0
                on C_BLUE
                        red = 0.0
                        green = 0.0
                        blue = 1.0
                on C_ORANGE
                        red = 1.0
                        green = 0.5
                        blue = 0.5
        off


func createPopupMenus

        fontMenu = glutCreateMenu(:processFontMenu)

        glutAddMenuEntry("BITMAP_8_BY_13 ",C_INT_GLUT_BITMAP_8_BY_13 )
        glutAddMenuEntry("BITMAP_9_BY_15",C_INT_GLUT_BITMAP_9_BY_15 )
        glutAddMenuEntry("BITMAP_TIMES_ROMAN_10 ",C_INT_GLUT_BITMAP_TIMES_ROMAN_10  )
        glutAddMenuEntry("BITMAP_TIMES_ROMAN_24",C_INT_GLUT_BITMAP_TIMES_ROMAN_24  )
        glutAddMenuEntry("BITMAP_HELVETICA_10 ",C_INT_GLUT_BITMAP_HELVETICA_10  )
        glutAddMenuEntry("BITMAP_HELVETICA_12",C_INT_GLUT_BITMAP_HELVETICA_12  )
        glutAddMenuEntry("BITMAP_HELVETICA_18",C_INT_GLUT_BITMAP_HELVETICA_18  )

        fillMenu = glutCreateMenu(:processFillMenu)

        glutAddMenuEntry("Fill",C_FILL)
        glutAddMenuEntry("Line",C_LINE)

        colorMenu = glutCreateMenu(:processColorMenu)
        glutAddMenuEntry("Red",C_RED);
        glutAddMenuEntry("Blue",C_BLUE);
        glutAddMenuEntry("Green",C_GREEN);
        glutAddMenuEntry("Orange",C_ORANGE);

        mainMenu = glutCreateMenu(:processMainMenu)

        glutAddSubMenu("Polygon Mode", fillMenu)
        glutAddSubMenu("Color", colorMenu)
```

(continues on next page)

(continued from previous page)

```
        glutAddSubMenu("Font",fontMenu)
        // attach the menu to the right button
        glutAttachMenu(GLUT_RIGHT_BUTTON)

        // this will allow us to know if the menu is active
        glutMenuStatusFunc(:processMenuStatus)


// -----------------------------------
//             MAIN
// -----------------------------------

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test - 9 SnowMan")

        // register callbacks
        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        glutIgnoreKeyRepeat(1)
        glutKeyboardFunc(:processNormalKeys)
        glutSpecialFunc(:pressKey)
        glutSpecialUpFunc(:releaseKey)

        // here are the two new functions
        glutMouseFunc(:mouseButton)
        glutMotionFunc(:mouseMove)

        // OpenGL init
        glEnable(GL_DEPTH_TEST)
        glEnable(GL_CULL_FACE)

        // init Menus
        createPopupMenus()

        // enter GLUT event processing cycle
        glutMainLoop()
```

Screen Shots:

The First screen shot

The Second screen shot

58.15 Make a Cube using RingOpenGL and RingFreeGLUT

Example:

```
load "freeglut.ring"
load "opengl21lib.ring"

// -----------------------------------------------------------
// Global Variables
// -----------------------------------------------------------
rotate_y=0
rotate_x=0

// -----------------------------------------------------------
// display() Callback function
// -----------------------------------------------------------
func display

  //  Clear screen and Z-buffer
  glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)

  // Reset transformations
  glLoadIdentity()

  // Rotate when user changes rotate_x and rotate_y
  glRotatef( rotate_x, 1.0, 0.0, 0.0 )
  glRotatef( rotate_y, 0.0, 1.0, 0.0 )

  //Multi-colored side - FRONT
  glBegin(GL_POLYGON)

  glColor3f( 1.0, 0.0, 0.0 )     glVertex3f(  0.5, -0.5, -0.5 )      # P1 is red
  glColor3f( 0.0, 1.0, 0.0 )     glVertex3f(  0.5,  0.5, -0.5 )      # P2 is green
  glColor3f( 0.0, 0.0, 1.0 )     glVertex3f( -0.5,  0.5, -0.5 )      # P3 is blue
  glColor3f( 1.0, 0.0, 1.0 )     glVertex3f( -0.5, -0.5, -0.5 )      # P4 is purple

  glEnd()

  // White side - BACK
  glBegin(GL_POLYGON)
  glColor3f(   1.0,  1.0, 1.0 )
  glVertex3f(  0.5, -0.5, 0.5 )
  glVertex3f(  0.5,  0.5, 0.5 )
  glVertex3f( -0.5,  0.5, 0.5 )
  glVertex3f( -0.5, -0.5, 0.5 )
  glEnd()

  // Purple side - RIGHT
  glBegin(GL_POLYGON)
  glColor3f(  1.0,  0.0,  1.0 )
  glVertex3f( 0.5, -0.5, -0.5 )
  glVertex3f( 0.5,  0.5, -0.5 )
  glVertex3f( 0.5,  0.5,  0.5 )
```

(continues on next page)

(continued from previous page)

```
  glVertex3f( 0.5, -0.5,  0.5 )
  glEnd()

  // Green side - LEFT
  glBegin(GL_POLYGON)
  glColor3f(   0.0,  1.0,  0.0 )
  glVertex3f( -0.5, -0.5,  0.5 )
  glVertex3f( -0.5,  0.5,  0.5 )
  glVertex3f( -0.5,  0.5, -0.5 )
  glVertex3f( -0.5, -0.5, -0.5 )
  glEnd()

  // Blue side - TOP
  glBegin(GL_POLYGON)
  glColor3f(   0.0,  0.0,  1.0 )
  glVertex3f(  0.5,  0.5,  0.5 )
  glVertex3f(  0.5,  0.5, -0.5 )
  glVertex3f( -0.5,  0.5, -0.5 )
  glVertex3f( -0.5,  0.5,  0.5 )
  glEnd()

  // Red side - BOTTOM
  glBegin(GL_POLYGON)
  glColor3f(   1.0,  0.0,  0.0 )
  glVertex3f(  0.5, -0.5, -0.5 )
  glVertex3f(  0.5, -0.5,  0.5 )
  glVertex3f( -0.5, -0.5,  0.5 )
  glVertex3f( -0.5, -0.5, -0.5 )
  glEnd()

  glFlush()
  glutSwapBuffers()


// -----------------------------------------------------------
// specialKeys() Callback Function
// -----------------------------------------------------------
func specialKeys

        key = glutEventKey()

  //  Right arrow - increase rotation by 5 degree
        switch Key

        on GLUT_KEY_RIGHT
                rotate_y += 5

        //  Left arrow - decrease rotation by 5 degree
        on GLUT_KEY_LEFT
                rotate_y -= 5

        on GLUT_KEY_UP
```

(continues on next page)

(continued from previous page)

```
                rotate_x += 5

        on GLUT_KEY_DOWN
                rotate_x -= 5

        off

  //  Request display update
  glutPostRedisplay()



// ----------------------------------------------------------
// main() function
// ----------------------------------------------------------
func main

  //  Initialize GLUT and process user parameters
  glutInit()

  //  Request double buffered true color window with Z-buffer
  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)

  // Create window
  glutCreateWindow("Awesome Cube")

  //  Enable Z-buffer depth test
  glEnable(GL_DEPTH_TEST)

  // Callback functions
  glutDisplayFunc(:display)
  glutSpecialFunc(:specialKeys)

  //  Pass control to GLUT for events
  glutMainLoop()

  //  Return to OS
```

Screen Shot:

Ring Notepad
File Edit View Program Browser Tools Help
Main File :
Project Files
Source Code : B:/ring/extensions/ringfreeglut/tests/cube.ring
extensions
codegen
ringallegro
ringcurl
ringfreeglut
freeglut.ring
gendoc.ring
ringfreeglutfuncs...
ring_freeglut.rh
tests
cube.ring
test.ring
test10.ring
test11.ring
test12.ring
test13.ring
test14.ring
test15.ring
test16.ring
test2.ring
test3.ring
test4.ring
test5.ring
test6.ring
test7.ring
test8.ring
test9.ring
ringinternet
ringmysql
ringodbc
ringopengl
ringopenssl
ringqt
ringsdl
ringsqlite
ringzip
graphics
include
22 glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
23
24 // Reset transformations
25 glLoadIdentity()
26
27 // Rotate when user changes rotate_x and rotate_y
28 glRotatef(rotate_x, 1.0, 0.0, 0.0)
29 glRotatef(rotate_y, 0.0, 1.0, 0.0)
30
31 //Multi-colored side - FRONT
32 glBegin(GL_POLYGON)
33
34 glColor3f(1.0, 0.0, 0.0) glVertex3f(0.5, -0.5, -0.5) # P1 is red
35 glColor3f(0.0, 1.0, 0.0) glVertex3f(0.5, 0.5, -0.5) # P2 is green
36 glColor3f(0.0, 0.0, 1.0) glVertex3f(-0.5, 0.5, -0.5) # P3 is blue
37 glColor3f(1.0, 0.0, 1.0) glVertex3f(-0.5, -0.5, -0.5) # P4 is purple
38
Awesome Cube
Line : 31 Column : 28 Total Lines : 145

CHAPTER

FIFTYNINE

USING RINGOPENGL AND RINGALLEGRO FOR 3D GRAPHICS

In this chapter we will learn about using RingOpenGL and RingAllegro

59.1 3D Cube and Texture

Source Code:

```
# Load Libraries
        load "gamelib.ring"             # RingAllegro Library
        load "opengl21lib.ring"         # RingOpenGL  Library

#==============================================================
# To Support MacOS X
        al_run_main()
        func al_game_start      # Called by al_run_main()
                main()          # Now we call our main function
#==============================================================

func main

        new GraphicsApp {
                start()
        }


class GraphicsApp from GraphicsAppBase

        TITLE = "Ring Cube"

        bitmap texture

        xrot = 0.0
        yrot = 0.0
        zrot = 0.0

        func loadresources

                bitmap = al_load_bitmap("ring.bmp")
                texture = al_get_opengl_texture(bitmap)
```

(continues on next page)

(continued from previous page)

```
        func destroyResources

                al_destroy_bitmap(bitmap)

        func drawScene

                w = 800 h = 600
                ratio =  w / h

                glViewport(0, 0, w, h)
                glMatrixMode(GL_PROJECTION)
                glLoadIdentity()

                gluPerspective(45,ratio,1,100)
                glMatrixMode(GL_MODELVIEW)
                glLoadIdentity()

                glEnable(GL_TEXTURE_2D)
                glShadeModel(GL_SMOOTH)
                glClearColor(0.0, 0.0, 0.0, 0.5)
                glClearDepth(1.0)
                glEnable(GL_DEPTH_TEST)
                glEnable(GL_CULL_FACE)
                glDepthFunc(GL_LEQUAL)
                glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)

                glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
                glLoadIdentity()
                glTranslatef(0.0,0.0,-5.0)

                glRotatef(xrot,1.0,0.0,0.0)
                glRotatef(yrot,0.0,1.0,0.0)
                glRotatef(zrot,0.0,0.0,1.0)

                glBindTexture(GL_TEXTURE_2D, texture)

                glBegin(GL_QUADS)
                        // Front Face
                        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
                        // Back Face
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
                        // Top Face
                        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
```

(continues on next page)

(continued from previous page)

```
                        // Bottom Face
                        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                        // Right face
                        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                        // Left Face
                        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                glEnd()

                xrot += 0.3
                yrot += 0.2
                zrot += 0.4


class GraphicsAppBase

        display event_queue ev timeout
        timer  redraw   = true

        FPS             = 60

        SCREEN_W        = 800
        SCREEN_H        = 600

        KEY_UP          = 1
        KEY_DOWN        = 2
        KEY_LEFT        = 3
        KEY_RIGHT       = 4

        Key = [false,false,false,false]

        TITLE = "Graphics Application"

        func start

                SetUp()
                loadResources()
                eventsLoop()
                destroy()

        func setup

                al_init()
                al_init_image_addon()
```

(continues on next page)

(continued from previous page)

```
            al_set_new_display_flags(ALLEGRO_OPENGL)
            display = al_create_display(SCREEN_W,SCREEN_H)
            al_set_Window_title(display,TITLE)
            al_clear_to_color(al_map_rgb(0,0,0))
            event_queue = al_create_event_queue()
            al_register_event_source(event_queue,
                    al_get_display_event_source(display))
            ev = al_new_allegro_event()
            timeout = al_new_allegro_timeout()
            al_init_timeout(timeout, 0.06)
            timer = al_create_timer(1.0 / FPS)
            al_register_event_source(event_queue,
                    al_get_timer_event_source(timer))
            al_start_timer(timer)
            al_install_mouse()
            al_register_event_source(event_queue,
                    al_get_mouse_event_source())
            al_install_keyboard()
            al_register_event_source(event_queue,
                    al_get_keyboard_event_source())

    func eventsLoop

            while true
                    al_init_timeout(timeout, 0.06)
                    al_wait_for_event_until(event_queue, ev, timeout)
                    switch al_get_allegro_event_type(ev)
                    on ALLEGRO_EVENT_DISPLAY_CLOSE
                            exit
                    on ALLEGRO_EVENT_TIMER
                            redraw = true
                    on ALLEGRO_EVENT_MOUSE_AXES
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                    on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                    on ALLEGRO_EVENT_MOUSE_BUTTON_UP
                            exit
                    on ALLEGRO_EVENT_KEY_DOWN
                            switch al_get_allegro_event_keyboard_keycode(ev)
                                    on ALLEGRO_KEY_UP
                                            key[KEY_UP] = true
                                    on ALLEGRO_KEY_DOWN
                                            key[KEY_DOWN] = true
                                    on ALLEGRO_KEY_LEFT
                                            key[KEY_LEFT] = true
                                    on ALLEGRO_KEY_RIGHT
                                            key[KEY_RIGHT] = true
                            off
                    on ALLEGRO_EVENT_KEY_UP
                            switch al_get_allegro_event_keyboard_keycode(ev)
```

(continues on next page)

(continued from previous page)

```
                                        on ALLEGRO_KEY_UP
                                                key[KEY_UP] = false
                                        on ALLEGRO_KEY_DOWN
                                                key[KEY_DOWN] = false
                                        on ALLEGRO_KEY_LEFT
                                                key[KEY_LEFT] = false
                                        on ALLEGRO_KEY_RIGHT
                                                key[KEY_RIGHT] = false
                                        on ALLEGRO_KEY_ESCAPE
                                                exit
                                off
                        off
                        if redraw and al_is_event_queue_empty(event_queue)
                                redraw = false
                                drawScene()
                                al_flip_display()
                        ok
                        callgc()
                end

        func destroy

                destroyResources()
                al_destroy_timer(timer)
                al_destroy_allegro_event(ev)
                al_destroy_allegro_timeout(timeout)
                al_destroy_event_queue(event_queue)
                al_destroy_display(display)
                al_exit()

        func loadresources

        func drawScene

        func destroyResources
```

Screen Shot:

59.2 Many Cubes

Source Code:

```
# Load Libraries
        load "gamelib.ring"             # RingAllegro Library
        load "opengl21lib.ring"         # RingOpenGL  Library

#==============================================================
# To Support MacOS X
        al_run_main()
        func al_game_start      # Called by al_run_main()
                main()          # Now we call our main function
#==============================================================


func main

        new GraphicsApp {
                start()
        }


class GraphicsApp from GraphicsAppBase

        TITLE = "Many Cubes"

        bitmap bitmap2 bitmap3
        texture texture2 texture3
```

(continues on next page)

(continued from previous page)

```
        fps = 120
        xrot = 0.0
        yrot = 0.0
        zrot = 0.0

        nPerspective = 100

        func loadresources

                bitmap = al_load_bitmap("sky1.jpg")
                texture = al_get_opengl_texture(bitmap)

                bitmap2 = al_load_bitmap("sky2.jpg")
                texture2 = al_get_opengl_texture(bitmap2)

                bitmap3 = al_load_bitmap("sky3.jpg")
                texture3 = al_get_opengl_texture(bitmap3)

        func destroyResources

                al_destroy_bitmap(bitmap)
                al_destroy_bitmap(bitmap2)
                al_destroy_bitmap(bitmap3)

        func drawScene

                prepare()
                cubes()
                rotate()

        func Prepare
                w = 800 h = 600
                ratio =  w / h
                glViewport(0, 0, w, h)
                glMatrixMode(GL_PROJECTION)
                glLoadIdentity()
                gluPerspective(-nPerspective,ratio,1,nPerspective)
                glMatrixMode(GL_MODELVIEW)
                glLoadIdentity()
                glEnable(GL_TEXTURE_2D)
                glShadeModel(GL_SMOOTH)
                glClearColor(0.0, 0.0, 0.0, 0.5)
                glClearDepth(1.0)
                glEnable(GL_DEPTH_TEST)
                glEnable(GL_CULL_FACE)
                glDepthFunc(GL_LEQUAL)
                glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)
                glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        func Cubes
                cube(5,-3.4,-5,:sky1)
                cube(0,-3,-5,:sky1)
```

(continues on next page)

(continued from previous page)

```
            cube(-5,-3,-5,:sky1)
            cube(5,0.5,-5,:sky2)
            cube(0,0.5,-5,:sky2)
            cube(-5,0.5,-5,:sky2)
            cube(5,4,-5,:sky3)
            cube(0,4,-5,:sky3)
            cube(-5,4,-5,:sky3)

    func Rotate
            xrot += 0.3 * 5
            yrot += 0.2 * 5
            zrot += 0.4 * 5
            nPerspective += 0.5


    func cube(x,y,z,nTexture)
            glLoadIdentity()
            glTranslatef(x,y,z)
            glRotatef(xrot,1.0,0.0,0.0)
            glRotatef(yrot,0.0,1.0,0.0)
            glRotatef(zrot,0.0,0.0,1.0)
            drawcube(nTexture)

    func drawcube(cTexture)

            switch cTexture
                    on :sky1
                            glBindTexture(GL_TEXTURE_2D, texture)
                    on :sky2
                            glBindTexture(GL_TEXTURE_2D, texture2)
                    on :sky3
                            glBindTexture(GL_TEXTURE_2D, texture3)
            off


            glBegin(GL_QUADS)
                    // Front Face
                    glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                    glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                    glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
                    glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
                    // Back Face
                    glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
                    glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                    glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                    glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
                    // Top Face
                    glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                    glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
                    glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
                    glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                    // Bottom Face
```

(continues on next page)

(continued from previous page)

```
                        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)

                        // Right face
                        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)

                        // Left Face
                        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                glEnd()


class GraphicsAppBase

        display event_queue ev timeout
        timer  redraw   = true

        FPS             = 60

        SCREEN_W        = 800
        SCREEN_H        = 600

        KEY_UP          = 1
        KEY_DOWN        = 2
        KEY_LEFT        = 3
        KEY_RIGHT       = 4

        Key = [false,false,false,false]

        TITLE = "Graphics Application"

        func start

                SetUp()
                loadResources()
                eventsLoop()
                destroy()

        func setup

                al_init()
                al_init_image_addon()
                al_set_new_display_flags(ALLEGRO_OPENGL)
                display = al_create_display(SCREEN_W,SCREEN_H)
                al_set_Window_title(display,TITLE)
```

(continues on next page)

(continued from previous page)

```
            al_clear_to_color(al_map_rgb(0,0,0))
            event_queue = al_create_event_queue()
            al_register_event_source(event_queue,
                    al_get_display_event_source(display))
            ev = al_new_allegro_event()
            timeout = al_new_allegro_timeout()
            al_init_timeout(timeout, 0.06)
            timer = al_create_timer(1.0 / FPS)
            al_register_event_source(event_queue,
                    al_get_timer_event_source(timer))
            al_start_timer(timer)
            al_install_mouse()
            al_register_event_source(event_queue,
                    al_get_mouse_event_source())
            al_install_keyboard()
            al_register_event_source(event_queue,
                    al_get_keyboard_event_source())

    func eventsLoop

            while true
                    al_init_timeout(timeout, 0.06)
                    al_wait_for_event_until(event_queue, ev, timeout)
                    switch al_get_allegro_event_type(ev)
                    on ALLEGRO_EVENT_DISPLAY_CLOSE
                            exit
                    on ALLEGRO_EVENT_TIMER
                            redraw = true
                    on ALLEGRO_EVENT_MOUSE_AXES
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                    on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                    on ALLEGRO_EVENT_MOUSE_BUTTON_UP
                            exit
                    on ALLEGRO_EVENT_KEY_DOWN
                            switch al_get_allegro_event_keyboard_keycode(ev)
                                    on ALLEGRO_KEY_UP
                                            key[KEY_UP] = true
                                    on ALLEGRO_KEY_DOWN
                                            key[KEY_DOWN] = true
                                    on ALLEGRO_KEY_LEFT
                                            key[KEY_LEFT] = true
                                    on ALLEGRO_KEY_RIGHT
                                            key[KEY_RIGHT] = true
                            off
                    on ALLEGRO_EVENT_KEY_UP
                            switch al_get_allegro_event_keyboard_keycode(ev)
                                    on ALLEGRO_KEY_UP
                                            key[KEY_UP] = false
                                    on ALLEGRO_KEY_DOWN
```

(continues on next page)

(continued from previous page)

```
                                        key[KEY_DOWN] = false
                                on ALLEGRO_KEY_LEFT
                                        key[KEY_LEFT] = false
                                on ALLEGRO_KEY_RIGHT
                                        key[KEY_RIGHT] = false
                                on ALLEGRO_KEY_ESCAPE
                                        exit
                        off
                off
                if redraw and al_is_event_queue_empty(event_queue)
                        redraw = false
                        drawScene()
                        al_flip_display()
                ok
                callgc()
        end

    func destroy

        destroyResources()
        al_destroy_timer(timer)
        al_destroy_allegro_event(ev)
        al_destroy_allegro_timeout(timeout)
        al_destroy_event_queue(event_queue)
        al_destroy_display(display)
        al_exit()

    func loadresources

    func drawScene

    func destroyResources
```

Screen Shot:

59.3 TicTacToe 3D Game

Source Code:

```
# Load Libraries
        load "gamelib.ring"             # RingAllegro Library
        load "opengl21lib.ring"         # RingOpenGL  Library

#==============================================================
# To Support MacOS X
        al_run_main()
        func al_game_start      # Called by al_run_main()
                main()          # Now we call our main function
#==============================================================

func main
        new TicTacToe3D {
                start()
        }
```

(continues on next page)

(continued from previous page)

```
class TicTacToe3D from GameLogic

        FPS = 60
        TITLE = "TicTacToe 3D"

        oBackground = new GameBackground
        oGameSound = new GameSound
        oGameCube = new GameCube
        oGameOver = new GameOver
        oGameInterface = new GameInterface

        func loadresources
                oGameOver.loadresources()
                oGameSound.loadresources()
                oBackGround.loadresources()
                oGameCube.loadresources()

        func destroyResources
                oGameOver.destroyResources()
                oGameSound.destroyResources()
                oBackGround.destroyResources()
                oGameCube.destroyResources()

        func drawScene
                oBackground.update()
                oGameInterface.update(self)

        func MouseClickEvent
                oGameInterface.MouseClickEvent(self)

class GameInterface

        func Update oGame
                prepare()
                cubes(oGame)

        func Prepare
                w = 1024 h = 768
                ratio =  w / h
                glViewport(0, 0, w, h)
                glMatrixMode(GL_PROJECTION)
                glLoadIdentity()
                gluPerspective(-120,ratio,1,120)
                glMatrixMode(GL_MODELVIEW)
                glLoadIdentity()
                glEnable(GL_TEXTURE_2D)
                glShadeModel(GL_SMOOTH)
                glClearColor(0.0, 0.0, 0.0, 0.5)
                glClearDepth(1.0)
                glEnable(GL_DEPTH_TEST)
                glEnable(GL_CULL_FACE)
                glDepthFunc(GL_LEQUAL)
```

(continues on next page)

(continued from previous page)

```
                glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)

        func Cubes oGame
                oGame.oGameCube {
                        aGameMap = oGame.aGameMap
                        cube( 5  , -3 , -5  , aGameMap[1][1] )
                        cube( 0  , -3 , -5  , aGameMap[1][2] )
                        cube( -5 , -3 , -5  , aGameMap[1][3] )
                        cube( 5  , 1  , -5  , aGameMap[2][1] )
                        cube( 0  , 1  , -5  , aGameMap[2][2] )
                        cube( -5 , 1  , -5  , aGameMap[2][3] )
                        cube( 5  , 5  , -5  , aGameMap[3][1] )
                        cube( 0  , 5  , -5  , aGameMap[3][2] )
                        cube( -5 , 5  , -5  , aGameMap[3][3] )
                        rotate()
                }

        func MouseClickEvent oGame
                oGame {
                        aBtn = Point2Button(Mouse_X,Mouse_Y)
                        nRow = aBtn[1]
                        nCol = aBtn[2]
                        if nRow != 0 and nCol != 0
                                if aGameMap[nRow][nCol] = :n
                                        aGameMap[nRow][nCol] = cActivePlayer
                                        ChangeActivePlayer()
                                        CheckGameOver()
                                ok
                        ok
                }

Class GameLogic from GraphicsAppBase

        aGameMap = [
                [ :n , :n , :n ] ,
                [ :n , :n , :n ] ,
                [ :n , :n , :n ]
        ]

        aGameButtons = [                        # x1,y1,x2,y2
                [176,88,375,261],               # [1,1]
                [423,88,591,261],               # [1,2]
                [645,88,876,261],               # [1,3]
                [176,282,375,428],              # [2,1]
                [423,282,591,428],              # [2,2]
                [645,282,876,428],              # [2,3]
                [176,454,375,678],              # [3,1]
                [423,454,591,678],              # [3,2]
                [645,454,876,678]               # [3,3]
        ]

        cActivePlayer = :x
```

(continues on next page)

(continued from previous page)

```
func point2button x,y
        nRow = 0
        nCol = 0
        for t = 1 to len(aGameButtons)
                rect = aGameButtons[t]
                if x >= rect[1] and x <= rect[3] and
                   y >= rect[2] and y <= rect[4]
                                switch t
                                        on 1  nRow = 1  nCol = 1
                                        on 2  nRow = 1  nCol = 2
                                        on 3  nRow = 1  nCol = 3
                                        on 4  nRow = 2  nCol = 1
                                        on 5  nRow = 2  nCol = 2
                                        on 6  nRow = 2  nCol = 3
                                        on 7  nRow = 3  nCol = 1
                                        on 8  nRow = 3  nCol = 2
                                        on 9  nRow = 3  nCol = 3
                                off
                                exit
                ok
        next
        return [nRow,nCol]

func ChangeActivePlayer()
        if cActivePlayer = :x
                cActivePlayer = :o
        else
                cActivePlayer = :x
        ok

func CheckGameOver
        aList = [
                aGameMap[1][1],
                aGameMap[1][2],
                aGameMap[1][3],
                aGameMap[2][1],
                aGameMap[2][2],
                aGameMap[2][3],
                aGameMap[3][1],
                aGameMap[3][2],
                aGameMap[3][3]
        ]
        for item in aList
                switch item
                        on :x   item = 1
                        on :o   item = 2
                        on :n   item = 0
                off
        next
        nStatus = CheckWinner(aList)
        if nStatus
```

(continues on next page)

(continued from previous page)

```
                        oGameOver {
                                Switch nStatus
                                        on 1 Player1Win(this)
                                        on 2 Player2Win(this)
                                        on 3 NoOneWin(this)
                                off
                        }
                        refreshGame()
                ok

        func refreshGame
                aGameMap = [
                        [ :n , :n , :n ] ,
                        [ :n , :n , :n ] ,
                        [ :n , :n , :n ]
                ]
                cActivePlayer = :x

        func CheckWinner lst
                //vertical check
                for v=1 to 9 step 3
                        if lst[v]!=0 and lst[v+1]!=0 and lst[v+2]!=0
                                if lst[v]=lst[v+1] and lst[v+1]=lst[v+2]
                                        return lst[v]
                                ok
                        ok
                next
                //horizontal
                for h=1 to 3
                        if lst[h]!=0 and lst[h+3]!=0 and lst[h+6]!=0
                                if lst[h]=lst[h+3] and lst[h+3]=lst[h+6]
                                        return lst[h]
                                ok
                        ok
                next
                //Cross
                if lst[1]!=0 and lst[5]!=0 and lst[9]!=0
                        if lst[1]=lst[5] and lst[5]=lst[9] return lst[1] ok
                ok
                if lst[3]!=0 and lst[5]!=0 and lst[7]!=0
                        if lst[3]=lst[5] and lst[5]=lst[7] return lst[3] ok
                ok
                //tie
                tie=true
                for i=1 to 9
                        if lst[i]=0 tie=false exit ok
                next
                if tie=true return 3 ok return 0



class GameOver
```

(continues on next page)

(continued from previous page)

```
        font bitmap

        func loadresources
                font = al_load_ttf_font("font/pirulen.ttf",54,0 )
                bitmap = al_load_bitmap("image/ballon.png")

        func destroyResources
                al_destroy_bitmap(bitmap)
                al_destroy_font(font)

        func Player1Win oGame
                showMsg(oGame,80,430,"Good job X you won!")

        func Player2Win oGame
                showMsg(oGame,80,430,"Good job O you won!")

        func NoOneWin oGame
                showMsg(oGame,150,430,"Oh no it's a tie!")

        func ShowMsg oGame,x,y,cMsg
                oGame {
                        drawScene()
                        al_flip_display()
                        al_rest(0.3)
                        newdisplay = al_create_display(SCREEN_W,SCREEN_H)
                        al_set_window_title(newdisplay,TITLE)
                        al_clear_to_color(al_map_rgb(255,255,255))
                        al_draw_bitmap(this.bitmap,200,50,1)
                        al_draw_text(this.font,
                                 al_map_rgb(0,0,255), x,y,
                                 ALLEGRO_ALIGN_LEFT,cMsg)
                        al_flip_display()
                        al_rest(2)
                        al_destroy_display(newdisplay)
                        al_set_target_backbuffer(display)
                }

class GameCube

        bitmap bitmap2 bitmap3
        textureX textureO textureN

        xrot = 0.0
        yrot = 0.0
        zrot = 0.0

        func loadresources
                bitmap = al_load_bitmap("image/o.png")
                textureO = al_get_opengl_texture(bitmap)
                bitmap2 = al_load_bitmap("image/x.png")
                textureX = al_get_opengl_texture(bitmap2)
```

(continues on next page)

(continued from previous page)

```
            bitmap3 = al_load_bitmap("image/empty.png")
            textureN = al_get_opengl_texture(bitmap3)

    func destroyResources
            al_destroy_bitmap(bitmap)
            al_destroy_bitmap(bitmap2)
            al_destroy_bitmap(bitmap3)

    func cube(x,y,z,nTexture)
            glLoadIdentity()
            glTranslatef(x,y,z)
            glRotatef(xrot,1.0,0.0,0.0)
            glRotatef(yrot,0.0,1.0,0.0)
            glRotatef(zrot,0.0,0.0,1.0)
            setCubeTexture(nTexture)
            drawCube()

    func setCubeTexture cTexture
            switch cTexture
                    on :x
                            glBindTexture(GL_TEXTURE_2D, textureX)
                    on :o
                            glBindTexture(GL_TEXTURE_2D, textureO)
                    on :n
                            glBindTexture(GL_TEXTURE_2D, textureN)
            off

    func Rotate
            xrot += 0.3 * 5
            yrot += 0.2 * 5
            zrot += 0.4 * 5

    func drawcube
            glBegin(GL_QUADS)
                    // Front Face
                    glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                    glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                    glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
                    glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
                    // Back Face
                    glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
                    glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                    glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                    glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
                    // Top Face
                    glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                    glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
                    glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
                    glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                    // Bottom Face
                    glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
                    glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
```

(continues on next page)

(continued from previous page)

```
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)

                        // Right face
                        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)

                        // Left Face
                        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                glEnd()


class GameBackground

        nBackX = 0
        nBackY = 0
        nBackDiffx = -1
        nBackDiffy = -1
        nBackMotion = 1
        aBackMotionList = [
                [ -1, -1 ] ,            # Down - Right
                [ 0 , 1  ] ,            # Up
                [ -1, -1 ] ,            # Down - Right
                [ 0 , 1  ] ,            # Up
                [ 1 , -1 ] ,            # Down - Left
                [ 0 , 1  ] ,            # Up
                [ 1 , -1 ] ,            # Down - Left
                [ 0 , 1  ]                      # Up
        ]

        bitmap

        func Update
                draw()
                motion()

        func draw
                al_draw_bitmap(bitmap,nBackX,nBackY,1)

        func motion
                nBackX += nBackDiffx
                nBackY += nBackDiffy
                if (nBackY = -350) or (nBackY = 0)
                        nBackMotion++
                        if nBackMotion > len(aBackMotionList)
                                nBackMotion = 1
                        ok
```

(continues on next page)

(continued from previous page)

```
                        nBackDiffx  = aBackMotionList[nBackMotion][1]
                        nBackDiffy  = aBackMotionList[nBackMotion][2]
                ok

        func loadResources
                bitmap = al_load_bitmap("image/back.jpg")

        func destroyResources
                al_destroy_bitmap(bitmap)


class GameSound

        sample sampleid

        func loadresources
                sample = al_load_sample( "sound/music1.wav" )
                sampleid = al_new_allegro_sample_id()
                al_play_sample(sample, 1.0, 0.0,1.0,ALLEGRO_PLAYMODE_LOOP,sampleid)

        func destroyResources
                al_destroy_allegro_sample_id(sampleid)
                al_destroy_sample(sample)


class GraphicsAppBase

        display event_queue ev timeout
        timer
        redraw                  = true
        FPS                     = 60
        SCREEN_W                = 1024
        SCREEN_H                = 700
        KEY_UP                  = 1
        KEY_DOWN                = 2
        KEY_LEFT                = 3
        KEY_RIGHT               = 4
        Key                     = [false,false,false,false]
        Mouse_X                 = 0
        Mouse_Y                 = 0
        TITLE                   = "Graphics Application"
        PRINT_MOUSE_XY  = False

        func start
                SetUp()
                loadResources()
                eventsLoop()
                destroy()

        func setup
                al_init()
                al_init_font_addon()
```

(continues on next page)

(continued from previous page)

```
            al_init_ttf_addon()
            al_init_image_addon()
            al_install_audio()
            al_init_acodec_addon()
            al_reserve_samples(1)
            al_set_new_display_flags(ALLEGRO_OPENGL)
            display = al_create_display(SCREEN_W,SCREEN_H)
            al_set_window_title(display,TITLE)
            al_clear_to_color(al_map_rgb(0,0,0))
            event_queue = al_create_event_queue()
            al_register_event_source(event_queue,
                    al_get_display_event_source(display))
            ev = al_new_allegro_event()
            timeout = al_new_allegro_timeout()
            al_init_timeout(timeout, 0.06)
            timer = al_create_timer(1.0 / FPS)
            al_register_event_source(event_queue,
                    al_get_timer_event_source(timer))
            al_start_timer(timer)
            al_install_mouse()
            al_register_event_source(event_queue,
                    al_get_mouse_event_source())
            al_install_keyboard()
            al_register_event_source(event_queue,
                    al_get_keyboard_event_source())

    func eventsLoop
            while true
                    al_init_timeout(timeout, 0.06)
                    al_wait_for_event_until(event_queue, ev, timeout)
                    switch al_get_allegro_event_type(ev)
                    on ALLEGRO_EVENT_DISPLAY_CLOSE
                            CloseEvent()
                    on ALLEGRO_EVENT_TIMER
                            redraw = true
                    on ALLEGRO_EVENT_MOUSE_AXES
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                            if PRINT_MOUSE_XY
                                    see "x = " + mouse_x + nl
                                    see "y = " + mouse_y + nl
                            ok
                    on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                    on ALLEGRO_EVENT_MOUSE_BUTTON_UP
                            MouseClickEvent()
                    on ALLEGRO_EVENT_KEY_DOWN
                            switch al_get_allegro_event_keyboard_keycode(ev)
                                    on ALLEGRO_KEY_UP
                                            key[KEY_UP] = true
                                    on ALLEGRO_KEY_DOWN
```

(continues on next page)

(continued from previous page)

```
                                        key[KEY_DOWN] = true
                                on ALLEGRO_KEY_LEFT
                                        key[KEY_LEFT] = true
                                on ALLEGRO_KEY_RIGHT
                                        key[KEY_RIGHT] = true
                        off
                on ALLEGRO_EVENT_KEY_UP
                        switch al_get_allegro_event_keyboard_keycode(ev)
                                on ALLEGRO_KEY_UP
                                        key[KEY_UP] = false
                                on ALLEGRO_KEY_DOWN
                                        key[KEY_DOWN] = false
                                on ALLEGRO_KEY_LEFT
                                        key[KEY_LEFT] = false
                                on ALLEGRO_KEY_RIGHT
                                        key[KEY_RIGHT] = false
                                on ALLEGRO_KEY_ESCAPE
                                        exit
                        off
                off
                if redraw and al_is_event_queue_empty(event_queue)
                        redraw = false
                        drawScene()
                        al_flip_display()
                ok
                callgc()
        end

    func destroy
        destroyResources()
        al_destroy_timer(timer)
        al_destroy_allegro_event(ev)
        al_destroy_allegro_timeout(timeout)
        al_destroy_event_queue(event_queue)
        al_destroy_display(display)
        al_exit()

    func loadresources

    func drawScene

    func destroyResources

    func MouseClickEvent
        exit                    # Exit from the Events Loop

    func CloseEvent
        exit                    # Exit from the Events Loop
```

Screen Shot:

59.4 More 3D Samples

You will find the samples in ring/samples/3D folder

The next screen shot for the Top-Down view - Many levels of cubes sample

The next screen shot for the Camera Sample

The next screen shot for the Camera and background sample

Developer : Azzeddine Remmal

Camera and Background Sample

CHAPTER

SIXTY

THE GOLD MAGIC 800 GAME

In this chapter we will learn about the Gold Magic 800 Game

The game is developed using Ring, RingAllegro and RingOpenGL

You will find the game in ring/applications/goldmagic800 folder

60.1 The Game Story

Your friend discovered a unique and special box, created by the best wizard in the world 7000 years ago, when you close this box and move it; you will find a new gold under the box, it's an infinite source of gold. The Box exists in a special environment full of puzzles, no one can enter this environment because it's protected by the magic. Your friend created a new robot using nanotechnology that can move the box using a remote control device. Your mission is to solve all of these puzzles and get this box forever to be the richest people in the world.

60.2 How to play?

The Gold Magic 800 is a puzzle game that will teach your unconscious mind to think like an entrepreneur.

The game is based on moving your box around to get gold score (=800) to open the First Door (Box Number 1) Then directly put your box on the Door (this will open the next door), Then continue to put your box on all of the next doors in the level, You need the score (800) only for the first door, The next doors doesn't require this condition, but your way of gold will be converted to a wall once you put the Box on any door, so select your path carefully.

60.3 What will you learn?

1- Plan First

2- Move to your target directly then go to get the required resources

3- Look to the future when you evaluate the different solutions

4- Try to avoid mistakes, Also learn from them

5- Respect the Cost (800)

6- Focus and be careful

7- Be patient and Enjoy!

60.4 Screen Shots

We can select the level

The next screen shot for level (1)

The Gold Magic 800 Level Editor

60.5 Source Code

You will find the Level Editor source code in this folder

https://github.com/ring-lang/ring/tree/master/applications/goldmagic800/editor

The user interface of the Level Editor is designed using the Ring Form Designer.

The next file contains the Level Editor Controller Class

https://github.com/ring-lang/ring/blob/master/applications/goldmagic800/editor/editorController.ring

You will find the Game Engine source code in this folder

https://github.com/ring-lang/ring/tree/master/applications/goldmagic800

CHAPTER

SIXTYONE

RINGTILENGINE EXTENSION

In this chapter we will learn about using the RingTilengine extension.

This extension provides complete support for Tilengine

The 2D retro graphics engine with raster effects

Tilengine URL: https://github.com/megamarc/Tilengine

Useful links (Original documentation for C programmers)

- Getting started: https://github.com/ring-lang/Tilengine/blob/master/docs/quick_guide.md
- Initialization: https://github.com/ring-lang/Tilengine/blob/master/docs/starting.md
- Using the window: https://github.com/ring-lang/Tilengine/blob/master/docs/windowing.md
- External rendering: https://github.com/ring-lang/Tilengine/blob/master/docs/rendering.md
- Background layers: https://github.com/ring-lang/Tilengine/blob/master/docs/layers.md
- Sprites: https://github.com/ring-lang/Tilengine/blob/master/docs/sprites.md

Useful tools

- Tiled Map Editor: https://www.mapeditor.org/
- aseprite: https://www.aseprite.org/
- Piskel: https://www.piskelapp.com/
- GrafX2: http://grafx2.chez.com/

Support

- Tilengine Forum: http://www.tilengine.org/forum/

61.1 Getting Started

Example (1):

```
load "tilengine.ring"

TLN_Init(400, 240, 1, 0, 0)
TLN_SetLoadPath("assets\sonic")
foreground = TLN_LoadTilemap ("Sonic_md_fg1.tmx", NULL)
TLN_SetLayerTilemap(0, foreground)
```

(continues on next page)

(continued from previous page)

```
TLN_CreateWindow(NULL, 0)
while TLN_ProcessWindow()
        TLN_DrawFrame(0)
end

TLN_DeleteTilemap(foreground)
TLN_Deinit()
```

Screen Shot:

61.2 More Samples

Example (2):

```
load "tilengine.ring"

WIDTH  = 400
HEIGHT = 240

frame = 0

/* setup engine */
TLN_Init(WIDTH, HEIGHT, 1,0,0)

/* load layer */
tilemap = TLN_LoadTilemap("assets/sonic/Sonic_md_fg1.tmx", NULL)
```

(continues on next page)

(continued from previous page)

```
/* setup the layer */
TLN_SetLayer(0, NULL, tilemap)
TLN_SetBGColor(32,32,128)

/* main loop */
TLN_CreateWindow(NULL, 0)
while TLN_ProcessWindow ()

        /* scroll the layer, one pixel per frame */
        TLN_SetLayerPosition (0, frame, 0)

        /* render to the window */
        TLN_DrawFrame (0)

        frame++
end

/* release resources */
TLN_DeleteTilemap(tilemap)
TLN_Deinit()
```

Screen Shot:

Example (3):

```
load "tilengine.ring"
```

(continues on next page)

(continued from previous page)

```
/* engine init */
TLN_Init(640,480,0,0,1)

/* load resources */
TLN_SetLoadPath("assets/color")
background = TLN_LoadBitmap("beach.png")
palette    = TLN_GetBitmapPalette(background)
sp         = TLN_LoadSequencePack("beach.sqx")
sequence   = TLN_FindSequence(sp, "beach")

/* setup */
TLN_SetBGBitmap(background)
TLN_SetPaletteAnimation(0, palette, sequence, true)

/* main loop */
TLN_CreateWindow(NULL, 0)
TLN_DisableCRTEffect()
while(TLN_ProcessWindow())
        TLN_DrawFrame(0)
end

TLN_DeleteBitmap(background)
TLN_DeleteSequencePack(sp)
TLN_Deinit()
```

Screen Shot:

Example (4):

```
load "tilengine.ring"
```

(continues on next page)

(continued from previous page)

```
WIDTH = 400
HEIGHT= 240

LAYER_FOREGROUND = 0
LAYER_BACKGROUND = 1
MAX_LAYER = 2


player_x = -16
player_y = 160

/* basic setup */
TLN_Init(WIDTH, HEIGHT, MAX_LAYER,1,0)
TLN_SetBGColor(0, 96, 184)

/* load resources */
TLN_SetLoadPath("assets/smw")
foreground = TLN_LoadTilemap("smw_foreground.tmx", NULL)
background = TLN_LoadTilemap("smw_background.tmx", NULL)
TLN_SetLayerTilemap(LAYER_FOREGROUND, foreground)
TLN_SetLayerTilemap(LAYER_BACKGROUND, background)
TLN_SetLayerPosition(LAYER_FOREGROUND, 0,48)
TLN_SetLayerPosition(LAYER_BACKGROUND, 0,80)

/* setup sprite */
spriteset = TLN_LoadSpriteset("smw_sprite")
TLN_SetSpriteSet(0, spriteset)
TLN_SetSpritePicture(0, 0)
TLN_SetSpritePosition(0, player_x, player_y)

/* setup animations */
seq_walking = TLN_CreateSpriteSequence(NULL, spriteset, "walking", 6)
TLN_SetSpriteAnimation(0, seq_walking, 0)

/* main loop */
TLN_CreateWindow(NULL, 0)
while TLN_ProcessWindow()
        player_x += 1
        if player_x >= WIDTH
                player_x = -16
        ok
        TLN_SetSpritePosition(0, player_x, player_y)
        TLN_DrawFrame(0)
end

/* deinit */
TLN_DeleteTilemap(foreground)
TLN_DeleteTilemap(background)
TLN_Deinit()
```

Screen Shot:

Example (5):

```
load "tilengine.ring"

HRES = 424
VRES = 240

LAYER_PROPS_FRONT        = 0
LAYER_FOREGROUND         = 1
LAYER_PROPS              = 2
LAYER_MIDDLEGROUND       = 3
LAYER_BACKGROUND         = 4
NUM_LAYERS               = 5

xworld = 0
oldx = -1

TLN_Init(HRES, VRES, NUM_LAYERS, 8, 0)

/* load assets */
TLN_SetLogLevel(TLN_LOG_ERRORS)
TLN_SetLoadPath("assets/forest")
foreground = TLN_LoadTilemap("map.tmx", "Main Layer")
middleground = TLN_LoadBitmap("middleground.png")
background = TLN_LoadBitmap("background.png")
atlas = TLN_LoadSpriteset("atlas.png")
props_list = TLN_LoadObjectList("map.tmx", NULL)

/* setup layers */
```

(continues on next page)

(continued from previous page)

```
TLN_SetLayer(LAYER_FOREGROUND, NULL, foreground)
TLN_SetLayerBitmap(LAYER_MIDDLEGROUND, middleground)
TLN_SetLayerBitmap(LAYER_BACKGROUND, background)
width = TLN_GetLayerWidth(LAYER_FOREGROUND)
height = TLN_GetLayerHeight(LAYER_FOREGROUND)

/* objects layer: add back objects (behind sprites) */
TLN_SetLayerObjects(LAYER_PROPS, props_list, NULL)

/* sync props layer positions to main layer */
TLN_SetLayerParent(LAYER_PROPS_FRONT, LAYER_FOREGROUND)
TLN_SetLayerParent(LAYER_PROPS, LAYER_FOREGROUND)

/* create sprite sequences */
idle = TLN_CreateSpriteSequence(NULL, atlas, "player-idle/player-idle-", 6)
skip = TLN_CreateSpriteSequence(NULL, atlas, "player-skip/player-skip-", 6)

/* setup main player sprite */
xplayer = 48
yplayer = 144
TLN_ConfigSprite(0, atlas, 0)
TLN_SetSpriteAnimation(0, idle, 0)

/* create window & main loop */
TLN_CreateWindow(NULL, 0)
TLN_DisableCRTEffect()
while TLN_ProcessWindow()
        TLN_DrawFrame(0)

        /* move 3 pixels right/left main layer */
        if (TLN_GetInput(INPUT_LEFT) && xworld > 0)
                xworld -= 3
        elseif (TLN_GetInput(INPUT_RIGHT) && xworld < width - HRES)
                xworld += 3
        ok

        /* update on change */
        if xworld != oldx
                TLN_SetLayerPosition(LAYER_FOREGROUND, xworld, 32)
                TLN_SetLayerPosition(LAYER_MIDDLEGROUND, xworld / 2, 0)
                TLN_SetLayerPosition(LAYER_BACKGROUND, xworld / 3, 0)
                TLN_SetSpritePosition(0, xplayer - xworld, yplayer)
                oldx = xworld
        ok
end

/* release resources */
TLN_DeleteTilemap(foreground)
TLN_DeleteBitmap(middleground)
TLN_CloseResourcePack()
TLN_DeleteSpriteset(atlas)
TLN_DeleteObjectList(props_list)
```

(continues on next page)

(continued from previous page)

```
TLN_DeleteWindow()
TLN_Deinit()
```

Screen Shot:

Example (6):

```
load "tilengine.ring"
load "sin.ring"

WIDTH            = 400
HEIGHT           = 240
FIXED_BITS       = 16
LAYER_FOREGROUND = 0
LAYER_BACKGROUND = 1
MAX_LAYER        = 2
MAP_HORIZON      = 0
MAP_TRACK        = 1
MAX_MAP          = 2

road             = NULL
x                = NULL
y                = NULL
angle            = NULL

func main

        /* setup engine */
```

(continues on next page)

(continued from previous page)

```
                TLN_Init (WIDTH,HEIGHT, MAX_LAYER, 0, 0)
                TLN_SetRasterCallback ("raster_callback()")
                TLN_SetBGColor (0,0,0)

        /* load resources*/

                TLN_SetLoadPath ("assets/smk")
                road = TLN_LoadTilemap ("track1.tmx", NULL)
                horizon = TLN_LoadTilemap ("track1_bg.tmx", NULL)

        /* startup display */

                TLN_CreateWindow (NULL, 0)
                x = int2fix(-136)
                y = int2fix(336)
                s = 0
                a = float2fix(0.2)
                angle = 0
                BuildSinTable ()

        /* main loop */

                while (TLN_ProcessWindow ())
                        TLN_SetLayerTilemap (LAYER_FOREGROUND, horizon)
                        TLN_SetLayerTilemap (LAYER_BACKGROUND, horizon)
                        TLN_SetLayerPosition (LAYER_FOREGROUND, lerp(angle*2, 0,360, 0,
↪256), 24)
                        TLN_SetLayerPosition (LAYER_BACKGROUND, lerp(angle, 0,360, 0,
↪256), 0)
                        TLN_ResetLayerMode (LAYER_BACKGROUND)

                        /* input */

                                if (TLN_GetInput (INPUT_LEFT))
                                        angle-=2
                                elseif (TLN_GetInput (INPUT_RIGHT))
                                        angle+=2
                                ok

                                if (TLN_GetInput (INPUT_UP))
                                        s += a
                                        if (s > int2fix(2))
                                                s = int2fix(2)
                                        ok
                                elseif (s >= a)
                                        s -= a
                                ok

                                if (TLN_GetInput (INPUT_DOWN))
                                        s -= a
                                        if (s < -int2fix(2))
                                                s = -int2fix(2)
```

(continues on next page)

(continued from previous page)

```
                                        ok
                                elseif (s <= -a)
                                        s += a
                                ok

                                if (s != 0)
                                        angle = angle%360
                                        if (angle < 0)
                                                angle += 360
                                        ok

                                        x += CalcSin (angle, s)
                                        y -= CalcCos (angle, s)
                                ok

                        /* render to window */

                                TLN_DrawFrame (0)
        end

        /* deinit */

                TLN_DeleteTilemap (road)
                TLN_DeleteTilemap (horizon)
                TLN_DeleteWindow ()
                TLN_Deinit ()


/* raster callback (virtual HBLANK) */

func raster_callback

        line = TLN_GETSCANLINE()

        if (line = 24)
                TLN_SetLayerTilemap (LAYER_BACKGROUND, road)
                TLN_SetLayerPosition (LAYER_BACKGROUND, fix2int(x), fix2int(y))
                TLN_DisableLayer (LAYER_FOREGROUND)
        elseif (line >= 24)
                s0 = float2fix (0.2)
                s1 = float2fix (5.0)
                s = lerp (line, 24,HEIGHT, s0,s1)
                scale = fix2float (s)
                TLN_SetLayerTransform (LAYER_BACKGROUND, angle, WIDTH/2, HEIGHT, scale,␣
↪scale)
        ok

func lerp x,x0,x1,fx0,fx1
        return (fx0) + ((fx1) - (fx0)) * ((x) - (x0))/((x1) - (x0))

func float2fix f        return f * (1 << FIXED_BITS)
func int2fix i          return i << FIXED_BITS
```

(continues on next page)

(continued from previous page)

```
func fix2int f          return f >> FIXED_BITS
func fix2float f        return f/(1 << FIXED_BITS)
```

Screen Shot:

Example (7):

```
load "tilengine.ring"
load "sin.ring"

WIDTH   =       320
HEIGHT  =       192
COLUMNS = WIDTH/8 + 2

/* layers */
LAYER_FOREGROUND = 0
LAYER_BACKGROUND = 1
MAX_LAYER = 2

frame = 0
column = list(COLUMNS)

/* entry point */
func main

        /* setup engine */

                TLN_Init (WIDTH,HEIGHT, MAX_LAYER, 0, 5)
```

(continues on next page)

(continued from previous page)

```
                TLN_SetRasterCallback ("raster_callback()")
                TLN_SetBGColor (0,0,0)

        /* load resources*/

                TLN_SetLoadPath ("assets/tf3")
                foreground = TLN_LoadTilemap ("tf3_bg2.tmx", NULL)
                background = TLN_LoadTilemap ("tf3_bg3.tmx", NULL)
                TLN_SetLayerTilemap (LAYER_FOREGROUND, foreground)
                TLN_SetLayerTilemap (LAYER_BACKGROUND, background)

                BuildSinTable ()

                # Convert the Column List to Array of Integers
                        intArray = "" for item in column intArray += int2Bytes(item) next

                TLN_SetLayerColumnOffset (LAYER_BACKGROUND, :intArray)

        /* main loop */

                TLN_CreateWindow (NULL, 0)
                while (TLN_ProcessWindow ())

                        /* scroll */

                                TLN_SetLayerPosition (LAYER_FOREGROUND, frame*3, 0)
                                TLN_SetLayerPosition (LAYER_BACKGROUND, frame, 0)

                        /* update column offset table */

                                for c=1 to COLUMNS
                                        column[c] = CalcSin (frame*5 + c*20, 3)
                                next

                        /* render to window */

                                TLN_DrawFrame (frame)
                                frame++

                end

                /* deinit */

                        TLN_DeleteTilemap (foreground)
                        TLN_DeleteTilemap (background)
                        TLN_Deinit ()

func raster_callback

        /* raster callback (virtual HBLANK) */

                line = TLN_GetScanLine()
```

(continues on next page)

(continued from previous page)

```
                TLN_SetLayerPosition (LAYER_FOREGROUND, frame*2, CalcSin((frame+line)<<1,
↪ 8) + 8)
                TLN_SetLayerPosition (LAYER_BACKGROUND, frame + CalcSin((frame + line)<
↪<1, 10), 0)

func lerp x,x0,x1,fx0,fx1

        /* linear interpolation */

                return (fx0) + ((fx1) - (fx0))*((x) - (x0))/((x1) - (x0))
```

Screen Shot:

Example (8):

```
load "tilengine.ring"

WIDTH  = 400
HEIGHT = 240

sky = [
        [0x1D, 0x44, 0x7B],
        [0x7F, 0xA4, 0xD9],
        [0x0B, 0x00, 0x4E],
        [0xEB, 0x99, 0x9D]
]
```

(continues on next page)

(continued from previous page)

```
sky_hi_r = 0
sky_hi_g = 0
sky_hi_b = 0
sky_lo_r = 0
sky_lo_g = 0
sky_lo_b = 0

frame = 0
xpos = 0
speed = 2
max_xpos = 4720

/* layers */
        LAYER_FOREGROUND = 0
        LAYER_BACKGROUND = 1
        MAX_LAYER = 2


/* entry point */
func main

        /* setup engine */
                TLN_Init (WIDTH, HEIGHT, 2,1,0)
                TLN_SetBGColor (0,128,238)
                TLN_SetRasterCallback ("raster_callback()")

        /* load resources */
                TLN_SetLoadPath ("assets/sotb")
                foreground = TLN_LoadTilemap ("SOTB_fg.tmx", NULL)
                background = TLN_LoadTilemap ("SOTB_bg.tmx", NULL)
                TLN_SetLayerTilemap (LAYER_FOREGROUND, foreground)
                TLN_SetLayerTilemap (LAYER_BACKGROUND, background)

        spriteset = TLN_LoadSpriteset ("SOTB")
        walk = TLN_CreateSpriteSequence (NULL, spriteset, "walk", 6)

        TLN_SetSpriteSet (0, spriteset)
        TLN_SetSpritePosition (0, 200,160)
        TLN_SetSpriteAnimation (0, walk, 0)

        xpos = 2000

        sky_hi_r = sky[1][1]
        sky_hi_g = sky[1][2]
        sky_hi_b = sky[1][3]
        sky_lo_r = sky[2][1]
        sky_lo_g = sky[2][2]
        sky_lo_b = sky[2][3]

        /* main loop */
        TLN_CreateWindow (NULL, 0)
        TLN_DisableCRTEffect()
```

(continues on next page)

(continued from previous page)

```
        while (TLN_ProcessWindow ())
        {
                if (xpos < max_xpos)
                {
                        xpos += speed
                        if (xpos >= max_xpos)
                        {
                                TLN_DisableSpriteAnimation (0)
                                TLN_SetSpritePicture (0, 0)
                        }
                }

                /* sky gradient */
                if (frame>=300 && frame<=900)
                {
                        /* interpolate upper color */
                        sky_hi_r = lerp (frame, 300,900, sky[1][1], sky[3][1])
                        sky_hi_g = lerp (frame, 300,900, sky[1][2], sky[3][2])
                        sky_hi_b = lerp (frame, 300,900, sky[1][3], sky[3][3])

                        /* interpolate lower color */
                        sky_lo_r = lerp (frame, 300,900, sky[2][1], sky[4][1])
                        sky_lo_g = lerp (frame, 300,900, sky[2][2], sky[4][2])
                        sky_lo_b = lerp (frame, 300,900, sky[2][3], sky[4][3])
                }

                TLN_SetLayerPosition (LAYER_FOREGROUND, xpos, 0)

                /* render to the window */
                TLN_DrawFrame (frame)
                frame++
        }

        /* release resources */
        TLN_DeleteSequence(walk)
        TLN_DeleteTilemap (foreground)
        TLN_DeleteTilemap (background)
        TLN_Deinit ()

func raster_callback

        line = TLN_GetScanLine()
        pos = 0

        /* sky color */
        if (line < 192)
        {
                //color = new RGB
                /* interpolate between upper and lower color */
                r = lerp (line, 0,191, sky_hi_r, sky_lo_r)
                g = lerp (line, 0,191, sky_hi_g, sky_lo_g)
                b = lerp (line, 0,191, sky_hi_b, sky_lo_b)
```

(continues on next page)

(continued from previous page)

```
                TLN_SetBGColor (r, g ,b)
        }

        /* background layer */
        pos = -1
        if (line = 0 || line = 24 || line = 64 || line = 88 || line = 96)
                pos = lerp (line, 0,96, xpos*0.7 , xpos*0.2 )
        elseif (line = 120)
                pos = xpos/2
        elseif (line = 208 || line = 216 || line = 224 || line = 232)
                pos = lerp (line,208,232,xpos*1,xpos*2)
        ok

        if (pos != -1)
                TLN_SetLayerPosition (LAYER_BACKGROUND, pos, 0)
        ok

        /* foreground layer */
        pos = -1
        if (line = 0)
                pos = xpos
        elseif (line = 216)
                pos = xpos*3
        ok
        if (pos != -1)
                TLN_SetLayerPosition (LAYER_FOREGROUND, pos, 0)
        ok

func lerp x,x0,x1,fx0,fx1
        /* linear interpolation */
        return (fx0 + (fx1-fx0)*(x-x0)/(x1-x0))
```

Screen Shot:

Example (9):

```
load "tilengine.ring"

WIDTH    = 400
HEIGHT = 240

MIN_SCALE = 50
MAX_SCALE       = 200

/* RGB sky colors */
sky = [
        [0x19, 0x54, 0x75],
        [0x2C, 0xB0, 0xDC]
]

/* layers */
LAYER_FOREGROUND = 0
LAYER_BACKGROUND = 1
MAX_LAYER = 2

xpos=0 ypos=0 scale=0

/* entry point */
func main

        /* setup engine */
        TLN_Init (WIDTH, HEIGHT, MAX_LAYER,0,0)
        TLN_SetBGColor (34,136,170)
```

(continues on next page)

(continued from previous page)

```
        TLN_SetRasterCallback ("raster_callback()")

        /* load resources */
        TLN_SetLoadPath ("assets/fox")
        foreground = TLN_LoadTilemap ("psycho.tmx", NULL)
        background = TLN_LoadTilemap ("rolo.tmx", NULL)
        TLN_SetLayerTilemap (LAYER_FOREGROUND, foreground)
        TLN_SetLayerTilemap (LAYER_BACKGROUND, background)

        /* initial values */
        xpos = 0
        ypos = 192
        scale = 100

        /* main loop */
        TLN_CreateWindow (NULL, 0)
        TLN_DisableCRTEffect()
        while TLN_ProcessWindow ()
                /* user input */
                if TLN_GetInput (INPUT_LEFT)
                        xpos-- ok
                if TLN_GetInput (INPUT_RIGHT)
                        xpos++ ok
                if TLN_GetInput (INPUT_UP) && ypos > 0
                        ypos-- ok
                if TLN_GetInput (INPUT_DOWN)
                        ypos++ ok
                if TLN_GetInput (INPUT_A) && scale < MAX_SCALE
                        scale += 1 ok
                if TLN_GetInput (INPUT_B) && scale > MIN_SCALE
                        scale -= 1 ok

                /* calculate scale factor from fixed point base */
                fgscale = scale/100.0f
                bgscale = lerp(scale, MIN_SCALE,MAX_SCALE, 0.75,1.5)

                /* scale dependant lower clipping */
                maxy = 640 - (240*100/scale)
                if ypos > maxy
                        ypos = maxy
                ok

                /* update position */
                bgypos = lerp(scale,MIN_SCALE,MAX_SCALE, 0,80)
                TLN_SetLayerPosition (LAYER_FOREGROUND, xpos*2, ypos)
                TLN_SetLayerPosition (LAYER_BACKGROUND, xpos, bgypos)
                TLN_SetLayerScaling (LAYER_FOREGROUND, fgscale, fgscale)
                TLN_SetLayerScaling (LAYER_BACKGROUND, bgscale, bgscale)

                /* render to the window */
                TLN_DrawFrame (0)
        end
```

(continues on next page)

(continued from previous page)

```
        /* release resources */
        TLN_DeleteTilemap (foreground)
        TLN_DeleteTilemap (background)
        TLN_Deinit ()


/* sky color gradient with raster effect */
func raster_callback
        line = TLN_GetScanLine()
        if line <= 152
                r = lerp (line, 0,152, sky[1][1], sky[2][1])
                g = lerp (line, 0,152, sky[1][2], sky[2][2])
                b = lerp (line, 0,152, sky[1][3], sky[2][3])
                TLN_SetBGColor (r, g, b)
        ok

/* linear interpolation */
func lerp x,x0,x1,fx0,fx1
        return (fx0) + ((fx1) - (fx0))*((x) - (x0))/((x1) - (x0))
```

Screen Shot:

Example (10):

```
load "tilengine.ring"

WIDTH    = 400
HEIGHT = 240
```

(continues on next page)

(continued from previous page)

```
sky = [
        [0x1B, 0x00, 0x8B],
        [0x00, 0x74, 0xD7],
        [0x24, 0x92, 0xDB],
        [0x1F, 0x7F, 0xBE]
]

/* layers */
LAYER_FOREGROUND = 0
LAYER_BACKGROUND = 1
MAX_LAYER = 2

pos_foreground = 0
pos_background = list(6)
inc_background = list(6)
speed = 0
ypos = 0

color = list(3)

/* entry point */
func main

        /* setup engine */
        TLN_Init (WIDTH,HEIGHT, MAX_LAYER, 0, 1)
        TLN_SetRasterCallback ("raster_callback()")
        TLN_SetBGColor (0,128,238)

        /* load resources*/
        TLN_SetLoadPath ("assets/sonic")
        foreground = TLN_LoadTilemap ("Sonic_md_fg1.tmx", NULL)
        background = TLN_LoadTilemap ("Sonic_md_bg1.tmx", NULL)
        TLN_SetLayerTilemap (LAYER_FOREGROUND, foreground)
        TLN_SetLayerTilemap (LAYER_BACKGROUND, background)
        sp = TLN_LoadSequencePack ("Sonic_md_seq.sqx")
        sequence = TLN_FindSequence (sp, "seq_water")

        /* assign color sequence to various entries in palette */
        palette = TLN_GetLayerPalette (LAYER_BACKGROUND)
        TLN_SetPaletteAnimation (TLN_GetAvailableAnimation(), palette, sequence, true)

        /* compute increments for variable background scrolling speeds */
        inc_background[1] = 0.562f
        inc_background[2] = 0.437f
        inc_background[3] = 0.375f
        inc_background[4] = 0.625f
        inc_background[5] = 1.0f
        inc_background[6] = 2.0f

        /* startup display */
        TLN_CreateWindow (NULL, 0)
```

(continues on next page)

(continued from previous page)

```
        /* main loop */
        while TLN_ProcessWindow ()
                if TLN_GetInput (INPUT_RIGHT)
                        speed += 0.02
                        if speed > 1.0
                                speed = 1.0f
                        ok
                elseif speed > 0.0
                        speed -= 0.02
                        if speed < 0.0
                                speed = 0.0f
                        ok
                ok

                if TLN_GetInput (INPUT_LEFT)
                        speed -= 0.02
                        if speed < -1
                                speed = -1
                        ok
                elseif speed < 0
                        speed += 0.02
                        if speed > 0
                                speed = 0
                        ok
                ok

                /* scroll */
                pos_foreground += 3*speed
                TLN_SetLayerPosition (LAYER_FOREGROUND, pos_foreground, ypos)
                for c=1 to 6
                        pos_background[c] += (inc_background[c] * speed)
                next

                /* render to window */
                TLN_DrawFrame (0)
        end

        /* deinit */
        TLN_DeleteTilemap (foreground)
        TLN_DeleteTilemap (background)
        TLN_DeleteSequencePack (sp)
        TLN_Deinit ()

/* raster callback (virtual HBLANK) */
func raster_callback

        line = TLN_GetScanLine()
        pos =- 1

        if line=0
                pos = pos_background[1]
```

(continues on next page)

(continued from previous page)

```
        elseif line=32
                pos = pos_background[2]
        elseif line=48
                pos = pos_background[3]
        elseif line=64
                pos = pos_background[4]
        elseif line=112
                pos = pos_background[5]
        elseif line >= 152
                pos = lerp (line, 152,224, pos_background[5], pos_background[6])
        ok

        if pos != -1
                TLN_SetLayerPosition (LAYER_BACKGROUND, pos, 0)
        ok

        /* background color gradients */
        if line < 112
                InterpolateColor (line, 0,112, sky[1], sky[2], color)
                TLN_SetBGColor (color[1], color[2], color[3])
        elseif line >= 144
                InterpolateColor (line, 144,HEIGHT, sky[3], sky[4], color)
                TLN_SetBGColor (color[1], color[2], color[3])
        ok

func InterpolateColor v,v1,v2,color1,color2,result
        result[1] = lerp (v, v1,v2, color1[1], color2[1])
        result[2] = lerp (v, v1,v2, color1[2], color2[2])
        result[3] = lerp (v, v1,v2, color1[3], color2[3])

/* linear interpolation */
func lerp x,x0,x1,fx0,fx1
        return (fx0) + ((fx1) - (fx0))*((x) - (x0))/((x1) - (x0))
```

Screen Shot:

61.3 Using LibSDL and Tilengine

Example (11):

```
load "tilengine.ring"
load "libsdl.ring"

width  = 640
height = 480

func main

        # Start Tilengine
                TLN_Init(width, height, 2, 80, 0)

        # Start LibSDL
                SDL_Init(SDL_INIT_EVERYTHING)
                win = SDL_CreateWindow("Using LibSDL and Tilengine", 100, 100, width,␣
↪height, SDL_WINDOW_SHOWN|SDL_WINDOW_OPENGL)
                ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_
↪RENDERER_PRESENTVSYNC )

        # Create the Surface in LibSDL that will be used by Tilengine for Drawing
                surface = SDL_CreateRGBSurface(0, width, height, 32, 0,0,0,0)
        # Get the Surface Pixels pointer from the Surface Structure
                pixels = SDL_Get_SDL_Surface_Pixels(surface)
        # Pass the Pixels pointer to Tilengine
```

(continues on next page)

(continued from previous page)

```
                TLN_SetRenderTarget(pixels, width * 4)

        # Using Tilengine
                TLN_SetLoadPath("assets\sonic")
                foreground = TLN_LoadTilemap ("Sonic_md_fg1.tmx", NULL)
                TLN_SetLayerTilemap(0, foreground)

        # Using LibSDL Events Loop
                oEvent = SDL_New_SDL_Event()

                while true

                        # Using Tilengine to Update the Frame
                                        TLN_UpdateFrame(0)

                        # Draw Tilengine results using LibSDL
                        # ( Surface --> Texture --> Render)
                                tex = SDL_CreateTextureFromSurface(ren, surface);
                                SDL_RenderCopy2(ren,tex)
                                SDL_DestroyTexture(tex)

                        # Draw Rectangle using LibSDL
                                rect = SDL_New_SDL_Rect()
                                SDL_Set_SDL_Rect_x(rect,10)
                                SDL_Set_SDL_Rect_y(rect,10)
                                SDL_Set_SDL_Rect_w(rect,100)
                                SDL_Set_SDL_Rect_h(rect,100)
                                SDL_SetRenderDrawColor(ren,255,255,255,255)
                                SDL_RenderDrawRect(ren,rect)
                                SDL_Destroy_SDL_Rect(rect)

                        # Display Results using LibSDL
                                SDL_RenderPresent(ren)

                        # Check Events

                                        SDL_PollEvent(oEvent)
                                        switch SDL_Get_SDL_Event_Type(oEvent)
                                                        on SDL_Get_SDL_Quit()
                                                                        exit
                                                        on SDL_Get_SDL_Keydown()
                                                                        Key = SDL_Get_
↪SDL_Event_key_Keysym_Sym(oEvent)
                                                                        if key = 27 exit␣
↪ok
                                        off
                end

        # End of Tilengine Usage
                TLN_Deinit()

        # End of LibSDL Usage
```

(continues on next page)

(continued from previous page)

```
                SDL_DestroyWindow(win)
                SDL_Quit()
```

Screen Shot:

Example (12):

```
load "tilengine.ring"
load "libsdl.ring"

factor = 3
width  = 400*factor
height = 220*factor

func main

        # Start Tilengine
                TLN_Init(width, height, 2, 80, 0)

        # Start LibSDL
                SDL_Init(SDL_INIT_EVERYTHING)
```

(continues on next page)

(continued from previous page)

```
                win = SDL_CreateWindow("Using LibSDL and Tilengine", 40, 40, width,␣
↪height, SDL_WINDOW_SHOWN|SDL_WINDOW_OPENGL)
                ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_
↪RENDERER_PRESENTVSYNC )

        # Create the Surface in LibSDL that will be used by Tilengine for Drawing
                surface = SDL_CreateRGBSurface(0, width, height, 32, 0,0,0,0)
        # Get the Surface Pixels pointer from the Surface Structure
                pixels = SDL_Get_SDL_Surface_Pixels(surface)
        # Pass the Pixels pointer to Tilengine
                TLN_SetRenderTarget(pixels, width * 4)

        # Using Tilengine
                TLN_SetLoadPath("assets\sonic")
                foreground = TLN_LoadTilemap ("Sonic_md_fg1.tmx", NULL)
                TLN_SetLayerTilemap(0, foreground)

        # Using LibSDL Events Loop
                oEvent = SDL_New_SDL_Event()

                while true

                        # Using Tilengine to Update the Frame
                                        TLN_UpdateFrame(0)

                        # Draw Tilengine results using LibSDL
                        # ( Surface --> Texture --> Render)
                                tex = SDL_CreateTextureFromSurface(ren, surface);

                                rectSource = SDL_New_SDL_Rect()
                                SDL_Set_SDL_Rect_x(rectSource,0)
                                SDL_Set_SDL_Rect_y(rectSource,0)
                                SDL_Set_SDL_Rect_w(rectSource,400)
                                SDL_Set_SDL_Rect_h(rectSource,220)

                                rectDest = SDL_New_SDL_Rect()
                                SDL_Set_SDL_Rect_x(rectDest,0)
                                SDL_Set_SDL_Rect_y(rectDest,0)
                                SDL_Set_SDL_Rect_w(rectDest,width)
                                SDL_Set_SDL_Rect_h(rectDest,height)

                                SDL_RenderCopy(ren,tex,rectSource,rectDest)
                                SDL_Destroy_SDL_Rect(rectSource)
                                SDL_Destroy_SDL_Rect(rectDest)
                                SDL_DestroyTexture(tex)

                        # Draw Rectangle using LibSDL
                                rect = SDL_New_SDL_Rect()
                                SDL_Set_SDL_Rect_x(rect,40)
                                SDL_Set_SDL_Rect_y(rect,5)
                                SDL_Set_SDL_Rect_w(rect,230)
                                SDL_Set_SDL_Rect_h(rect,200)
```

(continues on next page)

(continued from previous page)

```
                                SDL_SetRenderDrawColor(ren,255,255,255,255)
                                SDL_RenderDrawRect(ren,rect)
                                SDL_Destroy_SDL_Rect(rect)

                        # Display Results using LibSDL
                                SDL_RenderPresent(ren)

                        # Check Events

                                        SDL_PollEvent(oEvent)
                                        switch SDL_Get_SDL_Event_Type(oEvent)
                                                        on SDL_Get_SDL_Quit()
                                                                        exit
                                                        on SDL_Get_SDL_Keydown()
                                                                        Key = SDL_Get_
↪SDL_Event_key_Keysym_Sym(oEvent)
                                                                        if key = 27 exit␣
↪ok
                                        off
                end

        # End of Tilengine Usage
                TLN_Deinit()

        # End of LibSDL Usage
                SDL_DestroyWindow(win)
                SDL_Quit()
```

Screen Shot:

61.4 Reference

Constants:

- TILENGINE_VER_MAJ
- TILENGINE_VER_MIN
- TILENGINE_VER_REV
- TILENGINE_HEADER_VERSION
- FLAG_NONE
- FLAG_FLIPX
- FLAG_FLIPY
- FLAG_ROTATE
- FLAG_PRIORITY
- FLAG_MASKED
- BLEND_NONE
- BLEND_MIX25
- BLEND_MIX50
- BLEND_MIX75
- BLEND_ADD
- BLEND_SUB
- BLEND_MOD
- BLEND_CUSTOM
- MAX_BLEND
- BLEND_MIX
- TLN_OVERLAY_NONE
- TLN_OVERLAY_SHADOWMASK
- TLN_OVERLAY_APERTURE
- TLN_OVERLAY_SCANLINES
- TLN_OVERLAY_CUSTOM
- TLN_MAX_OVERLAY
- PLAYER1
- PLAYER2
- PLAYER3
- PLAYER4
- INPUT_NONE
- INPUT_UP
- INPUT_DOWN

- INPUT_LEFT
- INPUT_RIGHT
- INPUT_BUTTON1
- INPUT_BUTTON2
- INPUT_BUTTON3
- INPUT_BUTTON4
- INPUT_BUTTON5
- INPUT_BUTTON6
- INPUT_START
- INPUT_QUIT
- INPUT_CRT
- INPUT_P1
- INPUT_P2
- INPUT_P3
- INPUT_P4
- INPUT_A
- INPUT_B
- INPUT_C
- INPUT_D
- INPUT_E
- INPUT_F
- CWF_FULLSCREEN
- CWF_VSYNC
- CWF_S1
- CWF_S2
- CWF_S3
- CWF_S4
- CWF_S5
- TLN_ERR_OK
- TLN_ERR_OUT_OF_MEMORY
- TLN_ERR_IDX_LAYER
- TLN_ERR_IDX_SPRITE
- TLN_ERR_IDX_ANIMATION
- TLN_ERR_IDX_PICTURE
- TLN_ERR_REF_TILESET
- TLN_ERR_REF_TILEMAP

- TLN_ERR_REF_SPRITESET
- TLN_ERR_REF_PALETTE
- TLN_ERR_REF_SEQUENCE
- TLN_ERR_REF_SEQPACK
- TLN_ERR_REF_BITMAP
- TLN_ERR_NULL_POINTER
- TLN_ERR_FILE_NOT_FOUND
- TLN_ERR_WRONG_FORMAT
- TLN_ERR_WRONG_SIZE
- TLN_ERR_UNSUPPORTED
- TLN_ERR_REF_LIST
- TLN_MAX_ERR
- TLN_LOG_NONE
- TLN_LOG_ERRORS
- TLN_LOG_VERBOSE

Functions:

- TLN_Engine TLN_Init(int hres, int vres, int numlayers, int numsprites, int numanimations)
- void TLN_Deinit(void)
- bool TLN_DeleteContext(TLN_Engine context)
- bool TLN_SetContext(TLN_Engine context)
- TLN_Engine TLN_GetContext(void)
- int TLN_GetWidth(void)
- int TLN_GetHeight(void)
- uint32_t TLN_GetNumObjects(void)
- uint32_t TLN_GetUsedMemory(void)
- uint32_t TLN_GetVersion(void)
- int TLN_GetNumLayers(void)
- int TLN_GetNumSprites(void)
- void TLN_SetBGColor(uint8_t r, uint8_t g, uint8_t b)
- bool TLN_SetBGColorFromTilemap(TLN_Tilemap tilemap)
- void TLN_DisableBGColor(void)
- bool TLN_SetBGBitmap(TLN_Bitmap bitmap)
- bool TLN_SetBGPalette(TLN_Palette palette)
- void TLN_SetRenderTarget(uint8_t* data, int pitch)
- void TLN_UpdateFrame(int frame)
- void TLN_SetLoadPath(const char * path)

- void TLN_SetCustomBlendFunction(TLN_BlendFunction)
- void TLN_SetLogLevel(TLN_LogLevel log_level)
- bool TLN_OpenResourcePack(const char * filename, const char * key)
- void TLN_CloseResourcePack(void)
- void TLN_SetLastError(TLN_Error error)
- TLN_Error TLN_GetLastError(void)
- const char *TLN_GetErrorString(TLN_Error error)
- bool TLN_CreateWindow(const char * overlay, int flags)
- bool TLN_CreateWindowThread(const char * overlay, int flags)
- void TLN_SetWindowTitle(const char * title)
- bool TLN_ProcessWindow(void)
- bool TLN_IsWindowActive(void)
- bool TLN_GetInput(TLN_Input id)
- void TLN_EnableInput(TLN_Player player, bool enable)
- void TLN_AssignInputJoystick(TLN_Player player, int index)
- void TLN_DefineInputKey(TLN_Player player, TLN_Input input, uint32_t keycode)
- void TLN_DefineInputButton(TLN_Player player, TLN_Input input, uint8_t joybutton)
- void TLN_DrawFrame(int frame)
- void TLN_WaitRedraw(void)
- void TLN_DeleteWindow(void)
- void TLN_EnableBlur(bool mode)
- void TLN_EnableCRTEffect(TLN_Overlay overlay, uint8_t overlay_factor, uint8_t threshold, uint8_t v0, uint8_t v1, uint8_t v2, uint8_t v3, bool blur, uint8_t glow_factor)
- void TLN_DisableCRTEffect(void)
- void TLN_Delay(uint32_t msecs)
- uint32_t TLN_GetTicks(void)
- int TLN_GetWindowWidth(void)
- int TLN_GetWindowHeight(void)
- TLN_Spriteset TLN_CreateSpriteset(TLN_Bitmap bitmap, TLN_SpriteData* data, int num_entries)
- TLN_Spriteset TLN_LoadSpriteset(const char * name)
- TLN_Spriteset TLN_CloneSpriteset(TLN_Spriteset src)
- bool TLN_GetSpriteInfo(TLN_Spriteset spriteset, int entry, TLN_SpriteInfo* info)
- TLN_Palette TLN_GetSpritesetPalette(TLN_Spriteset spriteset)
- int TLN_FindSpritesetSprite(TLN_Spriteset spriteset, const char * name)
- bool TLN_SetSpritesetData(TLN_Spriteset spriteset, int entry, TLN_SpriteData* data, void* pixels, int pitch)
- bool TLN_DeleteSpriteset(TLN_Spriteset Spriteset)

- TLN_Tileset TLN_CreateTileset(int numtiles, int width, int height, TLN_Palette palette, TLN_SequencePack sp, TLN_TileAttributes* attributes)
- TLN_Tileset TLN_CreateImageTileset(int numtiles, TLN_TileImage* images)
- TLN_Tileset TLN_LoadTileset(const char * filename)
- TLN_Tileset TLN_CloneTileset(TLN_Tileset src)
- bool TLN_SetTilesetPixels(TLN_Tileset tileset, int entry, uint8_t* srcdata, int srcpitch)
- int TLN_GetTileWidth(TLN_Tileset tileset)
- int TLN_GetTileHeight(TLN_Tileset tileset)
- int TLN_GetTilesetNumTiles(TLN_Tileset tileset)
- TLN_Palette TLN_GetTilesetPalette(TLN_Tileset tileset)
- TLN_SequencePack TLN_GetTilesetSequencePack(TLN_Tileset tileset)
- bool TLN_DeleteTileset(TLN_Tileset tileset)
- TLN_Tilemap TLN_CreateTilemap(int rows, int cols, TLN_Tile tiles, uint32_t bgcolor, TLN_Tileset tileset)
- TLN_Tilemap TLN_LoadTilemap(const char * filename, const char * layername)
- TLN_Tilemap TLN_CloneTilemap(TLN_Tilemap src)
- int TLN_GetTilemapRows(TLN_Tilemap tilemap)
- int TLN_GetTilemapCols(TLN_Tilemap tilemap)
- TLN_Tileset TLN_GetTilemapTileset(TLN_Tilemap tilemap)
- bool TLN_GetTilemapTile(TLN_Tilemap tilemap, int row, int col, TLN_Tile tile)
- bool TLN_SetTilemapTile(TLN_Tilemap tilemap, int row, int col, TLN_Tile tile)
- bool TLN_CopyTiles(TLN_Tilemap src, int srcrow, int srccol, int rows, int cols, TLN_Tilemap dst, int dstrow, int dstcol)
- bool TLN_DeleteTilemap(TLN_Tilemap tilemap)
- TLN_Palette TLN_CreatePalette(int entries)
- TLN_Palette TLN_LoadPalette(const char * filename)
- TLN_Palette TLN_ClonePalette(TLN_Palette src)
- bool TLN_SetPaletteColor(TLN_Palette palette, int color, uint8_t r, uint8_t g, uint8_t b)
- bool TLN_MixPalettes(TLN_Palette src1, TLN_Palette src2, TLN_Palette dst, uint8_t factor)
- bool TLN_AddPaletteColor(TLN_Palette palette, uint8_t r, uint8_t g, uint8_t b, uint8_t start, uint8_t num)
- bool TLN_SubPaletteColor(TLN_Palette palette, uint8_t r, uint8_t g, uint8_t b, uint8_t start, uint8_t num)
- bool TLN_ModPaletteColor(TLN_Palette palette, uint8_t r, uint8_t g, uint8_t b, uint8_t start, uint8_t num)
- uint8_t* TLN_GetPaletteData(TLN_Palette palette, int index)
- bool TLN_DeletePalette(TLN_Palette palette)
- TLN_Bitmap TLN_CreateBitmap(int width, int height, int bpp)
- TLN_Bitmap TLN_LoadBitmap(const char * filename)
- TLN_Bitmap TLN_CloneBitmap(TLN_Bitmap src)
- uint8_t* TLN_GetBitmapPtr(TLN_Bitmap bitmap, int x, int y)

- int TLN_GetBitmapWidth(TLN_Bitmap bitmap)
- int TLN_GetBitmapHeight(TLN_Bitmap bitmap)
- int TLN_GetBitmapDepth(TLN_Bitmap bitmap)
- int TLN_GetBitmapPitch(TLN_Bitmap bitmap)
- TLN_Palette TLN_GetBitmapPalette(TLN_Bitmap bitmap)
- bool TLN_SetBitmapPalette(TLN_Bitmap bitmap, TLN_Palette palette)
- bool TLN_DeleteBitmap(TLN_Bitmap bitmap)
- TLN_ObjectList TLN_CreateObjectList(void)
- bool TLN_AddTileObjectToList(TLN_ObjectList list, uint16_t id, uint16_t gid, uint16_t flags, int x, int y)
- TLN_ObjectList TLN_LoadObjectList(const char * filename, const char * layername)
- TLN_ObjectList TLN_CloneObjectList(TLN_ObjectList src)
- int TLN_GetListNumObjects(TLN_ObjectList list)
- bool TLN_GetListObject(TLN_ObjectList list, TLN_ObjectInfo* info)
- bool TLN_DeleteObjectList(TLN_ObjectList list)
- bool TLN_SetLayer(int nlayer, TLN_Tileset tileset, TLN_Tilemap tilemap)
- bool TLN_SetLayerTilemap(int nlayer, TLN_Tilemap tilemap)
- bool TLN_SetLayerBitmap(int nlayer, TLN_Bitmap bitmap)
- bool TLN_SetLayerPalette(int nlayer, TLN_Palette palette)
- bool TLN_SetLayerPosition(int nlayer, int hstart, int vstart)
- bool TLN_SetLayerScaling(int nlayer, float xfactor, float yfactor)
- bool TLN_SetLayerAffineTransform(int nlayer, TLN_Affine *affine)
- bool TLN_SetLayerTransform(int layer, float angle, float dx, float dy, float sx, float sy)
- bool TLN_SetLayerPixelMapping(int nlayer, TLN_PixelMap* table)
- bool TLN_SetLayerBlendMode(int nlayer, TLN_Blend mode, uint8_t factor)
- bool TLN_SetLayerColumnOffset(int nlayer, int* offset)
- bool TLN_SetLayerClip(int nlayer, int x1, int y1, int x2, int y2)
- bool TLN_DisableLayerClip(int nlayer)
- bool TLN_SetLayerMosaic(int nlayer, int width, int height)
- bool TLN_DisableLayerMosaic(int nlayer)
- bool TLN_ResetLayerMode(int nlayer)
- bool TLN_SetLayerObjects(int nlayer, TLN_ObjectList objects, TLN_Tileset tileset)
- bool TLN_SetLayerPriority(int nlayer, bool enable)
- bool TLN_SetLayerParent(int nlayer, int parent)
- bool TLN_DisableLayerParent(int nlayer)
- bool TLN_DisableLayer(int nlayer)
- TLN_Palette TLN_GetLayerPalette(int nlayer)

- bool TLN_GetLayerTile(int nlayer, int x, int y, TLN_TileInfo* info)
- int TLN_GetLayerWidth(int nlayer)
- int TLN_GetLayerHeight(int nlayer)
- bool TLN_ConfigSprite(int nsprite, TLN_Spriteset spriteset, uint32_t flags)
- bool TLN_SetSpriteSet(int nsprite, TLN_Spriteset spriteset)
- bool TLN_SetSpriteFlags(int nsprite, uint32_t flags)
- bool TLN_EnableSpriteFlag(int nsprite, uint32_t flag, bool enable)
- bool TLN_SetSpritePosition(int nsprite, int x, int y)
- bool TLN_SetSpritePicture(int nsprite, int entry)
- bool TLN_SetSpritePalette(int nsprite, TLN_Palette palette)
- bool TLN_SetSpriteBlendMode(int nsprite, TLN_Blend mode, uint8_t factor)
- bool TLN_SetSpriteScaling(int nsprite, float sx, float sy)
- bool TLN_ResetSpriteScaling(int nsprite)
- int TLN_GetSpritePicture(int nsprite)
- int TLN_GetAvailableSprite(void)
- bool TLN_EnableSpriteCollision(int nsprite, bool enable)
- bool TLN_GetSpriteCollision(int nsprite)
- bool TLN_GetSpriteState(int nsprite, TLN_SpriteState* state)
- bool TLN_SetFirstSprite(int nsprite)
- bool TLN_SetNextSprite(int nsprite, int next)
- bool TLN_EnableSpriteMasking(int nsprite, bool enable)
- void TLN_SetSpritesMaskRegion(int top_line, int bottom_line)
- bool TLN_SetSpriteAnimation(int nsprite, TLN_Sequence sequence, int loop)
- bool TLN_DisableSpriteAnimation(int nsprite)
- bool TLN_DisableSprite(int nsprite)
- TLN_Palette TLN_GetSpritePalette(int nsprite)
- TLN_Sequence TLN_CreateSequence(const char * name, int target, int num_frames, TLN_SequenceFrame* frames)
- TLN_Sequence TLN_CreateCycle(const char * name, int num_strips, TLN_ColorStrip* strips)
- TLN_Sequence TLN_CreateSpriteSequence(const char * name, TLN_Spriteset spriteset, const char * basename, int delay)
- TLN_Sequence TLN_CloneSequence(TLN_Sequence src)
- bool TLN_GetSequenceInfo(TLN_Sequence sequence, TLN_SequenceInfo* info)
- bool TLN_DeleteSequence(TLN_Sequence sequence)
- TLN_SequencePack TLN_CreateSequencePack(void)
- TLN_SequencePack TLN_LoadSequencePack(const char * filename)
- TLN_Sequence TLN_GetSequence(TLN_SequencePack sp, int index)

- TLN_Sequence TLN_FindSequence(TLN_SequencePack sp, const char * name)
- int TLN_GetSequencePackCount(TLN_SequencePack sp)
- bool TLN_AddSequenceToPack(TLN_SequencePack sp, TLN_Sequence sequence)
- bool TLN_DeleteSequencePack(TLN_SequencePack sp)
- bool TLN_SetPaletteAnimation(int index, TLN_Palette palette, TLN_Sequence sequence, bool blend)
- bool TLN_SetPaletteAnimationSource(int index, TLN_Palette)
- bool TLN_GetAnimationState(int index)
- bool TLN_SetAnimationDelay(int index, int frame, int delay)
- int TLN_GetAvailableAnimation(void)
- bool TLN_DisablePaletteAnimation(int index)
- void TLN_SetRasterCallback(const char *cCode)
- void TLN_SetFrameCallback(const char *cCode)
- void TLN_SetSDLCallback(const char *cCode)
- int TLN_GetScanLine(void)
- SDL_Event *TLN_GetSDLEvent(void)

CHAPTER

SIXTYTWO

PERFORMANCE TIPS

In this chapter we will learn more about the Ring performance.

Tested using Victus Laptop [13th Gen Intel(R) Core(TM) i7-13700H, Windows 11, Ring 1.19]

62.1 Introduction

Ring is designed to be a simple, small and flexible language in the first place, but also it is fast enough for many applications.

Ring can do each of the following tasks in around one second.

(1) Compiling 100,000 lines of code

(2) Executing an empty loop that count from 1 to 100,000,000

(3) Creating list contains 7,000,000 items then summing all of the list items

(4) Printing numbers from 1 to 30,000 using command prompt

(5) Printing numbers from 1 to 400,000 using output redirection and Ring Notepad

(6) Adding 50,000 nodes to the TreeWidget in GUI applications

(7) Adding 60,000 items to the ListWidget in GUI applications

(8) Executing 3000 search operations using linear search in a list contains 100,000 items, trying to find the last item (The worst case)

Also when we need more speed we can use C/C++ extensions!

Example:

```
t1=clock()
for t=1 to 100_000_000 next
? (clock()-t1)/clockspersecond()
```

Output:

```
1.12
```

Example:

```
? "Create list contains 100,000 items"
aList = 1:100000
```

(continues on next page)

(continued from previous page)

```
? "Do 3000 search operations - Find the last item (Worst Case!)"
c = clock()

for t = 1 to 3000
        find(alist,100000)
next

? "Time: " + ( clock() - c ) / clockspersecond() + " seconds"
```

Output:

```
Create list contains 100,000 items
Do 3000 search operations - Find the last item (Worst Case!)
Time: 0.83 seconds
```

Example:

```
load "guilib.ring"

C_NODESCOUNT = 50000

func main
new QApp {
        win = new QWidget() {
                move(100,100) resize(500,500)
                setWindowTitle("Many Tree Items - Testing Performance")
                tree = new QTreeWidget(win) {
                        blocksignals(True) setUpdatesEnabled(False)
                        root = new qTreeWidgetItem()
                        root.setText(0,"The Root Node")
                        t1 = clock()
                        for t = 1 to C_NODESCOUNT
                                oItem = new qTreeWidgetItem()
                                oItem.settext(0,"Item " + t)
                                root.addchild(oItem)
                        next
                        cTime = (clock()-t1)/clockspersecond()
                        setHeaderLabel("Creating "+C_NODESCOUNT+" nodes in " + cTime + "␣
↪seconds.")
                        addTopLevelItem(root)
                        expanditem(root)
                        blocksignals(False) setUpdatesEnabled(True)
                }
                oLayout = new QVBoxLayout() {
                        addWidget(tree)
                }
                setLayout(oLayout)
                show()
        }
        exec()
}
```

Output:

62.2 Creating Lists

Example:

```
C_COUNT = 1_000_000

? "Create the list using the Range operator"
t1 = clock()
aList = 1:C_COUNT
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "Create the list using the For loop"
t1 = clock()
aList = []
for x = 1 to C_COUNT
        aList + x
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "Create the list using the list() function and the For loop"
t1 = clock()
aList = list(C_COUNT)
for x = 1 to C_COUNT
```

(continues on next page)

(continued from previous page)

```
        aList[x] = x
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"
```

Output:

```
Create the list using the Range operator
Time : 0.01 seconds
Create the list using the For loop
Time : 0.11 seconds
Create the list using the list() function and the For loop
Time : 0.13 seconds
```

Note: Creating lists using the Range operator is faster than using the For loop or the list() function

62.3 Arithmetic Operations

Example:

```
C_COUNT = 1_000_000

? "Using * operator"
t1 = clock()
out = 10
for x = 1 to C_COUNT
        out = out * 2
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "Using *= operator"
t1 = clock()
for x = 1 to C_COUNT
        out *= 2
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"
```

Output:

```
Using * operator
Time : 0.11 seconds
Using *= operator
Time : 0.09 seconds
```

Note: Using the *= operator is faster that using the * operator

62.4 Using len() and For Loops

Example:

```
aList = 1:1000000

? "Using len() in the For loop"
t1 = clock()
for x = 1 to len(aList)
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "Using len() before the For loop"
t1 = clock()
nMax = len(aList)
for x = 1 to nMax
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"
```

Output:

```
Using len() in the For loop
Time : 0.32 seconds
Using len() before the For loop
Time : 0.03 seconds
```

Note: Using len() before the For loop is faster than using len() in the For loop.

62.5 Calling Functions and Methods

Example:

```
? "calling 100000 functions"
t1 = clock()
for x = 1 to 100000
        test()
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

o1 = new test

? "calling 100000 methods using the dot operator"
t1 = clock()
for x = 1 to 100000
        o1.test()
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "calling 100000 methods using braces "
```

(continues on next page)

(continued from previous page)

```
t1 = clock()
for x = 1 to 100000
        o1 { test() }
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"


func test

class test
        func test
```

Output:

```
calling 100000 functions
Time : 0.03 seconds
calling 100000 methods using the dot operator
Time : 0.06 seconds
calling 100000 methods using braces
Time : 0.18 seconds
```

Note: Calling functions is faster than calling methods.

Note: Using the dot operator to call methods is faster than using braces.

CHAPTER

SIXTYTHREE

COMMAND LINE OPTIONS

The ring language takes source code file (*.ring) or the object file (*.ringo) as input to execute, also the language provide other options like

Option	Description
-tokens	Print a list of tokens in the source code file
-rules	Print grammar rules applied on the tokens
-ic	Print the intermediate byte code (before execution)
-icfinal	Print the final byte code (after execution)
-cgi	Print http response header before error messages
-norun	Don't run the program after compiling
-ins	Print instruction operation code before execution
-performance	Print clock before and after program execution
-go	Generate Object File
-geo	Generate embedded object file (C source code)
-w	Display Warnings

63.1 Printing Tokens

Example:

```
Func Main
        See "Hello World" + nl
        for x = 1 to 10
                see x + nl
        next
        test()

func test
        see "welcome" + nl
        o1 = new point { x=10 y=20 z=30 }
        see o1

class point x y z
```

Command:

```
ring test.ring -tokens -norun
```

Output:

```
=====================================================
Tokens - Generated by the Scanner
=====================================================

   Keyword : FUNC
Identifier : main
   EndLine
   Keyword : SEE
   Literal : Hello World
  Operator : +
Identifier : nl
   EndLine
   Keyword : FOR
Identifier : x
  Operator : =
    Number : 1
   Keyword : TO
    Number : 10
   EndLine
   Keyword : SEE
Identifier : x
  Operator : +
Identifier : nl
   EndLine
   Keyword : NEXT
   EndLine
Identifier : test
  Operator : (
  Operator : )
   EndLine
   Keyword : FUNC
Identifier : test
   EndLine
   Keyword : SEE
   Literal : welcome
  Operator : +
Identifier : nl
   EndLine
Identifier : o1
  Operator : =
   Keyword : NEW
Identifier : point
  Operator : {
Identifier : x
  Operator : =
    Number : 10
Identifier : y
  Operator : =
    Number : 20
Identifier : z
  Operator : =
```

(continues on next page)

(continued from previous page)

```
    Number : 30
  Operator : }
   EndLine
   Keyword : SEE
Identifier : o1
   EndLine
   Keyword : CLASS
Identifier : point
Identifier : x
Identifier : y
Identifier : z
   EndLine

=====================================================
```

63.2 Printing Rules

Command:

```
ring test.ring -rules -norun
```

Output:

```
=====================================================
Grammar Rules Used by The Parser
=====================================================

Rule : Program --> {Statement}

Line 1
Rule : Statement  --> 'Func' Identifier [ParaList]

Line 2
Rule : Factor --> Literal
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Arithmetic --> Arithmetic + Arithmetic
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> 'See' Expr
```

(continues on next page)

(continued from previous page)

```
Line 3
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> 'For' Identifier '=' Expr to Expr ['step' Expr]

Line 4
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Arithmetic --> Arithmetic + Arithmetic
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> 'See' Expr

Line 5
Rule : Next --> 'Next'

Line 6
Rule : Mixer -> '(' [Expr { ',' Expr} ] ')'

Line 8
```

(continues on next page)

(continued from previous page)

```
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr
Rule : Statement  --> 'Func' Identifier [ParaList]

Line 9
Rule : Factor --> Literal
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : Arithmetic --> Arithmetic + Arithmetic
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> 'See' Expr

Line 10
Rule : Factor --> New Identifier {'.' Identifier }
Rule : Mixer --> '{' {Statement} BraceEnd
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
```

(continues on next page)

(continued from previous page)

```
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr
Rule : Factor --> Number
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
```

(continues on next page)

(continued from previous page)

```
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr
Rule : BraceEnd --> '}'
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Assignment -> '=' Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr

Line 11
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> 'See' Expr

Line 13
Rule : Statement  --> 'Class' Identifier
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
```

(continues on next page)

(continued from previous page)

```
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr
Rule : Factor --> Identifier [ {Mixer} | Assignment | PlusPlus | MinusMinus]
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> Expr

====================================================
```

63.3 Printing Intermediate Code

Command:

```
ring test.ring -ic -norun
```

Output:

```
====================================================
Byte Code - Before Execution by the VM
====================================================

         PC      OPCode        Data

          1  ReturnNull
          2        Func    main
          3     NewLine       2
          4     FuncExE
```

(continues on next page)

(continued from previous page)

```
 5       PushC   Hello World
 6       LoadA      nl      0
 7       PushV
 8         SUM       0
 9       Print
10     NewLine       3
11    ExitMark      29     28
12  LoadAFirst       x
13       PushN   1.000000
14  BeforeEqual       0
15  Assignment
16       PushN   1.000000
17  StepNumber
18  JumpVarLENum       x  10.000000    29
19     NewLine       4
20     FuncExE
21       LoadA       x      0
22       PushV
23       LoadA      nl      0
24       PushV
25         SUM       0
26       Print
27     NewLine       5
28     IncJump       x     18
29  POPExitMark
30     POPStep
31     NewLine       6
32    LoadFunc    test
33        Call       0
34  NoOperation
35     NewLine       8
36       PushV
37   FreeStack
38  ReturnNull
39        Func    test
40     NewLine       9
41     FuncExE
42       PushC   welcome
43       LoadA      nl      0
44       PushV
45         SUM       0
46       Print
47     NewLine      10
48       LoadA      o1      0
49  AssignmentPointer
50         New   point
51    SetScope
52       PushV
53  BraceStart
54       LoadA       x      0     58
55  AssignmentPointer
56       PushN   10.000000
```

(continues on next page)

(continued from previous page)

```
         57  BeforeEqual       0
         58  Assignment       0      0
         59   FreeStack
         60       LoadA       y      0     64
         61  AssignmentPointer
         62       PushN   20.000000
         63  BeforeEqual       0
         64  Assignment       0      0
         65   FreeStack
         66       LoadA       z      0     70
         67  AssignmentPointer
         68       PushN   30.000000
         69  BeforeEqual       0
         70  Assignment       0      0
         71   FreeStack
         72    LoadFunc   ismethod
         73       LoadA    self      0
         74       PushV
         75       PushC   braceend
         76        Call
         77  NoOperation
         78       PushV
         79       JumpZ      85
         80    LoadFunc   braceend
         81        Call
         82  NoOperation
         83       PushV
         84   FreeStack
         85    BraceEnd
         86   FreeStack
         87     NewLine      11
         88     FuncExE
         89       LoadA      o1      0
         90       PushV
         91       Print
         92     NewLine      13
         93  ReturnNull
         94       Class   point  006E8BC0
         95    NewLabel
         96       LoadA       x      0
         97       PushV
         98   FreeStack
         99       LoadA       y      0
        100       PushV
        101   FreeStack
        102       LoadA       z      0
        103       PushV
        104   FreeStack
        105  ReturnNull

==================================================
```

63.4 Printing Final Intermediate Code

Command:

```
ring test.ring -icfinal
```

Output:

```
Hello World
1
2
3
4
5
6
7
8
9
10
welcome
x: 10.000000
y: 20.000000
z: 30.000000


====================================================
Byte Code - After Execution by the VM
====================================================

         PC      OPCode        Data

          1  ReturnNull
          2        Func    main
          3     NewLine       2
          4     FuncExE
          5       PushC   Hello World
          6       PushP   007D3670      0
          7       PushV
          8         SUM       0
          9       Print
         10     NewLine       3
         11    ExitMark      29     28
         12  LoadAFirst       x
         13       PushN   1.000000
         14  BeforeEqual       0
         15  Assignment
         16       PushN   1.000000
         17  StepNumber
         18  JumpVarLPLENum       x  10.000000    29
         19     NewLine       4
         20     FuncExE
         21  PushPLocal       x      0
         22       PushV
```

(continues on next page)

(continued from previous page)

```
      23       PushP   007D3670      0
      24       PushV
      25         SUM       0
      26       Print
      27     NewLine       5
      28   IncLPJump       x     18
      29  POPExitMark
      30     POPStep
      31     NewLine       6
      32   LoadFuncP    test
      33        Call       0
      34  NoOperation
      35     NewLine       8
      36       PushV
      37   FreeStack
      38  ReturnNull
      39        Func    test
      40     NewLine       9
      41     FuncExE
      42       PushC   welcome
      43       PushP   007D3670      0
      44       PushV
      45         SUM       0
      46       Print
      47     NewLine      10
      48  PushPLocal      o1      0
      49  AssignmentPointer
      50         New   point
      51    SetScope
      52       PushV
      53  BraceStart
      54       LoadA       x      0     58
      55  AssignmentPointer
      56       PushN   10.000000
      57  BeforeEqual       0
      58  SetProperty       0    106
      59   FreeStack
      60       LoadA       y      0     64
      61  AssignmentPointer
      62       PushN   20.000000
      63  BeforeEqual       0
      64  SetProperty       0    141
      65   FreeStack
      66       LoadA       z      0     70
      67  AssignmentPointer
      68       PushN   30.000000
      69  BeforeEqual       0
      70  SetProperty       0    176
      71   FreeStack
      72    LoadFunc   ismethod
      73       LoadA    self      0
      74       PushV
```

(continues on next page)

(continued from previous page)

```
 75       PushC   braceend
 76        Call
 77  NoOperation
 78       PushV
 79       JumpZ      85
 80    LoadFunc   braceend
 81        Call
 82  NoOperation
 83       PushV
 84   FreeStack
 85    BraceEnd
 86   FreeStack
 87     NewLine      11
 88     FuncExE
 89  PushPLocal      o1      0
 90       PushV
 91       Print
 92     NewLine      13
 93  ReturnNull
 94       Class   point  007D8470
 95    NewLabel
 96       LoadA       x      0
 97       PushV
 98   FreeStack
 99       LoadA       y      0
100       PushV
101   FreeStack
102       LoadA       z      0
103       PushV
104   FreeStack
105  ReturnNull
106    LoadFunc   ismethod
107       LoadA   ring_gettemp_var      0
108       PushV
109       PushC    setx
110        Call       0
111  NoOperation
112       PushV
113       JumpZ     132
114     NewLine       2
115       LoadA   ring_gettemp_var      0
116  LoadMethod    setx
117       LoadA   ring_settemp_var      0
118       PushV
119        Call       0      1
120  AfterCallMethod
121       PushV
122   FreeStack
123     NewLine       3
124       LoadA   ring_tempflag_var      0    128
125  AssignmentPointer
126       PushN   0.000000
```

(continues on next page)

(continued from previous page)

```
127  BeforeEqual       0
128  Assignment       0      0
129   FreeStack
130     NewLine       4
131        Jump     140
132     NewLine       5
133       PushP   007D37D8      0    137
134  AssignmentPointer
135       PushN   1.000000
136  BeforeEqual       0
137  Assignment       0      0
138   FreeStack
139     NewLine       6
140      Return
141    LoadFunc   ismethod
142       LoadA   ring_gettemp_var      0
143       PushV
144       PushC    sety
145        Call       0
146  NoOperation
147       PushV
148       JumpZ     167
149     NewLine       2
150       LoadA   ring_gettemp_var      0
151  LoadMethod    sety
152       LoadA   ring_settemp_var      0
153       PushV
154        Call       0      1
155  AfterCallMethod
156       PushV
157   FreeStack
158     NewLine       3
159       LoadA   ring_tempflag_var      0    163
160  AssignmentPointer
161       PushN   0.000000
162  BeforeEqual       0
163  Assignment       0      0
164   FreeStack
165     NewLine       4
166        Jump     175
167     NewLine       5
168       PushP   007D37D8      0    172
169  AssignmentPointer
170       PushN   1.000000
171  BeforeEqual       0
172  Assignment       0      0
173   FreeStack
174     NewLine       6
175      Return
176    LoadFunc   ismethod
177       LoadA   ring_gettemp_var      0
178       PushV
```

(continues on next page)

(continued from previous page)

```
        179       PushC    setz
        180        Call       0
        181  NoOperation
        182       PushV
        183       JumpZ     202
        184     NewLine       2
        185       LoadA   ring_gettemp_var      0
        186  LoadMethod    setz
        187       LoadA   ring_settemp_var      0
        188       PushV
        189        Call       0      1
        190  AfterCallMethod
        191       PushV
        192   FreeStack
        193     NewLine       3
        194       LoadA   ring_tempflag_var      0    198
        195  AssignmentPointer
        196       PushN   0.000000
        197  BeforeEqual       0
        198  Assignment       0      0
        199   FreeStack
        200     NewLine       4
        201        Jump     210
        202     NewLine       5
        203       PushP   007D37D8      0    207
        204  AssignmentPointer
        205       PushN   1.000000
        206  BeforeEqual       0
        207  Assignment       0      0
        208   FreeStack
        209     NewLine       6
        210      Return

====================================================
```

63.5 CGI Support

Command:

```
ring test.ring -cgi
```

63.6 No Run

Command:

```
ring test.ring -norun
```

63.7 Printing Instruction Operation Code

Command:

```
ring test.ring -ins
```

Output:

```
==================================================

Operation  : ReturnNull
PC         : 1
Line Number    : 1  , File test.ring

SP (After) : 0  - FuncSP : 0
 LineNumber 1
==================================================
.....
.....
.....
```

Tip: Output removed from the previous example because it's very large!

63.8 Performance

Command:

```
ring test.ring -performance
```

Output:

```
==================================================
Date  : 2015/09/15 Time : 15:56:17
Clock : 0
==================================================
Hello World
1
2
3
4
5
6
```

(continues on next page)

(continued from previous page)

```
7
8
9
10
welcome
x: 10.000000
y: 20.000000
z: 30.000000

====================================================
Date  : 2015/09/15 Time : 15:56:17
Clock : 0
====================================================
```

63.9 Generate Object File

You can generate object file (*.ringo) from your source code file (*.ring) using -go option

Tip: You will get one object file to use for distributing/running your application which may contains one or many ring source files that you can keep or distribute based on the application (commercial or open source).

Command:

```
ring test.ring -go
```

To run the compiled object file

```
ring test.ringo
```

63.10 Generate Embedded Object File

You can generate embedded object file (C source code) from your source code file (*.ring) using -geo option

Command:

```
ring test.ring -geo
```

This command will generate at least three files

```
test.c
ringappcode.c
ringappcode.h
```

More files could be generated based on the project size

The generated files will pass the byte code to Ring VM to be executed

CHAPTER

SIXTYFOUR

DISTRIBUTING RING APPLICATIONS (MANUAL)

In this chapter we will learn about distributing Ring applications.

The next method is old and was used in Ring 1.5 and previous versions!

Starting from Ring 1.6 we have a nice tool called Ring2EXE

Using Ring2EXE we can distribute applications quickly for Windows, Linux and macOS

Check the Ring2EXE chapter for more information!

64.1 Distributing Applications for Microsoft Windows

Step 1:

```
Copy c:\ring\bin folder to be for example c:\myapp
```

Step 2:

```
Rename c:\myapp\ring.exe to c:\myapp\myapp.exe
```

Step 3:

```
Create a file c:\myapp\ring.ring
```

And write

```
Load "myapp.ring"
```

When you run myapp.exe the file ring.ring will be executed automatically

So your file myapp.ring will be called and executed

Or just rename myapp.ring to ring.ring

It's a fast way to distribute applications.

64.2 Protecting the Source Code

Step 1:

Execute the next command

```
ring myapp.ring -go
```

This will generate one object file (myapp.ringo) from the project files (*.ring)

Step 2:

```
Rename myapp.ringo to ring.ringo
```

When you run the executable file (ring.exe) or (myapp.exe) the file ring.ringo will be executed.

64.3 The files ring.ring and ring.ringo

Ring will run ring.ring or ring.ringo only when

(1) The file exist in the current directory

(2) No file is passed to (ring.exe) to execute

We added the support to ring.ring (and ring.ringo) for a way to distribute Ring apps (before Ring2EXE)

64.4 Creating Windows Installer

There are many tools that you can use to distribute your application.

Check : nullsoft scriptable install system

URL : http://nsis.sourceforge.net/Main_Page

64.5 Using C/C++ Compiler and Linker

Another method to distribute applications is to use a C/C++ compiler.

Ring can be embedded in C/C++ projects, We can create executable files using a C/C++ compiler by embedding the Ring language in our project.

Check the "Embedding Ring Language in C/C++ Programs" chapter.

Using this way we will avoid using ring.ring or ring.ringo files.

64.6 Distributing Applications and Games for Mobile

Ring can be embedded in a Qt projects or LibSDL projects to build Mobile applications and Games.

You can build the Qt project or the LibSDL project and get the Android package directly (*.apk)

Check Ring distributions for Mobile development using Qt or LibSDL.

CHAPTER

SIXTYFIVE

DISTRIBUTING RING APPLICATIONS USING RING2EXE

In this chapter we will learn about distributing Ring applications.

Starting from Ring 1.6 we have a nice tool called Ring2EXE (Written in Ring itself)

Using Ring2EXE we can distribute applications quickly for Windows, Linux, macOS, WebAssembly and Mobile devices

Note: We can use the Distribute Menu in the Ring Notepad application (More Easy)

65.1 Using Ring2EXE

```
ring2exe filename.ring [Options]
```

This will set filename.ring as input to the program

The next files will be generated

```
filename.ringo          (The Ring Object File - by Ring Compiler)
filename.c              (The C Source code file
                         Contains the ringo file content
                         Will be generated by this program)
filename_buildvc.bat    (Will be executed to build filename.c using Visual C/C++)
filename_buildgcc.bat   (Will be executed to build filename.c using GNU C/C++)
filename_buildclang.bat (Will be executed to build filename.c using CLang C/C++)
filename.obj            (Will be generated by the Visual C/C++ compiler)
filename.exe            (Will be generated by the Visual C/C++ Linker)
filename                (Executable File - On Linux & MacOS X platforms)
```

65.2 How Ring2EXE works?

At first the Ring compiler will be used to generate the Ring object file (*.ringo)

If we have a C compiler (optional), This object file will be embedded inside a C source code file

Then using the C compiler and the Ring library (Contains the Ring Virtual Machine) the executable file

will be generated!

If we don't have a C compiler, the Ring executable will be copied and renamed to your application name

And your Ring object file (*.ringo) will become ring.ringo to be executed at startup of the executable file.

So it's better and easy to have a C compiler on your machine to be used by Ring2EXE.

65.3 Example

We have test.ring contains the next code

```
see "Hello, World!" + nl
```

To build the executable file for Windows, Linux or macOS

```
ring2exe test.ring
```

To run the program (Windows)

```
test
```

To run the program (Linux and macOS)

```
./test
```

65.4 Options

```
-keep            : Don't delete Temp. Files
-static          : Build Standalone Executable File
                   (Don't use ring.dll/ring.so/ring.dylib)
-gui             : Build GUI Application (Hide the Console Window)
-dist            : Prepare application for distribution
-allruntime      : Include all libraries in distribution
-mobileqt        : Prepare Qt Project to distribute Ring Application for Mobile
-webassemblyqt   : Prepare Qt Project to distribute Ring Application for the web␣
↪(WebAssembly)
-noqt            : Remove RingQt from distribution
-noallegro       : Remove RingAllegro from distribution
-noopenssl       : Remove RingOpenSSL from distribution
-nolibcurl       : Remove RingLibCurl from distribution
-nomysql         : Remove RingMySQL from distribution
-noodbc          : Remove RingODBC from distribution
-nosqlite        : Remove RingSQLite from distribution
-noopengl        : Remove RingOpenGL from distribution
-nofreeglut      : Remove RingFreeGLUT from distribution
-nolibzip        : Remove RingLibZip from distribution
-noconsolecolors : Remove RingConsoleColors from distribution
-nomurmuhash     : Remove RingMurmurHash from distribution
-nocruntime      : Remove C Runtime from distribution
-qt              : Add RingQt to distribution
-allegro         : Add RingAllegro to distribution
-openssl         : Add RingOpenSSL to distribution
-libcurl         : Add RingLibCurl to distribution
```

(continues on next page)

(continued from previous page)

```
-mysql           : Add RingMySQL to distribution
-odbc            : Add RingODBC to distribution
-sqlite          : Add RingSQLite to distribution
-postgresql      : Add RingPostgreSQL to distribution
-opengl          : Add RingOpenGL to distribution
-freeglut        : Add RingFreeGLUT to distribution
-libzip          : Add RingLibZip to distribution
-libuv           : Add RingLibuv to distribution
-consolecolors   : Add RingConsoleColors to distribution
-murmurhash      : Add RingMurmurHash to distribution
-cruntime        : Add C Runtime to distribution
```

65.5 Building standalone console application

Using the "-static" option we can build executable console application

So we don't have to use ring.dll, ring.so or ring.dylib

This avoid only the need to Ring dynamic link library

If you are using another libraries, You will need to include it with your application.

```
ring2exe test.ring -static
```

65.6 Distributing RingAllegro Applications

We have test2.ring contains the next code

```
# Just a simple program to test Ring2EXE Tool!
# Using RingAllegro

load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

        oGame = New Game        # Create the Game Object
        {
                title = "My First Game"
        }
```

To build the executable file and prepare for distributing the Game

We use "-dist" option and "-allruntime" to include all libraries

```
ring2exe test2.ring -dist -allruntime
```

After executing the previous command

On Windows we will have : target/windows folder

On Linux we will have : target/linux folder

On macOS we will have : target/macos folder

The previous command will add all of the Ring runtime libraries to our distribution

But we may need only RingAllegro, So it's better to use the next command

```
ring2exe test2.ring -dist -allegro -cruntime
```

This will produce smaller size distribution and will avoid the runtime files that we don't need!

Also we could use the "-gui" option to hide the console window

So it's better to use the next command

```
ring2exe test2.ring -dist -gui -allegro -cruntime
```

65.7 Distributing RingQt Applications

We have test3.ring contains the next code

```
# Just a simple program to test Ring2EXE Tool!
# Using RingQt

load "guilib.ring"

new qApp {
        new qWidget() {
                setwindowtitle("Hello, World!")
                resize(400,400)
                show()
        }
        exec()
}
```

To build the executable file and prepare for distributing the GUI application

We use "-dist" option and "-allruntime" to include all libraries

```
ring2exe test3.ring -dist -allruntime
```

After executing the previous command

On Windows we will have : target/windows folder

On Linux we will have : target/linux folder

On macOS we will have : target/macos folder

The previous command will add all of the Ring runtime libraries to our distribution

But we may need only RingQt, So it's better to use the next command

```
ring2exe test3.ring -dist -qt -cruntime
```

This will produce smaller size distribution and will avoid the runtime files that we don't need!

Also we could use the "-gui" option to hide the console window

So it's better to use the next command

```
ring2exe test3.ring -dist -gui -qt -cruntime
```

65.8 Distributing Applications for Mobile using RingQt

To prepare a Qt project for your RingQt application (test3.ring) use the "-mobileqt" option

Example :

```
ring2exe test3.ring -dist -mobileqt
```

After executing the previous command, We will have the Qt project in target/mobile/qtproject folder

The main project file will be project.pro which we can open using the Qt Creator IDE.

Also we will have the resource file : project.qrc

Another important file is our C++ main file : main.cpp

65.9 Distributing Applications for WebAssembly using RingQt

To prepare a Qt project (WebAssembly) for your RingQt application (myapp.ring) use the "-webassemblyqt" option

Example :

```
ring2exe myapp.ring -dist -webassemblyqt
```

After executing the previous command, We will have the Qt project in target/webassembly/qtproject folder

The main project file will be project.pro which we can open using the Qt Creator IDE.

Also we will have the resource file : project.qrc

Another important file is our C++ main file : main.cpp

65.10 Building the Cards Game for Mobile using RingQt

For a better example, consider building an Android package for the Cards game that comes with the

Ring language in this folder : ring/application/cards

The Cards game folder contains three files

cards.ring : The Game source code

cards.jpg : The image file used by the game

project.qrc : Resource file to be used with the Qt project

The resource file contains the next content

```
<RCC>
        <qresource>
                <file>cards.ringo</file>
                <file>cards.jpg</file>
```

(continues on next page)

(continued from previous page)

```
        </qresource>
</RCC>
```

We have two files in the resource file

The first file is cards.ringo (The Ring Object File) and the second file is cards.jpg (The image file)

As a start, Ring2EXE will generate this resource file in target/mobile/qtproject/project.qrc

But this file will contains only cards.ringo (That Ring2EXE will generate by calling Ring compiler)

We need to update this resource file to add the image file : cards.jpg

After this update, we copy the resource file to the main application folder

So when we use Ring2EXE again, Our updated resource file will be used!

Now to build the cards game for Mobile

(1) Run the next command

```
ring2exe cards.ring -dist -mobileqt
```

(2) Open target/mobile/qtproject/project.pro using Qt creator

(3) Build and Run using Qt Creator

How the Cards game will find the image file ?

RingQt comes with a simple function : AppFile() that we can use to determine the files that we may access on Desktop or Mobile platforms

The next code from cards.ring

```
mypic = new QPixmap(AppFile("cards.jpg"))
```

So all what you need is using AppFile() function around your image files!

65.11 Building the Weight History Application for Mobile using RingQt

Another example to distribute your application for Mobile Devices using Ring2EXE and Qt

consider building an Android package for the Weight History application that comes with the Ring language in this folder : ring/application/weighthistory

The Weight History application folder contains four files

weighthistory.ring : The application source code

weighthistory.db : The SQLite database

project.qrc : The resource file for the Qt project

main.cpp : The main C++ source file for the Qt project

To build the Weight History application for Mobile

(1) Run the next command

```
ring2exe weighthistory.ring -dist -mobileqt
```

(2) Open target/mobile/qtproject/project.pro using Qt creator

(3) Build and Run using Qt Creator

The resource file (project.qrc) contains two files

```
<RCC>
        <qresource>
                <file>weighthistory.ringo</file>
                <file>weighthistory.db</file>
        </qresource>
</RCC>
```

The first file is weighthistory.ringo (Ring Object File - Generated by Ring2EXE by calling Ring compiler)

The database file : weighthistory.db

The main.cpp contains the next little update, To copy the database file from resources to a writable location on the mobile device

```
QString path3 ;
path3 = path+"/weighthistory.db";
QFile::copy(":/weighthistory.db",path3);
```

You will need to do this with database files only!

When we use Ring2EXE, the tool will check for project.qrc and main.cpp, if they exist then your updated files will be used in target/mobile/qtproject instead of the default version generated by Ring2EXE

So Use Ring2EXE to generate these files, Then copy them to your application folder when you update them.

65.12 Building the Form Designer for Mobile using RingQt

To build the Form Designer application (ring/tools/formdesigner) for Mobile

(1) Run the next command

```
ring2exe formdesigner.ring -dist -mobileqt
```

(2) Open target/mobile/qtproject/project.pro using Qt creator

(3) Build and Run using Qt Creator

in the folder ring/application/formdesigner You will find the resource file : project.qrc

It will be used automatically by Ring2EXE

```
<RCC>
        <qresource>
                <file>formdesigner.ringo</file>
                <file>image/allevents.png</file>
                <file>image/checkbox.png</file>
                <file>image/close.png</file>
                <file>image/combobox.bmp</file>
```

(continues on next page)

(continued from previous page)

```
                <file>image/datepicker.bmp</file>
                <file>image/dial.png</file>
                <file>image/formdesigner.png</file>
                <file>image/frame.png</file>
                <file>image/grid.bmp</file>
                <file>image/hyperlink.png</file>
                <file>image/image.png</file>
                <file>image/label.png</file>
                <file>image/layout.png</file>
                <file>image/lcdnumber.png</file>
                <file>image/listview.png</file>
                <file>image/lock.png</file>
                <file>image/new.png</file>
                <file>image/open.png</file>
                <file>image/progressbar.png</file>
                <file>image/project.png</file>
                <file>image/pushbutton.png</file>
                <file>image/radiobutton.png</file>
                <file>image/save.png</file>
                <file>image/saveas.png</file>
                <file>image/select.png</file>
                <file>image/slider.png</file>
                <file>image/spinner.bmp</file>
                <file>image/statusbar.png</file>
                <file>image/tab.png</file>
                <file>image/textarea.png</file>
                <file>image/textfield.png</file>
                <file>image/timer.png</file>
                <file>image/toolbar.png</file>
                <file>image/tree.bmp</file>
                <file>image/videowidget.png</file>
                <file>image/webview.png</file>
        </qresource>
</RCC>
```

As we did in the Cards game, The Form Designer will use the AppFile() function to determine the name of the Image files.

The next code from ring/tools/formdesigner/mainwindow/formdesignerview.ring

```
func CreateToolBar
        aBtns = [
                        new qtoolbutton(win) {
                                setbtnimage(self,AppFile("image/new.png"))
                                setclickevent(Method(:NewAction))
                                settooltip("New File")
                        } ,
                        new qtoolbutton(win) {
                                setbtnimage(self,AppFile("image/open.png"))
                                setclickevent(Method(:OpenAction))
                                settooltip("Open File")
                        } ,
                        new qtoolbutton(win) {
```

(continues on next page)

(continued from previous page)

```
                                setbtnimage(self,AppFile("image/save.png"))
                                setclickevent(Method(:SaveAction))
                                settooltip("Save")
                        } ,
                        new qtoolbutton(win) {
                                setbtnimage(self,AppFile("image/saveas.png"))
                                setclickevent(Method(:SaveAsAction))
                                settooltip("Save As")
                        } ,
                        new qtoolbutton(win) {
                                setbtnimage(self,AppFile("image/close.png"))
                                setclickevent(Method(:ExitAction))
                                settooltip("Exit")
                        }
                ]

        tool1 = win.addtoolbar("files")  {
                for x in aBtns { addwidget(x) addseparator() }
        }
```

From this example, We know that we can use sub folders for images.

65.13 Creating the Qt resource file using Folder2qrc

When we have large RingQt project that contains a lot of images and files, We need to add these files to the resource file (*.qrc) when distributing applications for Mobile devices.

Instead of adding these files one by one, Ring 1.6 comes with a simple tool that save our time, It's called Folder2qrc.

Example:

```
folder2qrc formdesigner.ring
```

We determine the main source file while we are in the application folder, and Folder2qrc will check all of the files in the current folder and sub folders, Then add them to the resource file after the mainfile.ringo (In our example this will be formdesigner.ringo)

The output file will be : project.qrc

You can open it and remove the files that you don't need in the resources!

65.14 Important Information about Ring2EXE

- Using Ring2EXE to prepare distribution will delete all of the files in the old distribution

for example, if you have target/windows folder then used

```
ring2exe test3.ring -dist -allruntime
```

The files in target/windows will be deleted before adding the files again

This is important when you prepare a distribution for Mobile devices

```
ring2exe test3.ring -dist -mobileqt
```

If you modified the resource file : project.qrc or the main file : main.cpp

Don't forget to copy them to the application folder!

So Ring2EXE can use the updated version if you tried the previous command again!

- Ring2EXE is written in Ring, and you can read the source code from

  https://github.com/ring-lang/ring/blob/master/tools/ring2exe/ring2exe.ring

- The libraries information are stored in a separated files, So these files can be updated in the future

automatically to support new libraries

https://github.com/ring-lang/ring/blob/master/tools/ring2exe/libs

CHAPTER

SIXTYSIX

THE RING PACKAGE MANAGER (RINGPM)

In this chapter we will learn about using the Ring Package Manager (RingPM)

RingPM is a tool for discovering, installing and updating Ring packages.

66.1 Features

The Package Manager uses Semantic Versioning to check compatibility between packages

The Package Manager comes with the next options

```
Usage   : ringpm [command]
Command : search  [keywords...]
Command : refresh : Update the Registry (Packages List)
Command : install [ <packagename> [from <UserName>] [branch <branchname>] ]
Command : list    [-u : check updates]
Command : run     [packagename]
Command : update  <packagename>
Command : remove  <packagename>
Command : format  : Delete All Packages
Command : new     <packagename>
Command : package : Create package in the current folder
```

66.2 Discovering Packages

We can discover new packages using the Search command

Using this command we can search in the RingPM Registry (Packages Index)

The RingPM Registry is a local copy of all registered packages.

```
ringpm search [keywords...]
```

Example:

```
ringpm search notepad
```

Output:

```
Package : ringnotepad (Ring Notepad)
Package : notepadppeditorextension (Notepad++ Editor Extension package)
```

To print all packages in the RingPM Registry, use the search command without keywords.

Example:

```
ringpm search
```

66.3 Updating the RingPM Registry

The RingPM Registry is a local copy of all registered packages.

We can update the local copy using the Refresh command

Example:

```
ringpm refresh
```

Output:

```
No updates to the Registry, Nothing to do!
```

Or

```
The Registry is updated from revision 110 (2019/01/13) to revision 112 (2019/01/15)
```

66.4 Installing Packages

We can install new packages using the Install command

```
ringpm install [ <packagename> [from <UserName>] [branch <branchname>] ]
```

We can type only the package name to get the package information from the RingPM Registry or we can determine the user name (GitHub) and the branch name of the github project (optional).

If the current folder is a package folder then we don't need to write the package name.

Example (1) :

```
ringpm install ringnotepad
```

Example (2) :

```
ringpm install goldmagic800
```

Example (3) :

```
ringpm install gameoflife
```

If the package is not added to the RingPM Registry, We can install it directly from the GitHub user

Example (4) :

```
ringpm install firstpackage from mahmoudfayed
```

To run the package after installation

```
ringpm run firstpackage
```

To install a package in the current folder

Example (5) :

```
ringpm install
```

66.5 Printing List of Installed Packages

We can know the installed packages using the List command

```
ringpm list [-u : check updates]
```

Example

```
ringpm list
```

Output

```
(analogclock)          : The AnalogClock Package [master]         -- (1.0.0)
(androidringlibsdl)    : The AndroidRingLibSDL Package [master]   -- (1.0.0)
(androidringqt)        : The AndroidRingQt Package [master]       -- (1.0.0)
(atomeditorextension)  : The AtomEditorExtension Package [master] -- (1.0.0)
(bignumber)            : The BigNumber Package [master]           -- (1.0.0)
(calculator)           : The Calculator Package [master]          -- (1.0.0)
(cards)                : The Cards Package [master]               -- (1.0.0)
(checkers)             : The Checkers Package [master]            -- (1.0.0)
(chess)                : The Chess Package [master]               -- (1.0.0)
....
```

To check for new updates

```
ringpm list -u
```

66.6 Run Package

After installing a package, we can run it using the Run command.

```
ringpm run [packagename]
```

Example(1):

```
ringpm run ringnotepad
```

Example(2):

```
ringpm run goldmagic800
```

Example(3):

```
ringpm run gameoflife
```

To run a package in the current folder

Example(4):

```
ringpm run
```

66.7 Update Package

We can update a package using the Update command

```
ringpm update <packagename>
```

Example:

```
ringpm update ringnotepad
```

66.8 Remove Package

We can remove a package using the Remove command

```
ringpm remove <packagename>
```

Example:

```
ringpm remove ringnotepad
```

66.9 Deleting All Packages

We can delete all packages using the Format command

Example:

```
ringpm format
```

66.10 Creating New Package

We can create new package using the New command

```
ringpm new <packagename>
```

Example:

```
ringpm new myapp
```

This will create new folder called my myapp

The new folder will contains the next file

- package.ring : The package description and files
- main.ring : main program (used by the Run command)
- lib.ring : library file for the package

File : main.ring

```
# The Main File

load "lib.ring"

func main

        ? "Hello, World!"
```

File : lib.ring

```
# The Library File
```

File : package.ring

```
aPackageInfo = [
        :name = "The myapp Package",
        :description = "Our myapp package using the Ring programming language",
        :folder = "myapp",
        :developer = "",
        :email = "",
        :license = "MIT License",
        :version = "1.0.0",
        :ringversion = "1.10",
        :versions =     [
                [
                        :version = "1.0.0",
                        :branch = "master"
                ]
        ],
        :libs =         [
                [
                        :name = "",
                        :version = "",
                        :providerusername = ""
                ]
        ],
        :files =        [
                "lib.ring",
                "main.ring"
        ],
        :ringfolderfiles =      [

        ],
        :windowsfiles =         [
```

(continues on next page)

(continued from previous page)

```
        ],
        :linuxfiles =   [

        ],
        :ubuntufiles =  [

        ],
        :fedorafiles =  [

        ],
        :macosfiles =   [

        ],
        :windowsringfolderfiles =       [

        ],
        :linuxringfolderfiles =         [

        ],
        :ubunturingfolderfiles =        [

        ],
        :fedoraringfolderfiles =        [

        ],
        :macosringfolderfiles =         [

        ],
        :run = "ring main.ring",
        :windowsrun = "",
        :linuxrun = "",
        :macosrun = "",
        :ubunturun = "",
        :fedorarun = "",
        :setup = "",
        :windowssetup = "",
        :linuxsetup = "",
        :macossetup = "",
        :ubuntusetup = "",
        :fedorasetup = "",
        :remove = "",
        :windowsremove = "",
        :linuxremove = "",
        :macosremove = "",
        :ubunturemove = "",
        :fedoraremove = ""
]
```

66.11 The Package Description File

The package description file contains the package information defined in the list aPackageInfo

Attribute	Description
Name	Package Name
Description	Package Description
Folder	The Folder Name (Will be created in ring/ringpm/packages)
Developer	The Package Developer Name
Email	The Package Developer Email
License	The Package License
Version	The Current Version of the Package (Latest Release)
RingVersion	The Required Ring Language Version (Minimum Version)
Versions	List of different versions provided by different branches in the GitHub project
Libs	List of dependencies (Defined by name, version & GitHub user name)
Files	List of files (will be installed in ring/ringpm/packages/[Folder]
RingFolderFiles	List of files (will be installed in ring folder)
WindowsFiles	Like (Files) but for Microsoft Windows Only
LinuxFiles	Like (Files) but for Linux Only
MacOSFiles	Like (Files) but for macOS Only
UbuntuFiles	Like (Files) but for Ubuntu Only
FedoraFiles	Like (Files) but for Fedora Only
WindowsRingFolderFiles	Like (RingFolderFiles) but for Microsoft Windows Only
LinuxRingFolderFiles	Like (RingFolderFiles) but for Linux Only
MacOSRingFolderFiles	Like (RingFolderFiles) but for macOS Only
UbuntuRingFolderFiles	Like (RingFolderFiles) but for Ubuntu Only
FedoraRingFolderFiles	Like (RingFolderFiles) but for Fedora Only
Run	System Command (Command prompt or Terminal) to run the package
WindowsRun	Like (Run) but for Microsoft Windows Only
LinuxRun	Like (Run) but for Linux Only
MacOSRun	Like (Run) but for macOS Only
UbuntuRun	Like (Run) but for Ubuntu Only
FedoraRun	Like (Run) but for Fedora Only
Setup	System Command (Command prompt or Terminal) after downloading the package files
WindowsSetup	Like (Setup) but for Microsoft Windows Only
LinuxSetup	Like (Setup) but for Linux Only
MacOSSetup	Like (Setup) but for macOS Only
UbuntuSetup	Like (Setup) but for Ubuntu Only
FedoraSetup	Like (Setup) but for Fedora Only
Remove	System Command (Command prompt or Terminal) before removing the package files
WindowsRemove	Like (Remove) but for Microsoft Windows Only
LinuxRemove	Like (Remove) but for Linux Only
MacOSRemove	Like (Remove) but for macOS Only
UbuntuRemove	Like (Remove) but for Ubuntu Only
FedoraRemove	Like (Remove) but for Fedora Only

66.12 Create Package in the Current Folder

To create a package for an application that already exists, go to the application folder then type

```
ringpm package
```

This will create the package definition file (package.ring) and will add all of the application files to the package definition.

Each RingPM package contains the package definition file (package.ring)

The package definition file is a list that describe the package information and files.

Example :

The package definition file for the Ring Notepad package

```
aPackageInfo = [
        :name = "The RingNotepad Package",
        :description = "Our RingNotepad package using the Ring programming language",
        :folder = "ringnotepad",
        :developer = "Mahmoud Fayed",
        :email = "msfclipper@yahoo.com",
        :license = "MIT License",
        :version = "1.0.0",
        :ringversion = "1.10",
        :versions =     [
                [
                        :version = "1.0.0",
                        :branch = "master"
                ]
        ],
        :libs =         [
                [
                        :name = "stdlib",
                        :version = "1.0",
                        :providerusername = ""
                ],
                [
                        :name = "ringqt",
                        :version = "1.0",
                        :providerusername = ""
                ],
                [
                        :name = "findinfiles",
                        :version = "1.0",
                        :providerusername = ""
                ],
                [
                        :name = "formdesigner",
                        :version = "1.0",
                        :providerusername = ""
                ],
                [
                        :name = "libdepwin_apache",
```

(continues on next page)

(continued from previous page)

```
                        :version = "1.0",
                        :providerusername = ""
                ]
        ],
        :files =        [
                "main.ring",
                "README.md"
        ],
        :ringfolderfiles =      [
                "applications/rnote/batch/killwebserver.bat",
                "applications/rnote/batch/run.bat",
                "applications/rnote/batch/run2.bat",
                "applications/rnote/image/back.jpg",
                "applications/rnote/image/back2.jpg",
                "applications/rnote/image/close.png",
                "applications/rnote/image/colors.png",
                "applications/rnote/image/copy.png",
                "applications/rnote/image/cut.png",
                "applications/rnote/image/debug.png",
                "applications/rnote/image/font.png",
                "applications/rnote/image/formdesigner.png",
                "applications/rnote/image/new.png",
                "applications/rnote/image/notepad.png",
                "applications/rnote/image/open.png",
                "applications/rnote/image/paste.png",
                "applications/rnote/image/print.png",
                "applications/rnote/image/project.png",
                "applications/rnote/image/richtext.png",
                "applications/rnote/image/run.png",
                "applications/rnote/image/rungui.png",
                "applications/rnote/image/save.png",
                "applications/rnote/image/saveas.png",
                "applications/rnote/image/search.png",
                "applications/rnote/image/source.png",
                "applications/rnote/image/undo.png",
                "applications/rnote/image/web.png",
                "applications/rnote/README.md",
                "applications/rnote/rnote.ring",
                "applications/rnote/rnoteactivefile.ring",
                "applications/rnote/rnoteactivefolder.ring",
                "applications/rnote/rnoteautocomplete.ring",
                "applications/rnote/rnotebase.ring",
                "applications/rnote/rnotecontroller.ring",
                "applications/rnote/rnotedistribute.ring",
                "applications/rnote/rnoteeditmenu.ring",
                "applications/rnote/rnoteeditorevents.ring",
                "applications/rnote/rnotefilemenu.ring",
                "applications/rnote/rnotefiles.ring",
                "applications/rnote/rnotefilestabs.ring",
                "applications/rnote/rnotefind.ring",
                "applications/rnote/rnotefindinfiles.ring",
                "applications/rnote/rnoteformdesigner.ring",
```

(continues on next page)

(continued from previous page)

```
                "applications/rnote/rnotegoto.ring",
                "applications/rnote/rnotehelp.ring",
                "applications/rnote/rnotelists.ring",
                "applications/rnote/rnotemainfiletoolbar.ring",
                "applications/rnote/rnotemainwindow.ring",
                "applications/rnote/rnotemode.ring",
                "applications/rnote/rnoteoutputwindow.ring",
                "applications/rnote/rnoteprogrammenu.ring",
                "applications/rnote/rnoterun.ring",
                "applications/rnote/rnotesettings.ring",
                "applications/rnote/rnotestyle.ring",
                "applications/rnote/rnotetabwidth.ring",
                "applications/rnote/rnotetools.ring",
                "applications/rnote/rnotetreeviewevents.ring",
                "applications/rnote/rnoteviewmenu.ring",
                "applications/rnote/rnotewebbrowser.ring"
        ],
        :run = "ring main.ring"
]
```

66.13 The RingPM Registry File

The RingPM Registry is a local copy of all registered packages.

```
aPackagesRegistry = [
        [ :name = "ringpresentation",
          :description = "Powerpoint presentation for the Ring programming language",
          :ProviderUserName = "ringpackages"
        ]
        # ...
]
```

Each package is defined using a list that determine

- Package Name
- Package Description
- Provider User Name (GitHub User Name)

We can register new packages by updating the registry file then sending a Pull Request

URL : https://github.com/ring-lang/ring/blob/master/tools/ringpm/registry/registry.ring

CHAPTER

SIXTYSEVEN

USING ZEROLIB

In this chapter we will learn how to use the ZeroLib library.

67.1 Introduction

ZeroLib is a simple library written in Ring.

The library provide classes for Lists and String where the index starts from 0.

67.2 Z() function

Syntax:

```
Z(String|List) ---> New Object (ZeroBasedString|ZeroBasedList)
```

67.3 ZeroBasedList Class

Simple class provide a List where the index starts from zero.

Methods:

Method	Description/Output
Init(List)	
Add(Value)	Add item to the list
Insert(nIndex,Value)	Inset Item after nIndex
Find(Value)	Find item
Delete(nIndex)	Delete item from the list
Item(nIndex)	Get item from the list
First()	Get the first item in the list
Last()	Get the last item in the list
Set(nIndex,Value)	Set item value
FindInColumn(nCol,Value)	Find item in a column
Sort()	Sort items - return new list
Reverse()	Reverse items - return new list
Swap(nIndex1,nIndex2)	Swap two items

Example:

```
load "zerolib.ring"

? "Using List - Index start from 0"
List = Z( [1,2,3] )
List.Add(4)
List.Add(5)
? List[0]
? List[1]
? List[2]
? List[3]
? List[4]
nIndex = List.find(2)
? "Find(2) = " + nIndex
List.delete(0)
? "After deleting the first item : List[0]"
? "Now List[0] = " + List[0]
```

Output:

```
Using List - Index start from 0
1
2
3
4
5
Find(2) = 1
After deleting the first item : List[0]
Now List[0] = 2
```

67.4 ZeroBasedString Class

Simple class provide a String where the index starts from zero.

Method	Description/Output
Init(String|Number)	
Lower()	New String - Lower case characters
Upper()	New String - Upper case characters
Left(x)	New String - contains x characters from the left
Right(x)	New String - contains x characters from the right
Lines()	Number - Lines count
Trim()	New String - Remove Spaces
Copy(x)	New String - repeat string x times
strcmp(cString)	Compare string with cString
tolist()	List (String Lines to String Items)
tofile(cFileName)	Write string to file
mid(nPos1,nPos2)	New String - from nPos1 to nPos2
getfrom(nPos1)	New String - from nPos1 to the end of the string
replace(cStr1,cStr2,lCase)	New String - Replace cStr1 with cStr2 , lCase (True=Match Case)
split()	List - Each Word as list item
startswith(substring)	Return true if the start starts with a substring
endswith(substring)	Return true if the start ends with a substring

Example:

```
load "zerolib.ring"

? "Using String - Index start from 0"
String = Z( "Welcome" )
? String[0]
? String[1]
? String[2]
? String[3]
? String[4]
? String[5]
? String[6]
```

Output:

```
Using String - Index start from 0
W
e
l
c
o
m
e
```

67.5 Source Code

We can find the library source code in this folder

URL : https://github.com/ring-lang/ring/tree/master/libraries/zerolib

CHAPTER

SIXTYEIGHT

FOXRING FUNCTIONS REFERENCE

A class contains functions similar to FoxPro functions.

68.1 FoxRing functions

Function Name	Description
frAbs()	Returns the absolute value of the specified numeric expression.
frAddBs()	Adds a backslash (if needed) to a path expression.
frALines()	Creates an Array with the content of the specified string.
frAllTrim()	Removes all leading and trailing spaces of the specified string.
frAsc()	Returns the ANSI value for the leftmost character in a character expression.
frAt()	Searches a character expression for the occurrence of another character expression.
frAtC()	Searches a character expression for the occurrence of another character expression without regard for the case of these two expressions.
frBetween()	Determines whether the value of an expression is inclusively between the values of two expressions of the same type.
frChr()	Returns the character associated with the specified numeric ANSI code.
frEmpty()	Determines whether an expression evaluates to empty.
frFile()	Checks if a file exists on disk.
frFileToStr()	Returns the contents of a file as a character string.
frForceExt()	Returns a string with the old file name extension replaced by a new extension.
frForcePath()	Returns a file name with a new path name substituted for the old one.
frIif()	Returns one of two values depending on the value of a logical expression.
frInList()	Determines whether an expression matches another expression in a list.
frInt()	Evaluates a numeric expression and returns the integer portion of the expression.
frJustDrive()	Returns the drive letter from a complete path.
frJustExt()	Returns the characters of a file extension from a complete path.
frJustFName()	Returns the file name portion of a complete path and file name.
frJustPath()	Returns the path portion of a complete path and file name.
frJustStem()	Returns the stem name (the file name before the extension) from a complete path and file name.
frLen()	Determines the number of characters in a character expression, indicating the length of the expression.
frListToString()	Creates a string with the string elements of an Array.
frLTrim()	Removes all leading spaces or parsing characters from the specified character expression.
frPadL()	Returns a string from an expression, padded with spaces or characters to a

continues on next page

Table 1 – continued from previous page

Function Name	Description
	specified length on the left side.
frPadR()	Returns a string from an expression, padded with spaces or characters to a specified length on the right side.
frProper()	Returns from a character expression a string capitalized as appropriate for proper names.
frReplicate()	Returns a character string that contains a specified character expression repeated a specified number of times.
frRTrim()	Removes all trailing spaces or parsing characters from the specified character expression.
frSetIfEmpty()	Set a Value into a variable if the variable value is empty, null or zero.
frSetSeparatorTo()	Specifies the character for the numeric place separator.
frSpace()	Returns a character string composed of a specified number of spaces.
frStr()	Returns the character equivalent of a numeric expression.
frStringToList()	Creates a List with the content of the specified string.
frStrTran()	Searches a character expression for a second character expression and replaces each occurrence with a third character expression.
frStuff()	Returns a new character string replaced by a specified number of characters in a character expression with another character expression.
frSubStr()	Returns a character string from the given character expression, starting at a specified position in the character expression and continuing for a specified number of characters.
frTransform()	Returns a character string from an expression in a format determined by a format code.
frVal()	Returns a numeric value from a character expression composed of numbers.
frVarType()	Returns the data type of an expression.

68.2 frAbs() function

```
* Syntax       : lnReturnValue = frAbs(tnExpression)
* Description  : Returns the absolute value of the specified numeric expression.
*              :
* Arguments    : <tnExpression>
*              : Specifies the numeric expression whose absolute value frAbs()
*              : returns.
* Returns      : <lnReturnValue>
*              : Returns the absolute value of the specified numeric expression.
```

68.3 frAsc() function

```
* Syntax       : lnReturnValue = frAsc(tcExpression)
* Description  : Returns the ANSI value for the leftmost character in
*              : a character expression.
* Arguments    : <tcExpression>
*              : Specifies the character expression containing the character
*              : whose ANSI value frAsc()
```

(continues on next page)

(continued from previous page)

```
*               : returns. Any characters after the first character in
*               : tcExpression are ignored by frAsc().
* Returns       : <lnReturnValue>
*               : returns the position of the character in the character
*               : table of the current code page.
*               : Every character has a unique ANSI value in the
*               : range from 0 to 255.
```

68.4 frAddBs() function

```
* Syntax        : lcReturnValue = frAddBs(tcPath)
* Description   : Adds a backslash (if needed) to a path expression.
*               :
* Arguments     : <tcPath>
*               : Specifies the path name to which to add the backslash.
*               :
* Returns       : <lcReturnValue> The path with the backslash.
```

68.5 frAt() function

```
* Syntax        : lnPos = frAt(tcToSearch, tcString, tnOccurrence)
* Description   : Searches a character expression for the occurrence of
*               : another character expression.
*               : The search performed by frAt() is case-sensitive.
*               :
* Arguments     : <tcToSearch>
*               : Specifies the character expression to search
*               : for in <tcString>.
*               : <tcString>
*               : Specifies the character expression to search
*               : for <tcToSearch>.
*               : <tnOccurrence>
*               : Specifies which occurrence, first, second, third,
*               : and so on, of <tcToSearch> to search for
*               : in <tcString>.
*               : By default, frAt() searches for the first occurrence
*               : of <tcToSearch> (tnOccurrence = 1).
* Returns       : Numeric. frAt() returns an integer indicating the
*               : position of the first character for a
*               : character expression or memo field within another
*               : character expression or memo field,
*               : beginning from the leftmost character. If the
*               : expression or field is not found, or if
*               : <tnOccurrence> is greater than the number of
*               : times <tcToSearch> occurs in  <tcString>, frAt()
*               : returns 0.
```

68.6 frAtC() function

```
* Syntax       : lnPos = frAtC(tcToSearch, tcString, tnOccurrence)
* Description  : Searches a character expression for the occurrence
*               : of another character expression
*               : without regard for the case of these two expressions.
*               :
* Arguments    : <tcToSearch>
*               : Specifies the character expression to search
*               : for in <tcString>.
*               : <tcString>
*               : Specifies the character expression to search
*               : for <tcToSearch>.
*               : <tnOccurrence>
*               : Specifies which occurrence, first, second, third,
*               : and so on, of <tcToSearch> to search for
*               : in tcString.
*               : By default, frAtC() searches for the first occurrence
*               : of <tcToSearch> (tnOccurrence = 1).
* Returns      : Numeric. frAtC() returns an integer indicating the
*               : position of the first character for a
*               : character expression or memo field within
*               : another character expression or memo field,
*               : beginning from the leftmost character. If the
*               : expression or field is not found, or if
*               : <tnOccurrence> is greater than the number of
*               : times <tcToSearch> occurs in <tcString>, frAtC()
*               : returns 0.
```

68.7 frChr() function

```
* Syntax       : lcReturnValue = frChr(tnExpression)
* Description  : Returns the character associated with the specified numeric
*               :  ANSI code.
* Arguments    : <tnExpression>
*               : Specifies a number between 0 and 255 whose equivalent ANSI
*               : character frChr() returns.
* Returns      : <lcReturnValue>
*               : Returns a single character corresponding to the numeric
*               : position of the character in the
*               : character table of the current code page.
*               :
* Remarks      : tnExpression must be between 0 and 255
```

68.8 frEmpty() function

```
* Syntax       : llReturnValue = frEmpty(tuExpression)
* Description  : Determines whether an expression evaluates to empty.
*              :
* Arguments    : <tuExpression>
*              : Specifies the expression that EMPTY() evaluates.
*              : You can specify an expression with Character,
*              : Numeric, or logical type.
*              :
* Returns      : <llReturnValue> Logical
```

68.9 frFile() function

```
* Syntax       : llReturnValue = frFile(tcFileName, tnFlag)
* Description  : Checks if the specified file exists on disk.
*              :
* Arguments    : <tcFileName>
*              : Specifies the name of the file to check.
*              : tcFileName must include
*              : the file extension. You can include a path with
*              : the file name to
*              : search for a file in a directory or on a drive
*              : other than the current directory or drive.
*              :
*              : <tnFlag>
*              : tnFlag was included for future compatibility.
*              : In this version, It always returns true whenever
*              : the file exists on disk.
* Returns      : <llReturnValue> Logical
*              : True if file exists on disk.
*              : False if file doesn't exist on disk.
```

68.10 frFileToStr() function

```
* Syntax       : lcReturnValue = frFileToStr(tcFileName)
* Description  : Returns the contents of a file as a character string.
*              :
* Arguments    : <tcFileName>
*              : Specifies the name of the file whose contents are
*              : returned as a character
*              : string. If the file is in a directory other than
*              : the current default directory,
*              : include a path with the file name.
*              :
* Returns      : <lcReturnValue>
*              : A character string with the content of the specified file.
*              :
```

68.11 frStr() function

```
* Syntax       : lcReturnValue = frStr(tnValue, tnLen, tnDec)
* Description  : Returns the character equivalent of a numeric expression.
*               :
* Arguments    : <tnValue>
*               : Specifies the numeric expression to evaluate.
*               :
*               : <tnLen>
*               : Specifies the length of the character string returned.
*               : If tnLen is 0, tnLen defaults to 10 characters.
*               : If tnLen < 0 returns one string with same length as the number.
*               : Note
*               : If the expression contains a decimal point,
*               : the length includes one character for
*               : the decimal point and one character
*               : for each digit in the character string.
*               :
*               : <tnDec>
*               : Specifies the number of decimal places in the
*               : character string returned.
*               : To specify the number of decimal places using
*               : tnDec, you must include nLength. If nDecimalPlaces is omitted,
*               : the number of decimal places defaults to zero (0).
*               :
* Returns      : Character data type. frStr() returns a character string
*               : equivalent to the specified numeric expression.
*               : Depending on certain conditions, frStr() can return the following:
*               : If you specify fewer decimal places than exist in tnValue,
*               : the return value is rounded up. To round results to the nearest
*               : decimal place instead of upward, include the ROUND( ) function.
*               : For more information, see ROUND( ) Function.
*               : If nExpression is an integer, and nLength is less than
*               : the number of digits in nExpression, frStr( ) returns a string of
*               : asterisks, indicating numeric overflow.
*               : If nExpression contains a decimal point, and nLength is equal
*               : to or less than the number of digits to the left of the decimal
*               : point, frStr( ) returns a string of asterisks,
*               : indicating numeric overflow.
*               : If nLength is greater than the length of the value evaluated
*               : by nExpression, frStr( ) returns a character string padded with
*               : leading spaces.
*               : If nExpression has Numeric or Float type, and nLength
*               : is less than the number of digits in nExpression, and , frStr( )
*               : returns a value using scientific notation.
```

68.12 frSetIfEmpty() function

```
* Syntax       : tuReturnValue = frSetIfEmpty(tuValue, tuNewValue)
* Description  : Set a Value into a variable if the variable
*              : value is empty, null or zero.
* Arguments    : <tuValue>
*              : The value to evaluate.
*              :
*              : <tuNewValue>
*              : The value to set if tuValue is empty.
*              :
* Returns      : tuNewValue if tuValue is empty, otherwise
*              : returns the original value.
* Remarks      : This function doesn't exist in VFP.
```

68.13 frSpace() function

```
* Syntax       : lcReturnValue = frSpace(tnSpaces)
* Description  : Returns a character string composed of a
*              : specified number of spaces.
* Arguments    : <tnSpaces>
*              : Specifies the number of spaces that frSpace() returns.
*              :
* Returns      : <lcReturnValue>
*              : Character
```

68.14 frInList() function

```
* Syntax       : llReturnValue = frInList(tuExpression, taList)
* Description  : Determines whether an expression matches another
*              : expression in a set of expressions.
* Arguments    : <tuExpression>
*              : Specifies the expression frInList() searches for in the List.
*              :
*              : <taList>
*              : Specifies the List of expressions to search.
*              : You must include at least one element in the list.
*              : The expressions in the list of expressions needn't to be
*              : of the same data type.
*              :
* Returns      : <luReturnValue> Null or logical value.
```

68.15 frForcePath() function

```
* Syntax       : lcReturnValue = frForcePath(tcFileName, tcPath)
* Description  : Returns a file name with a new path name
*              : substituted for the old one.
* Arguments    : <tcFileName>
*              : Specifies the file name (with or without a path or extension),
*              : which will get a new path.
*              : <tcPath>
*              : Specifies the new path for tcFileName.
*              :
* Returns      : <lcReturnValue>
*              : Returns a file name with a new path name
*              : substituted for the old one.
```

68.16 frAllTrim() function

```
Syntax  : lcReturnValue = frAllTrim(tcExpression, tcCharacter)
```

68.17 frLTrim() function

```
Syntax  : lcRet = frLTrim(tcExpression, tcCharacter)
```

68.18 frJustDrive() function

```
* Syntax       : lcReturnValue = frJustDrive(tcPath)
* Description  : Returns the drive letter from a complete path.
*              :
* Arguments    : <tcPath>
*              : Specifies the complete path name for
*              : which you want only the drive.
* Returns      : <lcReturnValue>
*              : Returns the drive letter from a complete path.
```

68.19 frJustExt() function

```
* Syntax       : lcReturnValue = frJustExt(tcPath)
* Description  : Returns the characters of a file extension
*              : from a complete path.
* Arguments    : <tcPath>
*              : Specifies the name, which may include the full path,
*              : of the file for which you want only the extension.
* Returns      : <lcReturnValue>
```

(continues on next page)

(continued from previous page)

```
*               : Returns the drive characters of a file extension
*               : from a complete path.
```

68.20 frJustStem() function

```
* Syntax       : lcReturnValue = frJustStem(tcPath)
* Description  : Returns the stem name (the file name before the extension)
*               : from a complete path and file name.
* Arguments    : <tcPath>
*               : Specifies the name (including path) of the file
*               : for which you want only the stem.
* Returns      : <lcReturnValue>
*               : Returns the stem name of a file from a complete path.
```

68.21 frRTrim() function

```
Syntax : lcRet = frRTrim(tcExpression, tcCharacter)
```

68.22 frJustPath() function

```
Syntax  : tcReturnValue = frJustPath(tcExpression)
```

68.23 frForceExt() function

```
Syntax  : tcReturnValue = frForceExt(tcFileName, tcNewExtension)
```

68.24 frALines() function

```
Syntax  : tnReturnValue = frALines(taList, tcExpression, tcSeparator)
```

68.25 frJustFName() function

```
Syntax  : tcReturnValue = frJustFName(tcExpression)
```

68.26 frPadL() function

```
Syntax  : tcReturnValue = frPadL(tcString, tnLen, tcChar)
```

68.27 frPadR() function

```
Syntax  : tcReturnValue = frPadR(tcString, tnLen, tcChar)
```

68.28 frProper() function

```
* Syntax       : tcReturnValue = frProper(tcExpression)
* Description  : Returns from a character expression a string
*              : capitalized as appropriate for proper names.
* Arguments    : <tcExpression>
*              : Specifies the character expression from which
*              : frProper() returns a capitalized character string.
* Returns      : <tcReturnValue>
```

68.29 frReplicate() function

```
Syntax  : tcReturnValue = frReplicate(tcString, tnTimes)
```

68.30 frLen() function

```
Syntax  : tnReturnValue = frLen(tcString)
```

68.31 frStuff() function

```
* Syntax       : tcReturnValue = frStuff(tcExpression, tnStartRep,
                                         tnCharRep, tcToReplace)
* Description  : Returns a new character string replaced by a
*              : specified number of characters in a character
*              : expression with another character expression.
*              :
* Arguments    : <tcExpression>
*              : Specify the character expression in which the replacement occurs.
*              :
*              : <tnStartRep>
*              : Specify the position in <tcExpression> where the replacement begins.
*              :
```

(continues on next page)

(continued from previous page)

```
*               : <tnCharRep>
*               : Specifies the number of characters to be replaced.
*               : If <tnCharRep> is 0, the replacement string
*               : <tcToReplace> is inserted into <tcExpression>.
*               :
*               : <tcToReplace>
*               : Specifies the replacement character expression.
*               : If <tcToReplace> is an empty string, the number of
*               : characters specified by <tnCharRep> are removed from <tcExpression>.
*               :
* Returns       : Character
```

68.32 frSubStr() function

```
Syntax  : tcReturnValue = frSubStr(tcString, tnInitialPosition, tnNumberBytes)
```

68.33 frStrTran() function

```
Syntax  : tcReturnValue = frStrTran(tcString, tcOldString, tcNewString)
```

68.34 frListToString() function

```
* Syntax       : lcRet = frListToString(taList)
* Remarks      : This function doesn't exist in VFP.
```

68.35 frInt() function

```
Syntax          : lnInt = frInt(tnExpression)
```

68.36 frStringToList() function

```
* Syntax       : laList = frStringToList(tcExpression)
* Remarks      : This function doesn't exist in VFP.
```

68.37 frIIf() function

```
* Syntax       : luReturnValue = frIIf(tlExpression, tuReturnIfTrue, tuReturnIfFalse)
* Description  : Returns one of two values depending on the
*               : value of a logical expression.
* Arguments    : <tlExpression>
*               : Specifies the logical expression that frIIf() evaluates.
*               :
*               : <tuReturnTrue>, <tuReturnFalse>
*               : If tlExpression evaluates to True, tuReturnIfTrue is
*               : returned and tuReturnIfFalse is not evaluated.
*               : If tlExpression evaluates to False or Null, tuReturnIfFalse is
*               : returned and tuReturnIfTrue is not evaluated.
*               :
* Returns      : <luReturnValue> Defined by <tuReturnIfTrue> or <tuReturnIfFalse>
```

68.38 frVal() function

```
* Syntax       : luReturnValue = frVal(tcExpression)
* Description  : Returns a numeric value from a character expression
*               : composed of numbers
* Arguments    : <tcExpression>
*               : Specifies a character expression composed of up to 16 numbers.
*               :
* Returns      : <tnValue>
*               : Return a numeric value.
```

68.39 frBetween() function

```
* Syntax       : luReturnValue = frBetween(tuTestValue, tuLowValue, tuHighValue)
* Description  : Determines whether the value of an expression
*               : is inclusively between the
*               : values of two expressions of the same type.
*               :
* Arguments    : <tuTestValue>
*               : Specifies an expression to evaluate.
*               :
*               : <tuLowValue>
*               : Specifies the lower value in the range.
*               :
*               : <tuHighValue>
*               : Specifies the higher value in the range.
*               :
* Returns      : <luReturnValue>
*               : Returns a logical order null value.
```

68.40 frSetSeparatorTo() function

```
* Syntax       : frSetSeparatorTo(tuExpression)
* Description  : Specifies the character for the numeric place separator.
*              :
* Arguments    : <tuExpression>
*              : Specifies the character for the numeric place separator.
*              :
*              : Use frSetSeparatorTo() to change the numeric place
*              : separator from default, for example space " " or a comma ",".
*              : Issue frSetSeparatorTo(Null) to reset the value to its default.
*              :
* Returns      : None
```

68.41 frTransform() function

```
* Syntax       : tcReturnValue = frTransform(tuExpression, tcFormatCodes)
* Description  : Returns a character string from an expression in a
*              : format determined by a format code.
* Arguments    : <tuExpression>
*              : Specifies the expression to format.
*              :
*              : <tcFormatCodes>
*              : Specifies one or more format codes that determine how to
*              : format the expression.
*              :
* Returns      : <tcReturnValue>
```

The following table lists the available format codes for tcFormatCodes.

```
---------------------------------------------------------------------------
 Format Code   Description
---------------------------------------------------------------------------
 @!            Converts an entire character string to uppercase.
 @T            Trims leading and trailing spaces from character values.
 @B            Left-justifies Numeric data within the display region.
 @L            Pads numeric and string data with leading zeros.
 @C            Appends CR to positive numeric values to indicate a credit.
 @X            Appends DB to negative numeric values to indicate a debit.
---------------------------------------------------------------------------
```

68.42 frVarType() function

```
* Syntax       : lcRet = frVarType(tuExpression)
* Description  : Returns the data type of an expression.
*              :
* Arguments    : <tuExpression>
*              : Specifies the expression for which the data type is returned.
*              :  frVartype() returns a
*              : single character indicating the data type of the expression.
*              : The following table lists the characters that frVarType()
*              : returns for each data type.
*              :
*              : -------------------   -------------------------------------
*              : Return Value          Data Type
*              : -------------------   -------------------------------------
*              : C                     Character
*              : N                     Numeric
*              : A                     List
*              : O                     Object
*              : U                     Undefined type
*              : -------------------   -------------------------------------
*              :
* Returns      : Character
```

68.43 Example

```
Load "foxring.ring"


mf = new frFunctions

/*-----------------------------------------------------------*/
 * frProper() samples
/*-----------------------------------------------------------*/

lcStr1 = "ring is a good language"
?mf.frProper(lcStr1)
?mf.frProper(Upper(lcStr1))


/*-----------------------------------------------------------*/
 * frStuff() samples
/*-----------------------------------------------------------*/

lcStr1 = "abcdefghijklm"
lcStr2 = "12345"

// insert
?mf.frStuff(lcStr1, 4, 0, lcStr2)
// replace
```

(continues on next page)

(continued from previous page)

```
?mf.frStuff(lcStr1, 4, 3, lcStr2)
// delete
?mf.frStuff(lcStr1, 4, 6, "")
// replace and insert
?mf.frStuff(lcStr1, 4, 1, lcStr2)
// replace and delete
?mf.frStuff(lcStr1, 4, 4, lcStr2)
// replace, delete rest
?mf.frStuff(lcStr1, 4, Len(lcStr1), lcStr2)

/*-----------------------------------------------------------*/



?mf.frAbs(-45)
?mf.frAbs(10-30)
?mf.frAbs(30-10)

lcNumber1 = 40
lcNumber2 = 2

?mf.frAbs(lcNumber2-lcNumber1)




lcCompletFileName = "C:\ring\docs\source\contribute.txt"

?mf.frFile(lcCompletFileName, Null)
if mf.frFile(lcCompletFileName, Null) {
  ?mf.frFileToStr(lcCompletFileName)
else
  ?"File does not exist"
}

lcNewPath = "C:\ring_2\docs\source\"
?mf.frJustExt(lcCompletFileName)
?mf.frJustDrive(lcCompletFileName)
?mf.frJustStem(lcCompletFileName)
?mf.frForcePath(lcCompletFileName, lcNewPath)
?mf.frTransform("    Ring is a good language    ",
                "@! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!")
?mf.frAllTrim("    Ring is a good language    ", Null)
?mf._version
lnValue = 3125.54
?mf.frTransform(lnValue, "@B")+ "Euros"
?mf.frTransform(lnValue, "@C 9999,999,999,999.999")
mf.frSetSeparatorTo(" ")
?mf.frTransform(lnValue, "9999,999,999,999.999")
?mf.frInt(lnValue)
?mf.frForceExt("teste", "dbf")
// Format "@L" Added into frTransform() function
```

(continues on next page)

(continued from previous page)

```
?mf.frTransform("123", "@L 999999")
?mf.frTransform(123, "@L 999999")
```

CHAPTER

SIXTYNINE

BIGNUMBER LIBRARY

In this chapter we will learn about using the Big Number library.

69.1 Loading the library

Before using the next function load the bignumber.ring library

```
load "bignumber.ring"
# Use Big Number library functions
```

69.2 Examples

Using the BigNumber library we can do arithmetic operations on huge numbers.

Example:

```
load "bignumber.ring"

num1 = "6234567890123456789167834512345678 9"    ### Big
num2 =  "12378945678901234198712365 45"          ### Small
num3 =     "64"                                  ### Divide Small
num4 = "765432"
num5 =      "3"                                  ### Power

? "Add big numbers:"
a1 = new BigNumber(num1)        a1.Print()
a2 = new BigNumber(num2)        a2.Print()
a3 = a1 + a2                    a3.Print() ? nl

? "Substract big numbers:"
a1 = new BigNumber(num1)        a1.Print()
a2 = new BigNumber(num2)        a2.Print()
a3 = a1 - a2                    a3.Print() ? nl

? "Multiply big numbers:"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num2)        a2.print()
a3 = a1 * a2                    a3.print() ? nl
```

(continues on next page)

(continued from previous page)

```
? "Divide big numbers:"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num2)        a2.print()
a3 = a1 / a2                    a3.print() ? nl

? "Divide big numbers: by very small number"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num3)        a2.print()
a3 = a1 / a2                    a3.print() ? nl

? "Power of big number:"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num5)        a2.print()
a3 = a1 ^ a2                    a3.print() ? nl
```

Output:

```
Add big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
6234568013912913578180176499469333 4


Substract big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
5234568766334000000155492525222024 4


Multiply big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
7717737724326015010346217871419745473643247278011968230515400 5


Divide big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
50364288


Divide big numbers: by very small number
6234567890123456789167834512345678 9
64
9741512328317901233074741425540 12


Power of big number:
6234567890123456789167834512345678 9
3
2423366362614711720923471460317270 04 (Output continue in next line)
```

(continues on next page)

(continued from previous page)

```
371698195628343934238988256152289508 (Output continue in next line)
493964611043228971692389860897069
```

69.3 BigNumber Functions

The library contains the next functions

```
FuncAdd(num1,num2)
FuncSubtract(num1,num2)
FuncCompare(num1,num2)
FuncDivide(num1,num2)
FuncMultiply(num1,num2)
FuncPower(num1,num2)
FuncBinaryToDecimal(num1)
FuncDecimalToBinary(num1)
printBinaryDigits(binList)
printDecimalDigits(decList)
```

69.4 BigNumber Class

The library contains the next class

```
class BigNumber
        func init aPara
        func operator cOperator, Para
        func print
        func value
```

69.5 Library Source Code

You can see the library source code in : ring/libraries/bignumber folder

Source Code : https://github.com/ring-lang/ring/blob/master/libraries/bignumber/bignumber.ring

CHAPTER

SEVENTY

USING CSVLIB

In this chapter we will learn how to use the CSVLib library.

70.1 Introduction

CSVLib is a simple library written in Ring.

The library provide functions to read and write CSV Files.

70.2 Functions

The library comes with the next functions

```
List2CSV(aList) --> cCSVString
CSV2List(cCSVString) --> aList
```

70.3 Examples

Example(1)

```
load "csvlib.ring"

aList = [ ["number", "square" ] ]

for t=1 to 10
        aList + [ t, t*t ]
next

write( "squares.csv", list2CSV(aList) )
```

Output:

Example (2)

```
load "csvlib.ring"

if ! fexists("squares.csv")
        ? "The file squares.csv doesn't exist! - Run writeSquaresTable.ring to create it"
        return
ok

aList = CSV2List( read("squares.csv") )

for subList in aList
        ? "" + subList[1] + " - " + subList[2]
next
```

Output:

```
number - square
1 - 1
2 - 4
3 - 9
4 - 16
```

(continues on next page)

(continued from previous page)

```
5 - 25
6 - 36
7 - 49
8 - 64
9 - 81
10 - 100
```

CHAPTER

SEVENTYONE

USING JSONLIB

In this chapter we will learn how to use the JSONLib library.

71.1 Introduction

JSONLib is a simple library written in Ring.

The library provide functions to read and write JSON files.

71.2 Functions

The library comes with the next functions

```
List2JSON(aList) --> cJSONString
JSON2List(cJSONString) --> aList
```

71.3 Examples

Example (1):

File: sample.json

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 20,
  "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
  },
  "phoneNumbers": [
        { "type": "home", "number": "212 555-1234" },
        { "type": "fax", "number": "646 555-4567" }
  ]
}
```

Ring Code:

```
load "jsonlib.ring"

func main

        aList = JSON2List( read("sample.json") )

        ? aList[:FirstName]
        ? aList[:LastName]
        ? aList[:Age]
        ? aList[:Address][:city]
        ? aList[:phoneNumbers][1][:Type]
        ? aList[:phoneNumbers][1][:Number]
        ? aList[:phoneNumbers][2][:Type]
        ? aList[:phoneNumbers][2][:Number]
```

Output:

```
John
Smith
20
New York
home
212 555-1234
fax
646 555-4567
```

Example (2):

```
load "jsonlib.ring"

func main

        aList = [
                :name = "Ring",
                :year = 2016
        ]

        ? List2JSON(aList)
```

Output:

```
{
        "name": "Ring",
        "year": 2016
}
```

CHAPTER

SEVENTYTWO

USING HTTPLIB

In this chapter we will learn how to use the HTTPLib library.

72.1 Introduction

This extension provides support for the httplib library

URL: https://github.com/yhirose/cpp-httplib

72.2 Server Class Methods

- route(cType,cURL,cCode)
- setContent(cContent,cType)
- setHTMLPage(oPage)
- shareFolder(cFolder)
- setCookie(cStr)
- cookies() -> aList
- getFileContent(cFile) -> cString
- getFileName(cFile) -> cString

72.3 Example

```
load "httplib.ring"

oServer = new Server {

        ? "Try localhost:8080/hi"
        route(:Get,"/hi",:mytest)

        ? "Listen to port 8080"
        listen("0.0.0.0", 8080)

}
```

(continues on next page)

(continued from previous page)

```
func mytest
        oServer.setContent("Hello World!", "text/plain")
```

72.4 Samples

The samples exist in ring/samples/UsingHTTPLib folder

72.5 Printing Constants

The next example print the constants defined by the extension

```
load "httplib.ring"

? "Constants:"

? HTTPLIB_KEEPALIVE_TIMEOUT_SECOND
? HTTPLIB_KEEPALIVE_MAX_COUNT
? HTTPLIB_CONNECTION_TIMEOUT_SECOND
? HTTPLIB_CONNECTION_TIMEOUT_USECOND
? HTTPLIB_READ_TIMEOUT_SECOND
? HTTPLIB_READ_TIMEOUT_USECOND
? HTTPLIB_WRITE_TIMEOUT_SECOND
? HTTPLIB_WRITE_TIMEOUT_USECOND
? HTTPLIB_IDLE_INTERVAL_SECOND
? HTTPLIB_IDLE_INTERVAL_USECOND
? HTTPLIB_REQUEST_URI_MAX_LENGTH
? HTTPLIB_REDIRECT_MAX_COUNT
? HTTPLIB_PAYLOAD_MAX_LENGTH
? HTTPLIB_TCP_NODELAY
? HTTPLIB_COMPRESSION_BUFSIZ
? HTTPLIB_THREAD_POOL_COUNT
? HTTPLIB_RECV_FLAGS
? HTTPLIB_LISTEN_BACKLOG
```

72.6 Using HTTP GET

Example(1):

```
load "httplib.ring"

? "Start the server..."
oServer = new Server

? "Try localhost:8080/hi"
oServer.route(:Get,"/hi",:mytest)
```

(continues on next page)

(continued from previous page)

```
? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func mytest
        oServer.setContent("Hello World!", "text/plain")
```

Example(2):

```
load "httplib.ring"

? "Start the server..."
oServer = new Server

? "Try localhost:8080/one"
oServer.route(:Get,"/one",:one)
? "Try localhost:8080/two"
oServer.route(:Get,"/two",:two)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func one
        oServer.setContent("one", "text/plain")

func two
        oServer.setContent("two", "text/plain")
```

Example(3):

In this example we will use anonymous function

```
load "httplib.ring"

? "Try localhost:8080/hello"

oServer = new Server {

        route(:Get,"/hello",func {
                oServer.setContent("Hello, World!", "text/plain")
        })

        listen("0.0.0.0", 8080)

}
```

Example(4):

```
load "httplib.ring"

? "Try localhost:8080/hi - See output in console at Server-Side"
? "Try localhost:8080/hello - See output in web browser at Client-Side"
```

(continues on next page)

(continued from previous page)

```
oServer = new Server {

        route(:Get,"/hi",'? "Wow, I love Ring programming!"')
        route(:Get,"/hello",'oServer.setContent("Hello, World!", "text/plain")')

        listen("0.0.0.0", 8080)

}
```

Example(5):

```
load "httplib.ring"

new Client("localhost:8080") {
        ? download("/one")
        ? download("/two")
}
```

Tip: Using the Download() method in the InternetLib is faster

72.7 Using WebLib

Example(1):

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/report"
oServer.route(:Get,"/report",:report)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func report
        oPage = New HTMLPage
        {
                nRowsCount = 10
                title = "Report"
                h1 { text("Customers Report") }
                Table
                {
                        style = stylewidth("100%") + stylegradient(4)
                        TR
                        {
                                TD { WIDTH="10%" text("Customers Count : " )  }
```

(continues on next page)

(continued from previous page)

```
                                                        TD { text (nRowsCount) }
                        }
                }
                Table
                {
                        style = stylewidth("100%") + stylegradient(26)
                        TR
                        {
                                style = stylewidth("100%") + stylegradient(24)
                                TD { text("Name " )  }
                                TD { text("Age" ) }
                                TD { text("Country" ) }
                                TD { text("Job" ) }
                                TD { text("Company" ) }
                        }
                        for x =  1 to nRowsCount
                                TR
                                {
                                        TD { text("Test" )  }
                                        TD { text("30" ) }
                                        TD { text("Egypt" ) }
                                        TD { text("Sales" ) }
                                        TD { text("Future" ) }
                                }
                        next
                }
        }

        oServer.setHTMLPage(oPage)
```

72.8 Using HTTP Post

Example(1):

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/form"
oServer.route(:Get,"/form",:form)
oServer.route(:Post,"/formresponse",:formresponse)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func form
```

(continues on next page)

(continued from previous page)

```
        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Post Test")
                        newline()
                boxend()
                divstart([:style=StyleFloatLeft()+StyleWidth("100px") ])
                        newline()
                        text( "Number1 : " )    newline() newline()
                        text( "Number2 : " )    newline() newline()
                divend()
                formpost("formresponse")
                        divstart([ :style = styleFloatLeft()+StyleWidth("200px") ])
                                newline()
                                textbox([ :name = "Number1" ])  newline() newline()
                                textbox([ :name = "Number2" ])  newline() newline()
                                submit([ :value = "Send" ] )
                        divend()
                formend()
        }
        oServer.setHTMLPage(oPage)

func formresponse

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Post Result" )
                        newline()
                boxend()
                divstart([ :style = styleFloatLeft()+styleWidth("200px") ])
                        newline()
                        text( "Number1 : " + oServer["Number1"] )
                        newline() newline()
                        text( "Number2 : " + oServer["Number2"] )
                        newline() newline()
                        text( "Sum : " + (0 + oServer["Number1"] +
                                        oServer["Number2"] ) )
                        newline()
                divend()
        }

        oServer.setHTMLPage(oPage)
```

Example(2):

```
load "httplib.ring"
Load "openssllib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
```

(continues on next page)

(continued from previous page)

```
oServer = new Server

? "Try localhost:8080/hash"
oServer.route(:Get,"/hash",:hash)
oServer.route(:Post,"/hashresponse",:hashresponse)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func hash

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Hash Test")
                        newline()
                boxend()
                divstart([:style = StyleFloatLeft() + StyleWidth("100px") ])
                        newline()
                        text( "Value : " )
                        newline() newline()
                divend()
                formpost("/hashresponse")
                        divstart([:style = StyleFloatLeft() + StyleWidth("300px") ])
                                newline()
                                textbox([ :name = "Value" ])
                                newline() newline()
                                submit([ :value = "Send" ])
                        divend()
                formend()
        }
        oServer.setHTMLPage(oPage)

func hashresponse

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Hash Result" )
                        newline()
                boxend()
                divstart([:style = styleFloatLeft() + styleWidth("100%") ])
                        newline()
                        text( "Value : " + oServer["Value"] )
                        newline()
                        text( "MD5 : " + md5(oServer["Value"]) )
                        newline()
                        text( "SHA1 : " + SHA1(oServer["Value"]) )
                        newline()
                        text( "SHA256 : " + SHA256(oServer["Value"]) )
                        newline()
                        text( "SHA224 : " + SHA224(oServer["Value"]) )
```

(continues on next page)

(continued from previous page)

```
                        newline()
                        text( "SHA384 : " + SHA384(oServer["Value"]) )
                        newline()
                        text( "SHA512 : " + SHA512(oServer["Value"]) )
                        newline()
                divend()
        }

        oServer.setHTMLPage(oPage)
```

72.9 More Samples

Using Gradients:

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/gradient"
oServer.route(:Get,"/gradient",:gradient)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func gradient

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text("StyleGradient() Function")
                boxend()
                for x = 1 to 60
                        divstart([ :id = x , :align = "center" ,
                                   :style = stylefloatleft() +
                                                        stylesize(string(100/60*6)+"%",
↪"50px") +
                                                        stylegradient(x) ])
                                h3(x)
                        divend()
                next
        }
        oServer.setHTMLPage(oPage)
```

Using Lists:

```
load "httplib.ring"
load "weblib.ring"
```

(continues on next page)

(continued from previous page)

```
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/lists"
oServer.route(:Get,"/lists",:lists)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func lists

        oPage = New HTMLPageFunctions
        {
                ulstart([])
                        for x = 1 to 10
                                listart([])
                                        text(x)
                                liend()
                        next
                ulend()

                list2ul(["one","two","three","four","five"])

                ulstart([])
                        for x = 1 to 10
                                listart([])
                                        cFuncName = "btn"+x+"()"
                                        button([ :onclick = cFuncName , :value = x])
                                        script(scriptfuncalert(cFuncName,string(x)))
                                liend()
                        next
                ulend()
        }
        oServer.setHTMLPage(oPage)
```

Using Tables:

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/table"
oServer.route(:Get,"/table",:table)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)
```

(continues on next page)

(continued from previous page)

```
func table

        oPage = New HTMLPageFunctions
        {
                divstart([ :style = styledivcenter("400px","500px") ] )
                        style(styletable() + styletablerows("t01"))
                        tablestart([ :id = :t01 , :style = stylewidth("100%") ])
                                rowstart([])
                                        headerstart([]) text("Number") headerend()
                                        headerstart([]) text("square") headerend()
                                rowend()
                                for x = 1 to 10
                                        rowstart([])
                                                cellstart([]) text(x) cellend()
                                                cellstart([]) text(x*x) cellend()
                                        rowend()
                                next
                        tableend()
                divend()
        }
        oServer.setHTMLPage(oPage)
```

Play Video:

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/play"
oServer.route(:Get,"/play",:play)

? "We support files in the res folder like res/horse.ogg and res/movie.mp4"
oServer.shareFolder("res")

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func play

        oPage = New HTMLPage
        {
                Title = "Welcome"
                h1 { text("Play sound and video!") }
                div
                {
                        audio
                        {
                                src = "res/horse.ogg"
                                type = "audio/ogg"
```

(continues on next page)

(continued from previous page)

```
                        }

                        video
                        {
                                width = 320
                                height = 240
                                src = "res/movie.mp4"
                                type = "video/mp4"
                        }

                }
        }
        oServer.setHTMLPage(oPage)
```

72.10 Using Cookies

Example:

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/cookie"
oServer.route(:Get,"/cookie",:cookie)
oServer.route(:Get,"/cookieresponse",:cookieresponse)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func cookie

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Cookie Test" )
                        newline()
                boxend()
                link([ :url = "/cookieresponse", :title = "Use Cookies" ])
        }

        oServer.setCookie("programminglanguage=Ring")
        oServer.setCookie("library=HTTPLib")

        oServer.setHTMLPage(oPage)

func cookieresponse
```

(continues on next page)

(continued from previous page)

```
        aCookies = oServer.Cookies()

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Cookies Values" )
                        newline()
                boxend()
                link([ :url = "cookie", :title = "back" ])
                newline()
                divstart([:style="float:left;width:200px"])
                        text( "Programming Language : " + aCookies[:programminglanguage]␣
↪)
                        newline()
                        text( "Library : " + aCookies[:library] )
                        newline()
                divend()
        }
        oServer.setHTMLPage(oPage)
```

72.11 Uploading Files

Example:

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

cUploadFolder = "upload/"
oServer.shareFolder(cUploadFolder)

? "Try localhost:8080/upload"
oServer.route(:Get,"/upload",:upload)
oServer.route(:Post,"/uploadresponse",:uploadresponse)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func upload

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Upload File" )
                        newline()
                boxend()
                for x = 1 to 3 newline() next
```

(continues on next page)

(continued from previous page)

```
                formupload("/uploadresponse")
                        text( "Customer Name : " )
                        textbox([ :name = "custname" ])
                        newline() newline()
                        divstart([ :style = styleFloatLeft() + styleWidth("90px") ])
                                uploadfile("file1")  newline() newline()
                                uploadfile("file2") newline() newline()
                                submit([ :value = "Send" ])
                        divend()
                formend()
        }

        oServer.setHTMLPage(oPage)

func uploadresponse

        oPage = New HTMLPageFunctions
        {
                boxstart()
                        text( "Upload Result" )
                        newline()
                boxend()
                newline()
                divstart([ :style=  styleFloatLeft() + styleWidth("100px") ])
                        text( "Name : " + oServer["custname"] )
                        newline()
                divend()
                getuploadedfile(self,"file1")
                getuploadedfile(self,"file2")
        }

        oServer.setHTMLPage(oPage)


Func getUploadedFile oObj,cFile
        cNewFileName = oServer.getfilename(cFile)
        if cNewFileName = NULL return ok
        cNewFileContent = oServer.getFileContent(cFile)
        /*
                Here we use object.property instead of object { }
                To avoid executing braceend() method
        */
        cFileName = cUploadFolder + cNewFileName
        write(cFileName,cNewFileContent)
        if isLinux()
                system("chmod a+x "+cFileName)
        ok
        oObj.newline()
        oObj.text( "File "+cFileName+ " Uploaded ..." )
        oObj.newline()
        imageURL = cFileName
        oObj.link([ :url = imageURL, :title = "Download" ])
```

(continues on next page)

(continued from previous page)

```
        oObj.newline()
        oObj.image( [ :url = imageURL , :alt = :image  ] )
        oObj.newline()
```

72.12 Using Templates

Example:

```
load "httplib.ring"
load "weblib.ring"
import System.Web

? "Start the server..."
oServer = new Server

? "Try localhost:8080/template"
oServer.route(:Get,"/template"," new numbersController { start() } ")

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

class numbersController

        MyHeader aNumbers

        func Start

                MyHeader = New Header
                {
                        cColumn1 = "Number" cColumn2 = "Square"
                }

                aNumbers = list(20)

                for x = 1 to len(aNumbers)
                        aNumbers[x] = new number
                        {
                                nValue  = x   nSquare = x*x
                        }
                next

                cTemp = Template("templates/mynumbers.html",self)

                oPage = new HTMLPageFunctions
                {
                        boxstart()
                                text( "Test Templates" )
                                newline()
                        boxend()
                        html(cTemp)
```

(continues on next page)

(continued from previous page)

```
                }

                oServer.setHTMLPage(oPage)

Class Header cColumn1 cColumn2

Class Number nValue   nSquare
```

72.13 Regular Expressions

Example:

```
load "httplib.ring"

? "Start the server..."
oServer = new Server

? "Try localhost:8080/numbers/<number>"
? "Example: localhost:8080/numbers/123"
oServer.route(:Get,"(/numbers/(\d+))",:mytest)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func mytest
        cOutput = "Match(1): " + oServer.Match(1) + nl
        cOutput += "Match(2): " + oServer.Match(2) + nl
        oServer.setContent(cOutput, "text/plain")
```

72.14 Stop the Server

Example:

```
load "httplib.ring"

? "Start the server..."
oServer = new Server

? "Try localhost:8080/time"
? "Try localhost:8080/stop"
oServer.route(:Get,"/time",:gettime)
oServer.route(:Get,"/stop",:stop)

? "Listen to port 8080"
oServer.listen("0.0.0.0", 8080)

func gettime
        oServer.setContent("Time: " + time(), "text/plain")
```

(continues on next page)

(continued from previous page)

```
func stop
        oServer.stop()
```

CHAPTER

SEVENTYTHREE

USING TOKENSLIB

In this chapter we will learn how to use the TokensLib library.

73.1 Introduction

TokensLib is a simple library written in Ring.

The library provide classes to get Ring tokens from Ring source code.

73.2 RingTokens Class

The class comes with the next methods:

```
fromFile(cFileName)
fromString(cString)
getTokens() --> aTokensList
printTokens()
```

73.3 Example

```
load "tokenslib.ring"

func main
        oTokens = new RingTokens {
                fromFile("hello.ring")
                PrintTokens()
                ? Copy("=",50)
                fromString("? 1+1")
                PrintTokens()
        }
```

Output:

```
Keyword     : SEE
Literal     : Hello, World!
EndLine
```

(continues on next page)

(continued from previous page)

```
==================================================
Operator    : ? (23)
Number      : 1
Operator    : + (1)
Number      : 1
EndLine
```

73.4 Constants

When loading the library, We have access to the next constants

```
# Tokens Table
        C_TOKENTYPE     = 1
        C_TOKENVALUE    = 2
        C_TOKENINDEX    = 3

# Token Type
        C_KEYWORD       = 0
        C_OPERATOR      = 1
        C_LITERAL       = 2
        C_NUMBER        = 3
        C_IDENTIFIER    = 4
        C_ENDLINE       = 5

# Keywords List
        aKEYWORDS = [
                "IF","TO","OR","AND","NOT","FOR","NEW","FUNC",
                "FROM","NEXT","LOAD","ELSE","SEE","WHILE","OK",
                "CLASS","RETURN","BUT",
                "END","GIVE","BYE","EXIT","TRY","CATCH","DONE",
                "SWITCH","ON","OTHER","OFF",
                "IN","LOOP","PACKAGE","IMPORT","PRIVATE","STEP","DO",
                "AGAIN","CALL","ELSEIF",
                "PUT","GET","CASE","DEF","ENDFUNC","ENDCLASS","ENDPACKAGE",
                "CHANGERINGKEYWORD","CHANGERINGOPERATOR","LOADSYNTAX"
        ]

# Keywords Constants
        K_IF                    = 1
        K_TO                    = 2
        K_OR                    = 3
        K_AND                   = 4
        K_NOT                   = 5
        K_FOR                   = 6
        K_NEW                   = 7
        K_FUNC                  = 8
        K_FROM                  = 9
        K_NEXT                  = 10
        K_LOAD                  = 11
        K_ELSE                  = 12
```

(continues on next page)

(continued from previous page)

```
        K_SEE                   = 13
        K_WHILE                 = 14
        K_OK                    = 15
        K_CLASS                 = 16
        K_RETURN                = 17
        K_BUT                   = 18
        K_END                   = 19
        K_GIVE                  = 20
        K_BYE                   = 21
        K_EXIT                  = 22
        K_TRY                   = 23
        K_CATCH                 = 24
        K_DONE                  = 25
        K_SWITCH                = 26
        K_ON                    = 27
        K_OTHER                 = 28
        K_OFF                   = 29
        K_IN                    = 30
        K_LOOP                  = 31
        K_PACKAGE               = 32
        K_IMPORT                = 33
        K_PRIVATE               = 34
        K_STEP                  = 35
        K_DO                    = 36
        K_AGAIN                 = 37
        K_CALL                  = 38
        K_ELSEIF                = 39
        K_PUT                   = 40
        K_GET                   = 41
        K_CASE                  = 42
        K_DEF                   = 43
        K_ENDFUNC               = 44
        K_ENDCLASS              = 45
        K_ENDPACKAGE            = 46
        K_CHANGERINGKEYWORD     = 47
        K_CHANGERINGOPERATOR    = 48
        K_LOADSYNTAX            = 49

# Operators

        OP_PLUS         = 1
        OP_MINUS        = 2
        OP_MUL          = 3
        OP_DIV          = 4
        OP_REM          = 5
        OP_DOT          = 6
        OP_FOPEN        = 7
        OP_FCLOSE       = 8
        OP_EQUAL        = 9
        OP_COMMA        = 10
        OP_NOT          = 11
        OP_GREATER      = 12
```

(continues on next page)

(continued from previous page)

```
        OP_LESS         = 13
        OP_LOPEN        = 14
        OP_LCLOSE       = 15
        OP_RANGE        = 16
        OP_BRACEOPEN    = 17
        OP_BRACECLOSE   = 18
        OP_BITAND       = 19
        OP_BITOR        = 20
        OP_BITNOT       = 21
        OP_XOR          = 22
```

CHAPTER

SEVENTYFOUR

USING RINGLIBCURL

In this chapter we will learn about using RingLibCurl

74.1 Get Request

Example:

```
load "libcurl.ring"

curl = curl_easy_init()

curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)
curl_easy_setopt(curl, CURLOPT_URL, "http://ring-lang.sf.net")

curl_easy_perform(curl)

curl_easy_cleanup(curl)
```

74.2 Post Request

Example:

```
load "libcurl.ring"

curl = curl_easy_init()

cPostThis = "page=4&Number1=4&Number2=5"
curl_easy_setopt(curl, CURLOPT_URL, "http://localhost/ringapp/index.ring?page=3")
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, cPostThis)

curl_easy_perform(curl)

curl_easy_cleanup(curl)
```

74.3 Facebook Login

Example:

```
load "libcurl.ring"

see "Enter Email : " give $login_email
See "Enter Password : " give $login_pass

curl = curl_easy_init()

curl_easy_setopt(curl, CURLOPT_URL, 'https://www.facebook.com/login.php')
curl_easy_setopt(curl, CURLOPT_POSTFIELDS,'charset_test=j u s t a t e s t'+
' &email='+urlencode($login_email)+'&pass='+
urlencode($login_pass)+'&login=Login')
curl_easy_setopt(curl, CURLOPT_POST, 1)
curl_easy_setopt(curl, CURLOPT_HEADER, 0)
curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)
curl_easy_setopt(curl, CURLOPT_COOKIEJAR, "cookies.txt")
curl_easy_setopt(curl, CURLOPT_COOKIEFILE, "cookies.txt")
curl_easy_setopt(curl, CURLOPT_USERAGENT, "Mozilla/5.0 (Windows; U;"+
" Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3")
curl_easy_setopt(curl, CURLOPT_REFERER, "http://www.facebook.com")
curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, FALSE)
curl_easy_setopt(curl, CURLOPT_SSL_VERIFYHOST, 2)

mylist = curl_slist_append(NULL,'Accept-Charset: utf-8')
curl_slist_append(mylist,'Accept-Language: en-us,en;q=0.7,bn-bd;q=0.3')
curl_slist_append(mylist,'Accept: text/xml,application/xml,'+
'application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5')
curl_easy_setopt(curl, CURLOPT_HTTPHEADER, mylist)

curl_easy_setopt(curl, CURLOPT_COOKIESESSION, false)

curl_easy_perform(curl)

curl_easy_cleanup(curl)

Func URLEncode cStr
        cOut = ""
        for x in cStr
                if isalnum(x)
                        cOut += x
                but x = " "
                        cOut += "+"
                else
                        cOut += "%"+str2hex(x)
                ok
        next
        return cOut
```

74.4 Save Output to String

Example:

```
load "libcurl.ring"

curl = curl_easy_init()

curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)
curl_easy_setopt(curl, CURLOPT_URL, "http://ring-lang.sf.net")

cOutput = curl_easy_perform_silent(curl)

See "Output:" + nl
see cOutput

curl_easy_cleanup(curl)
```

74.5 Get Stock Data From Yahoo

Example:

```
Load "libcurl.ring"

### Part 1 --- Get Crumb and Cookie ----------------------------------------

See "Start curl_easy_init(): "+ nl
curl = curl_easy_init()                     ### >>> HANDLE >>> 01006BD0  CURL  0

        curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1)
        curl_easy_setopt(curl, CURLOPT_COOKIEJAR,  "cookies.txt")
        curl_easy_setopt(curl, CURLOPT_COOKIEFILE, "cookies.txt")
        curl_easy_setopt(curl, CURLOPT_URL, "https://finance.yahoo.com/quote/AMZN/history
↪")

        ###  HTML Data >>> STDOUT Window,  Use curl_easy_perform_silent >>> String

cOutput = curl_easy_perform_silent(curl)    ### GO Get Data >>> String


###   Extract Crumb from data
###  "CrumbStore":{"crumb":"abcdefghijk"},

if cOutput != NULL

        newStr1     = substr(cOutput, substr(cOutput, '"CrumbStore":{"crumb":"' ), 48 )
                nPosS   = substr(newStr1, ':"' ) ;  ### Start of crumb -2
                nPosE   = substr(newStr1, '"}' ) ;  ### End   of crumb
                nCount  = nPosE - nPosS -2          ### size  of crumb
        myCrumb     = substr(newStr1, nPosS +2, nCount)
```

(continues on next page)

(continued from previous page)

```
        See "myCrumb.: |"+ myCrumb +"|" +nl

        ### UniCode "\u002F" replace it with "/"
                if substr( myCrumb, "\u002F")
                   myCrumb = substr( myCrumb, "\u002F", "/")
                   See "myCrumb2: |"+ myCrumb +"|"+ nl
                ok

else
        See "No Connectivity to Yahoo. Looking for Cookie and Crumb." +nl +nl
ok


### Part 2 --- Send URL with Crumb, and Cookie ----------------------------------------

        ### Send URL+Crumb to Yahoo to fetch 1st stock history data,

        $url = "https://query1.finance.yahoo.com/v7/finance/download/AMZN"+
                "?period1=1277856000&period2=1498777545&interval=1wk" +
                "&events=history&crumb=" + myCrumb

        curl_easy_setopt(curl, CURLOPT_URL, $url);
        cStr = curl_easy_perform_silent(curl)
        See cStr

curl_easy_cleanup(curl)  ### REMEMBER to CLOSE  CURL
```

Output:

```
myCrumb.: |sEEeW97mxvN|
Date,Open,High,Low,Close,Adj Close,Volume
2010-07-05,110.650002,117.480003,109.000000,117.260002,117.260002,21000400
2010-07-12,117.809998,124.879997,117.320000,118.489998,118.489998,29407300
2010-07-19,118.379997,121.250000,105.800003,118.870003,118.870003,74252100
```

CHAPTER

SEVENTYFIVE

RINGLIBCURL FUNCTIONS REFERENCE

75.1 Introduction

In this chapter we have a list of the supported functions by this extension

75.2 Reference

- CURL_GLOBAL_ALL
- CURL_GLOBAL_SSL
- CURL_GLOBAL_WIN32
- CURL_GLOBAL_NOTHING
- CURL_GLOBAL_DEFAULT
- CURL_GLOBAL_ACK_EINTR
- CURLOPT_VERBOSE
- CURLOPT_HEADER
- CURLOPT_NOPROGRESS
- CURLOPT_NOSIGNAL
- CURLOPT_WILDCARDMATCH
- CURLOPT_WRITEFUNCTION
- CURLOPT_WRITEDATA
- CURLOPT_READFUNCTION
- CURLOPT_READDATA
- CURLOPT_IOCTLFUNCTION
- CURLOPT_IOCTLDATA
- CURLOPT_SEEKFUNCTION
- CURLOPT_SEEKDATA
- CURLOPT_SOCKOPTFUNCTION
- CURLOPT_SOCKOPTDATA
- CURLOPT_OPENSOCKETFUNCTION

- CURLOPT_OPENSOCKETDATA
- CURLOPT_CLOSESOCKETFUNCTION
- CURLOPT_CLOSESOCKETDATA
- CURLOPT_PROGRESSFUNCTION
- CURLOPT_PROGRESSDATA
- CURLOPT_HEADERFUNCTION
- CURLOPT_HEADERDATA
- CURLOPT_DEBUGFUNCTION
- CURLOPT_DEBUGDATA
- CURLOPT_SSL_CTX_FUNCTION
- CURLOPT_SSL_CTX_DATA
- CURLOPT_CONV_TO_NETWORK_FUNCTION
- CURLOPT_CONV_FROM_NETWORK_FUNCTION
- CURLOPT_CONV_FROM_UTF8_FUNCTION
- CURLOPT_INTERLEAVEFUNCTION
- CURLOPT_INTERLEAVEDATA
- CURLOPT_CHUNK_BGN_FUNCTION
- CURLOPT_CHUNK_END_FUNCTION
- CURLOPT_CHUNK_DATA
- CURLOPT_FNMATCH_FUNCTION
- CURLOPT_FNMATCH_DATA
- CURLOPT_ERRORBUFFER
- CURLOPT_STDERR
- CURLOPT_FAILONERROR
- CURLOPT_URL
- CURLOPT_PROTOCOLS
- CURLOPT_REDIR_PROTOCOLS
- CURLOPT_PROXY
- CURLOPT_PROXYPORT
- CURLOPT_PROXYTYPE
- CURLOPT_NOPROXY
- CURLOPT_HTTPPROXYTUNNEL
- CURLOPT_SOCKS5_GSSAPI_SERVICE
- CURLOPT_SOCKS5_GSSAPI_NEC
- CURLOPT_INTERFACE
- CURLOPT_LOCALPORT

- CURLOPT_LOCALPORTRANGE
- CURLOPT_DNS_CACHE_TIMEOUT
- CURLOPT_DNS_USE_GLOBAL_CACHE
- CURLOPT_BUFFERSIZE
- CURLOPT_PORT
- CURLOPT_TCP_NODELAY
- CURLOPT_ADDRESS_SCOPE
- CURLOPT_NETRC
- CURLOPT_NETRC_FILE
- CURLOPT_USERPWD
- CURLOPT_PROXYUSERPWD
- CURLOPT_USERNAME
- CURLOPT_PASSWORD
- CURLOPT_PROXYUSERNAME
- CURLOPT_PROXYPASSWORD
- CURLOPT_HTTPAUTH
- CURLOPT_TLSAUTH_USERNAME
- CURLOPT_TLSAUTH_PASSWORD
- CURLOPT_TLSAUTH_TYPE
- CURLOPT_PROXYAUTH
- CURLOPT_AUTOREFERER
- CURLOPT_ACCEPT_ENCODING
- CURLOPT_TRANSFER_ENCODING
- CURLOPT_FOLLOWLOCATION
- CURLOPT_UNRESTRICTED_AUTH
- CURLOPT_MAXREDIRS
- CURLOPT_POSTREDIR
- CURLOPT_PUT
- CURLOPT_POST
- CURLOPT_POSTFIELDS
- CURLOPT_POSTFIELDSIZE
- CURLOPT_POSTFIELDSIZE_LARGE
- CURLOPT_COPYPOSTFIELDS
- CURLOPT_HTTPPOST
- CURLOPT_REFERER
- CURLOPT_USERAGENT

- CURLOPT_HTTPHEADER
- CURLOPT_HTTP200ALIASES
- CURLOPT_COOKIE
- CURLOPT_COOKIEFILE
- CURLOPT_COOKIEJAR
- CURLOPT_COOKIESESSION
- CURLOPT_COOKIELIST
- CURLOPT_HTTPGET
- CURLOPT_HTTP_VERSION
- CURLOPT_IGNORE_CONTENT_LENGTH
- CURLOPT_HTTP_CONTENT_DECODING
- CURLOPT_HTTP_TRANSFER_DECODING
- CURLOPT_MAIL_FROM
- CURLOPT_MAIL_RCPT
- CURLOPT_TFTP_BLKSIZE
- CURLOPT_FTPPORT
- CURLOPT_QUOTE
- CURLOPT_POSTQUOTE
- CURLOPT_PREQUOTE
- CURLOPT_APPEND
- CURLOPT_FTP_USE_EPRT
- CURLOPT_FTP_USE_EPSV
- CURLOPT_FTP_USE_PRET
- CURLOPT_FTP_CREATE_MISSING_DIRS
- CURLOPT_FTP_RESPONSE_TIMEOUT
- CURLOPT_FTP_ALTERNATIVE_TO_USER
- CURLOPT_FTP_SKIP_PASV_IP
- CURLOPT_FTPSSLAUTH
- CURLOPT_FTP_SSL_CCC
- CURLOPT_FTP_ACCOUNT
- CURLOPT_FTP_FILEMETHOD
- CURLOPT_RTSP_REQUEST
- CURLOPT_RTSP_SESSION_ID
- CURLOPT_RTSP_STREAM_URI
- CURLOPT_RTSP_TRANSPORT
- CURLOPT_RTSP_CLIENT_CSEQ

- CURLOPT_RTSP_SERVER_CSEQ
- CURLOPT_TRANSFERTEXT
- CURLOPT_PROXY_TRANSFER_MODE
- CURLOPT_CRLF
- CURLOPT_RANGE
- CURLOPT_RESUME_FROM
- CURLOPT_RESUME_FROM_LARGE
- CURLOPT_CUSTOMREQUEST
- CURLOPT_FILETIME
- CURLOPT_DIRLISTONLY
- CURLOPT_NOBODY
- CURLOPT_INFILESIZE
- CURLOPT_INFILESIZE_LARGE
- CURLOPT_UPLOAD
- CURLOPT_MAXFILESIZE
- CURLOPT_MAXFILESIZE_LARGE
- CURLOPT_TIMECONDITION
- CURLOPT_TIMEVALUE
- CURLOPT_TIMEOUT
- CURLOPT_TIMEOUT_MS
- CURLOPT_LOW_SPEED_LIMIT
- CURLOPT_LOW_SPEED_TIME
- CURLOPT_MAX_SEND_SPEED_LARGE
- CURLOPT_MAX_RECV_SPEED_LARGE
- CURLOPT_MAXCONNECTS
- CURLOPT_FRESH_CONNECT
- CURLOPT_FORBID_REUSE
- CURLOPT_CONNECTTIMEOUT
- CURLOPT_CONNECTTIMEOUT_MS
- CURLOPT_IPRESOLVE
- CURLOPT_CONNECT_ONLY
- CURLOPT_USE_SSL
- CURLOPT_RESOLVE
- CURLOPT_SSLCERT
- CURLOPT_SSLCERTTYPE
- CURLOPT_SSLKEY

- CURLOPT_SSLKEYTYPE
- CURLOPT_KEYPASSWD
- CURLOPT_SSLENGINE
- CURLOPT_SSLENGINE_DEFAULT
- CURLOPT_SSLVERSION
- CURLOPT_SSL_VERIFYHOST
- CURLOPT_SSL_VERIFYPEER
- CURLOPT_CAINFO
- CURLOPT_ISSUERCERT
- CURLOPT_CAPATH
- CURLOPT_CRLFILE
- CURLOPT_CERTINFO
- CURLOPT_RANDOM_FILE
- CURLOPT_EGDSOCKET
- CURLOPT_SSL_CIPHER_LIST
- CURLOPT_SSL_SESSIONID_CACHE
- CURLOPT_KRBLEVEL
- CURLOPT_GSSAPI_DELEGATION
- CURLOPT_SSH_AUTH_TYPES
- CURLOPT_SSH_HOST_PUBLIC_KEY_MD5
- CURLOPT_SSH_PUBLIC_KEYFILE
- CURLOPT_SSH_PRIVATE_KEYFILE
- CURLOPT_SSH_KNOWNHOSTS
- CURLOPT_SSH_KEYFUNCTION
- CURLOPT_SSH_KEYDATA
- CURLOPT_PRIVATE
- CURLOPT_SHARE
- CURLOPT_NEW_FILE_PERMS
- CURLOPT_NEW_DIRECTORY_PERMS
- CURLOPT_TELNETOPTIONS
- CURLE_OK
- CURLE_UNKNOWN_OPTION
- CURLE_NOT_BUILT_IN
- CURLINFO_EFFECTIVE_URL
- CURLINFO_RESPONSE_CODE
- CURLINFO_HTTP_CONNECTCODE

- CURLINFO_FILETIME
- CURLINFO_TOTAL_TIME
- CURLINFO_NAMELOOKUP_TIME
- CURLINFO_CONNECT_TIME
- CURLINFO_APPCONNECT_TIME
- CURLINFO_PRETRANSFER_TIME
- CURLINFO_STARTTRANSFER_TIME
- CURLINFO_REDIRECT_TIME
- CURLINFO_REDIRECT_COUNT
- CURLINFO_REDIRECT_URL
- CURLINFO_SIZE_UPLOAD
- CURLINFO_SIZE_DOWNLOAD
- CURLINFO_SPEED_DOWNLOAD
- CURLINFO_SPEED_UPLOAD
- CURLINFO_HEADER_SIZE
- CURLINFO_REQUEST_SIZE
- CURLINFO_SSL_VERIFYRESULT
- CURLINFO_SSL_ENGINES
- CURLINFO_CONTENT_LENGTH_DOWNLOAD
- CURLINFO_CONTENT_LENGTH_UPLOAD
- CURLINFO_CONTENT_TYPE
- CURLINFO_PRIVATE
- CURLINFO_HTTPAUTH_AVAIL
- CURLINFO_PROXYAUTH_AVAIL
- CURLINFO_OS_ERRNO
- CURLINFO_NUM_CONNECTS
- CURLINFO_PRIMARY_IP
- CURLINFO_PRIMARY_PORT
- CURLINFO_LOCAL_IP
- CURLINFO_LOCAL_PORT
- CURLINFO_COOKIELIST
- CURLINFO_LASTSOCKET
- CURLINFO_FTP_ENTRY_PATH
- CURLINFO_CERTINFO
- CURLINFO_CONDITION_UNMET
- CURLINFO_RTSP_SESSION_ID

- CURLINFO_RTSP_CLIENT_CSEQ
- CURLINFO_RTSP_SERVER_CSEQ
- CURLINFO_RTSP_CSEQ_RECV
- CURLFORM_COPYNAME
- CURLFORM_PTRNAME
- CURLFORM_COPYCONTENTS
- CURLFORM_PTRCONTENTS
- CURLFORM_CONTENTSLENGTH
- CURLFORM_FILECONTENT
- CURLFORM_FILE
- CURLFORM_CONTENTTYPE
- CURLFORM_FILENAME
- CURLFORM_BUFFER
- CURLFORM_BUFFERPTR
- CURLFORM_BUFFERLENGTH
- CURLFORM_STREAM
- CURLFORM_ARRAY
- CURLFORM_CONTENTHEADER
- CURLcode curl_global_init(long flags)
- CURL *curl_easy_init(void)
- void curl_easy_cleanup(CURL * handle)
- CURLcode curl_easy_setopt_1(CURL *handle, CURLoption option, int)
- CURLcode curl_easy_setopt_2(CURL *handle, CURLoption option, const char *)
- CURLcode curl_easy_setopt_3(CURL *handle, CURLoption option, void *)
- CURLcode curl_easy_setopt_4(CURL *handle, CURLoption option, CURLLIST *)
- CURLcode curl_easy_perform(CURL * easy_handle)
- String *curl_easy_perform_silent(CURL * easy_handle)
- CURLcode curl_easy_getinfo_1(CURL *handle, CURLINFO info, char **urlp)
- CURLcode curl_easy_getinfo_2(CURL *handle, CURLINFO info, long *codep)
- CURLcode curl_easy_getinfo_3(CURL *handle, CURLINFO info, double *timep)
- CURLcode curl_easy_getinfo_4(CURL *handle, CURLINFO info, CURLLIST **engine_list)
- CURLcode curl_easy_getinfo_5(CURL *handle, CURLINFO info, struct curl_certinfo *chainp)
- CURLcode curl_easy_getinfo_6(CURL *handle, CURLINFO info, struct curl_tlssessioninfo **session)
- char *curl_version(void)
- time_t curl_getdate(char * datestring , time_t *now)

- CURLFORMcode curl_formadd_1(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, const char *, CURLformoption)
- CURLFORMcode curl_formadd_2(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, const char *,CURLformoption, const char *, CURLformoption)
- CURLFORMcode curl_formadd_3(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, void *, CURLformoption)
- CURLFORMcode curl_formadd_4(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, void *,CURLformoption, long , CURLformoption)
- CURLFORMcode curl_formadd_5(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, void *,CURLformoption, long , CURLformoption, const char* , CURLformoption)
- CURLFORMcode curl_formadd_6(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, const char *,CURLformoption, void * , CURLformoption, long , CURLformoption)
- CURLFORMcode curl_formadd_7(struct curl_httppost **firstitem, struct curl_httppost **lastitem, CURLformoption, const char *, CURLformoption, struct curl_forms [], CURLformoption)
- void curl_formfree(struct curl_httppost * form)
- CURLLIST *curl_slist_append(CURLLIST * list, const char * string)
- void curl_slist_free_all(CURLLIST * list)
- char *curl_easy_escape(CURL * curl , const char * string , int length)
- char *curl_easy_unescape(CURL * curl , const char * url , int inlength , int * outlength)

CHAPTER

SEVENTYSIX

RINGSOCKETS EXTENSION

In this chapter we will learn about using the RingSockets extension.

76.1 TCP Server

Example (TCP Server Code):

```
load "sockets.ring"

sock = socket(AF_INET,SOCK_STREAM,0)
bind(sock,"127.0.0.1",5050)
listen(sock,5)
ns = accept(sock)

send(ns,"Hello Client")
msg = recv(ns,1024)
? "Client Say >> " + msg
close(ns)
close(sock)
? "socket connection closed"
```

76.2 TCP Client

Example (TCP Client Code):

```
load "sockets.ring"

sock = socket(AF_INET,SOCK_STREAM)
connect(sock,"127.0.0.1",5050)

send(sock,"Hello Server")
msg = recv(sock,1024)
? "Server Say >> " + msg

close(sock)
? "socket connection closed"
```

76.3 Functions

The next functions are provided by this extension

```
socket(nAddressFamily, nConnectionType) -> SocketHandle
bind(pSocketHandle, cHost , nPort)
listen(pSocketHandle, nBacklog)
accept(pSocketHandle) -> pConnectionHandle
send(pConnectionHandle|pSocketHandle,cMessage)
sendto(pSocketHandle,cMessage)
recv(pConnectionHandle|pSocketHandle,nBuffer) -> cData
recvfrom(pSocketHandle, nBuffer) -> cData
connect(pSocketHandle, cHost , nPort) -> return <0> if successful
close(pSocketHandle|pConnectionHandle)
gethostbyname(cHostName) -> cIPAddress
gethostbyaddr(cIPAddress) -> aListOfHostInfo
gethostname() -> cHostName
getservbyname(cName) -> nPort
getservbyport(nPort) -> cName
inet_pton(AdressFamily, IP) ->  packed_address
inet_ntop(AdressFamily, packed_address) -> IP
socketsCleanup()
```

Example:

```
load "sockets.ring"

host = gethostbyname("google.com")
? host
line()
? gethostbyaddr(host)
line()
? gethostname()
line()
? getservbyname("ftp")
line()
? getservbyport(21)

func line ? copy("=",30)
```

76.4 Constants

Address Family:

```
AF_INET       #  mean use IPV4
AF_INET6      #  ........ IPV6
```

Connection type:

```
SOCK_STREAM   #  mean use TCP Protocol
SOCK_DGRAM    #  ........ UDP ........
```

CHAPTER

SEVENTYSEVEN

THREADS EXTENSION

In this chapter we will learn about using the Threads extension.

Supported platforms: Windows (32bit), Linux & macOS

If you are developing for Windows (64bit), Use threads from other extensions like GameLib (RingAllegro) or RingLibuv.

77.1 Creating Threads

Example (1):

```
load "threads.ring"

func main

        t = new_thrd_t()
        thrd_create(t,"Hello()")
        res = 0
        thrd_join(t, :res)

func Hello

        ? "Message from the Hello() function"
```

77.2 More Examples

Example (2):

```
load "threads.ring"

func main

        t = new_thrd_t()
        thrd_create(t,"Hello()")
        res = 0
        thrd_join(t, :res)

        t2 = new_thrd_t()
```

(continues on next page)

(continued from previous page)

```
        thrd_create(t2,"Hello2()")
        res2 = 0
        thrd_join(t2, :res2)

        ? :done

func Hello

        ? "Message from the Hello() function"

func Hello2

        ? "Message from the Hello2() function"
```

Example (3):

```
load "threads.ring"

func main

        nThreads = 2
        aList = list(nThreads)

        for x=1 to nThreads
                aList[x] = new_thrd_t()
                thrd_create(aList[x],"Hello("+x+")")
        next

        for x=1 to nThreads
                res= 0
                thrd_join(aList[x],:res)
        next

        ? :Done

        shutdown()

func Hello x

        for r=1 to 100
                ? "Message from the Hello("+x+") function"
        next
```

Example (4):

```
load "threads.ring"

func main

        nThreads = 10
        aList = list(nThreads)
```

(continues on next page)

(continued from previous page)

```
        for x=1 to nThreads
                aList[x] = new_thrd_t()
                thrd_create(aList[x],"Hello("+x+")")
        next

        for k=1 to nThreads
                 ? "*** Before join: " + k + " *** "
                 res = 0
                 thrd_join(aList[K],:res)
                 ? "*** After join: " + k + " *** "
        next

        ? " ===== Done ========== "

func Hello x

        for r=1 to 100
                ? "("+r+") Message from the Hello("+x+") function"
        next
```

77.3 Reference

Constants:

- TIME_UTC
- TINYCTHREAD_VERSION_MAJOR
- TINYCTHREAD_VERSION_MINOR
- TINYCTHREAD_VERSION
- thrd_error
- thrd_success
- thrd_timedout
- thrd_busy
- thrd_nomem
- mtx_plain
- mtx_timed
- mtx_recursive

Functions:

- int mtx_init(mtx_t *mtx, int type)
- void mtx_destroy(mtx_t *mtx)
- int mtx_lock(mtx_t *mtx)
- int mtx_timedlock(mtx_t *mtx, const struct timespec *ts)
- int mtx_trylock(mtx_t *mtx)

- int mtx_unlock(mtx_t *mtx)
- int cnd_init(cnd_t *cond)
- void cnd_destroy(cnd_t *cond)
- int cnd_signal(cnd_t *cond)
- int cnd_broadcast(cnd_t *cond)
- int cnd_wait(cnd_t *cond, mtx_t *mtx)
- int cnd_timedwait(cnd_t *cond, mtx_t *mtx, const struct timespec *ts)
- int thrd_create(thrd_t *thr, const char *cEvent)
- thrd_t thrd_current(void)
- int thrd_detach(thrd_t thr)
- int thrd_equal(thrd_t thr0, thrd_t thr1)
- void thrd_exit(int res)
- int thrd_join(thrd_t thr, int *res)
- int thrd_sleep(const struct timespec *duration, struct timespec *remaining)
- void thrd_yield(void)
- void tss_delete(tss_t key)
- void *tss_get(tss_t key)
- int tss_set(tss_t key, void *val)

CHAPTER

SEVENTYEIGHT

RINGLIBUI EXTENSION

In this chapter we will learn about using the RingLibUI extension.

This extension provides complete support for Libui

Using this extension we can develop and distribute lightweight GUI Applications using Ring (Less than 1 MB)

Runtime files and their size (For Ring 1.14)

- Ring.dll (448 KB)
- Libui.dll (210 KB)
- Ring_Libui.dll (633 KB)
- Total : 1,291 KB without compressing the files
- After compressing the files (To ZIP file) - Total : 504 KB

78.1 Hello World

```
load "libui.ring"

oWindow = uiNewWindow( "Hello, World", 400, 400, True)
uiWindowOnClosing(oWindow,"closeApp()")

btn1 = uiNewButton("SayHello")
uiButtonOnClicked(btn1,"sayHello()")

btn2 = uiNewButton("Close")
uiButtonOnClicked(btn2,"closeApp()")

g = uiNewGrid() uiGridSetPadded(g, 1) uiWindowSetChild(oWindow, g)
uiGridAppend(g, btn1, 0, 0, 2, 1, 1, uiAlignFill, 0, uiAlignFill)
uiGridAppend(g, btn2, 0, 1, 1, 1, 1, uiAlignFill, 0, uiAlignFill)

uiControlShow( oWindow )
uiMain()

func sayHello
        uiMsgBox(oWindow,"Hi","Hello")

func closeApp
        uiQuit()
```

Screen Shots:

78.2 Say Hello

```
load "libui.ring"

oWindow = uiNewWindow( "Say Hello", 500, 80, True)
uiWindowOnClosing(oWindow,"closeApp()")

lbl1 = uiNewLabel("Name: ")
text1 = uiNewEntry()

btn1 = uiNewButton("SayHello")
uiButtonOnClicked(btn1,"sayHello()")

btn2 = uiNewButton("Close")
uiButtonOnClicked(btn2,"closeApp()")

lbl2 = uiNewLabel("")

g = uiNewGrid() uiGridSetPadded(g, 1) uiWindowSetChild(oWindow, g)
uiGridAppend(g, lbl1, 0, 0, 2, 1, 1, uiAlignCenter, 0, uiAlignCenter)
uiGridAppend(g, text1, 1, 0, 2, 1, 1, uiAlignFill, 0, uiAlignFill)
uiGridAppend(g, btn1, 0, 1, 1, 2, 1, uiAlignFill, 0, uiAlignFill)
uiGridAppend(g, btn2, 2, 1, 1, 1, 1, uiAlignFill, 0, uiAlignFill)
```

(continues on next page)

(continued from previous page)

```
uiGridAppend(g, lbl2, 0, 3, 2, 1, 1, uiAlignCenter, 0, uiAlignCenter)

uiControlShow( oWindow )
uiMain()

func sayHello
        uiLabelSetText(lbl2,"Hello " + uiEntryText(text1))

func closeApp
        uiQuit()
```

Screen Shot:

78.3 Control Gallery

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

# Global Variables

        slider = NULL
        spinbox = NULL
        pBar = NULL
```

(continues on next page)

(continued from previous page)

```
        entryOpen = NULL
        entrySave = NULL

# Main Window

        mainwin = uiNewWindow("libui Control Gallery", 640, 480, 1)
        uiWindowOnClosing(mainwin, "onClosing()")

        tab = uiNewTab()
        uiWindowSetChild(mainwin, tab)
        uiWindowSetMargined(mainwin, 1)

        uiTabAppend(tab, "Basic Controls", makeBasicControlsPage())
        uiTabSetMargined(tab, 0, 1)

        uiTabAppend(tab, "Numbers and Lists", makeNumbersPage())
        uiTabSetMargined(tab, 1, 1)

        uiTabAppend(tab, "Data Choosers", makeDataChoosersPage())
        uiTabSetMargined(tab, 2, 1)

        uiControlShow(mainwin)
        uiMain()

func onClosing
        uiQuit()

func makeDataChoosersPage

        hbox = uiNewHorizontalBox()
        uiBoxSetPadded(hbox, 1)

        vbox = uiNewVerticalBox()
        uiBoxSetPadded(vbox, 1)
        uiBoxAppend(hbox, vbox, 0)

        uiBoxAppend(vbox,
                uiNewDatePicker(),
                0)
        uiBoxAppend(vbox,
                uiNewTimePicker(),
                0)
        uiBoxAppend(vbox,
                uiNewDateTimePicker(),
                0)

        uiBoxAppend(vbox,
                uiNewFontButton(),
                0)
        uiBoxAppend(vbox,
                uiNewColorButton(),
                0)
```

(continues on next page)

(continued from previous page)

```
uiBoxAppend(hbox,
        uiNewVerticalSeparator(),
        0)

vbox = uiNewVerticalBox()
uiBoxSetPadded(vbox, 1)
uiBoxAppend(hbox, vbox, 1)

grid = uiNewGrid()
uiGridSetPadded(grid, 1)
uiBoxAppend(vbox, grid, 0)

button = uiNewButton("Open File")
entryOpen = uiNewEntry()
uiEntrySetReadOnly(entryOpen, 1)
uiButtonOnClicked(button, "onOpenFileClicked()")
uiGridAppend(grid, button,
        0, 0, 1, 1,
        0, uiAlignFill, 0, uiAlignFill)
uiGridAppend(grid, entryOpen,
        1, 0, 1, 1,
        1, uiAlignFill, 0, uiAlignFill)

button = uiNewButton("Save File")
entrySave = uiNewEntry()
uiEntrySetReadOnly(entrySave, 1)
uiButtonOnClicked(button, "onSaveFileClicked()")
uiGridAppend(grid, button,
        0, 1, 1, 1,
        0, uiAlignFill, 0, uiAlignFill)
uiGridAppend(grid, entrySave,
        1, 1, 1, 1,
        1, uiAlignFill, 0, uiAlignFill)

msggrid = uiNewGrid()
uiGridSetPadded(msggrid, 1)
uiGridAppend(grid, msggrid,
        0, 2, 2, 1,
        0, uiAlignCenter, 0, uiAlignStart)

button = uiNewButton("Message Box")
uiButtonOnClicked(button, "onMsgBoxClicked()")
uiGridAppend(msggrid, button,
        0, 0, 1, 1,
        0, uiAlignFill, 0, uiAlignFill)
button = uiNewButton("Error Box")
uiButtonOnClicked(button, "onMsgBoxErrorClicked()")
uiGridAppend(msggrid, button,
        1, 0, 1, 1,
        0, uiAlignFill, 0, uiAlignFill)
```

(continues on next page)

(continued from previous page)

```
        return hbox

func makeNumbersPage

        hbox = uiNewHorizontalBox()
        uiBoxSetPadded(hbox, 1)

        group = uiNewGroup("Numbers")
        uiGroupSetMargined(group, 1)
        uiBoxAppend(hbox, group, 1)

        vbox = uiNewVerticalBox()
        uiBoxSetPadded(vbox, 1)
        uiGroupSetChild(group, vbox)

        spinbox = uiNewSpinbox(0, 100)
        slider = uiNewSlider(0, 100)
        pbar = uiNewProgressBar()
        uiSpinboxOnChanged(spinbox, "onSpinboxChanged()")
        uiSliderOnChanged(slider, "onSliderChanged()")
        uiBoxAppend(vbox, spinbox, 0)
        uiBoxAppend(vbox, slider, 0)
        uiBoxAppend(vbox, pbar, 0)

        ip = uiNewProgressBar()
        uiProgressBarSetValue(ip, -1)
        uiBoxAppend(vbox, ip, 0)

        group = uiNewGroup("Lists")
        uiGroupSetMargined(group, 1)
        uiBoxAppend(hbox, group, 1)

        vbox = uiNewVerticalBox()
        uiBoxSetPadded(vbox, 1)
        uiGroupSetChild(group, vbox)

        cbox = uiNewCombobox()
        uiComboboxAppend(cbox, "Combobox Item 1")
        uiComboboxAppend(cbox, "Combobox Item 2")
        uiComboboxAppend(cbox, "Combobox Item 3")
        uiBoxAppend(vbox, cbox, 0)

        ecbox = uiNewEditableCombobox()
        uiEditableComboboxAppend(ecbox, "Editable Item 1")
        uiEditableComboboxAppend(ecbox, "Editable Item 2")
        uiEditableComboboxAppend(ecbox, "Editable Item 3")
        uiBoxAppend(vbox, ecbox, 0)

        rb = uiNewRadioButtons()
        uiRadioButtonsAppend(rb, "Radio Button 1")
        uiRadioButtonsAppend(rb, "Radio Button 2")
        uiRadioButtonsAppend(rb, "Radio Button 3")
```

(continues on next page)

(continued from previous page)

```
        uiBoxAppend(vbox, rb, 0)

        return hbox


func makeBasicControlsPage

        vbox = uiNewVerticalBox()
        uiBoxSetPadded(vbox, 1)

        hbox = uiNewHorizontalBox()
        uiBoxSetPadded(hbox, 1)
        uiBoxAppend(vbox, hbox, 0)

        uiBoxAppend(hbox,
                uiNewButton("Button"),
                0)
        uiBoxAppend(hbox,
                uiNewCheckbox("Checkbox"),
                0)

        uiBoxAppend(vbox,
                uiNewLabel("This is a label. Right now, labels can only span one line."),
                0)

        uiBoxAppend(vbox,
                uiNewHorizontalSeparator(),
                0)

        group = uiNewGroup("Entries")
        uiGroupSetMargined(group, 1)
        uiBoxAppend(vbox, group, 1)

        entryForm = uiNewForm()
        uiFormSetPadded(entryForm, 1)
        uiGroupSetChild(group, entryForm)

        uiFormAppend(entryForm,
                "Entry",
                uiNewEntry(),
                0)
        uiFormAppend(entryForm,
                "Password Entry",
                uiNewPasswordEntry(),
                0)
        uiFormAppend(entryForm,
                "Search Entry",
                uiNewSearchEntry(),
                0)
        uiFormAppend(entryForm,
                "Multiline Entry",
                uiNewMultilineEntry(),
```

(continues on next page)

(continued from previous page)

```
                1)
        uiFormAppend(entryForm,
                "Multiline Entry No Wrap",
                uiNewNonWrappingMultilineEntry(),
                1)

        return vbox


func onSpinboxChanged
        s = uiEventSpinBox()
        uiSliderSetValue(slider, uiSpinboxValue(s));
        uiProgressBarSetValue(pbar, uiSpinboxValue(s));

func onSliderChanged
        s = uiEventSlider()
        uiSpinboxSetValue(spinbox, uiSliderValue(s));
        uiProgressBarSetValue(pbar, uiSliderValue(s));


func onOpenFileClicked
        filename = uiOpenFile(mainwin)
        if ISNULL(filename)
                uiEntrySetText(entryOpen, "(cancelled)")
                return
        ok
        uiEntrySetText(entryOpen, filename)

func onSaveFileClicked
        filename = uiSaveFile(mainwin)
        if ISNULL(filename)
                uiEntrySetText(entrySave, "(cancelled)")
                return
        ok
        uiEntrySetText(entrySave, filename)

func onMsgBoxClicked
        uiMsgBox(mainwin,
                "This is a normal message box.",
                "More detailed information can be shown here.")

func onMsgBoxErrorClicked
        uiMsgBoxError(mainwin,
                "This message box describes an error.",
                "More detailed information can be shown here.")
```

Screen Shot:

libui Control Gallery
Basic Controls
Numbers and Lists
Data Choosers
Button
Checkbox
This is a label. Right now, labels can only span one line.
Entries
Entry
Password Entry
Search Entry
Multiline Entry
Multiline Entry No Wrap

libui Control Gallery
Basic Controls
Numbers and Lists
Data Choosers
Numbers
0
Lists
Radio Button 1
Radio Button 2
Radio Button 3

libui Control Gallery
Basic Controls
Numbers and Lists
Data Choosers
12/24/2020
10:29:47 PM
12/24/2020 10:29:47 PM
Arial Regular 10
Open File
Save File
Message Box
Error Box

libui Control Gallery
Basic Controls
Numbers and Lists
Data Choosers
Color
Preview:
Opacity:
H: 0
S: 0
V: 0
R: 0 0
G: 0 0
B: 0 0
A: 1 255
Hex: #FF000000
OK
Cancel

78.4 Say Something

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

w = uiNewWindow("Hello", 320, 240, 0)
uiWindowSetMargined(w, 1)

b = uiNewVerticalBox()
uiBoxSetPadded(b, 1)
uiWindowSetChild(w, b)

e = uiNewMultilineEntry()
uiMultilineEntrySetReadOnly(e, 1)

btn = uiNewButton("Say Something")
uiButtonOnClicked(btn, "saySomething()")
uiBoxAppend(b, btn, 0)

uiBoxAppend(b, e, 1)

uiTimer(1000, "sayTime()")

uiWindowOnClosing(w, "onClosing()")
uiControlShow(w)
uiMain()

func saySomething
        uiMultilineEntryAppend(e, "Saying something"+nl)

func sayTime
        uiMultilineEntryAppend(e, Time()+nl)

func onClosing
        uiQuit()
```

Screen Shot:

78.5 Using the Menubar

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

# Create the Menubar

        fileMenu = uiNewMenu("File")
        newItem = uiMenuAppendItem(fileMenu, "New")
        openItem = uiMenuAppendItem(fileMenu, "Open")
        uiMenuAppendSeparator(fileMenu)
        shouldQuitItem = uiMenuAppendCheckItem(fileMenu, "Should Quit")
        quitItem = uiMenuAppendQuitItem(fileMenu)
```

(continues on next page)

(continued from previous page)

```
        editMenu = uiNewMenu("Edit")
        undoItem = uiMenuAppendItem(editMenu, "Undo")
        uiMenuItemDisable(undoItem)
        uiMenuAppendSeparator(editMenu)
        checkItem = uiMenuAppendCheckItem(editMenu, "Check Me\tTest")
        accelItem = uiMenuAppendItem(editMenu, "A&ccele&&rator T_es__t")
        prefsItem = uiMenuAppendPreferencesItem(editMenu)

        testMenu = uiNewMenu("Test")
        enabledItem = uiMenuAppendCheckItem(testMenu, "Enable Below Item")
        uiMenuItemSetChecked(enabledItem, 1)
        enableThisItem = uiMenuAppendItem(testMenu, "This Will Be Enabled")
        uiMenuItemOnClicked(enabledItem, "enableItemTest(enableThisItem)")
        forceCheckedItem = uiMenuAppendItem(testMenu, "Force Above Checked")
        uiMenuItemOnClicked(forceCheckedItem, "forceOn()")
        forceUncheckedItem = uiMenuAppendItem(testMenu, "Force Above Unchecked")
        uiMenuItemOnClicked(forceUncheckedItem, "forceOff()")
        uiMenuAppendSeparator(testMenu)
        whatWindowItem = uiMenuAppendItem(testMenu, "What Window?")
        uiMenuItemOnClicked(whatWindowItem, "whatWindow()")

        moreTestsMenu = uiNewMenu("More Tests")
        quitEnabledItem = uiMenuAppendCheckItem(moreTestsMenu, "Quit Item Enabled")
        uiMenuItemSetChecked(quitEnabledItem, 1)
        prefsEnabledItem = uiMenuAppendCheckItem(moreTestsMenu, "Preferences Item Enabled
↪")
        uiMenuItemSetChecked(prefsEnabledItem, 1)
        aboutEnabledItem = uiMenuAppendCheckItem(moreTestsMenu, "About Item Enabled")
        uiMenuItemSetChecked(aboutEnabledItem, 1)
        uiMenuAppendSeparator(moreTestsMenu)
        checkEnabledItem = uiMenuAppendCheckItem(moreTestsMenu, "Check Me Item Enabled")
        uiMenuItemSetChecked(checkEnabledItem, 1)

        multiMenu = uiNewMenu("Multi")
        uiMenuAppendSeparator(multiMenu)
        uiMenuAppendSeparator(multiMenu)
        uiMenuAppendItem(multiMenu, "Item && Item && Item")
        uiMenuAppendSeparator(multiMenu)
        uiMenuAppendSeparator(multiMenu)
        uiMenuAppendItem(multiMenu, "Item __ Item __ Item")
        uiMenuAppendSeparator(multiMenu)
        uiMenuAppendSeparator(multiMenu)

        helpMenu = uiNewMenu("Help")
        helpItem = uiMenuAppendItem(helpMenu, "Help")
        aboutItem = uiMenuAppendAboutItem(helpMenu)

        uiMenuItemOnClicked(quitEnabledItem, "enableItemTest(quitItem)")
        uiMenuItemOnClicked(prefsEnabledItem, "enableItemTest(prefsItem)")
        uiMenuItemOnClicked(aboutEnabledItem, "enableItemTest(aboutItem)")
        uiMenuItemOnClicked(checkEnabledItem, "enableItemTest(checkItem)")
```

(continues on next page)

(continued from previous page)

```
# Create the Window

        oWindow = uiNewWindow( "Using the Menubar", 400, 400, True)
        uiWindowOnClosing(oWindow,"closeApp()")

        uiControlShow( oWindow )
        uiMain()

func enableItemTest(data)
        item = uiEventMenuItem()
        if uiMenuItemChecked(item)
                uiMenuItemEnable(data)
        else
                uiMenuItemDisable(data)
        ok

func forceOn
        uiMenuItemSetChecked(enabledItem, 1)

func forceOff
        uiMenuItemSetChecked(enabledItem, 0)

func whatWindow
        ? "menu item clicked on window "
        ? oWindow

func closeApp
        uiQuit()
```

Screen Shot:

Using the Menubar
File
Edit
Test
More Tests
Multi
Help

78.6 Drawing Sample

```
load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 420, 450, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()
```

(continues on next page)

(continued from previous page)

```
func draw
        Rectangle(0, 0, uiEventAreaWidth(), uiEventAreaHeight(), colorGray)
        Rectangle(0, 0, 400, 400, colorWhite)
        Rectangle(10, 10, 20, 20, colorRed)
        Rectangle(30, 30, 30, 30, colorGreen)
        Rectangle(60, 60, 40, 40, colorBlue)

# The Rectangle function is now part of RingLibUI as uiRectangle()
func Rectangle x,y,width,height,color
        oContext = uiEventContext()
        oBrush = uiNewSolidBrush(color)
        oPath = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathAddRectangle(oPath, x, y, width, height)
        uiDrawPathEnd(oPath)
        uiDrawFill(oContext, oPath, oBrush)
        uiDrawFreePath(oPath)

func closeApp
        uiQuit()
```

Screen Shot:

78.7 Draw Gradient

```
load "libui.ring"

oWindow = uiNewWindow( "Draw Gradient", 500, 500, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()

func draw
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()
        uiRectangle(0, 0, nWidth, nHeight, colorBlue)
        for y=0 to 255 step 2
                customColor = y
                uiRectangle(0, y, nWidth, y+1, customColor)
        next

func closeApp
        uiQuit()
```

Screen Shot:

78.8 Histogram

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

datapoints      = list(10)
currentPoint    = -1

// some metrics
xoffLeft        = 20                    /* histogram margins */
yoffTop         = 20
xoffRight       = 20
yoffBottom      = 20
pointRadius     = 5
```

(continues on next page)

(continued from previous page)

```
histogram       = NULL
mainwin         = NULL
colorButton     = NULL

func pointLocations width, height, xs, ys
        xincr = width / 9               // 10 - 1 to make the last point be at the end
        yincr = height / 100
        for i = 1 to 10
                // get the value of the point
                n = uiSpinboxValue(datapoints[i])
                // because y=0 is the top but n=0 is the bottom, we need to flip
                n = 100 - n;
                xs[i] = xincr * i
                ys[i] = yincr * n
        next

func constructGraph width, height, extend
        xs = list(10)
        ys = list(10)
        pointLocations(width, height, xs, ys)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, xs[1], ys[1])
        for i = 2 to 10
                uiDrawPathLineTo(path, xs[i], ys[i])
        next
        if extend
                uiDrawPathLineTo(path, width, height)
                uiDrawPathLineTo(path, 0, height)
                uiDrawPathCloseFigure(path)
        ok
        uiDrawPathEnd(path)
        return path


func graphSize clientWidth, clientHeight
        graphWidth = clientWidth - xoffLeft - xoffRight
        graphHeight = clientHeight - yoffTop - yoffBottom
        return [graphWidth,graphHeight]

func handlerDraw
        // fill the area with white
        Brush = uiNewSolidBrush(0)
        setSolidBrush(brush, colorWhite, 1.0)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathAddRectangle(path, 0, 0, uiEventAreaWidth(), uiEventAreaHeight())
        uiDrawPathEnd(path)
        uiDrawFill(uiEventContext(), path, brush)
        uiDrawFreePath(path)

        // figure out dimensions
        aOut = graphSize(uiEventAreaWidth(), uiEventAreaHeight())
        graphWidth = aOut[1]
```

(continues on next page)

(continued from previous page)

```
        graphHeight = aOut[2]

        sp = new_managed_uiDrawStrokeParams()
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_Thickness(sp,2)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)

        // draw the axes
        setSolidBrush(brush, colorBlack, 1.0)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path,
                xoffLeft, yoffTop)
        uiDrawPathLineTo(path,
                xoffLeft, yoffTop + graphHeight)
        uiDrawPathLineTo(path,
                xoffLeft + graphWidth, yoffTop + graphHeight)
        uiDrawPathEnd(path)
        uiDrawStroke(uiEventContext(), path, brush, sp)
        uiDrawFreePath(path)

        // now transform the coordinate space so (0, 0) is the top-left corner of the␣
↪graph
        m = new_managed_uiDrawMatrix()
        uiDrawMatrixSetIdentity(m)
        uiDrawMatrixTranslate(m, xoffLeft, yoffTop)
        uiDrawTransform(uiEventContext(), m)

        // now get the color for the graph itself and set up the brush

        GraphR=0
        GraphG=0
        GraphB=0
        GraphA=0

        uiColorButtonColor(colorButton, :graphR,
                 :graphG,
                 :graphB,
                 :graphA)

        uiSetBrushType(brush,uiDrawBrushTypeSolid)
        uiSetBrushR(brush,graphR)
        uiSetBrushG(brush,graphG)
        uiSetBrushB(brush,graphB)

        // we set brush->A below to different values for the fill and stroke

        // now create the fill for the graph below the graph line
        path = constructGraph(graphWidth, graphHeight, 1)

  uiSetBrushA(brush, graphA / 2)
        uiDrawFill(uiEventContext(), path, brush)
```

(continues on next page)

(continued from previous page)

```
        uiDrawFreePath(path)

        // now draw the histogram line
        path = constructGraph(graphWidth, graphHeight, 0)
        uiSetBrushA(brush,graphA)
        uiDrawStroke(uiEventContext(), path, brush, sp)
        uiDrawFreePath(path)

        // now draw the point being hovered over
        if currentPoint != -1
                xs = list(10)
                ys = list(10)
                pointLocations(graphWidth, graphHeight, xs, ys)
                path = uiDrawNewPath(uiDrawFillModeWinding)

                uiDrawPathNewFigureWithArc(path,
                        xs[currentPoint], ys[currentPoint],
                        pointRadius,
                        0, 6.23,                // TODO pi
                        0)
                uiDrawPathEnd(path)
                // use the same brush as for the histogram lines
                uiDrawFill(uiEventContext(), path, brush)
                uiDrawFreePath(path)
        ok

func inPoint x, y, xtest, ytest
        // TODO switch to using a matrix
        x -= xoffLeft
        y -= yoffTop
        return (x >= xtest - pointRadius) &&
                (x <= xtest + pointRadius) &&
                (y >= ytest - pointRadius) &&
                (y <= ytest + pointRadius)

func handlerMouseEvent
        xs = list(10)
        ys = list(10)

        aOut = graphSize(uiEventAreaWidth(), uiEventAreaHeight())
        graphWidth = aOut[1]
        graphHeight = aOut[2]

        pointLocations(graphWidth, graphHeight, xs, ys)

        e = uiEventAreaMouseEvent()
        eX = get_uiAreaMouseEvent_X(e)
        eY = get_uiAreaMouseEvent_Y(e)
        for i=1 to 10
                if inPoint(eX, eY, xs[i], ys[i])
                        exit
                ok
```

(continues on next page)

(continued from previous page)

```
        next
        if i = 11               // not in a point
                i = -1
        ok

        currentPoint = i

        uiAreaQueueRedrawAll(histogram)

func onDatapointChanged
        uiAreaQueueRedrawAll(histogram)

func onColorChanged
        uiAreaQueueRedrawAll(histogram);

func onClosing
        uiControlDestroy(uiControl(mainwin))
        uiQuit()
        return 0

func shouldQuit
        uiControlDestroy(uiControl(mainwin))

func main

        uiOnShouldQuit("shouldQuit()")

        mainwin = uiNewWindow("Histogram Sample", 800, 480, 1)
        uiWindowSetMargined(mainwin, 1)
        uiWindowOnClosing(mainwin, "onClosing()")

        Brush = uiNewSolidBrush(0)

        hbox = uiNewHorizontalBox()
        uiBoxSetPadded(hbox, 1)
        uiWindowSetChild(mainwin, uiControl(hbox))

        vbox = uiNewVerticalBox()
        uiBoxSetPadded(vbox, 1)
        uiBoxAppend(hbox, uiControl(vbox), 0)

        srandom(clock());
        for i=1 to 10
                datapoints[i] = uiNewSpinbox(0, 100)
                uiSpinboxSetValue(datapoints[i], random() % 101)
                uiSpinboxOnChanged(datapoints[i], "onDatapointChanged()")
                uiBoxAppend(vbox, uiControl(datapoints[i]), 0)
        next

        colorButton = uiNewColorButton()

        setSolidBrush(brush, colorDodgerBlue, 1.0)
```

(continues on next page)

(continued from previous page)

```
        uiColorButtonSetColor(colorButton,
                uiBrushR(brush),
                uiBrushG(brush),
                uiBrushB(brush),
                uiBrushA(brush))

        uiColorButtonOnChanged(colorButton, "onColorChanged()")
        uiBoxAppend(vbox, uiControl(colorButton), 0)

        oAreaHandler = uiNewAreaHandler("handlerDraw()","handlerMouseEvent()","","","")
        histogram = uiNewArea(oAreaHandler)
        uiBoxAppend(hbox, uiControl(histogram), 1)

        uiControlShow(uiControl(mainwin))
        uiMain()
```

Screen Shot:

78.9 Text Drawing

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

# Global Variables
        mainwin         = NULL
        area            = NULL
        handler         = NULL
        fontButton      = NULL
        alignment       = NULL
        attrstr         = NULL

func appendWithAttribute what, attr, attr2
        nStart = uiAttributedStringLen(attrstr)
        nEnd = nStart + len(what)
        uiAttributedStringAppendUnattributed(attrstr, what)
        uiAttributedStringSetAttribute(attrstr, attr, nStart, nEnd)
        if ! ISNULL(attr2)
                uiAttributedStringSetAttribute(attrstr, attr2, nStart, nEnd)
        ok

func makeAttributedString

        attrstr = uiNewAttributedString("Drawing strings with libui is done with the␣
↪uiAttributedString and uiDrawTextLayout objects."+nl+
                "uiAttributedString lets you have a variety of attributes: ")

        attr = uiNewFamilyAttribute("Courier New")
        appendWithAttribute("font family", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, ", ")

        attr = uiNewSizeAttribute(18)
        appendWithAttribute("font size", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, ", ")

        attr = uiNewWeightAttribute(uiTextWeightBold)
        appendWithAttribute("font weight", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, ", ")

        attr = uiNewItalicAttribute(uiTextItalicItalic)
        appendWithAttribute("font italicness", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, ", ")

        attr = uiNewStretchAttribute(uiTextStretchCondensed)
        appendWithAttribute("font stretch", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, ", ")

        attr = uiNewColorAttribute(0.75, 0.25, 0.5, 0.75)
        appendWithAttribute("text color", attr, NULL)
```

(continues on next page)

(continued from previous page)

```
        uiAttributedStringAppendUnattributed(attrstr, ", ")

        attr = uiNewBackgroundAttribute(0.5, 0.5, 0.25, 0.5)
        appendWithAttribute("text background color", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, ", ")


        attr = uiNewUnderlineAttribute(uiUnderlineSingle)
        appendWithAttribute("underline style", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, ", ")

        uiAttributedStringAppendUnattributed(attrstr, "and ")
        attr = uiNewUnderlineAttribute(uiUnderlineDouble)
        attr2 = uiNewUnderlineColorAttribute(uiUnderlineColorCustom, 1.0, 0.0, 0.5, 1.0)
        appendWithAttribute("underline color", attr, attr2)
        uiAttributedStringAppendUnattributed(attrstr, ". ")

        uiAttributedStringAppendUnattributed(attrstr, "Furthermore, there are attributes␣
↪allowing for ")
        attr = uiNewUnderlineAttribute(uiUnderlineSuggestion)
        attr2 = uiNewUnderlineColorAttribute(uiUnderlineColorSpelling, 0, 0, 0, 0)
        appendWithAttribute("special underlines for indicating spelling errors", attr,␣
↪attr2)
        uiAttributedStringAppendUnattributed(attrstr, " (and other types of errors) ")

        uiAttributedStringAppendUnattributed(attrstr, "and control over OpenType␣
↪features such as ligatures (for instance, ")
        otf = uiNewOpenTypeFeatures()
        uiOpenTypeFeaturesAdd(otf, ASCII('l'), ASCII('i'), ASCII('g'), ASCII('a'), 0)
        attr = uiNewFeaturesAttribute(otf)
        appendWithAttribute("afford", attr, NULL)
        uiAttributedStringAppendUnattributed(attrstr, " vs. ")
        uiOpenTypeFeaturesAdd(otf, ASCII('l'), ASCII('i'), ASCII('g'), ASCII('a'), 1)
        attr = uiNewFeaturesAttribute(otf)
        appendWithAttribute("afford", attr, NULL)
        uiFreeOpenTypeFeatures(otf)
        uiAttributedStringAppendUnattributed(attrstr, ").\n")

        uiAttributedStringAppendUnattributed(attrstr, "Use the controls opposite to the␣
↪text to control properties of the text.")

func handlerDraw

        defaultfont = new_uiFontDescriptor()
        params = new_uiDrawTextLayoutParams()

        set_uiDrawTextLayoutParams_String(params,attrstr)
        uiFontButtonFont(fontButton, defaultFont)
        set_uiDrawTextLayoutParams_DefaultFont(params,defaultFont)
        set_uiDrawTextLayoutParams_Width(params,uiEventAreaWidth())
        set_uiDrawTextLayoutParams_Align(params,uiComboboxSelected(alignment))
        textLayout = uiDrawNewTextLayout(params)
```

(continues on next page)

(continued from previous page)

```
        uiDrawText(uiEventContext(), textLayout, 0, 0)
        uiDrawFreeTextLayout(textLayout)
        uiFreeFontButtonFont(defaultFont)

func onFontChanged
        uiAreaQueueRedrawAll(area)

func onComboboxSelected
        uiAreaQueueRedrawAll(area)

func onClosing
        uiControlDestroy(mainwin)
        uiQuit()

func shouldQuit
        uiControlDestroy(mainwin)

func main

        uiOnShouldQuit("shouldQuit()")

        makeAttributedString()

        mainwin = uiNewWindow("libui Text-Drawing Example", 640, 480, 1)
        uiWindowSetMargined(mainwin, 1)
        uiWindowOnClosing(mainwin, "onClosing()")

        hbox = uiNewHorizontalBox()
        uiBoxSetPadded(hbox, 1)
        uiWindowSetChild(mainwin, hbox)

        vbox = uiNewVerticalBox()
        uiBoxSetPadded(vbox, 1)
        uiBoxAppend(hbox, vbox, 0)

        fontButton = uiNewFontButton()
        uiFontButtonOnChanged(fontButton, "onFontChanged()")
        uiBoxAppend(vbox, fontButton, 0)

        form = uiNewForm()
        uiFormSetPadded(form, 1)
        uiBoxAppend(vbox, form, 0)

        alignment = uiNewCombobox()
        uiComboboxAppend(alignment, "Left")
        uiComboboxAppend(alignment, "Center")
        uiComboboxAppend(alignment, "Right")
        uiComboboxSetSelected(alignment, 0)             // start with left alignment
        uiComboboxOnSelected(alignment, "onComboboxSelected()")
        uiFormAppend(form, "Alignment", alignment, 0)

        oAreaHandler = uiNewAreaHandler("handlerDraw()","","","","")
```

(continues on next page)

(continued from previous page)

```
        area = uiNewArea(oAreaHandler)
        uiBoxAppend(hbox, area, 1)

        uiControlShow(mainwin)
        uiMain()
        uiFreeAttributedString(attrstr)
```

Screen Shot:

libui Text-Drawing Example
Arial Regular 16
Alignment
Left
Drawing strings with libui is done with the uiAttributedString and uiDrawTextLayout objects. uiAttributedString lets you have a variety of attributes: font family, font size, font weight, font italicness, font stretch, text color, text background color, underline style, and underline color. Furthermore, there are attributes allowing for special underlines for indicating spelling errors (and other types of errors) and control over OpenType features such as ligatures (for instance, afford vs. afford).\nUse the controls opposite to the text to control properties of the text.

78.10 More Drawing Samples

Example (1):

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 400, 400, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
```

(continues on next page)

(continued from previous page)

```
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()

func draw
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()
        source = new_uiDrawBrush()
        sp = new_uiDrawStrokeParams()
        source = uiNewSolidBrush(colorBlue)
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)
        set_uiDrawStrokeParams_NumDashes(sp,0)
        set_uiDrawStrokeParams_DashPhase(sp,0)
        set_uiDrawStrokeParams_Thickness(sp,40.96)

        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, 76.8, 84.48)
        uiDrawPathLineTo(path, 76.8 + 51.2, 84.48 -51.2)
        uiDrawPathLineTo(path, 76.8 + 51.2 + 51.2, 84.48 - 51.2 + 51.2)
        uiDrawPathEnd(path)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, 76.8, 161.28)
        uiDrawPathLineTo(path, 76.8 + 51.2, 161.28 -51.2)
        uiDrawPathLineTo(path, 76.8 + 51.2 + 51.2, 161.28 - 51.2 + 51.2)
        uiDrawPathEnd(path)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinBevel)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, 76.8, 238.08)
        uiDrawPathLineTo(path, 76.8 + 51.2, 238.08 -51.2)
        uiDrawPathLineTo(path, 76.8 + 51.2 + 51.2, 238.08 - 51.2 + 51.2)
        uiDrawPathEnd(path)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinRound)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

func closeApp
        uiQuit()
```

Screen Shot:

Example (2):

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 400, 400, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()

func draw
```

(continues on next page)

(continued from previous page)

```
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()

        source = new_uiDrawBrush()
        sp = new_uiDrawStrokeParams()
        source = uiNewSolidBrush(colorBlack)
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)
        set_uiDrawStrokeParams_NumDashes(sp,0)
        set_uiDrawStrokeParams_DashPhase(sp,0)
        set_uiDrawStrokeParams_Thickness(sp,30)

        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, 64.0, 50.0)
        uiDrawPathLineTo(path, 64.0, 200.0)
        uiDrawPathEnd(path)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapRound)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, 128.0, 50.0)
        uiDrawPathLineTo(path, 128.0, 200.0)
        uiDrawPathEnd(path)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapSquare)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, 192.0, 50.0)
        uiDrawPathLineTo(path, 192.0, 200.0)
        uiDrawPathEnd(path)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

        // draw helping lines
        // keep the square cap to match the reference picture on the cairo website
        uiCrSourceRGBA(source, 1, 0.2, 0.2, 1)
        set_uiDrawStrokeParams_Thickness(sp,2.56)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, 64.0, 50.0)
        uiDrawPathLineTo(path, 64.0, 200.0)
        uiDrawPathNewFigure(path, 128.0, 50.0)
        uiDrawPathLineTo(path, 128.0, 200.0)
        uiDrawPathNewFigure(path, 192.0, 50.0)
        uiDrawPathLineTo(path, 192.0, 200.0)
        uiDrawPathEnd(path)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

func closeApp
```

(continues on next page)

(continued from previous page)

```
        uiQuit()
```

Screen Shot:

Example (3):

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 260, 300, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)
```

(continues on next page)

(continued from previous page)

```
uiControlShow( oWindow )
uiMain()

func draw
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()

        source = new_uiDrawBrush()
        sp = new_uiDrawStrokeParams()

        x         = 25.6
        y         = 25.6
        width         = 204.8
        height        = 204.8
        aspect        = 1.0
        corner_radius = height

        radius = corner_radius / aspect
        degrees = uiPi / 180.0

        source = uiNewSolidBrush(colorBlue)
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)
        set_uiDrawStrokeParams_NumDashes(sp,0)
        set_uiDrawStrokeParams_DashPhase(sp,0)
        set_uiDrawStrokeParams_Thickness(sp,30)

        path = uiDrawNewPath(uiDrawFillModeWinding)

        // top right corner
        uiDrawPathNewFigureWithArc(path,
                x + width - radius, y + radius,
                radius,
                -90 * degrees, uiPi / 2,
                0)
        // bottom right corner
        uiDrawPathArcTo(path,
                x + width - radius, y + height - radius,
                radius,
                0 * degrees, uiPi / 2,
                0)
        // bottom left corner
        uiDrawPathArcTo(path,
                x + radius, y + height - radius,
                radius,
                90 * degrees, uiPi / 2,
                0)
        // top left corner
        uiDrawPathArcTo(path,
                x + radius, y + radius,
                radius,
```

(continues on next page)

(continued from previous page)

```
                180 * degrees, uiPi / 2,
                0)
        uiDrawPathCloseFigure(path)
        uiDrawPathEnd(path)

        uiCrSourceRGBA(source, 0.5, 0.5, 1, 1)
        uiDrawFill(uiEventContext(), path, source)
        uiCrSourceRGBA(source, 0.5, 0, 0, 0.5)
        set_uiDrawStrokeParams_Thickness(sp,10)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

func closeApp
        uiQuit()
```

Screen Shot:

Example (4):

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 300, 300, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
```

(continues on next page)

(continued from previous page)

```
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()

func draw
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()

        source = new_uiDrawBrush()
        sp = new_uiDrawStrokeParams()

        source = uiNewSolidBrush(colorBlue)
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)
        set_uiDrawStrokeParams_NumDashes(sp,0)
        set_uiDrawStrokeParams_DashPhase(sp,0)

        path = uiDrawNewPath(uiDrawFillModeWinding)

        uiDrawPathNewFigure(path, 50.0, 75.0)
        uiDrawPathLineTo(path, 200.0, 75.0)

        uiDrawPathNewFigure(path, 50.0, 125.0)
        uiDrawPathLineTo(path, 200.0, 125.0)

        uiDrawPathNewFigure(path, 50.0, 175.0)
        uiDrawPathLineTo(path, 200.0, 175.0)
        uiDrawPathEnd(path)

        set_uiDrawStrokeParams_Thickness(sp,30)
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapRound)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

func closeApp
        uiQuit()
```

Screen Shot:

Example (5):

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 260, 300, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()

func draw
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()

        source = new_uiDrawBrush()
        sp = new_uiDrawStrokeParams()
        m = new_uiDrawMatrix()
        source = uiNewSolidBrush(colorBlue)
```

(continues on next page)

(continued from previous page)

```
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)
        set_uiDrawStrokeParams_NumDashes(sp,0)
        set_uiDrawStrokeParams_DashPhase(sp,0)
        set_uiDrawStrokeParams_Thickness(sp,6)

        path = uiDrawNewPath(uiDrawFillModeAlternate)
        uiDrawPathAddRectangle(path, 12, 12, 232, 70)
        uiDrawPathNewFigureWithArc(path,
                64, 64,
                40,
                0, 2*uiPi,
                0)
        uiDrawPathNewFigureWithArc(path,
                192, 64,
                40,
                0, -2*uiPi,
                1)
        uiDrawPathEnd(path)

        uicrsourcergba(source, 0, 0.7, 0, 1)
        uiDrawFill(uiEventContext(), path, source)
        uicrsourcergba(source, 0, 0, 0, 1)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

        uiDrawMatrixSetIdentity(m)
        uiDrawMatrixTranslate(m, 0, 128)
        uiDrawTransform(uiEventContext(), m)

        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathAddRectangle(path, 12, 12, 232, 70)
        uiDrawPathNewFigureWithArc(path,
                64, 64,
                40,
                0, 2*uiPi,
                0)
        uiDrawPathNewFigureWithArc(path,
                192, 64,
                40,
                0, -2*uiPi,
                1)
        uiDrawPathEnd(path)

        uicrsourcergba(source, 0, 0, 0.9, 1)
        uiDrawFill(uiEventContext(), path, source)
        uicrsourcergba(source, 0, 0, 0, 1)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

func closeApp
```

(continues on next page)

(continued from previous page)

```
        uiQuit()
```

Screen Shot:

Example (6):

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 260, 300, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()

func draw
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()
```

(continues on next page)

(continued from previous page)

```
        source = new_uiDrawBrush()
        sp = new_uiDrawStrokeParams()
        source = uiNewSolidBrush(colorBlue)
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)
        set_uiDrawStrokeParams_NumDashes(sp,0)
        set_uiDrawStrokeParams_DashPhase(sp,0)

        path = uiDrawNewPath(uiDrawFillModeWinding)

        uiDrawPathNewFigure(path, 128.0, 25.6)
        uiDrawPathLineTo(path, 230.4, 230.4)
        uiDrawPathLineTo(path, 230.4 - 102.4, 230.4 + 0.0)
        uiDrawPathBezierTo(path, 51.2, 230.4, 51.2, 128.0, 128.0, 128.0)
        uiDrawPathCloseFigure(path)

        uiDrawPathNewFigure(path, 64.0, 25.6)
        uiDrawPathLineTo(path, 64.0 + 51.2, 25.6 + 51.2)
        uiDrawPathLineTo(path, 64.0 + 51.2 -51.2, 25.6 + 51.2 + 51.2)
        uiDrawPathLineTo(path, 64.0 + 51.2 -51.2 -51.2, 25.6 + 51.2 + 51.2 -51.2)
        uiDrawPathCloseFigure(path)

        uiDrawPathEnd(path)

        set_uiDrawStrokeParams_Thickness(sp,10)
        uicrsourcergba(source, 0, 0, 1, 1)
        uiDrawFill(uiEventContext(), path, source)
        uicrsourcergba(source, 0, 0, 0, 1)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

func closeApp
        uiQuit()
```

Screen Shot:

Example (7):

```
# Sample ported to Ring
# Based on original sample from : https://github.com/andlabs/libui

load "libui.ring"

oWindow = uiNewWindow( "Drawing Sample", 260, 300, True)
uiWindowOnClosing(oWindow,"closeApp()")

oAreaHandler = uiNewAreaHandler("draw()","","","","")
area = uiNewArea(oAreaHandler)

btnClose = uiNewButton("Close Application")
uiButtonOnClicked(btnClose,"closeApp()")

hbox = uiNewVerticalBox()
uiBoxSetPadded(hbox, 1)
uiBoxAppend(hbox,btnClose,0)
uiBoxAppend(hbox,area,1)
uiWindowSetChild(oWindow, hbox)

uiControlShow( oWindow )
uiMain()

func draw
        nWidth = uiEventAreaWidth()             nHeight = uiEventAreaHeight()

        source = new_uiDrawBrush()
        sp = new_uiDrawStrokeParams()
        source = uiNewSolidBrush(colorBlue)
```

(continues on next page)

(continued from previous page)

```
        x=25.6   y=128.0
        x1=102.4 y1=230.4
        x2=153.6 y2=25.6
        x3=230.4 y3=128.0

        uicrsourcergba(source, 0, 0, 0, 1)
        set_uiDrawStrokeParams_Cap(sp,uiDrawLineCapFlat)
        set_uiDrawStrokeParams_Join(sp,uiDrawLineJoinMiter)
        set_uiDrawStrokeParams_MiterLimit(sp,uiDrawDefaultMiterLimit)
        set_uiDrawStrokeParams_NumDashes(sp,0)
        set_uiDrawStrokeParams_DashPhase(sp,0)

        path = uiDrawNewPath(uiDrawFillModeWinding)

        uiDrawPathNewFigure(path, x, y)
        uiDrawPathBezierTo(path, x1, y1, x2, y2, x3, y3)
        uiDrawPathEnd(path)
        set_uiDrawStrokeParams_Thickness(sp,10)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

        uicrsourcergba(source, 1, 0.2, 0.2, 0.6)
        set_uiDrawStrokeParams_Thickness(sp,6)
        path = uiDrawNewPath(uiDrawFillModeWinding)
        uiDrawPathNewFigure(path, x, y)
        uiDrawPathLineTo(path, x1, y1)
        uiDrawPathNewFigure(path, x2, y2)
        uiDrawPathLineTo(path, x3, y3)
        uiDrawPathEnd(path)
        uiDrawStroke(uiEventContext(), path, source, sp)
        uiDrawFreePath(path)

func closeApp
        uiQuit()
```

Screen Shot:

Drawing Sample
Close Application

CHAPTER

SEVENTYNINE

USING RINGZIP

In this chapter we will learn about using RingZip

79.1 Create Zip File

Example : Create myfile.zip contains 4 files

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip",'w')
zip_addfile(oZip,"test.c")
zip_addfile(oZip,"zip.c")
zip_addfile(oZip,"zip.h")
zip_addfile(oZip,"miniz.h")
zip_close(oZip)
```

79.2 Extract Zip File

Example : Extract myfile.zip to myfolder folder.

```
load "ziplib.ring"
zip_extract_allfiles("myfile.zip","myfolder")
```

79.3 Print Files in Zip file

Example : Print file names in the myfile.zip

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip",'r')
for x=1 to zip_filescount(oZip)
       see zip_getfilenamebyindex(oZip,x) + nl
next
zip_close(oZip)
```

79.4 Using RingZip Classes

The RingZip library comes with two classes. The Zip class and the ZipEntry class.

Example (1):

```
load "ziplib.ring"

new Zip {
        setFileName("myfile.zip")
        open("w")
        newEntry() {
                open("test.c")
                writefile("test.c")
                close()
        }
        close()
}
```

Example (2):

```
load "ziplib.ring"

new Zip {
        SetFileName("myfile.zip")
        Open("w")
        AddFile("test.c")
        AddFile("zip.c")
        AddFile("zip.h")
        AddFile("miniz.h")
        Close()
}
```

Example (3):

```
load "ziplib.ring"

new zip {
        SetFileName("myfile.zip")
        ExtractAllFiles("myfolder")
}
```

Example (4):

```
load "ziplib.ring"

new Zip {
        SetFileName("myfile.zip")
        Open("r")
        see FilesCount()
        Close()
}
```

Example (5):

```
load "ziplib.ring"

new Zip {
        SetFileName("myfile.zip")
        Open("r")
        for x = 1 to filescount()
                See GetFileNameByIndex(x) + nl
        next
        Close()
}
```

79.5 Zip Class Reference

Methods:

Method	Description/Output
SetFileName(cName)	Set the Zip file name
GetFileName()	Return the Zip file name
Open(cMode)	Open File, cMode = "a", "w" or "r"
Close()	Close the Zip File
AddFile(cFileName)	Add file to the Zip file
ExtractAllFiles(cFolder)	Extract all files from the Zip file
FilesCount()	Return files count in the Zip file
GetFileNameByIndex(nIndex)	Return file name in the Zip file by file index
NewEntry()	Create new ZipEntry object

79.6 ZipEntry Class Reference

Methods:

Method	Description/Output
Open(cFileName)	Open new Entry
WriteFile(cFileName)	Write File to the Entry
WriteString(cString)	Write String to the Entry
Close()	Close the Entry

CHAPTER

EIGHTY

RINGLIBZIP FUNCTIONS REFERENCE

80.1 Introduction

In this chapter we have a list of the supported functions by this extension

80.2 Reference

- ZIP_T *zip_openfile(const char *, const char *)
- int zip_entry_open(ZIP_T*, const char *)
- int zip_entry_write(ZIP_T*, const char *,int)
- int zip_entry_fwrite(ZIP_T*, const char *)
- int zip_entry_read(ZIP_T*, void *, size_t *)
- int zip_entry_fread(ZIP_T*, const char *cFile)
- int zip_entry_close(ZIP_T*)
- void zip_extract_allfiles(const char *cFile, const char *cFolder)
- void zip_close(ZIP_T*)
- int zip_filescount(ZIP_T *)
- const char *zip_getfilenamebyindex(ZIP_T *pZip,int index)
- void zip_extract_file(const char *cZIPFile,const char *cFile)

CHAPTER

EIGHTYONE

RINGMURMURHASH FUNCTIONS REFERENCE

Murmurhash extension is an extension written to implement a full implementation for the MurmurHash library.

81.1 MurmurHash1 functions

uint32_t murmurhash1(string key, int seed, [bool return_type]);

uint32_t murmurhash1_aligned(string key, int seed, [bool return_type]);

81.2 MurmurHash2 functions

uint32_t murmurhash2(string key, int seed, [bool return_type]);

uint32_t murmurhash2a(string key, int seed, [bool return_type]);

uint64_t murmurhash64a(string key, int seed, [bool return_type]);

uint64_t murmurhash64b(string key, int seed, [bool return_type]);

uint32_t murmurhash_neutral2(string key, int seed, [bool return_type]);

uint32_t murmurhash_aligned2(string key, int seed, [bool return_type]);

81.3 MurmurHash3 functions

uint32_t murmurhash3_x86_32(string key, int seed, [bool return_type]);

list murmurhash3_x86_128(string key, int seed, [bool return_type]);

list murmurhash3_x64_128(string key, int seed, [bool return_type]);

The third optional parameter is to set the type of the returned value, this parameter accepts a bool value [true, false], true will return a Hex value, while false will return a integer value.

81.4 Example

```
load "murmurhashlib.ring"

key = "Ring Language"

see murmurhash3_x86_32(key, 0, 0) + nl // Output: 1894444853
see murmurhash3_x86_32(key, 0, 1) + nl // Output: 70eaef35
```

CHAPTER

EIGHTYTWO

RINGCONSOLECOLORS FUNCTIONS REFERENCE

82.1 Introduction

In this chapter we have a list of the supported functions by this extension

82.2 Reference

- CC_FG_NONE
- CC_FG_BLACK
- CC_FG_DARK_RED
- CC_FG_DARK_GREEN
- CC_FG_DARK_YELLOW
- CC_FG_DARK_BLUE
- CC_FG_DARK_MAGENTA
- CC_FG_DARK_CYAN
- CC_FG_GRAY
- CC_FG_DARK_GRAY
- CC_FG_RED
- CC_FG_GREEN
- CC_FG_YELLOW
- CC_FG_BLUE
- CC_FG_MAGENTA
- CC_FG_CYAN
- CC_FG_WHITE
- CC_BG_NONE
- CC_BG_BLACK
- CC_BG_DARK_RED
- CC_BG_DARK_GREEN
- CC_BG_DARK_YELLOW

- CC_BG_DARK_BLUE
- CC_BG_DARK_MAGENTA
- CC_BG_DARK_CYAN
- CC_BG_GRAY
- CC_BG_DARK_GRAY
- CC_BG_RED
- CC_BG_GREEN
- CC_BG_YELLOW
- CC_BG_BLUE
- CC_BG_MAGENTA
- CC_BG_CYAN
- CC_BG_WHITE
- void cc_print(int color,const char *string)

CHAPTER

EIGHTYTHREE

RINGALLEGRO FUNCTIONS REFERENCE

83.1 Introduction

In this chapter we have a list of the supported functions by this extension

83.2 Reference

- void al_exit(void)
- void al_run_main(void)
- int al_init(void)
- ALLEGRO_CONFIG *al_create_config(void)
- void al_destroy_config(ALLEGRO_CONFIG *config)
- ALLEGRO_CONFIG *al_load_config_file(const char *filename)
- ALLEGRO_CONFIG *al_load_config_file_f(ALLEGRO_FILE *file)
- bool al_save_config_file(const char *filename, const ALLEGRO_CONFIG *config)
- bool al_save_config_file_f(ALLEGRO_FILE *file, const ALLEGRO_CONFIG *config)
- void al_add_config_section(ALLEGRO_CONFIG *config, const char *name)
- void al_add_config_comment(ALLEGRO_CONFIG *config, const char *section, const char *comment)
- const char *al_get_config_value(const ALLEGRO_CONFIG *config,const char *section, const char *key)
- void al_set_config_value(ALLEGRO_CONFIG *config,const char *section, const char *key, const char *value)
- char const *al_get_first_config_section(ALLEGRO_CONFIG const *config,ALLEGRO_CONFIG_SECTION **iterator)
- char const *al_get_next_config_section(ALLEGRO_CONFIG_SECTION **iterator)
- char const *al_get_first_config_entry(ALLEGRO_CONFIG const *config,char const *section, ALLEGRO_CONFIG_ENTRY **iterator)
- char const *al_get_next_config_entry(ALLEGRO_CONFIG_ENTRY **iterator)
- ALLEGRO_CONFIG *al_merge_config(const ALLEGRO_CONFIG *cfg1,const ALLEGRO_CONFIG *cfg2)
- void al_merge_config_into(ALLEGRO_CONFIG *master, const ALLEGRO_CONFIG *add)
- ALLEGRO_DISPLAY *al_create_display(int w, int h)

- void al_destroy_display(ALLEGRO_DISPLAY *display)
- int al_get_new_display_flags(void)
- void al_set_new_display_flags(int flags)
- int al_get_new_display_option(int option, int *importance)
- void al_set_new_display_option(int option, int value, int importance)
- void al_reset_new_display_options(void)
- void al_get_new_window_position(int *x, int *y)
- void al_set_new_window_position(int x, int y)
- int al_get_new_display_refresh_rate(void)
- void al_set_new_display_refresh_rate(int refresh_rate)
- ALLEGRO_EVENT_SOURCE *al_get_display_event_source(ALLEGRO_DISPLAY *display)
- ALLEGRO_BITMAP *al_get_backbuffer(ALLEGRO_DISPLAY *display)
- void al_flip_display(void)
- void al_update_display_region(int x, int y, int width, int height)
- bool al_wait_for_vsync(void)
- int al_get_display_width(ALLEGRO_DISPLAY *display)
- int al_get_display_height(ALLEGRO_DISPLAY *display)
- bool al_resize_display(ALLEGRO_DISPLAY *display, int width, int height)
- bool al_acknowledge_resize(ALLEGRO_DISPLAY *display)
- void al_get_window_position(ALLEGRO_DISPLAY *display, int *x, int *y)
- void al_set_window_position(ALLEGRO_DISPLAY *display, int x, int y)
- int al_get_display_flags(ALLEGRO_DISPLAY *display)
- bool al_set_display_flag(ALLEGRO_DISPLAY *display, int flag, bool onoff)
- int al_get_display_option(ALLEGRO_DISPLAY *display, int option)
- int al_get_display_format(ALLEGRO_DISPLAY *display)
- int al_get_display_refresh_rate(ALLEGRO_DISPLAY *display)
- void al_set_window_title(ALLEGRO_DISPLAY *display, const char *title)
- void al_set_display_icon(ALLEGRO_DISPLAY *display, ALLEGRO_BITMAP *icon)
- void al_set_display_icons(ALLEGRO_DISPLAY *display,int num_icons, ALLEGRO_BITMAP *icons[])
- bool al_inhibit_screensaver(bool inhibit)
- void al_acknowledge_drawing_halt(ALLEGRO_DISPLAY *display)
- void al_acknowledge_drawing_resume(ALLEGRO_DISPLAY *display)
- int al_get_display_orientation(ALLEGRO_DISPLAY* display)
- void al_set_display_option(ALLEGRO_DISPLAY *display, int option, int value)
- bool al_get_window_constraints(ALLEGRO_DISPLAY *display,int *min_w, int *min_h, int *max_w, int *max_h)

- bool al_set_window_constraints(ALLEGRO_DISPLAY *display,int min_w, int min_h, int max_w, int max_h)
- ALLEGRO_EVENT_QUEUE *al_create_event_queue(void)
- void al_destroy_event_queue(ALLEGRO_EVENT_QUEUE *queue)
- void al_register_event_source(ALLEGRO_EVENT_QUEUE *queue,ALLEGRO_EVENT_SOURCE *source)
- void al_unregister_event_source(ALLEGRO_EVENT_QUEUE *queue,ALLEGRO_EVENT_SOURCE *source)
- bool al_is_event_queue_empty(ALLEGRO_EVENT_QUEUE *queue)
- bool al_get_next_event(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event)
- bool al_peek_next_event(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event)
- bool al_drop_next_event(ALLEGRO_EVENT_QUEUE *queue)
- void al_flush_event_queue(ALLEGRO_EVENT_QUEUE *queue)
- void al_wait_for_event(ALLEGRO_EVENT_QUEUE *queue, ALLEGRO_EVENT *ret_event)
- bool al_wait_for_event_timed(ALLEGRO_EVENT_QUEUE *queue,ALLEGRO_EVENT *ret_event, float secs)
- bool al_wait_for_event_until(ALLEGRO_EVENT_QUEUE *queue,ALLEGRO_EVENT *ret_event, ALLEGRO_TIMEOUT *timeout)
- void al_init_user_event_source(ALLEGRO_EVENT_SOURCE *src)
- void al_destroy_user_event_source(ALLEGRO_EVENT_SOURCE *src)
- intptr_t al_get_event_source_data(const ALLEGRO_EVENT_SOURCE *source)
- void al_set_event_source_data(ALLEGRO_EVENT_SOURCE *source, intptr_t data)
- ALLEGRO_FILE *al_fopen(const char *path, const char *mode)
- ALLEGRO_FILE *al_fopen_interface(const ALLEGRO_FILE_INTERFACE *drv,const char *path, const char *mode)
- ALLEGRO_FILE *al_fopen_slice(ALLEGRO_FILE *fp, size_t initial_size, const char *mode)
- void al_fclose(ALLEGRO_FILE *f)
- size_t al_fread(ALLEGRO_FILE *f, void *ptr, size_t size)
- size_t al_fwrite(ALLEGRO_FILE *f, const void *ptr, size_t size)
- bool al_fflush(ALLEGRO_FILE *f)
- int64_t al_ftell(ALLEGRO_FILE *f)
- bool al_fseek(ALLEGRO_FILE *f, int64_t offset, int whence)
- bool al_feof(ALLEGRO_FILE *f)
- bool al_ferror(ALLEGRO_FILE *f)
- void al_fclearerr(ALLEGRO_FILE *f)
- int al_fungetc(ALLEGRO_FILE *f, int c)
- int64_t al_fsize(ALLEGRO_FILE *f)
- int al_fgetc(ALLEGRO_FILE *f)
- int al_fputc(ALLEGRO_FILE *f, int c)

- int16_t al_fread16le(ALLEGRO_FILE *f)
- int16_t al_fread16be(ALLEGRO_FILE *f)
- size_t al_fwrite16le(ALLEGRO_FILE *f, int16_t w)
- size_t al_fwrite16be(ALLEGRO_FILE *f, int16_t w)
- int32_t al_fread32le(ALLEGRO_FILE *f)
- int32_t al_fread32be(ALLEGRO_FILE *f)
- size_t al_fwrite32le(ALLEGRO_FILE *f, int32_t l)
- size_t al_fwrite32be(ALLEGRO_FILE *f, int32_t l)
- char *al_fgets(ALLEGRO_FILE *f, char * const buf, size_t max)
- ALLEGRO_USTR *al_fget_ustr(ALLEGRO_FILE *f)
- int al_fputs(ALLEGRO_FILE *f, char const *p)
- ALLEGRO_FILE *al_fopen_fd(int fd, const char *mode)
- ALLEGRO_FILE *al_make_temp_file(const char *template, ALLEGRO_PATH **ret_path)
- void al_set_new_file_interface(const ALLEGRO_FILE_INTERFACE *file_interface
- void al_set_standard_file_interface(void)
- const ALLEGRO_FILE_INTERFACE *al_get_new_file_interface(void)
- ALLEGRO_FILE *al_create_file_handle(const ALLEGRO_FILE_INTERFACE *drv,void *userdata)
- void *al_get_file_userdata(ALLEGRO_FILE *f)
- ALLEGRO_FS_ENTRY *al_create_fs_entry(const char *path)
- void al_destroy_fs_entry(ALLEGRO_FS_ENTRY *fh)
- const char *al_get_fs_entry_name(ALLEGRO_FS_ENTRY *e)
- bool al_update_fs_entry(ALLEGRO_FS_ENTRY *e)
- uint32_t al_get_fs_entry_mode(ALLEGRO_FS_ENTRY *e)
- time_t al_get_fs_entry_atime(ALLEGRO_FS_ENTRY *e)
- time_t al_get_fs_entry_ctime(ALLEGRO_FS_ENTRY *e)
- time_t al_get_fs_entry_mtime(ALLEGRO_FS_ENTRY *e)
- off_t al_get_fs_entry_size(ALLEGRO_FS_ENTRY *e)
- bool al_fs_entry_exists(ALLEGRO_FS_ENTRY *e)
- bool al_remove_fs_entry(ALLEGRO_FS_ENTRY *e)
- bool al_filename_exists(const char *path)
- bool al_remove_filename(const char *path)
- bool al_open_directory(ALLEGRO_FS_ENTRY *e)
- ALLEGRO_FS_ENTRY *al_read_directory(ALLEGRO_FS_ENTRY *e)
- bool al_close_directory(ALLEGRO_FS_ENTRY *e)
- char *al_get_current_directory(void)
- bool al_change_directory(const char *path)

- bool al_make_directory(const char *path)
- ALLEGRO_FILE *al_open_fs_entry(ALLEGRO_FS_ENTRY *e, const char *mode)
- void al_set_fs_interface(const ALLEGRO_FS_INTERFACE *fs_interface)
- void al_set_standard_fs_interface(void)
- const ALLEGRO_FS_INTERFACE *al_get_fs_interface(void)
- al_fixed al_itofix(int x);
- int al_fixtoi(al_fixed x);
- int al_fixfloor(al_fixed x);
- int al_fixceil(al_fixed x);
- al_fixed al_ftofix(double x);
- double al_fixtof(al_fixed x);
- al_fixed al_fixmul(al_fixed x, al_fixed y);
- al_fixed al_fixdiv(al_fixed x, al_fixed y);
- al_fixed al_fixadd(al_fixed x, al_fixed y);
- al_fixed al_fixsub(al_fixed x, al_fixed y);
- al_fixed al_fixsin(al_fixed x);
- al_fixed al_fixcos(al_fixed x);
- al_fixed al_fixtan(al_fixed x);
- al_fixed al_fixasin(al_fixed x);
- al_fixed al_fixacos(al_fixed x);
- al_fixed al_fixatan(al_fixed x)
- al_fixed al_fixatan2(al_fixed y, al_fixed x)
- al_fixed al_fixsqrt(al_fixed x)
- al_fixed al_fixhypot(al_fixed x, al_fixed y)
- ALLEGRO_DISPLAY_MODE *al_get_display_mode(int index, ALLEGRO_DISPLAY_MODE *mode)
- int al_get_num_display_modes(void)
- ALLEGRO_COLOR al_map_rgb(unsigned char r, unsigned char g, unsigned char b)
- ALLEGRO_COLOR al_map_rgb_f(float r, float g, float b)
- ALLEGRO_COLOR al_map_rgba(unsigned char r, unsigned char g, unsigned char b, unsigned char a)
- ALLEGRO_COLOR al_map_rgba_f(float r, float g, float b, float a)
- void al_unmap_rgb(ALLEGRO_COLOR color,unsigned char *r, unsigned char *g, unsigned char *b)
- void al_unmap_rgb_f(ALLEGRO_COLOR color, float *r, float *g, float *b)
- void al_unmap_rgba(ALLEGRO_COLOR color,unsigned char *r, unsigned char *g, unsigned char *b, unsigned char *a)
- void al_unmap_rgba_f(ALLEGRO_COLOR color,float *r, float *g, float *b, float *a)
- int al_get_pixel_size(int format)

- int al_get_pixel_format_bits(int format)
- ALLEGRO_LOCKED_REGION *al_lock_bitmap(ALLEGRO_BITMAP *bitmap,int format, int flags)
- ALLEGRO_LOCKED_REGION *al_lock_bitmap_region(ALLEGRO_BITMAP *bitmap,int x, int y, int width, int height, int format, int flags)
- void al_unlock_bitmap(ALLEGRO_BITMAP *bitmap)
- ALLEGRO_BITMAP *al_create_bitmap(int w, int h)
- ALLEGRO_BITMAP *al_create_sub_bitmap(ALLEGRO_BITMAP *parent,int x, int y, int w, int h)
- ALLEGRO_BITMAP *al_clone_bitmap(ALLEGRO_BITMAP *bitmap)
- void al_destroy_bitmap(ALLEGRO_BITMAP *bitmap)
- int al_get_new_bitmap_flags(void)
- int al_get_new_bitmap_format(void)
- void al_set_new_bitmap_flags(int flags)
- void al_add_new_bitmap_flag(int flag)
- void al_set_new_bitmap_format(int format)
- int al_get_bitmap_flags(ALLEGRO_BITMAP *bitmap)
- int al_get_bitmap_format(ALLEGRO_BITMAP *bitmap)
- int al_get_bitmap_height(ALLEGRO_BITMAP *bitmap)
- int al_get_bitmap_width(ALLEGRO_BITMAP *bitmap)
- ALLEGRO_COLOR al_get_pixel(ALLEGRO_BITMAP *bitmap, int x, int y)
- bool al_is_bitmap_locked(ALLEGRO_BITMAP *bitmap)
- bool al_is_compatible_bitmap(ALLEGRO_BITMAP *bitmap)
- bool al_is_sub_bitmap(ALLEGRO_BITMAP *bitmap)
- ALLEGRO_BITMAP *al_get_parent_bitmap(ALLEGRO_BITMAP *bitmap)
- void al_clear_to_color(ALLEGRO_COLOR color)
- void al_draw_bitmap(ALLEGRO_BITMAP *bitmap, float dx, float dy, int flags)
- void al_draw_tinted_bitmap(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR tint,float dx, float dy, int flags)
- void al_draw_bitmap_region(ALLEGRO_BITMAP *bitmap,float sx, float sy, float sw, float sh, float dx, float dy, int flags)
- void al_draw_tinted_bitmap_region(ALLEGRO_BITMAP *bitmap,ALLEGRO_COLOR tint,float sx, float sy, float sw, float sh, float dx, float dy,int flags)
- void al_draw_pixel(float x, float y, ALLEGRO_COLOR color)
- void al_draw_rotated_bitmap(ALLEGRO_BITMAP *bitmap,float cx, float cy, float dx, float dy, float angle, int flags)
- void al_draw_tinted_rotated_bitmap(ALLEGRO_BITMAP *bitmap,ALLEGRO_COLOR tint,float cx, float cy, float dx, float dy, float angle, int flags)
- void al_draw_scaled_rotated_bitmap(ALLEGRO_BITMAP *bitmap,float cx, float cy, float dx, float dy, float xscale, float yscale,float angle, int flags)

- void al_draw_tinted_scaled_rotated_bitmap(ALLEGRO_BITMAP *bitmap,ALLEGRO_COLOR tint,float cx, float cy, float dx, float dy, float xscale, float yscale,float angle, int flags)
- void al_draw_tinted_scaled_rotated_bitmap_region(ALLEGRO_BITMAP *bitmap,float sx, float sy, float sw, float sh,ALLEGRO_COLOR tint,float cx, float cy, float dx, float dy, float xscale, float yscale,float angle, int flags)
- void al_draw_scaled_bitmap(ALLEGRO_BITMAP *bitmap,float sx, float sy, float sw, float sh,float dx, float dy, float dw, float dh, int flags)
- void al_draw_tinted_scaled_bitmap(ALLEGRO_BITMAP *bitmap,ALLEGRO_COLOR tint,float sx, float sy, float sw, float sh,float dx, float dy, float dw, float dh, int flags)
- ALLEGRO_BITMAP *al_get_target_bitmap(void)
- void al_put_pixel(int x, int y, ALLEGRO_COLOR color)
- void al_put_blended_pixel(int x, int y, ALLEGRO_COLOR color)
- void al_set_target_bitmap(ALLEGRO_BITMAP *bitmap)
- void al_set_target_backbuffer(ALLEGRO_DISPLAY *display)
- ALLEGRO_DISPLAY *al_get_current_display(void)
- void al_get_blender(int *op, int *src, int *dst)
- void al_get_separate_blender(int *op, int *src, int *dst,int *alpha_op, int *alpha_src, int *alpha_dst)
- void al_set_blender(int op, int src, int dst)
- void al_set_separate_blender(int op, int src, int dst,int alpha_op, int alpha_src, int alpha_dst)
- void al_get_clipping_rectangle(int *x, int *y, int *w, int *h)
- void al_set_clipping_rectangle(int x, int y, int width, int height)
- void al_reset_clipping_rectangle(void)
- void al_convert_mask_to_alpha(ALLEGRO_BITMAP *bitmap, ALLEGRO_COLOR mask_color)
- void al_hold_bitmap_drawing(bool hold)
- bool al_is_bitmap_drawing_held(void)
- ALLEGRO_BITMAP *al_load_bitmap_f(ALLEGRO_FILE *fp, const char *ident)
- bool al_save_bitmap(const char *filename, ALLEGRO_BITMAP *bitmap)
- bool al_save_bitmap_f(ALLEGRO_FILE *fp, const char *ident,ALLEGRO_BITMAP *bitmap)
- bool al_install_joystick(void)
- void al_uninstall_joystick(void)
- bool al_is_joystick_installed(void)
- bool al_reconfigure_joysticks(void)
- int al_get_num_joysticks(void)
- ALLEGRO_JOYSTICK * al_get_joystick(int num)
- void al_release_joystick(ALLEGRO_JOYSTICK *joy)
- bool al_get_joystick_active(ALLEGRO_JOYSTICK *joy)
- const char *al_get_joystick_name(ALLEGRO_JOYSTICK *joy)
- const char *al_get_joystick_stick_name(ALLEGRO_JOYSTICK *joy, int stick)

- const char *al_get_joystick_axis_name(ALLEGRO_JOYSTICK *joy, int stick, int axis)
- const char *al_get_joystick_button_name(ALLEGRO_JOYSTICK *joy, int button)
- int al_get_joystick_stick_flags(ALLEGRO_JOYSTICK *joy, int stick)
- int al_get_joystick_num_sticks(ALLEGRO_JOYSTICK *joy)
- int al_get_joystick_num_axes(ALLEGRO_JOYSTICK *joy, int stick)
- int al_get_joystick_num_buttons(ALLEGRO_JOYSTICK *joy)
- void al_get_joystick_state(ALLEGRO_JOYSTICK *joy, ALLEGRO_JOYSTICK_STATE *ret_state)
- ALLEGRO_EVENT_SOURCE *al_get_joystick_event_source(void)
- bool al_install_keyboard(void)
- bool al_is_keyboard_installed(void)
- void al_uninstall_keyboard(void)
- void al_get_keyboard_state(ALLEGRO_KEYBOARD_STATE *ret_state)
- bool al_key_down(const ALLEGRO_KEYBOARD_STATE *state, int keycode)
- const char *al_keycode_to_name(int keycode)
- bool al_set_keyboard_leds(int leds)
- ALLEGRO_EVENT_SOURCE *al_get_keyboard_event_source(void)
- void *al_malloc_with_context(size_t n,int line, const char *file, const char *func)
- void al_free_with_context(void *ptr, int line, const char *file, const char *func)
- void *al_realloc_with_context(void *ptr, size_t n,int line, const char *file, const char *func)
- void *al_calloc_with_context(size_t count, size_t n,int line, const char *file, const char *func)
- void al_set_memory_interface(ALLEGRO_MEMORY_INTERFACE *memory_interface)
- int al_get_new_display_adapter(void)
- void al_set_new_display_adapter(int adapter)
- bool al_get_monitor_info(int adapter, ALLEGRO_MONITOR_INFO *info)
- int al_get_num_video_adapters(void)
- bool al_install_mouse(void)
- bool al_is_mouse_installed(void)
- void al_uninstall_mouse(void)
- unsigned int al_get_mouse_num_axes(void)
- unsigned int al_get_mouse_num_buttons(void)
- void al_get_mouse_state(ALLEGRO_MOUSE_STATE *ret_state)
- int al_get_mouse_state_axis(const ALLEGRO_MOUSE_STATE *state, int axis)
- bool al_mouse_button_down(const ALLEGRO_MOUSE_STATE *state, int button)
- bool al_set_mouse_xy(ALLEGRO_DISPLAY *display, int x, int y)
- bool al_set_mouse_z(int z)
- bool al_set_mouse_w(int w)

- bool al_set_mouse_axis(int which, int value)
- ALLEGRO_EVENT_SOURCE *al_get_mouse_event_source(void)
- ALLEGRO_MOUSE_CURSOR *al_create_mouse_cursor(ALLEGRO_BITMAP *bmp,int x_focus, int y_focus)
- void al_destroy_mouse_cursor(ALLEGRO_MOUSE_CURSOR *cursor)
- bool al_set_mouse_cursor(ALLEGRO_DISPLAY *display, ALLEGRO_MOUSE_CURSOR *cursor)
- bool al_set_system_mouse_cursor(ALLEGRO_DISPLAY *display,ALLEGRO_SYSTEM_MOUSE_CURSOR cursor_id)
- bool al_get_mouse_cursor_position(int *ret_x, int *ret_y)
- bool al_hide_mouse_cursor(ALLEGRO_DISPLAY *display)
- bool al_show_mouse_cursor(ALLEGRO_DISPLAY *display)
- bool al_grab_mouse(ALLEGRO_DISPLAY *display)
- bool al_ungrab_mouse(void)
- ALLEGRO_PATH *al_create_path(const char *str)
- ALLEGRO_PATH *al_create_path_for_directory(const char *str)
- void al_destroy_path(ALLEGRO_PATH *path)
- ALLEGRO_PATH *al_clone_path(const ALLEGRO_PATH *path)
- bool al_join_paths(ALLEGRO_PATH *path, const ALLEGRO_PATH *tail)
- bool al_rebase_path(const ALLEGRO_PATH *head, ALLEGRO_PATH *tail)
- const char *al_get_path_drive(const ALLEGRO_PATH *path)
- int al_get_path_num_components(const ALLEGRO_PATH *path)
- const char *al_get_path_component(const ALLEGRO_PATH *path, int i)
- const char *al_get_path_tail(const ALLEGRO_PATH *path)
- const char *al_get_path_filename(const ALLEGRO_PATH *path)
- const char *al_get_path_basename(const ALLEGRO_PATH *path)
- const char *al_get_path_extension(const ALLEGRO_PATH *path)
- void al_set_path_drive(ALLEGRO_PATH *path, const char *drive)
- void al_append_path_component(ALLEGRO_PATH *path, const char *s)
- void al_insert_path_component(ALLEGRO_PATH *path, int i, const char *s)
- void al_replace_path_component(ALLEGRO_PATH *path, int i, const char *s)
- void al_remove_path_component(ALLEGRO_PATH *path, int i)
- void al_drop_path_tail(ALLEGRO_PATH *path)
- void al_set_path_filename(ALLEGRO_PATH *path, const char *filename)
- bool al_set_path_extension(ALLEGRO_PATH *path, char const *extension)
- const char *al_path_cstr(const ALLEGRO_PATH *path, char delim)
- bool al_make_path_canonical(ALLEGRO_PATH *path)
- void al_restore_state(ALLEGRO_STATE const *state)

- void al_store_state(ALLEGRO_STATE *state, int flags
- int al_get_errno(void)
- void al_set_errno(int errnum)
- void al_uninstall_system(void)
- bool al_is_system_installed(void)
- uint32_t al_get_allegro_version(void)
- ALLEGRO_PATH *al_get_standard_path(int id)
- void al_set_exe_name(char const *path)
- void al_set_app_name(const char *app_name)
- void al_set_org_name(const char *org_name)
- const char *al_get_app_name(void)
- const char *al_get_org_name(void)
- ALLEGRO_CONFIG *al_get_system_config(void)
- ALLEGRO_THREAD *al_create_thread(void)
- void al_run_detached_thread(void)
- void al_start_thread(ALLEGRO_THREAD *thread)
- void al_join_thread(ALLEGRO_THREAD *thread, void **ret_value)
- void al_set_thread_should_stop(ALLEGRO_THREAD *thread)
- bool al_get_thread_should_stop(ALLEGRO_THREAD *thread)
- void al_destroy_thread(ALLEGRO_THREAD *thread)
- ALLEGRO_MUTEX *al_create_mutex(void)
- ALLEGRO_MUTEX *al_create_mutex_recursive(void)
- void al_lock_mutex(ALLEGRO_MUTEX *mutex)
- void al_unlock_mutex(ALLEGRO_MUTEX *mutex)
- void al_destroy_mutex(ALLEGRO_MUTEX *mutex)
- ALLEGRO_COND *al_create_cond(void)
- void al_destroy_cond(ALLEGRO_COND *cond)
- void al_wait_cond(ALLEGRO_COND *cond, ALLEGRO_MUTEX *mutex)
- int al_wait_cond_until(ALLEGRO_COND *cond, ALLEGRO_MUTEX *mutex,const ALLEGRO_TIMEOUT *timeout)
- void al_broadcast_cond(ALLEGRO_COND *cond)
- void al_signal_cond(ALLEGRO_COND *cond)
- double al_get_time(void)
- void al_init_timeout(ALLEGRO_TIMEOUT *timeout, double seconds)
- void al_rest(double seconds)
- ALLEGRO_TIMER *al_create_timer(double speed_secs)

- void al_start_timer(ALLEGRO_TIMER *timer)
- void al_stop_timer(ALLEGRO_TIMER *timer)
- bool al_get_timer_started(const ALLEGRO_TIMER *timer)
- void al_destroy_timer(ALLEGRO_TIMER *timer)
- int64_t al_get_timer_count(const ALLEGRO_TIMER *timer)
- void al_set_timer_count(ALLEGRO_TIMER *timer, int64_t new_count)
- void al_add_timer_count(ALLEGRO_TIMER *timer, int64_t diff)
- double al_get_timer_speed(const ALLEGRO_TIMER *timer)
- void al_set_timer_speed(ALLEGRO_TIMER *timer, double new_speed_secs)
- ALLEGRO_EVENT_SOURCE *al_get_timer_event_source(ALLEGRO_TIMER *timer)
- void al_copy_transform(ALLEGRO_TRANSFORM *dest, const ALLEGRO_TRANSFORM *src)
- void al_use_transform(const ALLEGRO_TRANSFORM *trans)
- const ALLEGRO_TRANSFORM *al_get_current_transform(void)
- void al_invert_transform(ALLEGRO_TRANSFORM *trans)
- int al_check_inverse(const ALLEGRO_TRANSFORM *trans, float tol)
- void al_identity_transform(ALLEGRO_TRANSFORM *trans)
- void al_build_transform(ALLEGRO_TRANSFORM *trans, float x, float y,float sx, float sy, float theta)
- void al_translate_transform(ALLEGRO_TRANSFORM *trans, float x, float y)
- void al_rotate_transform(ALLEGRO_TRANSFORM *trans, float theta)
- void al_scale_transform(ALLEGRO_TRANSFORM *trans, float sx, float sy)
- void al_transform_coordinates(const ALLEGRO_TRANSFORM *trans, float *x, float *y)
- void al_compose_transform(ALLEGRO_TRANSFORM *trans, const ALLEGRO_TRANSFORM *other)
- ALLEGRO_USTR *al_ustr_new(const char *s)
- ALLEGRO_USTR *al_ustr_new_from_buffer(const char *s, size_t size)
- void al_ustr_free(ALLEGRO_USTR *us)
- const char *al_cstr(const ALLEGRO_USTR *us)
- void al_ustr_to_buffer(const ALLEGRO_USTR *us, char *buffer, int size)
- char *al_cstr_dup(const ALLEGRO_USTR *us)
- ALLEGRO_USTR *al_ustr_dup(const ALLEGRO_USTR *us)
- ALLEGRO_USTR *al_ustr_dup_substr(const ALLEGRO_USTR *us, int start_pos,int end_pos)
- const ALLEGRO_USTR *al_ustr_empty_string(void)
- const ALLEGRO_USTR *al_ref_cstr(ALLEGRO_USTR_INFO *info, const char *s)
- const ALLEGRO_USTR *al_ref_buffer(ALLEGRO_USTR_INFO *info, const char *s, size_t size)
- const ALLEGRO_USTR *al_ref_ustr(ALLEGRO_USTR_INFO *info, const ALLEGRO_USTR *us,int start_pos, int end_pos)
- size_t al_ustr_size(const ALLEGRO_USTR *us)

- size_t al_ustr_length(const ALLEGRO_USTR *us)
- int al_ustr_offset(const ALLEGRO_USTR *us, int index)
- bool al_ustr_next(const ALLEGRO_USTR *us, int *pos)
- bool al_ustr_prev(const ALLEGRO_USTR *us, int *pos)
- int32_t al_ustr_get(const ALLEGRO_USTR *ub, int pos)
- int32_t al_ustr_get_next(const ALLEGRO_USTR *us, int *pos)
- int32_t al_ustr_prev_get(const ALLEGRO_USTR *us, int *pos)
- bool al_ustr_insert(ALLEGRO_USTR *us1, int pos, const ALLEGRO_USTR *us2)
- bool al_ustr_insert_cstr(ALLEGRO_USTR *us, int pos, const char *s)
- size_t al_ustr_insert_chr(ALLEGRO_USTR *us, int pos, int32_t c)
- bool al_ustr_append(ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)
- bool al_ustr_append_cstr(ALLEGRO_USTR *us, const char *s)
- size_t al_ustr_append_chr(ALLEGRO_USTR *us, int32_t c)
- bool al_ustr_remove_chr(ALLEGRO_USTR *us, int pos)
- bool al_ustr_remove_range(ALLEGRO_USTR *us, int start_pos, int end_pos)
- bool al_ustr_truncate(ALLEGRO_USTR *us, int start_pos)
- bool al_ustr_ltrim_ws(ALLEGRO_USTR *us)
- bool al_ustr_rtrim_ws(ALLEGRO_USTR *us)
- bool al_ustr_trim_ws(ALLEGRO_USTR *us)
- bool al_ustr_assign(ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)
- bool al_ustr_assign_substr(ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2,int start_pos, int end_pos)
- bool al_ustr_assign_cstr(ALLEGRO_USTR *us1, const char *s)
- size_t al_ustr_set_chr(ALLEGRO_USTR *us, int start_pos, int32_t c)
- bool al_ustr_replace_range(ALLEGRO_USTR *us1, int start_pos1, int end_pos1,const ALLEGRO_USTR *us2)
- int al_ustr_find_chr(const ALLEGRO_USTR *us, int start_pos, int32_t c)
- int al_ustr_rfind_chr(const ALLEGRO_USTR *us, int end_pos, int32_t c)
- int al_ustr_find_set(const ALLEGRO_USTR *us, int start_pos,const ALLEGRO_USTR *accept)
- int al_ustr_find_set_cstr(const ALLEGRO_USTR *us, int start_pos,const char *accept)
- int al_ustr_find_cset(const ALLEGRO_USTR *us, int start_pos,const ALLEGRO_USTR *reject)
- int al_ustr_find_cset_cstr(const ALLEGRO_USTR *us, int start_pos,const char *reject)
- int al_ustr_find_str(const ALLEGRO_USTR *haystack, int start_pos,const ALLEGRO_USTR *needle)
- int al_ustr_find_cstr(const ALLEGRO_USTR *haystack, int start_pos,const char *needle)
- int al_ustr_rfind_str(const ALLEGRO_USTR *haystack, int end_pos,const ALLEGRO_USTR *needle)
- int al_ustr_rfind_cstr(const ALLEGRO_USTR *haystack, int end_pos,const char *needle)
- bool al_ustr_find_replace(ALLEGRO_USTR *us, int start_pos,const ALLEGRO_USTR *find, const ALLEGRO_USTR *replace)

- bool al_ustr_find_replace_cstr(ALLEGRO_USTR *us, int start_pos,const char *find, const char *replace)
- int al_ustr_compare(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)
- int al_ustr_ncompare(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2, int n)
- bool al_ustr_equal(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)
- bool al_ustr_has_prefix(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)
- bool al_ustr_has_prefix_cstr(const ALLEGRO_USTR *us1, const char *s2)
- bool al_ustr_has_suffix(const ALLEGRO_USTR *us1, const ALLEGRO_USTR *us2)
- bool al_ustr_has_suffix_cstr(const ALLEGRO_USTR *us1, const char *s2)
- ALLEGRO_USTR *al_ustr_new_from_utf16(uint16_t const *s)
- size_t al_ustr_size_utf16(const ALLEGRO_USTR *us)
- size_t al_ustr_encode_utf16(const ALLEGRO_USTR *us, uint16_t *s,size_t n)
- size_t al_utf8_width(int c)
- size_t al_utf8_encode(char s[], int32_t c)
- size_t al_utf16_width(int c)
- LPDIRECT3DDEVICE9 al_get_d3d_device(ALLEGRO_DISPLAY *display)
- LPDIRECT3DTEXTURE9 al_get_d3d_system_texture(ALLEGRO_BITMAP *bitmap)
- LPDIRECT3DTEXTURE9 al_get_d3d_video_texture(ALLEGRO_BITMAP *bitmap)
- bool al_have_d3d_non_pow2_texture_support(void)
- bool al_have_d3d_non_square_texture_support(void)
- void al_get_d3d_texture_position(ALLEGRO_BITMAP *bitmap, int *u, int *v)
- bool al_is_d3d_device_lost(ALLEGRO_DISPLAY *display)
- ALLEGRO_OGL_EXT_LIST *al_get_opengl_extension_list(void)
- void *al_get_opengl_proc_address(const char *name)
- GLuint al_get_opengl_texture(ALLEGRO_BITMAP *bitmap)
- void al_get_opengl_texture_size(ALLEGRO_BITMAP *bitmap, int *w, int *h)
- void al_get_opengl_texture_position(ALLEGRO_BITMAP *bitmap, int *u, int *v)
- GLuint al_get_opengl_fbo(ALLEGRO_BITMAP *bitmap)
- void al_remove_opengl_fbo(ALLEGRO_BITMAP *bitmap)
- bool al_have_opengl_extension(const char *extension);
- uint32_t al_get_opengl_version(void)
- int al_get_opengl_variant(void)
- void al_set_current_opengl_context(ALLEGRO_DISPLAY *display)
- bool al_install_audio(void)
- void al_uninstall_audio(void)
- bool al_is_audio_installed(void)
- bool al_reserve_samples(int reserve_samples)

- uint32_t al_get_allegro_audio_version(void)
- size_t al_get_audio_depth_size(ALLEGRO_AUDIO_DEPTH depth)
- size_t al_get_channel_count(ALLEGRO_CHANNEL_CONF conf)
- ALLEGRO_VOICE *al_create_voice(unsigned int freq,ALLEGRO_AUDIO_DEPTH depth, ALLEGRO_CHANNEL_CONF chan_conf)
- void al_destroy_voice(ALLEGRO_VOICE *voice)
- void al_detach_voice(ALLEGRO_VOICE *voice)
- bool al_attach_audio_stream_to_voice(ALLEGRO_AUDIO_STREAM *stream,ALLEGRO_VOICE *voice)
- bool al_attach_mixer_to_voice(ALLEGRO_MIXER *mixer, ALLEGRO_VOICE *voice)
- bool al_attach_sample_instance_to_voice(ALLEGRO_SAMPLE_INSTANCE *spl,ALLEGRO_VOICE *voice)
- unsigned int al_get_voice_frequency(const ALLEGRO_VOICE *voice)
- ALLEGRO_CHANNEL_CONF al_get_voice_channels(const ALLEGRO_VOICE *voice)
- ALLEGRO_AUDIO_DEPTH al_get_voice_depth(const ALLEGRO_VOICE *voice)
- bool al_get_voice_playing(const ALLEGRO_VOICE *voice)
- bool al_set_voice_playing(ALLEGRO_VOICE *voice, bool val)
- unsigned int al_get_voice_position(const ALLEGRO_VOICE *voice)
- bool al_set_voice_position(ALLEGRO_VOICE *voice, unsigned int val)
- ALLEGRO_SAMPLE *al_create_sample(void *buf, unsigned int samples,unsigned int freq, ALLEGRO_AUDIO_DEPTH depth,ALLEGRO_CHANNEL_CONF chan_conf, bool free_buf)
- void al_destroy_sample(ALLEGRO_SAMPLE *spl)
- bool al_play_sample(ALLEGRO_SAMPLE *spl, float gain, float pan, float speed,int loop, ALLEGRO_SAMPLE_ID *ret_id)
- void al_stop_sample(ALLEGRO_SAMPLE_ID *spl_id)
- void al_stop_samples(void)
- ALLEGRO_CHANNEL_CONF al_get_sample_channels(const ALLEGRO_SAMPLE *spl)
- ALLEGRO_AUDIO_DEPTH al_get_sample_depth(const ALLEGRO_SAMPLE *spl)
- unsigned int al_get_sample_frequency(const ALLEGRO_SAMPLE *spl)
- unsigned int al_get_sample_length(const ALLEGRO_SAMPLE *spl)
- void *al_get_sample_data(const ALLEGRO_SAMPLE *spl)
- ALLEGRO_SAMPLE_INSTANCE *al_create_sample_instance(ALLEGRO_SAMPLE *sample_data)
- void al_destroy_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_play_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_stop_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)
- ALLEGRO_CHANNEL_CONF al_get_sample_instance_channels(const ALLEGRO_SAMPLE_INSTANCE *spl)
- ALLEGRO_AUDIO_DEPTH al_get_sample_instance_depth(const ALLEGRO_SAMPLE_INSTANCE *spl)
- unsigned int al_get_sample_instance_frequency(const ALLEGRO_SAMPLE_INSTANCE *spl)

- unsigned int al_get_sample_instance_length(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample_instance_length(ALLEGRO_SAMPLE_INSTANCE *spl,unsigned int val)
- unsigned int al_get_sample_instance_position(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample_instance_position(ALLEGRO_SAMPLE_INSTANCE *spl,unsigned int val)
- float al_get_sample_instance_speed(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample_instance_speed(ALLEGRO_SAMPLE_INSTANCE *spl, float val)
- float al_get_sample_instance_gain(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample_instance_gain(ALLEGRO_SAMPLE_INSTANCE *spl, float val)
- float al_get_sample_instance_pan(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample_instance_pan(ALLEGRO_SAMPLE_INSTANCE *spl, float val)
- float al_get_sample_instance_time(const ALLEGRO_SAMPLE_INSTANCE *spl)
- ALLEGRO_PLAYMODE al_get_sample_instance_playmode(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample_instance_playmode(ALLEGRO_SAMPLE_INSTANCE *spl,ALLEGRO_PLAYMODE val)
- bool al_get_sample_instance_playing(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample_instance_playing(ALLEGRO_SAMPLE_INSTANCE *spl, bool val)
- bool al_get_sample_instance_attached(const ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_detach_sample_instance(ALLEGRO_SAMPLE_INSTANCE *spl)
- ALLEGRO_SAMPLE *al_get_sample(ALLEGRO_SAMPLE_INSTANCE *spl)
- bool al_set_sample(ALLEGRO_SAMPLE_INSTANCE *spl, ALLEGRO_SAMPLE *data)
- ALLEGRO_MIXER *al_create_mixer(unsigned int freq,ALLEGRO_AUDIO_DEPTH depth, ALLEGRO_CHANNEL_CONF chan_conf)
- void al_destroy_mixer(ALLEGRO_MIXER *mixer)
- ALLEGRO_MIXER *al_get_default_mixer(void)
- bool al_set_default_mixer(ALLEGRO_MIXER *mixer)
- bool al_restore_default_mixer(void)
- bool al_attach_mixer_to_mixer(ALLEGRO_MIXER *stream, ALLEGRO_MIXER *mixer)
- bool al_attach_sample_instance_to_mixer(ALLEGRO_SAMPLE_INSTANCE *spl,ALLEGRO_MIXER *mixer)
- bool al_attach_audio_stream_to_mixer(ALLEGRO_AUDIO_STREAM *stream, ALLEGRO_MIXER *mixer)
- unsigned int al_get_mixer_frequency(const ALLEGRO_MIXER *mixer)
- bool al_set_mixer_frequency(ALLEGRO_MIXER *mixer, unsigned int val)
- ALLEGRO_CHANNEL_CONF al_get_mixer_channels(const ALLEGRO_MIXER *mixer)
- ALLEGRO_AUDIO_DEPTH al_get_mixer_depth(const ALLEGRO_MIXER *mixer)
- float al_get_mixer_gain(const ALLEGRO_MIXER *mixer)
- bool al_set_mixer_gain(ALLEGRO_MIXER *mixer, float new_gain)
- ALLEGRO_MIXER_QUALITY al_get_mixer_quality(const ALLEGRO_MIXER *mixer)

- bool al_set_mixer_quality(ALLEGRO_MIXER *mixer, ALLEGRO_MIXER_QUALITY new_quality)
- bool al_get_mixer_playing(const ALLEGRO_MIXER *mixer)
- bool al_set_mixer_playing(ALLEGRO_MIXER *mixer, bool val)
- bool al_get_mixer_attached(const ALLEGRO_MIXER *mixer)
- bool al_detach_mixer(ALLEGRO_MIXER *mixer)
- void al_destroy_audio_stream(ALLEGRO_AUDIO_STREAM *stream)
- ALLEGRO_EVENT_SOURCE *al_get_audio_stream_event_source(ALLEGRO_AUDIO_STREAM *stream)
- void al_drain_audio_stream(ALLEGRO_AUDIO_STREAM *stream)
- bool al_rewind_audio_stream(ALLEGRO_AUDIO_STREAM *stream)
- unsigned int al_get_audio_stream_frequency(const ALLEGRO_AUDIO_STREAM *stream)
- ALLEGRO_CHANNEL_CONF al_get_audio_stream_channels(const ALLEGRO_AUDIO_STREAM *stream)
- ALLEGRO_AUDIO_DEPTH al_get_audio_stream_depth(const ALLEGRO_AUDIO_STREAM *stream)
- unsigned int al_get_audio_stream_length(const ALLEGRO_AUDIO_STREAM *stream)
- float al_get_audio_stream_speed(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_set_audio_stream_speed(ALLEGRO_AUDIO_STREAM *stream, float val)
- float al_get_audio_stream_gain(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_set_audio_stream_gain(ALLEGRO_AUDIO_STREAM *stream, float val)
- float al_get_audio_stream_pan(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_set_audio_stream_pan(ALLEGRO_AUDIO_STREAM *stream, float val)
- bool al_get_audio_stream_playing(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_set_audio_stream_playing(ALLEGRO_AUDIO_STREAM *stream, bool val)
- ALLEGRO_PLAYMODE al_get_audio_stream_playmode(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_set_audio_stream_playmode(ALLEGRO_AUDIO_STREAM *stream,ALLEGRO_PLAYMODE val)
- bool al_get_audio_stream_attached(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_detach_audio_stream(ALLEGRO_AUDIO_STREAM *stream)
- void *al_get_audio_stream_fragment(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_set_audio_stream_fragment(ALLEGRO_AUDIO_STREAM *stream, void *val)
- unsigned int al_get_audio_stream_fragments(const ALLEGRO_AUDIO_STREAM *stream)
- unsigned int al_get_available_audio_stream_fragments(const ALLEGRO_AUDIO_STREAM *stream)
- bool al_seek_audio_stream_secs(ALLEGRO_AUDIO_STREAM *stream, double time)
- double al_get_audio_stream_position_secs(ALLEGRO_AUDIO_STREAM *stream)
- double al_get_audio_stream_length_secs(ALLEGRO_AUDIO_STREAM *stream)
- bool al_set_audio_stream_loop_secs(ALLEGRO_AUDIO_STREAM *stream,double start, double end)
- ALLEGRO_SAMPLE *al_load_sample(const char *filename)
- ALLEGRO_SAMPLE *al_load_sample_f(ALLEGRO_FILE* fp, const char *ident)

- ALLEGRO_AUDIO_STREAM *al_load_audio_stream(const char *filename,size_t buffer_count, unsigned int samples)
- ALLEGRO_AUDIO_STREAM *al_load_audio_stream_f(ALLEGRO_FILE* fp, const char *ident,size_t buffer_count, unsigned int samples)
- bool al_save_sample(const char *filename, ALLEGRO_SAMPLE *spl)
- bool al_save_sample_f(ALLEGRO_FILE *fp, const char *ident, ALLEGRO_SAMPLE *spl)
- bool al_init_acodec_addon(void)
- uint32_t al_get_allegro_acodec_version(void)
- ALLEGRO_COLOR al_color_cmyk(float c, float m, float y, float k)
- void al_color_cmyk_to_rgb(float cyan, float magenta, float yellow,float key, float *red, float *green, float *blue)
- ALLEGRO_COLOR al_color_hsl(float h, float s, float l)
- void al_color_hsl_to_rgb(float hue, float saturation, float lightness,float *red, float *green, float *blue)
- ALLEGRO_COLOR al_color_hsv(float h, float s, float v)
- void al_color_hsv_to_rgb(float hue, float saturation, float value,float *red, float *green, float *blue)
- ALLEGRO_COLOR al_color_html(char const *string)
- void al_color_html_to_rgb(char const *string,float *red, float *green, float *blue)
- void al_color_rgb_to_html(float red, float green, float blue,char *string)
- ALLEGRO_COLOR al_color_name(char const *name)
- bool al_color_name_to_rgb(char const *name, float *r, float *g, float *b)
- void al_color_rgb_to_cmyk(float red, float green, float blue,float *cyan, float *magenta, float *yellow, float *key)
- void al_color_rgb_to_hsl(float red, float green, float blue,float *hue, float *saturation, float *lightness)
- void al_color_rgb_to_hsv(float red, float green, float blue,float *hue, float *saturation, float *value)
- char const *al_color_rgb_to_name(float r, float g, float b)
- void al_color_rgb_to_yuv(float red, float green, float blue,float *y, float *u, float *v)
- ALLEGRO_COLOR al_color_yuv(float y, float u, float v)
- void al_color_yuv_to_rgb(float y, float u, float v,float *red, float *green, float *blue)
- uint32_t al_get_allegro_color_version(void)
- void al_init_font_addon(void)
- void al_shutdown_font_addon(void)
- ALLEGRO_FONT *al_load_font(char const *filename, int size, int flags)
- void al_destroy_font(ALLEGRO_FONT *f)
- int al_get_font_ascent(const ALLEGRO_FONT *f)
- int al_get_font_descent(const ALLEGRO_FONT *f)
- int al_get_text_width(const ALLEGRO_FONT *f, const char *str)
- int al_get_ustr_width(const ALLEGRO_FONT *f, ALLEGRO_USTR const *ustr)
- void al_draw_text(const ALLEGRO_FONT *font,ALLEGRO_COLOR color, float x, float y, int flags,char const *text)

- void al_draw_ustr(const ALLEGRO_FONT *font,ALLEGRO_COLOR color, float x, float y, int flags,const ALLEGRO_USTR *ustr)
- void al_draw_justified_text(const ALLEGRO_FONT *font,ALLEGRO_COLOR color, float x1, float x2,float y, float diff, int flags, const char *text)
- void al_draw_justified_ustr(const ALLEGRO_FONT *font,ALLEGRO_COLOR color, float x1, float x2,float y, float diff, int flags, const ALLEGRO_USTR *ustr)
- void al_get_text_dimensions(const ALLEGRO_FONT *f,char const *text,int *bbx, int *bby, int *bbw, int *bbh)
- void al_get_ustr_dimensions(const ALLEGRO_FONT *f,ALLEGRO_USTR const *ustr,int *bbx, int *bby, int *bbw, int *bbh)
- uint32_t al_get_allegro_font_version(void)
- ALLEGRO_FONT *al_grab_font_from_bitmap(ALLEGRO_BITMAP *bmp,int ranges_n, const int ranges[])
- ALLEGRO_FONT *al_load_bitmap_font(const char *fname)
- ALLEGRO_FONT *al_create_builtin_font(void)
- bool al_init_ttf_addon(void)
- void al_shutdown_ttf_addon(void)
- ALLEGRO_FONT *al_load_ttf_font(char const *filename, int size, int flags)
- ALLEGRO_FONT *al_load_ttf_font_f(ALLEGRO_FILE *file,char const *filename, int size, int flags)
- ALLEGRO_FONT *al_load_ttf_font_stretch(char const *filename, int w, int h,int flags)
- ALLEGRO_FONT *al_load_ttf_font_stretch_f(ALLEGRO_FILE *file,char const *filename, int w, int h, int flags)
- uint32_t al_get_allegro_ttf_version(void)
- bool al_init_image_addon(void)
- void al_shutdown_image_addon(void)
- uint32_t al_get_allegro_image_version(void)
- ALLEGRO_FILE *al_open_memfile(void *mem, int64_t size, const char *mode)
- uint32_t al_get_allegro_memfile_version(void)
- bool al_init_native_dialog_addon(void)
- void al_shutdown_native_dialog_addon(void)
- ALLEGRO_FILECHOOSER *al_create_native_file_dialog(char const *initial_path,char const *title,char const *patterns,int mode)
- bool al_show_native_file_dialog(ALLEGRO_DISPLAY *display,ALLEGRO_FILECHOOSER *dialog)
- int al_get_native_file_dialog_count(const ALLEGRO_FILECHOOSER *dialog)
- const char *al_get_native_file_dialog_path(const ALLEGRO_FILECHOOSER *dialog, size_t i)
- void al_destroy_native_file_dialog(ALLEGRO_FILECHOOSER *dialog)
- int al_show_native_message_box(ALLEGRO_DISPLAY *display,char const *title, char const *heading, char const *text,char const *buttons, int flags)
- ALLEGRO_TEXTLOG *al_open_native_text_log(char const *title, int flags)
- void al_close_native_text_log(ALLEGRO_TEXTLOG *textlog)

- uint32_t al_get_allegro_native_dialog_version(void)
- void al_set_physfs_file_interface(void)
- uint32_t al_get_allegro_physfs_version(void)
- uint32_t al_get_allegro_primitives_version(void)
- bool al_init_primitives_addon(void)
- void al_shutdown_primitives_addon(void)
- void al_draw_line(float x1, float y1, float x2, float y2,ALLEGRO_COLOR color, float thickness)
- void al_draw_triangle(float x1, float y1, float x2, float y2,float x3, float y3, ALLEGRO_COLOR color, float thickness)
- void al_draw_filled_triangle(float x1, float y1, float x2, float y2,float x3, float y3, ALLEGRO_COLOR color)
- void al_draw_rectangle(float x1, float y1, float x2, float y2,ALLEGRO_COLOR color, float thickness)
- void al_draw_filled_rectangle(float x1, float y1, float x2, float y2,ALLEGRO_COLOR color)
- void al_draw_rounded_rectangle(float x1, float y1, float x2, float y2,float rx, float ry, ALLEGRO_COLOR color, float thickness)
- void al_draw_filled_rounded_rectangle(float x1, float y1, float x2, float y2,float rx, float ry, ALLEGRO_COLOR color)
- void al_calculate_arc(float* dest, int stride, float cx, float cy,float rx, float ry, float start_theta, float delta_theta, float thickness,int num_points)
- void al_draw_pieslice(float cx, float cy, float r, float start_theta,float delta_theta, ALLEGRO_COLOR color, float thickness)
- void al_draw_filled_pieslice(float cx, float cy, float r, float start_theta,float delta_theta, ALLEGRO_COLOR color)
- void al_draw_ellipse(float cx, float cy, float rx, float ry,ALLEGRO_COLOR color, float thickness)
- void al_draw_filled_ellipse(float cx, float cy, float rx, float ry,ALLEGRO_COLOR color)
- void al_draw_circle(float cx, float cy, float r, ALLEGRO_COLOR color,float thickness)
- void al_draw_filled_circle(float cx, float cy, float r, ALLEGRO_COLOR color)
- void al_draw_arc(float cx, float cy, float r, float start_theta,float delta_theta, ALLEGRO_COLOR color, float thickness)
- void al_draw_elliptical_arc(float cx, float cy, float rx, float ry, float start_theta,float delta_theta, ALLEGRO_COLOR color, float thickness)
- void al_draw_ribbon(const float *points, int points_stride, ALLEGRO_COLOR color,float thickness, int num_segments)
- int al_draw_prim(const void* vtxs, const ALLEGRO_VERTEX_DECL* decl,ALLEGRO_BITMAP* texture, int start, int end, int type)
- int al_draw_indexed_prim(const void* vtxs, const ALLEGRO_VERTEX_DECL* decl,ALLEGRO_BITMAP* texture, const int* indices, int num_vtx, int type)
- ALLEGRO_VERTEX_DECL* al_create_vertex_decl(const ALLEGRO_VERTEX_ELEMENT* elements, int stride)
- void al_destroy_vertex_decl(ALLEGRO_VERTEX_DECL* decl)
- ALLEGRO_SHADER *al_create_shader(ALLEGRO_SHADER_PLATFORM platform)

- bool al_attach_shader_source(ALLEGRO_SHADER *shader, ALLEGRO_SHADER_TYPE type,const char *source)
- bool al_attach_shader_source_file(ALLEGRO_SHADER *shader,ALLEGRO_SHADER_TYPE type, const char *filename)
- bool al_build_shader(ALLEGRO_SHADER *shader)
- const char *al_get_shader_log(ALLEGRO_SHADER *shader)
- ALLEGRO_SHADER_PLATFORM al_get_shader_platform(ALLEGRO_SHADER *shader)
- bool al_use_shader(ALLEGRO_SHADER *shader)
- void al_destroy_shader(ALLEGRO_SHADER *shader)
- bool al_set_shader_sampler(const char *name,ALLEGRO_BITMAP *bitmap, int unit)
- bool al_set_shader_matrix(const char *name,const ALLEGRO_TRANSFORM *matrix)
- bool al_set_shader_int(const char *name, int i)
- bool al_set_shader_float(const char *name, float f)
- bool al_set_shader_bool(const char *name, bool b)
- bool al_set_shader_int_vector(const char *name,int num_components, const int *i, int num_elems)
- bool al_set_shader_float_vector(const char *name,int num_components, const float *f, int num_elems)
- char const *al_get_default_shader_source(ALLEGRO_SHADER_PLATFORM platform,ALLEGRO_SHADER_TYPE type)

CHAPTER

EIGHTYFOUR

USING RINGLIBSDL

In this chapter we will learn about using RingLibSDL to create games based on the LibSDL, SDLImage, SDLTTF and SDLMixer libraries.

Tip: RingLibSDL is not distributed with the binary releases for desktop which uses RingAllegro

Note: To use RingLibSDL, Check ring/android/ringlibsdl folder.

84.1 Create Window

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
SDL_Delay(2000)
SDL_DestroyWindow(win)
SDL_Quit()
```

84.2 Display Image

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
bmp = SDL_LoadBMP("hello.bmp")
tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
SDL_RenderClear(ren)
SDL_RenderCopy2(ren,tex)
SDL_RenderPresent(ren)
```

(continues on next page)

(continued from previous page)

```
SDL_Delay(2000)
SDL_DestroyTexture(tex)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

84.3 Switch between two images

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
bmp = SDL_LoadBMP("hello.bmp")
tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
bmp = SDL_LoadBMP("hello2.bmp")
tex2 = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)

for x = 1 to 10 showtex(tex) showtex(tex2) next

SDL_DestroyTexture(tex)
SDL_DestroyTexture(tex2)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()

func showtex oTex
        SDL_RenderClear(ren)
        SDL_RenderCopy2(ren,oTex)
        SDL_RenderPresent(ren)
        SDL_Delay(200)
```

84.4 Draw Rectangle

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
SDL_RenderClear(ren)
rect = sdl_new_sdl_rect()
sdl_set_sdl_rect_x(rect,10)
```

(continues on next page)

(continued from previous page)

```
sdl_set_sdl_rect_y(rect,10)
sdl_set_sdl_rect_w(rect,100)
sdl_set_sdl_rect_h(rect,100)
SDL_SetRenderDrawColor(ren,255,255,255,255)
SDL_RenderDrawRect(ren,rect)
sdl_destroy_sdl_rect(rect)
SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

84.5 Display PNG Images

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
bmp = IMG_Load("hello3.png")
tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
SDL_RenderClear(ren)
SDL_RenderCopy2(ren,tex)
SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyTexture(tex)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

84.6 Use TTF Fonts

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )
SDL_RenderClear(ren)

TTF_Init()
font = TTF_OpenFont("pirulen.ttf", 16)
color = sdl_new_sdl_color()
sdl_set_sdl_color_r(color,0)
```

(continues on next page)

(continued from previous page)

```
sdl_set_sdl_color_g(color,255)
sdl_set_sdl_color_b(color,0)
text = TTF_RenderText_Solid(font,"Welcome to the Ring language",color)
surface = SDL_GetWindowSurface(win)
SDL_BlitSurface(text, NULL, surface, NULL)
SDL_UpdateWindowSurface(win)
SDL_Delay(2000)

SDL_Destroy_SDL_Color(color)
SDL_FreeSurface(text)
TTF_CloseFont(font)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

84.7 Display Transparent Images

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)

flags = IMG_INIT_JPG | IMG_INIT_PNG
IMG_Init(flags)

win = SDL_CreateWindow("Hello World!", 100, 100, 800, 600, SDL_WINDOW_SHOWN)
ren = SDL_CreateRenderer(win, -1, SDL_RENDERER_ACCELERATED | SDL_RENDERER_PRESENTVSYNC )

bmp = IMG_Load("stars.jpg")
tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
SDL_RenderClear(ren)
SDL_RenderCopy(ren,tex,NULL,NULL)
SDL_DestroyTexture(tex)

bmp = IMG_Load("player.png")
# Image - Set Transparent color (white)
myformat = sdl_get_sdl_surface_format(bmp)
white = SDL_MapRGB(myformat, 255, 255, 255)
SDL_SetColorKey(bmp, SDL_True, white)

tex = SDL_CreateTextureFromSurface(ren,bmp)
SDL_FreeSurface(bmp)
rect = sdl_new_sdl_rect()
sdl_set_sdl_rect_x(rect,0)
sdl_set_sdl_rect_y(rect,0)
sdl_set_sdl_rect_w(rect,100)
sdl_set_sdl_rect_h(rect,100)
SDL_RenderCopy(ren,tex,NULL,rect)
```

(continues on next page)

(continued from previous page)

```
SDL_SetTextureBlendMode(tex,2)
SDL_SetTextureAlphaMod(tex,255)
sdl_set_sdl_rect_x(rect,200)
sdl_set_sdl_rect_y(rect,200)
sdl_set_sdl_rect_w(rect,100)
sdl_set_sdl_rect_h(rect,100)
SDL_RenderCopy(ren,tex,NULL,rect)

SDL_DestroyTexture(tex)
SDL_Destroy_SDL_Rect(rect)

SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

84.8 Close Window Event

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)

myevent = sdl_new_sdl_event()
while true
        thevent = sdl_pollevent(myevent)
        switch sdl_get_sdl_event_type(myevent)
                on sdl_get_sdl_quit()
                        exit
                on sdl_get_sdl_keydown()
                        Key = SDL_GET_SDL_Event_key_keysym_sym(myevent)
                        if key = 27 exit ok

        off
end

SDL_DestroyWindow(win)
SDL_Quit()
```

84.9 Mouse Events

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)

win = SDL_CreateWindow("Mouse Events ", 100, 100, 640, 480, SDL_WINDOW_SHOWN)

TTF_Init()
font = TTF_OpenFont("pirulen.ttf", 16)
color = sdl_new_sdl_color()
sdl_set_sdl_color_r(color,0)
sdl_set_sdl_color_g(color,255)
sdl_set_sdl_color_b(color,0)

surface = SDL_GetWindowSurface(win)

myevent = sdl_new_sdl_event()
while true
        cMsg = ""
        sdl_pollevent(myevent)
        switch sdl_get_sdl_event_type(myevent)
                on SDL_QUIT
                        exit
                on SDL_KEYDOWN
                        Key = SDL_GET_SDL_Event_key_keysym_sym(myevent)
                        if key = 27 exit ok
                on SDL_MOUSEBUTTONDOWN
                        if sdl_get_Sdl_Event_button_button(myevent) = SDL_BUTTON_LEFT
                                SDL_SETWINDOWTITLE(win, " Button_Left_Down " )
                        but sdl_get_Sdl_Event_button_button(myevent) = SDL_BUTTON_MIDDLE
                                SDL_SETWINDOWTITLE(win,  " Button_Middle_Down " )
                        but sdl_get_Sdl_Event_button_button(myevent) = SDL_BUTTON_RIGHT
                                SDL_SETWINDOWTITLE(win,  " Button_Right_Down " )
                        ok
                on SDL_MOUSEMOTION
                        sdl_fillrect(surface,NULL,0)
                        if sdl_get_sdl_event_motion_xrel(myevent) < 0
                                cMsg += " Left "
                        else
                                cMsg += " Right "
                        ok
                        if sdl_get_sdl_event_motion_yrel(myevent) < 0
                                cMsg += " Up "
                        else
                                cMsg += " Down "
                        ok
                        cMsg += " x = " + sdl_get_sdl_event_motion_x(myevent)
                        cMsg += " y = " + sdl_get_sdl_event_motion_y(myevent)
                        showmsg(cMsg)
        off
```

(continues on next page)

(continued from previous page)

```
end

SDL_Destroy_SDL_Color(Color)
TTF_CloseFont(font)
SDL_DestroyWindow(win)
SDL_Quit()

func showmsg mymsg
        text = TTF_RenderText_Solid(font,mymsg,color)
        SDL_BlitSurface(text, NULL, surface, NULL)
        SDL_UpdateWindowSurface(win)
        SDL_FreeSurface(text)
```

84.10 Play Sound

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
Mix_OpenAudio( 44100, MIX_DEFAULT_FORMAT , 2, 10000)
Mix_AllocateChannels(4)
sound = Mix_LoadWav( "sound.wav" )
Mix_VolumeChunk(sound,1)
Mix_PlayChannel(1,sound,0)

myevent = sdl_new_sdl_event()
while true
        thevent = sdl_pollevent(myevent)
        switch sdl_get_sdl_event_type(myevent)
                on sdl_get_sdl_quit()
                        exit
                on sdl_get_sdl_keydown()
                        Key = SDL_GET_SDL_Event_key_keysym_sym(myevent)
                        if key = 27 exit ok
        off
end

Mix_FreeChunk( sound )
Mix_CloseAudio()
Mix_Quit()
SDL_DestroyWindow(win)
SDL_Quit()
```

CHAPTER

EIGHTYFIVE

RINGLIBSDL FUNCTIONS REFERENCE

85.1 Introduction

In this chapter we have a list of the supported functions by this extension

85.2 Reference

- MIX_DEFAULT_FORMAT
- SDL_QUIT
- SDL_BUTTON_LEFT
- SDL_BUTTON_MIDDLE
- SDL_BUTTON_RIGHT
- SDL_PRESSED
- SDL_RELEASED
- SDL_APP_TERMINATING
- SDL_APP_LOWMEMORY
- SDL_APP_WILLENTERBACKGROUND
- SDL_APP_DIDENTERBACKGROUND
- SDL_APP_WILLENTERFOREGROUND
- SDL_APP_DIDENTERFOREGROUND
- SDL_WINDOWEVENT
- SDL_SYSWMEVENT
- SDL_KEYDOWN
- SDL_KEYUP
- SDL_TEXTEDITING
- SDL_TEXTINPUT
- SDL_MOUSEMOTION
- SDL_MOUSEBUTTONDOWN
- SDL_MOUSEBUTTONUP

- SDL_MOUSEWHEEL
- SDL_JOYAXISMOTION
- SDL_JOYBALLMOTION
- SDL_JOYHATMOTION
- SDL_JOYBUTTONDOWN
- SDL_JOYBUTTONUP
- SDL_JOYDEVICEADDED
- SDL_JOYDEVICEREMOVED
- SDL_CONTROLLERAXISMOTION
- SDL_CONTROLLERBUTTONDOWN
- SDL_CONTROLLERBUTTONUP
- SDL_CONTROLLERDEVICEADDED
- SDL_CONTROLLERDEVICEREMOVED
- SDL_CONTROLLERDEVICEREMAPPED
- SDL_FINGERDOWN
- SDL_FINGERUP
- SDL_FINGERMOTION
- SDL_DOLLARGESTURE
- SDL_DOLLARRECORD
- SDL_MULTIGESTURE
- SDL_CLIPBOARDUPDATE
- SDL_DROPFILE
- SDL_RENDER_TARGETS_RESET
- SDL_USEREVENT
- SDL_LASTEVENT
- SDL_NET_MAJOR_VERSION
- SDL_NET_MINOR_VERSION
- SDL_NET_PATCHLEVEL
- INADDR_ANY
- INADDR_NONE
- INADDR_BROADCAST
- SDLNET_MAX_UDPCHANNELS
- SDLNET_MAX_UDPADDRESSES
- SDLK_0
- SDLK_1
- SDLK_2

- SDLK_3
- SDLK_4
- SDLK_5
- SDLK_6
- SDLK_7
- SDLK_8
- SDLK_9
- SDLK_a
- SDLK_AC_BACK
- SDLK_AC_BOOKMARKS
- SDLK_AC_FORWARD
- SDLK_AC_HOME
- SDLK_AC_REFRESH
- SDLK_AC_SEARCH
- SDLK_AC_STOP
- SDLK_AGAIN
- SDLK_ALTERASE
- SDLK_QUOTE
- SDLK_APPLICATION
- SDLK_AUDIOMUTE
- SDLK_AUDIONEXT
- SDLK_AUDIOPLAY
- SDLK_AUDIOPREV
- SDLK_BRIGHTNESSDOWN
- SDLK_BRIGHTNESSUP
- SDLK_c
- SDLK_CALCULATOR
- SDLK_CANCEL
- SDLK_CAPSLOCK
- SDLK_CLEAR
- SDLK_CLEARAGAIN
- SDLK_COMMA
- SDLK_COMPUTER
- SDLK_COPY
- SDLK_CRSEL
- SDLK_CURRENCYSUBUNIT

- SDLK_CURRENCYUNIT
- SDLK_CUT
- SDLK_d
- SDLK_DECIMALSEPARATOR
- SDLK_DELETE
- SDLK_DISPLAYSWITCH
- SDLK_DOWN
- SDLK_e
- SDLK_EJECT
- SDLK_END
- SDLK_EQUALS
- SDLK_ESCAPE
- SDLK_EXECUTE
- SDLK_EXSEL
- SDLK_f
- SDLK_F1
- SDLK_F10
- SDLK_F11
- SDLK_F12
- SDLK_F13
- SDLK_F14
- SDLK_F15
- SDLK_F16
- SDLK_F17
- SDLK_F18
- SDLK_F19
- SDLK_F2
- SDLK_F20
- SDLK_F21
- SDLK_F22
- SDLK_F23
- SDLK_F24
- SDLK_F3
- SDLK_F4
- SDLK_F5
- SDLK_F6

- SDLK_F7
- SDLK_F8
- SDLK_F9
- SDLK_FIND
- SDLK_g
- SDLK_BACKQUOTE
- SDLK_h
- SDLK_HELP
- SDLK_HOME
- SDLK_i
- SDLK_INSERT
- SDLK_j
- SDLK_k
- SDLK_KBDILLUMDOWN
- SDLK_KBDILLUMTOGGLE
- SDLK_KBDILLUMUP
- SDLK_KP_0
- SDLK_KP_00
- SDLK_KP_000
- SDLK_KP_1
- SDLK_KP_2
- SDLK_KP_3
- SDLK_KP_4
- SDLK_KP_5
- SDLK_KP_6
- SDLK_KP_7
- SDLK_KP_8
- SDLK_KP_9
- SDLK_KP_A
- SDLK_KP_AMPERSAND
- SDLK_KP_AT
- SDLK_KP_B
- SDLK_KP_BACKSPACE
- SDLK_KP_BINARY
- SDLK_KP_C
- SDLK_KP_CLEAR

- SDLK_KP_CLEARENTRY
- SDLK_KP_COLON
- SDLK_KP_COMMA
- SDLK_KP_D
- SDLK_KP_DBLAMPERSAND
- SDLK_KP_DBLVERTICALBAR
- SDLK_KP_DECIMAL
- SDLK_KP_DIVIDE
- SDLK_KP_E
- SDLK_KP_ENTER
- SDLK_KP_EQUALS
- SDLK_KP_EQUALSAS400
- SDLK_KP_EXCLAM
- SDLK_KP_F
- SDLK_KP_GREATER
- SDLK_KP_HASH
- SDLK_KP_HEXADECIMAL
- SDLK_KP_LEFTBRACE
- SDLK_KP_LEFTPAREN
- SDLK_KP_LESS
- SDLK_KP_MEMADD
- SDLK_KP_MEMCLEAR
- SDLK_KP_MEMDIVIDE
- SDLK_KP_MEMMULTIPLY
- SDLK_KP_MEMRECALL
- SDLK_KP_MEMSTORE
- SDLK_KP_MEMSUBTRACT
- SDLK_KP_MINUS
- SDLK_KP_MULTIPLY
- SDLK_KP_OCTAL
- SDLK_KP_PERCENT
- SDLK_KP_PERIOD
- SDLK_KP_PLUS
- SDLK_KP_PLUSMINUS
- SDLK_KP_POWER
- SDLK_KP_RIGHTBRACE

- SDLK_KP_RIGHTPAREN
- SDLK_KP_SPACE
- SDLK_KP_TAB
- SDLK_KP_VERTICALBAR
- SDLK_KP_XOR
- SDLK_l
- SDLK_LALT
- SDLK_LCTRL
- SDLK_LEFT
- SDLK_LEFTBRACKET
- SDLK_LGUI
- SDLK_LSHIFT
- SDLK_m
- SDLK_MAIL
- SDLK_MEDIASELECT
- SDLK_MENU
- SDLK_MINUS
- SDLK_MODE
- SDLK_MUTE
- SDLK_n
- SDLK_NUMLOCKCLEAR
- SDLK_o
- SDLK_OPER
- SDLK_OUT
- SDLK_p
- SDLK_PAGEDOWN
- SDLK_PAGEUP
- SDLK_PASTE
- SDLK_PAUSE
- SDLK_PERIOD
- SDLK_POWER
- SDLK_PRINTSCREEN
- SDLK_PRIOR
- SDLK_q
- SDLK_r
- SDLK_RALT

- SDLK_RCTRL
- SDLK_RETURN
- SDLK_RETURN2
- SDLK_RGUI
- SDLK_RIGHT
- SDLK_RIGHTBRACKET
- SDLK_RSHIFT
- SDLK_s
- SDLK_SCROLLLOCK
- SDLK_SELECT
- SDLK_SEMICOLON
- SDLK_SEPARATOR
- SDLK_SLASH
- SDLK_SLEEP
- SDLK_SPACE
- SDLK_STOP
- SDLK_SYSREQ
- SDLK_t
- SDLK_TAB
- SDLK_THOUSANDSSEPARATOR
- SDLK_u
- SDLK_UNDO
- SDLK_UNKNOWN
- SDLK_UP
- SDLK_v
- SDLK_VOLUMEDOWN
- SDLK_VOLUMEUP
- SDLK_w
- SDLK_WWW
- SDLK_x
- SDLK_y
- SDLK_z
- SDLK_AMPERSAND
- SDLK_ASTERISK
- SDLK_AT
- SDLK_CARET

- SDLK_COLON
- SDLK_DOLLAR
- SDLK_EXCLAIM
- SDLK_GREATER
- SDLK_HASH
- SDLK_LEFTPAREN
- SDLK_LESS
- SDLK_PERCENT
- SDLK_PLUS
- SDLK_QUESTION
- SDLK_QUOTEDBL
- SDLK_RIGHTPAREN
- SDLK_UNDERSCORE
- SDL_THREAD_PRIORITY_LOW
- SDL_THREAD_PRIORITY_NORMAL
- SDL_THREAD_PRIORITY_HIGH
- void SDL_RenderCopy2(SDL_Renderer *,SDL_Texture *)
- void SDL_Delay(int)
- void SDL_Init(int)
- int SDL_InitSubSystem(Uint32 flags)
- void SDL_Quit(void)
- void SDL_QuitSubSystem(Uint32 flags)
- void SDL_SetMainReady(void)
- Uint32 SDL_WasInit(Uint32 flags)
- SDL_bool SDL_SetHint(const char *name,const char *value)
- SDL_bool SDL_SetHintWithPriority(const char *name,const char *value,SDL_HintPriority priority)
- void SDL_ClearError(void)
- const char *SDL_GetError(void)
- SDL_LogPriority SDL_LogGetPriority(int category)
- void SDL_LogResetPriorities(void)
- void SDL_LogSetAllPriority(SDL_LogPriority priority)
- SDL_AssertionHandler SDL_GetDefaultAssertionHandler(void)
- void SDL_ResetAssertionReport(void)
- void SDL_SetAssertionHandler(SDL_AssertionHandler handler,void *userdata)
- void SDL_TriggerBreakpoint(void)
- void SDL_assert(int)

- void SDL_assert_paranoid(int)
- void SDL_assert_release(int)
- const char * SDL_GetRevision(void)
- int SDL_GetRevisionNumber(void)
- void SDL_GetVersion(SDL_version *ver)
- SDL_Window *SDL_CreateWindow(const char * title,int x, int y,int w,int h,Uint32 flags)
- void SDL_DestroyWindow(SDL_Window *window)
- void SDL_DisableScreenSaver(void)
- void SDL_EnableScreenSaver(void)
- SDL_GLContext SDL_GL_CreateContext(SDL_Window *window)
- void SDL_GL_DeleteContext(SDL_GLContext context)
- SDL_bool SDL_GL_ExtensionSupported(const char *extension)
- int SDL_GL_GetAttribute(SDL_GLattr attr,int *value)
- SDL_GLContext SDL_GL_GetCurrentContext(void)
- SDL_Window *SDL_GL_GetCurrentWindow(void)
- void SDL_GL_GetDrawableSize(SDL_Window *window,int *w,int *h)
- void *SDL_GL_GetProcAddress(const char *proc)
- int SDL_GL_GetSwapInterval(void)
- int SDL_GL_LoadLibrary(const char *path)
- int SDL_GL_MakeCurrent(SDL_Window *window,SDL_GLContext context)
- void SDL_GL_ResetAttributes(void)
- int SDL_GL_SetAttribute(SDL_GLattr attr,int value)
- int SDL_GL_SetSwapInterval(int interval)
- void SDL_GL_SwapWindow(SDL_Window *window)
- void SDL_GL_UnloadLibrary(void)
- SDL_DisplayMode *SDL_GetClosestDisplayMode(int displayIndex,SDL_DisplayMode *mode,SDL_DisplayMode *closest)
- int SDL_GetCurrentDisplayMode(int displayIndex,SDL_DisplayMode *mode)
- const char *SDL_GetCurrentVideoDriver(void)
- int SDL_GetDesktopDisplayMode(int displayIndex,SDL_DisplayMode *mode)
- int SDL_GetDisplayBounds(int displayIndex,SDL_Rect *rect)
- int SDL_GetNumVideoDisplays(void)
- int SDL_GetNumVideoDrivers(void)
- const char * SDL_GetVideoDriver(int index)
- void *SDL_GetWindowData(SDL_Window *window,const char *name)
- int SDL_GetWindowDisplayIndex(SDL_Window *window)

- int SDL_GetWindowDisplayMode(SDL_Window *window,SDL_DisplayMode *mode)
- Uint32 SDL_GetWindowFlags(SDL_Window *window)
- SDL_Window *SDL_GetWindowFromID(Uint32 id)
- int SDL_GetWindowGammaRamp(SDL_Window *window,Uint16 *red,Uint16 *green,Uint16 *blue)
- SDL_bool SDL_GetWindowGrab(SDL_Window *window)
- Uint32 SDL_GetWindowID(SDL_Window* window)
- void SDL_GetWindowMaximumSize(SDL_Window *window,int *w,int *h)
- void SDL_GetWindowMinimumSize(SDL_Window *window,int *w,int *h)
- void SDL_GetWindowPosition(SDL_Window *window,int *x,int *y)
- void SDL_GetWindowSize(SDL_Window *window,int *w,int *h)
- SDL_Surface *SDL_GetWindowSurface(SDL_Window *window)
- const char *SDL_GetWindowTitle(SDL_Window *window)
- SDL_bool SDL_IsScreenSaverEnabled(void)
- void SDL_MaximizeWindow(SDL_Window *window)
- void SDL_MinimizeWindow(SDL_Window *window)
- void SDL_RaiseWindow(SDL_Window *window)
- void SDL_RestoreWindow(SDL_Window *window)
- void SDL_SetWindowBordered(SDL_Window *window,SDL_bool bordered)
- int SDL_SetWindowBrightness(SDL_Window *window,float brightness)
- void *SDL_SetWindowData(SDL_Window *window,const char *name,void *userdata)
- int SDL_SetWindowDisplayMode(SDL_Window *window,const SDL_DisplayMode *mode)
- int SDL_SetWindowFullscreen(SDL_Window *window,Uint32 flags)
- int SDL_SetWindowGammaRamp(SDL_Window *window,const Uint16 *red,const Uint16 *green,const Uint16* blue)
- void SDL_SetWindowGrab(SDL_Window *window,SDL_bool grabbed)
- void SDL_SetWindowMinimumSize(SDL_Window* window,int min_w,int min_h)
- void SDL_SetWindowSize(SDL_Window *window,int w,int h)
- void SDL_SetWindowTitle(SDL_Window *window,const char *title)
- int SDL_ShowMessageBox(const SDL_MessageBoxData *messageboxdata,int *buttonid)
- int SDL_ShowSimpleMessageBox(Uint32 flags,const char *title,const char *message,SDL_Window *window)
- void SDL_ShowWindow(SDL_Window *window)
- int SDL_UpdateWindowSurface(SDL_Window *window)
- int SDL_UpdateWindowSurfaceRects(SDL_Window *window,const SDL_Rect *rects,int numrects)
- int SDL_VideoInit(const char *driver_name)
- void SDL_VideoQuit(void)
- SDL_Renderer *SDL_CreateRenderer(SDL_Window *window,int index,Uint32 flags)

- SDL_Renderer *SDL_CreateSoftwareRenderer(SDL_Surface *surface)
- SDL_Texture *SDL_CreateTexture(SDL_Renderer *renderer,Uint32 format,int access,int w,int h)
- SDL_Texture *SDL_CreateTextureFromSurface(SDL_Renderer *renderer,SDL_Surface *surface)
- void SDL_DestroyTexture(SDL_Texture *texture)
- int SDL_GL_BindTexture(SDL_Texture *texture,float *texw,float *texh)
- int SDL_GL_UnbindTexture(SDL_Texture *texture)
- int SDL_GetNumRenderDrivers(void)
- int SDL_GetRenderDrawBlendMode(SDL_Renderer *renderer,SDL_BlendMode *blendMode)
- int SDL_GetRenderDrawColor(SDL_Renderer *renderer,Uint8 *r,Uint8 *g,Uint8 *b,Uint8 *a)
- int SDL_GetRenderDriverInfo(int index,SDL_RendererInfo *info)
- SDL_Texture *SDL_GetRenderTarget(SDL_Renderer *renderer)
- SDL_Renderer *SDL_GetRenderer(SDL_Window *window)
- int SDL_GetRendererInfo(SDL_Renderer *renderer,SDL_RendererInfo *info)
- int SDL_GetRendererOutputSize(SDL_Renderer *renderer,int *w,int *h)
- int SDL_GetTextureAlphaMod(SDL_Texture *texture,Uint8 *alpha)
- int SDL_GetTextureBlendMode(SDL_Texture *texture,SDL_BlendMode *blendMode)
- int SDL_GetTextureColorMod(SDL_Texture *texture,Uint8 *r,Uint8 *g,Uint8 *b)
- int SDL_LockTexture(SDL_Texture *texture,const SDL_Rect *rect,void **pixels,int *pitch)
- int SDL_QueryTexture(SDL_Texture *texture,int *format,int *access,int *w,int *h)
- int SDL_RenderClear(SDL_Renderer *renderer)
- int SDL_RenderCopy(SDL_Renderer *renderer,SDL_Texture *texture,const SDL_Rect *srcrect,const SDL_Rect *dstrect)
- int SDL_RenderCopyEx(SDL_Renderer *renderer,SDL_Texture *texture,const SDL_Rect *srcrect,const SDL_Rect *dstrect,const double angle,const SDL_Point *center,const SDL_RendererFlip flip)
- int SDL_RenderDrawLine(SDL_Renderer *renderer,int x1,int y1,int x2,int y2)
- int SDL_RenderDrawLines(SDL_Renderer *renderer,const SDL_Point *points,int count)
- int SDL_RenderDrawPoint(SDL_Renderer *renderer,int x, int y)
- int SDL_RenderDrawPoints(SDL_Renderer *renderer,const SDL_Point *points,int count)
- int SDL_RenderDrawRect(SDL_Renderer *renderer,const SDL_Rect *rect)
- int SDL_RenderDrawRects(SDL_Renderer *renderer,const SDL_Rect *rects,int count)
- int SDL_RenderFillRect(SDL_Renderer *renderer,const SDL_Rect *rect)
- int SDL_RenderFillRects(SDL_Renderer *renderer,const SDL_Rect* rects,int count)
- void SDL_RenderGetClipRect(SDL_Renderer *renderer,SDL_Rect *rect)
- void SDL_RenderGetScale(SDL_Renderer *renderer,float *scaleX,float *scaleY)
- void SDL_RenderGetViewport(SDL_Renderer *renderer,SDL_Rect *rect)
- int SDL_RenderReadPixels(SDL_Renderer *renderer,const SDL_Rect *rect,Uint32 format,void *pixels,int pitch)

- int SDL_RenderSetClipRect(SDL_Renderer *renderer,const SDL_Rect *rect)
- int SDL_RenderSetScale(SDL_Renderer *renderer,float scaleX,float scaleY)
- int SDL_RenderSetViewport(SDL_Renderer *renderer,const SDL_Rect *rect)
- SDL_bool SDL_RenderTargetSupported(SDL_Renderer *renderer)
- int SDL_SetRenderDrawBlendMode(SDL_Renderer *renderer,SDL_BlendMode blendMode)
- int SDL_SetRenderDrawColor(SDL_Renderer *renderer,Uint8 r,Uint8 g,Uint8 b,Uint8 a)
- int SDL_SetRenderTarget(SDL_Renderer *renderer,SDL_Texture *texture)
- int SDL_SetTextureAlphaMod(SDL_Texture *texture,Uint8 alpha)
- int SDL_SetTextureBlendMode(SDL_Texture *texture,SDL_BlendMode blendMode)
- int SDL_SetTextureColorMod(SDL_Texture *texture,Uint8 r,Uint8 g,Uint8 b)
- void SDL_UnlockTexture(SDL_Texture *texture)
- int SDL_UpdateTexture(SDL_Texture *texture,const SDL_Rect *rect,const void*pixels,int pitch)
- int SDL_UpdateYUVTexture(SDL_Texture *texture,const SDL_Rect *rect,const Uint8 *Yplane,int Ypitch,const Uint8 *Uplane,int Upitch,const Uint8 *Vplane,int Vpitch)
- SDL_PixelFormat *SDL_AllocFormat(Uint32 pixel_format)
- SDL_Palette *SDL_AllocPalette(int ncolors)
- void SDL_CalculateGammaRamp(float gamma,Uint16 *ramp)
- void SDL_FreeFormat(SDL_PixelFormat *format)
- void SDL_FreePalette(SDL_Palette *palette)
- const char *SDL_GetPixelFormatName(Uint32 format)
- void SDL_GetRGB(Uint32 pixel,const SDL_PixelFormat* format,Uint8 *r,Uint8 *g,Uint8 *b)
- void SDL_GetRGBA(Uint32 pixel,const SDL_PixelFormat* format,Uint8 *r,Uint8 *g,Uint8 *b,Uint8 *a)
- Uint32 SDL_MapRGB(const SDL_PixelFormat* format,Uint8 r,Uint8 g, Uint8 b)
- Uint32 SDL_MapRGBA(const SDL_PixelFormat* format,Uint8 r,Uint8 g, Uint8 b, Uint8 a)
- Uint32 SDL_MasksToPixelFormatEnum(int bpp,Uint32 Rmask,Uint32 Gmask,Uint32 Bmask,Uint32 Amask)
- SDL_bool SDL_PixelFormatEnumToMasks(Uint32 format,int *bpp,Uint32 *Rmask,Uint32 *Gmask,Uint32 *Bmask,Uint32 *Amask)
- int SDL_SetPaletteColors(SDL_Palette *palette,const SDL_Color *colors,int firstcolor,int ncolors)
- int SDL_SetPixelFormatPalette(SDL_PixelFormat *format,SDL_Palette *palette)
- SDL_bool SDL_EnclosePoints(const SDL_Point* points,int count,const SDL_Rect *clip,SDL_Rect *result)
- SDL_bool SDL_HasIntersection(const SDL_Rect *A,const SDL_Rect *B)
- SDL_bool SDL_IntersectRect(const SDL_Rect *A,const SDL_Rect *B,SDL_Rect *result)
- SDL_bool SDL_IntersectRectAndLine(const SDL_Rect *rect,int *X1,int *Y1,int *X2,int *Y2)
- SDL_bool SDL_RectEquals(const SDL_Rect *a,const SDL_Rect *b)
- void SDL_UnionRect(const SDL_Rect *A,const SDL_Rect *B,SDL_Rect *result)
- int SDL_BlitScaled(SDL_Surface *src,const SDL_Rect *srcrect,SDL_Surface *dst,SDL_Rect *dstrect)
- int SDL_BlitSurface(SDL_Surface *src,const SDL_Rect* srcrect,SDL_Surface *dst,SDL_Rect *dstrect)

- int SDL_ConvertPixels(int width,int height,Uint32 src_format,const void *src,int src_pitch,Uint32 dst_format,void *dst,int dst_pitch)
- SDL_Surface *SDL_ConvertSurface(SDL_Surface *src,const SDL_PixelFormat *fmt,Uint32 flags)
- SDL_Surface *SDL_ConvertSurfaceFormat(SDL_Surface *src,Uint32 pixel_format,Uint32 flags)
- SDL_Surface *SDL_CreateRGBSurface(Uint32 flags,int width,int height,int depth,Uint32 Rmask,Uint32 Gmask,Uint32 Bmask,Uint32 Amask)
- SDL_Surface* SDL_CreateRGBSurfaceFrom(void *pixels,int width,int height,int depth,int pitch,Uint32 Rmask,Uint32 Gmask,Uint32 Bmask,Uint32 Amask)
- int SDL_FillRect(SDL_Surface *dst,const SDL_Rect *rect,Uint32 color)
- int SDL_FillRects(SDL_Surface *dst,const SDL_Rect *rects,int count,Uint32 color)
- void SDL_FreeSurface(SDL_Surface *surface)
- void SDL_GetClipRect(SDL_Surface *surface,SDL_Rect *rect)
- int SDL_GetColorKey(SDL_Surface *surface,Uint32 *key)
- int SDL_GetSurfaceAlphaMod(SDL_Surface *surface,Uint8 *alpha)
- int SDL_GetSurfaceBlendMode(SDL_Surface *surface,SDL_BlendMode *blendMode)
- int SDL_GetSurfaceColorMod(SDL_Surface *surface,Uint8 *r,Uint8 *g,Uint8 *b)
- SDL_Surface *SDL_LoadBMP(const char *file)
- SDL_Surface *SDL_LoadBMP_RW(SDL_RWops *src,int freesrc)
- int SDL_LockSurface(SDL_Surface *surface)
- int SDL_LowerBlit(SDL_Surface *src,SDL_Rect *srcrect,SDL_Surface *dst,SDL_Rect *dstrect)
- int SDL_LowerBlitScaled(SDL_Surface *src,SDL_Rect *srcrect,SDL_Surface *dst,SDL_Rect *dstrect)
- SDL_bool SDL_MUSTLOCK(SDL_Surface *surface)
- int SDL_SaveBMP(SDL_Surface *surface,const char *file)
- int SDL_SaveBMP_RW(SDL_Surface *surface,SDL_RWops *dst,int freedst)
- SDL_bool SDL_SetClipRect(SDL_Surface *surface,const SDL_Rect *rect)
- int SDL_SetColorKey(SDL_Surface *surface,int flag,Uint32 key)
- int SDL_SetSurfaceAlphaMod(SDL_Surface *surface,Uint8 alpha)
- int SDL_SetSurfaceBlendMode(SDL_Surface *surface,SDL_BlendMode blendMode)
- int SDL_SetSurfaceColorMod(SDL_Surface *surface,Uint8 r,Uint8 g,Uint8 b)
- int SDL_SetSurfacePalette(SDL_Surface *surface,SDL_Palette *palette)
- int SDL_SetSurfaceRLE(SDL_Surface *surface,int flag)
- void SDL_UnlockSurface(SDL_Surface* surface)
- SDL_bool SDL_GetWindowWMInfo(SDL_Window *window,SDL_SysWMinfo *info)
- char *SDL_GetClipboardText(void)
- SDL_bool SDL_HasClipboardText(void)
- int SDL_SetClipboardText(const char *text)
- void SDL_AddEventWatch(SDL_EventFilter filter,void *userdata)

- void SDL_DelEventWatch(SDL_EventFilter filter,void *userdata)
- Uint8 SDL_EventState(Uint32 type,int state)
- void SDL_FilterEvents(SDL_EventFilter filter,void *userdata)
- void SDL_FlushEvent(Uint32 type)
- void SDL_FlushEvents(Uint32 minType,Uint32 maxType)
- SDL_bool SDL_GetEventFilter(SDL_EventFilter *filter,void **userdata)
- Uint8 SDL_GetEventState(Uint32 type)
- int SDL_GetNumTouchDevices(void)
- int SDL_GetNumTouchFingers(SDL_TouchID touchID)
- SDL_TouchID SDL_GetTouchDevice(int index)
- SDL_Finger* SDL_GetTouchFinger(SDL_TouchID touchID,int index)
- SDL_bool SDL_HasEvent(Uint32 type)
- SDL_bool SDL_HasEvents(Uint32 minType,Uint32 maxType)
- int SDL_LoadDollarTemplates(SDL_TouchID touchId,SDL_RWops *src)
- int SDL_PeepEvents(SDL_Event *events,int numevents,SDL_eventaction action,Uint32 minType,Uint32 maxType)
- int SDL_PollEvent(SDL_Event *event)
- void SDL_PumpEvents(void)
- int SDL_PushEvent(SDL_Event *event)
- SDL_bool SDL_QuitRequested(void)
- int SDL_RecordGesture(SDL_TouchID touchId)
- Uint32 SDL_RegisterEvents(int numevents)
- int SDL_SaveAllDollarTemplates(SDL_RWops *dst)
- int SDL_SaveDollarTemplate(SDL_GestureID gestureId,SDL_RWops *dst)
- void SDL_SetEventFilter(SDL_EventFilter filter,void *userdata)
- int SDL_WaitEvent(SDL_Event *event)
- int SDL_WaitEventTimeout(SDL_Event *event,int timeout)
- SDL_Keycode SDL_GetKeyFromName(const char * name)
- SDL_Keycode SDL_GetKeyFromScancode(SDL_Scancode scancode)
- const char * SDL_GetKeyName(SDL_Keycode key)
- SDL_Window* SDL_GetKeyboardFocus(void)
- const Uint8* SDL_GetKeyboardState(int* numkeys)
- SDL_Keymod SDL_GetModState(void)
- SDL_Scancode SDL_GetScancodeFromKey(SDL_Keycode key)
- SDL_Scancode SDL_GetScancodeFromName(const char * name)
- const char * SDL_GetScancodeName(SDL_Scancode scancode)

- SDL_bool SDL_HasScreenKeyboardSupport(void)
- SDL_bool SDL_IsScreenKeyboardShown(SDL_Window* window)
- SDL_bool SDL_IsTextInputActive(void)
- void SDL_SetModState(SDL_Keymod modstate)
- void SDL_SetTextInputRect(SDL_Rect* rect)
- void SDL_StartTextInput(void)
- void SDL_StopTextInput(void)
- SDL_Cursor *SDL_CreateCursor(const Uint8 *data,const Uint8 *mask,int w,int h,int hot_x,int hot_y)
- void SDL_FreeCursor(SDL_Cursor *cursor)
- SDL_Cursor *SDL_GetCursor(void)
- SDL_Cursor *SDL_GetDefaultCursor(void)
- Uint32 SDL_GetMouseState(int *x,int * y)
- SDL_bool SDL_GetRelativeMouseMode(void)
- Uint32 SDL_GetRelativeMouseState(int *x,int *y)
- void SDL_SetCursor(SDL_Cursor *cursor)
- int SDL_SetRelativeMouseMode(SDL_bool enabled)
- int SDL_ShowCursor(int toggle)
- void SDL_JoystickClose(SDL_Joystick *joystick)
- SDL_bool SDL_JoystickGetAttached(SDL_Joystick *joystick)
- Sint16 SDL_JoystickGetAxis(SDL_Joystick *joystick,int axis)
- int SDL_JoystickGetBall(SDL_Joystick *joystick,int ball,int *dx,int *dy)
- Uint8 SDL_JoystickGetButton(SDL_Joystick *joystick,int button)
- SDL_JoystickGUID SDL_JoystickGetDeviceGUID(int device_index)
- SDL_JoystickGUID SDL_JoystickGetGUID(SDL_Joystick *joystick)
- SDL_JoystickGUID SDL_JoystickGetGUIDFromString(const char *pchGUID)
- void SDL_JoystickGetGUIDString(SDL_JoystickGUID guid,char *pszGUID,int cbGUID)
- Uint8 SDL_JoystickGetHat(SDL_Joystick *joystick,int hat)
- SDL_JoystickID SDL_JoystickInstanceID(SDL_Joystick *joystick)
- const char *SDL_JoystickName(SDL_Joystick *joystick)
- const char *SDL_JoystickNameForIndex(int device_index)
- int SDL_JoystickNumAxes(SDL_Joystick *joystick)
- int SDL_JoystickNumBalls(SDL_Joystick *joystick)
- int SDL_JoystickNumButtons(SDL_Joystick *joystick)
- int SDL_JoystickNumHats(SDL_Joystick *joystick)
- SDL_Joystick *SDL_JoystickOpen(int device_index)
- void SDL_JoystickUpdate(void)

- int SDL_NumJoysticks(void)
- int SDL_GameControllerAddMapping(const char *mappingString)
- int SDL_GameControllerAddMappingsFromFile(const char *filename)
- int SDL_GameControllerAddMappingsFromRW(SDL_RWops *rw,int freerw)
- void SDL_GameControllerClose(SDL_GameController *gamecontroller)
- int SDL_GameControllerEventState(int state)
- Sint16 SDL_GameControllerGetAxis(SDL_GameController *gamecontroller,SDL_GameControllerAxis axis)
- SDL_GameControllerAxis SDL_GameControllerGetAxisFromString(const char *pchString)
- SDL_GameControllerButtonBind SDL_GameControllerGetBindForAxis(SDL_GameController *gamecontroller,SDL_GameControllerAxis axis)
- SDL_GameControllerButtonBind SDL_GameControllerGetBindForButton(SDL_GameController *gamecontroller,SDL_GameControllerButton button)
- Uint8 SDL_GameControllerGetButton(SDL_GameController *gamecontroller,SDL_GameControllerButton button)
- SDL_GameControllerButton SDL_GameControllerGetButtonFromString(const char *pchString)
- SDL_Joystick *SDL_GameControllerGetJoystick(SDL_GameController *gamecontroller)
- const char *SDL_GameControllerGetStringForAxis(SDL_GameControllerAxis axis)
- const char *SDL_GameControllerGetStringForButton(SDL_GameControllerButton button)
- char *SDL_GameControllerMapping(SDL_GameController *gamecontroller)
- char *SDL_GameControllerMappingForGUID(SDL_JoystickGUID guid)
- const char *SDL_GameControllerName(SDL_GameController *gamecontroller)
- const char *SDL_GameControllerNameForIndex(int joystick_index)
- SDL_GameController* SDL_GameControllerOpen(int joystick_index)
- void SDL_GameControllerUpdate(void)
- SDL_bool SDL_IsGameController(int joystick_index)
- void SDL_HapticClose(SDL_Haptic* haptic)
- void SDL_HapticDestroyEffect(SDL_Haptic *haptic,int effect)
- int SDL_HapticEffectSupported(SDL_Haptic *haptic,SDL_HapticEffect *effect)
- int SDL_HapticGetEffectStatus(SDL_Haptic *haptic,int effect)
- int SDL_HapticIndex(SDL_Haptic *haptic)
- const char *SDL_HapticName(int device_index)
- int SDL_HapticNewEffect(SDL_Haptic *haptic,SDL_HapticEffect *effect)
- int SDL_HapticNumAxes(SDL_Haptic *haptic)
- int SDL_HapticNumEffects(SDL_Haptic *haptic)
- int SDL_HapticNumEffectsPlaying(SDL_Haptic *haptic)
- SDL_Haptic *SDL_HapticOpen(int device_index)
- SDL_Haptic *SDL_HapticOpenFromJoystick(SDL_Joystick *joystick)

- SDL_Haptic *SDL_HapticOpenFromMouse(void)
- int SDL_HapticOpened(int device_index)
- int SDL_HapticPause(SDL_Haptic *haptic)
- unsigned int SDL_HapticQuery(SDL_Haptic *haptic)
- int SDL_HapticRumbleInit(SDL_Haptic *haptic)
- int SDL_HapticRumblePlay(SDL_Haptic *haptic,float strength,Uint32 length)
- int SDL_HapticRumbleStop(SDL_Haptic *haptic)
- int SDL_HapticRumbleSupported(SDL_Haptic *haptic)
- int SDL_HapticRunEffect(SDL_Haptic *haptic,int effect,Uint32 iterations)
- int SDL_HapticSetAutocenter(SDL_Haptic *haptic,int autocenter)
- int SDL_HapticSetGain(SDL_Haptic *haptic,int gain)
- int SDL_HapticStopAll(SDL_Haptic *haptic)
- int SDL_HapticStopEffect(SDL_Haptic *haptic,int effect)
- int SDL_HapticUnpause(SDL_Haptic *haptic)
- int SDL_HapticUpdateEffect(SDL_Haptic *haptic,int effect,SDL_HapticEffect *data)
- int SDL_JoystickIsHaptic(SDL_Joystick *joystick)
- int SDL_MouseIsHaptic(void)
- int SDL_NumHaptics(void)
- int SDL_AudioInit(const char * driver_name)
- void SDL_AudioQuit(void)
- int SDL_BuildAudioCVT(SDL_AudioCVT *cvt,SDL_AudioFormat src_format,Uint8 src_channels,int src_rate,SDL_AudioFormat dst_format,Uint8 dst_channels,int dst_rate)
- void SDL_CloseAudioDevice(SDL_AudioDeviceID dev)
- int SDL_ConvertAudio(SDL_AudioCVT *cvt)
- void SDL_FreeWAV(Uint8 *audio_buf)
- const char * SDL_GetAudioDeviceName(int index,int iscapture)
- SDL_AudioStatus SDL_GetAudioDeviceStatus(SDL_AudioDeviceID dev)
- const char * SDL_GetAudioDriver(int index)
- SDL_AudioStatus SDL_GetAudioStatus(void)
- const char * SDL_GetCurrentAudioDriver(void)
- int SDL_GetNumAudioDevices(int iscapture)
- int SDL_GetNumAudioDrivers(void)
- SDL_AudioSpec *SDL_LoadWAV_RW(SDL_RWops *src,int freesrc,SDL_AudioSpec *spec,Uint8 **audio_buf,Uint32 *audio_len)
- void SDL_LockAudio(void)
- void SDL_LockAudioDevice(SDL_AudioDeviceID dev)
- void SDL_MixAudio(Uint8 *dst,const Uint8* src,Uint32 len,int volume)

- void SDL_MixAudioFormat(Uint8 *dst,const Uint8 *src,SDL_AudioFormat format,Uint32 len,int volume)
- int SDL_OpenAudio(SDL_AudioSpec *desired,SDL_AudioSpec *obtained)
- SDL_AudioDeviceID SDL_OpenAudioDevice(const char *device,int iscapture,const SDL_AudioSpec *desired,SDL_AudioSpec *obtained,int allowed_changes)
- void SDL_PauseAudio(int pause_on)
- void SDL_PauseAudioDevice(SDL_AudioDeviceID dev,int pause_on)
- void SDL_UnlockAudio(void)
- void SDL_UnlockAudioDevice(SDL_AudioDeviceID dev)
- char *SDL_GetBasePath(void)
- char *SDL_GetPrefPath(const char *org,const char *app)
- SDL_RWops *SDL_AllocRW(void)
- void SDL_FreeRW(SDL_RWops *area)
- SDL_RWops *SDL_RWFromConstMem(const void* mem,int size)
- SDL_RWops *SDL_RWFromFP(void *fp,SDL_bool autoclose)
- SDL_RWops *SDL_RWFromFile(const char *file,const char *mode)
- SDL_RWops *SDL_RWFromMem(void *mem,int size)
- int SDL_RWclose(struct SDL_RWops *context)
- size_t SDL_RWread(struct SDL_RWops *context,void *ptr,size_t size,size_t maxnum)
- Sint64 SDL_RWseek(SDL_RWops *context,Sint64 offset,int whence)
- Sint64 SDL_RWsize(SDL_RWops *context)
- Sint64 SDL_RWtell(struct SDL_RWops *context)
- size_t SDL_RWwrite(struct SDL_RWops *context,const void *ptr,size_t size,size_t num)
- Uint16 SDL_ReadBE16(SDL_RWops *src)
- Uint32 SDL_ReadBE32(SDL_RWops *src)
- Uint64 SDL_ReadBE64(SDL_RWops *src)
- Uint16 SDL_ReadLE16(SDL_RWops *src)
- Uint32 SDL_ReadLE32(SDL_RWops *src)
- Uint64 SDL_ReadLE64(SDL_RWops *src)
- Uint8 SDL_ReadU8(SDL_RWops *src)
- size_t SDL_WriteBE16(SDL_RWops *dst,Uint16 value)
- size_t SDL_WriteBE32(SDL_RWops *dst,Uint32 value)
- size_t SDL_WriteBE64(SDL_RWops *dst,Uint64 value)
- size_t SDL_WriteLE16(SDL_RWops *dst,Uint16 value)
- size_t SDL_WriteLE32(SDL_RWops *dst,Uint32 value)
- size_t SDL_WriteLE64(SDL_RWops *dst,Uint64 value)
- size_t SDL_WriteU8(SDL_RWops *dst,Uint8 value)

- void *SDL_LoadFunction(void *handle,const char *name)
- void *SDL_LoadObject(const char *sofile)
- void SDL_UnloadObject(void *handle)
- const char *SDL_GetPlatform(void)
- int SDL_GetCPUCacheLineSize(void)
- int SDL_GetCPUCount(void)
- int SDL_GetSystemRAM(void)
- SDL_bool SDL_Has3DNow(void)
- SDL_bool SDL_HasAVX(void)
- SDL_bool SDL_HasMMX(void)
- SDL_bool SDL_HasRDTSC(void)
- SDL_bool SDL_HasSSE(void)
- SDL_bool SDL_HasSSE2(void)
- SDL_bool SDL_HasSSE3(void)
- SDL_bool SDL_HasSSE41(void)
- SDL_bool SDL_HasSSE42(void)
- SDL_PowerState SDL_GetPowerInfo(int *secs,int *pct)
- double SDL_acos(double x)
- int IMG_Init(int flags)
- void IMG_Quit(void)
- SDL_Surface *IMG_Load(const char *file)
- SDL_Surface *IMG_Load_RW(SDL_RWops *src, int freesrc)
- SDL_Surface *IMG_LoadTyped_RW(SDL_RWops *src, int freesrc, char *type)
- SDL_Surface *IMG_LoadCUR_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadBMP_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadPNM_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadXPM_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadXCF_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadPCX_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadGIF_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadJPG_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadTIF_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadPNG_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadTGA_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadLBM_RW(SDL_RWops *src)
- SDL_Surface *IMG_LoadXV_RW(SDL_RWops *src)

- SDL_Surface *IMG_ReadXPMFromArray(char **xpm)
- int IMG_isCUR(SDL_RWops *src)
- int IMG_isICO(SDL_RWops *src)
- int IMG_isBMP(SDL_RWops *src)
- int IMG_isPNM(SDL_RWops *src)
- int IMG_isXPM(SDL_RWops *src)
- int IMG_isXCF(SDL_RWops *src)
- int IMG_isPCX(SDL_RWops *src)
- int IMG_isGIF(SDL_RWops *src)
- int IMG_isJPG(SDL_RWops *src)
- int IMG_isTIF(SDL_RWops *src)
- int IMG_isPNG(SDL_RWops *src)
- int IMG_isLBM(SDL_RWops *src)
- int IMG_isXV(SDL_RWops *src)
- int TTF_Init(void)
- int TTF_WasInit(void)
- void TTF_Quit(void)
- TTF_Font *TTF_OpenFont(const char *file, int ptsize)
- TTF_Font *TTF_OpenFontRW(SDL_RWops *src, int freesrc, int ptsize)
- TTF_Font *TTF_OpenFontIndex(const char *file, int ptsize, long index)
- TTF_Font *TTF_OpenFontIndexRW(SDL_RWops *src, int freesrc, int ptsize, long index)
- void TTF_CloseFont(TTF_Font *font)
- void TTF_ByteSwappedUNICODE(int swapped)
- int TTF_GetFontStyle(TTF_Font *font)
- void TTF_SetFontStyle(TTF_Font *font, int style)
- int TTF_GetFontOutline(TTF_Font *font)
- void TTF_SetFontOutline(TTF_Font *font, int outline)
- int TTF_GetFontHinting(TTF_Font *font)
- void TTF_SetFontHinting(TTF_Font *font, int hinting)
- int TTF_GetFontKerning(TTF_Font *font)
- void TTF_SetFontKerning(TTF_Font *font, int allowed)
- int TTF_FontHeight(const TTF_Font *font)
- int TTF_FontAscent(const TTF_Font *font)
- int TTF_FontDescent(const TTF_Font *font)
- int TTF_FontLineSkip(const TTF_Font *font)
- long TTF_FontFaces(const TTF_Font *font)

- int TTF_FontFaceIsFixedWidth(const TTF_Font *font)
- char *TTF_FontFaceFamilyName(const TTF_Font *font)
- char *TTF_FontFaceStyleName(const TTF_Font *font)
- int TTF_GlyphIsProvided(const TTF_Font *font, Uint16 ch)
- int TTF_GlyphMetrics(TTF_Font *font, Uint16 ch, int *minx, int *maxx, int *miny, int *maxy, int *advance)
- int TTF_SizeText(TTF_Font *font, const char *text, int *w, int *h)
- int TTF_SizeUTF8(TTF_Font *font, const char *text, int *w, int *h)
- int TTF_SizeUNICODE(TTF_Font *font, const Uint16 *text, int *w, int *h)
- SDL_Surface *TTF_RenderText_Solid(TTF_Font *font, const char *text, SDL_Color fg)
- SDL_Surface *TTF_RenderUTF8_Solid(TTF_Font *font, const char *text,SDL_Color fg)
- SDL_Surface *TTF_RenderUNICODE_Solid(TTF_Font *font, const Uint16 *text,SDL_Color fg)
- SDL_Surface *TTF_RenderGlyph_Solid(TTF_Font *font, Uint16 ch, SDL_Color fg)
- SDL_Surface *TTF_RenderText_Shaded(TTF_Font *font, const char *text,SDL_Color fg, SDL_Color bg)
- SDL_Surface *TTF_RenderUTF8_Shaded(TTF_Font *font, const char *text,SDL_Color fg, SDL_Color bg)
- SDL_Surface *TTF_RenderUNICODE_Shaded(TTF_Font *font, const Uint16 *text,SDL_Color fg, SDL_Color bg)
- SDL_Surface *TTF_RenderGlyph_Shaded(TTF_Font *font, Uint16 ch, SDL_Color fg,SDL_Color bg)
- SDL_Surface *TTF_RenderText_Blended(TTF_Font *font, const char *text,SDL_Color fg)
- SDL_Surface *TTF_RenderUTF8_Blended(TTF_Font *font, const char *text,SDL_Color fg)
- SDL_Surface *TTF_RenderUNICODE_Blended(TTF_Font *font, const Uint16 *text,SDL_Color fg)
- SDL_Surface *TTF_RenderGlyph_Blended(TTF_Font *font, Uint16 ch, SDL_Color fg)
- int Mix_Init(int flags)
- void Mix_Quit(void)
- int Mix_OpenAudio(int frequency, Uint16 format, int channels, int chunksize)
- void Mix_CloseAudio(void)
- int Mix_QuerySpec(int *frequency, Uint16 *format, int *channels)
- int Mix_GetNumChunkDecoders(void)
- const char *Mix_GetChunkDecoder(int index)
- Mix_Chunk *Mix_LoadWAV(char *file)
- Mix_Chunk *Mix_LoadWAV_RW(SDL_RWops *src, int freesrc)
- Mix_Chunk *Mix_QuickLoad_WAV(Uint8 *mem)
- void Mix_FreeChunk(Mix_Chunk *chunk)
- int Mix_AllocateChannels(int numchans)
- int Mix_Volume(int channel, int volume)
- int Mix_PlayChannel(int channel, Mix_Chunk *chunk, int loops)
- int Mix_PlayChannelTimed(int channel, Mix_Chunk *chunk, int loops, int ticks)

- int Mix_FadeInChannel(int channel, Mix_Chunk *chunk, int loops, int ms)
- int Mix_FadeInChannelTimed(int channel, Mix_Chunk *chunk,int loops, int ms, int ticks)
- void Mix_Pause(int channel)
- void Mix_Resume(int channel)
- int Mix_HaltChannel(int channel)
- int Mix_ExpireChannel(int channel, int ticks)
- int Mix_FadeOutChannel(int channel, int ms)
- int Mix_Paused(int channel)
- Mix_Fading Mix_FadingChannel(int which)
- Mix_Chunk *Mix_GetChunk(int channel)
- int Mix_ReserveChannels(int num)
- int Mix_GroupChannel(int which, int tag)
- int Mix_GroupChannels(int from, int to, int tag)
- int Mix_GroupCount(int tag)
- int Mix_GroupAvailable(int tag)
- int Mix_GroupOldest(int tag)
- int Mix_GroupNewer(int tag)
- int Mix_FadeOutGroup(int tag, int ms)
- int Mix_HaltGroup(int tag)
- int Mix_GetNumMusicDecoders(void)
- const char *Mix_GetMusicDecoder(int index)
- Mix_Music *Mix_LoadMUS(const char *file)
- void Mix_FreeMusic(Mix_Music *music)
- int Mix_PlayMusic(Mix_Music *music, int loops)
- int Mix_FadeInMusic(Mix_Music *music, int loops, int ms)
- int Mix_FadeInMusicPos(Mix_Music *music, int loops, int ms, double position)
- int Mix_PlayingMusic(void)
- int Mix_PausedMusic(void)
- Mix_Fading Mix_FadingMusic(void)
- void *Mix_GetMusicHookData(void)
- int Mix_RegisterEffect(int chan, Mix_EffectFunc_t f, Mix_EffectDone_t d,void *arg)
- int Mix_UnregisterEffect(int channel, Mix_EffectFunc_t f)
- int Mix_UnregisterAllEffects(int channel)
- int Mix_SetDistance(int channel, Uint8 distance)
- int Mix_SetPosition(int channel, Sint16 angle, Uint8 distance)
- int Mix_SetReverseStereo(int channel, int flip)

- int SDLNet_Init(void)
- void SDLNet_Quit(void)
- char *SDLNet_GetError(void)
- void SDLNet_Write16(Uint16 value, void *area)
- void SDLNet_Write32(Uint32 value, void *area)
- Uint16 SDLNet_Read16(void *area)
- Uint32 SDLNet_Read32(void *area)
- int SDLNet_ResolveHost(IPaddress *address, const char *host, Uint16 port)
- const char *SDLNet_ResolveIP(IPaddress *address)
- TCPsocket SDLNet_TCP_Open(IPaddress *ip)
- void SDLNet_TCP_Close(TCPsocket sock)
- TCPsocket SDLNet_TCP_Accept(TCPsocket server)
- int SDLNet_TCP_Send(TCPsocket sock, const void *data, int len)
- int SDLNet_TCP_Recv(TCPsocket sock, void *data, int maxlen)
- UDPsocket SDLNet_UDP_Open(Uint16 port)
- void SDLNet_UDP_Close(UDPsocket sock)
- int SDLNet_UDP_Bind(UDPsocket sock, int channel, IPaddress *address)
- void SDLNet_UDP_Unbind(UDPsocket sock, int channel)
- IPaddress *SDLNet_UDP_GetPeerAddress(UDPsocket sock, int channel)
- int SDLNet_UDP_Send(UDPsocket sock, int channel, UDPpacket *packet)
- int SDLNet_UDP_Recv(UDPsocket sock, UDPpacket *packet)
- int SDLNet_UDP_SendV(UDPsocket sock, UDPpacket **packetV, int npackets)
- int SDLNet_UDP_RecvV(UDPsocket sock, UDPpacket **packetV)
- UDPpacket *SDLNet_AllocPacket(int size)
- int SDLNet_ResizePacket(UDPpacket *packet, int size)
- void SDLNet_FreePacket(UDPpacket *packet)
- UDPpacket **SDLNet_AllocPacketV(int howmany, int size)
- void SDLNet_FreePacketV(UDPpacket **packetV)
- SDLNet_SocketSet SDLNet_AllocSocketSet(int maxsockets)
- void SDLNet_FreeSocketSet(SDLNet_SocketSet set)
- int SDLNet_AddSocket(SDLNet_SocketSet set, SDLNet_GenericSocket sock)
- int SDLNet_TCP_AddSocket(SDLNet_SocketSet set, TCPsocket sock)
- int SDLNet_UDP_AddSocket(SDLNet_SocketSet set, UDPsocket sock)
- int SDLNet_DelSocket(SDLNet_SocketSet set, SDLNet_GenericSocket sock)
- int SDLNet_TCP_DelSocket(SDLNet_SocketSet set, TCPsocket sock)
- int SDLNet_UDP_DelSocket(SDLNet_SocketSet set, UDPsocket sock)

- int SDLNet_CheckSockets(SDLNet_SocketSet set, Uint32 timeout)
- int SDLNet_SocketReady(TCPsocket sock)
- int circleRGBA(SDL_Renderer * renderer, Sint16 x, Sint16 y, Sint16 rad, Uint8 r, Uint8 g, Uint8 b, Uint8 a)
- SDL_Thread *SDL_CreateThread(SDL_ThreadFunction fn,const char *name,void *data)
- void SDL_DetachThread(SDL_Thread *thread)
- SDL_threadID SDL_GetThreadID(SDL_Thread *thread)
- const char *SDL_GetThreadName(SDL_Thread* thread)
- int SDL_SetThreadPriority(SDL_ThreadPriority priority)
- SDL_TLSID SDL_TLSCreate(void)
- void *SDL_TLSGet(SDL_TLSID id)
- int SDL_TLSSet(SDL_TLSID id,const void *value,void *)
- SDL_threadID SDL_ThreadID(void)
- void SDL_WaitThread(SDL_Thread *thread,int *status)
- int SDL_CondBroadcast(SDL_cond *cond)
- int SDL_CondSignal(SDL_cond *cond)
- int SDL_CondWait(SDL_cond *cond,SDL_mutex *mutex)
- int SDL_CondWaitTimeout(SDL_cond *cond,SDL_mutex *mutex,Uint32 ms)
- SDL_cond *SDL_CreateCond(void)
- SDL_mutex *SDL_CreateMutex(void)
- SDL_sem *SDL_CreateSemaphore(Uint32 initial_value)
- void SDL_DestroyCond(SDL_cond *cond)
- void SDL_DestroyMutex(SDL_mutex *mutex)
- void SDL_DestroySemaphore(SDL_sem *sem)
- int SDL_LockMutex(SDL_mutex *mutex)
- int SDL_SemPost(SDL_sem *sem)
- int SDL_SemTryWait(SDL_sem *sem)
- Uint32 SDL_SemValue(SDL_sem *sem)
- int SDL_SemWait(SDL_sem *sem)
- int SDL_SemWaitTimeout(SDL_sem *sem,Uint32 ms)
- int SDL_TryLockMutex(SDL_mutex *mutex)
- int SDL_UnlockMutex(SDL_mutex *mutex)

CHAPTER

EIGHTYSIX

USING RINGLIBUV

In this chapter we will learn about using RingLibuv

Note: To use RingLibuv, Check ring/extensions/ringlibuv folder.

Information from the library website: http://libuv.org/

Libuv is a multi-platform support library with a focus on asynchronous I/O.

Feature highlights

- Full-featured event loop backed by epoll, kqueue, IOCP, event ports.
- Asynchronous TCP and UDP sockets
- Asynchronous DNS resolution
- Asynchronous file and file system operations
- File system events
- ANSI escape code controlled TTY
- IPC with socket sharing, using Unix domain sockets or named pipes (Windows)
- Child processes
- Thread pool
- Signal handling
- High resolution clock
- Threading and synchronization primitives

86.1 First Application using RingLibuv

Example:

```
load "libuv.ring"

func main

        myloop = new_uv_loop_t()
        uv_loop_init(myloop)
        ? "Now quitting"
```

(continues on next page)

(continued from previous page)

```
        uv_run(myloop, UV_RUN_DEFAULT)
        uv_loop_close(myloop)
        destroy_uv_loop_t(myloop)
```

Output:

```
Now quitting
```

86.2 The Events Loop

Example:

```
load "libuv.ring"

counter = 0
idler = NULL

func main
        idler = new_uv_idle_t()
        uv_idle_init(uv_default_loop(), idler)
        uv_idle_start(idler, "wait()")
        ? "Idling..."
        uv_run(uv_default_loop(), UV_RUN_DEFAULT);
        uv_loop_close(uv_default_loop());
        destroy_uv_idle_t(idler)

func wait
        counter++
        if counter >= 100000
                uv_idle_stop(idler)
        ok
```

Output:

```
Idling...
```

86.3 Server Example

Example:

```
load "libuv.ring"

? "Testing RingLibuv - Server Side"

DEFAULT_PORT    = 13370
DEFAULT_BACKLOG = 1024

addr    = new_sockaddr_in()
```

(continues on next page)

(continued from previous page)

```
server  = NULL
client  = NULL
myloop  = NULL

func main
        myloop = uv_default_loop()
        server = new_uv_tcp_t()
        uv_tcp_init(myloop, server)
        uv_ip4_addr("127.0.0.1", DEFAULT_PORT, addr)
        uv_tcp_bind(server, addr, 0)
        r = uv_listen(server, DEFAULT_BACKLOG, "newconnection()")
        if r
                ? "Listen error " + uv_strerror(r)
                return 1
        ok
        uv_run(myloop, UV_RUN_DEFAULT)
        destroy_uv_tcp_t(server)
        destroy_uv_sockaddr_in(addr)

func newconnection
        ? "New Connection"
        aPara   = uv_Eventpara(server,:connect)
        nStatus = aPara[2]
        if nStatus < 0
                ? "New connection error : " + nStatus
                return
        ok
        client = new_uv_tcp_t()
        uv_tcp_init(myloop, client)
        if uv_accept(server, client) = 0
                        uv_read_start(client, uv_myalloccallback(), "echo_read()")
        ok

func echo_read
        aPara = uv_Eventpara(client,:read)
        nRead = aPara[2]
        buf   = aPara[3]
        if nRead > 0
                req = new_uv_write_t()
                        wrbuf = uv_buf_init(get_uv_buf_t_base(buf), nread)
                uv_write(req, client, wrbuf, 1, "echo_write()")
                ? uv_buf2str(wrbuf)
                message = "message from the server to the client"
                buf = new_uv_buf_t()
                set_uv_buf_t_len(buf,len(message))
                set_uv_buf_t_base(buf,varptr("message",:char))
                uv_write(req, client, buf, 1, "echo_write()")
        ok

func echo_write
        aPara = uv_Eventpara(client,:read)
        req   = aPara[1]
```

Output:

When we run the client, We will see the message "New Connection"

Then the message "hello from the client"

```
Testing RingLibuv - Server Side
New Connection
hello from the client
```

86.4 Client Example

Example:

```
load "libuv.ring"

? "Testing RingLibuv - Client Side"

DEFAULT_PORT    = 13370
DEFAULT_BACKLOG = 1024

addr    = new_sockaddr_in()
connect = NULL
buffer  = null
socket  = null

func main
        myloop  = uv_default_loop()
        Socket  = new_uv_tcp_t()
        connect = new_uv_connect_t()
        uv_tcp_init(myloop, Socket)
        uv_ip4_addr("127.0.0.1", DEFAULT_PORT, addr)
        uv_tcp_connect(connect,Socket, addr, "connect()")
        uv_run(myloop, UV_RUN_DEFAULT)
        destroy_uv_tcp_t(socket)
        destroy_uv_connect_t(connect)

func connect
        ? "Client: Start Connection"
        aPara   = uv_Eventpara(connect,:connect)
        req     = aPara[1]
        nStatus = aPara[2]
        if nStatus = -1
                ? "Error : on_write_end "
                return
        ok
        buf = new_uv_buf_t()
        message = "hello from the client"
        set_uv_buf_t_len(buf,len(message))
        set_uv_buf_t_base(buf,varptr("message",:char))
        tcp       = get_uv_connect_t_handle(req)
        write_req = new_uv_write_t()
```

(continues on next page)

(continued from previous page)

```
        buf_count = 1
        uv_write(write_req, tcp, buf, buf_count, "on_write_end()")

func on_write_end
                uv_read_start(socket, uv_myalloccallback(), "echo_read()")

func echo_read
        aPara = uv_Eventpara(socket,:read)
        nRead = aPara[2]
        buf   = aPara[3]
        if nRead > 0
                        wrbuf = uv_buf_init(get_uv_buf_t_base(buf), nread);
                ? uv_buf2str(wrbuf)
        ok
```

Output:

We will run the client after the server

```
Testing RingLibuv - Client Side
Client: Start Connection
hello from the client
message from the server to the client
```

86.5 Server Example Using Classes

Example:

```
load "libuv.ring"
load "objectslib.ring"

? "Testing RingLibuv - Server Side - Using Classes"

open_object(:MyServer)

class MyServer from ObjectControllerParent

        DEFAULT_PORT    = 13370
        DEFAULT_BACKLOG = 1024

        addr    = new_sockaddr_in()
        server  = NULL
        client  = NULL
        myloop  = NULL

        func start
                myloop = uv_default_loop()
                server = new_uv_tcp_t()
                uv_tcp_init(myloop, server)
                uv_ip4_addr("127.0.0.1", DEFAULT_PORT, addr)
                uv_tcp_bind(server, addr, 0)
```

(continues on next page)

(continued from previous page)

```
                r = uv_listen(server, DEFAULT_BACKLOG, Method(:newconnection) )
                if r
                        ? "Listen error " + uv_strerror(r)
                        return 1
                ok
                uv_run(myloop, UV_RUN_DEFAULT)
                destroy_uv_tcp_t(server)
                destroy_uv_sockaddr_in(addr)

        func newconnection
                ? "New Connection"
                aPara   = uv_Eventpara(server,:connect)
                nStatus = aPara[2]
                if nStatus < 0
                        ? "New connection error : " + nStatus
                        return
                ok
                client = new_uv_tcp_t()
                uv_tcp_init(myloop, client)
                if uv_accept(server, client) = 0
                                uv_read_start(client, uv_myalloccallback(),
                                                        Method(:echo_read))
                ok

        func echo_read
                aPara = uv_Eventpara(client,:read)
                nRead = aPara[2]
                buf   = aPara[3]
                if nRead > 0
                        req = new_uv_write_t()
                                wrbuf = uv_buf_init(get_uv_buf_t_base(buf), nread)
                        uv_write(req, client, wrbuf, 1, Method(:echo_write))
                        ? uv_buf2str(wrbuf)
                        message = "message from the server to the client"
                        buf = new_uv_buf_t()
                        set_uv_buf_t_len(buf,len(message))
                        set_uv_buf_t_base(buf,varptr("message",:char))
                        uv_write(req, client, buf, 1, Method(:echo_write))
                ok

        func echo_write
                aPara = uv_Eventpara(client,:read)
                req   = aPara[1]
```

Output:

When we run the client, We will see the message “New Connection”

Then the message “hello from the client”

```
Testing RingLibuv - Server Side - Using Classes
New Connection
hello from the client
```

86.6 Client Example Using Classes

Example:

```
load "libuv.ring"
load "objectslib.ring"

? "Testing RingLibuv - Client Side - Using Classes"

open_object(:MyClient)

Class MyClient from ObjectControllerParent

        DEFAULT_PORT    = 13370
        DEFAULT_BACKLOG = 1024

        addr    = new_sockaddr_in()
        connect = NULL
        buffer  = null
        socket  = null

        func start
                myloop  = uv_default_loop()
                Socket  = new_uv_tcp_t()
                connect = new_uv_connect_t()
                uv_tcp_init(myloop, Socket)
                uv_ip4_addr("127.0.0.1", DEFAULT_PORT, addr)
                uv_tcp_connect(connect,Socket, addr, Method(:connect))
                uv_run(myloop, UV_RUN_DEFAULT)
                destroy_uv_tcp_t(socket)
                destroy_uv_connect_t(connect)

        func connect
                ? "Client: Start Connection"
                aPara   = uv_Eventpara(connect,:connect)
                req     = aPara[1]
                nStatus = aPara[2]
                if nStatus = -1
                        ? "Error : on_write_end "
                        return
                ok
                buf = new_uv_buf_t()
                message = "hello from the client"
                set_uv_buf_t_len(buf,len(message))
                set_uv_buf_t_base(buf,varptr("message",:char))
                tcp       = get_uv_connect_t_handle(req)
                write_req = new_uv_write_t()
                buf_count = 1
                uv_write(write_req, tcp, buf, buf_count, Method(:on_write_end))

        func on_write_end
                        uv_read_start(socket, uv_myalloccallback(), Method(:echo_read))
```

(continues on next page)

(continued from previous page)

```
        func echo_read
                aPara = uv_Eventpara(socket,:read)
                nRead = aPara[2]
                buf   = aPara[3]
                if nRead > 0
                                wrbuf = uv_buf_init(get_uv_buf_t_base(buf), nread);
                        ? uv_buf2str(wrbuf)
                ok
```

Output:

We will run the client after the server

```
Testing RingLibuv - Client Side - Using Classes
Client: Start Connection
hello from the client
message from the server to the client
```

86.7 Threads Example

Example:

```
load "libuv.ring"

? "Testing RingLibuv - Threads"

func main
        one_id = new_uv_thread_t()
        two_id = new_uv_thread_t()
        uv_thread_create(one_id, "one()")
        uv_thread_create(two_id, "two()")
        uv_thread_join(one_id)
        uv_thread_join(two_id)
        destroy_uv_thread_t(one_id)
        destroy_uv_thread_t(two_id)

func one
        ? "Message from the First Thread!"

func two
        ? "Message from the Second Thread!"
```

Output:

```
Testing RingLibuv - Threads
Message from the First Thread!
Message from the Second Thread!
```

86.8 Threads Example - Using Classes

Example:

```
load "libuv.ring"
load "objectslib.ring"

? "Testing RingLibuv - Threads - Using Classes"

open_object(:MyThreads)

class MyThreads from ObjectControllerParent

        func Start
                one_id = new_uv_thread_t()
                two_id = new_uv_thread_t()
                uv_thread_create(one_id, Method(:One))
                uv_thread_create(two_id, Method(:Two))
                uv_thread_join(one_id)
                uv_thread_join(two_id)
                destroy_uv_thread_t(one_id)
                destroy_uv_thread_t(two_id)

        func one
                ? "Message from the First Thread!"

        func Two
                ? "Message from the Second Thread!"
```

Output:

```
Testing RingLibuv - Threads - Using Classes
Message from the First Thread!
Message from the Second Thread!
```

CHAPTER

EIGHTYSEVEN

RINGLIBUV FUNCTIONS REFERENCE

87.1 Introduction

In this chapter we have a list of the supported functions by this extension

87.2 Reference

- int uv_loop_init(uv_loop_t* loop)
- int uv_loop_configure(uv_loop_t* loop, uv_loop_option option, int)
- int uv_loop_close(uv_loop_t* loop)
- uv_loop_t* uv_default_loop(void)
- int uv_run(uv_loop_t* loop, uv_run_mode mode)
- int uv_loop_alive(const uv_loop_t* loop)
- void uv_stop(uv_loop_t* loop)
- size_t uv_loop_size(void)
- int uv_backend_fd(const uv_loop_t* loop)
- int uv_backend_timeout(const uv_loop_t* loop)
- uint64_t uv_now(const uv_loop_t* loop)
- void uv_update_time(uv_loop_t* loop)
- void uv_walk(uv_loop_t* loop, uv_walk_cb walk_cb, void* arg)
- void uv_walk_2(uv_loop_t* loop, uv_walk_cb walk_cb, void* arg)
- int uv_loop_fork(uv_loop_t* loop)
- int uv_is_active(const uv_handle_t* handle)
- int uv_is_closing(const uv_handle_t* handle)
- void uv_close(uv_handle_t* handle, uv_close_cb close_cb)
- void uv_close_2(uv_handle_t* handle, uv_close_cb close_cb)
- void uv_ref(uv_handle_t* handle)
- void uv_unref(uv_handle_t* handle)
- int uv_has_ref(const uv_handle_t* handle)

- size_t uv_handle_size(uv_handle_type type)
- int uv_send_buffer_size(uv_handle_t* handle, int* value)
- int uv_recv_buffer_size(uv_handle_t* handle, int* value)
- int uv_fileno(const uv_handle_t* handle, uv_os_fd_t* fd)
- int uv_cancel(uv_req_t* req)
- size_t uv_req_size(uv_req_type type)
- int uv_timer_init(uv_loop_t* loop, uv_timer_t* handle)
- int uv_timer_start(uv_timer_t* handle, uv_timer_cb cb, uint64_t timeout, uint64_t repeat)
- int uv_timer_start_2(uv_timer_t* handle, uv_timer_cb cb, uint64_t timeout, uint64_t repeat)
- int uv_timer_stop(uv_timer_t* handle)
- int uv_timer_again(uv_timer_t* handle)
- void uv_timer_set_repeat(uv_timer_t* handle, uint64_t repeat)
- uint64_t uv_timer_get_repeat(const uv_timer_t* handle)
- int uv_prepare_init(uv_loop_t* loop, uv_prepare_t* prepare)
- int uv_prepare_start(uv_prepare_t* prepare, uv_prepare_cb cb)
- int uv_prepare_start_2(uv_prepare_t* prepare, uv_prepare_cb cb)
- int uv_prepare_stop(uv_prepare_t* prepare)
- int uv_check_init(uv_loop_t* loop, uv_check_t* check)
- int uv_check_start(uv_check_t* check, uv_check_cb cb)
- int uv_check_start_2(uv_check_t* check, uv_check_cb cb)
- int uv_check_stop(uv_check_t* check)
- int uv_idle_init(uv_loop_t* loop, uv_idle_t* idle)
- int uv_idle_start(uv_idle_t* idle, uv_idle_cb cb)
- int uv_idle_start_2(uv_idle_t* idle, uv_idle_cb cb)
- int uv_idle_stop(uv_idle_t* idle)
- int uv_async_init(uv_loop_t* loop, uv_async_t* async, uv_async_cb async_cb)
- int uv_async_init_2(uv_loop_t* loop, uv_async_t* async, uv_async_cb async_cb)
- int uv_async_send(uv_async_t* async)
- int uv_poll_init(uv_loop_t* loop, uv_poll_t* handle, int fd)
- int uv_poll_init_socket(uv_loop_t* loop, uv_poll_t* handle, uv_os_sock_t socket)
- int uv_poll_start(uv_poll_t* handle, int events, uv_poll_cb cb)
- int uv_poll_start_2(uv_poll_t* handle, int events, uv_poll_cb cb)
- int uv_poll_stop(uv_poll_t* poll)
- int uv_signal_init(uv_loop_t* loop, uv_signal_t* signal)
- int uv_signal_start(uv_signal_t* signal, uv_signal_cb cb, int signum)
- int uv_signal_start_2(uv_signal_t* signal, uv_signal_cb cb, int signum)

- int uv_signal_start_oneshot(uv_signal_t* signal, uv_signal_cb cb, int signum)
- int uv_signal_start_oneshot_2(uv_signal_t* signal, uv_signal_cb cb, int signum)
- int uv_signal_stop(uv_signal_t* signal)
- void uv_disable_stdio_inheritance(void)
- int uv_spawn(uv_loop_t* loop, uv_process_t* handle, const uv_process_options_t* options)
- int uv_process_kill(uv_process_t* handle, int signum)
- int uv_kill(int pid, int signum)
- int uv_shutdown(uv_shutdown_t* req, uv_stream_t* handle, uv_shutdown_cb cb)
- int uv_shutdown_2(uv_shutdown_t* req, uv_stream_t* handle, uv_shutdown_cb cb)
- int uv_listen(uv_stream_t* stream, int backlog, uv_connection_cb cb)
- int uv_listen_2(uv_stream_t* stream, int backlog, uv_connection_cb cb)
- int uv_accept(uv_stream_t* server, uv_stream_t* client)
- int uv_read_start(uv_stream_t* stream, uv_alloc_cb alloc_cb, uv_read_cb read_cb)
- int uv_read_start_2(uv_stream_t* stream, uv_alloc_cb alloc_cb, uv_read_cb read_cb)
- int uv_read_stop(uv_stream_t*)
- int uv_write(uv_write_t* req, uv_stream_t* handle, uv_buf_t *bufs, unsigned int nbufs, uv_write_cb cb)
- int uv_write_2(uv_write_t* req, uv_stream_t* handle, uv_buf_t *bufs, unsigned int nbufs, uv_write_cb cb)
- int uv_write2(uv_write_t* req, uv_stream_t* handle, uv_buf_t *bufs, unsigned int nbufs, uv_stream_t* send_handle, uv_write_cb cb)
- int uv_write2_2(uv_write_t* req, uv_stream_t* handle, uv_buf_t *bufs, unsigned int nbufs, uv_stream_t* send_handle, uv_write_cb cb)
- int uv_try_write(uv_stream_t* handle, uv_buf_t *bufs, unsigned int nbufs)
- int uv_is_readable(const uv_stream_t* handle)
- int uv_is_writable(const uv_stream_t* handle)
- int uv_stream_set_blocking(uv_stream_t* handle, int blocking)
- int uv_tcp_init(uv_loop_t* loop, uv_tcp_t* handle)
- int uv_tcp_init_ex(uv_loop_t* loop, uv_tcp_t* handle, unsigned int flags)
- int uv_tcp_open(uv_tcp_t* handle, uv_os_sock_t sock)
- int uv_tcp_nodelay(uv_tcp_t* handle, int enable)
- int uv_tcp_keepalive(uv_tcp_t* handle, int enable, unsigned int delay)
- int uv_tcp_simultaneous_accepts(uv_tcp_t* handle, int enable)
- int uv_tcp_bind(uv_tcp_t *handle,sockaddr *addr,unsigned int flags)
- int uv_tcp_getsockname(const uv_tcp_t* handle, struct sockaddr* name, int* namelen)
- int uv_tcp_getpeername(const uv_tcp_t* handle, struct sockaddr* name, int* namelen)
- int uv_tcp_connect(uv_connect_t* req, uv_tcp_t* handle, sockaddr * addr, uv_connect_cb cb)
- int uv_tcp_connect_2(uv_connect_t* req, uv_tcp_t* handle, sockaddr * addr, uv_connect_cb cb)
- int uv_pipe_init(uv_loop_t* loop, uv_pipe_t* handle, int ipc)

- int uv_pipe_open(uv_pipe_t* handle, uv_file file)
- int uv_pipe_bind(uv_pipe_t* handle, const char * name)
- void uv_pipe_connect(uv_connect_t* req, uv_pipe_t* handle, const char * name, uv_connect_cb cb)
- void uv_pipe_connect_2(uv_connect_t* req, uv_pipe_t* handle, const char * name, uv_connect_cb cb)
- int uv_pipe_getsockname(const uv_pipe_t* handle, char* buffer, size_t* size)
- int uv_pipe_getpeername(const uv_pipe_t* handle, char* buffer, size_t* size)
- void uv_pipe_pending_instances(uv_pipe_t* handle, int count)
- int uv_pipe_pending_count(uv_pipe_t* handle)
- uv_handle_type uv_pipe_pending_type(uv_pipe_t* handle)
- int uv_pipe_chmod(uv_pipe_t* handle, int flags)
- int uv_tty_init(uv_loop_t* loop, uv_tty_t* handle, uv_file fd, int readable)
- int uv_tty_set_mode(uv_tty_t* handle, uv_tty_mode_t mode)
- int uv_tty_reset_mode(void)
- int uv_tty_get_winsize(uv_tty_t* handle, int* width, int* height)
- int uv_udp_init(uv_loop_t* loop, uv_udp_t* handle)
- int uv_udp_init_ex(uv_loop_t* loop, uv_udp_t* handle, unsigned int flags)
- int uv_udp_open(uv_udp_t* handle, uv_os_sock_t sock)
- int uv_udp_bind(uv_udp_t* handle, sockaddr * addr, unsigned int flags)
- int uv_udp_getsockname(const uv_udp_t* handle, struct sockaddr* name, int* namelen)
- int uv_udp_set_membership(uv_udp_t* handle, const char * multicast_addr, const char * interface_addr, uv_membership membership)
- int uv_udp_set_multicast_loop(uv_udp_t* handle, int on)
- int uv_udp_set_multicast_ttl(uv_udp_t* handle, int ttl)
- int uv_udp_set_multicast_interface(uv_udp_t* handle, const char * interface_addr)
- int uv_udp_set_broadcast(uv_udp_t* handle, int on)
- int uv_udp_set_ttl(uv_udp_t* handle, int ttl)
- int uv_udp_send(uv_udp_send_t* req, uv_udp_t* handle, uv_buf_t *bufs, unsigned int nbufs, sockaddr * addr, uv_udp_send_cb send_cb)
- int uv_udp_send_2(uv_udp_send_t* req, uv_udp_t* handle, uv_buf_t *bufs, unsigned int nbufs, sockaddr * addr, uv_udp_send_cb send_cb)
- int uv_udp_try_send(uv_udp_t* handle, uv_buf_t *bufs, unsigned int nbufs, sockaddr * addr)
- int uv_udp_recv_start(uv_udp_t* handle, uv_alloc_cb alloc_cb, uv_udp_recv_cb recv_cb)
- int uv_udp_recv_start_2(uv_udp_t* handle, uv_alloc_cb alloc_cb, uv_udp_recv_cb recv_cb)
- int uv_udp_recv_stop(uv_udp_t* handle)
- int uv_fs_event_init(uv_loop_t* loop, uv_fs_event_t* handle)
- int uv_fs_event_start(uv_fs_event_t* handle, uv_fs_event_cb cb, const char * path, unsigned int flags)
- int uv_fs_event_start_2(uv_fs_event_t* handle, uv_fs_event_cb cb, const char * path, unsigned int flags)

- int uv_fs_event_stop(uv_fs_event_t* handle)
- int uv_fs_event_getpath(uv_fs_event_t* handle, char* buffer, size_t* size)
- int uv_fs_poll_init(uv_loop_t* loop, uv_fs_poll_t* handle)
- int uv_fs_poll_start(uv_fs_poll_t* handle, uv_fs_poll_cb poll_cb, const char * path, unsigned int interval)
- int uv_fs_poll_start_2(uv_fs_poll_t* handle, uv_fs_poll_cb poll_cb, const char * path, unsigned int interval)
- int uv_fs_poll_stop(uv_fs_poll_t* handle)
- int uv_fs_poll_getpath(uv_fs_poll_t* handle, char* buffer, size_t* size)
- void uv_fs_req_cleanup(uv_fs_t* req)
- int uv_fs_close(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_open(uv_loop_t* loop, uv_fs_t* req, const char * path, int flags, int mode, uv_fs_cb cb)
- int uv_fs_read(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_buf_t *bufs, unsigned int nbufs, int64_t offset, uv_fs_cb cb)
- int uv_fs_unlink(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_write(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_buf_t *bufs, unsigned int nbufs, int64_t offset, uv_fs_cb cb)
- int uv_fs_mkdir(uv_loop_t* loop, uv_fs_t* req, const char * path, int mode, uv_fs_cb cb)
- int uv_fs_mkdtemp(uv_loop_t* loop, uv_fs_t* req, const char * tpl, uv_fs_cb cb)
- int uv_fs_rmdir(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_scandir(uv_loop_t* loop, uv_fs_t* req, const char * path, int flags, uv_fs_cb cb)
- int uv_fs_scandir_next(uv_fs_t* req, uv_dirent_t* ent)
- int uv_fs_stat(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_fstat(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_lstat(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_rename(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, uv_fs_cb cb)
- int uv_fs_fsync(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_fdatasync(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_ftruncate(uv_loop_t* loop, uv_fs_t* req, uv_file file, int64_t offset, uv_fs_cb cb)
- int uv_fs_copyfile(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, int flags, uv_fs_cb cb)
- int uv_fs_sendfile(uv_loop_t* loop, uv_fs_t* req, uv_file out_fd, uv_file in_fd, int64_t in_offset, size_t length, uv_fs_cb cb)
- int uv_fs_access(uv_loop_t* loop, uv_fs_t* req, const char * path, int mode, uv_fs_cb cb)
- int uv_fs_chmod(uv_loop_t* loop, uv_fs_t* req, const char * path, int mode, uv_fs_cb cb)
- int uv_fs_fchmod(uv_loop_t* loop, uv_fs_t* req, uv_file file, int mode, uv_fs_cb cb)
- int uv_fs_utime(uv_loop_t* loop, uv_fs_t* req, const char * path, double atime, double mtime, uv_fs_cb cb)
- int uv_fs_futime(uv_loop_t* loop, uv_fs_t* req, uv_file file, double atime, double mtime, uv_fs_cb cb)
- int uv_fs_link(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, uv_fs_cb cb)

- int uv_fs_symlink(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, int flags, uv_fs_cb cb)
- int uv_fs_readlink(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_realpath(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_chown(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_uid_t uid, uv_gid_t gid, uv_fs_cb cb)
- int uv_fs_fchown(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_uid_t uid, uv_gid_t gid, uv_fs_cb cb)
- int uv_fs_close_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_open_2(uv_loop_t* loop, uv_fs_t* req, const char * path, int flags, int mode, uv_fs_cb cb)
- int uv_fs_read_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_buf_t *bufs, unsigned int nbufs, int64_t offset, uv_fs_cb cb)
- int uv_fs_unlink_2(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_write_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_buf_t *bufs, unsigned int nbufs, int64_t offset, uv_fs_cb cb)
- int uv_fs_mkdir_2(uv_loop_t* loop, uv_fs_t* req, const char * path, int mode, uv_fs_cb cb)
- int uv_fs_mkdtemp_2(uv_loop_t* loop, uv_fs_t* req, const char * tpl, uv_fs_cb cb)
- int uv_fs_rmdir_2(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_scandir_2(uv_loop_t* loop, uv_fs_t* req, const char * path, int flags, uv_fs_cb cb)
- int uv_fs_stat_2(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_fstat_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_lstat_2(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_rename_2(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, uv_fs_cb cb)
- int uv_fs_fsync_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_fdatasync_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_fs_cb cb)
- int uv_fs_ftruncate_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, int64_t offset, uv_fs_cb cb)
- int uv_fs_copyfile_2(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, int flags, uv_fs_cb cb)
- int uv_fs_sendfile_2(uv_loop_t* loop, uv_fs_t* req, uv_file out_fd, uv_file in_fd, int64_t in_offset, size_t length, uv_fs_cb cb)
- int uv_fs_access_2(uv_loop_t* loop, uv_fs_t* req, const char * path, int mode, uv_fs_cb cb)
- int uv_fs_chmod_2(uv_loop_t* loop, uv_fs_t* req, const char * path, int mode, uv_fs_cb cb)
- int uv_fs_fchmod_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, int mode, uv_fs_cb cb)
- int uv_fs_utime_2(uv_loop_t* loop, uv_fs_t* req, const char * path, double atime, double mtime, uv_fs_cb cb)
- int uv_fs_futime_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, double atime, double mtime, uv_fs_cb cb)
- int uv_fs_link_2(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, uv_fs_cb cb)
- int uv_fs_symlink_2(uv_loop_t* loop, uv_fs_t* req, const char * path, const char * new_path, int flags, uv_fs_cb cb)
- int uv_fs_readlink_2(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)
- int uv_fs_realpath_2(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_fs_cb cb)

- int uv_fs_chown_2(uv_loop_t* loop, uv_fs_t* req, const char * path, uv_uid_t uid, uv_gid_t gid, uv_fs_cb cb)
- int uv_fs_fchown_2(uv_loop_t* loop, uv_fs_t* req, uv_file file, uv_uid_t uid, uv_gid_t gid, uv_fs_cb cb)
- int uv_queue_work(uv_loop_t* loop, uv_work_t* req, uv_work_cb work_cb, uv_after_work_cb after_work_cb)
- int uv_queue_work_2(uv_loop_t* loop, uv_work_t* req, uv_work_cb work_cb, uv_after_work_cb after_work_cb)
- int uv_getaddrinfo(uv_loop_t* loop, uv_getaddrinfo_t* req, uv_getaddrinfo_cb getaddrinfo_cb, const char * node, const char * service, const struct addrinfo* hints)
- int uv_getaddrinfo_2(uv_loop_t* loop, uv_getaddrinfo_t* req, uv_getaddrinfo_cb getaddrinfo_cb, const char * node, const char * service, const struct addrinfo* hints)
- void uv_freeaddrinfo(struct addrinfo* ai)
- int uv_getnameinfo(uv_loop_t* loop, uv_getnameinfo_t* req, uv_getnameinfo_cb getnameinfo_cb, sockaddr * addr, int flags)
- int uv_getnameinfo_2(uv_loop_t* loop, uv_getnameinfo_t* req, uv_getnameinfo_cb getnameinfo_cb, sockaddr * addr, int flags)
- int uv_dlopen(const char * filename, uv_lib_t* lib)
- void uv_dlclose(uv_lib_t* lib)
- int uv_dlsym(uv_lib_t* lib, const char * name, void** ptr)
- const char * uv_dlerror(const uv_lib_t* lib)
- int uv_thread_create(uv_thread_t* tid, uv_thread_cb entry, void* arg)
- int uv_thread_create_2(uv_thread_t* tid, uv_thread_cb entry, void* arg)
- uv_thread_t uv_thread_self(void)
- int uv_thread_join(uv_thread_t *tid)
- int uv_thread_equal(const uv_thread_t* t1, const uv_thread_t* t2)
- int uv_key_create(uv_key_t* key)
- void uv_key_delete(uv_key_t* key)
- void* uv_key_get(uv_key_t* key)
- void uv_key_set(uv_key_t* key, void* value)
- int uv_mutex_init(uv_mutex_t* handle)
- int uv_mutex_init_recursive(uv_mutex_t* handle)
- void uv_mutex_destroy(uv_mutex_t* handle)
- void uv_mutex_lock(uv_mutex_t* handle)
- int uv_mutex_trylock(uv_mutex_t* handle)
- void uv_mutex_unlock(uv_mutex_t* handle)
- int uv_rwlock_init(uv_rwlock_t* rwlock)
- void uv_rwlock_destroy(uv_rwlock_t* rwlock)
- void uv_rwlock_rdlock(uv_rwlock_t* rwlock)
- int uv_rwlock_tryrdlock(uv_rwlock_t* rwlock)
- void uv_rwlock_rdunlock(uv_rwlock_t* rwlock)

- void uv_rwlock_wrlock(uv_rwlock_t* rwlock)
- int uv_rwlock_trywrlock(uv_rwlock_t* rwlock)
- void uv_rwlock_wrunlock(uv_rwlock_t* rwlock)
- int uv_sem_init(uv_sem_t* sem, unsigned int value)
- void uv_sem_destroy(uv_sem_t* sem)
- void uv_sem_post(uv_sem_t* sem)
- void uv_sem_wait(uv_sem_t* sem)
- int uv_sem_trywait(uv_sem_t* sem)
- int uv_cond_init(uv_cond_t* cond)
- void uv_cond_destroy(uv_cond_t* cond)
- void uv_cond_signal(uv_cond_t* cond)
- void uv_cond_broadcast(uv_cond_t* cond)
- void uv_cond_wait(uv_cond_t* cond, uv_mutex_t* mutex)
- int uv_cond_timedwait(uv_cond_t* cond, uv_mutex_t* mutex, uint64_t timeout)
- int uv_barrier_init(uv_barrier_t* barrier, unsigned int count)
- void uv_barrier_destroy(uv_barrier_t* barrier)
- int uv_barrier_wait(uv_barrier_t* barrier)
- uv_handle_type uv_guess_handle(uv_file file)
- int uv_replace_allocator(uv_malloc_func malloc_func, uv_realloc_func realloc_func, uv_calloc_func calloc_func, uv_free_func free_func)
- uv_buf_t uv_buf_init(char* base, unsigned int len)
- char** uv_setup_args(int argc, char** argv)
- int uv_get_process_title(char* buffer, size_t size)
- int uv_set_process_title(const char * title)
- int uv_resident_set_memory(size_t* rss)
- int uv_uptime(double* uptime)
- int uv_getrusage(uv_rusage_t* rusage)
- uv_pid_t uv_os_getpid(void)
- uv_pid_t uv_os_getppid(void)
- int uv_cpu_info(uv_cpu_info_t** cpu_infos, int* count)
- void uv_free_cpu_info(uv_cpu_info_t* cpu_infos, int count)
- int uv_interface_addresses(uv_interface_address_t** addresses, int* count)
- void uv_free_interface_addresses(uv_interface_address_t* addresses, int count)
- int uv_ip6_addr(const char * ip, int port, sockaddr_in6* addr)
- int uv_ip4_name(sockaddr_in* src, char* dst, size_t size)
- int uv_ip6_name(sockaddr_in6* src, char* dst, size_t size)

- int uv_inet_ntop(int af, const void* src, char* dst, size_t size)
- int uv_inet_pton(int af, const char * src, void* dst)
- int uv_if_indextoname(unsigned int ifindex, char* buffer, size_t* size)
- int uv_if_indextoiid(unsigned int ifindex, char* buffer, size_t* size)
- int uv_exepath(char* buffer, size_t* size)
- int uv_cwd(char* buffer, size_t* size)
- int uv_chdir(const char * dir)
- int uv_os_homedir(char* buffer, size_t* size)
- int uv_os_tmpdir(char* buffer, size_t* size)
- int uv_os_get_passwd(uv_passwd_t* pwd)
- void uv_os_free_passwd(uv_passwd_t* pwd)
- uint64_t uv_get_total_memory(void)
- uint64_t uv_hrtime(void)
- void uv_print_all_handles(uv_loop_t* loop, FILE* stream)
- void uv_print_active_handles(uv_loop_t* loop, FILE* stream)
- int uv_os_getenv(const char * name, char* buffer, size_t* size)
- int uv_os_setenv(const char * name, const char * value)
- int uv_os_unsetenv(const char * name)
- int uv_os_gethostname(char* buffer, size_t* size)

CHAPTER

EIGHTYEIGHT

RINGFREEGLUT FUNCTIONS REFERENCE

88.1 Introduction

In this chapter we have a list of the supported functions by this extension

88.2 Reference

- GLUT_RGB
- GLUT_RGBA
- GLUT_INDEX
- GLUT_SINGLE
- GLUT_DOUBLE
- GLUT_ACCUM
- GLUT_ALPHA
- GLUT_DEPTH
- GLUT_STENCIL
- GLUT_MULTISAMPLE
- GLUT_STEREO
- GLUT_LUMINANCE
- GLUT_KEY_F1
- GLUT_KEY_F2
- GLUT_KEY_F3
- GLUT_KEY_F4
- GLUT_KEY_F5
- GLUT_KEY_F6
- GLUT_KEY_F7
- GLUT_KEY_F8
- GLUT_KEY_F9
- GLUT_KEY_F10

- GLUT_KEY_F11
- GLUT_KEY_F12
- GLUT_KEY_LEFT
- GLUT_KEY_UP
- GLUT_KEY_RIGHT
- GLUT_KEY_DOWN
- GLUT_KEY_PAGE_UP
- GLUT_KEY_PAGE_DOWN
- GLUT_KEY_HOME
- GLUT_KEY_END
- GLUT_KEY_INSERT
- GLUT_LEFT_BUTTON
- GLUT_MIDDLE_BUTTON
- GLUT_RIGHT_BUTTON
- GLUT_DOWN
- GLUT_UP
- GLUT_LEFT
- GLUT_ENTERED
- GLUT_MENU_NOT_IN_USE
- GLUT_MENU_IN_USE
- GLUT_NOT_VISIBLE
- GLUT_VISIBLE
- GLUT_HIDDEN
- GLUT_FULLY_RETAINED
- GLUT_PARTIALLY_RETAINED
- GLUT_FULLY_COVERED
- GLUT_WINDOW_X
- GLUT_WINDOW_Y
- GLUT_WINDOW_WIDTH
- GLUT_WINDOW_HEIGHT
- GLUT_WINDOW_BUFFER_SIZE
- GLUT_WINDOW_STENCIL_SIZE
- GLUT_WINDOW_DEPTH_SIZE
- GLUT_WINDOW_RED_SIZE
- GLUT_WINDOW_GREEN_SIZE
- GLUT_WINDOW_BLUE_SIZE

- GLUT_WINDOW_ALPHA_SIZE
- GLUT_WINDOW_ACCUM_RED_SIZE
- GLUT_WINDOW_ACCUM_GREEN_SIZE
- GLUT_WINDOW_ACCUM_BLUE_SIZE
- GLUT_WINDOW_ACCUM_ALPHA_SIZE
- GLUT_WINDOW_DOUBLEBUFFER
- GLUT_WINDOW_RGBA
- GLUT_WINDOW_PARENT
- GLUT_WINDOW_NUM_CHILDREN
- GLUT_WINDOW_COLORMAP_SIZE
- GLUT_WINDOW_NUM_SAMPLES
- GLUT_WINDOW_STEREO
- GLUT_WINDOW_CURSOR
- GLUT_SCREEN_WIDTH
- GLUT_SCREEN_HEIGHT
- GLUT_SCREEN_WIDTH_MM
- GLUT_SCREEN_HEIGHT_MM
- GLUT_MENU_NUM_ITEMS
- GLUT_DISPLAY_MODE_POSSIBLE
- GLUT_INIT_WINDOW_X
- GLUT_INIT_WINDOW_Y
- GLUT_INIT_WINDOW_WIDTH
- GLUT_INIT_WINDOW_HEIGHT
- GLUT_INIT_DISPLAY_MODE
- GLUT_ELAPSED_TIME
- GLUT_WINDOW_FORMAT_ID
- GLUT_HAS_KEYBOARD
- GLUT_HAS_MOUSE
- GLUT_HAS_SPACEBALL
- GLUT_HAS_DIAL_AND_BUTTON_BOX
- GLUT_HAS_TABLET
- GLUT_NUM_MOUSE_BUTTONS
- GLUT_NUM_SPACEBALL_BUTTONS
- GLUT_NUM_BUTTON_BOX_BUTTONS
- GLUT_NUM_DIALS
- GLUT_NUM_TABLET_BUTTONS

- GLUT_DEVICE_IGNORE_KEY_REPEAT
- GLUT_DEVICE_KEY_REPEAT
- GLUT_HAS_JOYSTICK
- GLUT_OWNS_JOYSTICK
- GLUT_JOYSTICK_BUTTONS
- GLUT_JOYSTICK_AXES
- GLUT_JOYSTICK_POLL_RATE
- GLUT_OVERLAY_POSSIBLE
- GLUT_LAYER_IN_USE
- GLUT_HAS_OVERLAY
- GLUT_TRANSPARENT_INDEX
- GLUT_NORMAL_DAMAGED
- GLUT_OVERLAY_DAMAGED
- GLUT_VIDEO_RESIZE_POSSIBLE
- GLUT_VIDEO_RESIZE_IN_USE
- GLUT_VIDEO_RESIZE_X_DELTA
- GLUT_VIDEO_RESIZE_Y_DELTA
- GLUT_VIDEO_RESIZE_WIDTH_DELTA
- GLUT_VIDEO_RESIZE_HEIGHT_DELTA
- GLUT_VIDEO_RESIZE_X
- GLUT_VIDEO_RESIZE_Y
- GLUT_VIDEO_RESIZE_WIDTH
- GLUT_VIDEO_RESIZE_HEIGHT
- GLUT_NORMAL
- GLUT_OVERLAY
- GLUT_ACTIVE_SHIFT
- GLUT_ACTIVE_CTRL
- GLUT_ACTIVE_ALT
- GLUT_CURSOR_RIGHT_ARROW
- GLUT_CURSOR_LEFT_ARROW
- GLUT_CURSOR_INFO
- GLUT_CURSOR_DESTROY
- GLUT_CURSOR_HELP
- GLUT_CURSOR_CYCLE
- GLUT_CURSOR_SPRAY
- GLUT_CURSOR_WAIT

- GLUT_CURSOR_TEXT
- GLUT_CURSOR_CROSSHAIR
- GLUT_CURSOR_UP_DOWN
- GLUT_CURSOR_LEFT_RIGHT
- GLUT_CURSOR_TOP_SIDE
- GLUT_CURSOR_BOTTOM_SIDE
- GLUT_CURSOR_LEFT_SIDE
- GLUT_CURSOR_RIGHT_SIDE
- GLUT_CURSOR_TOP_LEFT_CORNER
- GLUT_CURSOR_TOP_RIGHT_CORNER
- GLUT_CURSOR_BOTTOM_RIGHT_CORNER
- GLUT_CURSOR_BOTTOM_LEFT_CORNER
- GLUT_CURSOR_INHERIT
- GLUT_CURSOR_NONE
- GLUT_CURSOR_FULL_CROSSHAIR
- GLUT_RED
- GLUT_GREEN
- GLUT_BLUE
- GLUT_KEY_REPEAT_OFF
- GLUT_KEY_REPEAT_ON
- GLUT_KEY_REPEAT_DEFAULT
- GLUT_JOYSTICK_BUTTON_A
- GLUT_JOYSTICK_BUTTON_B
- GLUT_JOYSTICK_BUTTON_C
- GLUT_JOYSTICK_BUTTON_D
- GLUT_GAME_MODE_ACTIVE
- GLUT_GAME_MODE_POSSIBLE
- GLUT_GAME_MODE_WIDTH
- GLUT_GAME_MODE_HEIGHT
- GLUT_GAME_MODE_PIXEL_DEPTH
- GLUT_GAME_MODE_REFRESH_RATE
- GLUT_GAME_MODE_DISPLAY_CHANGED
- GLUT_STROKE_ROMAN
- GLUT_STROKE_MONO_ROMAN
- GLUT_BITMAP_9_BY_15
- GLUT_BITMAP_8_BY_13

- GLUT_BITMAP_TIMES_ROMAN_10
- GLUT_BITMAP_TIMES_ROMAN_24
- GLUT_BITMAP_HELVETICA_10
- GLUT_BITMAP_HELVETICA_12
- GLUT_BITMAP_HELVETICA_18
- void glutInit(void)
- void glutDisplayFunc(const char *)
- void glutReshapeFunc(const char *)
- int glutEventWidth(void)
- int glutEventHeight(void)
- void glutIdleFunc(const char *)
- void glutKeyboardFunc(const char *)
- void glutSpecialFunc(const char *)
- void glutSpecialUpFunc(const char *)
- void glutMouseFunc(const char *)
- void glutMotionFunc(const char *)
- int glutCreateMenu(const char *)
- void glutMenuStatusFunc(const char *)
- int glutEventKey(void)
- int glutEventX(void)
- int glutEventY(void)
- int glutEventButton(void)
- int glutEventState(void)
- int glutEventValue(void)
- int glutEventStatus(void)
- void test_draw(void)
- void glutInitWindowPosition(int x, int y)
- void glutInitWindowSize(int width, int height)
- void glutInitDisplayMode(unsigned displayMode)
- void glutInitDisplayString(const char * displayMode)
- int glutCreateWindow(const char * title)
- int glutCreateSubWindow(int window, int x, int y, int width, int height)
- void glutDestroyWindow(int window)
- void glutSetWindow(int window)
- int glutGetWindow(void)
- void glutSetWindowTitle(const char * title)

- void glutSetIconTitle(const char * title)
- void glutReshapeWindow(int width, int height)
- void glutPositionWindow(int x, int y)
- void glutShowWindow(void)
- void glutHideWindow(void)
- void glutIconifyWindow(void)
- void glutPushWindow(void)
- void glutPopWindow(void)
- void glutFullScreen(void)
- void glutPostWindowRedisplay(int window)
- void glutPostRedisplay(void)
- void glutSwapBuffers(void)
- void glutWarpPointer(int x, int y)
- void glutSetCursor(int cursor)
- void glutEstablishOverlay(void)
- void glutRemoveOverlay(void)
- void glutUseLayer(GLenum layer)
- void glutPostOverlayRedisplay(void)
- void glutPostWindowOverlayRedisplay(int window)
- void glutShowOverlay(void)
- void glutHideOverlay(void)
- void glutDestroyMenu(int menu)
- int glutGetMenu(void)
- void glutSetMenu(int menu)
- void glutAddMenuEntry(const char * label, int value)
- void glutAddSubMenu(const char * label, int subMenu)
- void glutChangeToMenuEntry(int item, const char * label, int value)
- void glutChangeToSubMenu(int item, const char * label, int value)
- void glutRemoveMenuItem(int item)
- void glutAttachMenu(int button)
- void glutDetachMenu(int button)
- int glutGet(GLenum query)
- int glutDeviceGet(GLenum query)
- int glutGetModifiers(void)
- int glutLayerGet(GLenum query)
- void glutBitmapCharacter(void *font, int character)

- int glutBitmapWidth(void *font, int character)
- void glutStrokeCharacter(void *font, int character)
- int glutStrokeWidth(void *font, int character)
- int glutStrokeLength(void *font, char * string)
- GLfloat glutStrokeWidthf(void *font, int character)
- GLfloat glutStrokeLengthf(void *font, char *string)
- int glutBitmapLength(void *font, char * string)
- void glutWireCube(double size)
- void glutSolidCube(double size)
- void glutWireSphere(double radius, GLint slices, GLint stacks)
- void glutSolidSphere(double radius, GLint slices, GLint stacks)
- void glutWireCone(double base, double height, GLint slices, GLint stacks)
- void glutSolidCone(double base, double height, GLint slices, GLint stacks)
- void glutWireTorus(double innerRadius, double outerRadius, GLint sides, GLint rings)
- void glutSolidTorus(double innerRadius, double outerRadius, GLint sides, GLint rings)
- void glutWireDodecahedron(void)
- void glutSolidDodecahedron(void)
- void glutWireOctahedron(void)
- void glutSolidOctahedron(void)
- void glutWireTetrahedron(void)
- void glutSolidTetrahedron(void)
- void glutWireIcosahedron(void)
- void glutSolidIcosahedron(void)
- void glutWireTeapot(double size)
- void glutSolidTeapot(double size)
- void glutGameModeString(const char * string)
- int glutEnterGameMode(void)
- void glutLeaveGameMode(void)
- int glutGameModeGet(GLenum query)
- int glutVideoResizeGet(GLenum query)
- void glutSetupVideoResizing(void)
- void glutStopVideoResizing(void)
- void glutVideoResize(int x, int y, int width, int height)
- void glutVideoPan(int x, int y, int width, int height)
- void glutSetColor(int color, GLfloat red, GLfloat green, GLfloat blue)
- GLfloat glutGetColor(int color, int component)

- void glutCopyColormap(int window)
- void glutIgnoreKeyRepeat(int ignore)
- void glutSetKeyRepeat(int repeatMode)
- void glutForceJoystickFunc(void)
- int glutExtensionSupported(const char * extension)
- void glutReportErrors(void)
- void glutMainLoop(void)
- void glutCloseFunc(const char *)
- GLUT_KEY_NUM_LOCK
- GLUT_KEY_BEGIN
- GLUT_KEY_DELETE
- GLUT_KEY_SHIFT_L
- GLUT_KEY_SHIFT_R
- GLUT_KEY_CTRL_L
- GLUT_KEY_CTRL_R
- GLUT_KEY_ALT_L
- GLUT_KEY_ALT_R
- GLUT_ACTION_EXIT
- GLUT_ACTION_GLUTMAINLOOP_RETURNS
- GLUT_ACTION_CONTINUE_EXECUTION
- GLUT_CREATE_NEW_CONTEXT
- GLUT_USE_CURRENT_CONTEXT
- GLUT_FORCE_INDIRECT_CONTEXT
- GLUT_ALLOW_DIRECT_CONTEXT
- GLUT_TRY_DIRECT_CONTEXT
- GLUT_FORCE_DIRECT_CONTEXT
- GLUT_INIT_STATE
- GLUT_ACTION_ON_WINDOW_CLOSE
- GLUT_WINDOW_BORDER_WIDTH
- GLUT_WINDOW_BORDER_HEIGHT
- GLUT_WINDOW_HEADER_HEIGHT
- GLUT_VERSION
- GLUT_RENDERING_CONTEXT
- GLUT_DIRECT_RENDERING
- GLUT_FULL_SCREEN
- GLUT_SKIP_STALE_MOTION_EVENTS

- GLUT_AUX
- GLUT_AUX1
- GLUT_AUX2
- GLUT_AUX3
- GLUT_AUX4
- GLUT_INIT_MAJOR_VERSION
- GLUT_INIT_MINOR_VERSION
- GLUT_INIT_FLAGS
- GLUT_INIT_PROFILE
- GLUT_DEBUG
- GLUT_FORWARD_COMPATIBLE
- GLUT_CORE_PROFILE
- GLUT_COMPATIBILITY_PROFILE
- GLUT_CAPTIONLESS
- GLUT_BORDERLESS
- GLUT_SRGB
- GLUT_HAS_MULTI
- void glutMainLoopEvent(void)
- void glutLeaveMainLoop(void)
- void glutExit(void)
- void glutFullScreenToggle(void)
- void glutLeaveFullScreen(void)
- void glutSetMenuFont(int menuID, void *font)
- void glutSetOption(GLenum option_flag, int value)
- int *glutGetModeValues(GLenum mode, int * size)
- void *glutGetWindowData(void)
- void glutSetWindowData(void *data)
- void *glutGetMenuData(void)
- void glutSetMenuData(void *data)
- int glutBitmapHeight(void* font)
- GLfloat glutStrokeHeight(void* font)
- void glutBitmapString(void* font, char *string)
- void glutStrokeString(void* font, char *string)
- void glutWireRhombicDodecahedron(void)
- void glutSolidRhombicDodecahedron(void)
- void glutWireSierpinskiSponge(int num_levels, double *offset, double scale)

- void glutSolidSierpinskiSponge(int num_levels, double *offset, double scale)
- void glutWireCylinder(double radius, double height, GLint slices, GLint stacks)
- void glutSolidCylinder(double radius, double height, GLint slices, GLint stacks)
- void glutWireTeacup(double size)
- void glutSolidTeacup(double size)
- void glutWireTeaspoon(double size)
- void glutSolidTeaspoon(double size)
- void glutInitContextVersion(int majorVersion, int minorVersion)
- void glutInitContextFlags(int flags)
- void glutInitContextProfile(int profile)
- void glutSetVertexAttribCoord3(GLint attrib)
- void glutSetVertexAttribNormal(GLint attrib)
- void glutSetVertexAttribTexCoord2(GLint attrib)

CHAPTER

EIGHTYNINE

RINGSTBIMAGE FUNCTIONS REFERENCE

89.1 Introduction

In this chapter we have a list of the supported functions by this extension

Example (1):

```
# Load the library
        load "stbimage.ring"
# Image Information
        width=0 height=0 channels=0
# Ring will Free cData automatically in the end of the program
        cData = stbi_load("ring.jpg",:width,:height,:channels,STBI_rgb)
# Display the output
        ? "Size (bytes): " + len(cData)
        ? "Width : " + width
        ? "Height: " + height
        ? "Channels: " + channels
```

Output:

```
Size (bytes): 557371
Width : 563
Height: 330
Channels: 3
```

Example (2):

```
load "stbimage.ring"

width    = 640
height   = 480
channels = 3
cData    = space(width*height*channels)

? "Creating the image..."
t1 = clock()
nIndex=0
for x=1 to height
        for y=1 to width
                cData[nIndex++] = x*x
```

(continues on next page)

(continued from previous page)

```
                cData[nIndex++] = x*y
                cData[nIndex++] = x*2
        next
next
t2 = clock()

# Write the image
? "Writing mynewimage.bmp"
stbi_write_bmp("mynewimage.bmp", width, height, channels, cData)
t3 = clock()

? "Time to create the image : " + ((t2-t1)/clockspersecond()) + " seconds"
? "Time to save the image   : " + ((t3-t2)/clockspersecond()) + " seconds"

system("mynewimage.bmp")
```

Output:

```
Creating the image...
Writing mynewimage.bmp
Time to create the image : 0.12 seconds
Time to save the image   : 0.00 seconds
```

Screen Shot:

89.2 Constants

- STBI_default
- STBI_grey
- STBI_grey_alpha
- STBI_rgb
- STBI_rgb_alpha

89.3 Functions

- stbi_uc *stbi_load_from_memory(stbi_uc const *buffer, int len,int *x, int *y, int *channels_in_file, int desired_channels)
- stbi_uc *stbi_load(char const *filename, int *x, int *y, int *channels_in_file, int desired_channels)
- stbi_uc *stbi_load_from_file(FILE *f, int *x, int *y, int *channels_in_file, int desired_channels)
- stbi_us *stbi_load_16_from_memory(stbi_uc const *buffer, int len, int *x, int *y, int *channels_in_file, int desired_channels)
- stbi_us *stbi_load_16(char const *filename, int *x, int *y, int *channels_in_file, int desired_channels)
- stbi_us *stbi_load_from_file_16(FILE *f, int *x, int *y, int *channels_in_file, int desired_channels)
- float *stbi_loadf_from_memory(stbi_uc const *buffer, int len, int *x, int *y, int *channels_in_file, int desired_channels)
- float *stbi_loadf(char const *filename, int *x, int *y, int *channels_in_file, int desired_channels)
- float *stbi_loadf_from_file(FILE *f, int *x, int *y, int *channels_in_file, int desired_channels)
- void stbi_hdr_to_ldr_gamma(float gamma)
- void stbi_hdr_to_ldr_scale(float scale)
- void stbi_ldr_to_hdr_gamma(float gamma)
- void stbi_ldr_to_hdr_scale(float scale)
- int stbi_is_hdr_from_memory(stbi_uc const *buffer, int len)
- int stbi_is_hdr(char const *filename)
- int stbi_is_hdr_from_file(FILE *f)
- const char *stbi_failure_reason(void)
- int stbi_info_from_memory(stbi_uc const *buffer, int len, int *x, int *y, int *comp)
- int stbi_is_16_bit_from_memory(stbi_uc const *buffer, int len)
- int stbi_info(char const *filename,int *x,int *y,int *comp)
- int stbi_info_from_file(FILE *f,int *x,int *y,int *comp)
- int stbi_is_16_bit(char const *filename)
- int stbi_is_16_bit_from_file(FILE *f)
- void stbi_set_unpremultiply_on_load(int flag_true_if_should_unpremultiply)
- void stbi_convert_iphone_png_to_rgb(int flag_true_if_should_convert)

- void stbi_set_flip_vertically_on_load(int flag_true_if_should_flip)
- void stbi_set_flip_vertically_on_load_thread(int flag_true_if_should_flip)
- char *stbi_zlib_decode_malloc_guesssize(const char *buffer, int len, int initial_size, int *outlen)
- char *stbi_zlib_decode_malloc_guesssize_headerflag(const char *buffer, int len, int initial_size, int *outlen, int parse_header)
- char *stbi_zlib_decode_malloc(const char *buffer, int len, int *outlen)
- int stbi_zlib_decode_buffer(char *obuffer,int olen, const char *ibuffer, int ilen)
- char *stbi_zlib_decode_noheader_malloc(const char *buffer, int len, int *outlen)
- int stbi_zlib_decode_noheader_buffer(char *obuffer, int olen, const char *ibuffer, int ilen)
- int stbi_write_png(char const *filename, int w, int h, int comp, const void *data, int stride_in_bytes)
- int stbi_write_bmp(char const *filename, int w, int h, int comp, const void *data)
- int stbi_write_tga(char const *filename, int w, int h, int comp, const void *data)
- int stbi_write_jpg(char const *filename, int w, int h, int comp, const void *data, int quality)
- void stbi_flip_vertically_on_write(int flag)

CHAPTER

NINETY

RINGOPENGL (OPENGL 3.2) FUNCTIONS REFERENCE

90.1 Introduction

In this chapter we have a list of the supported functions by this extension

90.2 Reference

- GL_ZERO
- GL_FALSE
- GL_LOGIC_OP
- GL_NONE
- GL_TEXTURE_COMPONENTS
- GL_NO_ERROR
- GL_POINTS
- GL_CURRENT_BIT
- GL_TRUE
- GL_ONE
- GL_CLIENT_PIXEL_STORE_BIT
- GL_LINES
- GL_LINE_LOOP
- GL_POINT_BIT
- GL_CLIENT_VERTEX_ARRAY_BIT
- GL_LINE_STRIP
- GL_LINE_BIT
- GL_TRIANGLES
- GL_TRIANGLE_STRIP
- GL_TRIANGLE_FAN
- GL_QUADS
- GL_QUAD_STRIP

- GL_POLYGON_BIT
- GL_POLYGON
- GL_POLYGON_STIPPLE_BIT
- GL_PIXEL_MODE_BIT
- GL_LIGHTING_BIT
- GL_FOG_BIT
- GL_DEPTH_BUFFER_BIT
- GL_ACCUM
- GL_LOAD
- GL_RETURN
- GL_MULT
- GL_ADD
- GL_NEVER
- GL_ACCUM_BUFFER_BIT
- GL_LESS
- GL_EQUAL
- GL_LEQUAL
- GL_GREATER
- GL_NOTEQUAL
- GL_GEQUAL
- GL_ALWAYS
- GL_SRC_COLOR
- GL_ONE_MINUS_SRC_COLOR
- GL_SRC_ALPHA
- GL_ONE_MINUS_SRC_ALPHA
- GL_DST_ALPHA
- GL_ONE_MINUS_DST_ALPHA
- GL_DST_COLOR
- GL_ONE_MINUS_DST_COLOR
- GL_SRC_ALPHA_SATURATE
- GL_STENCIL_BUFFER_BIT
- GL_FRONT_LEFT
- GL_FRONT_RIGHT
- GL_BACK_LEFT
- GL_BACK_RIGHT
- GL_FRONT

- GL_BACK
- GL_LEFT
- GL_RIGHT
- GL_FRONT_AND_BACK
- GL_AUX0
- GL_AUX1
- GL_AUX2
- GL_AUX3
- GL_INVALID_ENUM
- GL_INVALID_VALUE
- GL_INVALID_OPERATION
- GL_STACK_OVERFLOW
- GL_STACK_UNDERFLOW
- GL_OUT_OF_MEMORY
- GL_2D
- GL_3D
- GL_3D_COLOR
- GL_3D_COLOR_TEXTURE
- GL_4D_COLOR_TEXTURE
- GL_PASS_THROUGH_TOKEN
- GL_POINT_TOKEN
- GL_LINE_TOKEN
- GL_POLYGON_TOKEN
- GL_BITMAP_TOKEN
- GL_DRAW_PIXEL_TOKEN
- GL_COPY_PIXEL_TOKEN
- GL_LINE_RESET_TOKEN
- GL_EXP
- GL_VIEWPORT_BIT
- GL_EXP2
- GL_CW
- GL_CCW
- GL_COEFF
- GL_ORDER
- GL_DOMAIN
- GL_CURRENT_COLOR

- GL_CURRENT_INDEX
- GL_CURRENT_NORMAL
- GL_CURRENT_TEXTURE_COORDS
- GL_CURRENT_RASTER_COLOR
- GL_CURRENT_RASTER_INDEX
- GL_CURRENT_RASTER_TEXTURE_COORDS
- GL_CURRENT_RASTER_POSITION
- GL_CURRENT_RASTER_POSITION_VALID
- GL_CURRENT_RASTER_DISTANCE
- GL_POINT_SMOOTH
- GL_POINT_SIZE
- GL_POINT_SIZE_RANGE
- GL_POINT_SIZE_GRANULARITY
- GL_LINE_SMOOTH
- GL_LINE_WIDTH
- GL_LINE_WIDTH_RANGE
- GL_LINE_WIDTH_GRANULARITY
- GL_LINE_STIPPLE
- GL_LINE_STIPPLE_PATTERN
- GL_LINE_STIPPLE_REPEAT
- GL_LIST_MODE
- GL_MAX_LIST_NESTING
- GL_LIST_BASE
- GL_LIST_INDEX
- GL_POLYGON_MODE
- GL_POLYGON_SMOOTH
- GL_POLYGON_STIPPLE
- GL_EDGE_FLAG
- GL_CULL_FACE
- GL_CULL_FACE_MODE
- GL_FRONT_FACE
- GL_LIGHTING
- GL_LIGHT_MODEL_LOCAL_VIEWER
- GL_LIGHT_MODEL_TWO_SIDE
- GL_LIGHT_MODEL_AMBIENT
- GL_SHADE_MODEL

- GL_COLOR_MATERIAL_FACE
- GL_COLOR_MATERIAL_PARAMETER
- GL_COLOR_MATERIAL
- GL_FOG
- GL_FOG_INDEX
- GL_FOG_DENSITY
- GL_FOG_START
- GL_FOG_END
- GL_FOG_MODE
- GL_FOG_COLOR
- GL_DEPTH_RANGE
- GL_DEPTH_TEST
- GL_DEPTH_WRITEMASK
- GL_DEPTH_CLEAR_VALUE
- GL_DEPTH_FUNC
- GL_ACCUM_CLEAR_VALUE
- GL_STENCIL_TEST
- GL_STENCIL_CLEAR_VALUE
- GL_STENCIL_FUNC
- GL_STENCIL_VALUE_MASK
- GL_STENCIL_FAIL
- GL_STENCIL_PASS_DEPTH_FAIL
- GL_STENCIL_PASS_DEPTH_PASS
- GL_STENCIL_REF
- GL_STENCIL_WRITEMASK
- GL_MATRIX_MODE
- GL_NORMALIZE
- GL_VIEWPORT
- GL_MODELVIEW_STACK_DEPTH
- GL_PROJECTION_STACK_DEPTH
- GL_TEXTURE_STACK_DEPTH
- GL_MODELVIEW_MATRIX
- GL_PROJECTION_MATRIX
- GL_TEXTURE_MATRIX
- GL_ATTRIB_STACK_DEPTH
- GL_CLIENT_ATTRIB_STACK_DEPTH

- GL_ALPHA_TEST
- GL_ALPHA_TEST_FUNC
- GL_ALPHA_TEST_REF
- GL_DITHER
- GL_BLEND_DST
- GL_BLEND_SRC
- GL_BLEND
- GL_LOGIC_OP_MODE
- GL_INDEX_LOGIC_OP
- GL_COLOR_LOGIC_OP
- GL_AUX_BUFFERS
- GL_DRAW_BUFFER
- GL_READ_BUFFER
- GL_SCISSOR_BOX
- GL_SCISSOR_TEST
- GL_INDEX_CLEAR_VALUE
- GL_INDEX_WRITEMASK
- GL_COLOR_CLEAR_VALUE
- GL_COLOR_WRITEMASK
- GL_INDEX_MODE
- GL_RGBA_MODE
- GL_DOUBLEBUFFER
- GL_STEREO
- GL_RENDER_MODE
- GL_PERSPECTIVE_CORRECTION_HINT
- GL_POINT_SMOOTH_HINT
- GL_LINE_SMOOTH_HINT
- GL_POLYGON_SMOOTH_HINT
- GL_FOG_HINT
- GL_TEXTURE_GEN_S
- GL_TEXTURE_GEN_T
- GL_TEXTURE_GEN_R
- GL_TEXTURE_GEN_Q
- GL_PIXEL_MAP_I_TO_I
- GL_PIXEL_MAP_S_TO_S
- GL_PIXEL_MAP_I_TO_R

- GL_PIXEL_MAP_I_TO_G
- GL_PIXEL_MAP_I_TO_B
- GL_PIXEL_MAP_I_TO_A
- GL_PIXEL_MAP_R_TO_R
- GL_PIXEL_MAP_G_TO_G
- GL_PIXEL_MAP_B_TO_B
- GL_PIXEL_MAP_A_TO_A
- GL_PIXEL_MAP_I_TO_I_SIZE
- GL_PIXEL_MAP_S_TO_S_SIZE
- GL_PIXEL_MAP_I_TO_R_SIZE
- GL_PIXEL_MAP_I_TO_G_SIZE
- GL_PIXEL_MAP_I_TO_B_SIZE
- GL_PIXEL_MAP_I_TO_A_SIZE
- GL_PIXEL_MAP_R_TO_R_SIZE
- GL_PIXEL_MAP_G_TO_G_SIZE
- GL_PIXEL_MAP_B_TO_B_SIZE
- GL_PIXEL_MAP_A_TO_A_SIZE
- GL_UNPACK_SWAP_BYTES
- GL_UNPACK_LSB_FIRST
- GL_UNPACK_ROW_LENGTH
- GL_UNPACK_SKIP_ROWS
- GL_UNPACK_SKIP_PIXELS
- GL_UNPACK_ALIGNMENT
- GL_PACK_SWAP_BYTES
- GL_PACK_LSB_FIRST
- GL_PACK_ROW_LENGTH
- GL_PACK_SKIP_ROWS
- GL_PACK_SKIP_PIXELS
- GL_PACK_ALIGNMENT
- GL_MAP_COLOR
- GL_MAP_STENCIL
- GL_INDEX_SHIFT
- GL_INDEX_OFFSET
- GL_RED_SCALE
- GL_RED_BIAS
- GL_ZOOM_X

- GL_ZOOM_Y
- GL_GREEN_SCALE
- GL_GREEN_BIAS
- GL_BLUE_SCALE
- GL_BLUE_BIAS
- GL_ALPHA_SCALE
- GL_ALPHA_BIAS
- GL_DEPTH_SCALE
- GL_DEPTH_BIAS
- GL_MAX_EVAL_ORDER
- GL_MAX_LIGHTS
- GL_MAX_CLIP_PLANES
- GL_MAX_TEXTURE_SIZE
- GL_MAX_PIXEL_MAP_TABLE
- GL_MAX_ATTRIB_STACK_DEPTH
- GL_MAX_MODELVIEW_STACK_DEPTH
- GL_MAX_NAME_STACK_DEPTH
- GL_MAX_PROJECTION_STACK_DEPTH
- GL_MAX_TEXTURE_STACK_DEPTH
- GL_MAX_VIEWPORT_DIMS
- GL_MAX_CLIENT_ATTRIB_STACK_DEPTH
- GL_SUBPIXEL_BITS
- GL_INDEX_BITS
- GL_RED_BITS
- GL_GREEN_BITS
- GL_BLUE_BITS
- GL_ALPHA_BITS
- GL_DEPTH_BITS
- GL_STENCIL_BITS
- GL_ACCUM_RED_BITS
- GL_ACCUM_GREEN_BITS
- GL_ACCUM_BLUE_BITS
- GL_ACCUM_ALPHA_BITS
- GL_NAME_STACK_DEPTH
- GL_AUTO_NORMAL
- GL_MAP1_COLOR_4

- GL_MAP1_INDEX
- GL_MAP1_NORMAL
- GL_MAP1_TEXTURE_COORD_1
- GL_MAP1_TEXTURE_COORD_2
- GL_MAP1_TEXTURE_COORD_3
- GL_MAP1_TEXTURE_COORD_4
- GL_MAP1_VERTEX_3
- GL_MAP1_VERTEX_4
- GL_MAP2_COLOR_4
- GL_MAP2_INDEX
- GL_MAP2_NORMAL
- GL_MAP2_TEXTURE_COORD_1
- GL_MAP2_TEXTURE_COORD_2
- GL_MAP2_TEXTURE_COORD_3
- GL_MAP2_TEXTURE_COORD_4
- GL_MAP2_VERTEX_3
- GL_MAP2_VERTEX_4
- GL_MAP1_GRID_DOMAIN
- GL_MAP1_GRID_SEGMENTS
- GL_MAP2_GRID_DOMAIN
- GL_MAP2_GRID_SEGMENTS
- GL_TEXTURE_1D
- GL_TEXTURE_2D
- GL_FEEDBACK_BUFFER_POINTER
- GL_FEEDBACK_BUFFER_SIZE
- GL_FEEDBACK_BUFFER_TYPE
- GL_SELECTION_BUFFER_POINTER
- GL_SELECTION_BUFFER_SIZE
- GL_TEXTURE_WIDTH
- GL_TRANSFORM_BIT
- GL_TEXTURE_HEIGHT
- GL_TEXTURE_INTERNAL_FORMAT
- GL_TEXTURE_BORDER_COLOR
- GL_TEXTURE_BORDER
- GL_DONT_CARE
- GL_FASTEST

- GL_NICEST
- GL_AMBIENT
- GL_DIFFUSE
- GL_SPECULAR
- GL_POSITION
- GL_SPOT_DIRECTION
- GL_SPOT_EXPONENT
- GL_SPOT_CUTOFF
- GL_CONSTANT_ATTENUATION
- GL_LINEAR_ATTENUATION
- GL_QUADRATIC_ATTENUATION
- GL_COMPILE
- GL_COMPILE_AND_EXECUTE
- GL_BYTE
- GL_UNSIGNED_BYTE
- GL_SHORT
- GL_UNSIGNED_SHORT
- GL_INT
- GL_UNSIGNED_INT
- GL_FLOAT
- GL_2_BYTES
- GL_3_BYTES
- GL_4_BYTES
- GL_DOUBLE
- GL_CLEAR
- GL_AND
- GL_AND_REVERSE
- GL_COPY
- GL_AND_INVERTED
- GL_NOOP
- GL_XOR
- GL_OR
- GL_NOR
- GL_EQUIV
- GL_INVERT
- GL_OR_REVERSE

- GL_COPY_INVERTED
- GL_OR_INVERTED
- GL_NAND
- GL_SET
- GL_EMISSION
- GL_SHININESS
- GL_AMBIENT_AND_DIFFUSE
- GL_COLOR_INDEXES
- GL_MODELVIEW
- GL_PROJECTION
- GL_TEXTURE
- GL_COLOR
- GL_DEPTH
- GL_STENCIL
- GL_COLOR_INDEX
- GL_STENCIL_INDEX
- GL_DEPTH_COMPONENT
- GL_RED
- GL_GREEN
- GL_BLUE
- GL_ALPHA
- GL_RGB
- GL_RGBA
- GL_LUMINANCE
- GL_LUMINANCE_ALPHA
- GL_BITMAP
- GL_POINT
- GL_LINE
- GL_FILL
- GL_RENDER
- GL_FEEDBACK
- GL_SELECT
- GL_FLAT
- GL_SMOOTH
- GL_KEEP
- GL_REPLACE

- GL_INCR
- GL_DECR
- GL_VENDOR
- GL_RENDERER
- GL_VERSION
- GL_EXTENSIONS
- GL_S
- GL_ENABLE_BIT
- GL_T
- GL_R
- GL_Q
- GL_MODULATE
- GL_DECAL
- GL_TEXTURE_ENV_MODE
- GL_TEXTURE_ENV_COLOR
- GL_TEXTURE_ENV
- GL_EYE_LINEAR
- GL_OBJECT_LINEAR
- GL_SPHERE_MAP
- GL_TEXTURE_GEN_MODE
- GL_OBJECT_PLANE
- GL_EYE_PLANE
- GL_NEAREST
- GL_LINEAR
- GL_NEAREST_MIPMAP_NEAREST
- GL_LINEAR_MIPMAP_NEAREST
- GL_NEAREST_MIPMAP_LINEAR
- GL_LINEAR_MIPMAP_LINEAR
- GL_TEXTURE_MAG_FILTER
- GL_TEXTURE_MIN_FILTER
- GL_TEXTURE_WRAP_S
- GL_TEXTURE_WRAP_T
- GL_CLAMP
- GL_REPEAT
- GL_POLYGON_OFFSET_UNITS
- GL_POLYGON_OFFSET_POINT

- GL_POLYGON_OFFSET_LINE
- GL_R3_G3_B2
- GL_V2F
- GL_V3F
- GL_C4UB_V2F
- GL_C4UB_V3F
- GL_C3F_V3F
- GL_N3F_V3F
- GL_C4F_N3F_V3F
- GL_T2F_V3F
- GL_T4F_V4F
- GL_T2F_C4UB_V3F
- GL_T2F_C3F_V3F
- GL_T2F_N3F_V3F
- GL_T2F_C4F_N3F_V3F
- GL_T4F_C4F_N3F_V4F
- GL_CLIP_PLANE0
- GL_CLIP_PLANE1
- GL_CLIP_PLANE2
- GL_CLIP_PLANE3
- GL_CLIP_PLANE4
- GL_CLIP_PLANE5
- GL_LIGHT0
- GL_COLOR_BUFFER_BIT
- GL_LIGHT1
- GL_LIGHT2
- GL_LIGHT3
- GL_LIGHT4
- GL_LIGHT5
- GL_LIGHT6
- GL_LIGHT7
- GL_HINT_BIT
- GL_POLYGON_OFFSET_FILL
- GL_POLYGON_OFFSET_FACTOR
- GL_ALPHA4
- GL_ALPHA8

- GL_ALPHA12
- GL_ALPHA16
- GL_LUMINANCE4
- GL_LUMINANCE8
- GL_LUMINANCE12
- GL_LUMINANCE16
- GL_LUMINANCE4_ALPHA4
- GL_LUMINANCE6_ALPHA2
- GL_LUMINANCE8_ALPHA8
- GL_LUMINANCE12_ALPHA4
- GL_LUMINANCE12_ALPHA12
- GL_LUMINANCE16_ALPHA16
- GL_INTENSITY
- GL_INTENSITY4
- GL_INTENSITY8
- GL_INTENSITY12
- GL_INTENSITY16
- GL_RGB4
- GL_RGB5
- GL_RGB8
- GL_RGB10
- GL_RGB12
- GL_RGB16
- GL_RGBA2
- GL_RGBA4
- GL_RGB5_A1
- GL_RGBA8
- GL_RGB10_A2
- GL_RGBA12
- GL_RGBA16
- GL_TEXTURE_RED_SIZE
- GL_TEXTURE_GREEN_SIZE
- GL_TEXTURE_BLUE_SIZE
- GL_TEXTURE_ALPHA_SIZE
- GL_TEXTURE_LUMINANCE_SIZE
- GL_TEXTURE_INTENSITY_SIZE

- GL_PROXY_TEXTURE_1D
- GL_PROXY_TEXTURE_2D
- GL_TEXTURE_PRIORITY
- GL_TEXTURE_RESIDENT
- GL_TEXTURE_BINDING_1D
- GL_TEXTURE_BINDING_2D
- GL_VERTEX_ARRAY
- GL_NORMAL_ARRAY
- GL_COLOR_ARRAY
- GL_INDEX_ARRAY
- GL_TEXTURE_COORD_ARRAY
- GL_EDGE_FLAG_ARRAY
- GL_VERTEX_ARRAY_SIZE
- GL_VERTEX_ARRAY_TYPE
- GL_VERTEX_ARRAY_STRIDE
- GL_NORMAL_ARRAY_TYPE
- GL_NORMAL_ARRAY_STRIDE
- GL_COLOR_ARRAY_SIZE
- GL_COLOR_ARRAY_TYPE
- GL_COLOR_ARRAY_STRIDE
- GL_INDEX_ARRAY_TYPE
- GL_INDEX_ARRAY_STRIDE
- GL_TEXTURE_COORD_ARRAY_SIZE
- GL_TEXTURE_COORD_ARRAY_TYPE
- GL_TEXTURE_COORD_ARRAY_STRIDE
- GL_EDGE_FLAG_ARRAY_STRIDE
- GL_VERTEX_ARRAY_POINTER
- GL_NORMAL_ARRAY_POINTER
- GL_COLOR_ARRAY_POINTER
- GL_INDEX_ARRAY_POINTER
- GL_TEXTURE_COORD_ARRAY_POINTER
- GL_EDGE_FLAG_ARRAY_POINTER
- GL_COLOR_INDEX1_EXT
- GL_COLOR_INDEX2_EXT
- GL_COLOR_INDEX4_EXT
- GL_COLOR_INDEX8_EXT

- GL_COLOR_INDEX12_EXT
- GL_COLOR_INDEX16_EXT
- GL_EVAL_BIT
- GL_LIST_BIT
- GL_TEXTURE_BIT
- GL_SCISSOR_BIT
- GL_ALL_ATTRIB_BITS
- GL_CLIENT_ALL_ATTRIB_BITS
- GL_SMOOTH_POINT_SIZE_RANGE
- GL_SMOOTH_POINT_SIZE_GRANULARITY
- GL_SMOOTH_LINE_WIDTH_RANGE
- GL_SMOOTH_LINE_WIDTH_GRANULARITY
- GL_UNSIGNED_BYTE_3_3_2
- GL_UNSIGNED_SHORT_4_4_4_4
- GL_UNSIGNED_SHORT_5_5_5_1
- GL_UNSIGNED_INT_8_8_8_8
- GL_UNSIGNED_INT_10_10_10_2
- GL_RESCALE_NORMAL
- GL_TEXTURE_BINDING_3D
- GL_PACK_SKIP_IMAGES
- GL_PACK_IMAGE_HEIGHT
- GL_UNPACK_SKIP_IMAGES
- GL_UNPACK_IMAGE_HEIGHT
- GL_TEXTURE_3D
- GL_PROXY_TEXTURE_3D
- GL_TEXTURE_DEPTH
- GL_TEXTURE_WRAP_R
- GL_MAX_3D_TEXTURE_SIZE
- GL_BGR
- GL_BGRA
- GL_MAX_ELEMENTS_VERTICES
- GL_MAX_ELEMENTS_INDICES
- GL_CLAMP_TO_EDGE
- GL_TEXTURE_MIN_LOD
- GL_TEXTURE_MAX_LOD
- GL_TEXTURE_BASE_LEVEL

- GL_TEXTURE_MAX_LEVEL
- GL_LIGHT_MODEL_COLOR_CONTROL
- GL_SINGLE_COLOR
- GL_SEPARATE_SPECULAR_COLOR
- GL_UNSIGNED_BYTE_2_3_3_REV
- GL_UNSIGNED_SHORT_5_6_5
- GL_UNSIGNED_SHORT_5_6_5_REV
- GL_UNSIGNED_SHORT_4_4_4_4_REV
- GL_UNSIGNED_SHORT_1_5_5_5_REV
- GL_UNSIGNED_INT_8_8_8_8_REV
- GL_ALIASED_POINT_SIZE_RANGE
- GL_ALIASED_LINE_WIDTH_RANGE
- GL_MULTISAMPLE
- GL_SAMPLE_ALPHA_TO_COVERAGE
- GL_SAMPLE_ALPHA_TO_ONE
- GL_SAMPLE_COVERAGE
- GL_SAMPLE_BUFFERS
- GL_SAMPLES
- GL_SAMPLE_COVERAGE_VALUE
- GL_SAMPLE_COVERAGE_INVERT
- GL_CLAMP_TO_BORDER
- GL_TEXTURE0
- GL_TEXTURE1
- GL_TEXTURE2
- GL_TEXTURE3
- GL_TEXTURE4
- GL_TEXTURE5
- GL_TEXTURE6
- GL_TEXTURE7
- GL_TEXTURE8
- GL_TEXTURE9
- GL_TEXTURE10
- GL_TEXTURE11
- GL_TEXTURE12
- GL_TEXTURE13
- GL_TEXTURE14

- GL_TEXTURE15
- GL_TEXTURE16
- GL_TEXTURE17
- GL_TEXTURE18
- GL_TEXTURE19
- GL_TEXTURE20
- GL_TEXTURE21
- GL_TEXTURE22
- GL_TEXTURE23
- GL_TEXTURE24
- GL_TEXTURE25
- GL_TEXTURE26
- GL_TEXTURE27
- GL_TEXTURE28
- GL_TEXTURE29
- GL_TEXTURE30
- GL_TEXTURE31
- GL_ACTIVE_TEXTURE
- GL_CLIENT_ACTIVE_TEXTURE
- GL_MAX_TEXTURE_UNITS
- GL_TRANSPOSE_MODELVIEW_MATRIX
- GL_TRANSPOSE_PROJECTION_MATRIX
- GL_TRANSPOSE_TEXTURE_MATRIX
- GL_TRANSPOSE_COLOR_MATRIX
- GL_SUBTRACT
- GL_COMPRESSED_ALPHA
- GL_COMPRESSED_LUMINANCE
- GL_COMPRESSED_LUMINANCE_ALPHA
- GL_COMPRESSED_INTENSITY
- GL_COMPRESSED_RGB
- GL_COMPRESSED_RGBA
- GL_TEXTURE_COMPRESSION_HINT
- GL_NORMAL_MAP
- GL_REFLECTION_MAP
- GL_TEXTURE_CUBE_MAP
- GL_TEXTURE_BINDING_CUBE_MAP

- GL_TEXTURE_CUBE_MAP_POSITIVE_X
- GL_TEXTURE_CUBE_MAP_NEGATIVE_X
- GL_TEXTURE_CUBE_MAP_POSITIVE_Y
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Y
- GL_TEXTURE_CUBE_MAP_POSITIVE_Z
- GL_TEXTURE_CUBE_MAP_NEGATIVE_Z
- GL_PROXY_TEXTURE_CUBE_MAP
- GL_MAX_CUBE_MAP_TEXTURE_SIZE
- GL_COMBINE
- GL_COMBINE_RGB
- GL_COMBINE_ALPHA
- GL_RGB_SCALE
- GL_ADD_SIGNED
- GL_INTERPOLATE
- GL_CONSTANT
- GL_PRIMARY_COLOR
- GL_PREVIOUS
- GL_SOURCE0_RGB
- GL_SOURCE1_RGB
- GL_SOURCE2_RGB
- GL_SOURCE0_ALPHA
- GL_SOURCE1_ALPHA
- GL_SOURCE2_ALPHA
- GL_OPERAND0_RGB
- GL_OPERAND1_RGB
- GL_OPERAND2_RGB
- GL_OPERAND0_ALPHA
- GL_OPERAND1_ALPHA
- GL_OPERAND2_ALPHA
- GL_TEXTURE_COMPRESSED_IMAGE_SIZE
- GL_TEXTURE_COMPRESSED
- GL_NUM_COMPRESSED_TEXTURE_FORMATS
- GL_COMPRESSED_TEXTURE_FORMATS
- GL_DOT3_RGB
- GL_DOT3_RGBA
- GL_MULTISAMPLE_BIT

- GL_BLEND_DST_RGB
- GL_BLEND_SRC_RGB
- GL_BLEND_DST_ALPHA
- GL_BLEND_SRC_ALPHA
- GL_POINT_SIZE_MIN
- GL_POINT_SIZE_MAX
- GL_POINT_FADE_THRESHOLD_SIZE
- GL_POINT_DISTANCE_ATTENUATION
- GL_GENERATE_MIPMAP
- GL_GENERATE_MIPMAP_HINT
- GL_DEPTH_COMPONENT16
- GL_DEPTH_COMPONENT24
- GL_DEPTH_COMPONENT32
- GL_MIRRORED_REPEAT
- GL_FOG_COORDINATE_SOURCE
- GL_FOG_COORDINATE
- GL_FRAGMENT_DEPTH
- GL_CURRENT_FOG_COORDINATE
- GL_FOG_COORDINATE_ARRAY_TYPE
- GL_FOG_COORDINATE_ARRAY_STRIDE
- GL_FOG_COORDINATE_ARRAY_POINTER
- GL_FOG_COORDINATE_ARRAY
- GL_COLOR_SUM
- GL_CURRENT_SECONDARY_COLOR
- GL_SECONDARY_COLOR_ARRAY_SIZE
- GL_SECONDARY_COLOR_ARRAY_TYPE
- GL_SECONDARY_COLOR_ARRAY_STRIDE
- GL_SECONDARY_COLOR_ARRAY_POINTER
- GL_SECONDARY_COLOR_ARRAY
- GL_MAX_TEXTURE_LOD_BIAS
- GL_TEXTURE_FILTER_CONTROL
- GL_TEXTURE_LOD_BIAS
- GL_INCR_WRAP
- GL_DECR_WRAP
- GL_TEXTURE_DEPTH_SIZE
- GL_DEPTH_TEXTURE_MODE

- GL_TEXTURE_COMPARE_MODE
- GL_TEXTURE_COMPARE_FUNC
- GL_COMPARE_R_TO_TEXTURE
- GL_CURRENT_FOG_COORD
- GL_FOG_COORD
- GL_FOG_COORD_ARRAY
- GL_FOG_COORD_ARRAY_BUFFER_BINDING
- GL_FOG_COORD_ARRAY_POINTER
- GL_FOG_COORD_ARRAY_STRIDE
- GL_FOG_COORD_ARRAY_TYPE
- GL_FOG_COORD_SRC
- GL_SRC0_ALPHA
- GL_SRC0_RGB
- GL_SRC1_ALPHA
- GL_SRC1_RGB
- GL_SRC2_ALPHA
- GL_SRC2_RGB
- GL_BUFFER_SIZE
- GL_BUFFER_USAGE
- GL_QUERY_COUNTER_BITS
- GL_CURRENT_QUERY
- GL_QUERY_RESULT
- GL_QUERY_RESULT_AVAILABLE
- GL_ARRAY_BUFFER
- GL_ELEMENT_ARRAY_BUFFER
- GL_ARRAY_BUFFER_BINDING
- GL_ELEMENT_ARRAY_BUFFER_BINDING
- GL_VERTEX_ARRAY_BUFFER_BINDING
- GL_NORMAL_ARRAY_BUFFER_BINDING
- GL_COLOR_ARRAY_BUFFER_BINDING
- GL_INDEX_ARRAY_BUFFER_BINDING
- GL_TEXTURE_COORD_ARRAY_BUFFER_BINDING
- GL_EDGE_FLAG_ARRAY_BUFFER_BINDING
- GL_SECONDARY_COLOR_ARRAY_BUFFER_BINDING
- GL_FOG_COORDINATE_ARRAY_BUFFER_BINDING
- GL_WEIGHT_ARRAY_BUFFER_BINDING

- GL_VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
- GL_READ_ONLY
- GL_WRITE_ONLY
- GL_READ_WRITE
- GL_BUFFER_ACCESS
- GL_BUFFER_MAPPED
- GL_BUFFER_MAP_POINTER
- GL_STREAM_DRAW
- GL_STREAM_READ
- GL_STREAM_COPY
- GL_STATIC_DRAW
- GL_STATIC_READ
- GL_STATIC_COPY
- GL_DYNAMIC_DRAW
- GL_DYNAMIC_READ
- GL_DYNAMIC_COPY
- GL_SAMPLES_PASSED
- GL_BLEND_EQUATION_RGB
- GL_VERTEX_ATTRIB_ARRAY_ENABLED
- GL_VERTEX_ATTRIB_ARRAY_SIZE
- GL_VERTEX_ATTRIB_ARRAY_STRIDE
- GL_VERTEX_ATTRIB_ARRAY_TYPE
- GL_CURRENT_VERTEX_ATTRIB
- GL_VERTEX_PROGRAM_POINT_SIZE
- GL_VERTEX_PROGRAM_TWO_SIDE
- GL_VERTEX_ATTRIB_ARRAY_POINTER
- GL_STENCIL_BACK_FUNC
- GL_STENCIL_BACK_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_FAIL
- GL_STENCIL_BACK_PASS_DEPTH_PASS
- GL_MAX_DRAW_BUFFERS
- GL_DRAW_BUFFER0
- GL_DRAW_BUFFER1
- GL_DRAW_BUFFER2
- GL_DRAW_BUFFER3
- GL_DRAW_BUFFER4

- GL_DRAW_BUFFER5
- GL_DRAW_BUFFER6
- GL_DRAW_BUFFER7
- GL_DRAW_BUFFER8
- GL_DRAW_BUFFER9
- GL_DRAW_BUFFER10
- GL_DRAW_BUFFER11
- GL_DRAW_BUFFER12
- GL_DRAW_BUFFER13
- GL_DRAW_BUFFER14
- GL_DRAW_BUFFER15
- GL_BLEND_EQUATION_ALPHA
- GL_POINT_SPRITE
- GL_COORD_REPLACE
- GL_MAX_VERTEX_ATTRIBS
- GL_VERTEX_ATTRIB_ARRAY_NORMALIZED
- GL_MAX_TEXTURE_COORDS
- GL_MAX_TEXTURE_IMAGE_UNITS
- GL_FRAGMENT_SHADER
- GL_VERTEX_SHADER
- GL_MAX_FRAGMENT_UNIFORM_COMPONENTS
- GL_MAX_VERTEX_UNIFORM_COMPONENTS
- GL_MAX_VARYING_FLOATS
- GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS
- GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS
- GL_SHADER_TYPE
- GL_FLOAT_VEC2
- GL_FLOAT_VEC3
- GL_FLOAT_VEC4
- GL_INT_VEC2
- GL_INT_VEC3
- GL_INT_VEC4
- GL_BOOL
- GL_BOOL_VEC2
- GL_BOOL_VEC3
- GL_BOOL_VEC4

- GL_FLOAT_MAT2
- GL_FLOAT_MAT3
- GL_FLOAT_MAT4
- GL_SAMPLER_1D
- GL_SAMPLER_2D
- GL_SAMPLER_3D
- GL_SAMPLER_CUBE
- GL_SAMPLER_1D_SHADOW
- GL_SAMPLER_2D_SHADOW
- GL_DELETE_STATUS
- GL_COMPILE_STATUS
- GL_LINK_STATUS
- GL_VALIDATE_STATUS
- GL_INFO_LOG_LENGTH
- GL_ATTACHED_SHADERS
- GL_ACTIVE_UNIFORMS
- GL_ACTIVE_UNIFORM_MAX_LENGTH
- GL_SHADER_SOURCE_LENGTH
- GL_ACTIVE_ATTRIBUTES
- GL_ACTIVE_ATTRIBUTE_MAX_LENGTH
- GL_FRAGMENT_SHADER_DERIVATIVE_HINT
- GL_SHADING_LANGUAGE_VERSION
- GL_CURRENT_PROGRAM
- GL_POINT_SPRITE_COORD_ORIGIN
- GL_LOWER_LEFT
- GL_UPPER_LEFT
- GL_STENCIL_BACK_REF
- GL_STENCIL_BACK_VALUE_MASK
- GL_STENCIL_BACK_WRITEMASK
- GL_CURRENT_RASTER_SECONDARY_COLOR
- GL_PIXEL_PACK_BUFFER
- GL_PIXEL_UNPACK_BUFFER
- GL_PIXEL_PACK_BUFFER_BINDING
- GL_PIXEL_UNPACK_BUFFER_BINDING
- GL_FLOAT_MAT2x3
- GL_FLOAT_MAT2x4

- GL_FLOAT_MAT3x2
- GL_FLOAT_MAT3x4
- GL_FLOAT_MAT4x2
- GL_FLOAT_MAT4x3
- GL_SRGB
- GL_SRGB8
- GL_SRGB_ALPHA
- GL_SRGB8_ALPHA8
- GL_SLUMINANCE_ALPHA
- GL_SLUMINANCE8_ALPHA8
- GL_SLUMINANCE
- GL_SLUMINANCE8
- GL_COMPRESSED_SRGB
- GL_COMPRESSED_SRGB_ALPHA
- GL_COMPRESSED_SLUMINANCE
- GL_COMPRESSED_SLUMINANCE_ALPHA
- GL_CLIP_DISTANCE0
- GL_CLIP_DISTANCE1
- GL_CLIP_DISTANCE2
- GL_CLIP_DISTANCE3
- GL_CLIP_DISTANCE4
- GL_CLIP_DISTANCE5
- GL_COMPARE_REF_TO_TEXTURE
- GL_MAX_CLIP_DISTANCES
- GL_MAX_VARYING_COMPONENTS
- GL_CONTEXT_FLAG_FORWARD_COMPATIBLE_BIT
- GL_MAJOR_VERSION
- GL_MINOR_VERSION
- GL_NUM_EXTENSIONS
- GL_CONTEXT_FLAGS
- GL_DEPTH_BUFFER
- GL_STENCIL_BUFFER
- GL_RGBA32F
- GL_RGB32F
- GL_RGBA16F
- GL_RGB16F

- GL_VERTEX_ATTRIB_ARRAY_INTEGER
- GL_MAX_ARRAY_TEXTURE_LAYERS
- GL_MIN_PROGRAM_TEXEL_OFFSET
- GL_MAX_PROGRAM_TEXEL_OFFSET
- GL_CLAMP_VERTEX_COLOR
- GL_CLAMP_FRAGMENT_COLOR
- GL_CLAMP_READ_COLOR
- GL_FIXED_ONLY
- GL_TEXTURE_RED_TYPE
- GL_TEXTURE_GREEN_TYPE
- GL_TEXTURE_BLUE_TYPE
- GL_TEXTURE_ALPHA_TYPE
- GL_TEXTURE_LUMINANCE_TYPE
- GL_TEXTURE_INTENSITY_TYPE
- GL_TEXTURE_DEPTH_TYPE
- GL_TEXTURE_1D_ARRAY
- GL_PROXY_TEXTURE_1D_ARRAY
- GL_TEXTURE_2D_ARRAY
- GL_PROXY_TEXTURE_2D_ARRAY
- GL_TEXTURE_BINDING_1D_ARRAY
- GL_TEXTURE_BINDING_2D_ARRAY
- GL_R11F_G11F_B10F
- GL_UNSIGNED_INT_10F_11F_11F_REV
- GL_RGB9_E5
- GL_UNSIGNED_INT_5_9_9_9_REV
- GL_TEXTURE_SHARED_SIZE
- GL_TRANSFORM_FEEDBACK_VARYING_MAX_LENGTH
- GL_TRANSFORM_FEEDBACK_BUFFER_MODE
- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_COMPONENTS
- GL_TRANSFORM_FEEDBACK_VARYINGS
- GL_TRANSFORM_FEEDBACK_BUFFER_START
- GL_TRANSFORM_FEEDBACK_BUFFER_SIZE
- GL_PRIMITIVES_GENERATED
- GL_TRANSFORM_FEEDBACK_PRIMITIVES_WRITTEN
- GL_RASTERIZER_DISCARD
- GL_MAX_TRANSFORM_FEEDBACK_INTERLEAVED_COMPONENTS

- GL_MAX_TRANSFORM_FEEDBACK_SEPARATE_ATTRIBS
- GL_INTERLEAVED_ATTRIBS
- GL_SEPARATE_ATTRIBS
- GL_TRANSFORM_FEEDBACK_BUFFER
- GL_TRANSFORM_FEEDBACK_BUFFER_BINDING
- GL_RGBA32UI
- GL_RGB32UI
- GL_RGBA16UI
- GL_RGB16UI
- GL_RGBA8UI
- GL_RGB8UI
- GL_RGBA32I
- GL_RGB32I
- GL_RGBA16I
- GL_RGB16I
- GL_RGBA8I
- GL_RGB8I
- GL_RED_INTEGER
- GL_GREEN_INTEGER
- GL_BLUE_INTEGER
- GL_ALPHA_INTEGER
- GL_RGB_INTEGER
- GL_RGBA_INTEGER
- GL_BGR_INTEGER
- GL_BGRA_INTEGER
- GL_SAMPLER_1D_ARRAY
- GL_SAMPLER_2D_ARRAY
- GL_SAMPLER_1D_ARRAY_SHADOW
- GL_SAMPLER_2D_ARRAY_SHADOW
- GL_SAMPLER_CUBE_SHADOW
- GL_UNSIGNED_INT_VEC2
- GL_UNSIGNED_INT_VEC3
- GL_UNSIGNED_INT_VEC4
- GL_INT_SAMPLER_1D
- GL_INT_SAMPLER_2D
- GL_INT_SAMPLER_3D

- GL_INT_SAMPLER_CUBE
- GL_INT_SAMPLER_1D_ARRAY
- GL_INT_SAMPLER_2D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_1D
- GL_UNSIGNED_INT_SAMPLER_2D
- GL_UNSIGNED_INT_SAMPLER_3D
- GL_UNSIGNED_INT_SAMPLER_CUBE
- GL_UNSIGNED_INT_SAMPLER_1D_ARRAY
- GL_UNSIGNED_INT_SAMPLER_2D_ARRAY
- GL_QUERY_WAIT
- GL_QUERY_NO_WAIT
- GL_QUERY_BY_REGION_WAIT
- GL_QUERY_BY_REGION_NO_WAIT
- GL_TEXTURE_RECTANGLE
- GL_TEXTURE_BINDING_RECTANGLE
- GL_PROXY_TEXTURE_RECTANGLE
- GL_MAX_RECTANGLE_TEXTURE_SIZE
- GL_SAMPLER_2D_RECT
- GL_SAMPLER_2D_RECT_SHADOW
- GL_TEXTURE_BUFFER
- GL_MAX_TEXTURE_BUFFER_SIZE
- GL_TEXTURE_BINDING_BUFFER
- GL_TEXTURE_BUFFER_DATA_STORE_BINDING
- GL_TEXTURE_BUFFER_FORMAT
- GL_SAMPLER_BUFFER
- GL_INT_SAMPLER_2D_RECT
- GL_INT_SAMPLER_BUFFER
- GL_UNSIGNED_INT_SAMPLER_2D_RECT
- GL_UNSIGNED_INT_SAMPLER_BUFFER
- GL_RED_SNORM
- GL_RG_SNORM
- GL_RGB_SNORM
- GL_RGBA_SNORM
- GL_R8_SNORM
- GL_RG8_SNORM
- GL_RGB8_SNORM

- GL_RGBA8_SNORM
- GL_R16_SNORM
- GL_RG16_SNORM
- GL_RGB16_SNORM
- GL_RGBA16_SNORM
- GL_SIGNED_NORMALIZED
- GL_PRIMITIVE_RESTART
- GL_PRIMITIVE_RESTART_INDEX
- GL_BUFFER_ACCESS_FLAGS
- GL_BUFFER_MAP_LENGTH
- GL_BUFFER_MAP_OFFSET
- GL_CONTEXT_CORE_PROFILE_BIT
- GL_CONTEXT_COMPATIBILITY_PROFILE_BIT
- GL_LINES_ADJACENCY
- GL_LINE_STRIP_ADJACENCY
- GL_TRIANGLES_ADJACENCY
- GL_TRIANGLE_STRIP_ADJACENCY
- GL_PROGRAM_POINT_SIZE
- GL_GEOMETRY_VERTICES_OUT
- GL_GEOMETRY_INPUT_TYPE
- GL_GEOMETRY_OUTPUT_TYPE
- GL_MAX_GEOMETRY_TEXTURE_IMAGE_UNITS
- GL_FRAMEBUFFER_ATTACHMENT_LAYERED
- GL_FRAMEBUFFER_INCOMPLETE_LAYER_TARGETS
- GL_GEOMETRY_SHADER
- GL_MAX_GEOMETRY_UNIFORM_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_VERTICES
- GL_MAX_GEOMETRY_TOTAL_OUTPUT_COMPONENTS
- GL_MAX_VERTEX_OUTPUT_COMPONENTS
- GL_MAX_GEOMETRY_INPUT_COMPONENTS
- GL_MAX_GEOMETRY_OUTPUT_COMPONENTS
- GL_MAX_FRAGMENT_INPUT_COMPONENTS
- GL_CONTEXT_PROFILE_MASK
- void glAccum(GLenum op,GLfloat value)
- void glActiveTexture(GLenum texture)
- void glAlphaFunc(GLenum func,GLclampf ref)

- GLboolean glAreTexturesResident(GLsizei n,const GLuint * textures,GLboolean * residences)
- void glArrayElement(GLint i)
- void glAttachShader(GLuint program,GLuint shader)
- void glBegin(GLenum mode)
- void glBeginQuery(GLenum target,GLuint id)
- void glBindAttribLocation(GLuint program,GLuint index,const GLchar *name)
- void glBindBuffer(GLenum target,GLuint buffer)
- void glBindTexture(GLenum target,GLuint texture)
- void glBitmap(GLsizei width,GLsizei height,GLfloat xorig,GLfloat yorig,GLfloat xmove,GLfloat ymove,const GLubyte * bitmap)
- void glBlendColor(GLclampf red,GLclampf green,GLclampf blue,GLclampf alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB,GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor,GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB,GLenum dstRGB,GLenum srcAlpha,GLenum dstAlpha)
- void glBufferData(GLenum target,GLsizeiptr size,const GLvoid * data,GLenum usage)
- void glBufferSubData(GLenum target,GLintptr offset,GLsizeiptr size,const GLvoid * data)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n,GLenum type,const GLvoid * lists)
- void glClear(GLbitfield mask)
- void glClearAccum(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)
- void glClearColor(GLclampf red,GLclampf green,GLclampf blue,GLclampf alpha)
- void glClearDepth(GLclampd depth)
- void glClearIndex(GLfloat c)
- void glClearStencil(GLint s)
- void glClientActiveTexture(GLenum texture)
- void glClipPlane(GLenum plane,const GLdouble * equation)
- void glColor3b(GLbyte red,GLbyte green,GLbyte blue)
- void glColor3s(GLshort red,GLshort green,GLshort blue)
- void glColor3i(GLint red,GLint green,GLint blue)
- void glColor3f(GLfloat red,GLfloat green,GLfloat blue)
- void glColor3d(GLdouble red,GLdouble green,GLdouble blue)
- void glColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glColor3us(GLushort red,GLushort green,GLushort blue)
- void glColor3ui(GLuint red,GLuint green,GLuint blue)
- void glColor4b(GLbyte red,GLbyte green,GLbyte blue,GLbyte alpha)

- void glColor4s(GLshort red,GLshort green,GLshort blue,GLshort alpha)
- void glColor4i(GLint red,GLint green,GLint blue,GLint alpha)
- void glColor4f(GLfloat red,GLfloat green,GLfloat blue,GLfloat alpha)
- void glColor4d(GLdouble red,GLdouble green,GLdouble blue,GLdouble alpha)
- void glColor4ub(GLubyte red,GLubyte green,GLubyte blue,GLubyte alpha)
- void glColor4us(GLushort red,GLushort green,GLushort blue,GLushort alpha)
- void glColor4ui(GLuint red,GLuint green,GLuint blue,GLuint alpha)
- void glColor3bv(const GLbyte * v)
- void glColor3sv(const GLshort * v)
- void glColor3iv(const GLint * v)
- void glColor3fv(const GLfloat * v)
- void glColor3dv(const GLdouble * v)
- void glColor3ubv(const GLubyte * v)
- void glColor3usv(const GLushort * v)
- void glColor3uiv(const GLuint * v)
- void glColor4bv(const GLbyte * v)
- void glColor4sv(const GLshort * v)
- void glColor4iv(const GLint * v)
- void glColor4fv(const GLfloat * v)
- void glColor4dv(const GLdouble * v)
- void glColor4ubv(const GLubyte * v)
- void glColor4usv(const GLushort * v)
- void glColor4uiv(const GLuint * v)
- void glColorMask(GLboolean red,GLboolean green,GLboolean blue,GLboolean alpha)
- void glColorMaterial(GLenum face,GLenum mode)
- void glColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glColorSubTable(GLenum target,GLsizei start,GLsizei count,GLenum format,GLenum type,const GLvoid * data)
- void glColorTable(GLenum target,GLenum internalformat,GLsizei width,GLenum format,GLenum type,const GLvoid * data)
- void glColorTableParameterfv(GLenum target,GLenum pname,const GLfloat * params)
- void glColorTableParameteriv(GLenum target,GLenum pname,const GLint * params)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target,GLint level,GLenum internalformat,GLsizei width,GLint border,GLsizei imageSize,const GLvoid * data)
- void glCompressedTexImage2D(GLenum target,GLint level,GLenum internalformat,GLsizei width,GLsizei height,GLint border,GLsizei imageSize,const GLvoid * data)

- void glCompressedTexImage3D(GLenum target,GLint level,GLenum internalformat,GLsizei width,GLsizei height,GLsizei depth,GLint border,GLsizei imageSize,const GLvoid * data)
- void glCompressedTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLsizei width,GLenum format,GLsizei imageSize,const GLvoid * data)
- void glCompressedTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLsizei width,GLsizei height,GLenum format,GLsizei imageSize,const GLvoid * data)
- void glCompressedTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLsizei width,GLsizei height,GLsizei depth,GLenum format,GLsizei imageSize,const GLvoid * data)
- void glConvolutionFilter1D(GLenum target,GLenum internalformat,GLsizei width,GLenum format,GLenum type,const GLvoid * data)
- void glConvolutionFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glConvolutionParameterf(GLenum target,GLenum pname,GLfloat params)
- void glConvolutionParameteri(GLenum target,GLenum pname,GLint params)
- void glConvolutionParameterfv(GLenum target,GLenum pname,const GLfloat * params)
- void glConvolutionParameteriv(GLenum target,GLenum pname,const GLint * params)
- void glCopyColorSubTable(GLenum target,GLsizei start,GLint x,GLint y,GLsizei width)
- void glCopyColorTable(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width)
- void glCopyConvolutionFilter1D(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width)
- void glCopyConvolutionFilter2D(GLenum target,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum type)
- void glCopyTexImage1D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLint border)
- void glCopyTexImage2D(GLenum target,GLint level,GLenum internalformat,GLint x,GLint y,GLsizei width,GLsizei height,GLint border)
- void glCopyTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLint x,GLint y,GLsizei width)
- void glCopyTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- void glCopyTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLint x,GLint y,GLsizei width,GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum shaderType)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n,const GLuint * buffers)
- void glDeleteLists(GLuint list,GLsizei range)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n,const GLuint * ids)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n,const GLuint * textures)

- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd nearVal,GLclampd farVal)
- void glDetachShader(GLuint program,GLuint shader)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode,GLint first,GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n,const GLenum *bufs)
- void glDrawElements(GLenum mode,GLsizei count,GLenum type,const GLvoid * indices)
- void glDrawPixels(GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glDrawRangeElements(GLenum mode,GLuint start,GLuint end,GLsizei count,GLenum type,const GLvoid * indices)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride,const GLvoid * pointer)
- void glEnd(void)
- void glEndList(void)
- void glEndQuery(GLenum target)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord2f(GLfloat u,GLfloat v)
- void glEvalCoord2d(GLdouble u,GLdouble v)
- void glEvalMesh1(GLenum mode,GLint i1,GLint i2)
- void glEvalPoint1(GLint i)
- void glEvalPoint2(GLint i,GLint j)
- void glFeedbackBuffer(GLsizei size,GLenum type,GLfloat * buffer)
- void glFinish(void)
- void glFlush(void)
- void glFogf(GLenum pname,GLfloat param)
- void glFogi(GLenum pname,GLint param)
- void glFogfv(GLenum pname,const GLfloat * params)
- void glFogiv(GLenum pname,const GLint * params)
- void glFogCoordd(GLdouble coord)
- void glFogCoordf(GLfloat coord)

- void glFogCoorddv(GLdouble * coord)
- void glFogCoordfv(GLfloat * coord)
- void glFogCoordPointer(GLenum type,GLsizei stride,GLvoid * pointer)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glGenBuffers(GLsizei n,GLuint * buffers)
- GLuint glGenLists(GLsizei range)
- void glGenQueries(GLsizei n,GLuint * ids)
- void glGenTextures(GLsizei n,GLuint * textures)
- void glGetBooleanv(GLenum pname,GLboolean * params)
- void glGetDoublev(GLenum pname,GLdouble * params)
- void glGetFloatv(GLenum pname,GLfloat * params)
- void glGetIntegerv(GLenum pname,GLint * params)
- void glGetActiveAttrib(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetActiveUniform(GLuint program,GLuint index,GLsizei bufSize,GLsizei *length,GLint *size,GLenum *type,GLchar *name)
- void glGetAttachedShaders(GLuint program,GLsizei maxCount,GLsizei *count,GLuint *shaders)
- GLint glGetAttribLocation(GLuint program,const GLchar *name)
- void glGetBufferParameteriv(GLenum target,GLenum value,GLint * data)
- void glGetBufferPointerv(GLenum target,GLenum pname,GLvoid ** params)
- void glGetBufferSubData(GLenum target,GLintptr offset,GLsizeiptr size,GLvoid * data)
- void glGetClipPlane(GLenum plane,GLdouble * equation)
- void glGetColorTable(GLenum target,GLenum format,GLenum type,GLvoid * table)
- void glGetColorTableParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetColorTableParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetCompressedTexImage(GLenum target,GLint lod,GLvoid * img)
- void glGetConvolutionFilter(GLenum target,GLenum format,GLenum type,GLvoid * image)
- void glGetConvolutionParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetConvolutionParameteriv(GLenum target,GLenum pname,GLint * params)
- GLenum glGetError(void)
- void glGetHistogram(GLenum target,GLboolean reset,GLenum format,GLenum type,GLvoid * values)
- void glGetHistogramParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetHistogramParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetLightfv(GLenum light,GLenum pname,GLfloat * params)
- void glGetLightiv(GLenum light,GLenum pname,GLint * params)

- void glGetMapdv(GLenum target,GLenum query,GLdouble * v)
- void glGetMapfv(GLenum target,GLenum query,GLfloat * v)
- void glGetMapiv(GLenum target,GLenum query,GLint * v)
- void glGetMaterialfv(GLenum face,GLenum pname,GLfloat * params)
- void glGetMaterialiv(GLenum face,GLenum pname,GLint * params)
- void glGetMinmax(GLenum target,GLboolean reset,GLenum format,GLenum types,GLvoid * values)
- void glGetMinmaxParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetMinmaxParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetPixelMapfv(GLenum map,GLfloat * data)
- void glGetPixelMapuiv(GLenum map,GLuint * data)
- void glGetPixelMapusv(GLenum map,GLushort * data)
- void glGetPointerv(GLenum pname,GLvoid ** params)
- void glGetPolygonStipple(GLubyte * pattern)
- void glGetProgramiv(GLuint program,GLenum pname,GLint *params)
- void glGetProgramInfoLog(GLuint program,GLsizei maxLength,GLsizei *length,GLchar *infoLog)
- void glGetQueryObjectiv(GLuint id,GLenum pname,GLint * params)
- void glGetQueryObjectuiv(GLuint id,GLenum pname,GLuint * params)
- void glGetQueryiv(GLenum target,GLenum pname,GLint * params)
- void glGetSeparableFilter(GLenum target,GLenum format,GLenum type,GLvoid * row,GLvoid * column,GLvoid * span)
- void glGetShaderiv(GLuint shader,GLenum pname,GLint *params)
- void glGetShaderInfoLog(GLuint shader,GLsizei maxLength,GLsizei *length,GLchar *infoLog)
- void glGetShaderSource(GLuint shader,GLsizei bufSize,GLsizei *length,GLchar *source)
- const GLubyte *glGetString(GLenum name)
- void glGetTexEnvfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetTexEnviv(GLenum target,GLenum pname,GLint * params)
- void glGetTexGendv(GLenum coord,GLenum pname,GLdouble * params)
- void glGetTexGenfv(GLenum coord,GLenum pname,GLfloat * params)
- void glGetTexGeniv(GLenum coord,GLenum pname,GLint * params)
- void glGetTexImage(GLenum target,GLint level,GLenum format,GLenum type,GLvoid * img)
- void glGetTexLevelParameterfv(GLenum target,GLint level,GLenum pname,GLfloat * params)
- void glGetTexLevelParameteriv(GLenum target,GLint level,GLenum pname,GLint * params)
- void glGetTexParameterfv(GLenum target,GLenum pname,GLfloat * params)
- void glGetTexParameteriv(GLenum target,GLenum pname,GLint * params)
- void glGetUniformfv(GLuint program,GLint location,GLfloat *params)
- void glGetUniformiv(GLuint program,GLint location,GLint *params)

- GLint glGetUniformLocation(GLuint program,const GLchar *name)
- void glGetVertexAttribdv(GLuint index,GLenum pname,GLdouble *params)
- void glGetVertexAttribfv(GLuint index,GLenum pname,GLfloat *params)
- void glGetVertexAttribiv(GLuint index,GLenum pname,GLint *params)
- void glGetVertexAttribPointerv(GLuint index,GLenum pname,GLvoid **pointer)
- void glHint(GLenum target,GLenum mode)
- void glHistogram(GLenum target,GLsizei width,GLenum internalformat,GLboolean sink)
- void glIndexs(GLshort c)
- void glIndexi(GLint c)
- void glIndexf(GLfloat c)
- void glIndexd(GLdouble c)
- void glIndexub(GLubyte c)
- void glIndexsv(const GLshort * c)
- void glIndexiv(const GLint * c)
- void glIndexfv(const GLfloat * c)
- void glIndexdv(const GLdouble * c)
- void glIndexubv(const GLubyte * c)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type,GLsizei stride,const GLvoid * pointer)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format,GLsizei stride,const GLvoid * pointer)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)
- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLightf(GLenum light,GLenum pname,GLfloat param)
- void glLighti(GLenum light,GLenum pname,GLint param)
- void glLightfv(GLenum light,GLenum pname,const GLfloat * params)
- void glLightiv(GLenum light,GLenum pname,const GLint * params)
- void glLightModelf(GLenum pname,GLfloat param)
- void glLightModeli(GLenum pname,GLint param)
- void glLightModelfv(GLenum pname,const GLfloat * params)
- void glLightModeliv(GLenum pname,const GLint * params)

- void glLineStipple(GLint factor,GLushort pattern)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble * m)
- void glLoadMatrixf(const GLfloat * m)
- void glLoadName(GLuint name)
- void glLoadTransposeMatrixd(const GLdouble * m)
- void glLoadTransposeMatrixf(const GLfloat * m)
- void glLogicOp(GLenum opcode)
- void glMap1f(GLenum target,GLfloat u1,GLfloat u2,GLint stride,GLint order,const GLfloat * points)
- void glMap1d(GLenum target,GLdouble u1,GLdouble u2,GLint stride,GLint order,const GLdouble * points)
- void glMap2f(GLenum target,GLfloat u1,GLfloat u2,GLint ustride,GLint uorder,GLfloat v1,GLfloat v2,GLint vstride,GLint vorder,const GLfloat * points)
- void glMap2d(GLenum target,GLdouble u1,GLdouble u2,GLint ustride,GLint uorder,GLdouble v1,GLdouble v2,GLint vstride,GLint vorder,const GLdouble * points)
- void * glMapBuffer(GLenum target,GLenum access)
- void glMapGrid1d(GLint un,GLdouble u1,GLdouble u2)
- void glMapGrid1f(GLint un,GLfloat u1,GLfloat u2)
- void glMapGrid2d(GLint un,GLdouble u1,GLdouble u2,GLint vn,GLdouble v1,GLdouble v2)
- void glMapGrid2f(GLint un,GLfloat u1,GLfloat u2,GLint vn,GLfloat v1,GLfloat v2)
- void glMaterialf(GLenum face,GLenum pname,GLfloat param)
- void glMateriali(GLenum face,GLenum pname,GLint param)
- void glMatrixMode(GLenum mode)
- void glMinmax(GLenum target,GLenum internalformat,GLboolean sink)
- void glMultMatrixd(const GLdouble * m)
- void glMultMatrixf(const GLfloat * m)
- void glMultTransposeMatrixd(const GLdouble * m)
- void glMultTransposeMatrixf(const GLfloat * m)
- void glMultiDrawArrays(GLenum mode,GLint * first,GLsizei * count,GLsizei primcount)
- void glMultiDrawElements(GLenum mode,const GLsizei * count,GLenum type,const GLvoid ** indices,GLsizei primcount)
- void glMultiTexCoord1s(GLenum target,GLshort s)
- void glMultiTexCoord1i(GLenum target,GLint s)
- void glMultiTexCoord1f(GLenum target,GLfloat s)
- void glMultiTexCoord1d(GLenum target,GLdouble s)

- void glMultiTexCoord2s(GLenum target,GLshort s,GLshort t)
- void glMultiTexCoord2i(GLenum target,GLint s,GLint t)
- void glMultiTexCoord2f(GLenum target,GLfloat s,GLfloat t)
- void glMultiTexCoord2d(GLenum target,GLdouble s,GLdouble t)
- void glMultiTexCoord3s(GLenum target,GLshort s,GLshort t,GLshort r)
- void glMultiTexCoord3i(GLenum target,GLint s,GLint t,GLint r)
- void glMultiTexCoord3f(GLenum target,GLfloat s,GLfloat t,GLfloat r)
- void glMultiTexCoord3d(GLenum target,GLdouble s,GLdouble t,GLdouble r)
- void glMultiTexCoord4s(GLenum target,GLshort s,GLshort t,GLshort r,GLshort q)
- void glMultiTexCoord4i(GLenum target,GLint s,GLint t,GLint r,GLint q)
- void glMultiTexCoord4f(GLenum target,GLfloat s,GLfloat t,GLfloat r,GLfloat q)
- void glMultiTexCoord4d(GLenum target,GLdouble s,GLdouble t,GLdouble r,GLdouble q)
- void glMultiTexCoord1sv(GLenum target,const GLshort * v)
- void glMultiTexCoord1iv(GLenum target,const GLint * v)
- void glMultiTexCoord1fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord1dv(GLenum target,const GLdouble * v)
- void glMultiTexCoord2sv(GLenum target,const GLshort * v)
- void glMultiTexCoord2iv(GLenum target,const GLint * v)
- void glMultiTexCoord2fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord2dv(GLenum target,const GLdouble * v)
- void glMultiTexCoord3sv(GLenum target,const GLshort * v)
- void glMultiTexCoord3iv(GLenum target,const GLint * v)
- void glMultiTexCoord3fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord3dv(GLenum target,const GLdouble * v)
- void glMultiTexCoord4sv(GLenum target,const GLshort * v)
- void glMultiTexCoord4iv(GLenum target,const GLint * v)
- void glMultiTexCoord4fv(GLenum target,const GLfloat * v)
- void glMultiTexCoord4dv(GLenum target,const GLdouble * v)
- void glNewList(GLuint list,GLenum mode)
- void glNormal3b(GLbyte nx,GLbyte ny,GLbyte nz)
- void glNormal3d(GLdouble nx,GLdouble ny,GLdouble nz)
- void glNormal3f(GLfloat nx,GLfloat ny,GLfloat nz)
- void glNormal3i(GLint nx,GLint ny,GLint nz)
- void glNormal3s(GLshort nx,GLshort ny,GLshort nz)
- void glNormal3bv(const GLbyte * v)
- void glNormal3dv(const GLdouble * v)

- void glNormal3fv(const GLfloat * v)
- void glNormal3iv(const GLint * v)
- void glNormal3sv(const GLshort * v)
- void glNormalPointer(GLenum type,GLsizei stride,const GLvoid * pointer)
- void glOrtho(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top,GLdouble nearVal,GLdouble farVal)
- void glPassThrough(GLfloat token)
- void glPixelMapfv(GLenum map,GLsizei mapsize,const GLfloat * values)
- void glPixelMapuiv(GLenum map,GLsizei mapsize,const GLuint * values)
- void glPixelMapusv(GLenum map,GLsizei mapsize,const GLushort * values)
- void glPixelStoref(GLenum pname,GLfloat param)
- void glPixelStorei(GLenum pname,GLint param)
- void glPixelTransferf(GLenum pname,GLfloat param)
- void glPixelTransferi(GLenum pname,GLint param)
- void glPixelZoom(GLfloat xfactor,GLfloat yfactor)
- void glPointParameterf(GLenum pname,GLfloat param)
- void glPointParameteri(GLenum pname,GLint param)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face,GLenum mode)
- void glPolygonOffset(GLfloat factor,GLfloat units)
- void glPolygonStipple(const GLubyte * pattern)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glPrioritizeTextures(GLsizei n,const GLuint * textures,const GLclampf * priorities)
- void glPopMatrix(void)
- void glRasterPos2s(GLshort x,GLshort y)
- void glRasterPos2i(GLint x,GLint y)
- void glRasterPos2f(GLfloat x,GLfloat y)
- void glRasterPos2d(GLdouble x,GLdouble y)
- void glRasterPos3s(GLshort x,GLshort y,GLshort z)
- void glRasterPos3i(GLint x,GLint y,GLint z)
- void glRasterPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glRasterPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glRasterPos4s(GLshort x,GLshort y,GLshort z,GLshort w)

- void glRasterPos4i(GLint x,GLint y,GLint z,GLint w)
- void glRasterPos4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)
- void glRasterPos4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x,GLint y,GLsizei width,GLsizei height,GLenum format,GLenum type,GLvoid * data)
- void glRectd(GLdouble x1,GLdouble y1,GLdouble x2,GLdouble y2)
- void glRectf(GLfloat x1,GLfloat y1,GLfloat x2,GLfloat y2)
- void glRecti(GLint x1,GLint y1,GLint x2,GLint y2)
- void glRects(GLshort x1,GLshort y1,GLshort x2,GLshort y2)
- void glRectdv(const GLdouble * v1,const GLdouble * v2)
- void glRectfv(const GLfloat * v1,const GLfloat * v2)
- void glRectiv(const GLint * v1,const GLint * v2)
- void glRectsv(const GLshort * v1,const GLshort * v2)
- GLint glRenderMode(GLenum mode)
- void glResetHistogram(GLenum target)
- void glRotated(GLdouble angle,GLdouble x,GLdouble y,GLdouble z)
- void glRotatef(GLfloat angle,GLfloat x,GLfloat y,GLfloat z)
- void glSampleCoverage(GLclampf value,GLboolean invert)
- void glScaled(GLdouble x,GLdouble y,GLdouble z)
- void glScalef(GLfloat x,GLfloat y,GLfloat z)
- void glScissor(GLint x,GLint y,GLsizei width,GLsizei height)
- void glSecondaryColor3b(GLbyte red,GLbyte green,GLbyte blue)
- void glSecondaryColor3s(GLshort red,GLshort green,GLshort blue)
- void glSecondaryColor3i(GLint red,GLint green,GLint blue)
- void glSecondaryColor3f(GLfloat red,GLfloat green,GLfloat blue)
- void glSecondaryColor3d(GLdouble red,GLdouble green,GLdouble blue)
- void glSecondaryColor3ub(GLubyte red,GLubyte green,GLubyte blue)
- void glSecondaryColor3us(GLushort red,GLushort green,GLushort blue)
- void glSecondaryColor3ui(GLuint red,GLuint green,GLuint blue)
- void glSecondaryColor3bv(const GLbyte * v)
- void glSecondaryColor3sv(const GLshort * v)
- void glSecondaryColor3iv(const GLint * v)
- void glSecondaryColor3fv(const GLfloat * v)
- void glSecondaryColor3dv(const GLdouble * v)
- void glSecondaryColor3ubv(const GLubyte * v)
- void glSecondaryColor3usv(const GLushort * v)

- void glSecondaryColor3uiv(const GLuint * v)
- void glSecondaryColorPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glSelectBuffer(GLsizei size,GLuint * buffer)
- void glSeparableFilter2D(GLenum target,GLenum internalformat,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * row,const GLvoid * column)
- void glShadeModel(GLenum mode)
- void glShaderSource(GLuint shader,GLsizei count,const GLchar **string,const GLint *length)
- void glStencilFunc(GLenum func,GLint ref,GLuint mask)
- void glStencilFuncSeparate(GLenum face,GLenum func,GLint ref,GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face,GLuint mask)
- void glStencilOp(GLenum sfail,GLenum dpfail,GLenum dppass)
- void glStencilOpSeparate(GLenum face,GLenum sfail,GLenum dpfail,GLenum dppass)
- void glTexCoord1s(GLshort s)
- void glTexCoord1i(GLint s)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1d(GLdouble s)
- void glTexCoord2s(GLshort s,GLshort t)
- void glTexCoord2i(GLint s,GLint t)
- void glTexCoord2f(GLfloat s,GLfloat t)
- void glTexCoord2d(GLdouble s,GLdouble t)
- void glTexCoord3s(GLshort s,GLshort t,GLshort r)
- void glTexCoord3i(GLint s,GLint t,GLint r)
- void glTexCoord3f(GLfloat s,GLfloat t,GLfloat r)
- void glTexCoord3d(GLdouble s,GLdouble t,GLdouble r)
- void glTexCoord4s(GLshort s,GLshort t,GLshort r,GLshort q)
- void glTexCoord4i(GLint s,GLint t,GLint r,GLint q)
- void glTexCoord4f(GLfloat s,GLfloat t,GLfloat r,GLfloat q)
- void glTexCoord4d(GLdouble s,GLdouble t,GLdouble r,GLdouble q)
- void glTexCoord1sv(const GLshort * v)
- void glTexCoord1iv(const GLint * v)
- void glTexCoord1fv(const GLfloat * v)
- void glTexCoord1dv(const GLdouble * v)
- void glTexCoord2sv(const GLshort * v)
- void glTexCoord2iv(const GLint * v)
- void glTexCoord2fv(const GLfloat * v)

- void glTexCoord2dv(const GLdouble * v)
- void glTexCoord3sv(const GLshort * v)
- void glTexCoord3iv(const GLint * v)
- void glTexCoord3fv(const GLfloat * v)
- void glTexCoord3dv(const GLdouble * v)
- void glTexCoord4sv(const GLshort * v)
- void glTexCoord4iv(const GLint * v)
- void glTexCoord4fv(const GLfloat * v)
- void glTexCoord4dv(const GLdouble * v)
- void glTexCoordPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glTexEnvf(GLenum target,GLenum pname,GLfloat param)
- void glTexEnvi(GLenum target,GLenum pname,GLint param)
- void glTexGeni(GLenum coord,GLenum pname,GLint param)
- void glTexGenf(GLenum coord,GLenum pname,GLfloat param)
- void glTexGend(GLenum coord,GLenum pname,GLdouble param)
- void glTexGeniv(GLenum coord,GLenum pname,const GLint * params)
- void glTexGenfv(GLenum coord,GLenum pname,const GLfloat * params)
- void glTexGendv(GLenum coord,GLenum pname,const GLdouble * params)
- void glTexImage1D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLint border,GLenum format,GLenum type,const GLvoid * data)
- void glTexImage2D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLint border,GLenum format,GLenum type,const GLvoid * data)
- void glTexImage3D(GLenum target,GLint level,GLint internalFormat,GLsizei width,GLsizei height,GLsizei depth,GLint border,GLenum format,GLenum type,const GLvoid * data)
- void glTexParameterf(GLenum target,GLenum pname,GLfloat param)
- void glTexParameteri(GLenum target,GLenum pname,GLint param)
- void glTexParameterfv(GLenum target,GLenum pname,const GLfloat * params)
- void glTexParameteriv(GLenum target,GLenum pname,const GLint * params)
- void glTexSubImage1D(GLenum target,GLint level,GLint xoffset,GLsizei width,GLenum format,GLenum type,const GLvoid * data)
- void glTexSubImage2D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLsizei width,GLsizei height,GLenum format,GLenum type,const GLvoid * data)
- void glTexSubImage3D(GLenum target,GLint level,GLint xoffset,GLint yoffset,GLint zoffset,GLsizei width,GLsizei height,GLsizei depth,GLenum format,GLenum type,const GLvoid * data)
- void glTranslated(GLdouble x,GLdouble y,GLdouble z)
- void glTranslatef(GLfloat x,GLfloat y,GLfloat z)
- void glUniform1f(GLint location,GLfloat v0)
- void glUniform2f(GLint location,GLfloat v0,GLfloat v1)

- void glUniform3f(GLint location,GLfloat v0,GLfloat v1,GLfloat v2)
- void glUniform4f(GLint location,GLfloat v0,GLfloat v1,GLfloat v2,GLfloat v3)
- void glUniform1i(GLint location,GLint v0)
- void glUniform2i(GLint location,GLint v0,GLint v1)
- void glUniform3i(GLint location,GLint v0,GLint v1,GLint v2)
- void glUniform4i(GLint location,GLint v0,GLint v1,GLint v2,GLint v3)
- void glUniform1fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform2fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform3fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform4fv(GLint location,GLsizei count,const GLfloat *value)
- void glUniform1iv(GLint location,GLsizei count,const GLint *value)
- void glUniform2iv(GLint location,GLsizei count,const GLint *value)
- void glUniform3iv(GLint location,GLsizei count,const GLint *value)
- void glUniform4iv(GLint location,GLsizei count,const GLint *value)
- void glUniformMatrix2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix2x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x2fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix3x4fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUniformMatrix4x3fv(GLint location,GLsizei count,GLboolean transpose,const GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertex2s(GLshort x,GLshort y)
- void glVertex2i(GLint x,GLint y)
- void glVertex2f(GLfloat x,GLfloat y)
- void glVertex2d(GLdouble x,GLdouble y)
- void glVertex3s(GLshort x,GLshort y,GLshort z)
- void glVertex3i(GLint x,GLint y,GLint z)
- void glVertex3f(GLfloat x,GLfloat y,GLfloat z)
- void glVertex3d(GLdouble x,GLdouble y,GLdouble z)
- void glVertex4s(GLshort x,GLshort y,GLshort z,GLshort w)
- void glVertex4i(GLint x,GLint y,GLint z,GLint w)
- void glVertex4f(GLfloat x,GLfloat y,GLfloat z,GLfloat w)

- void glVertex4d(GLdouble x,GLdouble y,GLdouble z,GLdouble w)
- void glVertex2sv(const GLshort * v)
- void glVertex2iv(const GLint * v)
- void glVertex2fv(const GLfloat * v)
- void glVertex2dv(const GLdouble * v)
- void glVertex3sv(const GLshort * v)
- void glVertex3iv(const GLint * v)
- void glVertex3fv(const GLfloat * v)
- void glVertex3dv(const GLdouble * v)
- void glVertex4sv(const GLshort * v)
- void glVertex4iv(const GLint * v)
- void glVertex4fv(const GLfloat * v)
- void glVertex4dv(const GLdouble * v)
- void glVertexAttrib1f(GLuint index,GLfloat v0)
- void glVertexAttrib1s(GLuint index,GLshort v0)
- void glVertexAttrib1d(GLuint index,GLdouble v0)
- void glVertexAttrib2f(GLuint index,GLfloat v0,GLfloat v1)
- void glVertexAttrib2s(GLuint index,GLshort v0,GLshort v1)
- void glVertexAttrib2d(GLuint index,GLdouble v0,GLdouble v1)
- void glVertexAttrib3f(GLuint index,GLfloat v0,GLfloat v1,GLfloat v2)
- void glVertexAttrib3s(GLuint index,GLshort v0,GLshort v1,GLshort v2)
- void glVertexAttrib3d(GLuint index,GLdouble v0,GLdouble v1,GLdouble v2)
- void glVertexAttrib4f(GLuint index,GLfloat v0,GLfloat v1,GLfloat v2,GLfloat v3)
- void glVertexAttrib4s(GLuint index,GLshort v0,GLshort v1,GLshort v2,GLshort v3)
- void glVertexAttrib4d(GLuint index,GLdouble v0,GLdouble v1,GLdouble v2,GLdouble v3)
- void glVertexAttrib4Nub(GLuint index,GLubyte v0,GLubyte v1,GLubyte v2,GLubyte v3)
- void glVertexAttrib1fv(GLuint index,const GLfloat *v)
- void glVertexAttrib1sv(GLuint index,const GLshort *v)
- void glVertexAttrib1dv(GLuint index,const GLdouble *v)
- void glVertexAttrib2fv(GLuint index,const GLfloat *v)
- void glVertexAttrib2sv(GLuint index,const GLshort *v)
- void glVertexAttrib2dv(GLuint index,const GLdouble *v)
- void glVertexAttrib3fv(GLuint index,const GLfloat *v)
- void glVertexAttrib3sv(GLuint index,const GLshort *v)
- void glVertexAttrib3dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4fv(GLuint index,const GLfloat *v)

- void glVertexAttrib4sv(GLuint index,const GLshort *v)
- void glVertexAttrib4dv(GLuint index,const GLdouble *v)
- void glVertexAttrib4iv(GLuint index,const GLint *v)
- void glVertexAttrib4bv(GLuint index,const GLbyte *v)
- void glVertexAttrib4ubv(GLuint index,const GLubyte *v)
- void glVertexAttrib4usv(GLuint index,const GLushort *v)
- void glVertexAttrib4uiv(GLuint index,const GLuint *v)
- void glVertexAttribPointer(GLuint index,GLint size,GLenum type,GLboolean normalized,GLsizei stride,const GLvoid * pointer)
- void glVertexPointer(GLint size,GLenum type,GLsizei stride,const GLvoid * pointer)
- void glViewport(GLint x,GLint y,GLsizei width,GLsizei height)
- void glWindowPos2s(GLshort x,GLshort y)
- void glWindowPos2i(GLint x,GLint y)
- void glWindowPos2f(GLfloat x,GLfloat y)
- void glWindowPos2d(GLdouble x,GLdouble y)
- void glWindowPos3s(GLshort x,GLshort y,GLshort z)
- void glWindowPos3i(GLint x,GLint y,GLint z)
- void glWindowPos3f(GLfloat x,GLfloat y,GLfloat z)
- void glWindowPos3d(GLdouble x,GLdouble y,GLdouble z)
- void glWindowPos2sv(const GLshort * v)
- void glWindowPos2iv(const GLint * v)
- void glWindowPos2fv(const GLfloat * v)
- void glWindowPos2dv(const GLdouble * v)
- void glWindowPos3sv(const GLshort * v)
- void glWindowPos3iv(const GLint * v)
- void glWindowPos3fv(const GLfloat * v)
- void glWindowPos3dv(const GLdouble * v)
- void gluBeginCurve(GLUnurbs* nurb)
- void gluBeginPolygon(GLUtesselator* tess)
- void gluBeginSurface(GLUnurbs* nurb)
- void gluBeginTrim(GLUnurbs* nurb)
- void gluCylinder(GLUquadric* quad,GLdouble base,GLdouble top,GLdouble height,GLint slices,GLint stacks)
- void gluDeleteNurbsRenderer(GLUnurbs* nurb)
- void gluDeleteQuadric(GLUquadric* quad)
- void gluDeleteTess(GLUtesselator* tess)
- void gluDisk(GLUquadric* quad,GLdouble inner,GLdouble outer,GLint slices,GLint loops)

- void gluEndCurve(GLUnurbs* nurb)
- void gluEndPolygon(GLUtesselator* tess)
- void gluEndSurface(GLUnurbs* nurb)
- void gluEndTrim(GLUnurbs* nurb)
- const GLubyte * gluErrorString(GLenum error)
- void gluGetNurbsProperty(GLUnurbs* nurb,GLenum property,GLfloat* data)
- const GLubyte * gluGetString(GLenum name)
- void gluGetTessProperty(GLUtesselator* tess,GLenum which,GLdouble* data)
- void gluLoadSamplingMatrices(GLUnurbs* nurb,const GLfloat * model,const GLfloat * perspective,const GLint * view)
- void gluLookAt(GLdouble eyeX,GLdouble eyeY,GLdouble eyeZ,GLdouble centerX,GLdouble centerY,GLdouble centerZ,GLdouble upX,GLdouble upY,GLdouble upZ)
- GLUnurbs *gluNewNurbsRenderer(void)
- GLUquadric *gluNewQuadric(void)
- GLUtesselator* gluNewTess(void)
- void gluNextContour(GLUtesselator* tess,GLenum type)
- void gluNurbsCurve(GLUnurbs* nurb,GLint knotCount,GLfloat * knots,GLint stride,GLfloat * control,GLint order,GLenum type)
- void gluNurbsProperty(GLUnurbs* nurb,GLenum property,GLfloat value)
- void gluNurbsSurface(GLUnurbs* nurb,GLint sKnotCount,GLfloat* sKnots,GLint tKnotCount,GLfloat* tKnots,GLint sStride,GLint tStride,GLfloat* control,GLint sOrder,GLint tOrder,GLenum type)
- void gluOrtho2D(GLdouble left,GLdouble right,GLdouble bottom,GLdouble top)
- void gluPartialDisk(GLUquadric* quad,GLdouble inner,GLdouble outer,GLint slices,GLint loops,GLdouble start,GLdouble sweep)
- void gluPerspective(GLdouble fovy,GLdouble aspect,GLdouble zNear,GLdouble zFar)
- void gluPickMatrix(GLdouble x,GLdouble y,GLdouble delX,GLdouble delY,GLint * viewport)
- GLint gluProject(GLdouble objX,GLdouble objY,GLdouble objZ,const GLdouble * model,const GLdouble * proj,const GLint * view,GLdouble* winX,GLdouble* winY,GLdouble* winZ)
- void gluPwlCurve(GLUnurbs* nurb,GLint count,GLfloat* data,GLint stride,GLenum type)
- void gluQuadricDrawStyle(GLUquadric* quad,GLenum draw)
- void gluQuadricNormals(GLUquadric* quad,GLenum normal)
- void gluQuadricOrientation(GLUquadric* quad,GLenum orientation)
- void gluQuadricTexture(GLUquadric* quad,GLboolean texture)
- GLint gluScaleImage(GLenum format,GLsizei wIn,GLsizei hIn,GLenum typeIn,const void * dataIn,GLsizei wOut,GLsizei hOut,GLenum typeOut,GLvoid* dataOut)
- void gluSphere(GLUquadric* quad,GLdouble radius,GLint slices,GLint stacks)
- void gluTessBeginContour(GLUtesselator* tess)
- void gluTessBeginPolygon(GLUtesselator* tess,GLvoid* data)

- void gluTessEndContour(GLUtesselator* tess)
- void gluTessEndPolygon(GLUtesselator* tess)
- void gluTessNormal(GLUtesselator* tess,GLdouble valueX,GLdouble valueY,GLdouble valueZ)
- void gluTessProperty(GLUtesselator* tess,GLenum which,GLdouble data)
- void gluTessVertex(GLUtesselator* tess,GLdouble * location,GLvoid* data)
- GLint gluUnProject(GLdouble winX,GLdouble winY,GLdouble winZ,const GLdouble * model,const GLdouble * proj,const GLint * view,GLdouble* objX,GLdouble* objY,GLdouble* objZ)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glBindVertexArray(GLuint array)
- void glGenVertexArrays(GLsizei n,const GLuint * arrays)
- GLenum glewInit(void)
- GLboolean glewIsSupported(const char *name)
- GLboolean glewGetExtension(const char *name)
- const GLubyte *glewGetErrorString(GLenum error)
- const GLubyte *glewGetString(GLenum name)

CHAPTER

NINETYONE

RINGQT CLASSES REFERENCE

91.1 AbstractAxis Class

Parameters : void

- Qt::Alignment alignment(void)
- QColor gridLineColor(void)
- QPen gridLinePen(void)
- void hide(void)
- bool isGridLineVisible(void)
- bool isLineVisible(void)
- bool isMinorGridLineVisible(void)
- bool isReverse(void)
- bool isTitleVisible(void)
- bool isVisible(void)
- int labelsAngle(void)
- QBrush labelsBrush(void)
- QColor labelsColor(void)
- bool labelsVisible(void)
- QPen linePen(void)
- QColor linePenColor(void)
- QColor minorGridLineColor(void)
- QPen minorGridLinePen(void)
- Qt::Orientation orientation(void)
- void setGridLineColor(QColor color)
- void setGridLinePen(QPen pen)
- void setGridLineVisible(bool visible)
- void setLabelsAngle(int angle)
- void setLabelsBrush(QBrush brush)
- void setLabelsColor(QColor color)

- void setLabelsVisible(bool visible)
- void setLinePen(QPen pen)
- void setLinePenColor(QColor color)
- void setLineVisible(bool visible)
- void setMax(QVariant max)
- void setMin(QVariant min)
- void setMinorGridLineColor(QColor color)
- void setMinorGridLinePen(QPen pen)
- void setMinorGridLineVisible(bool visible)
- void setRange(QVariant min, QVariant max)
- void setReverse(bool reverse)
- void setShadesBorderColor(QColor color)
- void setShadesBrush(QBrush brush)
- void setShadesColor(QColor color)
- void setShadesPen(QPen pen)
- void setShadesVisible(bool visible)
- void setTitleBrush(QBrush brush)
- void setTitleFont(QFont font)
- void setTitleText(QString title)
- void setTitleVisible(bool visible)
- void setVisible(bool visible)
- QColor shadesBorderColor(void)
- QBrush shadesBrush(void)
- QColor shadesColor(void)
- QPen shadesPen(void)
- bool shadesVisible(void)
- void show(void)
- QBrush titleBrush(void)
- QFont titleFont(void)
- QString titleText(void)
- void setcolorChangedEvent(const char *)
- void setgridLineColorChangedEvent(const char *)
- void setgridLinePenChangedEvent(const char *)
- void setgridVisibleChangedEvent(const char *)
- void setlabelsAngleChangedEvent(const char *)
- void setlabelsBrushChangedEvent(const char *)

- void setlabelsColorChangedEvent(const char *)
- void setlabelsEditableChangedEvent(const char *)
- void setlabelsFontChangedEvent(const char *)
- void setlabelsVisibleChangedEvent(const char *)
- void setlinePenChangedEvent(const char *)
- void setlineVisibleChangedEvent(const char *)
- void setminorGridLineColorChangedEvent(const char *)
- void setminorGridLinePenChangedEvent(const char *)
- void setminorGridVisibleChangedEvent(const char *)
- void setreverseChangedEvent(const char *)
- void setshadesBorderColorChangedEvent(const char *)
- void setshadesBrushChangedEvent(const char *)
- void setshadesColorChangedEvent(const char *)
- void setshadesPenChangedEvent(const char *)
- void setshadesVisibleChangedEvent(const char *)
- void settitleBrushChangedEvent(const char *)
- void settitleFontChangedEvent(const char *)
- void settitleTextChangedEvent(const char *)
- void settitleVisibleChangedEvent(const char *)
- void setvisibleChangedEvent(const char *)
- const char *getcolorChangedEvent(void)
- const char *getgridLineColorChangedEvent(void)
- const char *getgridLinePenChangedEvent(void)
- const char *getgridVisibleChangedEvent(void)
- const char *getlabelsAngleChangedEvent(void)
- const char *getlabelsBrushChangedEvent(void)
- const char *getlabelsColorChangedEvent(void)
- const char *getlabelsEditableChangedEvent(void)
- const char *getlabelsFontChangedEvent(void)
- const char *getlabelsVisibleChangedEvent(void)
- const char *getlinePenChangedEvent(void)
- const char *getlineVisibleChangedEvent(void)
- const char *getminorGridLineColorChangedEvent(void)
- const char *getminorGridLinePenChangedEvent(void)
- const char *getminorGridVisibleChangedEvent(void)
- const char *getreverseChangedEvent(void)

- const char *getshadesBorderColorChangedEvent(void)
- const char *getshadesBrushChangedEvent(void)
- const char *getshadesColorChangedEvent(void)
- const char *getshadesPenChangedEvent(void)
- const char *getshadesVisibleChangedEvent(void)
- const char *gettitleBrushChangedEvent(void)
- const char *gettitleFontChangedEvent(void)
- const char *gettitleTextChangedEvent(void)
- const char *gettitleVisibleChangedEvent(void)
- const char *getvisibleChangedEvent(void)

91.2 AbstractBarSeries Class

Parameters : void

Parent Class : AbstractSeries

- bool append(QBarSet *set)
- void clear(void)
- int count(void)
- bool insert(int index, QBarSet *set)
- bool isLabelsVisible(void)
- qreal labelsAngle(void)
- QString labelsFormat(void)
- QAbstractBarSeries::LabelsPosition labelsPosition(void)
- int labelsPrecision(void)
- bool remove(QBarSet *set)
- void setBarWidth(qreal width)
- void setLabelsAngle(qreal angle)
- void setLabelsFormat(QString format)
- void setLabelsPosition(QAbstractBarSeries::LabelsPosition position)
- void setLabelsPrecision(int precision)
- void setLabelsVisible(bool visible)
- bool take(QBarSet *set)
- void setbarsetsAddedEvent(const char *)
- void setbarsetsRemovedEvent(const char *)
- void setclickedEvent(const char *)
- void setcountChangedEvent(const char *)

- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setlabelsAngleChangedEvent(const char *)
- void setlabelsFormatChangedEvent(const char *)
- void setlabelsPositionChangedEvent(const char *)
- void setlabelsPrecisionChangedEvent(const char *)
- void setlabelsVisibleChangedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- const char *getbarsetsAddedEvent(void)
- const char *getbarsetsRemovedEvent(void)
- const char *getclickedEvent(void)
- const char *getcountChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getlabelsAngleChangedEvent(void)
- const char *getlabelsFormatChangedEvent(void)
- const char *getlabelsPositionChangedEvent(void)
- const char *getlabelsPrecisionChangedEvent(void)
- const char *getlabelsVisibleChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)

91.3 CodeEditor Class

Parameters : QWidget *

Parent Class : QPlainTextEdit

- void setCompleter(QCompleter *c)
- QCompleter *completer(void)
- void setLineNumbersAreaColor(QColor oColor)
- void setLineNumbersAreaBackColor(QColor oColor)

91.4 QAbstractAspect Class

C++ Reference : http://doc.qt.io/qt-5/qt3dcore-qabstractaspect.html

Parameters : QObject *

Parent Class : QObject

- void scheduleSingleShotJob(Qt3DCore::QAspectJobPtr job)

91.5 QAbstractButton Class

C++ Reference : http://doc.qt.io/qt-5/qabstractbutton.html

Parameters : QWidget *parent

Parent Class : QWidget

- bool autoExclusive(void)
- bool autoRepeat(void)
- int autoRepeatDelay(void)
- int autoRepeatInterval(void)
- QButtonGroup *group(void)
- QIcon icon(void)
- QSize iconSize(void)
- bool isCheckable(void)
- bool isChecked(void)
- bool isDown(void)
- void setAutoExclusive(bool)
- void setAutoRepeat(bool)
- void setAutoRepeatDelay(int)
- void setAutoRepeatInterval(int)
- void setCheckable(bool)
- void setDown(bool)
- void setIcon(QIcon)
- void setShortcut(QKeySequence)
- void setText(QString)
- QKeySequence shortcut(void)
- QString text(void)
- void animateClick(int msec)
- void click(void)
- void setChecked(bool)
- void setIconSize(QSize)

- void toggle(void)

91.6 QAbstractCameraController Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qabstractcameracontroller.html

Parent Class : QEntity

- float acceleration(void)
- Qt3DRender::QCamera * camera(void)
- float deceleration(void)
- float linearSpeed(void)
- float lookSpeed(void)
- void setAcceleration(float acceleration)
- void setCamera(Qt3DRender::QCamera *camera)
- void setDeceleration(float deceleration)
- void setLinearSpeed(float linearSpeed)
- void setLookSpeed(float lookSpeed)

91.7 QAbstractItemView Class

C++ Reference : http://doc.qt.io/qt-5/qabstractitemview.html

Parameters : QWidget *parent

Parent Class : QAbstractScrollArea

- bool alternatingRowColors(void)
- int autoScrollMargin(void)
- void closePersistentEditor(QModelIndex)
- QModelIndex currentIndex(void)
- int defaultDropAction(void)
- int dragDropMode(void)
- bool dragDropOverwriteMode(void)
- bool dragEnabled(void)
- int editTriggers(void)
- bool hasAutoScroll(void)
- int horizontalScrollMode(void)
- QSize iconSize(void)
- QModelIndex indexAt(QPoint)
- QWidget *indexWidget(QModelIndex)
- QAbstractItemDelegate *itemDelegate(QModelIndex)

- QAbstractItemDelegate *itemDelegateForColumn(int column)
- QAbstractItemDelegate *itemDelegateForRow(int row)
- void keyboardSearch(QString)
- QAbstractItemModel *model(void)
- void openPersistentEditor(QModelIndex)
- QModelIndex rootIndex(void)
- void scrollTo(QModelIndex,QAbstractItemView::ScrollHint)
- int selectionBehavior(void)
- int selectionMode(void)
- QItemSelectionModel *selectionModel(void)
- void setAlternatingRowColors(bool enable)
- void setAutoScroll(bool enable)
- void setAutoScrollMargin(int margin)
- void setDefaultDropAction(Qt::DropAction dropAction)
- void setDragDropMode(QAbstractItemView::DragDropMode behavior)
- void setDragDropOverwriteMode(bool overwrite)
- void setDragEnabled(bool enable)
- void setDropIndicatorShown(bool enable)
- void setEditTriggers(QAbstractItemView::EditTrigger triggers)
- void setHorizontalScrollMode(QAbstractItemView::ScrollMode mode)
- void setIconSize(QSize)
- void setIndexWidget(QModelIndex, QWidget *widget)
- void setItemDelegate(QAbstractItemDelegate *delegate)
- void setItemDelegateForColumn(int column, QAbstractItemDelegate *delegate)
- void setItemDelegateForRow(int row, QAbstractItemDelegate *delegate)
- void setModel(QAbstractItemModel *model)
- void setSelectionBehavior(QAbstractItemView::SelectionBehavior behavior)
- void setSelectionMode(QAbstractItemView::SelectionMode mode)
- void setSelectionModel(QItemSelectionModel *selectionModel)
- void setTabKeyNavigation(bool enable)
- void setTextElideMode(Qt::TextElideMode mode)
- void setVerticalScrollMode(QAbstractItemView::ScrollMode mode)
- bool showDropIndicator(void)
- int sizeHintForColumn(int column)
- QSize sizeHintForIndex(QModelIndex)
- int sizeHintForRow(int row)

- bool tabKeyNavigation(void)
- int textElideMode(void)
- int verticalScrollMode(void)
- QRect visualRect(QModelIndex)
- void clearSelection(void)
- void edit(QModelIndex)
- void scrollToBottom(void)
- void scrollToTop(void)
- void setCurrentIndex(QModelIndex)
- void update(QModelIndex)

91.8 QAbstractPrintDialog Class

C++ Reference : http://doc.qt.io/qt-5/qabstractprintdialog.html

Parameters : QPrinter *,QWidget *

Parent Class : QDialog

- int fromPage(void)
- int maxPage(void)
- int minPage(void)
- QAbstractPrintDialog::PrintRange printRange(void)
- QPrinter * printer(void)
- void setFromTo(int from, int to)
- void setMinMax(int min, int max)
- void setOptionTabs(QList<QWidget *> tabs)
- void setPrintRange(QAbstractPrintDialog::PrintRange range)
- int toPage(void)

91.9 QAbstractScrollArea Class

C++ Reference : http://doc.qt.io/qt-5/qabstractscrollarea.html

Parameters : QWidget *parent

Parent Class : QFrame

- void addScrollBarWidget(QWidget *widget, Qt::AlignmentFlag alignment)
- QWidget *cornerWidget(void)
- QScrollBar *horizontalScrollBar(void)
- int horizontalScrollBarPolicy(void)
- QSize maximumViewportSize(void)

- QWidgetList scrollBarWidgets(Qt::AlignmentFlag)
- void setCornerWidget(QWidget *widget)
- void setHorizontalScrollBar(QScrollBar *scrollBar)
- void setHorizontalScrollBarPolicy(Qt::ScrollBarPolicy)
- void setVerticalScrollBar(QScrollBar *scrollBar)
- void setVerticalScrollBarPolicy(Qt::ScrollBarPolicy)
- void setViewport(QWidget *widget)
- QScrollBar *verticalScrollBar(void)
- int verticalScrollBarPolicy(void)
- QWidget *viewport(void)

91.10 QAbstractSeries Class

C++ Reference : http://doc.qt.io/qt-5/qabstractseries.html

Parameters : void

Parent Class : QObject

- bool attachAxis(QAbstractAxis *axis)
- QList<QAbstractAxis *> attachedAxes(void)
- QChart *chart(void)
- bool detachAxis(QAbstractAxis *axis)
- void hide(void)
- bool isVisible(void)
- QString name(void)
- qreal opacity(void)
- void setName(QString name)
- void setOpacity(qreal opacity)
- void setUseOpenGL(bool enable)
- void setVisible(bool visible)
- void show(void)
- bool useOpenGL(void)
- void setnameChangedEvent(const char *)
- void setopacityChangedEvent(const char *)
- void setuseOpenGLChangedEvent(const char *)
- void setvisibleChangedEvent(const char *)
- const char *getnameChangedEvent(void)
- const char *getopacityChangedEvent(void)

- const char *getuseOpenGLChangedEvent(void)
- const char *getvisibleChangedEvent(void)

91.11 QAbstractSlider Class

C++ Reference : http://doc.qt.io/qt-5/qabstractslider.html

Parameters : QWidget *parent

Parent Class : QWidget

- bool hasTracking(void)
- bool invertedAppearance(void)
- bool invertedControls(void)
- bool isSliderDown(void)
- int maximum(void)
- int minimum(void)
- int orientation(void)
- int pageStep(void)
- void setInvertedAppearance(bool)
- void setInvertedControls(bool)
- void setMaximum(int)
- void setMinimum(int)
- void setPageStep(int)
- void setSingleStep(int)
- void setSliderDown(bool)
- void setSliderPosition(int)
- void setTracking(bool enable)
- int singleStep(void)
- int sliderPosition(void)
- void triggerAction(QAbstractSlider::SliderAction action)
- int value(void)
- void setOrientation(Qt::Orientation)
- void setRange(int min, int max)
- void setValue(int)

91.12 QAbstractSocket Class

C++ Reference : http://doc.qt.io/qt-5/qabstractsocket.html

Parameters : void

Parent Class : QIODevice

- void abort(void)
- bool bind(QHostAddress address, int port, QAbstractSocket::BindFlag mode)
- void connectToHost(QString hostName, int port, QIODevice::OpenModeFlag openMode, QAbstractSocket::NetworkLayerProtocol protocol)
- void disconnectFromHost(void)
- int error(void)
- bool flush(void)
- bool isValid(void)
- QHostAddress localAddress(void)
- int localPort(void)
- int pauseMode(void)
- QHostAddress peerAddress(void)
- QString peerName(void)
- int peerPort(void)
- QNetworkProxy proxy(void)
- int readBufferSize(void)
- void resume(void)
- void setPauseMode(QAbstractSocket::PauseMode pauseMode)
- void setProxy(QNetworkProxy networkProxy)
- void setReadBufferSize(int size)
- bool setSocketDescriptor(qintptr socketDescriptor, QAbstractSocket::SocketState socketState, QIODevice::OpenModeFlag openMode)
- void setSocketOption(QAbstractSocket::SocketOption option, QVariant value)
- int *socketDescriptor(void)
- QVariant socketOption(QAbstractSocket::SocketOption option)
- int socketType(void)
- int state(void)
- bool waitForConnected(int msecs)
- bool waitForDisconnected(int msecs)
- bool atEnd(void)
- int bytesAvailable(void)
- int bytesToWrite(void)

- bool canReadLine(void)
- void close(void)
- bool isSequential(void)
- bool waitForBytesWritten(int msecs)
- bool waitForReadyRead(int msecs)
- void setconnectedEvent(const char *)
- void setdisconnectedEvent(const char *)
- void seterrorEvent(const char *)
- void sethostFoundEvent(const char *)
- void setproxyAuthenticationRequiredEvent(const char *)
- void setstateChangedEvent(const char *)
- const char *getconnectedEvent(void)
- const char *getdisconnectedEvent(void)
- const char *geterrorEvent(void)
- const char *gethostFoundEvent(void)
- const char *getproxyAuthenticationRequiredEvent(void)
- const char *getstateChangedEvent(void)

91.13 QAbstractSpinBox Class

C++ Reference : http://doc.qt.io/qt-5/qabstractspinbox.html

Parameters : QWidget *parent

Parent Class : QWidget

- int alignment(void)
- int buttonSymbols(void)
- int correctionMode(void)
- bool hasAcceptableInput(void)
- bool hasFrame(void)
- void interpretText(void)
- bool isAccelerated(void)
- bool keyboardTracking(void)
- void setAccelerated(bool on)
- void setAlignment(Qt::AlignmentFlag flag)
- void setButtonSymbols(QAbstractSpinBox::ButtonSymbols bs)
- void setCorrectionMode(QAbstractSpinBox::CorrectionMode cm)
- void setFrame(bool)

- void setReadOnly(bool r)
- void setSpecialValueText(QString)
- void setWrapping(bool w)
- QString specialValueText(void)
- void stepBy(int steps)
- QString text(void)
- bool wrapping(void)
- void clear(void)
- void selectAll(void)
- void stepDown(void)
- void stepUp(void)

91.14 QAction Class

C++ Reference : http://doc.qt.io/qt-5/qaction.html

Parameters : QWidget *parent

- QActionGroup *actionGroup(void)
- void activate(QAction::ActionEvent event)
- bool autoRepeat(void)
- QVariant data(void)
- QFont font(void)
- QIcon icon(void)
- QString iconText(void)
- bool isCheckable(void)
- bool isChecked(void)
- bool isEnabled(void)
- bool isIconVisibleInMenu(void)
- bool isSeparator(void)
- bool isVisible(void)
- QMenu *menu(void)
- int menuRole(void)
- QWidget *parentWidget(void)
- int priority(void)
- void setActionGroup(QActionGroup *group)
- void setAutoRepeat(bool)
- void setCheckable(bool)

- void setData(QVariant)
- void setFont(QFont)
- void setIcon(QIcon)
- void setIconText(QString)
- void setIconVisibleInMenu(bool visible)
- void setMenu(QMenu *menu)
- void setMenuRole(QAction::MenuRole menuRole)
- void setPriority(QAction::Priority priority)
- void setSeparator(bool b)
- void setShortcut(QKeySequence)
- void setShortcutContext(Qt::ShortcutContext context)
- void setShortcuts(QKeySequence::StandardKey key)
- void setStatusTip(QString)
- void setText(QString)
- void setToolTip(QString)
- void setWhatsThis(QString)
- QKeySequence shortcut(void)
- int shortcutContext(void)
- bool showStatusText(QWidget *widget)
- QString statusTip(void)
- QString text(void)
- QString toolTip(void)
- QString whatsThis(void)
- void hover(void)
- void setChecked(bool)
- void setDisabled(bool)
- void setEnabled(bool)
- void setVisible(bool)
- void toggle(void)
- void trigger(void)
- void setClickEvent(const char *)
- const char *getClickEvent(void)

91.15 QAllEvents Class

Parameters : QWidget *

Parent Class : QWidget

- void accept(void)
- void ignore(void)
- int getKeyCode(void)
- QString getKeyText(void)
- int getModifiers(void)
- int getx(void)
- int gety(void)
- int getglobalx(void)
- int getglobaly(void)
- int getbutton(void)
- int getbuttons(void)
- void setKeyPressEvent(const char *cStr)
- void setMouseButtonPressEvent(const char *cStr)
- void setMouseButtonReleaseEvent(const char *cStr)
- void setMouseButtonDblClickEvent(const char *cStr)
- void setMouseMoveEvent(const char *cStr)
- void setCloseEvent(const char *cStr)
- void setContextMenuEvent(const char *cStr)
- void setDragEnterEvent(const char *cStr)
- void setDragLeaveEvent(const char *cStr)
- void setDragMoveEvent(const char *cStr)
- void setDropEvent(const char *cStr)
- void setEnterEvent(const char *cStr)
- void setFocusInEvent(const char *cStr)
- void setFocusOutEvent(const char *cStr)
- void setKeyReleaseEvent(const char *cStr)
- void setLeaveEvent(const char *cStr)
- void setNonClientAreaMouseButtonDblClickEvent(const char *cStr)
- void setNonClientAreaMouseButtonPressEvent(const char *cStr)
- void setNonClientAreaMouseButtonReleaseEvent(const char *cStr)
- void setNonClientAreaMouseMoveEvent(const char *cStr)
- void setMoveEvent(const char *cStr)

- void setResizeEvent(const char *cStr)
- void setWindowActivateEvent(const char *cStr)
- void setWindowBlockedEvent(const char *cStr)
- void setWindowDeactivateEvent(const char *cStr)
- void setWindowStateChangeEvent(const char *cStr)
- void setWindowUnblockedEvent(const char *cStr)
- void setPaintEvent(const char *cStr)
- void setChildAddedEvent(const char *cStr)
- void setChildPolishedEvent(const char *cStr)
- void setChildRemovedEvent(const char *cStr)
- const char *getKeyPressEvent(void)
- const char *getMouseButtonPressEvent(void)
- const char *getMouseButtonReleaseEvent(void)
- const char *getMouseButtonDblClickEvent(void)
- const char *getMouseMoveEvent(void)
- const char *getCloseEvent(void)
- const char *getContextMenuEvent(void)
- const char *getDragEnterEvent(void)
- const char *getDragLeaveEvent(void)
- const char *getDragMoveEvent(void)
- const char *getDropEvent(void)
- const char *getEnterEvent(void)
- const char *getFocusInEvent(void)
- const char *getFocusOutEvent(void)
- const char *getKeyReleaseEvent(void)
- const char *getLeaveEvent(void)
- const char *getNonClientAreaMouseButtonDblClickEvent(void)
- const char *getNonClientAreaMouseButtonPressEvent(void)
- const char *getNonClientAreaMouseButtonReleaseEvent(void)
- const char *getNonClientAreaMouseMoveEvent(void)
- const char *getMoveEvent(void)
- const char *getResizeEvent(void)
- const char *getWindowActivateEvent(void)
- const char *getWindowBlockedEvent(void)
- const char *getWindowDeactivateEvent(void)
- const char *getWindowStateChangeEvent(void)

- const char *getWindowUnblockedEvent(void)
- const char *getPaintEvent(void)
- const char *getChildAddedEvent(void)
- const char *getChildPolishedEvent(void)
- const char *getChildRemovedEvent(void)
- void setEventOutput(bool x)
- QObject *getParentObject(void)
- QWidget *getParentWidget(void)
- void setKeyPressFunc(const char *cStr)
- void setMouseButtonPressFunc(const char *cStr)
- void setMouseButtonReleaseFunc(const char *cStr)
- void setMouseButtonDblClickFunc(const char *cStr)
- void setMouseMoveFunc(const char *cStr)
- void setCloseFunc(const char *cStr)
- void setContextMenuFunc(const char *cStr)
- void setDragEnterFunc(const char *cStr)
- void setDragLeaveFunc(const char *cStr)
- void setDragMoveFunc(const char *cStr)
- void setDropFunc(const char *cStr)
- void setEnterFunc(const char *cStr)
- void setFocusInFunc(const char *cStr)
- void setFocusOutFunc(const char *cStr)
- void setKeyReleaseFunc(const char *cStr)
- void setLeaveFunc(const char *cStr)
- void setNonClientAreaMouseButtonDblClickFunc(const char *cStr)
- void setNonClientAreaMouseButtonPressFunc(const char *cStr)
- void setNonClientAreaMouseButtonReleaseFunc(const char *cStr)
- void setNonClientAreaMouseMoveFunc(const char *cStr)
- void setMoveFunc(const char *cStr)
- void setResizeFunc(const char *cStr)
- void setWindowActivateFunc(const char *cStr)
- void setWindowBlockedFunc(const char *cStr)
- void setWindowDeactivateFunc(const char *cStr)
- void setWindowStateChangeFunc(const char *cStr)
- void setWindowUnblockedFunc(const char *cStr)
- void setPaintFunc(const char *cStr)

- void setChildAddedFunc(const char *cStr)
- void setChildPolishedFunc(const char *cStr)
- void setChildRemovedFunc(const char *cStr)
- const char *getKeyPressFunc(void)
- const char *getMouseButtonPressFunc(void)
- const char *getMouseButtonReleaseFunc(void)
- const char *getMouseButtonDblClickFunc(void)
- const char *getMouseMoveFunc(void)
- const char *getCloseFunc(void)
- const char *getContextMenuFunc(void)
- const char *getDragEnterFunc(void)
- const char *getDragLeaveFunc(void)
- const char *getDragMoveFunc(void)
- const char *getDropFunc(void)
- const char *getEnterFunc(void)
- const char *getFocusInFunc(void)
- const char *getFocusOutFunc(void)
- const char *getKeyReleaseFunc(void)
- const char *getLeaveFunc(void)
- const char *getNonClientAreaMouseButtonDblClickFunc(void)
- const char *getNonClientAreaMouseButtonPressFunc(void)
- const char *getNonClientAreaMouseButtonReleaseFunc(void)
- const char *getNonClientAreaMouseMoveFunc(void)
- const char *getMoveFunc(void)
- const char *getResizeFunc(void)
- const char *getWindowActivateFunc(void)
- const char *getWindowBlockedFunc(void)
- const char *getWindowDeactivateFunc(void)
- const char *getWindowStateChangeFunc(void)
- const char *getWindowUnblockedFunc(void)
- const char *getPaintFunc(void)
- const char *getChildAddedFunc(void)
- const char *getChildPolishedFunc(void)
- const char *getChildRemovedFunc(void)
- QDropEvent *getDropEventObject(void)
- QDragMoveEvent *getDragMoveEventObject(void)

- QDragEnterEvent *getDragEnterEventObject(void)
- QDragLeaveEvent *getDragLeaveEventObject(void)
- QChildEvent *getChildEventObject(void)

91.16 QApp Class

C++ Reference : http://doc.qt.io/qt-5/qapplication.html

Parent Class : QGuiApplication

- void exec(void)
- void quit(void)
- void processEvents(void)
- void styleWindows(void)
- void styleWindowsVista(void)
- void styleFusion(void)
- void styleFusionBlack(void)
- void styleFusionCustom(QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor,QColor)
- void closeAllWindows(void)
- Qt::KeyboardModifiers keyboardModifiers(void)
- QClipboard *clipboard(void)
- QStyle *style(void)
- void aboutQt(void)
- QWidget *activeModalWidget(void)
- QWidget *activePopupWidget(void)
- QWidget *activeWindow(void)
- QWidget *focusWidget(void)
- double titlebarHeight(void)

91.17 QAreaLegendMarker Class

C++ Reference : http://doc.qt.io/qt-5/qarealegendmarker.html

Parent Class : QLegendMarker

- QAreaSeries * series(void)
- QLegendMarker::LegendMarkerType type(void)

91.18 QAreaSeries Class

C++ Reference : http://doc.qt.io/qt-5/qareaseries.html

Parameters : QObject *

Parent Class : QAbstractSeries

- QColor borderColor(void)
- QBrush brush(void)
- QColor color(void)
- QLineSeries * lowerSeries(void)
- QPen pen(void)
- bool pointLabelsClipping(void)
- QColor pointLabelsColor(void)
- QFont pointLabelsFont(void)
- QString pointLabelsFormat(void)
- bool pointLabelsVisible(void)
- bool pointsVisible(void)
- void setBorderColor(QColor color)
- void setBrush(QBrush brush)
- void setColor(QColor color)
- void setLowerSeries(QLineSeries *series)
- void setPen(QPen pen)
- void setPointLabelsClipping(bool enabled)
- void setPointLabelsColor(QColor color)
- void setPointLabelsFont(QFont font)
- void setPointLabelsFormat(QString format)
- void setPointLabelsVisible(bool visible)
- void setPointsVisible(bool visible)
- void setUpperSeries(QLineSeries *series)
- QLineSeries * upperSeries(void)
- void setborderColorChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setcolorChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setpointLabelsClippingChangedEvent(const char *)
- void setpointLabelsColorChangedEvent(const char *)

- void setpointLabelsFontChangedEvent(const char *)
- void setpointLabelsFormatChangedEvent(const char *)
- void setpointLabelsVisibilityChangedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- const char *getborderColorChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getcolorChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getpointLabelsClippingChangedEvent(void)
- const char *getpointLabelsColorChangedEvent(void)
- const char *getpointLabelsFontChangedEvent(void)
- const char *getpointLabelsFormatChangedEvent(void)
- const char *getpointLabelsVisibilityChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)

91.19 QAspectEngine Class

C++ Reference : http://doc.qt.io/qt-5/qt3dcore-qaspectengine.html

Parameters : QObject *

Parent Class : QObject

- QVector<Qt3DCore::QAbstractAspect *> aspects(void)
- QVariant executeCommand(QString command)
- void registerAspect(Qt3DCore::QAbstractAspect *aspect)
- void registerAspect_2(QString name)
- Qt3DCore::QEntityPtr rootEntity(void)
- void setRootEntity(Qt3DCore::QEntityPtr root)
- void unregisterAspect(Qt3DCore::QAbstractAspect *aspect)
- void unregisterAspect_2(QString name)

91.20 QAxBase Class

C++ Reference : http://doc.qt.io/qt-5/qaxbase.html

Parameters : QWidget *

Parent Class : QObject

- QVariant asVariant(void)
- QString control(void)
- void disableClassInfo(void)
- void disableEventSink(void)
- void disableMetaObject(void)
- QVariant dynamicCall(char *function)
- QVariant dynamicCall_2(char *function,QVariant)
- QVariant dynamicCall_3(char *function,QVariant,QVariant)
- QVariant dynamicCall_4(char *function,QVariant,QVariant,QVariant)
- QVariant dynamicCall_5(char *function,QVariant,QVariant,QVariant,QVariant)
- QVariant dynamicCall_6(char *function,QVariant,QVariant,QVariant,QVariant,QVariant)
- QVariant dynamicCall_7(char *function,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QVariant dynamicCall_8(char *function,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QVariant dynamicCall_9(char *function,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QVariant dynamicCall_10(const char *function, QList<QVariant>)
- QString generateDocumentation(void)
- bool isNull(void)
- QAxObject *querySubObject(const char *name)
- QAxObject *querySubObject_2(const char *name,QVariant)
- QAxObject *querySubObject_3(const char *name,QVariant,QVariant)
- QAxObject *querySubObject_4(const char *name,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_5(const char *name,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_6(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_7(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_8(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_9(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- bool setControl(QString)
- QStringList verbs(void)

91.21 QAxObject Class

C++ Reference : http://doc.qt.io/qt-5/qaxobject.html

Parameters : QString

Parent Class : QAxBase

- bool doVerb(QString)

91.22 QAxWidget Class

C++ Reference : http://doc.qt.io/qt-5/qaxwidget.html

Parameters : QWidget *parent, Qt::WindowFlags f

Parent Class : QAxBase

- bool doVerb(QString)

91.23 QAxWidget2 Class

Parameters : QString c, QWidget *parent, Qt::WindowFlags f

Parent Class : QAxWidget

91.24 QBarCategoryAxis Class

C++ Reference : http://doc.qt.io/qt-5/qbarcategoryaxis.html

Parameters : QObject *

Parent Class : QAbstractAxis

- void append(QStringList categories)
- void append_2(QString category)
- QString at(int index)
- QStringList categories(void)
- void clear(void)
- int count(void)
- void insert(int index, QString category)
- QString max(void)
- QString min(void)
- void remove(QString category)
- void replace(QString oldCategory, QString newCategory)
- void setCategories(QStringList categories)
- void setMax(QString max)

- void setMin(QString min)
- void setRange(QString minCategory, QString maxCategory)
- void setcategoriesChangedEvent(const char *)
- void setcountChangedEvent(const char *)
- void setmaxChangedEvent(const char *)
- void setminChangedEvent(const char *)
- void setrangeChangedEvent(const char *)
- const char *getcategoriesChangedEvent(void)
- const char *getcountChangedEvent(void)
- const char *getmaxChangedEvent(void)
- const char *getminChangedEvent(void)
- const char *getrangeChangedEvent(void)

91.25 QBarLegendMarker Class

C++ Reference : http://doc.qt.io/qt-5/qbarlegendmarker.html

Parent Class : QLegendMarker

- QBarSet * barset(void)
- QAbstractBarSeries * series(void)

91.26 QBarSeries Class

C++ Reference : http://doc.qt.io/qt-5/qbarseries.html

Parameters : QObject *

Parent Class : QAbstractBarSeries

- QAbstractSeries::SeriesType type(void)

91.27 QBarSet Class

C++ Reference : http://doc.qt.io/qt-5/qbarset.html

Parameters : QString,QObject *

Parent Class : QObject

- void append(qreal value)
- void append_2(QList<qreal> values)
- qreal at(int index)
- QColor borderColor(void)
- QBrush brush(void)

- QColor color(void)
- int count(void)
- void insert(int index, qreal value)
- QString label(void)
- QBrush labelBrush(void)
- QColor labelColor(void)
- QFont labelFont(void)
- QPen pen(void)
- void remove(int index, int count)
- void replace(int index, qreal value)
- void setBorderColor(QColor color)
- void setBrush(QBrush brush)
- void setColor(QColor color)
- void setLabel(QString label)
- void setLabelBrush(QBrush brush)
- void setLabelColor(QColor color)
- void setLabelFont(QFont font)
- void setPen(QPen pen)
- qreal sum(void)
- void setborderColorChangedEvent(const char *)
- void setbrushChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setcolorChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setlabelBrushChangedEvent(const char *)
- void setlabelChangedEvent(const char *)
- void setlabelColorChangedEvent(const char *)
- void setlabelFontChangedEvent(const char *)
- void setpenChangedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void setvalueChangedEvent(const char *)
- void setvaluesAddedEvent(const char *)
- void setvaluesRemovedEvent(const char *)
- const char *getborderColorChangedEvent(void)

- const char *getbrushChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getcolorChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getlabelBrushChangedEvent(void)
- const char *getlabelChangedEvent(void)
- const char *getlabelColorChangedEvent(void)
- const char *getlabelFontChangedEvent(void)
- const char *getpenChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *getvalueChangedEvent(void)
- const char *getvaluesAddedEvent(void)
- const char *getvaluesRemovedEvent(void)

91.28 QBitmap Class

C++ Reference : http://doc.qt.io/qt-5/qbitmap.html

Parameters : void

Parent Class : QPixmap

- void clear(void)
- void swap(QBitmap)
- QBitmap transformed(QTransform)
- QBitmap fromData(QSize, const uchar * bits, QImage::Format monoFormat)
- QBitmap fromImage(QImage, Qt::ImageConversionFlags flags)

91.29 QBluetoothAddress Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothaddress.html

Parameters : void

- void clear(void)
- bool isNull(void)
- QString toString(void)
- quint64 toUInt64(void)

91.30 QBluetoothDeviceDiscoveryAgent Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothdevicediscoveryagent.html

Parameters : QObject *

- QList<QBluetoothDeviceInfo> discoveredDevices(void)
- QBluetoothDeviceDiscoveryAgent::Error error(void)
- QString errorString(void)
- QBluetoothDeviceDiscoveryAgent::InquiryType inquiryType(void)
- bool isActive(void)
- void setInquiryType(QBluetoothDeviceDiscoveryAgent::InquiryType type)
- void start(void)
- void stop(void)
- void setcanceledEvent(const char *)
- void setdeviceDiscoveredEvent(const char *)
- void seterrorEvent(const char *)
- void setfinishedEvent(const char *)
- const char *getcanceledEvent(void)
- const char *getdeviceDiscoveredEvent(void)
- const char *geterrorEvent(void)
- const char *getfinishedEvent(void)

91.31 QBluetoothDeviceInfo Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothdeviceinfo.html

Parameters : void

- QBluetoothAddress address(void)
- bool isValid(void)
- QBluetoothDeviceInfo::MajorDeviceClass majorDeviceClass(void)
- quint8 minorDeviceClass(void)
- QString name(void)
- qint16 rssi(void)
- QBluetoothDeviceInfo::ServiceClasses serviceClasses(void)
- QList<QBluetoothUuid> serviceUuids(QBluetoothDeviceInfo::DataCompleteness *completeness)
- QBluetoothDeviceInfo::DataCompleteness serviceUuidsCompleteness(void)
- void setCached(bool cached)
- void setServiceUuids(QList<QBluetoothUuid> uuids, QBluetoothDeviceInfo::DataCompleteness completeness)

91.32 QBluetoothHostInfo Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothhostinfo.html

Parameters : void

- QBluetoothAddress address(void)
- QString name(void)
- void setAddress(QBluetoothAddress address)
- void setName(QString name)

91.33 QBluetoothLocalDevice Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothlocaldevice.html

Parameters : QObject *

- QBluetoothAddress address(void)
- QBluetoothLocalDevice::HostMode hostMode(void)
- bool isValid(void)
- QString name(void)
- QBluetoothLocalDevice::Pairing pairingStatus(QBluetoothAddress address)
- void powerOn(void)
- void requestPairing(QBluetoothAddress address, QBluetoothLocalDevice::Pairing pairing)
- void setHostMode(QBluetoothLocalDevice::HostMode mode)
- void pairingConfirmation(bool accept)
- QList<QBluetoothHostInfo> allDevices(void)
- void setdeviceConnectedEvent(const char *)
- void setdeviceDisconnectedEvent(const char *)
- void seterrorEvent(const char *)
- void sethostModeStateChangedEvent(const char *)
- void setpairingDisplayConfirmationEvent(const char *)
- void setpairingDisplayPinCodeEvent(const char *)
- void setpairingFinishedEvent(const char *)
- const char *getdeviceConnectedEvent(void)
- const char *getdeviceDisconnectedEvent(void)
- const char *geterrorEvent(void)
- const char *gethostModeStateChangedEvent(void)
- const char *getpairingDisplayConfirmationEvent(void)
- const char *getpairingDisplayPinCodeEvent(void)
- const char *getpairingFinishedEvent(void)

91.34 QBluetoothServer Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothserver.html

Parameters : QBluetoothServiceInfo::Protocol,QObject *

Parent Class : QObject

- void close(void)
- QBluetoothServer::Error error(void)
- bool hasPendingConnections(void)
- bool isListening(void)
- bool listen(QBluetoothAddress address, quint16 port)
- QBluetoothServiceInfo listen_2(QBluetoothUuid uuid, QString serviceName))
- int maxPendingConnections(void)
- QBluetoothSocket * nextPendingConnection(void)
- QBluetooth::SecurityFlags securityFlags(void)
- QBluetoothAddress serverAddress(void)
- quint16 serverPort(void)
- QBluetoothServiceInfo::Protocol serverType(void)
- void setMaxPendingConnections(int numConnections)
- void setSecurityFlags(QBluetooth::SecurityFlags security)
- void seterrorEvent(const char *)
- void setnewConnectionEvent(const char *)
- const char *geterrorEvent(void)
- const char *getnewConnectionEvent(void)

91.35 QBluetoothServiceDiscoveryAgent Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothservicediscoveryagent.html

Parameters : QObject *

Parent Class : QObject

- QList<QBluetoothServiceInfo> discoveredServices(void)
- QBluetoothServiceDiscoveryAgent::Error error(void)
- QString errorString(void)
- bool isActive(void)
- QBluetoothAddress remoteAddress(void)
- bool setRemoteAddress(QBluetoothAddress address)
- void setUuidFilter(QList<QBluetoothUuid> uuids)
- void setUuidFilter_2(QBluetoothUuid uuid)

- QList<QBluetoothUuid> uuidFilter(void)
- void clear(void)
- void start(QBluetoothServiceDiscoveryAgent::DiscoveryMode mode)
- void stop(void)
- void setcanceledEvent(const char *)
- void seterrorEvent(const char *)
- void setfinishedEvent(const char *)
- void setserviceDiscoveredEvent(const char *)
- const char *getcanceledEvent(void)
- const char *geterrorEvent(void)
- const char *getfinishedEvent(void)
- const char *getserviceDiscoveredEvent(void)

91.36 QBluetoothServiceInfo Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothserviceinfo.html

Parameters : void

- QVariant attribute(quint16 attributeId)
- QList<quint16> attributes(void)
- bool contains(quint16 attributeId)
- QBluetoothDeviceInfo device(void)
- bool isComplete(void)
- bool isRegistered(void)
- bool isValid(void)
- QBluetoothServiceInfo::Sequence protocolDescriptor(QBluetoothUuid::ProtocolUuid protocol)
- int protocolServiceMultiplexer(void)
- bool registerService(QBluetoothAddress localAdapter))
- void removeAttribute(quint16 attributeId)
- int serverChannel(void)
- quint8 serviceAvailability(void)
- QList<QBluetoothUuid> serviceClassUuids(void)
- QString serviceDescription(void)
- QString serviceName(void)
- QString serviceProvider(void)
- QBluetoothUuid serviceUuid(void)
- void setAttribute(quint16 attributeId, QVariant value)

- void setAttribute_2(quint16 attributeId, QBluetoothUuid value)
- void setAttribute_3(quint16 attributeId, QBluetoothServiceInfo::Sequence value)
- void setDevice(QBluetoothDeviceInfo device)
- void setServiceAvailability(quint8 availability)
- void setServiceDescription(QString description)
- void setServiceName(QString name)
- void setServiceProvider(QString provider)
- void setServiceUuid(QBluetoothUuid uuid)
- bool unregisterService(void)

91.37 QBluetoothSocket Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothsocket.html

Parameters : QBluetoothServiceInfo::Protocol,QObject *

Parent Class : QIODevice

- void abort(void)
- void connectToService(QBluetoothServiceInfo service, QIODevice::OpenMode openMode)
- void connectToService_2(QBluetoothAddress address, QBluetoothUuid uuid, QIODevice::OpenMode openMode)
- void connectToService_3(QBluetoothAddress address, quint16 port, QIODevice::OpenMode openMode)
- void disconnectFromService(void)
- QBluetoothSocket::SocketError error(void)
- QString errorString(void)
- QBluetoothAddress localAddress(void)
- QString localName(void)
- quint16 localPort(void)
- QBluetoothAddress peerAddress(void)
- QString peerName(void)
- quint16 peerPort(void)
- bool setSocketDescriptor(int socketDescriptor, QBluetoothServiceInfo::Protocol socketType, QBluetoothSocket::SocketState socketState, QIODevice::OpenMode openMode)
- int socketDescriptor(void)
- QBluetoothServiceInfo::Protocol socketType(void)
- QBluetoothSocket::SocketState state(void)
- void setconnectedEvent(const char *)
- void setdisconnectedEvent(const char *)
- void seterrorEvent(const char *)

- void setstateChangedEvent(const char *)
- const char *getconnectedEvent(void)
- const char *getdisconnectedEvent(void)
- const char *geterrorEvent(void)
- const char *getstateChangedEvent(void)

91.38 QBluetoothTransferManager Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothtransfermanager.html

Parameters : QObject *

Parent Class : QObject

- QBluetoothTransferReply * put(QBluetoothTransferRequest request, QIODevice *data)
- void setfinishedEvent(const char *)
- const char *getfinishedEvent(void)

91.39 QBluetoothTransferReply Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothtransferreply.html

Parameters : QObject *

Parent Class : QObject

- QBluetoothTransferManager * manager(void)
- QBluetoothTransferRequest request(void)
- void abort(void)
- void seterrorEvent(const char *)
- void setfinishedEvent(const char *)
- void settransferProgressEvent(const char *)
- const char *geterrorEvent(void)
- const char *getfinishedEvent(void)
- const char *gettransferProgressEvent(void)

91.40 QBluetoothTransferRequest Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothtransferrequest.html

Parameters : QBluetoothAddress

- QBluetoothAddress address(void)
- QVariant attribute(QBluetoothTransferRequest::Attribute code, QVariant defaultValue))
- void setAttribute(QBluetoothTransferRequest::Attribute code, QVariant value)

91.41 QBluetoothUuid Class

C++ Reference : http://doc.qt.io/qt-5/qbluetoothuuid.html

Parameters : void

Parent Class : QUuid

- int minimumSize(void)
- quint16 toUInt16(bool *ok)
- quint32 toUInt32(bool *ok)
- quint128 toUInt128(void)

91.42 QBoxLayout Class

C++ Reference : http://doc.qt.io/qt-5/qboxlayout.html

Parameters : QBoxLayout::Direction dir, QWidget *parent

Parent Class : QLayout

- void addLayout(QLayout * layout, int stretch)
- void addSpacerItem(QSpacerItem * spacerItem)
- void addSpacing(int size)
- void addStretch(int stretch)
- void addStrut(int size)
- void addWidget(QWidget * widget, int stretch , Qt::Alignment alignment)
- QBoxLayout::Direction direction(void)
- void insertLayout(int index, QLayout * layout, int stretch)
- void insertSpacerItem(int index, QSpacerItem * spacerItem)
- void insertSpacing(int index, int size)
- void insertStretch(int index, int stretch)
- void insertWidget(int index, QWidget * widget, int stretch , Qt::Alignment alignment)
- void setDirection(QBoxLayout::Direction direction)
- void setSpacing(int spacing)

- void setStretch(int index, int stretch)
- bool setStretchFactor(QWidget * widget, int stretch)
- bool setStretchFactor_2(QLayout * layout, int stretch)
- int spacing(void)
- int stretch(int index)

91.43 QBoxPlotLegendMarker Class

C++ Reference : http://doc.qt.io/qt-5/qboxplotlegendmarker.html

Parent Class : QLegendMarker

- QBoxPlotSeries * series(void)
- QLegendMarker::LegendMarkerType type(void)

91.44 QBoxPlotSeries Class

C++ Reference : http://doc.qt.io/qt-5/qboxplotseries.html

Parameters : QObject *

Parent Class : QAbstractSeries

- bool append(QBoxSet *set)
- bool append_2(QList<QBoxSet *> sets)
- bool boxOutlineVisible(void)
- QList<QBoxSet *> boxSets(void)
- qreal boxWidth(void)
- QBrush brush(void)
- void clear(void)
- int count(void)
- bool insert(int index, QBoxSet *set)
- QPen pen(void)
- bool remove(QBoxSet *set)
- void setBoxOutlineVisible(bool visible)
- void setBoxWidth(qreal width)
- void setBrush(QBrush brush)
- void setPen(QPen pen)
- bool take(QBoxSet *set)
- QAbstractSeries::SeriesType type(void)
- void setboxOutlineVisibilityChangedEvent(const char *)
- void setboxWidthChangedEvent(const char *)

- void setboxsetsAddedEvent(const char *)
- void setboxsetsRemovedEvent(const char *)
- void setbrushChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setcountChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setpenChangedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- const char *getboxOutlineVisibilityChangedEvent(void)
- const char *getboxWidthChangedEvent(void)
- const char *getboxsetsAddedEvent(void)
- const char *getboxsetsRemovedEvent(void)
- const char *getbrushChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getcountChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getpenChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)

91.45 QBoxSet Class

C++ Reference : http://doc.qt.io/qt-5/qboxset.html

Parameters : QString,QObject *

Parent Class : QObject

- void append(qreal value)
- void append_2(QList<qreal> values)
- qreal at(int index)
- QBrush brush(void)
- void clear(void)
- int count(void)
- QString label(void)
- QPen pen(void)

- void setBrush(QBrush brush)
- void setLabel(QString label)
- void setPen(QPen pen)
- void setValue(int index, qreal value)
- void setbrushChangedEvent(const char *)
- void setclearedEvent(const char *)
- void setclickedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setpenChangedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void setvalueChangedEvent(const char *)
- void setvaluesChangedEvent(const char *)
- const char *getbrushChangedEvent(void)
- const char *getclearedEvent(void)
- const char *getclickedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getpenChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *getvalueChangedEvent(void)
- const char *getvaluesChangedEvent(void)

91.46 QBrush Class

C++ Reference : http://doc.qt.io/qt-5/qbrush.html

Parameters : void

- QColor color(void)
- QGradient *gradient(void)
- bool isOpaque(void)
- void setColor(QColor)
- void setStyle(Qt::BrushStyle style)
- void setTexture(QPixmap)
- void setTextureImage(QImage)

- void setTransform(QTransform)
- int style(void)
- void swap(QBrush)
- QPixmap texture(void)
- QImage textureImage(void)
- QTransform transform(void)

91.47 QBuffer Class

C++ Reference : http://doc.qt.io/qt-5/qbuffer.html

Parameters : QObject *

Parent Class : QIODevice

- QByteArray buffer(void)
- QByteArray data(void)
- void setBuffer(QByteArray *byteArray)
- void setData(QByteArray data)
- void setData_2(char *data, int size)

91.48 QButtonGroup Class

C++ Reference : http://doc.qt.io/qt-5/qbuttongroup.html

Parameters : QObject *parent

- void addButton(QAbstractButton *button, int id)
- QAbstractButton *button(int id)
- QAbstractButton *checkedButton(void)
- int checkedId(void)
- bool exclusive(void)
- int id(QAbstractButton *button)
- void removeButton(QAbstractButton *button)
- void setExclusive(bool)
- void setId(QAbstractButton *button, int id)
- void setbuttonClickedEvent(const char *)
- void setbuttonPressedEvent(const char *)
- void setbuttonReleasedEvent(const char *)
- const char *getbuttonClickedEvent(void)
- const char *getbuttonPressedEvent(void)
- const char *getbuttonReleasedEvent(void)

91.49 QByteArray Class

C++ Reference : http://doc.qt.io/qt-5/qbytearray.html

Parameters : void

- QByteArray append(const char *str)
- QByteArray append_2(const char *str,int size)
- char at(int i)
- int capacity(void)
- void chop(int n)
- void clear(void)
- const char *constData(void)
- bool contains(const char *str)
- int count(const char *str)
- const char *data(void)
- bool endsWith(const char *str)
- QByteArray fill(char ch, int size)
- int indexOf(const char *str, int from)
- QByteArray insert(int i, const char *str, int len)
- bool isEmpty(void)
- bool isNull(void)
- int lastIndexOf(const char *str, int from)
- QByteArray left(int len)
- QByteArray leftJustified(int width, char fill, bool truncate)
- int length(void)
- QByteArray mid(int pos, int len)
- QByteArray prepend(const char *str, int len)
- void push_back(const char *str)
- void push_front(const char *str)
- QByteArray remove(int pos, int len)
- QByteArray repeated(int times)
- QByteArray replace(int pos, int len, const char *after, int alen)
- QByteArray replace_2(int pos, int len, QByteArray after)
- QByteArray replace_3(int pos, int len, const char *after)
- QByteArray replace_4(char before, const char *after)
- QByteArray replace_5(char before, QByteArray after)
- QByteArray replace_6(const char *before, const char *after)

- QByteArray replace_7(const char *before, int bsize, const char *after, int asize)
- QByteArray replace_8(const QByteArray before, QByteArray after)
- QByteArray replace_9(const QByteArray before, const char *after)
- QByteArray replace_10(const char *before, QByteArray after)
- QByteArray replace_11(char before, char after)
- void reserve(int size)
- void resize(int size)
- QByteArray right(int len)
- QByteArray rightJustified(int width, char fill, bool truncate)
- QByteArray setNum(int n, int base)
- QByteArray setRawData(const char *data, uint size)
- QByteArray simplified(void)
- int size(void)
- void squeeze(void)
- bool startsWith(const char *str)
- void swap(QByteArray other)
- QByteArray toBase64(void)
- double toDouble(bool * ok)
- float toFloat(bool * ok)
- QByteArray toHex(void)
- int toInt(bool *ok, int base)
- long toLong(bool *ok, int base)
- qlonglong toLongLong(bool *ok, int base)
- QByteArray toLower(void)
- QByteArray toPercentEncoding(QByteArray,QByteArray, char percent)
- short toShort(bool *ok, int base)
- int toUInt(bool *ok, int base)
- int toULong(bool *ok, int base)
- int toULongLong(bool * ok, int base)
- int toUShort(bool * ok, int base)
- QByteArray toUpper(void)
- QByteArray trimmed(void)
- void truncate(int pos)
- QByteArray fromBase64(QByteArray)
- QByteArray fromHex(QByteArray)
- QByteArray fromPercentEncoding(QByteArray, char percent)

- QByteArray fromRawData(const char *data, int size)
- QByteArray number(int n, int base)

91.50 QCalendarWidget Class

C++ Reference : http://doc.qt.io/qt-5/qcalendarwidget.html

Parameters : QWidget *

Parent Class : QWidget

- int dateEditAcceptDelay(void)
- QMap<QDate, QTextCharFormat> dateTextFormat(void)
- QTextCharFormat dateTextFormat_2(QDate date)
- Qt::DayOfWeek firstDayOfWeek(void)
- QTextCharFormat headerTextFormat(void)
- QCalendarWidget::HorizontalHeaderFormat horizontalHeaderFormat(void)
- bool isDateEditEnabled(void)
- bool isGridVisible(void)
- bool isNavigationBarVisible(void)
- QDate maximumDate(void)
- QDate minimumDate(void)
- int monthShown(void)
- QDate selectedDate(void)
- QCalendarWidget::SelectionMode selectionMode(void)
- void setDateEditAcceptDelay(int delay)
- void setDateEditEnabled(bool enable)
- void setDateTextFormat(QDate date, QTextCharFormat format)
- void setFirstDayOfWeek(Qt::DayOfWeek dayOfWeek)
- void setHeaderTextFormat(QTextCharFormat format)
- void setHorizontalHeaderFormat(QCalendarWidget::HorizontalHeaderFormat format)
- void setMaximumDate(QDate date)
- void setMinimumDate(QDate date)
- void setSelectionMode(QCalendarWidget::SelectionMode mode)
- void setVerticalHeaderFormat(QCalendarWidget::VerticalHeaderFormat format)
- void setWeekdayTextFormat(Qt::DayOfWeek dayOfWeek, QTextCharFormat format)
- QCalendarWidget::VerticalHeaderFormat verticalHeaderFormat(void)
- QTextCharFormat weekdayTextFormat(Qt::DayOfWeek dayOfWeek)
- int yearShown(void)

- void setCurrentPage(int year, int month)
- void setDateRange(QDate min, QDate max)
- void setGridVisible(bool show)
- void setNavigationBarVisible(bool visible)
- void setSelectedDate(QDate date)
- void showNextMonth(void)
- void showNextYear(void)
- void showPreviousMonth(void)
- void showPreviousYear(void)
- void showSelectedDate(void)
- void showToday(void)
- void setactivatedEvent(const char *)
- void setclickedEvent(const char *)
- void setcurrentPageChangedEvent(const char *)
- void setselectionChangedEvent(const char *)
- const char *getactivatedEvent(void)
- const char *getclickedEvent(void)
- const char *getcurrentPageChangedEvent(void)
- const char *getselectionChangedEvent(void)

91.51 QCamera Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qcamera.html

Parameters : void

Parent Class : QMediaObject

- QCamera::CaptureModes captureMode(void)
- QCamera::Error error(void)
- QString errorString(void)
- QCameraExposure * exposure(void)
- QCameraFocus * focus(void)
- QCameraImageProcessing * imageProcessing(void)
- bool isCaptureModeSupported(QCamera::CaptureModes mode)
- QCamera::LockStatus lockStatus(void)
- QCamera::LockStatus lockStatus_2(QCamera::LockType lockType)
- QCamera::LockTypes requestedLocks(void)
- void setViewfinder(QVideoWidget * viewfinder)

- void setViewfinder_2(QGraphicsVideoItem * viewfinder)
- void setViewfinder_3(QAbstractVideoSurface * surface)
- QCamera::State state(void)
- QCamera::Status status(void)
- QCamera::LockTypes supportedLocks(void)
- void load(void)
- void searchAndLock(void)
- void searchAndLock_2(QCamera::LockTypes locks)
- void setCaptureMode(QCamera::CaptureModes mode)
- void start(void)
- void stop(void)
- void unload(void)
- void unlock(void)
- void unlock_2(QCamera::LockTypes locks)

91.52 QCameraImageCapture Class

C++ Reference : http://doc.qt.io/qt-5/qcameraimagecapture.html

Parameters : QMediaObject * mediaObject

- QMultimedia::AvailabilityStatus availability(void)
- QVideoFrame::PixelFormat bufferFormat(void)
- QCameraImageCapture::CaptureDestinations captureDestination(void)
- QImageEncoderSettings encodingSettings(void)
- QCameraImageCapture::Error error(void)
- QString errorString(void)
- QString imageCodecDescription(QString codec)
- bool isAvailable(void)
- bool isCaptureDestinationSupported(QCameraImageCapture::CaptureDestinations destination)
- bool isReadyForCapture(void)
- void setBufferFormat(QVideoFrame::PixelFormat format)
- void setCaptureDestination(QCameraImageCapture::CaptureDestinations destination)
- void setEncodingSettings(QImageEncoderSettings settings)
- QList<QVideoFrame::PixelFormat> supportedBufferFormats(void)
- QStringList supportedImageCodecs(void)
- QList<QSize> supportedResolutions(QImageEncoderSettings settings , bool * continuous)
- void cancelCapture(void)

- int capture(QString file)

91.53 QCameraLens Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qcameralens.html

Parameters : Qt3DCore::QNode *

- float aspectRatio(void)
- float bottom(void)
- float exposure(void)
- float farPlane(void)
- float fieldOfView(void)
- float left(void)
- float nearPlane(void)
- QMatrix4x4 projectionMatrix(void)
- Qt3DRender::QCameraLens::ProjectionType projectionType(void)
- float right(void)
- void setFrustumProjection(float left, float right, float bottom, float top, float nearPlane, float farPlane)
- void setOrthographicProjection(float left, float right, float bottom, float top, float nearPlane, float farPlane)
- void setPerspectiveProjection(float fieldOfView, float aspectRatio, float nearPlane, float farPlane)
- float top(void)
- void setBottom(float bottom)
- void setExposure(float exposure)
- void setFarPlane(float farPlane)
- void setFieldOfView(float fieldOfView)
- void setLeft(float left)
- void setNearPlane(float nearPlane)
- void setProjectionMatrix(QMatrix4x4 projectionMatrix)
- void setProjectionType(Qt3DRender::QCameraLens::ProjectionType projectionType)
- void setRight(float right)
- void setTop(float top)

91.54 QCameraSelector Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qcameraselector .html

Parameters : Qt3DCore::QNode *

- Qt3DCore::QEntity * camera(void)
- void setCamera(Qt3DCore::QEntity *camera)

91.55 QCameraViewfinder Class

C++ Reference : http://doc.qt.io/qt-5/qcameraviewfinder.html

Parameters : QWidget *

Parent Class : QVideoWidget

91.56 QCandlestickLegendMarker Class

C++ Reference : http://doc.qt.io/qt-5/qcandlesticklegendmarker.html

Parent Class : QLegendMarker

- QCandlestickSeries * series(void)
- QLegendMarker::LegendMarkerType type(void)

91.57 QCandlestickModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qcandlestickmodelmapper.html

Parameters : QObject *

Parent Class : QObject

- QAbstractItemModel * model(void)
- Qt::Orientation orientation(void)
- QCandlestickSeries * series(void)
- void setModel(QAbstractItemModel *model)
- void setSeries(QCandlestickSeries *series)
- void setmodelReplacedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- const char *getmodelReplacedEvent(void)
- const char *getseriesReplacedEvent(void)

91.58 QCandlestickSeries Class

C++ Reference : http://doc.qt.io/qt-5/qcandlestickseries.html

Parameters : QObject *

Parent Class : QAbstractSeries

- bool append(QCandlestickSet *set)
- bool append_2(QList<QCandlestickSet *> sets)
- bool bodyOutlineVisible(void)
- qreal bodyWidth(void)
- QBrush brush(void)
- bool capsVisible(void)
- qreal capsWidth(void)
- void clear(void)
- int count(void)
- QColor decreasingColor(void)
- QColor increasingColor(void)
- bool insert(int index, QCandlestickSet *set)
- qreal maximumColumnWidth(void)
- qreal minimumColumnWidth(void)
- QPen pen(void)
- bool remove(QCandlestickSet *set)
- bool remove_2(QList<QCandlestickSet *> sets)
- void setBodyOutlineVisible(bool bodyOutlineVisible)
- void setBodyWidth(qreal bodyWidth)
- void setBrush(QBrush brush)
- void setCapsVisible(bool capsVisible)
- void setCapsWidth(qreal capsWidth)
- void setDecreasingColor(QColor decreasingColor)
- void setIncreasingColor(QColor increasingColor)
- void setMaximumColumnWidth(qreal maximumColumnWidth)
- void setMinimumColumnWidth(qreal minimumColumnWidth)
- void setPen(QPen pen)
- QList<QCandlestickSet *> sets(void)
- bool take(QCandlestickSet *set)
- void setbodyOutlineVisibilityChangedEvent(const char *)
- void setbodyWidthChangedEvent(const char *)

- void setbrushChangedEvent(const char *)
- void setcandlestickSetsAddedEvent(const char *)
- void setcandlestickSetsRemovedEvent(const char *)
- void setcapsVisibilityChangedEvent(const char *)
- void setcapsWidthChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setcountChangedEvent(const char *)
- void setdecreasingColorChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setincreasingColorChangedEvent(const char *)
- void setmaximumColumnWidthChangedEvent(const char *)
- void setminimumColumnWidthChangedEvent(const char *)
- void setpenChangedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- const char *getbodyOutlineVisibilityChangedEvent(void)
- const char *getbodyWidthChangedEvent(void)
- const char *getbrushChangedEvent(void)
- const char *getcandlestickSetsAddedEvent(void)
- const char *getcandlestickSetsRemovedEvent(void)
- const char *getcapsVisibilityChangedEvent(void)
- const char *getcapsWidthChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getcountChangedEvent(void)
- const char *getdecreasingColorChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getincreasingColorChangedEvent(void)
- const char *getmaximumColumnWidthChangedEvent(void)
- const char *getminimumColumnWidthChangedEvent(void)
- const char *getpenChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)

91.59 QCandlestickSet Class

C++ Reference : http://doc.qt.io/qt-5/qcandlestickset.html

Parameters : qreal,QObject *

Parent Class : QObject

- QBrush brush(void)
- qreal close(void)
- qreal high(void)
- qreal low(void)
- qreal open(void)
- QPen pen(void)
- void setBrush(QBrush brush)
- void setClose(qreal close)
- void setHigh(qreal high)
- void setLow(qreal low)
- void setOpen(qreal open)
- void setPen(QPen pen)
- void setTimestamp(qreal timestamp)
- qreal timestamp(void)
- void setbrushChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setcloseChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethighChangedEvent(const char *)
- void sethoveredEvent(const char *)
- void setlowChangedEvent(const char *)
- void setopenChangedEvent(const char *)
- void setpenChangedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void settimestampChangedEvent(const char *)
- const char *getbrushChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getcloseChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethighChangedEvent(void)

- const char *gethoveredEvent(void)
- const char *getlowChangedEvent(void)
- const char *getopenChangedEvent(void)
- const char *getpenChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *gettimestampChangedEvent(void)

91.60 QCategoryAxis Class

C++ Reference : http://doc.qt.io/qt-5/qcategoryaxis.html

Parameters : QObject *

Parent Class : QValueAxis

- void append(QString categoryLabel, qreal categoryEndValue)
- QStringList categoriesLabels(void)
- int count(void)
- qreal endValue(QString categoryLabel)
- QCategoryAxis::AxisLabelsPosition labelsPosition(void)
- void remove(QString categoryLabel)
- void replaceLabel(QString oldLabel, QString newLabel)
- void setLabelsPosition(QCategoryAxis::AxisLabelsPosition position)
- void setStartValue(qreal min)
- qreal startValue(QString categoryLabel)
- void setcategoriesChangedEvent(const char *)
- void setlabelsPositionChangedEvent(const char *)
- const char *getcategoriesChangedEvent(void)
- const char *getlabelsPositionChangedEvent(void)

91.61 QChar Class

C++ Reference : http://doc.qt.io/qt-5/qchar.html

Parameters : int

- QChar::Category category(void)
- uchar cell(void)
- unsigned char combiningClass(void)
- QString decomposition(void)
- QChar::Decomposition decompositionTag(void)

- int digitValue(void)
- QChar::Direction direction(void)
- bool hasMirrored(void)
- bool isDigit(void)
- bool isHighSurrogate(void)
- bool isLetter(void)
- bool isLetterOrNumber(void)
- bool isLowSurrogate(void)
- bool isLower(void)
- bool isMark(void)
- bool isNonCharacter(void)
- bool isNull(void)
- bool isNumber(void)
- bool isPrint(void)
- bool isPunct(void)
- bool isSpace(void)
- bool isSurrogate(void)
- bool isSymbol(void)
- bool isTitleCase(void)
- bool isUpper(void)
- QChar mirroredChar(void)
- uchar row(void)
- QChar::Script script(void)
- QChar toCaseFolded(void)
- char toLatin1(void)
- QChar toLower(void)
- QChar toTitleCase(void)
- QChar toUpper(void)
- ushort unicode(void)
- ushort unicode_2(void)
- QChar::UnicodeVersion unicodeVersion(void)
- QChar::Category category_2(uint ucs4)
- unsigned char combiningClass_2(uint ucs4)
- QChar::UnicodeVersion currentUnicodeVersion(void)
- QString decomposition_2(uint ucs4)
- QChar::Decomposition decompositionTag_2(uint ucs4)

- int digitValue_2(uint ucs4)
- QChar::Direction direction_2(uint ucs4)
- QChar fromLatin1(char c)
- bool hasMirrored_2(uint ucs4)
- ushort highSurrogate(uint ucs4)
- bool isDigit_2(uint ucs4)
- bool isHighSurrogate_2(uint ucs4)
- bool isLetter_2(uint ucs4)
- bool isLetterOrNumber_2(uint ucs4)
- bool isLowSurrogate_2(uint ucs4)
- bool isLower_2(uint ucs4)
- bool isMark_2(uint ucs4)
- bool isNonCharacter_2(uint ucs4)
- bool isNumber_2(uint ucs4)
- bool isPrint_2(uint ucs4)
- bool isPunct_2(uint ucs4)
- bool isSpace_2(uint ucs4)
- bool isSurrogate_2(uint ucs4)
- bool isSymbol_2(uint ucs4)
- bool isTitleCase_2(uint ucs4)
- bool isUpper_2(uint ucs4)
- ushort lowSurrogate(uint ucs4)
- uint mirroredChar_2(uint ucs4)
- bool requiresSurrogates(uint ucs4)
- QChar::Script script_2(uint ucs4)
- uint surrogateToUcs4(ushort high, ushort low)
- uint surrogateToUcs4_2(QChar high, QChar low)
- uint toCaseFolded_2(uint ucs4)
- uint toLower_2(uint ucs4)
- uint toTitleCase_2(uint ucs4)
- uint toUpper_2(uint ucs4)
- QChar::UnicodeVersion unicodeVersion_2(uint ucs4)

91.62 QChart Class

C++ Reference : http://doc.qt.io/qt-5/qchart.html

Parameters : QGraphicsItem *,Qt::WindowFlags

Parent Class : QGraphicsWidget

- void addAxis(QAbstractAxis *axis, Qt::Alignment alignment)
- void addSeries(QAbstractSeries *series)
- int animationDuration(void)
- QEasingCurve animationEasingCurve(void)
- QChart::AnimationOptions animationOptions(void)
- QList<QAbstractAxis *> axes(Qt::Orientations orientation, QAbstractSeries *series)
- QBrush backgroundBrush(void)
- QPen backgroundPen(void)
- qreal backgroundRoundness(void)
- QChart::ChartType chartType(void)
- void createDefaultAxes(void)
- bool isBackgroundVisible(void)
- bool isDropShadowEnabled(void)
- bool isPlotAreaBackgroundVisible(void)
- bool isZoomed(void)
- QLegend *legend(void)
- QLocale locale(void)
- bool localizeNumbers(void)
- QPointF mapToPosition(QPointF value, QAbstractSeries *series)
- QPointF mapToValue(QPointF position, QAbstractSeries *series)
- QMargins margins(void)
- QRectF plotArea(void)
- QBrush plotAreaBackgroundBrush(void)
- QPen plotAreaBackgroundPen(void)
- void removeAllSeries(void)
- void removeAxis(QAbstractAxis *axis)
- void removeSeries(QAbstractSeries *series)
- void scroll(qreal dx, qreal dy)
- QList<QAbstractSeries *> series(void)
- void setAnimationDuration(int msecs)
- void setAnimationEasingCurve(QEasingCurve curve)

- void setAnimationOptions(QChart::AnimationOptions options)
- void setBackgroundBrush(QBrush brush)
- void setBackgroundPen(QPen pen)
- void setBackgroundRoundness(qreal diameter)
- void setBackgroundVisible(bool visible)
- void setDropShadowEnabled(bool enabled)
- void setLocale(QLocale locale)
- void setLocalizeNumbers(bool localize)
- void setMargins(QMargins margins)
- void setPlotArea(QRectF rect)
- void setPlotAreaBackgroundBrush(QBrush brush)
- void setPlotAreaBackgroundPen(QPen pen)
- void setPlotAreaBackgroundVisible(bool visible)
- void setTheme(QChart::ChartTheme theme)
- void setTitle(QString title)
- void setTitleBrush(QBrush brush)
- void setTitleFont(QFont font)
- QChart::ChartTheme theme(void)
- QString title(void)
- QBrush titleBrush(void)
- QFont titleFont(void)
- void zoom(qreal factor)
- void zoomIn(void)
- void zoomIn_2(QRectF rect)
- void zoomOut(void)
- void zoomReset(void)
- void setplotAreaChangedEvent(const char *)
- const char *getplotAreaChangedEvent(void)

91.63 QChartView Class

C++ Reference : http://doc.qt.io/qt-5/qchartview.html

Parameters : QWidget *

Parent Class : QGraphicsView

- QChart * chart(void)
- QChartView::RubberBands rubberBand(void)

- void setChart(QChart *chart)
- void setRubberBand(QChartView::RubberBands rubberBand)

91.64 QCheckBox Class

C++ Reference : http://doc.qt.io/qt-5/qcheckbox.html

Parameters : QWidget *parent

Parent Class : QAbstractButton

- int checkState(void)
- bool isTristate(void)
- void setCheckState(Qt::CheckState state)
- void setTristate(bool y)
- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setstateChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void settoggledEvent(const char *)
- const char *getstateChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *gettoggledEvent(void)

91.65 QChildEvent Class

C++ Reference : http://doc.qt.io/qt-5/qchildevent.html

Parameters : QEvent::Type,QObject *

Parent Class : QEvent

- bool added(void)
- QObject *child(void)
- bool polished(void)
- bool removed(void)

91.66 QClipboard Class

C++ Reference : http://doc.qt.io/qt-5/qclipboard.html

- void clear(QClipboard::Mode mode)
- QImage image(QClipboard::Mode mode)
- QMimeData * mimeData(QClipboard::Mode mode)
- bool ownsClipboard(void)
- bool ownsFindBuffer(void)
- bool ownsSelection(void)
- QPixmap pixmap(QClipboard::Mode mode)
- void setImage(QImage image, QClipboard::Mode mode)
- void setMimeData(QMimeData * src, QClipboard::Mode mode)
- void setPixmap(QPixmap pixmap, QClipboard::Mode mode)
- void setText(QString text, QClipboard::Mode mode)
- bool supportsFindBuffer(void)
- bool supportsSelection(void)
- QString text(QClipboard::Mode mode)

91.67 QColor Class

C++ Reference : http://doc.qt.io/qt-5/qcolor.html

Parameters : void

- int alpha(void)
- double alphaF(void)
- int black(void)
- double blackF(void)
- int blue(void)
- double blueF(void)
- QColor convertTo(QColor::Spec colorSpec)
- int cyan(void)
- double cyanF(void)
- QColor darker(int factor)
- void getCmyk(int *c, int *m, int *y, int *k, int *a)
- void getCmykF(qreal *c, qreal *m, qreal *y, qreal *k, qreal *a)
- void getHsl(int *h, int *s, int *l, int *a)
- void getHslF(qreal *h, qreal *s, qreal *l, qreal *a)
- void getHsv(int *h, int *s, int *v, int *a)

- void getHsvF(qreal *h, qreal *s, qreal *v, qreal *a)
- void getRgb(int *r, int *g, int *b, int *a)
- void getRgbF(qreal *r, qreal *g, qreal *b, qreal *a)
- int green(void)
- double greenF(void)
- int hslHue(void)
- double hslHueF(void)
- int hslSaturation(void)
- double hslSaturationF(void)
- int hsvHue(void)
- double hsvHueF(void)
- int hsvSaturation(void)
- double hsvSaturationF(void)
- int hue(void)
- double hueF(void)
- bool isValid(void)
- QColor lighter(int factor)
- int lightness(void)
- double lightnessF(void)
- int magenta(void)
- double magentaF(void)
- QString name(void)
- int red(void)
- double redF(void)
- QRgb rgb(void)
- QRgb rgba(void)
- int saturation(void)
- double saturationF(void)
- void setAlpha(int alpha)
- void setAlphaF(double alpha)
- void setBlue(int blue)
- void setBlueF(double blue)
- void setCmyk(int c, int m, int y, int k, int a)
- void setCmykF(double c, double m, double y, double k, double a)
- void setGreen(int green)
- void setGreenF(double green)

- void setHsl(int h, int s, int l, int a)
- void setHslF(double h, double s, double l, double a)
- void setHsv(int h, int s, int v, int a)
- void setHsvF(double h, double s, double v, double a)
- void setNamedColor(QString)
- void setRed(int red)
- void setRedF(double red)
- void setRgb(int r, int g, int b, int a)
- void setRgbF(double r, double g, double b, double a)
- void setRgba(QRgb rgba)
- int spec(void)
- QColor toCmyk(void)
- QColor toHsl(void)
- QColor toHsv(void)
- QColor toRgb(void)
- int value(void)
- double valueF(void)
- int yellow(void)
- double yellowF(void)
- QStringList colorNames(void)
- QColor fromCmyk(int c, int m, int y, int k, int a)
- QColor fromCmykF(double c, double m, double y, double k, double a)
- QColor fromHsl(int h, int s, int l, int a)
- QColor fromHslF(double h, double s, double l, double a)
- QColor fromHsv(int h, int s, int v, int a)
- QColor fromHsvF(double h, double s, double v, double a)
- QColor fromRgb(int r, int g, int b, int a)
- QColor fromRgbF(double r, double g, double b, double a)
- QColor fromRgba(QRgb rgba)
- bool isValidColor(QString)

91.68 QColorDialog Class

C++ Reference : http://doc.qt.io/qt-5/qcolordialog.html

Parameters : void

Parent Class : QDialog

- QColor currentColor(void)
- void open(QObject * receiver, char * member)
- QColorDialog::ColorDialogOptions options(void)
- QColor selectedColor(void)
- void setCurrentColor(QColor color)
- void setOption(QColorDialog::ColorDialogOption option, bool on)
- void setOptions(QColorDialog::ColorDialogOptions options)
- bool testOption(QColorDialog::ColorDialogOption option)
- QColor customColor(int index)
- int customCount(void)
- QColor getColor_2(QColor initial, QWidget * parent, QString title, QColorDialog::ColorDialogOptions options)
- void setCustomColor(int index, QColor color)
- void setStandardColor(int index, QColor color)
- QColor standardColor(int index)
- void setcolorSelectedEvent(const char *)
- void setcurrentColorChangedEvent(const char *)
- const char *getcolorSelectedEvent(void)
- const char *getcurrentColorChangedEvent(void)
- int getcolor(void)

91.69 QComboBox Class

C++ Reference : http://doc.qt.io/qt-5/qcombobox.html

Parameters : QWidget *

Parent Class : QWidget

- void addItem(QString,int)
- void addItems(QStringList)
- QCompleter *completer(void)
- int count(void)
- int currentIndex(void)
- QString currentText(void)

- bool duplicatesEnabled(void)
- int findData(QVariant, int role, Qt::MatchFlag flags)
- int findText(QString, Qt::MatchFlag flags)
- bool hasFrame(void)
- void hidePopup(void)
- QSize iconSize(void)
- void insertItem(int index, QString, QVariant)
- bool isEditable(void)
- QVariant itemData(int index, int role)
- QAbstractItemDelegate *itemDelegate(void)
- QIcon itemIcon(int index)
- QString itemText(int index)
- QLineEdit *lineEdit(void)
- int maxCount(void)
- int maxVisibleItems(void)
- int minimumContentsLength(void)
- QAbstractItemModel *model(void)
- int modelColumn(void)
- void removeItem(int index)
- QModelIndex rootModelIndex(void)
- void setCompleter(QCompleter *completer)
- void setDuplicatesEnabled(bool enable)
- void setEditable(bool editable)
- void setFrame(bool)
- void setIconSize(QSize)
- void setItemDelegate(QAbstractItemDelegate *delegate)
- void setItemIcon(int index, QIcon)
- void setItemText(int index, QString)
- void setLineEdit(QLineEdit *edit)
- void setMaxCount(int max)
- void setMaxVisibleItems(int maxItems)
- void setMinimumContentsLength(int characters)
- void setModel(QAbstractItemModel *model)
- void setModelColumn(int visibleColumn)
- void setRootModelIndex(QModelIndex)
- void setView(QAbstractItemView *itemView)

- void showPopup(void)
- QAbstractItemView *view(void)
- void clear(void)
- void clearEditText(void)
- void setCurrentIndex(int index)
- void setCurrentText(QString)
- void setEditText(QString)
- void setactivatedEvent(const char *)
- void setcurrentIndexChangedEvent(const char *)
- void seteditTextChangedEvent(const char *)
- void sethighlightedEvent(const char *)
- const char *getactivatedEvent(void)
- const char *getcurrentIndexChangedEvent(void)
- const char *geteditTextChangedEvent(void)
- const char *gethighlightedEvent(void)

91.70 QCompleter Class

C++ Reference : http://doc.qt.io/qt-5/qcompleter.html

Parameters : QObject *parent

Parent Class : QObject

- Qt::CaseSensitivity caseSensitivity(void)
- int completionColumn(void)
- int completionCount(void)
- QCompleter::CompletionMode completionMode(void)
- QAbstractItemModel *completionModel(void)
- QString completionPrefix(void)
- int completionRole(void)
- QString currentCompletion(void)
- QModelIndex currentIndex(void)
- int currentRow(void)
- Qt::MatchFlags filterMode(void)
- int maxVisibleItems(void)
- QAbstractItemModel * model(void)
- QCompleter::ModelSorting modelSorting(void)
- QAbstractItemView * popup(void)

- void setCaseSensitivity(Qt::CaseSensitivity caseSensitivity)
- void setCompletionColumn(int column)
- void setCompletionMode(QCompleter::CompletionMode mode)
- void setCompletionRole(int role)
- bool setCurrentRow(int row)
- void setFilterMode(Qt::MatchFlags filterMode)
- void setMaxVisibleItems(int maxItems)
- void setModel(QAbstractItemModel *model)
- void setModelSorting(QCompleter::ModelSorting sorting)
- void setPopup(QAbstractItemView *popup)
- void setWidget(QWidget *widget)
- QWidget * widget(void)
- bool wrapAround(void)
- void complete(QRect rect)
- void setCompletionPrefix(QString prefix)
- void setWrapAround(bool wrap)

91.71 QCompleter2 Class

Parameters : QAbstractItemModel *model, QObject *parent

Parent Class : QCompleter

91.72 QCompleter3 Class

Parameters : QStringList list, QObject *parent

Parent Class : QCompleter

91.73 QCompleter4 Class

Parameters : QStringList list, QObject *parent

Parent Class : QCompleter

91.74 QConeGeometry Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qconegeometry.html

Parameters : Qt3DCore::QNode *

- float bottomRadius(void)
- bool hasBottomEndcap(void)
- bool hasTopEndcap(void)
- Qt3DRender::QAttribute * indexAttribute(void)
- float length(void)
- Qt3DRender::QAttribute * normalAttribute(void)
- Qt3DRender::QAttribute * positionAttribute(void)
- int rings(void)
- int slices(void)
- Qt3DRender::QAttribute * texCoordAttribute(void)
- float topRadius(void)
- void updateIndices(void)
- void updateVertices(void)
- void setBottomRadius(float bottomRadius)
- void setHasBottomEndcap(bool hasBottomEndcap)
- void setHasTopEndcap(bool hasTopEndcap)
- void setLength(float length)
- void setRings(int rings)
- void setSlices(int slices)
- void setTopRadius(float topRadius)

91.75 QConeMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qconemesh.html

Parameters : Qt3DCore::QNode *

- float bottomRadius(void)
- bool hasBottomEndcap(void)
- bool hasTopEndcap(void)
- float length(void)
- int rings(void)
- int slices(void)
- float topRadius(void)
- void setBottomRadius(float bottomRadius)

- void setHasBottomEndcap(bool hasBottomEndcap)
- void setHasTopEndcap(bool hasTopEndcap)
- void setLength(float length)
- void setRings(int rings)
- void setSlices(int slices)
- void setTopRadius(float topRadius)

91.76 QCoreApplication Class

C++ Reference : http://doc.qt.io/qt-5/qcoreapplication.html

Parent Class : QObject

- void installNativeEventFilter(QAbstractNativeEventFilter *filterObj)
- void removeNativeEventFilter(QAbstractNativeEventFilter *filterObject)
- void quit(void)
- void addLibraryPath(QString path)
- QString applicationDirPath(void)
- QString applicationFilePath(void)
- QString applicationName(void)
- qint64 applicationPid(void)
- QString applicationVersion(void)
- QStringList arguments(void)
- bool closingDown(void)
- QAbstractEventDispatcher * eventDispatcher(void)
- int exec(void)
- void exit(int returnCode)
- bool installTranslator(QTranslator *translationFile)
- QCoreApplication * instance(void)
- bool isQuitLockEnabled(void)
- QStringList libraryPaths(void)
- QString organizationDomain(void)
- QString organizationName(void)
- void postEvent(QObject *receiver, QEvent *event, int priority)
- void processEvents(QEventLoop::ProcessEventsFlags flags)
- void processEvents_2(QEventLoop::ProcessEventsFlags flags, int maxtime)
- void removeLibraryPath(QString path)
- void removePostedEvents(QObject *receiver, int eventType)

- bool removeTranslator(QTranslator *translationFile)
- bool sendEvent(QObject *receiver, QEvent *event)
- void sendPostedEvents(QObject *receiver, int event_type)
- void setApplicationName(QString application)
- void setApplicationVersion(QString version)
- void setAttribute(Qt::ApplicationAttribute attribute, bool on)
- void setEventDispatcher(QAbstractEventDispatcher *eventDispatcher)
- void setLibraryPaths(QStringList paths)
- void setOrganizationDomain(QString orgDomain)
- void setOrganizationName(QString orgName)
- void setQuitLockEnabled(bool enabled)
- bool startingUp(void)
- bool testAttribute(Qt::ApplicationAttribute attribute)
- QString translate(char *context, char *sourceText, char *disambiguation, int n)

91.77 QCuboidMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qcuboidmesh.html

Parameters : Qt3DCore::QNode *

- float xExtent(void)
- QSize xyMeshResolution(void)
- QSize xzMeshResolution(void)
- float yExtent(void)
- QSize yzMeshResolution(void)
- float zExtent(void)
- void setXExtent(float xExtent)
- void setXYMeshResolution(QSize resolution)
- void setXZMeshResolution(QSize resolution)
- void setYExtent(float yExtent)
- void setYZMeshResolution(QSize resolution)
- void setZExtent(float zExtent)

91.78 QCullFace Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qcullface.html

Parameters : Qt3DCore::QNode *

- Qt3DRender::QCullFace::CullingMode mode(void)
- void setMode(Qt3DRender::QCullFace::CullingMode mode)

91.79 QCursor Class

C++ Reference : http://doc.qt.io/qt-5/qcursor.html

Parameters : void

- QBitmap *bitmap(void)
- QBitmap *mask(void)
- QPoint hotSpot(void)
- QPixmap pixmap(void)
- void setShape(Qt::CursorShape shape)
- Qt::CursorShape shape(void)
- QPoint pos(void)
- QPoint pos_2(QScreen *)
- void setPos(int x, int y)
- void setPos_2(QScreen *screen, int x, int y)
- void setPos_3(QPoint)
- void setPos_4(QScreen *screen, QPoint)

91.80 QCylinderMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qcylindermesh.html

Parameters : Qt3DCore::QNode *

- float length(void)
- float radius(void)
- int rings(void)
- int slices(void)
- void setLength(float length)
- void setRadius(float radius)
- void setRings(int rings)
- void setSlices(int slices)

91.81 QDate Class

C++ Reference : http://doc.qt.io/qt-5/qdate.html

Parameters : void

- QDate addDays(int ndays)
- QDate addMonths(int nmonths)
- QDate addYears(int nyears)
- int day(void)
- int dayOfWeek(void)
- int dayOfYear(void)
- int daysInMonth(void)
- int daysInYear(void)
- int daysTo(QDate)
- void getDate(int * year, int * month, int * day)
- bool isNull(void)
- bool isValid(void)
- int month(void)
- bool setDate(int year, int month, int day)
- int toJulianDay(void)
- QString toString(QString)
- int weekNumber(int * yearNumber)
- int year(void)
- QDate currentDate(void)
- QDate fromJulianDay(int jd)
- QDate fromString(QString, QString)
- bool isLeapYear(int year)

91.82 QDateEdit Class

C++ Reference : http://doc.qt.io/qt-5/qdateedit.html

Parameters : QWidget *parent

Parent Class : QDateTimeEdit

91.83 QDateTime Class

C++ Reference : http://doc.qt.io/qt-5/qdatetime.html

Parameters : void

- QDateTime addDays(int ndays)
- QDateTime addMSecs(qint64 msecs)
- QDateTime addMonths(int nmonths)
- QDateTime addSecs(int s)
- QDateTime addYears(int nyears)
- QDate date(void)
- int daysTo(QDateTime other)
- bool isNull(void)
- bool isValid(void)
- qint64 msecsTo(QDateTime other)
- int secsTo(QDateTime other)
- void setDate(QDate date)
- void setMSecsSinceEpoch(qint64 msecs)
- void setTime(QTime time)
- void setTimeSpec(Qt::TimeSpec spec)
- QTime time(void)
- Qt::TimeSpec timeSpec(void)
- QDateTime toLocalTime(void)
- qint64 toMSecsSinceEpoch(void)
- QString toString(QString format)
- QString toString_2(Qt::DateFormat format)
- QDateTime toTimeSpec(Qt::TimeSpec specification)
- QDateTime toUTC(void)
- QDateTime currentDateTime(void)
- QDateTime currentDateTimeUtc(void)
- qint64 currentMSecsSinceEpoch(void)
- QDateTime fromMSecsSinceEpoch(qint64 msecs)
- QDateTime fromString(QString string, Qt::DateFormat format)
- QDateTime fromString_2(QString string, QString format)

91.84 QDateTimeAxis Class

C++ Reference : http://doc.qt.io/qt-5/qdatetimeaxis.html

Parameters : QObject *

Parent Class : QAbstractAxis

- QString format(void)
- QDateTime max(void)
- QDateTime min(void)
- void setFormat(QString format)
- void setMax(QDateTime max)
- void setMin(QDateTime min)
- void setRange(QDateTime min, QDateTime max)
- void setTickCount(int count)
- int tickCount(void)
- void setformatChangedEvent(const char *)
- void setmaxChangedEvent(const char *)
- void setminChangedEvent(const char *)
- void setrangeChangedEvent(const char *)
- void settickCountChangedEvent(const char *)
- const char *getformatChangedEvent(void)
- const char *getmaxChangedEvent(void)
- const char *getminChangedEvent(void)
- const char *getrangeChangedEvent(void)
- const char *gettickCountChangedEvent(void)

91.85 QDateTimeEdit Class

C++ Reference : http://doc.qt.io/qt-5/qdatetimeedit.html

Parameters : QWidget *parent

Parent Class : QAbstractSpinBox

- bool calendarPopup(void)
- QCalendarWidget *calendarWidget(void)
- void clearMaximumDate(void)
- void clearMaximumDateTime(void)
- void clearMaximumTime(void)
- void clearMinimumDate(void)
- void clearMinimumDateTime(void)

- void clearMinimumTime(void)
- int currentSection(void)
- int currentSectionIndex(void)
- QDate date(void)
- QDateTime dateTime(void)
- QString displayFormat(void)
- int displayedSections(void)
- QDate maximumDate(void)
- QDateTime maximumDateTime(void)
- QTime maximumTime(void)
- QDate minimumDate(void)
- QDateTime minimumDateTime(void)
- QTime minimumTime(void)
- int sectionAt(int index)
- int sectionCount(void)
- QString sectionText(QDateTimeEdit::Section section)
- void setCalendarPopup(bool enable)
- void setCalendarWidget(QCalendarWidget *calendarWidget)
- void setCurrentSection(QDateTimeEdit::Section section)
- void setCurrentSectionIndex(int index)
- void setDateRange(QDate,QDate)
- void setDateTimeRange(QDateTime,QDateTime)
- void setDisplayFormat(QString)
- void setMaximumDate(QDate)
- void setMaximumDateTime(QDateTime)
- void setMaximumTime(QTime)
- void setMinimumDate(QDate)
- void setMinimumDateTime(QDateTime)
- void setMinimumTime(QTime)
- void setSelectedSection(QDateTimeEdit::Section section)
- void setTimeRange(QTime,QTime)
- void setTimeSpec(Qt::TimeSpec spec)
- QTime time(void)
- Qt::TimeSpec timeSpec(void)
- void setDate(QDate)
- void setDateTime(QDateTime)

- void setTime(QTime)

91.86 QDepthTest Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qdepthtest.html

Parameters : Qt3DCore::QNode *

- Qt3DRender::QDepthTest::DepthFunction depthFunction(void)
- void setDepthFunction(Qt3DRender::QDepthTest::DepthFunction depthFunction)

91.87 QDesktopServices Class

C++ Reference : http://doc.qt.io/qt-5/qdesktopservices.html

- bool openUrl(QUrl)
- void setUrlHandler(QString, QObject *receiver, const char *method)
- void unsetUrlHandler(QString)

91.88 QDesktopWidget Class

C++ Reference : http://doc.qt.io/qt-5/qdesktopwidget.html

Parameters : void

Parent Class : QWidget

- QRect availableGeometry(QWidget *)
- QScreen *screen(void)
- QRect screenGeometry(QWidget *)
- int screenNumber(QWidget *widget)

91.89 QDial Class

C++ Reference : http://doc.qt.io/qt-5/qdial.html

Parameters : QWidget *parent

Parent Class : QAbstractSlider

- int notchSize(void)
- qreal notchTarget(void)
- bool notchesVisible(void)
- void setNotchTarget(double target)
- bool wrapping(void)
- QSize minimumSizeHint(void)

- QSize sizeHint(void)
- void setNotchesVisible(bool visible)
- void setWrapping(bool on)
- void setactionTriggeredEvent(const char *)
- void setrangeChangedEvent(const char *)
- void setsliderMovedEvent(const char *)
- void setsliderPressedEvent(const char *)
- void setsliderReleasedEvent(const char *)
- void setvalueChangedEvent(const char *)
- const char *getactionTriggeredEvent(void)
- const char *getrangeChangedEvent(void)
- const char *getsliderMovedEvent(void)
- const char *getsliderPressedEvent(void)
- const char *getsliderReleasedEvent(void)
- const char *getvalueChangedEvent(void)

91.90 QDialog Class

C++ Reference : http://doc.qt.io/qt-5/qdialog.html

Parameters : QWidget *parent

Parent Class : QWidget

- bool isSizeGripEnabled(void)
- int result(void)
- void setModal(bool modal)
- void setResult(int i)
- void setSizeGripEnabled(bool)
- void accept(void)
- void done(int r) # In RingQt use : void donedialog(int r)
- int exec(void)
- void open(void)
- void reject(void)

91.91 QDiffuseSpecularMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qdiffusespecularmaterial.html

Parameters : Qt3DCore::QNode *

- QColor ambient(void)
- QVariant diffuse(void)
- bool isAlphaBlendingEnabled(void)
- QVariant normal(void)
- float shininess(void)
- QVariant specular(void)
- float textureScale(void)
- void setAlphaBlendingEnabled(bool enabled)
- void setAmbient(QColor ambient)
- void setDiffuse(QVariant diffuse)
- void setNormal(QVariant normal)
- void setShininess(float shininess)
- void setSpecular(QVariant specular)
- void setTextureScale(float textureScale)

91.92 QDir Class

C++ Reference : http://doc.qt.io/qt-5/qdir.html

Parameters : void

- QString absoluteFilePath(QString fileName)
- QString absolutePath(void)
- QString canonicalPath(void)
- bool cd(QString dirName)
- bool cdUp(void)
- uint count(void)
- QString dirName(void)
- QFileInfoList entryInfoList(QStringList nameFilters, QDir::Filters filters, QDir::SortFlags sort)
- QFileInfoList entryInfoList_2(QDir::Filters filters, QDir::SortFlags sort)
- QStringList entryList(QStringList nameQDir::Filters, QDir::Filters filters, QDir::SortFlags sort)
- QStringList entryList_2(QDir::Filters filters, QDir::SortFlags sort)
- bool exists(QString name)
- bool exists_2(void)
- QString filePath(QString fileName)

- QDir::Filters filter(void)
- bool isAbsolute(void)
- bool isReadable(void)
- bool isRelative(void)
- bool isRoot(void)
- bool makeAbsolute(void)
- bool mkdir(QString dirName)
- bool mkpath(QString dirPath)
- QStringList nameFilters(void)
- QString path(void)
- void refresh(void)
- QString relativeFilePath(QString fileName)
- bool remove(QString fileName)
- bool removeRecursively(void)
- bool rename(QString oldName, QString newName)
- bool rmdir(QString dirName)
- bool rmpath(QString dirPath)
- void setFilter(QDir::Filters filters)
- void setNameFilters(QStringList nameQDir::Filters)
- void setPath(QString path)
- void setSorting(QDir::SortFlags sort)
- QDir::SortFlags sorting(void)
- void swap(QDir other)
- void addSearchPath(QString prefix, QString path)
- QString cleanPath(QString path)
- QDir current(void)
- QString currentPath(void)
- QFileInfoList drives(void)
- QString fromNativeSeparators(QString pathName)
- QDir home(void)
- QString homePath(void)
- bool isAbsolutePath(QString path)
- bool isRelativePath(QString path)
- bool match(QString filter, QString fileName)
- bool match_2(QStringList filters, QString fileName)
- QDir root(void)

- QString rootPath(void)
- QStringList searchPaths(QString prefix)
- QChar separator(void)
- bool setCurrent(QString path)
- void setSearchPaths(QString prefix, QStringList searchPaths)
- QDir temp(void)
- QString tempPath(void)
- QString toNativeSeparators(QString pathName)

91.93 QDockWidget Class

C++ Reference : http://doc.qt.io/qt-5/qdockwidget.html

Parameters : QWidget *parent,Qt::WindowType flag

Parent Class : QWidget

- int allowedAreas(void)
- int features(void)
- bool isAreaAllowed(Qt::DockWidgetArea area)
- bool isFloating(void)
- void setAllowedAreas(Qt::DockWidgetArea areas)
- void setFeatures(QDockWidget::DockWidgetFeature features)
- void setFloating(bool floating)
- void setTitleBarWidget(QWidget *widget)
- void setWidget(QWidget *widget)
- QWidget *titleBarWidget(void)
- QAction *toggleViewAction(void)
- QWidget *widget(void)
- void allowedAreasChanged(Qt::DockWidgetArea allowedAreas)
- void dockLocationChanged(Qt::DockWidgetArea area)
- void featuresChanged(QDockWidget::DockWidgetFeature features)
- void topLevelChanged(bool topLevel)
- void visibilityChanged(bool visible)
- void setallowedAreasChangedEvent(const char *)
- void setdockLocationChangedEvent(const char *)
- void setfeaturesChangedEvent(const char *)
- void settopLevelChangedEvent(const char *)
- void setvisibilityChangedEvent(const char *)

- const char *getallowedAreasChangedEvent(void)
- const char *getdockLocationChangedEvent(void)
- const char *getfeaturesChangedEvent(void)
- const char *gettopLevelChangedEvent(void)
- const char *getvisibilityChangedEvent(void)

91.94 QDrag Class

C++ Reference : http://doc.qt.io/qt-5/qdrag.html

Parameters : QObject *

Parent Class : QObject

- Qt::DropAction defaultAction(void)
- QPixmap dragCursor(Qt::DropAction action)
- Qt::DropAction exec(Qt::DropActions supportedActions)
- Qt::DropAction exec_2(Qt::DropActions supportedActions, Qt::DropAction defaultDropAction)
- QPoint hotSpot(void)
- QMimeData *mimeData(void)
- QPixmap pixmap(void)
- void setDragCursor(QPixmap cursor, Qt::DropAction action)
- void setHotSpot(QPoint hotspot)
- void setMimeData(QMimeData * data)
- void setPixmap(QPixmap pixmap)
- QObject * source(void)
- Qt::DropActions supportedActions(void)
- QObject * target(void)
- void setactionChangedEvent(const char *)
- void settargetChangedEvent(const char *)
- const char *getactionChangedEvent(void)
- const char *gettargetChangedEvent(void)

91.95 QDragEnterEvent Class

C++ Reference : http://doc.qt.io/qt-5/qdragenterevent.html

Parameters : QPoint,Qt::DropActions,const QMimeData *,Qt::MouseButtons,Qt::KeyboardModifiers

Parent Class : QDragMoveEvent

91.96 QDragLeaveEvent Class

C++ Reference : http://doc.qt.io/qt-5/qdragleaveevent.html

Parameters : void

Parent Class : QEvent

91.97 QDragMoveEvent Class

C++ Reference : http://doc.qt.io/qt-5/qdragmoveevent.html

Parameters : QPoint,Qt::DropActions,const QMimeData *,Qt::MouseButtons,Qt::KeyboardModifiers,QEvent::Type

Parent Class : QDropEvent

- void accept(QRect rectangle)
- void accept_2(void)
- QRect answerRect(void)
- void ignore(QRect rectangle)
- void ignore_2(void)

91.98 QDropEvent Class

C++ Reference : http://doc.qt.io/qt-5/qdropevent.html

Parameters : QPointF,Qt::DropActions,const QMimeData *,Qt::MouseButtons,Qt::KeyboardModifiers,QEvent::Type

Parent Class : QEvent

- void acceptProposedAction(void)
- Qt::DropAction dropAction(void)
- Qt::KeyboardModifiers keyboardModifiers(void)
- QMimeData * mimeData(void)
- Qt::MouseButtons mouseButtons(void)
- QPoint pos(void)
- QPointF posF(void)
- Qt::DropActions possibleActions(void)
- Qt::DropAction proposedAction(void)

- void setDropAction(Qt::DropAction action)
- QObject * source(void)

91.99 QEffect Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qeffect.html

Parameters : Qt3DCore::QNode *

- void addParameter(Qt3DRender::QParameter *parameter)
- void addTechnique(Qt3DRender::QTechnique *t)
- QVector<Qt3DRender::QParameter *> parameters(void)
- void removeParameter(Qt3DRender::QParameter *parameter)
- void removeTechnique(Qt3DRender::QTechnique *t)
- QVector<Qt3DRender::QTechnique *> techniques(void)

91.100 QEntity Class

C++ Reference : http://doc.qt.io/qt-5/qt3dcore-qentity.html

Parameters : Qt3DCore::QNode *

Parent Class : QNode

- void addComponent(Qt3DCore::QComponent *comp)
- Qt3DCore::QComponentVector components(void)
- Qt3DCore::QEntity * parentEntity(void)
- void removeComponent(Qt3DCore::QComponent *comp)

91.101 QEvent Class

C++ Reference : http://doc.qt.io/qt-5/qevent.html

Parameters : QEvent::Type Type

- void accept(void)
- void ignore(void)
- bool isAccepted(void)
- void setAccepted(bool accepted)
- bool spontaneous(void)
- int type(void)

91.102 QExtrudedTextMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qextrudedtextmesh.html

Parameters : Qt3DCore::QNode *

- float depth(void)
- QFont font(void)
- QString text(void)
- void setDepth(float depth)
- void setFont(QFont font)
- void setText(QString text)

91.103 QFile Class

C++ Reference : http://doc.qt.io/qt-5/qfile.html

Parameters : void

Parent Class : QFileDevice

- bool copy(QString newName)
- bool exists(void)
- bool link(QString linkName)
- bool open(FILE *fh, QIODevice::OpenMode mode, QFile::FileHandleFlags handleFlags)
- bool open_2(int fd, QIODevice::OpenMode mode, QFile::FileHandleFlags handleFlags)
- bool open_3(QIODevice::OpenMode mode)
- bool remove(void)
- bool rename(QString newName)
- void setFileName(QString name)
- QString symLinkTarget(void)
- bool copy_2(QString fileName, QString newName)
- QString decodeName(QByteArray localFileName)
- QString decodeName_2(char *localFileName)
- QByteArray encodeName(QString fileName)
- bool exists_2(QString fileName)
- bool link_2(QString fileName, QString linkName)
- QFile::Permissions permissions(QString fileName)
- bool remove_2(QString fileName)
- bool rename_2(QString oldName, QString newName)
- bool resize(QString fileName, qint64 sz)
- bool setPermissions(QString fileName, QFile::Permissions permissions)

- QString symLinkTarget_2(QString fileName)

91.104 QFile2 Class

Parameters : QString

Parent Class : QFile

91.105 QFileDevice Class

C++ Reference : http://doc.qt.io/qt-5/qfiledevice.html

Parent Class : QIODevice

- QFileDevice::FileError error(void)
- bool flush(void)
- int handle(void)
- uchar * map(qint64 offset, qint64 size, QFileDevice::MemoryMapFlags flags)
- QFileDevice::Permissions permissions(void)
- bool resize(qint64 sz)
- QString fileName(void)
- bool unmap(uchar *address)
- void unsetError(void)

91.106 QFileDialog Class

C++ Reference : http://doc.qt.io/qt-5/qfiledialog.html

Parameters : QWidget *parent

Parent Class : QDialog

- QFileDialog::AcceptMode acceptMode(void)
- QString defaultSuffix(void)
- QDir directory(void)
- QUrl directoryUrl(void)
- QFileDialog::FileMode fileMode(void)
- QDir::Filters filter(void)
- QStringList history(void)
- QFileIconProvider * iconProvider(void)
- QAbstractItemDelegate * itemDelegate(void)
- QString labelText(QFileDialog::DialogLabel label)
- QStringList mimeTypeFilters(void)

- QStringList nameFilters(void)
- void open(QObject * receiver, char * member)
- QFileDialog::Options options(void)
- QAbstractProxyModel * proxyModel(void)
- bool restoreState(QByteArray state)
- QByteArray saveState(void)
- void selectFile(QString filename)
- void selectMimeTypeFilter(QString filter)
- void selectNameFilter(QString filter)
- void selectUrl(QUrl url)
- QStringList selectedFiles(void)
- QString selectedNameFilter(void)
- QList<QUrl> selectedUrls(void)
- void setAcceptMode(QFileDialog::AcceptMode mode)
- void setDefaultSuffix(QString suffix)
- void setDirectory(QString directory)
- void setDirectory_2(QDir directory)
- void setDirectoryUrl(QUrl directory)
- void setFileMode(QFileDialog::FileMode mode)
- void setFilter(QDir::Filters filters)
- void setHistory(QStringList paths)
- void setIconProvider(QFileIconProvider * provider)
- void setItemDelegate(QAbstractItemDelegate * delegate)
- void setLabelText(QFileDialog::DialogLabel label, QString text)
- void setMimeTypeFilters(QStringList filters)
- void setNameFilter(QString filter)
- void setNameFilters(QStringList filters)
- void setOption(QFileDialog::Option option, bool on)
- void setOptions(QFileDialog::Options options)
- void setProxyModel(QAbstractProxyModel * proxyModel)
- void setSidebarUrls(QList<QUrl> urls)
- void setViewMode(QFileDialog::ViewMode mode)
- QList<QUrl> sidebarUrls(void)
- bool testOption(QFileDialog::Option option)
- QFileDialog::ViewMode viewMode(void)
- QString getExistingDirectory(QWidget * parent, QString caption, QString dir, QFileDialog::Options options)

- QUrl getExistingDirectoryUrl(QWidget * parent, QString caption, QUrl dir, QFileDialog::Options options, QStringList supportedSchemes)
- QString getOpenFileName(QWidget * parent, QString caption, QString dir, QString filter)
- QString getOpenFileName_2(QWidget * parent, QString caption, QString dir, QString filter, QString * selectedFilter, QFileDialog::Options options)
- QStringList getOpenFileNames(QWidget * parent, QString caption, QString dir, QString filter, QString * selectedFilter, QFileDialog::Options options)
- QUrl getOpenFileUrl(QWidget * parent, QString caption, QUrl dir, QString filter, QString * selectedFilter, QFileDialog::Options options, QStringList supportedSchemes)
- QList<QUrl> getOpenFileUrls(QWidget * parent, QString caption, QUrl dir, QString filter, QString * selectedFilter, QFileDialog::Options options, QStringList supportedSchemes)
- QString getSaveFileName(QWidget * parent, QString caption, QString dir, QString filter)
- QString getSaveFileName_2(QWidget * parent, QString caption, QString dir, QString filter, QString * selectedFilter, QFileDialog::Options options)
- QUrl getSaveFileUrl(QWidget * parent, QString caption, QUrl dir, QString filter, QString * selectedFilter, QFileDialog::Options options, QStringList supportedSchemes)
- void setcurrentChangedEvent(const char *)
- void setcurrentUrlChangedEvent(const char *)
- void setdirectoryEnteredEvent(const char *)
- void setdirectoryUrlEnteredEvent(const char *)
- void setfileSelectedEvent(const char *)
- void setfilesSelectedEvent(const char *)
- void setfilterSelectedEvent(const char *)
- void seturlSelectedEvent(const char *)
- void seturlsSelectedEvent(const char *)
- const char *getcurrentChangedEvent(void)
- const char *getcurrentUrlChangedEvent(void)
- const char *getdirectoryEnteredEvent(void)
- const char *getdirectoryUrlEnteredEvent(void)
- const char *getfileSelectedEvent(void)
- const char *getfilesSelectedEvent(void)
- const char *getfilterSelectedEvent(void)
- const char *geturlSelectedEvent(void)
- const char *geturlsSelectedEvent(void)
- void saveFileContent(QByteArray fileContent,QString fileNameHint)
- void getOpenFileContent(const char *cFileTypes, const char *cCode)
- List *FileContentList(void)

91.107 QFileInfo Class

C++ Reference : http://doc.qt.io/qt-5/qfileinfo.html

Parameters : void

- QDir absoluteDir(void)
- QString absoluteFilePath(void)
- QString absolutePath(void)
- QString baseName(void)
- QString bundleName(void)
- bool caching(void)
- QString canonicalFilePath(void)
- QString canonicalPath(void)
- QString completeBaseName(void)
- QString completeSuffix(void)
- QDir dir(void)
- bool exists(void)
- QString fileName(void)
- QString filePath(void)
- QString group(void)
- int groupId(void)
- bool isAbsolute(void)
- bool isBundle(void)
- bool isDir(void)
- bool isExecutable(void)
- bool isFile(void)
- bool isHidden(void)
- bool isNativePath(void)
- bool isReadable(void)
- bool isRelative(void)
- bool isRoot(void)
- bool isSymLink(void)
- bool isWritable(void)
- QDateTime lastModified(void)
- QDateTime lastRead(void)
- bool makeAbsolute(void)
- QString owner(void)

- uint ownerId(void)
- QString path(void)
- bool permission(QFileDevice::Permission permissions)
- int permissions(void)
- void refresh(void)
- void setCaching(bool enable)
- void setFile(QString)
- int size(void)
- QString suffix(void)
- void swap(QFileInfo)
- QString symLinkTarget(void)

91.108 QFileSystemModel Class

C++ Reference : http://doc.qt.io/qt-5/qfilesystemmodel.html

Parameters : void

- QIcon fileIcon(QModelIndex)
- QFileInfo fileInfo(QModelIndex)
- QString fileName(QModelIndex)
- QString filePath(QModelIndex)
- int filter(void)
- QFileIconProvider *iconProvider(void)
- QModelIndex index(QString, int column)
- bool isDir(QModelIndex)
- bool isReadOnly(void)
- QDateTime lastModified(QModelIndex)
- QModelIndex mkdir(QModelIndex,QString)
- QVariant myComputer(int role)
- bool nameFilterDisables(void)
- QStringList nameFilters(void)
- int permissions(QModelIndex)
- bool remove(QModelIndex)
- bool resolveSymlinks(void)
- bool rmdir(QModelIndex)
- QDir rootDirectory(void)
- QString rootPath(void)

- void setFilter(QDir::Filter filters)
- void setIconProvider(QFileIconProvider *provider)
- void setNameFilterDisables(bool enable)
- void setNameFilters(QStringList)
- void setReadOnly(bool enable)
- void setResolveSymlinks(bool enable)
- QModelIndex setRootPath(QString)
- int size(QModelIndex)
- QString type(QModelIndex)
- bool canFetchMore(QModelIndex)
- int columnCount(void)
- QVariant data(QModelIndex index, int role)
- bool dropMimeData(QMimeData *data, Qt::DropAction action, int row, int column, QModelIndex parent)
- void fetchMore(QModelIndex parent)
- int flags(QModelIndex index)
- bool hasChildren(QModelIndex parent)
- QVariant headerData(int section, Qt::Orientation orientation, int role)
- QMimeData * mimeData(QModelIndexList indexes)
- QStringList mimeTypes(void)
- QModelIndex parent(QModelIndex index)
- int rowCount(QModelIndex parent)
- bool setData(QModelIndex idx, QVariant value, int role)
- void sort(int column, Qt::SortOrder order)
- int supportedDropActions(void)

91.109 QFirstPersonCameraController Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qfirstpersoncameracontroller.html

Parameters : Qt3DCore::QNode *

Parent Class : QAbstractCameraController

91.110 QFont Class

C++ Reference : http://doc.qt.io/qt-5/qfont.html

Parameters : QString, int, int, bool

- bool bold(void)
- int capitalization(void)
- QString defaultFamily(void)
- bool exactMatch(void)
- QString family(void)
- bool fixedPitch(void)
- bool fromString(QString)
- int hintingPreference(void)
- bool isCopyOf(QFont)
- bool italic(void)
- bool kerning(void)
- QString key(void)
- double letterSpacing(void)
- int letterSpacingType(void)
- bool overline(void)
- int pixelSize(void)
- int pointSize(void)
- double pointSizeF(void)
- QFont resolve(QFont)
- void setBold(bool enable)
- void setCapitalization(QFont::Capitalization caps)
- void setFamily(QString)
- void setFixedPitch(bool enable)
- void setHintingPreference(QFont::HintingPreference hintingPreference)
- void setItalic(bool enable)
- void setKerning(bool enable)
- void setLetterSpacing(QFont::SpacingType type, double spacing)
- void setOverline(bool enable)
- void setPixelSize(int pixelSize)
- void setPointSize(int pointSize)
- void setPointSizeF(double pointSize)
- void setStretch(int factor)

- void setStrikeOut(bool enable)
- void setStyle(QFont::Style style)
- void setStyleHint(QFont::StyleHint hint, QFont::StyleStrategy strategy)
- void setStyleName(QString)
- void setStyleStrategy(QFont::StyleStrategy s)
- void setUnderline(bool enable)
- void setWeight(int weight)
- void setWordSpacing(double spacing)
- int stretch(void)
- bool strikeOut(void)
- int style(void)
- int styleHint(void)
- QString styleName(void)
- int styleStrategy(void)
- QString toString(void)
- bool underline(void)
- int weight(void)
- double wordSpacing(void)
- void insertSubstitution(QString,QString)
- void insertSubstitutions(QString,QStringList)
- QString substitute(QString)
- QStringList substitutes(QString)
- QStringList substitutions(void)

91.111 QFontDialog Class

C++ Reference : http://doc.qt.io/qt-5/qfontdialog.html

Parameters : void

Parent Class : QDialog

- QFont currentFont(void)
- void open(QObject * receiver, char * member)
- QFontDialog::FontDialogOptions options(void)
- QFont selectedFont(void)
- void setCurrentFont(QFont font)
- void setOption(QFontDialog::FontDialogOption option, bool on)
- void setOptions(QFontDialog::FontDialogOptions options)

- bool testOption(QFontDialog::FontDialogOption option)
- QFont getFont_2(bool * ok, QFont initial, QWidget * parent, QString title, QFontDialog::FontDialogOptions options)
- QFont getFont_3(bool * ok, QWidget * parent)
- void setcurrentFontChangedEvent(const char *)
- void setfontSelectedEvent(const char *)
- const char *getcurrentFontChangedEvent(void)
- const char *getfontSelectedEvent(void)
- int getfont(void)

91.112 QFontMetrics Class

C++ Reference : http://doc.qt.io/qt-5/qfontmetrics.html

Parameters : QFont

- int ascent(void)
- int averageCharWidth(void)
- QRect boundingRect(QChar ch)
- QRect boundingRect_2(QString text)
- QRect boundingRect_3(int x, int y, int width, int height, int flags, QString text, int tabStops , int * tabArray)
- QRect boundingRect_4(QRect rect, int flags, QString text, int tabStops , int * tabArray)
- int descent(void)
- QString elidedText(QString text, Qt::TextElideMode mode, int width, int flags)
- int height(void)
- bool inFont(QChar ch)
- bool inFontUcs4(uint character)
- int leading(void)
- int leftBearing(QChar ch)
- int lineSpacing(void)
- int lineWidth(void)
- int maxWidth(void)
- int minLeftBearing(void)
- int minRightBearing(void)
- int overlinePos(void)
- int rightBearing(QChar ch)
- QSize size(int flags, QString text, int tabStops , int * tabArray)
- int strikeOutPos(void)
- QRect tightBoundingRect(QString text)

- int underlinePos(void)
- int xHeight(void)
- int horizontalAdvance(QString text, int len)
- int horizontalAdvance_2(QChar ch)

91.113 QForwardRenderer Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qforwardrenderer.html

Parameters : Qt3DCore::QNode *

- Qt3DCore::QEntity * camera(void)
- QColor clearColor(void)
- QSize externalRenderTargetSize(void)
- float gamma(void)
- bool isFrustumCullingEnabled(void)
- QObject * surface(void)
- QRectF viewportRect(void)
- void setCamera(Qt3DCore::QEntity *camera)
- void setClearColor(QColor clearColor)
- void setExternalRenderTargetSize(QSize size)
- void setFrustumCullingEnabled(bool enabled)
- void setGamma(float gamma)
- void setSurface(QObject *surface)
- void setViewportRect(QRectF viewportRect)

91.114 QFrame Class

C++ Reference : http://doc.qt.io/qt-5/qframe.html

Parameters : QWidget *parent, Qt::WindowType flag

Parent Class : QWidget

- QRect frameRect(void)
- int frameShadow(void)
- int frameShape(void)
- int frameStyle(void)
- int frameWidth(void)
- int lineWidth(void)
- int midLineWidth(void)
- void setFrameRect(QRect)

- void setFrameShadow(QFrame::Shadow)
- void setFrameShape(QFrame::Shape)
- void setFrameStyle(int style)
- void setLineWidth(int)
- void setMidLineWidth(int)
- QSize sizeHint(void)

91.115 QFrame2 Class

Parameters : void

Parent Class : QFrame

91.116 QFrame3 Class

Parameters : QWidget *parent

Parent Class : QFrame

91.117 QFrameAction Class

C++ Reference : http://doc.qt.io/qt-5/qt3dlogic-qframeaction.html

Parameters : Qt3DCore::QNode *

- void settriggeredEvent(const char *)
- const char *gettriggeredEvent(void)

91.118 QGeoAddress Class

C++ Reference : http://doc.qt.io/qt-5/qgeoaddress.html

Parameters : void

- QString city(void)
- void clear(void)
- QString country(void)
- QString countryCode(void)
- QString county(void)
- QString district(void)
- bool isEmpty(void)
- bool isTextGenerated(void)
- QString postalCode(void)

- void setCity(QString city)
- void setCountry(QString country)
- void setCountryCode(QString countryCode)
- void setCounty(QString county)
- void setDistrict(QString district)
- void setPostalCode(QString postalCode)
- void setState(QString state)
- void setStreet(QString street)
- void setText(QString text)
- QString state(void)
- QString street(void)
- QString text(void)

91.119 QGeoAreaMonitorInfo Class

C++ Reference : http://doc.qt.io/qt-5/qgeoareamonitorinfo.html

Parameters : QString

- QGeoShape area(void)
- QDateTime expiration(void)
- QString identifier(void)
- bool isPersistent(void)
- bool isValid(void)
- QString name(void)
- QVariantMap notificationParameters(void)
- void setArea(QGeoShape newShape)
- void setExpiration(QDateTime expiry)
- void setName(QString name)
- void setNotificationParameters(QVariantMap parameters)
- void setPersistent(bool isPersistent)

91.120 QGeoAreaMonitorSource Class

C++ Reference : http://doc.qt.io/qt-5/qgeoareamonitorsource.html

Parameters : QObject *

Parent Class : QObject

- QString sourceName(void)
- QStringList availableSources(void)
- QGeoAreaMonitorSource * createDefaultSource(QObject * parent)
- QGeoAreaMonitorSource * createSource(QString sourceName, QObject * parent)

91.121 QGeoCircle Class

C++ Reference : http://doc.qt.io/qt-5/qgeocircle.html

Parameters : void

- QGeoCoordinate center(void)
- qreal radius(void)
- void setCenter(QGeoCoordinate center)
- void setRadius(qreal radius)
- void translate(double degreesLatitude, double degreesLongitude)
- QGeoCircle translated(double degreesLatitude, double degreesLongitude)

91.122 QGeoCoordinate Class

C++ Reference : http://doc.qt.io/qt-5/qgeocoordinate.html

Parameters : void

- double altitude(void)
- QGeoCoordinate atDistanceAndAzimuth(qreal distance, qreal azimuth, qreal distanceUp)
- qreal azimuthTo(QGeoCoordinate other)
- qreal distanceTo(QGeoCoordinate other)
- bool isValid(void)
- double latitude(void)
- double longitude(void)
- void setAltitude(double altitude)
- void setLatitude(double latitude)
- void setLongitude(double longitude)
- QString toString(QGeoCoordinate::CoordinateFormat format)
- QGeoCoordinate::CoordinateType type(void)

91.123 QGeoPositionInfo Class

C++ Reference : http://doc.qt.io/qt-5/qgeopositioninfo.html

Parameters : void

- qreal attribute(QGeoPositionInfo::Attribute attribute)
- QGeoCoordinate coordinate(void)
- bool hasAttribute(QGeoPositionInfo::Attribute attribute)
- bool isValid(void)
- void removeAttribute(QGeoPositionInfo::Attribute attribute)
- void setAttribute(QGeoPositionInfo::Attribute attribute, qreal value)
- void setCoordinate(QGeoCoordinate coordinate)
- void setTimestamp(QDateTime timestamp)
- QDateTime timestamp(void)

91.124 QGeoPositionInfoSource Class

C++ Reference : http://doc.qt.io/qt-5/qgeopositioninfosource.html

Parent Class : QObject

- QGeoPositionInfoSource::PositioningMethods preferredPositioningMethods(void)
- QString sourceName(void)
- int updateInterval(void)
- void seterrorEvent(const char *)
- void setpositionUpdatedEvent(const char *)
- void setupdateTimeoutEvent(const char *)
- const char *geterrorEvent(void)
- const char *getpositionUpdatedEvent(void)
- const char *getupdateTimeoutEvent(void)
- QStringList availableSources(void)
- QGeoPositionInfoSource *createDefaultSource(QObject *parent)
- QGeoPositionInfoSource *createSource(QString sourceName, QObject *parent)

91.125 QGeoRectangle Class

C++ Reference : http://doc.qt.io/qt-5/qgeorectangle.html

Parameters : void

Parent Class : QGeoShape

- QGeoCoordinate bottomLeft(void)
- QGeoCoordinate bottomRight(void)
- QGeoCoordinate center(void)
- bool contains(QGeoRectangle rectangle)
- double height(void)
- bool intersects(QGeoRectangle rectangle)
- void setBottomLeft(QGeoCoordinate bottomLeft)
- void setBottomRight(QGeoCoordinate bottomRight)
- void setCenter(QGeoCoordinate center)
- void setHeight(double degreesHeight)
- void setTopLeft(QGeoCoordinate topLeft)
- void setTopRight(QGeoCoordinate topRight)
- void setWidth(double degreesWidth)
- QGeoCoordinate topLeft(void)
- QGeoCoordinate topRight(void)
- void translate(double degreesLatitude, double degreesLongitude)
- QGeoRectangle translated(double degreesLatitude, double degreesLongitude)
- QGeoRectangle united(QGeoRectangle rectangle)
- double width(void)

91.126 QGeoSatelliteInfo Class

C++ Reference : http://doc.qt.io/qt-5/qgeosatelliteinfo.html

Parameters : void

- qreal attribute(QGeoSatelliteInfo::Attribute attribute)
- bool hasAttribute(QGeoSatelliteInfo::Attribute attribute)
- void removeAttribute(QGeoSatelliteInfo::Attribute attribute)
- int satelliteIdentifier(void)
- QGeoSatelliteInfo::SatelliteSystem satelliteSystem(void)
- void setAttribute(QGeoSatelliteInfo::Attribute attribute, qreal value)
- void setSatelliteIdentifier(int satId)
- void setSatelliteSystem(QGeoSatelliteInfo::SatelliteSystem system)

- void setSignalStrength(int signalStrength)
- int signalStrength(void)

91.127 QGeoSatelliteInfoSource Class

C++ Reference : http://doc.qt.io/qt-5/qgeosatelliteinfosource.html

- QString sourceName(void)
- int updateInterval(void)

91.128 QGeoShape Class

C++ Reference : http://doc.qt.io/qt-5/qgeoshape.html

Parameters : void

- bool contains(QGeoCoordinate coordinate)
- bool isEmpty(void)
- bool isValid(void)
- QGeoShape::ShapeType type(void)

91.129 QGoochMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qgoochmaterial.html

Parameters : Qt3DCore::QNode *

- float alpha(void)
- float beta(void)
- QColor cool(void)
- QColor diffuse(void)
- float shininess(void)
- QColor specular(void)
- QColor warm(void)
- void setAlpha(float alpha)
- void setBeta(float beta)
- void setCool(QColor cool)
- void setDiffuse(QColor diffuse)
- void setShininess(float shininess)
- void setSpecular(QColor specular)
- void setWarm(QColor warm)

91.130 QGradient Class

C++ Reference : http://doc.qt.io/qt-5/qgradient.html

Parameters : void

- QGradient::CoordinateMode coordinateMode(void)
- void setColorAt(qreal position, QColor color)
- void setCoordinateMode(QGradient::CoordinateMode mode)
- void setSpread(QGradient::Spread method)
- void setStops(QGradientStops stopPoints)
- QGradient::Spread spread(void)
- QGradientStops stops(void)
- QGradient::Type type(void)

91.131 QGraphicsScene Class

C++ Reference : http://doc.qt.io/qt-5/qgraphicsscene.html

Parameters : QObject *

Parent Class : QObject

- QGraphicsItem * activePanel(void)
- QGraphicsWidget * activeWindow(void)
- QGraphicsEllipseItem * addEllipse(QRectF rect, QPen pen, QBrush brush)
- QGraphicsEllipseItem * addEllipse_2(qreal x, qreal y, qreal w, qreal h, QPen pen, QBrush brush)
- void addItem(QGraphicsItem *item)
- QGraphicsLineItem * addLine(QLineF line, QPen pen)
- QGraphicsLineItem * addLine_2(qreal x1, qreal y1, qreal x2, qreal y2, QPen pen)
- QGraphicsPathItem * addPath(QPainterPath path, QPen pen, QBrush brush)
- QGraphicsPixmapItem * addPixmap(QPixmap pixmap)
- QGraphicsPolygonItem * addPolygon(QPolygonF polygon, QPen pen, QBrush brush)
- QGraphicsRectItem * addRect(QRectF rect, QPen pen, QBrush brush)
- QGraphicsRectItem * addRect_2(qreal x, qreal y, qreal w, qreal h, QPen pen, QBrush brush)
- QGraphicsSimpleTextItem * addSimpleText(QString text, QFont font)
- QGraphicsTextItem * addText(QString text, QFont font)
- QGraphicsProxyWidget * addWidget(QWidget *widget, Qt::WindowFlags wFlags)
- QBrush backgroundBrush(void)
- int bspTreeDepth(void)
- void clearFocus(void)
- QList<QGraphicsItem *> collidingItems(QGraphicsItem *item, Qt::ItemSelectionMode mode)

- QGraphicsItemGroup * createItemGroup(QList<QGraphicsItem *> items)
- void destroyItemGroup(QGraphicsItemGroup *group)
- QGraphicsItem * focusItem(void)
- QBrush foregroundBrush(void)
- bool hasFocus(void)
- qreal height(void)
- QVariant inputMethodQuery(Qt::InputMethodQuery query)
- void invalidate(qreal x, qreal y, qreal w, qreal h, QGraphicsScene::SceneLayers layers)
- bool isActive(void)
- QGraphicsItem * itemAt(QPointF position, QTransform deviceTransform)
- QGraphicsItem * itemAt_2(qreal x, qreal y, QTransform deviceTransform)
- QGraphicsScene::ItemIndexMethod itemIndexMethod(void)
- QList<QGraphicsItem *> items(Qt::SortOrder order)
- QList<QGraphicsItem *> items_2(QPointF pos, Qt::ItemSelectionMode mode, Qt::SortOrder order, QTransform deviceTransform)
- QList<QGraphicsItem *> items_3(QRectF rect, Qt::ItemSelectionMode mode, Qt::SortOrder order, QTransform deviceTransform)
- QList<QGraphicsItem *> items_4(QPolygonF polygon, Qt::ItemSelectionMode mode, Qt::SortOrder order, QTransform deviceTransform)
- QList<QGraphicsItem *> items_5(QPainterPath path, Qt::ItemSelectionMode mode, Qt::SortOrder order, QTransform deviceTransform)
- QList<QGraphicsItem *> items_6(qreal x, qreal y, qreal w, qreal h, Qt::ItemSelectionMode mode, Qt::SortOrder order, QTransform deviceTransform)
- QRectF itemsBoundingRect(void)
- qreal minimumRenderSize(void)
- QGraphicsItem * mouseGrabberItem(void)
- QPalette palette(void)
- void removeItem(QGraphicsItem *item)
- void render(QPainter *painter, QRectF target, QRectF source, Qt::AspectRatioMode aspectRatioMode)
- QRectF sceneRect(void)
- QList<QGraphicsItem *> selectedItems(void)
- QPainterPath selectionArea(void)
- bool sendEvent(QGraphicsItem *item, QEvent *event)
- void setActivePanel(QGraphicsItem *item)
- void setActiveWindow(QGraphicsWidget *widget)
- void setBackgroundBrush(QBrush brush)
- void setBspTreeDepth(int depth)
- void setFocus(Qt::FocusReason focusReason)

- void setFocusItem(QGraphicsItem *item, Qt::FocusReason focusReason)
- void setForegroundBrush(QBrush brush)
- void setItemIndexMethod(QGraphicsScene::ItemIndexMethod method)
- void setMinimumRenderSize(qreal minSize)
- void setPalette(QPalette palette)
- void setSceneRect(QRectF rect)
- void setSceneRect_2(qreal x, qreal y, qreal w, qreal h)
- void setSelectionArea(QPainterPath path, QTransform deviceTransform)
- void setSelectionArea_2(QPainterPath path, Qt::ItemSelectionMode mode, QTransform deviceTransform)
- void setSelectionArea_3(QPainterPath path, Qt::ItemSelectionOperation selectionOperation, Qt::ItemSelectionMode mode, QTransform deviceTransform)
- void setStickyFocus(bool enabled)
- void setStyle(QStyle *style)
- bool stickyFocus(void)
- QStyle * style(void)
- void update(qreal x, qreal y, qreal w, qreal h)
- QList<QGraphicsView *> views(void)
- qreal width(void)
- void advance(void)
- void clear(void)
- void clearSelection(void)
- void invalidate_2(QRectF rect, QGraphicsScene::SceneLayers layers)
- void update_2(QRectF rect)
- void setchangedEvent(const char *)
- void setfocusItemChangedEvent(const char *)
- void setsceneRectChangedEvent(const char *)
- void setselectionChangedEvent(const char *)
- const char *getchangedEvent(void)
- const char *getfocusItemChangedEvent(void)
- const char *getsceneRectChangedEvent(void)
- const char *getselectionChangedEvent(void)

91.132 QGraphicsVideoItem Class

C++ Reference : http://doc.qt.io/qt-5/qgraphicsvideoitem.html

Parameters : void

- Qt::AspectRatioMode aspectRatioMode(void)
- QSizeF nativeSize(void)
- QPointF offset(void)
- void setAspectRatioMode(Qt::AspectRatioMode mode)
- void setOffset(QPointF offset)
- void setSize(QSizeF size)
- QSizeF size(void)

91.133 QGraphicsView Class

C++ Reference : http://doc.qt.io/qt-5/qgraphicsview.html

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- Qt::Alignment alignment(void)
- QBrush backgroundBrush(void)
- QGraphicsView::CacheMode cacheMode(void)
- void centerOn(QPointF pos)
- void centerOn_2(qreal x, qreal y)
- void centerOn_3(QGraphicsItem *item)
- QGraphicsView::DragMode dragMode(void)
- void ensureVisible(QRectF rect, int xmargin, int ymargin)
- void ensureVisible_2(qreal x, qreal y, qreal w, qreal h, int xmargin, int ymargin)
- void ensureVisible_3(QGraphicsItem *item, int xmargin, int ymargin)
- void fitInView(QRectF rect, Qt::AspectRatioMode aspectRatioMode)
- void fitInView_2(qreal x, qreal y, qreal w, qreal h, Qt::AspectRatioMode aspectRatioMode)
- void fitInView_3(QGraphicsItem *item, Qt::AspectRatioMode aspectRatioMode)
- QBrush foregroundBrush(void)
- bool isInteractive(void)
- bool isTransformed(void)
- QGraphicsItem * itemAt(QPoint pos)
- QGraphicsItem * itemAt_2(int x, int y)
- QList<QGraphicsItem *> items(void)
- QList<QGraphicsItem *> items_2(QPoint pos)

- QList<QGraphicsItem *> items_3(int x, int y)
- QList<QGraphicsItem *> items_4(QRect rect, Qt::ItemSelectionMode mode)
- QList<QGraphicsItem *> items_5(int x, int y, int w, int h, Qt::ItemSelectionMode mode)
- QList<QGraphicsItem *> items_6(QPolygon polygon, Qt::ItemSelectionMode mode)
- QList<QGraphicsItem *> items_7(QPainterPath path, Qt::ItemSelectionMode mode)
- QPoint mapFromScene(QPointF point)
- QPolygon mapFromScene_2(QRectF rect)
- QPolygon mapFromScene_3(QPolygonF polygon)
- QPainterPath mapFromScene_4(QPainterPath path)
- QPoint mapFromScene_5(qreal x, qreal y)
- QPolygon mapFromScene_6(qreal x, qreal y, qreal w, qreal h)
- QPointF mapToScene(QPoint point)
- QPolygonF mapToScene_2(QRect rect)
- QPolygonF mapToScene_3(QPolygon polygon)
- QPainterPath mapToScene_4(QPainterPath path)
- QPointF mapToScene_5(int x, int y)
- QPolygonF mapToScene_6(int x, int y, int w, int h)
- QGraphicsView::OptimizationFlags optimizationFlags(void)
- void render(QPainter *painter, QRectF target, QRect source, Qt::AspectRatioMode aspectRatioMode)
- QPainter::RenderHints renderHints(void)
- void resetCachedContent(void)
- void resetTransform(void)
- QGraphicsView::ViewportAnchor resizeAnchor(void)
- void rotate(qreal angle)
- QRect rubberBandRect(void)
- Qt::ItemSelectionMode rubberBandSelectionMode(void)
- void scale(qreal sx, qreal sy)
- QGraphicsScene * scene(void)
- QRectF sceneRect(void)
- void setAlignment(Qt::Alignment alignment)
- void setBackgroundBrush(QBrush brush)
- void setCacheMode(QGraphicsView::CacheMode mode)
- void setDragMode(QGraphicsView::DragMode mode)
- void setForegroundBrush(QBrush brush)
- void setInteractive(bool allowed)
- void setOptimizationFlag(QGraphicsView::OptimizationFlag flag, bool enabled)

- void setOptimizationFlags(QGraphicsView::OptimizationFlags flags)
- void setRenderHint(QPainter::RenderHint hint, bool enabled)
- void setRenderHints(QPainter::RenderHints hints)
- void setResizeAnchor(QGraphicsView::ViewportAnchor anchor)
- void setRubberBandSelectionMode(Qt::ItemSelectionMode mode)
- void setScene(QGraphicsScene *scene)
- void setSceneRect(QRectF rect)
- void setSceneRect_2(qreal x, qreal y, qreal w, qreal h)
- void setTransform(QTransform matrix, bool combine)
- void setTransformationAnchor(QGraphicsView::ViewportAnchor anchor)
- void setViewportUpdateMode(QGraphicsView::ViewportUpdateMode mode)
- void shear(qreal sh, qreal sv)
- QTransform transform(void)
- QGraphicsView::ViewportAnchor transformationAnchor(void)
- void translate(qreal dx, qreal dy)
- QTransform viewportTransform(void)
- QGraphicsView::ViewportUpdateMode viewportUpdateMode(void)
- void invalidateScene(QRectF rect, QGraphicsScene::SceneLayers layers)
- void updateScene(QList<QRectF> rects)
- void updateSceneRect(QRectF rect)

91.134 QGridLayout Class

C++ Reference : http://doc.qt.io/qt-5/qgridlayout.html

Parameters : void

- void addItem(QLayoutItem * item, int row, int column, int rowSpan , int columnSpan , Qt::Alignment alignment)
- void addLayout(QLayout * layout, int row, int column, Qt::Alignment alignment)
- void addLayout_2(QLayout * layout, int row, int column, int rowSpan, int columnSpan, Qt::Alignment alignment)
- void addWidget(QWidget * widget, int row, int column, Qt::Alignment alignment)
- void addWidget_2(QWidget * widget, int fromRow, int fromColumn, int rowSpan, int columnSpan, Qt::Alignment alignment)
- QRect cellRect(int row, int column)
- int columnCount(void)
- int columnMinimumWidth(int column)
- int columnStretch(int column)

- void getItemPosition(int index, int * row, int * column, int * rowSpan, int * columnSpan)
- int horizontalSpacing(void)
- QLayoutItem * itemAtPosition(int row, int column)
- Qt::Corner originCorner(void)
- int rowCount(void)
- int rowMinimumHeight(int row)
- int rowStretch(int row)
- void setColumnMinimumWidth(int column, int minSize)
- void setColumnStretch(int column, int stretch)
- void setHorizontalSpacing(int spacing)
- void setOriginCorner(Qt::Corner corner)
- void setRowMinimumHeight(int row, int minSize)
- void setRowStretch(int row, int stretch)
- void setSpacing(int spacing)
- void setVerticalSpacing(int spacing)
- int spacing(void)
- int verticalSpacing(void)

91.135 QGuiApplication Class

C++ Reference : http://doc.qt.io/qt-5/qguiapplication.html

Parent Class : QCoreApplication

Parameters : int,char **

- qreal devicePixelRatio(void)
- bool isSavingSession(void)
- bool isSessionRestored(void)
- QString sessionId(void)
- QString sessionKey(void)
- QWindowList allWindows(void)
- QString applicationDisplayName(void)
- Qt::ApplicationState applicationState(void)
- void changeOverrideCursor(QCursor cursor)
- QClipboard * clipboard(void)
- bool desktopSettingsAware(void)
- int exec(void)
- QObject * focusObject(void)

- QWindow * focusWindow(void)
- QFont font(void)
- QInputMethod * inputMethod(void)
- bool isLeftToRight(void)
- bool isRightToLeft(void)
- Qt::KeyboardModifiers keyboardModifiers(void)
- Qt::LayoutDirection layoutDirection(void)
- QWindow * modalWindow(void)
- Qt::MouseButtons mouseButtons(void)
- QCursor * overrideCursor(void)
- QPalette palette(void)
- QString platformName(void)
- QPlatformNativeInterface * platformNativeInterface(void)
- QScreen * primaryScreen(void)
- Qt::KeyboardModifiers queryKeyboardModifiers(void)
- bool quitOnLastWindowClosed(void)
- void restoreOverrideCursor(void)
- QList<QScreen *> screens(void)
- void setApplicationDisplayName(QString name)
- void setDesktopSettingsAware(bool on)
- void setFont(QFont font)
- void setLayoutDirection(Qt::LayoutDirection direction)
- void setOverrideCursor(QCursor cursor)
- void setPalette(QPalette pal)
- void setQuitOnLastWindowClosed(bool quit)
- QStyleHints * styleHints(void)
- void sync(void)
- QWindow * topLevelAt(QPoint pos)
- QWindowList topLevelWindows(void)
- void setapplicationDisplayNameChangedEvent(const char *)
- void setapplicationStateChangedEvent(const char *)
- void setcommitDataRequestEvent(const char *)
- void setfocusObjectChangedEvent(const char *)
- void setfocusWindowChangedEvent(const char *)
- void setfontDatabaseChangedEvent(const char *)
- void setlastWindowClosedEvent(const char *)

- void setlayoutDirectionChangedEvent(const char *)
- void setpaletteChangedEvent(const char *)
- void setprimaryScreenChangedEvent(const char *)
- void setsaveStateRequestEvent(const char *)
- void setscreenAddedEvent(const char *)
- void setscreenRemovedEvent(const char *)
- const char *getapplicationDisplayNameChangedEvent(void)
- const char *getapplicationStateChangedEvent(void)
- const char *getcommitDataRequestEvent(void)
- const char *getfocusObjectChangedEvent(void)
- const char *getfocusWindowChangedEvent(void)
- const char *getfontDatabaseChangedEvent(void)
- const char *getlastWindowClosedEvent(void)
- const char *getlayoutDirectionChangedEvent(void)
- const char *getpaletteChangedEvent(void)
- const char *getprimaryScreenChangedEvent(void)
- const char *getsaveStateRequestEvent(void)
- const char *getscreenAddedEvent(void)
- const char *getscreenRemovedEvent(void)

91.136 QHBarModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qhbarmodelmapper.html

Parameters : QObject *

Parent Class : QObject

- int columnCount(void)
- int firstBarSetRow(void)
- int firstColumn(void)
- int lastBarSetRow(void)
- QAbstractItemModel * model(void)
- QAbstractBarSeries * series(void)
- void setColumnCount(int columnCount)
- void setFirstBarSetRow(int firstBarSetRow)
- void setFirstColumn(int firstColumn)
- void setLastBarSetRow(int lastBarSetRow)
- void setModel(QAbstractItemModel *model)

- void setSeries(QAbstractBarSeries *series)
- void setcolumnCountChangedEvent(const char *)
- void setfirstBarSetRowChangedEvent(const char *)
- void setfirstColumnChangedEvent(const char *)
- void setlastBarSetRowChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- const char *getcolumnCountChangedEvent(void)
- const char *getfirstBarSetRowChangedEvent(void)
- const char *getfirstColumnChangedEvent(void)
- const char *getlastBarSetRowChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getseriesReplacedEvent(void)

91.137 QHBoxLayout Class

C++ Reference : http://doc.qt.io/qt-5/qhboxlayout.html

Parameters : void

Parent Class : QBoxLayout

- void addWidget(QWidget *)
- void addLayout(QLayout *)

91.138 QHBoxPlotModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qhboxplotmodelmapper.html

Parameters : QObject *

Parent Class : QObject

- int columnCount(void)
- int firstBoxSetRow(void)
- int firstColumn(void)
- int lastBoxSetRow(void)
- QAbstractItemModel * model(void)
- QBoxPlotSeries * series(void)
- void setColumnCount(int rowCount)
- void setFirstBoxSetRow(int firstBoxSetRow)
- void setFirstColumn(int firstColumn)
- void setLastBoxSetRow(int lastBoxSetRow)

- void setModel(QAbstractItemModel *model)
- void setSeries(QBoxPlotSeries *series)
- void setcolumnCountChangedEvent(const char *)
- void setfirstBoxSetRowChangedEvent(const char *)
- void setfirstColumnChangedEvent(const char *)
- void setlastBoxSetRowChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- const char *getcolumnCountChangedEvent(void)
- const char *getfirstBoxSetRowChangedEvent(void)
- const char *getfirstColumnChangedEvent(void)
- const char *getlastBoxSetRowChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getseriesReplacedEvent(void)

91.139 QHCandlestickModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qhcandlestickmodelmapper.html

Parameters : QObject *

Parent Class : QCandlestickModelMapper

- int closeColumn(void)
- int firstSetRow(void)
- int highColumn(void)
- int lastSetRow(void)
- int lowColumn(void)
- int openColumn(void)
- void setCloseColumn(int closeColumn)
- void setFirstSetRow(int firstSetRow)
- void setHighColumn(int highColumn)
- void setLastSetRow(int lastSetRow)
- void setLowColumn(int lowColumn)
- void setOpenColumn(int openColumn)
- void setTimestampColumn(int timestampColumn)
- int timestampColumn(void)
- void setcloseColumnChangedEvent(const char *)
- void setfirstSetRowChangedEvent(const char *)

- void sethighColumnChangedEvent(const char *)
- void setlastSetRowChangedEvent(const char *)
- void setlowColumnChangedEvent(const char *)
- void setopenColumnChangedEvent(const char *)
- void settimestampColumnChangedEvent(const char *)
- const char *getcloseColumnChangedEvent(void)
- const char *getfirstSetRowChangedEvent(void)
- const char *gethighColumnChangedEvent(void)
- const char *getlastSetRowChangedEvent(void)
- const char *getlowColumnChangedEvent(void)
- const char *getopenColumnChangedEvent(void)
- const char *gettimestampColumnChangedEvent(void)

91.140 QHPieModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qhpiemodelmapper.html

Parameters : QObject *

Parent Class : QPieModelMapper

- int columnCount(void)
- int firstColumn(void)
- int labelsRow(void)
- QAbstractItemModel * model(void)
- QPieSeries * series(void)
- void setColumnCount(int columnCount)
- void setFirstColumn(int firstColumn)
- void setLabelsRow(int labelsRow)
- void setModel(QAbstractItemModel *model)
- void setSeries(QPieSeries *series)
- void setValuesRow(int valuesRow)
- int valuesRow(void)
- void setcolumnCountChangedEvent(const char *)
- void setfirstColumnChangedEvent(const char *)
- void setlabelsRowChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- void setvaluesRowChangedEvent(const char *)

- const char *getcolumnCountChangedEvent(void)
- const char *getfirstColumnChangedEvent(void)
- const char *getlabelsRowChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getseriesReplacedEvent(void)
- const char *getvaluesRowChangedEvent(void)

91.141 QHXYModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qhxymodelmapper.html

Parameters : QObject *

Parent Class : QXYModelMapper

- int columnCount(void)
- int firstColumn(void)
- QAbstractItemModel * model(void)
- QXYSeries * series(void)
- void setColumnCount(int columnCount)
- void setFirstColumn(int firstColumn)
- void setModel(QAbstractItemModel *model)
- void setSeries(QXYSeries *series)
- void setXRow(int xRow)
- void setYRow(int yRow)
- int xRow(void)
- int yRow(void)
- void setcolumnCountChangedEvent(const char *)
- void setfirstColumnChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- void setxRowChangedEvent(const char *)
- void setyRowChangedEvent(const char *)
- const char *getcolumnCountChangedEvent(void)
- const char *getfirstColumnChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getseriesReplacedEvent(void)
- const char *getxRowChangedEvent(void)
- const char *getyRowChangedEvent(void)

91.142 QHeaderView Class

C++ Reference : http://doc.qt.io/qt-5/qheaderview.html

Parameters : Qt::Orientation, QWidget *

Parent Class : QAbstractItemView

- bool cascadingSectionResizes(void)
- int count(void)
- Qt::Alignment defaultAlignment(void)
- int defaultSectionSize(void)
- int hiddenSectionCount(void)
- void hideSection(int logicalIndex)
- bool highlightSections(void)
- bool isSectionHidden(int logicalIndex)
- bool isSortIndicatorShown(void)
- int length(void)
- int logicalIndex(int visualIndex)
- int logicalIndexAt(int position)
- int logicalIndexAt_2(int x, int y)
- int logicalIndexAt_3(QPoint pos)
- int maximumSectionSize(void)
- int minimumSectionSize_2(void)
- void moveSection(int from, int to)
- int offset(void)
- Qt::Orientation orientation(void)
- int resizeContentsPrecision(void)
- void resizeSection(int logicalIndex, int size)
- void resizeSections(QHeaderView::ResizeMode mode)
- bool restoreState(QByteArray state)
- QByteArray saveState(void)
- int sectionPosition(int logicalIndex)
- QHeaderView::ResizeMode sectionResizeMode(int logicalIndex)
- int sectionSize(int logicalIndex)
- int sectionSizeHint(int logicalIndex)
- int sectionViewportPosition(int logicalIndex)
- bool sectionsClickable(void)
- bool sectionsHidden(void)

- bool sectionsMovable(void)
- bool sectionsMoved(void)
- void setCascadingSectionResizes(bool enable)
- void setDefaultAlignment(Qt::Alignment alignment)
- void setDefaultSectionSize(int size)
- void setHighlightSections(bool highlight)
- void setMaximumSectionSize(int size)
- void setMinimumSectionSize(int size)
- void setResizeContentsPrecision(int precision)
- void setSectionHidden(int logicalIndex, bool hide)
- void setSectionResizeMode(QHeaderView::ResizeMode mode)
- void setSectionResizeMode_2(int logicalIndex, QHeaderView::ResizeMode mode)
- void setSectionsClickable(bool clickable)
- void setSectionsMovable(bool movable)
- void setSortIndicator(int logicalIndex, Qt::SortOrder order)
- void setSortIndicatorShown(bool show)
- void setStretchLastSection(bool stretch)
- void showSection(int logicalIndex)
- Qt::SortOrder sortIndicatorOrder(void)
- int sortIndicatorSection(void)
- bool stretchLastSection(void)
- int stretchSectionCount(void)
- void swapSections(int first, int second)
- int visualIndex(int logicalIndex)
- int visualIndexAt(int position)
- void headerDataChanged(Qt::Orientation orientation, int logicalFirst, int logicalLast)
- void setOffset(int offset)
- void setOffsetToLastSection(void)
- void setOffsetToSectionPosition(int visualSectionNumber)
- void setgeometriesChangedEvent(const char *)
- void setsectionClickedEvent(const char *)
- void setsectionCountChangedEvent(const char *)
- void setsectionDoubleClickedEvent(const char *)
- void setsectionEnteredEvent(const char *)
- void setsectionHandleDoubleClickedEvent(const char *)
- void setsectionMovedEvent(const char *)

- void setsectionPressedEvent(const char *)
- void setsectionResizedEvent(const char *)
- void setsortIndicatorChangedEvent(const char *)
- const char *getgeometriesChangedEvent(void)
- const char *getsectionClickedEvent(void)
- const char *getsectionCountChangedEvent(void)
- const char *getsectionDoubleClickedEvent(void)
- const char *getsectionEnteredEvent(void)
- const char *getsectionHandleDoubleClickedEvent(void)
- const char *getsectionMovedEvent(void)
- const char *getsectionPressedEvent(void)
- const char *getsectionResizedEvent(void)
- const char *getsortIndicatorChangedEvent(void)
- void geteventparameters(void)

91.143 QHorizontalBarSeries Class

C++ Reference : http://doc.qt.io/qt-5/qhorizontalbarseries.html

Parameters : QObject *

Parent Class : QAbstractBarSeries

- QAbstractSeries::SeriesType type(void)

91.144 QHorizontalPercentBarSeries Class

C++ Reference : http://doc.qt.io/qt-5/qhorizontalpercentbarseries.html

Parameters : QObject *

Parent Class : QAbstractBarSeries

- QAbstractSeries::SeriesType type(void)

91.145 QHorizontalStackedBarSeries Class

C++ Reference : http://doc.qt.io/qt-5/qhorizontalstackedbarseries.html

Parameters : QObject *

Parent Class : QAbstractBarSeries

- QAbstractSeries::SeriesType type(void)

91.146 QHostAddress Class

C++ Reference : http://doc.qt.io/qt-5/qhostaddress.html

Parameters : void

- void clear(void)
- bool isInSubnet(QHostAddress, int netmask)
- bool isNull(void)
- int protocol(void)
- QString scopeId(void)
- bool setAddress(QString)
- int toIPv4Address(void)
- Q_IPV6ADDR toIPv6Address(void)
- QString toString(void)

91.147 QHostInfo Class

C++ Reference : http://doc.qt.io/qt-5/qhostinfo.html

Parameters : void

- int error(void)
- QString errorString(void)
- QString hostName(void)
- int lookupId(void)
- void setError(QHostInfo::HostInfoError error)
- void setErrorString(QString)
- void setHostName(QString)
- void setLookupId(int id)
- void abortHostLookup(int id)
- QHostInfo fromName(QString)
- QString localDomainName(void)
- QString localHostName(void)

91.148 QIODevice Class

C++ Reference : http://doc.qt.io/qt-5/qiodevice.html

Parameters : void

Parent Class : QObject

- QString errorString(void)
- bool getChar(char *c)
- bool isOpen(void)
- bool isReadable(void)
- bool isTextModeEnabled(void)
- bool isWritable(void)
- int openMode(void)
- int peek(char *data, int maxSize)
- int read(char *data, int maxSize)
- int readLine(char *data, int maxSize)
- void ungetChar(char c)
- int write(const char *data, int maxSize)
- bool atEnd(void)
- bool canReadLine(void)
- void close(void)
- bool open(QIODevice::OpenMode flags)
- qint64 pos(void)
- bool seek(qint64 pos)
- qint64 size(void)
- void setaboutToCloseEvent(const char *)
- void setbytesWrittenEvent(const char *)
- void setreadChannelFinishedEvent(const char *)
- void setreadyReadEvent(const char *)
- const char *getaboutToCloseEvent(void)
- const char *getbytesWrittenEvent(void)
- const char *getreadChannelFinishedEvent(void)
- const char *getreadyReadEvent(void)

91.149 QIcon Class

C++ Reference : http://doc.qt.io/qt-5/qicon.html

Parameters : QPixmap

91.150 QImage Class

C++ Reference : http://doc.qt.io/qt-5/qimage.html

Parameters : void

- bool allGray(void)
- int bitPlaneCount(void)
- uchar *bits(void)
- int bytesPerLine(void)
- qint64 cacheKey(void)
- QRgb color(int i)
- int colorCount(void)
- const uchar *constBits(void)
- const uchar *constScanLine(int i)
- QImage convertToFormat(QImage::Format format, Qt::ImageConversionFlags flags)
- QImage copy(int x, int y, int width, int height)
- QImage createAlphaMask(Qt::ImageConversionFlags flags)
- QImage createHeuristicMask(bool clipTight)
- QImage createMaskFromColor(QRgb color, Qt::MaskMode mode)
- int depth(void)
- int dotsPerMeterX(void)
- int dotsPerMeterY(void)
- void fill(QColor)
- QImage::Format format(void)
- bool hasAlphaChannel(void)
- int height(void)
- void invertPixels(QImage::InvertMode mode)
- bool isGrayscale(void)
- bool isNull(void)
- bool load(QString, const char *format) # In RingQt use : bool loadimage(QString, const char *format)
- bool loadFromData(QByteArray, const char * format)
- QImage mirrored(bool horizontal, bool vertical)
- QPoint offset(void)

- QRgb pixel(int x, int y)
- int pixelIndex(int x, int y)
- QRect rect(void)
- QImage rgbSwapped(void)
- bool save(QString, const char * format, int quality)
- QImage scaled(int width, int height, Qt::AspectRatioMode aspectRatioMode, Qt::TransformationMode transformMode)
- QImage scaledToHeight(int height, Qt::TransformationMode mode)
- QImage scaledToWidth(int width, Qt::TransformationMode mode)
- uchar *scanLine(int i)
- void setColor(int index, QRgb colorValue)
- void setColorCount(int colorCount)
- void setDotsPerMeterX(int x)
- void setDotsPerMeterY(int y)
- void setOffset(QPoint)
- void setPixel(int x, int y, uint index_or_rgb)
- void setText(QString,QString)
- QSize size(void)
- void swap(QImage)
- QString text(QString)
- QStringList textKeys(void)
- bool valid(int x, int y)
- int width(void)

91.151 QInputAspect Class

C++ Reference : http://doc.qt.io/qt-5/qt3dinput-qinputaspect.html

Parameters : QObject *

- QStringList availablePhysicalDevices(void)
- Qt3DInput::QAbstractPhysicalDevice * createPhysicalDevice(QString name)

91.152 QInputDialog Class

C++ Reference : http://doc.qt.io/qt-5/qinputdialog.html

Parameters : QWidget *

Parent Class : QDialog

- QString cancelButtonText(void)
- QStringList comboBoxItems(void)
- int doubleDecimals(void)
- double doubleMaximum(void)
- double doubleMinimum(void)
- double doubleValue(void)
- int inputMode(void)
- int intMaximum(void)
- int intMinimum(void)
- int intStep(void)
- int intValue(void)
- bool isComboBoxEditable(void)
- QString labelText(void)
- QString okButtonText(void)
- void open(QObject *receiver, const char *member)
- int options(void)
- void setCancelButtonText(QString)
- void setComboBoxEditable(bool editable)
- void setComboBoxItems(QStringList)
- void setDoubleDecimals(int decimals)
- void setDoubleMaximum(double max)
- void setDoubleMinimum(double min)
- void setDoubleRange(double min, double max)
- void setDoubleValue(double value)
- void setInputMode(QInputDialog::InputMode mode)
- void setIntMaximum(int max)
- void setIntMinimum(int min)
- void setIntRange(int min, int max)
- void setIntStep(int step)
- void setIntValue(int value)
- void setLabelText(QString)

- void setOkButtonText(QString)
- void setOption(QInputDialog::InputDialogOption option, bool on)
- void setOptions(QInputDialog::InputDialogOption options)
- void setTextEchoMode(QLineEdit::EchoMode mode)
- void setTextValue(QString)
- bool testOption(QInputDialog::InputDialogOption option)
- int textEchoMode(void)
- QString textValue(void)
- double getDouble(QWidget *parent,QString,QString, double value, double min, double max , int decimals, bool *ok, Qt::WindowType flags)
- int getInt(QWidget *parent,QString,QString, int value, int min, int max, int step, bool *ok, Qt::WindowType flags)

91.153 QJsonArray Class

C++ Reference : http://doc.qt.io/qt-5/qjsonarray.html

Parameters : void

- void append(QJsonValue value)
- QJsonValue at(int i)
- bool contains(QJsonValue value)
- int count(void)
- bool empty(void)
- QJsonValue first(void)
- void insert(int i, QJsonValue value)
- bool isEmpty(void)
- QJsonValue last(void)
- void pop_back(void)
- void pop_front(void)
- void prepend(QJsonValue value)
- void push_back(QJsonValue value)
- void push_front(QJsonValue value)
- void removeAt(int i)
- void removeFirst(void)
- void removeLast(void)
- void replace(int i, QJsonValue value)
- int size(void)
- QJsonValue takeAt(int i)

- QVariantList toVariantList(void)
- QJsonArray fromStringList(QStringList list)
- QJsonArray fromVariantList(QVariantList list)

91.154 QJsonDocument Class

C++ Reference : http://doc.qt.io/qt-5/qjsondocument.html

Parameters : void

- QJsonArray array(void)
- bool isArray(void)
- bool isEmpty(void)
- bool isNull(void)
- bool isObject(void)
- QJsonObject object(void)
- void setArray(QJsonArray array)
- void setObject(QJsonObject object)
- QByteArray toJson(QJsonDocument::JsonFormat format)
- QVariant toVariant(void)
- QJsonDocument fromJson(QByteArray json, QJsonParseError * error)
- QJsonDocument fromVariant(QVariant variant)

91.155 QJsonObject Class

C++ Reference : http://doc.qt.io/qt-5/qjsonobject.html

Parameters : void

- bool contains(QString key)
- int count(void)
- bool empty(void)
- bool isEmpty(void)
- QStringList keys(void)
- int length(void)
- void remove(QString key)
- int size(void)
- QJsonValue take(QString key)
- QVariantMap toVariantMap(void)
- QJsonValue value(QString key)
- QJsonObject fromVariantMap(QVariantMap map)

91.156 QJsonParseError Class

C++ Reference : http://doc.qt.io/qt-5/qjsonparseerror.html

Parameters : void

- QString errorString(void)

91.157 QJsonValue Class

C++ Reference : http://doc.qt.io/qt-5/qjsonvalue.html

Parameters : void

- bool isArray(void)
- bool isBool(void)
- bool isDouble(void)
- bool isNull(void)
- bool isObject(void)
- bool isString(void)
- bool isUndefined(void)
- QJsonArray toArray(QJsonArray defaultValue)
- QJsonArray toArray_2(void)
- bool toBool(bool defaultValue)
- double toDouble(double defaultValue)
- int toInt(int defaultValue)
- QJsonObject toObject(QJsonObject defaultValue)
- QJsonObject toObject_2(void)
- QString toString(QString defaultValue)
- QVariant toVariant(void)
- QJsonValue::Type type(void)
- QJsonValue fromVariant(QVariant variant)

91.158 QKeySequence Class

C++ Reference : http://doc.qt.io/qt-5/qkeysequence.html

Parameters : QString

91.159 QLCDNumber Class

C++ Reference : http://doc.qt.io/qt-5/qlcdnumber.html

Parameters : QWidget *

Parent Class : QFrame

- bool checkOverflow(double num)
- int digitCount(void)
- int intValue(void)
- int mode(void)
- int segmentStyle(void)
- void setDigitCount(int numDigits)
- void setMode(QLCDNumber::Mode)
- void setSegmentStyle(QLCDNumber::SegmentStyle)
- bool smallDecimalPoint(void)
- double value(void)
- void display(double)
- void setBinMode(void)
- void setDecMode(void)
- void setHexMode(void)
- void setOctMode(void)
- void setSmallDecimalPoint(bool)

91.160 QLabel Class

C++ Reference : http://doc.qt.io/qt-5/qlabel.html

Parameters : QWidget *

Parent Class : QFrame

- QPicture *picture(void)
- QPixmap *pixmap(void)
- int alignment(void)
- QWidget *buddy(void)
- bool hasScaledContents(void)
- bool hasSelectedText(void)
- int indent(void)
- int margin(void)
- QMovie *movie(void)
- bool openExternalLinks(void)

- QString selectedText(void)
- int selectionStart(void)
- void setAlignment(Qt::AlignmentFlag)
- void setBuddy(QWidget *buddy)
- void setIndent(int)
- void setMargin(int)
- void setOpenExternalLinks(bool open)
- void setScaledContents(bool)
- void setSelection(int start, int length)
- void setTextFormat(Qt::TextFormat)
- void setTextInteractionFlags(Qt::TextInteractionFlag flags)
- void setWordWrap(bool on)
- QString text(void)
- int textFormat(void)
- int textInteractionFlags(void)
- bool wordWrap(void)
- void clear(void)
- void setMovie(QMovie *movie)
- void setNum(double num)
- void setPicture(QPicture)
- void setPixmap(QPixmap)
- void setText(QString)

91.161 QLayout Class

C++ Reference : http://doc.qt.io/qt-5/qlayout.html

Parameters : QWidget *

Parent Class : QObject

- bool activate(void)
- void addWidget(QWidget *w)
- QMargins contentsMargins(void)
- QRect contentsRect(void)
- void getContentsMargins(int *left, int *top, int *right, int *bottom)
- bool isEnabled(void)
- QWidget *menuBar(void)
- QWidget *parentWidget(void)

- void removeItem(QLayoutItem *item)
- void removeWidget(QWidget *widget)
- bool setAlignment(QWidget *w, Qt::Alignment alignment)
- void setAlignment_2(Qt::Alignment alignment)
- bool setAlignment_3(QLayout *l, Qt::Alignment alignment)
- void setContentsMargins(int left, int top, int right, int bottom)
- void setContentsMargins_2(QMargins margins)
- void setEnabled(bool enable)
- void setMenuBar(QWidget *widget)
- void setSizeConstraint(QLayout::SizeConstraint)
- void setSpacing(int)
- QLayout::SizeConstraint sizeConstraint(void)
- int spacing(void)
- void update(void)
- QSize closestAcceptableSize(QWidget * widget, QSize size)

91.162 QLegend Class

C++ Reference : http://doc.qt.io/qt-5/qlegend.html

Parent Class : QGraphicsWidget

- Qt::Alignment alignment(void)
- void attachToChart(void)
- QColor borderColor(void)
- QBrush brush(void)
- QColor color(void)
- void detachFromChart(void)
- QFont font(void)
- bool isAttachedToChart(void)
- bool isBackgroundVisible(void)
- QBrush labelBrush(void)
- QColor labelColor(void)
- QLegend::MarkerShape markerShape(void)
- QList<QLegendMarker *> markers(QAbstractSeries *series)
- QPen pen(void)
- bool reverseMarkers(void)
- void setAlignment(Qt::Alignment alignment)

- void setBackgroundVisible(bool visible)
- void setBorderColor(QColor color)
- void setBrush(QBrush brush)
- void setColor(QColor color)
- void setFont(QFont font)
- void setLabelBrush(QBrush brush)
- void setLabelColor(QColor color)
- void setMarkerShape(QLegend::MarkerShape shape)
- void setPen(QPen pen)
- void setReverseMarkers(bool reverseMarkers)
- void setShowToolTips(bool show)
- bool showToolTips(void)

91.163 QLegendMarker Class

C++ Reference : http://doc.qt.io/qt-5/qlegendmarker.html

Parameters : void

Parent Class : QObject

- QBrush brush(void)
- QFont font(void)
- bool isVisible(void)
- QString label(void)
- QBrush labelBrush(void)
- QPen pen(void)
- QAbstractSeries * series(void)
- void setBrush(QBrush brush)
- void setFont(QFont font)
- void setLabel(QString label)
- void setLabelBrush(QBrush brush)
- void setPen(QPen pen)
- void setShape(QLegend::MarkerShape shape)
- void setVisible(bool visible)
- QLegend::MarkerShape shape(void)
- void setbrushChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setfontChangedEvent(const char *)

- void sethoveredEvent(const char *)
- void setlabelBrushChangedEvent(const char *)
- void setlabelChangedEvent(const char *)
- void setpenChangedEvent(const char *)
- void setshapeChangedEvent(const char *)
- void setvisibleChangedEvent(const char *)
- const char *getbrushChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getfontChangedEvent(void)
- const char *gethoveredEvent(void)
- const char *getlabelBrushChangedEvent(void)
- const char *getlabelChangedEvent(void)
- const char *getpenChangedEvent(void)
- const char *getshapeChangedEvent(void)
- const char *getvisibleChangedEvent(void)

91.164 QLineEdit Class

C++ Reference : http://doc.qt.io/qt-5/qlineedit.html

Parameters : QWidget *

Parent Class : QWidget

- int alignment(void)
- void backspace(void)
- QCompleter *completer(void)
- QMenu *createStandardContextMenu(void)
- void cursorBackward(bool mark, int steps)
- void cursorForward(bool mark, int steps)
- int cursorMoveStyle(void)
- int cursorPosition(void)
- int cursorPositionAt(QPoint)
- void cursorWordBackward(bool mark)
- void cursorWordForward(bool mark)
- void del(void)
- void deselect(void)
- QString displayText(void)
- bool dragEnabled(void)

- int echoMode(void)
- void end(bool mark) # In RingQt use : void endtext(bool mark)
- bool hasAcceptableInput(void)
- bool hasFrame(void)
- bool hasSelectedText(void)
- void home(bool mark)
- QString inputMask(void)
- void insert(QString)
- bool isModified(void)
- bool isReadOnly(void)
- bool isRedoAvailable(void)
- bool isUndoAvailable(void)
- int maxLength(void)
- QString placeholderText(void)
- QString selectedText(void)
- int selectionStart(void)
- void setAlignment(Qt::AlignmentFlag flag)
- void setCompleter(QCompleter *c)
- void setCursorMoveStyle(Qt::CursorMoveStyle style)
- void setCursorPosition(int)
- void setDragEnabled(bool b)
- void setEchoMode(QLineEdit::EchoMode)
- void setFrame(bool)
- void setInputMask(QString)
- void setMaxLength(int)
- void setModified(bool)
- void setPlaceholderText(QString)
- void setReadOnly(bool)
- void setSelection(int start, int length)
- void setTextMargins(int left, int top, int right, int bottom)
- void setValidator(QValidator *v)
- QString text(void)
- QMargins textMargins(void)
- QValidator *validator(void)
- void clear(void)
- void copy(void)

- void cut(void)
- void paste(void)
- void redo(void)
- void selectAll(void)
- void setText(QString)
- void undo(void)
- void setTextChangedEvent(const char *)
- void setcursorPositionChangedEvent(const char *)
- void seteditingFinishedEvent(const char *)
- void setreturnPressedEvent(const char *)
- void setselectionChangedEvent(const char *)
- void settextEditedEvent(const char *)
- const char *getTextChangedEvent(void)
- const char *getcursorPositionChangedEvent(void)
- const char *geteditingFinishedEvent(void)
- const char *getreturnPressedEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettextEditedEvent(void)

91.165 QLineSeries Class

C++ Reference : http://doc.qt.io/qt-5/qlineseries.html

Parameters : QObject *

Parent Class : QXYSeries

- QAbstractSeries::SeriesType type(void)

91.166 QLinearGradient Class

C++ Reference : http://doc.qt.io/qt-5/qlineargradient.html

Parameters : void

Parent Class : QGradient

- QPointF finalStop(void)
- void setFinalStop(QPointF stop)
- void setFinalStop_2(qreal x,qreal y)
- void setStart(QPointF start)
- void setStart_2(qreal x,qreal y)
- QPointF start(void)

91.167 QListView Class

C++ Reference : http://doc.qt.io/qt-5/qlistview.html

Parameters : QWidget *

Parent Class : QAbstractItemView

- int batchSize(void)
- void clearPropertyFlags(void)
- QListView::Flow flow(void)
- QSize gridSize(void)
- bool isRowHidden(int row)
- bool isSelectionRectVisible(void)
- bool isWrapping(void)
- QListView::LayoutMode layoutMode(void)
- int modelColumn(void)
- QListView::Movement movement(void)
- QListView::ResizeMode resizeMode(void)
- void setBatchSize(int batchSize)
- void setFlow(QListView::Flow flow)
- void setGridSize(QSize size)
- void setLayoutMode(QListView::LayoutMode mode)
- void setModelColumn(int column)
- void setMovement(QListView::Movement movement)
- void setResizeMode(QListView::ResizeMode mode)
- void setRowHidden(int row, bool hide)
- void setSelectionRectVisible(bool show)
- void setSpacing(int space)
- void setUniformItemSizes(bool enable)
- void setViewMode(QListView::ViewMode mode)
- void setWordWrap(bool on)
- void setWrapping(bool enable)
- int spacing(void)
- bool uniformItemSizes(void)
- QListView::ViewMode viewMode(void)
- bool wordWrap(void)

91.168 QListWidget Class

C++ Reference : http://doc.qt.io/qt-5/qlistwidget.html

Parameters : QWidget *

Parent Class : QListView

- void addItem(QString)
- void editItem(QListWidgetItem *item)
- bool isSortingEnabled(void)
- QListWidgetItem *item(int row)
- QListWidgetItem *itemAt(int x, int y)
- QWidget *itemWidget(QListWidgetItem *item)
- void openPersistentEditor(QListWidgetItem *item)
- void removeItemWidget(QListWidgetItem *item)
- int row(QListWidgetItem *item)
- void setCurrentRow(int row, QItemSelectionModel::SelectionFlag command)
- void setItemWidget(QListWidgetItem *item, QWidget *widget)
- void setSortingEnabled(bool enable)
- void sortItems(Qt::SortOrder order)
- QListWidgetItem *takeItem(int row)
- QRect visualItemRect(QListWidgetItem *item)
- void clear(void)
- void scrollToItem(QListWidgetItem *item,QAbstractItemView::ScrollHint hint)
- void setcurrentItemChangedEvent(const char *)
- void setcurrentRowChangedEvent(const char *)
- void setcurrentTextChangedEvent(const char *)
- void setitemActivatedEvent(const char *)
- void setitemChangedEvent(const char *)
- void setitemClickedEvent(const char *)
- void setitemDoubleClickedEvent(const char *)
- void setitemEnteredEvent(const char *)
- void setitemPressedEvent(const char *)
- void setitemSelectionChangedEvent(const char *)
- const char *getcurrentItemChangedEvent(void)
- const char *getcurrentRowChangedEvent(void)
- const char *getcurrentTextChangedEvent(void)
- const char *getitemActivatedEvent(void)

- const char *getitemChangedEvent(void)
- const char *getitemClickedEvent(void)
- const char *getitemDoubleClickedEvent(void)
- const char *getitemEnteredEvent(void)
- const char *getitemPressedEvent(void)
- const char *getitemSelectionChangedEvent(void)

91.169 QListWidgetItem Class

C++ Reference : http://doc.qt.io/qt-5/qlistwidgetitem.html

Parameters : void

- QBrush background(void)
- Qt::CheckState checkState(void)
- Qt::ItemFlags flags(void)
- QFont font(void)
- QBrush foreground(void)
- QIcon icon(void)
- bool isHidden(void)
- bool isSelected(void)
- QListWidget *listWidget(void)
- void setBackground(QBrush brush)
- void setCheckState(Qt::CheckState state)
- void setFlags(Qt::ItemFlags flags)
- void setFont(QFont font)
- void setForeground(QBrush brush)
- void setHidden(bool hide)
- void setIcon(QIcon icon)
- void setSelected(bool select)
- void setSizeHint(QSize size)
- void setStatusTip(QString statusTip)
- void setText(QString text)
- void setTextAlignment(int alignment)
- void setToolTip(QString toolTip)
- void setWhatsThis(QString whatsThis)
- QSize sizeHint(void)
- QString statusTip(void)

- QString text(void)
- int textAlignment(void)
- QString toolTip(void)
- int type(void)
- QString whatsThis(void)

91.170 QLocale Class

C++ Reference : http://doc.qt.io/qt-5/qlocale.html

Parameters : QString

- QString amText(void)
- QString bcp47Name(void)
- QString createSeparatedList(QStringList list)
- QString currencySymbol(QLocale::CurrencySymbolFormat format)
- QString dateFormat(QLocale::FormatType format)
- QString dateTimeFormat(QLocale::FormatType format)
- QString dayName(int day, QLocale::FormatType type)
- QChar decimalPoint(void)
- QChar exponential(void)
- Qt::DayOfWeek firstDayOfWeek(void)
- QLocale::Language language(void)
- QLocale::MeasurementSystem measurementSystem(void)
- QString monthName(int month, QLocale::FormatType type)
- QString name(void)
- QString nativeCountryName(void)
- QString nativeLanguageName(void)
- QChar negativeSign(void)
- QLocale::NumberOptions numberOptions(void)
- QChar percent(void)
- QString pmText(void)
- QChar positiveSign(void)
- QString quoteString(QString str, QLocale::QuotationStyle style)
- QString quoteString_2(QStringRef str, QLocale::QuotationStyle style)
- QLocale::Script script(void)
- void setNumberOptions(QLocale::NumberOptions options)
- QString standaloneDayName(int day, QLocale::FormatType type)

- QString standaloneMonthName(int month, QLocale::FormatType type)
- Qt::LayoutDirection textDirection(void)
- QString timeFormat(QLocale::FormatType format)
- double toDouble_2(QStringRef s, bool *ok)
- float toFloat(QString s, bool *ok)
- float toFloat_2(QStringRef s, bool *ok)
- int toInt(QString s, bool *ok)
- int toInt_2(QStringRef s, bool *ok)
- qlonglong toLongLong(QString s, bool *ok)
- qlonglong toLongLong_2(QStringRef s, bool *ok)
- QString toLower(QString str)
- short toShort(QString s, bool *ok)
- short toShort_2(QStringRef s, bool *ok)
- QString toString(qlonglong i)
- QString toString_2(qulonglong i)
- QString toString_4(short i)
- QString toString_5(ushort i)
- QString toString_6(int i)
- QString toString_7(uint i)
- QString toString_8(double i, char f, int prec)
- QString toString_9(float i, char f, int prec)
- QString toString_10(QDate date, QString format)
- QString toString_11(QTime time, QString format)
- QString toString_12(QDateTime dateTime, QString format)
- QString toString_13(QDate date, QLocale::FormatType format)
- QString toString_14(QTime time, QLocale::FormatType format)
- QString toString_15(QDateTime dateTime, QLocale::FormatType format)
- QTime toTime(QString string, QLocale::FormatType format)
- QTime toTime_2(QString string, QString format)
- uint toUInt(QString s, bool *ok)
- uint toUInt_2(QStringRef s, bool *ok)
- qulonglong toULongLong(QString s, bool *ok)
- qulonglong toULongLong_2(QStringRef s, bool *ok)
- ushort toUShort(QString s, bool *ok)
- ushort toUShort_2(QStringRef s, bool *ok)
- QString toUpper(QString str)

- QStringList uiLanguages(void)
- QList<Qt::DayOfWeek> weekdays(void)
- QChar zeroDigit(void)
- QLocale c(void)
- QString countryToString(QLocale::Country country)
- QString languageToString(QLocale::Language language)
- QList<QLocale> matchingLocales(QLocale::Language language, QLocale::Script script, QLocale::Country country)
- QString scriptToString(QLocale::Script script)
- void setDefault(QLocale locale)
- QLocale system(void)

91.171 QLogValueAxis Class

C++ Reference : http://doc.qt.io/qt-5/qlogvalueaxis.html

Parameters : QObject *

Parent Class : QAbstractAxis

- qreal base(void)
- QString labelFormat(void)
- qreal max(void)
- qreal min(void)
- int minorTickCount(void)
- void setBase(qreal base)
- void setLabelFormat(QString format)
- void setMax(qreal max)
- void setMin(qreal min)
- void setMinorTickCount(int minorTickCount)
- void setRange(qreal min, qreal max)
- int tickCount(void)
- void setbaseChangedEvent(const char *)
- void setlabelFormatChangedEvent(const char *)
- void setmaxChangedEvent(const char *)
- void setminChangedEvent(const char *)
- void setminorTickCountChangedEvent(const char *)
- void setrangeChangedEvent(const char *)
- void settickCountChangedEvent(const char *)
- const char *getbaseChangedEvent(void)

- const char *getlabelFormatChangedEvent(void)
- const char *getmaxChangedEvent(void)
- const char *getminChangedEvent(void)
- const char *getminorTickCountChangedEvent(void)
- const char *getrangeChangedEvent(void)
- const char *gettickCountChangedEvent(void)

91.172 QLogicAspect Class

C++ Reference : http://doc.qt.io/qt-5/qt3dlogic-qlogicaspect.html

Parameters : QObject *

Parent Class : Qt3DCore::QAbstractAspect

91.173 QMainWindow Class

C++ Reference : http://doc.qt.io/qt-5/qmainwindow.html

Parameters : void

Parent Class : QWidget

- void addDockWidget(Qt::DockWidgetArea area, QDockWidget *dockwidget, Qt::Orientation orientation)
- QToolBar *addToolBar(QString)
- void addToolBar_2(Qt::ToolBarArea area, QToolBar *toolbar)
- void addToolBarBreak(Qt::ToolBarArea)
- QWidget *centralWidget(void)
- int corner(Qt::Corner corner)
- QMenu *createPopupMenu(void)
- int dockOptions(void)
- int dockWidgetArea(QDockWidget *dockwidget)
- bool documentMode(void)
- QSize iconSize(void)
- void insertToolBar(QToolBar *before, QToolBar *toolbar)
- void insertToolBarBreak(QToolBar *before)
- bool isAnimated(void)
- bool isDockNestingEnabled(void)
- QMenuBar *menuBar(void)
- QWidget *menuWidget(void)
- void removeDockWidget(QDockWidget *dockwidget)
- void removeToolBar(QToolBar *toolbar)

- void removeToolBarBreak(QToolBar *before)
- bool restoreDockWidget(QDockWidget *dockwidget)
- bool restoreState(QByteArray state, int version)
- QByteArray saveState(int version)
- void setCentralWidget(QWidget *widget)
- void setCorner(Qt::Corner corner, Qt::DockWidgetArea area)
- void setDockOptions(QMainWindow::DockOption options)
- void setDocumentMode(bool enabled)
- void setIconSize(QSize)
- void setMenuBar(QMenuBar *menuBar)
- void setMenuWidget(QWidget *menuBar)
- void setStatusBar(QStatusBar *statusbar)
- void setTabPosition(Qt::DockWidgetArea areas, QTabWidget::TabPosition tabPosition)
- void setTabShape(QTabWidget::TabShape tabShape)
- void setToolButtonStyle(Qt::ToolButtonStyle toolButtonStyle)
- void setUnifiedTitleAndToolBarOnMac(bool set)
- void splitDockWidget(QDockWidget *first, QDockWidget *second, Qt::Orientation orientation)
- QStatusBar *statusBar(void)
- int tabPosition(Qt::DockWidgetArea area)
- int tabShape(void)
- void tabifyDockWidget(QDockWidget *first, QDockWidget *second)
- int toolBarArea(QToolBar *toolbar)
- bool toolBarBreak(QToolBar *toolbar)
- int toolButtonStyle(void)
- bool unifiedTitleAndToolBarOnMac(void)

91.174 QMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qmaterial.html

Parameters : Qt3DCore::QNode *

Parent Class : Qt3DCore::QComponent

- void addParameter(Qt3DRender::QParameter *parameter)
- Qt3DRender::QEffect * effect(void)
- QVector<Qt3DRender::QParameter *> parameters(void)
- void removeParameter(Qt3DRender::QParameter *parameter)
- void setEffect(Qt3DRender::QEffect *effect)

91.175 QMatrix4x4 Class

Parameters : qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal,qreal

- QVector4D column(int index)
- qreal * constData(void)
- qreal * data_2(void)
- qreal determinant(void)
- void fill(qreal value)
- void flipCoordinates(void)
- void frustum(qreal left, qreal right, qreal bottom, qreal top, qreal nearPlane, qreal farPlane)
- QMatrix4x4 inverted(bool * invertible)
- bool isIdentity(void)
- void lookAt(QVector3D eye, QVector3D center, QVector3D up)
- QPoint map(QPoint point)
- QPointF map_2(QPointF point)
- QVector3D map_3(QVector3D point)
- QVector4D map_4(QVector4D point)
- QRect mapRect(QRect rect)
- QRectF mapRect_2(QRectF rect)
- QVector3D mapVector(QVector3D vector)
- QMatrix3x3 normalMatrix(void)
- void optimize(void)
- void ortho(qreal left, qreal right, qreal bottom, qreal top, qreal nearPlane, qreal farPlane)
- void ortho_2(QRect rect)
- void ortho_3(QRectF rect)
- void perspective(qreal angle, qreal aspect, qreal nearPlane, qreal farPlane)
- void rotate(qreal angle, QVector3D vector)
- void rotate_2(QQuaternion quaternion)
- void rotate_3(qreal angle, qreal x, qreal y, qreal z)
- QVector4D row(int index)
- void scale(QVector3D vector)
- void scale_2(qreal x, qreal y)
- void scale_3(qreal x, qreal y, qreal z)
- void scale_4(qreal factor)
- void setColumn(int index, QVector4D value)
- void setRow(int index, QVector4D value)

- void setToIdentity(void)
- QTransform toTransform_2(qreal distanceToPlane)
- void translate(QVector3D vector)
- void translate_2(qreal x, qreal y)
- void translate_3(qreal x, qreal y, qreal z)
- QMatrix4x4 transposed(void)

91.176 QMdiArea Class

C++ Reference : http://doc.qt.io/qt-5/qmdiarea.html

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- QMdiArea::WindowOrder activationOrder(void)
- QMdiSubWindow * activeSubWindow(void)
- QMdiSubWindow * addSubWindow(QWidget * widget, Qt::WindowFlags windowFlags)
- QBrush background(void)
- QMdiSubWindow * currentSubWindow(void)
- bool documentMode(void)
- void removeSubWindow(QWidget * widget)
- void setActivationOrder(QMdiArea::WindowOrder order)
- void setBackground(QBrush background)
- void setDocumentMode(bool enabled)
- void setOption(QMdiArea::AreaOption option, bool on)
- void setTabPosition(QTabWidget::TabPosition position)
- void setTabShape(QTabWidget::TabShape shape)
- void setTabsClosable(bool closable)
- void setTabsMovable(bool movable)
- void setViewMode(QMdiArea::ViewMode mode)
- QList<QMdiSubWindow *> subWindowList(QMdiArea::WindowOrder order)
- QTabWidget::TabPosition tabPosition(void)
- QTabWidget::TabShape tabShape(void)
- bool tabsClosable(void)
- bool tabsMovable(void)
- bool testOption(QMdiArea::AreaOption option)
- QMdiArea::ViewMode viewMode(void)
- void activateNextSubWindow(void)

- void activatePreviousSubWindow(void)
- void cascadeSubWindows(void)
- void closeActiveSubWindow(void)
- void closeAllSubWindows(void)
- void setActiveSubWindow(QMdiSubWindow * window)
- void tileSubWindows(void)

91.177 QMdiSubWindow Class

C++ Reference : http://doc.qt.io/qt-5/qmdisubwindow.html

Parameters : QWidget *

Parent Class : QWidget

- bool isShaded(void)
- int keyboardPageStep(void)
- int keyboardSingleStep(void)
- QMdiArea * mdiArea(void)
- void setKeyboardPageStep(int step)
- void setKeyboardSingleStep(int step)
- void setOption(QMdiSubWindow::SubWindowOption option, bool on)
- void setSystemMenu(QMenu * systemMenu)
- void setWidget(QWidget * widget)
- QMenu * systemMenu(void)
- bool testOption(QMdiSubWindow::SubWindowOption option)
- QWidget * widget(void)
- void showShaded(void)
- void showSystemMenu(void)

91.178 QMediaObject Class

C++ Reference : http://doc.qt.io/qt-5/qmediaobject.html

Parameters : void

Parent Class : QWidget

- QStringList availableMetaData(void)
- bool isMetaDataAvailable(void)
- QVariant metaData(QString key)
- int notifyInterval(void)
- void setNotifyInterval(int milliSeconds)

91.179 QMediaPlayer Class

C++ Reference : http://doc.qt.io/qt-5/qmediaplayer.html

Parameters : void

- int bufferStatus(void)
- QMediaContent currentMedia(void)
- int duration(void)
- int error(void)
- QString errorString(void)
- bool isAudioAvailable(void)
- bool isMuted(void)
- bool isSeekable(void)
- bool isVideoAvailable(void)
- QMediaContent media(void)
- int mediaStatus(void)
- QIODevice *mediaStream(void)
- qreal playbackRate(void)
- QMediaPlaylist *playlist(void)
- int position(void)
- void setVideoOutput(QVideoWidget *output)
- int state(void)
- int volume(void)
- void pause(void)
- void play(void)
- void setMedia(QUrl)
- void setMuted(bool muted)
- void setPlaybackRate(qreal rate)
- void setPlaylist(QMediaPlaylist *playlist)
- void setPosition(int position)
- void setVolume(int volume)
- void stop(void)

91.180 QMediaPlaylist Class

C++ Reference : http://doc.qt.io/qt-5/qmediaplaylist.html

Parameters : void

- int currentIndex(void)
- QMediaContent currentMedia(void)
- int error(void)
- QString errorString(void)
- bool insertMedia(int pos, QMediaContent)
- bool isReadOnly(void)
- QMediaContent media(int index)
- int mediaCount(void)
- int nextIndex(int steps)
- int playbackMode(void)
- int previousIndex(int steps)
- bool save(QUrl, const char * format)
- void next(void) # In RingQt use : void movenext(void)
- void previous(void)
- void setCurrentIndex(int playlistPosition)
- void shuffle(void)

91.181 QMenu Class

C++ Reference : http://doc.qt.io/qt-5/qmenu.html

Parameters : QWidget *

Parent Class : QWidget

- QAction *actionAt(QPoint)
- QRect actionGeometry(QAction *act)
- QAction *activeAction(void)
- void addAction(QAction *)
- QMenu *addMenu(QString)
- QAction *addSeparator(void)
- void clear(void)
- QAction *defaultAction(void)
- QAction *exec(const QPoint &)
- QAction *exec_2(void)
- QAction *exec_3(const QPoint &,QAction *)

- void hideTearOffMenu(void)
- QIcon icon(void)
- QAction *insertMenu(QAction *before, QMenu *menu)
- QAction *insertSeparator(QAction *before)
- bool isEmpty(void)
- bool isTearOffEnabled(void)
- bool isTearOffMenuVisible(void)
- QAction *menuAction(void)
- void popup(QPoint, QAction *atAction)
- bool separatorsCollapsible(void)
- void setActiveAction(QAction *act)
- void setDefaultAction(QAction *act)
- void setIcon(QIcon)
- void setSeparatorsCollapsible(bool collapse)
- void setTearOffEnabled(bool)
- void setTitle(QString)
- QString title(void)

91.182 QMenuBar Class

C++ Reference : http://doc.qt.io/qt-5/qmenubar.html

Parameters : QWidget *

Parent Class : QWidget

- QAction *actionAt(QPoint)
- QRect actionGeometry(QAction *act)
- QAction *activeAction(void)
- QAction *addAction(QString)
- QAction *addSeparator(void)
- void clear(void)
- QWidget *cornerWidget(Qt::Corner)
- QAction *insertSeparator(QAction *before)
- bool isDefaultUp(void)
- bool isNativeMenuBar(void)
- void setActiveAction(QAction *act)
- void setCornerWidget(QWidget *widget, Qt::Corner)
- void setNativeMenuBar(bool nativeMenuBar)

91.183 QMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qmesh.html

Parameters : Qt3DCore::QNode *

- QString meshName(void)
- QUrl source(void)
- Qt3DRender::QMesh::Status status(void)
- void setMeshName(QString meshName)
- void setSource(QUrl source)

91.184 QMessageBox Class

C++ Reference : http://doc.qt.io/qt-5/qmessagebox.html

Parameters : QWidget *parent

Parent Class : QDialog

- void addButton(QAbstractButton *button, QMessageBox::ButtonRole role)
- QAbstractButton *button(QMessageBox::StandardButton which)
- int buttonRole(QAbstractButton *button)
- QAbstractButton *clickedButton(void)
- QPushButton *defaultButton(void)
- QString detailedText(void)
- QAbstractButton *escapeButton(void)
- QPixmap iconPixmap(void)
- QString informativeText(void)
- void open(QObject *receiver, const char *member)
- void removeButton(QAbstractButton *button)
- void setDefaultButton(QPushButton *button)
- void setDetailedText(QString)
- void setEscapeButton(QAbstractButton *button)
- void setIcon(QMessageBox::Icon)
- void setIconPixmap(QPixmap)
- void setInformativeText(QString)
- void setStandardButtons(QMessageBox::StandardButton buttons)
- void setText(QString)
- void setTextFormat(Qt::TextFormat format)
- void setWindowModality(Qt::WindowModality windowModality)
- void setWindowTitle(QString)

- int standardButton(QAbstractButton *button)
- int standardButtons(void)
- QString text(void)
- int textFormat(void)
- int exec(void)
- void about(QWidget *parent, QString,QString)
- void aboutQt(QWidget *parent, QString)
- int critical(QWidget * parent, QString , QString, int buttons, int defaultButton)
- int information(QWidget * parent, QString ,QString, int buttons,int defaultButton)
- int question(QWidget * parent,QString,QString, int buttons ,int defaultButton)
- int warning(QWidget *parent, QString,QString, int buttons,int defaultButton)

91.185 QMetalRoughMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qmetalroughmaterial.html

Parameters : Qt3DCore::QNode *

- QVariant ambientOcclusion(void)
- QVariant baseColor(void)
- QVariant metalness(void)
- QVariant normal(void)
- QVariant roughness(void)
- float textureScale(void)
- void setAmbientOcclusion(QVariant ambientOcclusion)
- void setBaseColor(QVariant baseColor)
- void setMetalness(QVariant metalness)
- void setNormal(QVariant normal)
- void setRoughness(QVariant roughness)
- void setTextureScale(float textureScale)

91.186 QMimeData Class

C++ Reference : http://doc.qt.io/qt-5/qmimedata.html

Parameters : void

Parent Class : QObject

- void clear(void)
- QVariant colorData(void)
- QByteArray data(QString mimeType)

- QStringList formats(void)
- bool hasColor(void)
- bool hasFormat(QString mimeType)
- bool hasHtml(void)
- bool hasImage(void)
- bool hasText(void)
- bool hasUrls(void)
- QString html(void)
- QVariant imageData(void)
- void removeFormat(QString mimeType)
- void setColorData(QVariant color)
- void setData(QString mimeType, QByteArray data)
- void setHtml(QString html)
- void setImageData(QVariant image)
- void setText(QString text)
- void setUrls(QList<QUrl> urls)
- QString text(void)
- QList<QUrl> urls(void)

91.187 QMorphPhongMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qmorphphongmaterial.html

Parameters : Qt3DCore::QNode *

- QColor ambient(void)
- QColor diffuse(void)
- float interpolator(void)
- float shininess(void)
- QColor specular(void)
- void setAmbient(QColor ambient)
- void setDiffuse(QColor diffuse)
- void setInterpolator(float interpolator)
- void setShininess(float shininess)
- void setSpecular(QColor specular)

91.188 QMovie Class

C++ Reference : http://doc.qt.io/qt-5/qmovie.html

Parameters : QObject *

Parent Class : QObject

- QColor backgroundColor(void)
- QMovie::CacheMode cacheMode(void)
- int currentFrameNumber(void)
- QImage currentImage(void)
- QPixmap currentPixmap(void)
- QIODevice * device(void)
- QString fileName(void)
- QByteArray format(void)
- int frameCount(void)
- QRect frameRect(void)
- bool isValid(void)
- bool jumpToFrame(int frameNumber)
- int nextFrameDelay(void)
- QSize scaledSize(void)
- void setBackgroundColor(QColor color)
- void setCacheMode(QMovie::CacheMode mode)
- void setDevice(QIODevice *device)
- void setFileName(QString fileName)
- void setFormat(QByteArray format)
- void setScaledSize(QSize size)
- int speed(void)
- QMovie::MovieState state(void)
- bool jumpToNextFrame(void)
- void setPaused(bool paused)
- void setSpeed(int percentSpeed)
- void start(void)
- void stop(void)
- void seterrorEvent(const char *)
- void setfinishedEvent(const char *)
- void setframeChangedEvent(const char *)
- void setresizedEvent(const char *)

- void setstartedEvent(const char *)
- void setstateChangedEvent(const char *)
- void setupdatedEvent(const char *)
- const char *geterrorEvent(void)
- const char *getfinishedEvent(void)
- const char *getframeChangedEvent(void)
- const char *getresizedEvent(void)
- const char *getstartedEvent(void)
- const char *getstateChangedEvent(void)
- const char *getupdatedEvent(void)

91.189 QMutex Class

C++ Reference : http://doc.qt.io/qt-5/qmutex.html

Parameters : QMutex::RecursionMode

- bool isRecursive(void)
- void lock(void)
- void unlock(void)

91.190 QMutexLocker Class

C++ Reference : http://doc.qt.io/qt-5/qmutexlocker.html

Parameters : QMutex *

- QMutex * mutex(void)
- void relock(void)
- void unlock(void)

91.191 QNetworkAccessManager Class

C++ Reference : http://doc.qt.io/qt-5/qnetworkaccessmanager.html

Parameters : QObject *

Parent Class : QObject

- void setfinishedEvent(const char *)
- const char *getfinishedEvent(void)
- QNetworkConfiguration activeConfiguration(void)
- QAbstractNetworkCache *cache(void)
- void clearAccessCache(void)

- QNetworkConfiguration configuration(void)
- void connectToHost(QString, quint16)
- QNetworkReply *deleteResource(QNetworkRequest)
- QNetworkReply *get(QNetworkRequest) # In RingQt use : QNetworkReply *getvalue(QNetworkRequest)
- QNetworkReply *head(QNetworkRequest)
- QNetworkAccessManager::NetworkAccessibility networkAccessible(void)
- QNetworkReply *post(QNetworkRequest, QByteArray)
- QNetworkProxy proxy(void)
- QNetworkProxyFactory *proxyFactory(void)
- QNetworkReply *put(QNetworkRequest, QByteArray) # In RingQt use : QNetworkReply *putvalue(QNetworkRequest, QByteArray)
- QNetworkReply *sendCustomRequest(QNetworkRequest, QByteArray, QIODevice *)
- void setCache(QAbstractNetworkCache *cache)
- void setConfiguration(QNetworkConfiguration)
- void setCookieJar(QNetworkCookieJar *cookieJar)
- void setNetworkAccessible(QNetworkAccessManager::NetworkAccessibility accessible)
- void setProxy(QNetworkProxy)
- void setProxyFactory(QNetworkProxyFactory *factory)
- QStringList supportedSchemes(void)
- void geteventparameters(void)

91.192 QNetworkProxy Class

C++ Reference : http://doc.qt.io/qt-5/qnetworkproxy.html

Parameters : void

- int capabilities(void)
- bool hasRawHeader(QByteArray headerName)
- QVariant header(QNetworkRequest::KnownHeaders header)
- QString hostName(void)
- bool isCachingProxy(void)
- bool isTransparentProxy(void)
- QString password(void)
- int port(void)
- QByteArray rawHeader(QByteArray headerName)
- void setCapabilities(QNetworkProxy::Capability capabilities)
- void setHeader(QNetworkRequest::KnownHeaders header, QVariant value)
- void setHostName(QString hostName)

- void setPassword(QString password)
- void setPort(int port)
- void setRawHeader(QByteArray headerName, QByteArray headerValue)
- void setType(QNetworkProxy::ProxyType type)
- void setUser(QString user)
- void swap(QNetworkProxy other)
- int type(void)
- QString user(void)
- QNetworkProxy applicationProxy(void)
- void setApplicationProxy(QNetworkProxy networkProxy)

91.193 QNetworkReply Class

C++ Reference : http://doc.qt.io/qt-5/qnetworkreply.html

Parameters : void

Parent Class : QIODevice

- QVariant attribute(QNetworkRequest::Attribute code)
- QNetworkReply::NetworkError error(void)
- bool hasRawHeader(QByteArray)
- QVariant header(QNetworkRequest::KnownHeaders header)
- bool isFinished(void)
- bool isRunning(void)
- QNetworkAccessManager *manager(void)
- QNetworkAccessManager::Operation operation(void)
- QByteArray rawHeader(QByteArray)
- qint64 readBufferSize(void)
- QNetworkRequest request(void)

91.194 QNetworkRequest Class

C++ Reference : http://doc.qt.io/qt-5/qnetworkrequest.html

Parameters : QUrl

- QVariant attribute(QNetworkRequest::Attribute, QVariant)
- bool hasRawHeader(QByteArray)
- QVariant header(QNetworkRequest::KnownHeaders)
- QObject *originatingObject(void)
- QNetworkRequest::Priority priority(void)

- QByteArray rawHeader(QByteArray)
- void setAttribute(QNetworkRequest::Attribute, QVariant)
- void setHeader(QNetworkRequest::KnownHeaders, QVariant)
- void setOriginatingObject(QObject *object)
- void setPriority(QNetworkRequest::Priority priority)
- void setRawHeader(QByteArray, QByteArray)
- void swap(QNetworkRequest)
- QUrl url(void)

91.195 QNmeaPositionInfoSource Class

C++ Reference : http://doc.qt.io/qt-5/qnmeapositioninfosource.html

Parameters : QNmeaPositionInfoSource::UpdateMode,QObject *

Parent Class : QGeoPositionInfoSource

- QIODevice * device(void)
- void setDevice(QIODevice * device)
- QNmeaPositionInfoSource::UpdateMode updateMode(void)

91.196 QNode Class

C++ Reference : http://doc.qt.io/qt-5/qt3dcore-qnode.html

Parameters : Qt3DCore::QNode *

Parent Class : QObject

- bool blockNotifications(bool block)
- Qt3DCore::QNodeVector childNodes(void)
- void clearPropertyTracking(QString propertyName)
- void clearPropertyTrackings(void)
- Qt3DCore::QNode::PropertyTrackingMode defaultPropertyTrackingMode(void)
- Qt3DCore::QNodeId id(void)
- bool isEnabled(void)
- bool notificationsBlocked(void)
- Qt3DCore::QNode * parentNode(void)
- Qt3DCore::QNode::PropertyTrackingMode propertyTracking(QString propertyName)
- void setPropertyTracking(QString propertyName, Qt3DCore::QNode::PropertyTrackingMode trackMode)
- void setDefaultPropertyTrackingMode(Qt3DCore::QNode::PropertyTrackingMode mode)
- void setEnabled(bool isEnabled)
- void setParent(Qt3DCore::QNode *parent)

91.197 QObject Class

C++ Reference : http://doc.qt.io/qt-5/qobject.html

Parameters : void

- bool blockSignals(bool block)
- QObjectList children(void)
- void dumpObjectInfo(void)
- void dumpObjectTree(void)
- bool inherits(const char *className)
- void installEventFilter(QObject *filterObj)
- bool isWidgetType(void)
- void killTimer(int id)
- void moveToThread(QThread *targetThread)
- QString objectName(void)
- QObject *parent(void)
- QVariant property(const char *name)
- void removeEventFilter(QObject *obj)
- void setObjectName(QString)
- void setParent(QObject *parent)
- bool setProperty(const char *name, QVariant)
- bool setProperty_2(const char *name, int)
- bool setProperty_3(const char *name, float)
- bool setProperty_4(const char *name, double)
- bool setProperty_5(const char *name, QString)
- bool setProperty_int(const char *name, int)
- bool setProperty_float(const char *name, float)
- bool setProperty_double(const char *name, double)
- bool setProperty_string(const char *name, QString)
- bool signalsBlocked(void)
- int startTimer(int interval)
- QThread *thread(void)
- void deleteLater(void)

91.198 QObjectPicker Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qobjectpicker.html

Parameters : Qt3DCore::QNode *

- bool containsMouse(void)
- bool isDragEnabled(void)
- bool isHoverEnabled(void)
- bool isPressed(void)
- void setDragEnabled(bool dragEnabled)
- void setHoverEnabled(bool hoverEnabled)
- void setclickedEvent(const char *)
- void setcontainsMouseChangedEvent(const char *)
- void setdragEnabledChangedEvent(const char *)
- void setenteredEvent(const char *)
- void setexitedEvent(const char *)
- void sethoverEnabledChangedEvent(const char *)
- void setmovedEvent(const char *)
- void setpressedEvent(const char *)
- void setpressedChangedEvent(const char *)
- void setreleasedEvent(const char *)
- const char *getclickedEvent(void)
- const char *getcontainsMouseChangedEvent(void)
- const char *getdragEnabledChangedEvent(void)
- const char *getenteredEvent(void)
- const char *getexitedEvent(void)
- const char *gethoverEnabledChangedEvent(void)
- const char *getmovedEvent(void)
- const char *getpressedEvent(void)
- const char *getpressedChangedEvent(void)
- const char *getreleasedEvent(void)

91.199 QOpenGLBuffer Class

C++ Reference : http://doc.qt.io/qt-5/qopenglbuffer.html

Parameters : void

- void allocate(void *data, int count)
- void allocate_2(int count)
- bool bind(void)
- GLuint bufferId(void)
- bool create(void)
- void destroy(void)
- bool isCreated(void)
- void * map(QOpenGLBuffer::Access access)
- void release(void)
- void setUsagePattern(QOpenGLBuffer::UsagePattern value)
- int size(void)
- QOpenGLBuffer::Type type(void)
- bool unmap(void)
- QOpenGLBuffer::UsagePattern usagePattern(void)
- void write(int offset, void *data, int count)
- void release_2(QOpenGLBuffer::Type type)

91.200 QOpenGLContext Class

C++ Reference : http://doc.qt.io/qt-5/qopenglcontext.html

Parameters : QObject *

Parent Class : QObject

- bool create(void)
- GLuint defaultFramebufferObject(void)
- void doneCurrent(void)
- QSet<QByteArray> extensions(void)
- QOpenGLFunctions * functions(void)
- QFunctionPointer getProcAddress(QByteArray procName)
- QFunctionPointer getProcAddress_2(char *procName)
- bool hasExtension(QByteArray extension)
- bool isOpenGLES(void)
- bool isValid(void)
- bool makeCurrent(QSurface *surface)

- QVariant nativeHandle(void)
- QScreen * screen(void)
- void setFormat(QSurfaceFormat format)
- void setNativeHandle(QVariant handle)
- void setScreen(QScreen *screen)
- void setShareContext(QOpenGLContext *shareContext)
- QOpenGLContext * shareContext(void)
- QOpenGLContextGroup * shareGroup(void)
- QSurface * surface(void)
- void swapBuffers(QSurface *surface)
- QAbstractOpenGLFunctions * versionFunctions(QOpenGLVersionProfile versionProfile))
- TYPE * versionFunctions_2(void)
- bool areSharing(QOpenGLContext *first, QOpenGLContext *second)
- QOpenGLContext * currentContext(void)
- QOpenGLContext * globalShareContext(void)
- void * openGLModuleHandle(void)
- QOpenGLContext::OpenGLModuleType openGLModuleType(void)
- bool supportsThreadedOpenGL(void)
- QOpenGLFunctions_3_2_Core *opengl32(void)

91.201 QOpenGLDebugLogger Class

C++ Reference : http://doc.qt.io/qt-5/qopengldebuglogger.html

Parameters : QObject *

- void disableMessages(QOpenGLDebugMessage::Sources sources, QOpenGLDebugMessage::Types types, QOpenGLDebugMessage::Severities severities)
- void disableMessages_2(QVector<GLuint> ids, QOpenGLDebugMessage::Sources sources, QOpenGLDebugMessage::Types types)
- void enableMessages(QOpenGLDebugMessage::Sources sources, QOpenGLDebugMessage::Types types, QOpenGLDebugMessage::Severities severities)
- void enableMessages_2(QVector<GLuint> ids, QOpenGLDebugMessage::Sources sources, QOpenGLDebugMessage::Types types)
- bool initialize(void)
- bool isLogging(void)
- QList<QOpenGLDebugMessage> loggedMessages(void)
- QOpenGLDebugLogger::LoggingMode loggingMode(void)
- qint64 maximumMessageLength(void)
- void popGroup(void)

- void pushGroup(QString name, GLuint id, QOpenGLDebugMessage::Source source)
- void logMessage(QOpenGLDebugMessage debugMessage)
- void startLogging(QOpenGLDebugLogger::LoggingMode loggingMode)
- void stopLogging(void)

91.202 QOpenGLFramebufferObject Class

C++ Reference : http://doc.qt.io/qt-5/qopenglframebufferobject.html

Parameters : int,int,GLenum

- bool bind(void)
- QOpenGLFramebufferObjectFormat format(void)
- GLuint handle(void)
- int height(void)
- bool isBound(void)
- bool isValid(void)
- bool release(void)
- void setAttachment(QOpenGLFramebufferObject::Attachment attachment)
- QSize size(void)
- QImage toImage(bool flipped)
- QImage toImage_3(bool flipped, int colorQOpenGLFramebufferObject::AttachmentIndex)
- QImage toImage_2(void)
- int width(void)
- bool bindDefault(void)
- bool hasOpenGLFramebufferObjects(void)

91.203 QOpenGLFunctions Class

C++ Reference : http://doc.qt.io/qt-5/qopenglfunctions.html

Parameters : void

- void glActiveTexture(GLenum texture)
- void glAttachShader(GLuint program, GLuint shader)
- void glBindAttribLocation(GLuint program, GLuint index, char *name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindFramebuffer(GLenum target, GLuint framebuffer)
- void glBindRenderbuffer(GLenum target, GLuint renderbuffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBlendColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)

- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum srcRGB, GLenum dstRGB, GLenum srcAlpha, GLenum dstAlpha)
- void glBufferData(GLenum target, qopengl_GLsizeiptr size, void *data, GLenum usage)
- void glBufferSubData(GLenum target, qopengl_GLintptr offset, qopengl_GLsizeiptr size, void *data)
- GLenum glCheckFramebufferStatus(GLenum target)
- void glClear(GLbitfield mask)
- void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha)
- void glClearDepthf(GLclampf depth)
- void glClearStencil(GLint s)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, void *data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, void *data)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum type)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n, GLuint *buffers)
- void glDeleteFramebuffers(GLsizei n, GLuint *framebuffers)
- void glDeleteProgram(GLuint program)
- void glDeleteRenderbuffers(GLsizei n, GLuint *renderbuffers)
- void glDeleteShader(GLuint shader)
- void glDeleteTextures(GLsizei n, GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRangef(GLclampf zNear, GLclampf zFar)
- void glDetachShader(GLuint program, GLuint shader)
- void glDisable(GLenum cap)
- void glDisableVertexAttribArray(GLuint index)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)

- void glDrawElements(GLenum mode, GLsizei count, GLenum type, GLvoid *indices)
- void glEnable(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glFinish(void)
- void glFlush(void)
- void glFramebufferRenderbuffer(GLenum target, GLenum attachment, GLenum renderbuffertarget, GLuint renderbuffer)
- void glFramebufferTexture2D(GLenum target, GLenum attachment, GLenum textarget, GLuint texture, GLint level)
- void glFrontFace(GLenum mode)
- void glGenBuffers(GLsizei n, GLuint *buffers)
- void glGenFramebuffers(GLsizei n, GLuint *framebuffers)
- void glGenRenderbuffers(GLsizei n, GLuint *renderbuffers)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGenerateMipmap(GLenum target)
- void glGetActiveAttrib(GLuint program, GLuint index, GLsizei bufsize, GLsizei *length, GLint *size, GLenum *type, char *name)
- void glGetActiveUniform(GLuint program, GLuint index, GLsizei bufsize, GLsizei *length, GLint *size, GLenum *type, char *name)
- void glGetAttachedShaders(GLuint program, GLsizei maxcount, GLsizei *count, GLuint *shaders)
- GLint glGetAttribLocation(GLuint program, char *name)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetBufferParameteriv(GLenum target, GLenum pname, GLint *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetFramebufferAttachmentParameteriv(GLenum target, GLenum attachment, GLenum pname, GLint *params)
- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetProgramInfoLog(GLuint program, GLsizei bufsize, GLsizei *length, char *infolog)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetRenderbufferParameteriv(GLenum target, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei bufsize, GLsizei *length, char *infolog)
- void glGetShaderPrecisionFormat(GLenum shadertype, GLenum precisiontype, GLint *range, GLint *precision)
- void glGetShaderSource(GLuint shader, GLsizei bufsize, GLsizei *length, char *source)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- GLubyte * glGetString(GLenum name)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)

- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)
- GLint glGetUniformLocation(GLuint program, char *name)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glHint(GLenum target, GLenum mode)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsFramebuffer(GLuint framebuffer)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsRenderbuffer(GLuint renderbuffer)
- GLboolean glIsShader(GLuint shader)
- GLboolean glIsTexture(GLuint texture)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glPixelStorei(GLenum pname, GLint param)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid *pixels)
- void glReleaseShaderCompiler(void)
- void glRenderbufferStorage(GLenum target, GLenum internalformat, GLsizei width, GLsizei height)
- void glSampleCoverage(GLclampf value, GLboolean invert)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glShaderBinary(GLint n, GLuint *shaders, GLenum binaryformat, void *binary, GLint length)
- void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face, GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glStencilOpSeparate(GLenum face, GLenum fail, GLenum zfail, GLenum zpass)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, GLvoid *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid *pixels)

- void glUniform1f(GLint location, GLfloat x)
- void glUniform1fv(GLint location, GLsizei count, GLfloat *v)
- void glUniform1i(GLint location, GLint x)
- void glUniform1iv(GLint location, GLsizei count, GLint *v)
- void glUniform2f(GLint location, GLfloat x, GLfloat y)
- void glUniform2fv(GLint location, GLsizei count, GLfloat *v)
- void glUniform2i(GLint location, GLint x, GLint y)
- void glUniform2iv(GLint location, GLsizei count, GLint *v)
- void glUniform3f(GLint location, GLfloat x, GLfloat y, GLfloat z)
- void glUniform3fv(GLint location, GLsizei count, GLfloat *v)
- void glUniform3i(GLint location, GLint x, GLint y, GLint z)
- void glUniform3iv(GLint location, GLsizei count, GLint *v)
- void glUniform4f(GLint location, GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glUniform4fv(GLint location, GLsizei count, GLfloat *v)
- void glUniform4i(GLint location, GLint x, GLint y, GLint z, GLint w)
- void glUniform4iv(GLint location, GLsizei count, GLint *v)
- void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glVertexAttrib1f(GLuint indx, GLfloat x)
- void glVertexAttrib1fv(GLuint indx, GLfloat *values)
- void glVertexAttrib2f(GLuint indx, GLfloat x, GLfloat y)
- void glVertexAttrib2fv(GLuint indx, GLfloat *values)
- void glVertexAttrib3f(GLuint indx, GLfloat x, GLfloat y, GLfloat z)
- void glVertexAttrib3fv(GLuint indx, GLfloat *values)
- void glVertexAttrib4f(GLuint indx, GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertexAttrib4fv(GLuint indx, GLfloat *values)
- void glVertexAttribPointer(GLuint indx, GLint size, GLenum type, GLboolean normalized, GLsizei stride, void *ptr)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)
- bool hasOpenGLFeature(QOpenGLFunctions::OpenGLFeature feature)
- void initializeOpenGLFunctions(void)
- QOpenGLFunctions::OpenGLFeatures openGLFeatures(void)

91.204 QOpenGLFunctions_3_2_Core Class

C++ Reference : http://doc.qt.io/qt-5/qopenglfunctions_3_2_core.html

Parameters : void

- void glActiveTexture(GLenum texture)
- void glAttachShader(GLuint program, GLuint shader)
- void glBeginConditionalRender(GLuint id, GLenum mode)
- void glBeginQuery(GLenum target, GLuint id)
- void glBeginTransformFeedback(GLenum primitiveMode)
- void glBindAttribLocation(GLuint program, GLuint index, GLchar *name)
- void glBindBuffer(GLenum target, GLuint buffer)
- void glBindBufferBase(GLenum target, GLuint index, GLuint buffer)
- void glBindBufferRange(GLenum target, GLuint index, GLuint buffer, GLintptr offset, GLsizeiptr size)
- void glBindFragDataLocation(GLuint program, GLuint color, GLchar *name)
- void glBindFramebuffer(GLenum target, GLuint framebuffer)
- void glBindRenderbuffer(GLenum target, GLuint renderbuffer)
- void glBindTexture(GLenum target, GLuint texture)
- void glBindVertexArray(GLuint array)
- void glBlendColor(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glBlendEquation(GLenum mode)
- void glBlendEquationSeparate(GLenum modeRGB, GLenum modeAlpha)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glBlendFuncSeparate(GLenum sfactorRGB, GLenum dfactorRGB, GLenum sfactorAlpha, GLenum dfactorAlpha)
- void glBlitFramebuffer(GLint srcX0, GLint srcY0, GLint srcX1, GLint srcY1, GLint dstX0, GLint dstY0, GLint dstX1, GLint dstY1, GLbitfield mask, GLenum filter)
- void glBufferData(GLenum target, GLsizeiptr size, GLvoid *data, GLenum usage)
- void glBufferSubData(GLenum target, GLintptr offset, GLsizeiptr size, GLvoid *data)
- GLenum glCheckFramebufferStatus(GLenum target)
- void glClampColor(GLenum target, GLenum clamp)
- void glClear(GLbitfield mask)
- void glClearBufferfi(GLenum buffer, GLint drawbuffer, GLfloat depth, GLint stencil)
- void glClearBufferfv(GLenum buffer, GLint drawbuffer, GLfloat *value)
- void glClearBufferiv(GLenum buffer, GLint drawbuffer, GLint *value)
- void glClearBufferuiv(GLenum buffer, GLint drawbuffer, GLuint *value)
- void glClearColor(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glClearDepth(GLdouble depth)

- void glClearStencil(GLint s)
- GLenum glClientWaitSync(GLsync sync, GLbitfield flags, GLuint64 timeout)
- void glColorMask(GLboolean red, GLboolean green, GLboolean blue, GLboolean alpha)
- void glColorMaski(GLuint index, GLboolean r, GLboolean g, GLboolean b, GLboolean a)
- void glCompileShader(GLuint shader)
- void glCompressedTexImage1D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLint border, GLsizei imageSize, GLvoid *data)
- void glCompressedTexImage2D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLint border, GLsizei imageSize, GLvoid *data)
- void glCompressedTexImage3D(GLenum target, GLint level, GLenum internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLsizei imageSize, GLvoid *data)
- void glCompressedTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLsizei imageSize, GLvoid *data)
- void glCompressedTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLsizei imageSize, GLvoid *data)
- void glCompressedTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLsizei imageSize, GLvoid *data)
- void glCopyBufferSubData(GLenum readTarget, GLenum writeTarget, GLintptr readOffset, GLintptr writeOffset, GLsizeiptr size)
- void glCopyTexImage1D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLint border)
- void glCopyTexImage2D(GLenum target, GLint level, GLenum internalformat, GLint x, GLint y, GLsizei width, GLsizei height, GLint border)
- void glCopyTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLint x, GLint y, GLsizei width)
- void glCopyTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- void glCopyTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLint x, GLint y, GLsizei width, GLsizei height)
- GLuint glCreateProgram(void)
- GLuint glCreateShader(GLenum type)
- void glCullFace(GLenum mode)
- void glDeleteBuffers(GLsizei n, GLuint *buffers)
- void glDeleteFramebuffers(GLsizei n, GLuint *framebuffers)
- void glDeleteProgram(GLuint program)
- void glDeleteQueries(GLsizei n, GLuint *ids)
- void glDeleteRenderbuffers(GLsizei n, GLuint *renderbuffers)
- void glDeleteShader(GLuint shader)
- void glDeleteSync(GLsync sync)
- void glDeleteTextures(GLsizei n, GLuint *textures)
- void glDeleteVertexArrays(GLsizei n, GLuint *arrays)

- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLdouble nearVal, GLdouble farVal)
- void glDetachShader(GLuint program, GLuint shader)
- void glDisable(GLenum cap)
- void glDisableVertexAttribArray(GLuint index)
- void glDisablei(GLenum target, GLuint index)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawArraysInstanced(GLenum mode, GLint first, GLsizei count, GLsizei instancecount)
- void glDrawBuffer(GLenum mode)
- void glDrawBuffers(GLsizei n, GLenum *bufs)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, GLvoid *indices)
- void glDrawElementsBaseVertex(GLenum mode, GLsizei count, GLenum type, GLvoid *indices, GLint basevertex)
- void glDrawElementsInstanced(GLenum mode, GLsizei count, GLenum type, GLvoid *indices, GLsizei instancecount)
- void glDrawElementsInstancedBaseVertex(GLenum mode, GLsizei count, GLenum type, GLvoid *indices, GLsizei instancecount, GLint basevertex)
- void glDrawRangeElements(GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, GLvoid *indices)
- void glDrawRangeElementsBaseVertex(GLenum mode, GLuint start, GLuint end, GLsizei count, GLenum type, GLvoid *indices, GLint basevertex)
- void glEnable(GLenum cap)
- void glEnableVertexAttribArray(GLuint index)
- void glEnablei(GLenum target, GLuint index)
- void glEndConditionalRender(void)
- void glEndQuery(GLenum target)
- void glEndTransformFeedback(void)
- GLsync glFenceSync(GLenum condition, GLbitfield flags)
- void glFinish(void)
- void glFlush(void)
- void glFlushMappedBufferRange(GLenum target, GLintptr offset, GLsizeiptr length)
- void glFramebufferRenderbuffer(GLenum target, GLenum attachment, GLenum renderbuffertarget, GLuint renderbuffer)
- void glFramebufferTexture(GLenum target, GLenum attachment, GLuint texture, GLint level)
- void glFramebufferTexture1D(GLenum target, GLenum attachment, GLenum textarget, GLuint texture, GLint level)
- void glFramebufferTexture2D(GLenum target, GLenum attachment, GLenum textarget, GLuint texture, GLint level)

- void glFramebufferTexture3D(GLenum target, GLenum attachment, GLenum textarget, GLuint texture, GLint level, GLint zoffset)
- void glFramebufferTextureLayer(GLenum target, GLenum attachment, GLuint texture, GLint level, GLint layer)
- void glFrontFace(GLenum mode)
- void glGenBuffers(GLsizei n, GLuint *buffers)
- void glGenFramebuffers(GLsizei n, GLuint *framebuffers)
- void glGenQueries(GLsizei n, GLuint *ids)
- void glGenRenderbuffers(GLsizei n, GLuint *renderbuffers)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGenVertexArrays(GLsizei n, GLuint *arrays)
- void glGenerateMipmap(GLenum target)
- void glGetActiveAttrib(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetActiveUniform(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLint *size, GLenum *type, GLchar *name)
- void glGetActiveUniformBlockName(GLuint program, GLuint uniformBlockIndex, GLsizei bufSize, GLsizei *length, GLchar *uniformBlockName)
- void glGetActiveUniformBlockiv(GLuint program, GLuint uniformBlockIndex, GLenum pname, GLint *params)
- void glGetActiveUniformName(GLuint program, GLuint uniformIndex, GLsizei bufSize, GLsizei *length, GLchar *uniformName)
- void glGetActiveUniformsiv(GLuint program, GLsizei uniformCount, GLuint *uniformIndices, GLenum pname, GLint *params)
- void glGetAttachedShaders(GLuint program, GLsizei maxCount, GLsizei *count, GLuint *obj)
- GLint glGetAttribLocation(GLuint program, GLchar *name)
- void glGetBooleani_v(GLenum target, GLuint index, GLboolean *data)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetBufferParameteri64v(GLenum target, GLenum pname, GLint64 *params)
- void glGetBufferParameteriv(GLenum target, GLenum pname, GLint *params)
- void glGetCompressedTexImage(GLenum target, GLint level, GLvoid *img)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- GLint glGetFragDataLocation(GLuint program, GLchar *name)
- void glGetFramebufferAttachmentParameteriv(GLenum target, GLenum attachment, GLenum pname, GLint *params)
- void glGetInteger64i_v(GLenum target, GLuint index, GLint64 *data)
- void glGetInteger64v(GLenum pname, GLint64 *params)
- void glGetIntegeri_v(GLenum target, GLuint index, GLint *data)

- void glGetIntegerv(GLenum pname, GLint *params)
- void glGetMultisamplefv(GLenum pname, GLuint index, GLfloat *val)
- void glGetProgramiv(GLuint program, GLenum pname, GLint *params)
- void glGetQueryObjectiv(GLuint id, GLenum pname, GLint *params)
- void glGetQueryObjectuiv(GLuint id, GLenum pname, GLuint *params)
- void glGetQueryiv(GLenum target, GLenum pname, GLint *params)
- void glGetRenderbufferParameteriv(GLenum target, GLenum pname, GLint *params)
- void glGetShaderInfoLog(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *infoLog)
- void glGetShaderSource(GLuint shader, GLsizei bufSize, GLsizei *length, GLchar *source)
- void glGetShaderiv(GLuint shader, GLenum pname, GLint *params)
- GLubyte * glGetString(GLenum name)
- GLubyte * glGetStringi(GLenum name, GLuint index)
- void glGetSynciv(GLsync sync, GLenum pname, GLsizei bufSize, GLsizei *length, GLint *values)
- void glGetTexImage(GLenum target, GLint level, GLenum format, GLenum type, GLvoid *pixels)
- void glGetTexLevelParameterfv(GLenum target, GLint level, GLenum pname, GLfloat *params)
- void glGetTexLevelParameteriv(GLenum target, GLint level, GLenum pname, GLint *params)
- void glGetTexParameterIiv(GLenum target, GLenum pname, GLint *params)
- void glGetTexParameterIuiv(GLenum target, GLenum pname, GLuint *params)
- void glGetTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glGetTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glGetTransformFeedbackVarying(GLuint program, GLuint index, GLsizei bufSize, GLsizei *length, GLsizei *size, GLenum *type, GLchar *name)
- GLuint glGetUniformBlockIndex(GLuint program, GLchar *uniformBlockName)
- void glGetUniformfv(GLuint program, GLint location, GLfloat *params)
- void glGetUniformiv(GLuint program, GLint location, GLint *params)
- void glGetUniformuiv(GLuint program, GLint location, GLuint *params)
- void glGetVertexAttribIiv(GLuint index, GLenum pname, GLint *params)
- void glGetVertexAttribIuiv(GLuint index, GLenum pname, GLuint *params)
- void glGetVertexAttribfv(GLuint index, GLenum pname, GLfloat *params)
- void glGetVertexAttribiv(GLuint index, GLenum pname, GLint *params)
- void glHint(GLenum target, GLenum mode)
- void glIndexub(GLubyte c)
- void glIndexubv(GLubyte *c)
- GLboolean glIsBuffer(GLuint buffer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsEnabledi(GLenum target, GLuint index)

- GLboolean glIsFramebuffer(GLuint framebuffer)
- GLboolean glIsProgram(GLuint program)
- GLboolean glIsQuery(GLuint id)
- GLboolean glIsRenderbuffer(GLuint renderbuffer)
- GLboolean glIsShader(GLuint shader)
- GLboolean glIsSync(GLsync sync)
- GLboolean glIsTexture(GLuint texture)
- GLboolean glIsVertexArray(GLuint array)
- void glLineWidth(GLfloat width)
- void glLinkProgram(GLuint program)
- void glLogicOp(GLenum opcode)
- GLvoid * glMapBuffer(GLenum target, GLenum access)
- GLvoid * glMapBufferRange(GLenum target, GLintptr offset, GLsizeiptr length, GLbitfield access)
- void glMultiDrawArrays(GLenum mode, GLint *first, GLsizei *count, GLsizei drawcount)
- void glPixelStorei(GLenum pname, GLint param)
- void glPointParameterf(GLenum pname, GLfloat param)
- void glPointParameterfv(GLenum pname, GLfloat *params)
- void glPointParameteri(GLenum pname, GLint param)
- void glPointParameteriv(GLenum pname, GLint *params)
- void glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPrimitiveRestartIndex(GLuint index)
- void glProvokingVertex(GLenum mode)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid *pixels)
- void glRenderbufferStorage(GLenum target, GLenum internalformat, GLsizei width, GLsizei height)
- void glRenderbufferStorageMultisample(GLenum target, GLsizei samples, GLenum internalformat, GLsizei width, GLsizei height)
- void glSampleCoverage(GLfloat value, GLboolean invert)
- void glSampleMaski(GLuint index, GLbitfield mask)
- void glScissor(GLint x, GLint y, GLsizei width, GLsizei height)
- void glStencilFuncSeparate(GLenum face, GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilMaskSeparate(GLenum face, GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)

- void glStencilOpSeparate(GLenum face, GLenum sfail, GLenum dpfail, GLenum dppass)
- void glTexBuffer(GLenum target, GLenum internalformat, GLuint buffer)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, GLvoid *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, GLvoid *pixels)
- void glTexImage2DMultisample(GLenum target, GLsizei samples, GLint internalformat, GLsizei width, GLsizei height, GLboolean fixedsamplelocations)
- void glTexImage3D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLsizei depth, GLint border, GLenum format, GLenum type, GLvoid *pixels)
- void glTexImage3DMultisample(GLenum target, GLsizei samples, GLint internalformat, GLsizei width, GLsizei height, GLsizei depth, GLboolean fixedsamplelocations)
- void glTexParameterIiv(GLenum target, GLenum pname, GLint *params)
- void glTexParameterIuiv(GLenum target, GLenum pname, GLuint *params)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, GLvoid *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, GLvoid *pixels)
- void glTexSubImage3D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLint zoffset, GLsizei width, GLsizei height, GLsizei depth, GLenum format, GLenum type, GLvoid *pixels)
- void glUniform1fv(GLint location, GLsizei count, GLfloat *value)
- void glUniform1i(GLint location, GLint v0)
- void glUniform1iv(GLint location, GLsizei count, GLint *value)
- void glUniform1ui(GLint location, GLuint v0)
- void glUniform1uiv(GLint location, GLsizei count, GLuint *value)
- void glUniform2f(GLint location, GLfloat v0, GLfloat v1)
- void glUniform2fv(GLint location, GLsizei count, GLfloat *value)
- void glUniform2i(GLint location, GLint v0, GLint v1)
- void glUniform2iv(GLint location, GLsizei count, GLint *value)
- void glUniform2ui(GLint location, GLuint v0, GLuint v1)
- void glUniform2uiv(GLint location, GLsizei count, GLuint *value)
- void glUniform3f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2)
- void glUniform3fv(GLint location, GLsizei count, GLfloat *value)
- void glUniform3i(GLint location, GLint v0, GLint v1, GLint v2)
- void glUniform3iv(GLint location, GLsizei count, GLint *value)

- void glUniform3ui(GLint location, GLuint v0, GLuint v1, GLuint v2)
- void glUniform3uiv(GLint location, GLsizei count, GLuint *value)
- void glUniform4f(GLint location, GLfloat v0, GLfloat v1, GLfloat v2, GLfloat v3)
- void glUniform4fv(GLint location, GLsizei count, GLfloat *value)
- void glUniform4i(GLint location, GLint v0, GLint v1, GLint v2, GLint v3)
- void glUniform4iv(GLint location, GLsizei count, GLint *value)
- void glUniform4ui(GLint location, GLuint v0, GLuint v1, GLuint v2, GLuint v3)
- void glUniform4uiv(GLint location, GLsizei count, GLuint *value)
- void glUniformBlockBinding(GLuint program, GLuint uniformBlockIndex, GLuint uniformBlockBinding)
- void glUniformMatrix2fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix2x3fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix2x4fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix3fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix3x2fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix3x4fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix4fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix4x2fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- void glUniformMatrix4x3fv(GLint location, GLsizei count, GLboolean transpose, GLfloat *value)
- GLboolean glUnmapBuffer(GLenum target)
- void glUseProgram(GLuint program)
- void glValidateProgram(GLuint program)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)
- void glWaitSync(GLsync sync, GLbitfield flags, GLuint64 timeout)

91.205 QOpenGLPaintDevice Class

C++ Reference : http://doc.qt.io/qt-5/qopenglpaintdevice.html

Parameters : void

Parent Class : QPaintDevice

- QOpenGLContext * context(void)
- qreal dotsPerMeterX(void)
- qreal dotsPerMeterY(void)
- void ensureActiveTarget(void)
- bool paintFlipped(void)
- void setDevicePixelRatio(qreal devicePixelRatio)
- void setDotsPerMeterX(qreal dpmx)

- void setDotsPerMeterY(qreal dpmy)
- void setPaintFlipped(bool flipped)
- void setSize(QSize size)
- QSize size(void)

91.206 QOpenGLShader Class

C++ Reference : http://doc.qt.io/qt-5/qopenglshader.html

Parameters : QOpenGLShader::ShaderType,QObject *

- bool compileSourceCode(char *source)
- bool compileSourceCode_2(QByteArray source)
- bool compileSourceCode_3(QString source)
- bool compileSourceFile(QString fileName)
- bool isCompiled(void)
- QString log(void)
- GLuint shaderId(void)
- QOpenGLShader::ShaderType shaderType(void)
- QByteArray sourceCode(void)
- bool hasOpenGLShaders(QOpenGLShader::ShaderType type, QOpenGLContext *context)

91.207 QOpenGLShaderProgram Class

C++ Reference : http://doc.qt.io/qt-5/qopenglshaderprogram.html

Parameters : QObject *

- bool addShaderFromSourceCode(QOpenGLShader::ShaderType type, char *source)
- bool addShaderFromSourceCode_2(QOpenGLShader::ShaderType type, QByteArray source)
- bool addShaderFromSourceCode_3(QOpenGLShader::ShaderType type, QString source)
- bool addShaderFromSourceFile(QOpenGLShader::ShaderType type, QString fileName)
- int attributeLocation(char *name)
- int attributeLocation_2(QByteArray name)
- int attributeLocation_3(QString name)
- bool bind(void)
- void bindAttributeLocation(char *name, int location)
- void bindAttributeLocation_2(QByteArray name, int location)
- void bindAttributeLocation_3(QString name, int location)
- QVector<float> defaultOuterTessellationLevels(void)
- void disableAttributeArray(int location)

- void disableAttributeArray_2(char *name)
- void enableAttributeArray(int location)
- void enableAttributeArray_2(char *name)
- bool isLinked(void)
- bool link(void)
- QString log(void)
- int maxGeometryOutputVertices(void)
- int patchVertexCount(void)
- GLuint programId(void)
- void release(void)
- void removeAllShaders(void)
- void removeShader(QOpenGLShader *shader)
- void setAttributeArray(int location, GLfloat *values, int tupleSize, int stride)
- void setAttributeArray_2(int location, QVector2D *values, int stride)
- void setAttributeArray_3(int location, QVector3D *values, int stride)
- void setAttributeArray_4(int location, QVector4D *values, int stride)
- void setAttributeArray_5(int location, GLenum type, void *values, int tupleSize, int stride)
- void setAttributeArray_6(char *name, GLfloat *values, int tupleSize, int stride)
- void setAttributeArray_7(char *name, QVector2D *values, int stride)
- void setAttributeArray_8(char *name, QVector3D *values, int stride)
- void setAttributeArray_9(char *name, QVector4D *values, int stride)
- void setAttributeArray_10(char *name, GLenum type, void *values, int tupleSize, int stride)
- void setAttributeBuffer(int location, GLenum type, int offset, int tupleSize, int stride)
- void setAttributeBuffer_2(char *name, GLenum type, int offset, int tupleSize, int stride)
- void setAttributeValue(int location, GLfloat value)
- void setAttributeValue_2(int location, GLfloat x, GLfloat y)
- void setAttributeValue_3(int location, GLfloat x, GLfloat y, GLfloat z)
- void setAttributeValue_4(int location, GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void setAttributeValue_5(int location, QVector2D value)
- void setAttributeValue_6(int location, QVector3D value)
- void setAttributeValue_7(int location, QVector4D value)
- void setAttributeValue_8(int location, QColor value)
- void setAttributeValue_9(int location, GLfloat *values, int columns, int rows)
- void setAttributeValue_10(char *name, GLfloat value)
- void setAttributeValue_11(char *name, GLfloat x, GLfloat y)
- void setAttributeValue_12(char *name, GLfloat x, GLfloat y, GLfloat z)

- void setAttributeValue_13(char *name, GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void setAttributeValue_14(char *name, QVector2D value)
- void setAttributeValue_15(char *name, QVector3D value)
- void setAttributeValue_16(char *name, QVector4D value)
- void setDefaultInnerTessellationLevels(QVector<float> levels)
- void setDefaultOuterTessellationLevels(QVector<float> levels)
- void setPatchVertexCount(int count)
- void setUniformValue(int location, GLfloat value)
- void setUniformValue_2(int location, GLint value)
- void setUniformValue_3(char *name, QColor color)
- void setUniformValue_4(char *name, QPoint point)
- void setUniformValue_5(char *name, QPointF point)
- void setUniformValue_6(char *name, QSize size)
- void setUniformValue_7(char *name, QSizeF size)
- void setUniformValue_8(char *name, QMatrix2x2 value)
- void setUniformValue_9(char *name, QMatrix2x3 value)
- void setUniformValue_10(char *name, QMatrix2x4 value)
- void setUniformValue_11(char *name, QMatrix3x2 value)
- void setUniformValue_12(char *name, QMatrix3x3 value)
- void setUniformValue_13(char *name, QMatrix3x4 value)
- void setUniformValue_14(char *name, QMatrix4x2 value)
- void setUniformValue_15(char *name, QMatrix4x3 value)
- void setUniformValue_16(char *name, QMatrix4x4 value)
- void setUniformValue_21(int location, GLuint value)
- void setUniformValue_22(int location, GLfloat x, GLfloat y)
- void setUniformValue_23(int location, GLfloat x, GLfloat y, GLfloat z)
- void setUniformValue_24(int location, GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void setUniformValue_25(int location, QVector2D value)
- void setUniformValue_26(int location, QVector3D value)
- void setUniformValue_27(int location, QVector4D value)
- void setUniformValue_28(int location, QColor color)
- void setUniformValue_29(int location, QPoint point)
- void setUniformValue_30(int location, QPointF point)
- void setUniformValue_31(int location, QSize size)
- void setUniformValue_32(int location, QSizeF size)
- void setUniformValue_33(int location, QMatrix2x2 value)

- void setUniformValue_34(int location, QMatrix2x3 value)
- void setUniformValue_35(int location, QMatrix2x4 value)
- void setUniformValue_36(int location, QMatrix3x2 value)
- void setUniformValue_37(int location, QMatrix3x3 value)
- void setUniformValue_38(int location, QMatrix3x4 value)
- void setUniformValue_39(int location, QMatrix4x2 value)
- void setUniformValue_40(int location, QMatrix4x3 value)
- void setUniformValue_41(int location, QMatrix4x4 value)
- void setUniformValue_46(char *name, GLfloat value)
- void setUniformValue_47(char *name, GLint value)
- void setUniformValue_48(char *name, GLuint value)
- void setUniformValue_49(char *name, GLfloat x, GLfloat y)
- void setUniformValue_50(char *name, GLfloat x, GLfloat y, GLfloat z)
- void setUniformValue_51(char *name, GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void setUniformValue_52(char *name, QVector2D value)
- void setUniformValue_53(char *name, QVector3D value)
- void setUniformValue_54(char *name, QVector4D value)
- void setUniformValueArray(int location, GLfloat *values, int count, int tupleSize)
- void setUniformValueArray_2(int location, GLint *values, int count)
- void setUniformValueArray_3(int location, GLuint *values, int count)
- void setUniformValueArray_4(int location, QVector2D *values, int count)
- void setUniformValueArray_5(int location, QVector3D *values, int count)
- void setUniformValueArray_6(int location, QVector4D *values, int count)
- void setUniformValueArray_7(int location, QMatrix2x2 *values, int count)
- void setUniformValueArray_8(int location, QMatrix2x3 *values, int count)
- void setUniformValueArray_9(int location, QMatrix2x4 *values, int count)
- void setUniformValueArray_10(int location, QMatrix3x2 *values, int count)
- void setUniformValueArray_11(int location, QMatrix3x3 *values, int count)
- void setUniformValueArray_12(int location, QMatrix3x4 *values, int count)
- void setUniformValueArray_13(int location, QMatrix4x2 *values, int count)
- void setUniformValueArray_14(int location, QMatrix4x3 *values, int count)
- void setUniformValueArray_15(int location, QMatrix4x4 *values, int count)
- void setUniformValueArray_16(char *name, GLfloat *values, int count, int tupleSize)
- void setUniformValueArray_17(char *name, GLint *values, int count)
- void setUniformValueArray_18(char *name, GLuint *values, int count)
- void setUniformValueArray_19(char *name, QVector2D *values, int count)

- void setUniformValueArray_20(char *name, QVector3D *values, int count)
- void setUniformValueArray_21(char *name, QVector4D *values, int count)
- void setUniformValueArray_22(char *name, QMatrix2x2 *values, int count)
- void setUniformValueArray_23(char *name, QMatrix2x3 *values, int count)
- void setUniformValueArray_24(char *name, QMatrix2x4 *values, int count)
- void setUniformValueArray_25(char *name, QMatrix3x2 *values, int count)
- void setUniformValueArray_26(char *name, QMatrix3x3 *values, int count)
- void setUniformValueArray_27(char *name, QMatrix3x4 *values, int count)
- void setUniformValueArray_28(char *name, QMatrix4x2 *values, int count)
- void setUniformValueArray_29(char *name, QMatrix4x3 *values, int count)
- void setUniformValueArray_30(char *name, QMatrix4x4 *values, int count)
- QList<QOpenGLShader *> shaders(void)
- int uniformLocation(char *name)
- int uniformLocation_2(QByteArray name)
- int uniformLocation_3(QString name)
- bool hasOpenGLShaderPrograms(QOpenGLContext *context)

91.208 QOpenGLTexture Class

C++ Reference : http://doc.qt.io/qt-5/qopengltexture.html

Parameters : QOpenGLTexture::Target

- void allocateStorage(void)
- void bind_2(uint unit, QOpenGLTexture::TextureUnitReset reset)
- QColor borderColor(void)
- void borderColor_2(float *border)
- void borderColor_3(int *border)
- void borderColor_4(unsigned int *border)
- QOpenGLTexture * createTextureView(QOpenGLTexture::Target target, QOpenGLTexture::TextureFormat viewFormat, int minimumMipmapLevel, int maximumMipmapLevel, int minimumLayer, int maximumLayer)
- int depth(void)
- QOpenGLTexture::DepthStencilMode depthStencilMode(void)
- void destroy(void)
- int faces(void)
- QOpenGLTexture::TextureFormat format(void)
- void generateMipMaps(void)
- void generateMipMaps_2(int baseLevel, bool resetBaseLevel)
- int height(void)

- bool isAutoMipMapGenerationEnabled(void)
- bool isBound(void)
- bool isBound_2(uint unit)
- bool isCreated(void)
- bool isTextureView(void)
- int layers(void)
- QPair<float, float> levelOfDetailRange(void)
- float levelofDetailBias(void)
- QOpenGLTexture::Filter magnificationFilter(void)
- float maximumAnisotropy(void)
- float maximumLevelOfDetail(void)
- int maximumMipLevels(void)
- float minimumLevelOfDetail(void)
- int mipBaseLevel(void)
- int mipMaxLevel(void)
- void release(void)
- void release_2(uint unit, QOpenGLTexture::TextureUnitReset reset)
- void setBorderColor(QColor color)
- void setBorderColor_4(uint r, uint g, uint b, uint a)
- void setCompressedData_4(int mipLevel, int dataSize, void *data, QOpenGLPixelTransferOptions * options)
- void setCompressedData_5(int dataSize, void *data, QOpenGLPixelTransferOptions * options)
- void setData(int mipLevel, int layer, QOpenGLTexture::CubeMapFace cubeFace, QOpenGLTexture::PixelFormat sourceFormat, QOpenGLTexture::PixelType sourceType, void *data, QOpenGLPixelTransferOptions * options)
- void setData_4(int mipLevel, QOpenGLTexture::PixelFormat sourceFormat, QOpenGLTexture::PixelType sourceType, void *data, QOpenGLPixelTransferOptions * options)
- void setData_5(QOpenGLTexture::PixelFormat sourceFormat, QOpenGLTexture::PixelType sourceType, void *data, QOpenGLPixelTransferOptions * options)
- void setData_6(QImage image, QOpenGLTexture::MipMapGeneration genMipMaps)
- void setDepthStencilMode(QOpenGLTexture::DepthStencilMode mode)
- void setLayers(int layers)
- void setLevelOfDetailRange(float min, float max)
- void setLevelofDetailBias(float bias)
- void setMagnificationFilter(QOpenGLTexture::Filter filter)
- void setMaximumAnisotropy(float anisotropy)
- void setMaximumLevelOfDetail(float value)
- void setMinMagFilters(QOpenGLTexture::Filter minificationQOpenGLTexture::Filter, QOpenGLTexture::Filter magnificationQOpenGLTexture::Filter)

- void setMinificationFilter(QOpenGLTexture::Filter filter)
- void setMinimumLevelOfDetail(float value)
- void setMipBaseLevel(int baseLevel)
- void setMipLevelRange(int baseLevel, int maxLevel)
- void setMipLevels(int levels)
- void setMipMaxLevel(int maxLevel)
- void setSwizzleMask(QOpenGLTexture::SwizzleComponent component, QOpenGLTexture::SwizzleValue value)
- void setSwizzleMask_2(QOpenGLTexture::SwizzleValue r, QOpenGLTexture::SwizzleValue g, QOpenGLTexture::SwizzleValue b, QOpenGLTexture::SwizzleValue a)
- void setWrapMode(QOpenGLTexture::WrapMode mode)
- void setWrapMode_2(QOpenGLTexture::CoordinateDirection direction, QOpenGLTexture::WrapMode mode)
- QOpenGLTexture::SwizzleValue swizzleMask(QOpenGLTexture::SwizzleComponent component)
- int width(void)
- QOpenGLTexture::WrapMode wrapMode(QOpenGLTexture::CoordinateDirection direction)
- GLuint boundTextureId(QOpenGLTexture::BindingTarget target)

91.209 QOpenGLTimerQuery Class

C++ Reference : http://doc.qt.io/qt-5/qopengltimerquery.html

Parameters : QObject *

- void begin(void)
- bool create(void)
- void destroy(void)
- void end(void)
- bool isCreated(void)
- bool isResultAvailable(void)
- GLuint objectId(void)
- void recordTimestamp(void)
- GLuint64 waitForResult(void)
- GLuint64 waitForTimestamp(void)

91.210 QOpenGLVersionProfile Class

C++ Reference : http://doc.qt.io/qt-5/qopenglversionprofile.html

Parameters : void

- bool hasProfiles(void)
- bool isLegacyVersion(void)
- bool isValid(void)
- QSurfaceFormat::OpenGLContextProfile profile(void)
- void setProfile(QSurfaceFormat::OpenGLContextProfile profile)
- void setVersion(int majorVersion, int minorVersion)
- QPair<int, int> version(void)

91.211 QOpenGLVertexArrayObject Class

C++ Reference : http://doc.qt.io/qt-5/qopenglvertexarrayobject.html

Parameters : QObject *

- void bind(void)
- bool create(void)
- void destroy(void)
- bool isCreated(void)
- GLuint objectId(void)
- void release(void)

91.212 QOpenGLWidget Class

C++ Reference : http://doc.qt.io/qt-5/qopenglwidget.html

Parameters : QWidget *

Parent Class : QWidget

- void geteventparameters(void)
- void setInitEvent(const char *cStr)
- const char *getInitEvent(void)
- void setPaintEvent(const char *cStr)
- const char *getPaintEvent(void)
- void setResizeEvent(const char *cStr)
- const char *getResizeEvent(void)
- QOpenGLContext * context(void)
- GLuint defaultFramebufferObject(void)

- void doneCurrent(void)
- QSurfaceFormat format(void)
- QImage grabFramebuffer(void)
- bool isValid(void)
- void makeCurrent(void)
- void setFormat(QSurfaceFormat format)
- QOpenGLWidget::UpdateBehavior updateBehavior(void)

91.213 QOrbitCameraController Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qorbitcameracontroller.html

Parameters : Qt3DCore::QNode *

Parent Class : QAbstractCameraController

- void setZoomInLimit(float zoomInLimit)
- float zoomInLimit(void)

91.214 QPageSetupDialog Class

C++ Reference : http://doc.qt.io/qt-5/qpagesetupdialog.html

Parameters : QPrinter *,QWidget *

Parent Class : QDialog

- void open(QObject * receiver, char * member)
- QPrinter * printer(void)

91.215 QPaintDevice Class

C++ Reference : http://doc.qt.io/qt-5/qpaintdevice.html

- int colorCount(void)
- int depth(void)
- int devicePixelRatio(void)
- int heightMM(void)
- int logicalDpiX(void)
- int logicalDpiY(void)
- QPaintEngine * paintEngine(void)
- bool paintingActive(void)
- int physicalDpiX(void)
- int physicalDpiY(void)

- int width(void)
- int widthMM(void)

91.216 QPainter Class

C++ Reference : http://doc.qt.io/qt-5/qpainter.html

Parameters : void

- QBrush background(void)
- int backgroundMode(void)
- bool begin(QPaintDevice *device)
- void beginNativePainting(void)
- QRect boundingRect(int x, int y, int w, int h, int flags, QString text)
- QBrush brush(void)
- QPoint brushOrigin(void)
- QRectF clipBoundingRect(void)
- QPainterPath clipPath(void)
- QRegion clipRegion(void)
- QTransform combinedTransform(void)
- int compositionMode(void)
- QPaintDevice *device(void)
- QTransform deviceTransform(void)
- void drawArc(int x, int y, int width, int height, int startAngle, int spanAngle)
- void drawChord(int x, int y, int width, int height, int startAngle, int spanAngle)
- void drawEllipse(int x, int y, int width, int height)
- void drawGlyphRun(QPointF position, QGlyphRun glyphs)
- void drawImage(int x, int y, QImage image)
- void drawLine(int x1, int y1, int x2, int y2)
- void drawLines(QLine *lines, int lineCount)
- void drawPath(QPainterPath path)
- void drawPicture(int x, int y, QPicture picture)
- void drawPie(int x, int y, int width, int height, int startAngle, int spanAngle)
- void drawPixmap(int x, int y, QPixmap)
- void drawPoint(int x, int y)
- void drawRect(int x, int y, int width, int height)
- void drawRects(QRectF *rectangles, int rectCount)
- void drawRoundedRect(int x, int y, int w, int h, qreal xRadius, qreal yRadius, Qt::SizeMode mode)

- void drawStaticText(int left, int top, QStaticText staticText)
- void drawText(int x, int y, QString text)
- void drawTiledPixmap(int x, int y, int width, int height, QPixmap pixmap, int sx, int sy)
- bool end(void) # In RingQt use : bool endpaint(void)
- void endNativePainting(void)
- void eraseRect(int x, int y, int width, int height)
- void fillPath(QPainterPath path, QBrush brush)
- void fillRect(int x, int y, int width, int height, QBrush)
- QFont font(void)
- QFontInfo fontInfo(void)
- bool hasClipping(void)
- bool isActive(void)
- int layoutDirection(void)
- double opacity(void)
- QPaintEngine *paintEngine(void)
- QPen pen(void)
- int renderHints(void)
- void resetTransform(void)
- void restore(void)
- void rotate(qreal angle)
- void save(void)
- void scale(double sx, double sy)
- void setBackground(QBrush brush)
- void setBackgroundMode(Qt::BGMode mode)
- void setBrush(QBrush brush)
- void setBrushOrigin(int x, int y)
- void setClipPath(QPainterPath path, Qt::ClipOperation operation)
- void setClipRect(int x, int y, int width, int height, Qt::ClipOperation operation)
- void setClipRegion(QRegion region, Qt::ClipOperation operation)
- void setClipping(bool enable)
- void setCompositionMode(QPainter::CompositionMode mode)
- void setFont(QFont font)
- void setLayoutDirection(Qt::LayoutDirection direction)
- void setOpacity(qreal opacity)
- void setPen(QPen pen)
- void setRenderHint(QPainter::RenderHint hint, bool on)

- void setTransform(QTransform transform, bool combine)
- void setViewTransformEnabled(bool enable)
- void setViewport(int x, int y, int width, int height)
- void setWindow(int x, int y, int width, int height)
- void setWorldMatrixEnabled(bool enable)
- void setWorldTransform(QTransform matrix, bool combine)
- void shear(double sh, double sv)
- void strokePath(QPainterPath path, QPen pen)
- bool testRenderHint(QPainter::RenderHint hint)
- QTransform transform(void)
- void translate(double dx, double dy)
- bool viewTransformEnabled(void)
- QRect viewport(void)
- QRect window(void)
- bool worldMatrixEnabled(void)
- QTransform worldTransform(void)
- void drawPolygon(List *pPoints, Qt::FillRule fillRule)
- void drawConvexPolygon(List *pPoints)
- void drawPoints(List *pPoints)
- void drawPolyline(List *pPoints)
- void drawHSVFList(List *pPoints)
- void drawRGBFList(List *pPoints)
- void drawHSVFListAtXY(List *pPoints, int x, int y)
- void drawRGBFListAtXY(List *pPoints, int x, int y)

91.217 QPainter2 Class

Parameters : QPaintDevice *

Parent Class : QPainter

91.218 QPainterPath Class

C++ Reference : http://doc.qt.io/qt-5/qpainterpath.html

Parameters : void

- void addEllipse(qreal x, qreal y, qreal width, qreal height)
- void addPath(QPainterPath)
- void addPolygon(QPolygonF)
- void addRect(qreal x, qreal y, qreal width, qreal height)
- void addRegion(QRegion)
- void addRoundedRect(qreal x, qreal y, qreal w, qreal h, qreal xRadius, qreal yRadius, Qt::SizeMode mode)
- void addText(qreal x, qreal y, QFont, QString)
- qreal angleAtPercent(qreal t)
- void arcMoveTo(qreal x, qreal y, qreal width, qreal height, qreal angle)
- void arcTo(qreal x, qreal y, qreal width, qreal height, qreal startAngle, qreal sweepLength)
- QRectF boundingRect(void)
- void closeSubpath(void)
- void connectPath(QPainterPath)
- bool contains(QPointF)
- QRectF controlPointRect(void)
- void cubicTo(qreal c1X, qreal c1Y, qreal c2X, qreal c2Y, qreal endPointX, qreal endPointY)
- QPointF currentPosition(void)
- QPainterPath::Element elementAt(int index)
- int elementCount(void)
- Qt::FillRule fillRule(void)
- QPainterPath intersected(QPainterPath)
- bool intersects(QRectF)
- bool isEmpty(void)
- qreal length(void)
- void lineTo(qreal x, qreal y)
- void moveTo(qreal x, qreal y)
- qreal percentAtLength(qreal len)
- QPointF pointAtPercent(qreal t)
- void quadTo(qreal cx, qreal cy, qreal endPointX, qreal endPointY)
- void setElementPositionAt(int index, qreal x, qreal y)
- void setFillRule(Qt::FillRule fillRule)
- QPainterPath simplified(void)

- qreal slopeAtPercent(qreal t)
- QPainterPath subtracted(QPainterPath)
- void swap(QPainterPath)
- QPolygonF toFillPolygon(QTransform)
- QPainterPath toReversed(void)
- void translate(qreal dx, qreal dy)
- QPainterPath translated(qreal dx, qreal dy)
- QPainterPath united(QPainterPath)

91.219 QPen Class

C++ Reference : http://doc.qt.io/qt-5/qpen.html

Parameters : void

- QBrush brush(void)
- int capStyle(void)
- QColor color(void)
- double dashOffset(void)
- bool isCosmetic(void)
- bool isSolid(void)
- int joinStyle(void)
- double miterLimit(void)
- void setBrush(QBrush)
- void setCapStyle(Qt::PenCapStyle style)
- void setColor(QColor)
- void setCosmetic(bool cosmetic)
- void setDashOffset(double offset)
- void setJoinStyle(Qt::PenJoinStyle style)
- void setMiterLimit(double limit)
- void setStyle(Qt::PenStyle style)
- void setWidth(int width)
- void setWidthF(double width)
- int style(void)
- void swap(QPen)
- int width(void)
- double widthF(void)

91.220 QPerVertexColorMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qpervertexcolormaterial.html

Parameters : Qt3DCore::QNode *

91.221 QPercentBarSeries Class

C++ Reference : http://doc.qt.io/qt-5/qpercentbarseries.html

Parameters : QObject *

Parent Class : QAbstractBarSeries

- QAbstractSeries::SeriesType type(void)

91.222 QPhongMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qphongmaterial.html

Parameters : Qt3DCore::QNode *

- QColor ambient(void)
- QColor diffuse(void)
- float shininess(void)
- QColor specular(void)
- void setAmbient(QColor ambient)
- void setDiffuse(QColor diffuse)
- void setShininess(float shininess)
- void setSpecular(QColor specular)

91.223 QPicture Class

C++ Reference : http://doc.qt.io/qt-5/qpicture.html

Parameters : void

- QRect boundingRect(void)
- const char *data(void)
- bool isNull(void)
- bool load(QString, const char *format) # In RingQt use : bool loadfile(QString, const char *format)
- bool play(QPainter *painter)
- bool save(QString , const char *format)
- void setBoundingRect(QRect)
- int size(void)

- void swap(QPicture)

91.224 QPieLegendMarker Class

C++ Reference : http://doc.qt.io/qt-5/qpielegendmarker.html

Parent Class : QLegendMarker

- QPieSlice * slice(void)

91.225 QPieSeries Class

C++ Reference : http://doc.qt.io/qt-5/qpieseries.html

Parameters : QObject *

Parent Class : QAbstractSeries

- bool append_2(QList<QPieSlice *> slices)
- QPieSlice * append_3(QString label, qreal value)
- void clear(void)
- int count(void)
- qreal holeSize(void)
- qreal horizontalPosition(void)
- bool insert(int index, QPieSlice *slice)
- bool isEmpty(void)
- qreal pieEndAngle(void)
- qreal pieSize(void)
- qreal pieStartAngle(void)
- bool remove(QPieSlice *slice)
- void setHoleSize(qreal holeSize)
- void setHorizontalPosition(qreal relativePosition)
- void setLabelsPosition(QPieSlice::LabelPosition position)
- void setLabelsVisible(bool visible)
- void setPieEndAngle(qreal angle)
- void setPieSize(qreal relativeSize)
- void setPieStartAngle(qreal startAngle)
- void setVerticalPosition(qreal relativePosition)
- QList<QPieSlice *> slices(void)
- qreal sum(void)
- bool take(QPieSlice *slice)
- qreal verticalPosition(void)

- void setaddedEvent(const char *)
- void setclickedEvent(const char *)
- void setcountChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void setremovedEvent(const char *)
- void setsumChangedEvent(const char *)
- const char *getaddedEvent(void)
- const char *getclickedEvent(void)
- const char *getcountChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *getremovedEvent(void)
- const char *getsumChangedEvent(void)

91.226 QPieSlice Class

C++ Reference : http://doc.qt.io/qt-5/qpieslice.html

Parameters : QObject *

Parent Class : QObject

- QColor borderColor(void)
- int borderWidth(void)
- QBrush brush(void)
- QColor color(void)
- qreal explodeDistanceFactor(void)
- bool isExploded(void)
- bool isLabelVisible(void)
- QString label(void)
- qreal labelArmLengthFactor(void)
- QBrush labelBrush(void)
- QColor labelColor(void)
- QFont labelFont(void)

- QPieSlice::LabelPosition labelPosition(void)
- QPen pen(void)
- qreal percentage(void)
- QPieSeries * series(void)
- void setBorderColor(QColor color)
- void setBorderWidth(int width)
- void setBrush(QBrush brush)
- void setColor(QColor color)
- void setExplodeDistanceFactor(qreal factor)
- void setExploded(bool exploded)
- void setLabel(QString label)
- void setLabelArmLengthFactor(qreal factor)
- void setLabelBrush(QBrush brush)
- void setLabelColor(QColor color)
- void setLabelFont(QFont font)
- void setLabelPosition(QPieSlice::LabelPosition position)
- void setLabelVisible(bool visible)
- void setPen(QPen pen)
- void setValue(qreal value)
- qreal startAngle(void)
- qreal value(void)
- void setangleSpanChangedEvent(const char *)
- void setborderColorChangedEvent(const char *)
- void setborderWidthChangedEvent(const char *)
- void setbrushChangedEvent(const char *)
- void setclickedEvent(const char *)
- void setcolorChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setlabelBrushChangedEvent(const char *)
- void setlabelChangedEvent(const char *)
- void setlabelColorChangedEvent(const char *)
- void setlabelFontChangedEvent(const char *)
- void setlabelVisibleChangedEvent(const char *)
- void setpenChangedEvent(const char *)
- void setpercentageChangedEvent(const char *)

- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void setstartAngleChangedEvent(const char *)
- void setvalueChangedEvent(const char *)
- const char *getangleSpanChangedEvent(void)
- const char *getborderColorChangedEvent(void)
- const char *getborderWidthChangedEvent(void)
- const char *getbrushChangedEvent(void)
- const char *getclickedEvent(void)
- const char *getcolorChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getlabelBrushChangedEvent(void)
- const char *getlabelChangedEvent(void)
- const char *getlabelColorChangedEvent(void)
- const char *getlabelFontChangedEvent(void)
- const char *getlabelVisibleChangedEvent(void)
- const char *getpenChangedEvent(void)
- const char *getpercentageChangedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *getstartAngleChangedEvent(void)
- const char *getvalueChangedEvent(void)

91.227 QPixmap Class

C++ Reference : http://doc.qt.io/qt-5/qpixmap.html

Parameters : const char *

- QPixmap transformed(QTransform transform, Qt::TransformationMode mode)
- QPixmap copy(int x, int y, int width, int height)
- QPixmap scaled(int width, int height, Qt::AspectRatioMode aspectRatioMode, Qt::TransformationMode transformMode)
- int width(void)
- int height(void)
- QBitmap createMaskFromColor(QColor , Qt::MaskMode)
- QBitmap mask(void)
- void setMask(QBitmap)

- void fill(QColor)
- QPixmap fromImage(QImage,Qt::ImageConversionFlags)
- bool load(QString, const char *, Qt::ImageConversionFlags)
- qint64 cacheKey(void)
- bool convertFromImage(QImage image, Qt::ImageConversionFlags flags)
- QPixmap copy_2(QRect rectangle)
- QBitmap createHeuristicMask(bool clipTight)
- int depth(void)
- void detach(void)
- qreal devicePixelRatio(void)
- bool hasAlpha(void)
- bool hasAlphaChannel(void)
- bool isNull(void)
- bool isQBitmap(void)
- bool loadFromData(uchar *data, uint len, char *format, Qt::ImageConversionFlags flags)
- bool loadFromData_2(QByteArray data, char *format, Qt::ImageConversionFlags flags)
- QRect rect(void)
- bool save(QString fileName, char *format, int quality)
- bool save_2(QIODevice *device, char *format, int quality)
- QPixmap scaled_2(QSize size, Qt::AspectRatioMode aspectRatioMode, Qt::TransformationMode transformMode)
- QPixmap scaledToHeight(int height, Qt::TransformationMode mode)
- QPixmap scaledToWidth(int width, Qt::TransformationMode mode)
- void scroll(int dx, int dy, int x, int y, int width, int height, QRegion *exposed)
- void scroll_2(int dx, int dy, QRect rect, QRegion *exposed)
- void setDevicePixelRatio(qreal scaleFactor)
- QSize size(void)
- void swap(QPixmap other)
- QImage toImage(void)
- QPixmap transformed_2(QTransform matrix, Qt::TransformationMode mode)
- int defaultDepth(void)
- QPixmap fromImage_2(QImage image, Qt::ImageConversionFlags flags)
- QPixmap fromImageReader(QImageReader *imageReader, Qt::ImageConversionFlags flags)
- QTransform trueMatrix(QTransform matrix, int width, int height)

91.228 QPixmap2 Class

Parameters : int width, int height

Parent Class : QPixmap

91.229 QPlainTextEdit Class

C++ Reference : http://doc.qt.io/qt-5/qplaintextedit.html

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- void setTabStopDistance(qreal width)
- qreal tabStopDistance(void)
- QString anchorAt(QPoint pos)
- bool backgroundVisible(void)
- int blockCount(void)
- bool canPaste(void)
- bool centerOnScroll(void)
- QMenu * createStandardContextMenu(void)
- QTextCharFormat currentCharFormat(void)
- QTextCursor cursorForPosition(QPoint pos)
- QRect cursorRect(QTextCursor cursor)
- QRect cursorRect_2(void)
- int cursorWidth(void)
- QTextDocument * document(void)
- QString documentTitle(void)
- void ensureCursorVisible(void)
- QList<QTextEdit::ExtraSelection> extraSelections(void)
- bool find(QString exp, QTextDocument::FindFlags options)
- bool isReadOnly(void)
- bool isUndoRedoEnabled(void)
- QPlainTextEdit::LineWrapMode lineWrapMode(void)
- int maximumBlockCount(void)
- void mergeCurrentCharFormat(QTextCharFormat modifier)
- void moveCursor(QTextCursor::MoveOperation operation, QTextCursor::MoveMode mode)
- bool overwriteMode(void)
- void print(QPagedPaintDevice *printer)
- void setBackgroundVisible(bool visible)

- void setCenterOnScroll(bool enabled)
- void setCurrentCharFormat(QTextCharFormat format)
- void setCursorWidth(int width)
- void setDocument(QTextDocument *document)
- void setDocumentTitle(QString title)
- void setExtraSelections(QList<QTextEdit::ExtraSelection> selections)
- void setLineWrapMode(QPlainTextEdit::LineWrapMode mode)
- void setMaximumBlockCount(int maximum)
- void setOverwriteMode(bool overwrite)
- void setReadOnly(bool ro)
- void setTabChangesFocus(bool b)
- void setTextCursor(QTextCursor cursor)
- void setTextInteractionFlags(Qt::TextInteractionFlags flags)
- void setUndoRedoEnabled(bool enable)
- void setWordWrapMode(QTextOption::WrapMode policy)
- bool tabChangesFocus(void)
- QTextCursor textCursor(void)
- Qt::TextInteractionFlags textInteractionFlags(void)
- QString toPlainText(void)
- QTextOption::WrapMode wordWrapMode(void)
- void appendHtml(QString html)
- void appendPlainText(QString text)
- void centerCursor(void)
- void clear(void)
- void copy(void)
- void cut(void)
- void insertPlainText(QString text)
- void paste(void)
- void redo(void)
- void selectAll(void)
- void setPlainText(QString text)
- void undo(void)
- void zoomIn(int range)
- void zoomOut(int range)
- void setblockCountChangedEvent(const char *cStr)
- void setcopyAvailableEvent(const char *cStr)

- void setcursorPositionChangedEvent(const char *cStr)
- void setmodificationChangedEvent(const char *cStr)
- void setredoAvailableEvent(const char *cStr)
- void setselectionChangedEvent(const char *cStr)
- void settextChangedEvent(const char *cStr)
- void setundoAvailableEvent(const char *cStr)
- void setupdateRequestEvent(const char *cStr)
- const char *getblockCountChangedEvent(void)
- const char *getcopyAvailableEvent(void)
- const char *getcursorPositionChangedEvent(void)
- const char *getmodificationChangedEvent(void)
- const char *getredoAvailableEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettextChangedEvent(void)
- const char *getundoAvailableEvent(void)
- const char *getupdateRequestEvent(void)
- void cyanline(void)
- void setactivelinecolor(QColor)

91.230 QPlaneMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qplanemesh.html

Parameters : Qt3DCore::QNode *

- float height(void)
- QSize meshResolution(void)
- bool mirrored(void)
- float width(void)
- void setHeight(float height)
- void setMeshResolution(QSize resolution)
- void setMirrored(bool mirrored)
- void setWidth(float width)

91.231 QPoint Class

C++ Reference : http://doc.qt.io/qt-5/qpoint.html

Parameters : void

- bool isNull(void)
- int manhattanLength(void)
- int rx(void)
- int ry(void)
- void setX(int x)
- void setY(int y)
- int x(void)
- int y(void)

91.232 QPointF Class

C++ Reference : http://doc.qt.io/qt-5/qpointf.html

Parameters : void

- bool isNull(void)
- qreal manhattanLength(void)
- qreal rx(void)
- qreal ry(void)
- void setX(qreal x)
- void setY(qreal y)
- QPoint toPoint(void)
- qreal x(void)
- qreal y(void)

91.233 QPointLight Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qpointlight.html

Parameters : Qt3DCore::QNode *

- float constantAttenuation(void)
- float linearAttenuation(void)
- float quadraticAttenuation(void)
- void setConstantAttenuation(float value)
- void setLinearAttenuation(float value)
- void setQuadraticAttenuation(float value)

- void setColor(QColor)
- void setIntensity(float intensity)

91.234 QPolarChart Class

C++ Reference : http://doc.qt.io/qt-5/qpolarchart.html

Parameters : QGraphicsItem *,Qt::WindowFlags

Parent Class : QChart

- QList<QAbstractAxis *> axes(QPolarChart::PolarOrientations polarOrientation, QAbstractSeries *series)

91.235 QPrintDialog Class

C++ Reference : http://doc.qt.io/qt-5/qprintdialog.html

Parameters : QPrinter *,QWidget *

Parent Class : QAbstractPrintDialog

- void open(QObject * receiver, char * member)
- QAbstractPrintDialog::PrintDialogOptions options(void)
- QPrinter * printer(void)
- void setOption(QAbstractPrintDialog::PrintDialogOption option, bool on)
- void setOptions(QAbstractPrintDialog::PrintDialogOptions options)
- bool testOption(QAbstractPrintDialog::PrintDialogOption option)
- void setacceptedEvent(const char *)
- const char *getacceptedEvent(void)

91.236 QPrintPreviewDialog Class

C++ Reference : http://doc.qt.io/qt-5/qprintpreviewdialog.html

Parameters : QPrinter *

Parent Class : QDialog

- void open(QObject * receiver, char * member)
- QPrinter * printer(void)
- void setpaintRequestedEvent(const char *)
- const char *getpaintRequestedEvent(void)

91.237 QPrintPreviewWidget Class

C++ Reference : http://doc.qt.io/qt-5/qprintpreviewwidget.html

Parameters : QPrinter *

Parent Class : QWidget

- int currentPage(void)
- QPrinter::Orientation orientation(void)
- int pageCount(void)
- QPrintPreviewWidget::ViewMode viewMode(void)
- qreal zoomFactor(void)
- QPrintPreviewWidget::ZoomMode zoomMode(void)
- void fitInView(void)
- void fitToWidth(void)
- void print(void)
- void setAllPagesViewMode(void)
- void setCurrentPage(int page)
- void setFacingPagesViewMode(void)
- void setLandscapeOrientation(void)
- void setOrientation(QPrinter::Orientation orientation)
- void setPortraitOrientation(void)
- void setSinglePageViewMode(void)
- void setViewMode(QPrintPreviewWidget::ViewMode mode)
- void setZoomFactor(qreal factor)
- void setZoomMode(QPrintPreviewWidget::ZoomMode zoomMode)
- void updatePreview(void)
- void zoomIn(qreal factor)
- void zoomOut(qreal factor)
- void setpaintRequestedEvent(const char *)
- void setpreviewChangedEvent(const char *)
- const char *getpaintRequestedEvent(void)
- const char *getpreviewChangedEvent(void)

91.238 QPrinter Class

C++ Reference : http://doc.qt.io/qt-5/qprinter.html

Parameters : QPrinter::PrinterMode

- bool abort(void)
- bool collateCopies(void)
- int colorMode(void)
- int copyCount(void)
- QString creator(void)
- QString docName(void)
- int duplex(void)
- bool fontEmbeddingEnabled(void)
- int fromPage(void)
- bool fullPage(void)
- bool isValid(void)
- QString outputFileName(void)
- int outputFormat(void)
- QRectF pageRect(QPrinter::Unit unit)
- QRectF paperRect(QPrinter::Unit unit)
- int paperSource(void)
- QPrintEngine *printEngine(void)
- QString printProgram(void)
- int printRange(void)
- QString printerName(void)
- QString printerSelectionOption(void)
- int printerState(void)
- int resolution(void)
- void setCollateCopies(bool collate)
- void setColorMode(QPrinter::ColorMode newColorMode)
- void setCopyCount(int count)
- void setCreator(QString)
- void setDocName(QString)
- void setDuplex(QPrinter::DuplexMode duplex)
- void setFontEmbeddingEnabled(bool enable)
- void setFromTo(int from, int to)
- void setFullPage(bool fp)

- void setOutputFileName(QString)
- void setOutputFormat(QPrinter::OutputFormat format)
- void setPrintProgram(QString)
- void setPrintRange(QPrinter::PrintRange)
- void setPrinterName(QString)
- void setPrinterSelectionOption(QString)
- void setResolution(int dpi)
- bool supportsMultipleCopies(void)
- int toPage(void)
- bool newPage(void)
- QPaintEngine *paintEngine(void)
- void setPageSizeMM(QSizeF)

91.239 QPrinterInfo Class

C++ Reference : http://doc.qt.io/qt-5/qprinterinfo.html

Parameters : void

- QString description(void)
- bool isDefault(void)
- bool isNull(void)
- QString location(void)
- QString makeAndModel(void)
- QString printerName(void)
- QPrinterInfo printerInfo(QString printerName

91.240 QProcess Class

C++ Reference : http://doc.qt.io/qt-5/qprocess.html

Parameters : QObject *

Parent Class : QIODevice

- QStringList arguments(void)
- void closeReadChannel(QProcess::ProcessChannel channel)
- void closeWriteChannel(void)
- QProcess::ProcessError error(void)
- int exitCode(void)
- QProcess::ExitStatus exitStatus(void)
- QProcess::InputChannelMode inputChannelMode(void)

- QProcess::ProcessChannelMode processChannelMode(void)
- QProcessEnvironment processEnvironment(void)
- QString program(void)
- QByteArray readAllStandardError(void)
- QByteArray readAllStandardOutput(void)
- QProcess::ProcessChannel readChannel(void)
- void setArguments(QStringList arguments)
- void setInputChannelMode(QProcess::InputChannelMode mode)
- void setProcessChannelMode(QProcess::ProcessChannelMode mode)
- void setProcessEnvironment(QProcessEnvironment environment)
- void setProgram(QString program)
- void setReadChannel(QProcess::ProcessChannel channel)
- void setStandardErrorFile(QString fileName, QIODevice::OpenMode mode)
- void setStandardInputFile(QString fileName)
- void setStandardOutputFile(QString fileName, QIODevice::OpenMode mode)
- void setStandardOutputProcess(QProcess *destination)
- void setWorkingDirectory(QString dir)
- void start(QString program, QStringList arguments, QIODevice::OpenMode mode)
- void start_3(QIODevice::OpenMode mode)
- QProcess::ProcessState state(void)
- bool waitForFinished(int msecs)
- bool waitForStarted(int msecs)
- QString workingDirectory(void)
- void kill(void)
- void terminate(void)
- void setreadyReadStandardErrorEvent(const char *)
- void setreadyReadStandardOutputEvent(const char *)
- const char *getreadyReadStandardErrorEvent(void)
- const char *getreadyReadStandardOutputEvent(void)

91.241 QProgressBar Class

C++ Reference : http://doc.qt.io/qt-5/qprogressbar.html

Parameters : QWidget *parent

Parent Class : QWidget

- int alignment(void)
- QString format(void)
- bool invertedAppearance(void)
- bool isTextVisible(void)
- int maximum(void)
- int minimum(void)
- int orientation(void)
- void resetFormat(void)
- void setAlignment(Qt::AlignmentFlag alignment)
- void setFormat(QString)
- void setInvertedAppearance(bool invert)
- void setTextDirection(QProgressBar::Direction textDirection)
- void setTextVisible(bool visible)
- QString text(void)
- int textDirection(void)
- int value(void)
- void reset(void)
- void setMaximum(int maximum)
- void setMinimum(int minimum)
- void setOrientation(Qt::Orientation)
- void setRange(int minimum, int maximum)
- void setValue(int value)
- void setvalueChangedEvent(const char *)
- const char *getvalueChangedEvent(void)

91.242 QPushButton Class

C++ Reference : http://doc.qt.io/qt-5/qpushbutton.html

Parameters : QWidget *

Parent Class : QAbstractButton

- void setText(const char *)
- void setClickEvent(const char *)
- void setIcon(QIcon)
- void setIconSize(QSize)
- const char *getClickEvent(void)

91.243 QQmlEngine Class

C++ Reference : http://doc.qt.io/qt-5/qqmlengine.html

Parameters : QObject *

- void addImageProvider(QString providerId, QQmlImageProviderBase *provider)
- void addImportPath(QString path)
- void addPluginPath(QString path)
- QUrl baseUrl(void)
- void clearComponentCache(void)
- QQmlImageProviderBase *imageProvider(QString providerId)
- QStringList importPathList(void)
- bool importPlugin(QString filePath, QString uri, QList<QQmlError> *errors)
- QQmlIncubationController *incubationController(void)
- QNetworkAccessManager *networkAccessManager(void)
- QQmlNetworkAccessManagerFactory *networkAccessManagerFactory(void)
- QString offlineStorageDatabaseFilePath(QString databaseName)
- QString offlineStoragePath(void)
- bool outputWarningsToStandardError(void)
- QStringList pluginPathList(void)
- void removeImageProvider(QString providerId)
- QQmlContext *rootContext(void)
- void setBaseUrl(QUrl url)
- void setImportPathList(QStringList paths)
- void setIncubationController(QQmlIncubationController *controller)
- void setNetworkAccessManagerFactory(QQmlNetworkAccessManagerFactory *factory)
- void setOfflineStoragePath(QString dir)

- void setOutputWarningsToStandardError(bool enabled)
- void setPluginPathList(QStringList paths)
- void trimComponentCache(void)
- void retranslate(void)
- QQmlContext * contextForObject(QObject *object)
- QQmlEngine::ObjectOwnership objectOwnership(QObject *object)
- void setContextForObject(QObject *object, QQmlContext *context)
- void setObjectOwnership(QObject *object, QQmlEngine::ObjectOwnership ownership)

91.244 QQmlError Class

C++ Reference : http://doc.qt.io/qt-5/qqmlerror.html

Parameters : void

- int column(void)
- QString description(void)
- bool isValid(void)
- int line(void)
- QObject * object(void)
- void setColumn(int column)
- void setDescription(QString description)
- void setLine(int line)
- void setObject(QObject *object)
- void setUrl(QUrl url)
- QString toString(void)
- QUrl url(void)

91.245 QQuaternion Class

C++ Reference : http://doc.qt.io/qt-5/qquaternion.html

Parameters : float,float,float,float

- bool isNull(void)
- float length(void)
- float lengthSquared(void)
- void normalize(void)
- QQuaternion normalized(void)
- QVector3D rotatedVector(QVector3D vector)
- float scalar(void)

- void setScalar(float scalar)
- void setVector(QVector3D vector)
- void setVector_2(float x, float y, float z)
- void setX(float x)
- void setY(float y)
- void setZ(float z)
- QVector3D vector(void)
- float x(void)
- float y(void)
- float z(void)
- QQuaternion fromAxisAndAngle_2(float x, float y, float z, float angle)
- QQuaternion nlerp(QQuaternion q1, QQuaternion q2, float t)
- QQuaternion slerp(QQuaternion q1, QQuaternion q2, float t)

91.246 QQuickView Class

C++ Reference : http://doc.qt.io/qt-5/qquickview.html

Parameters : void

Parent Class : QWindow

- QQmlEngine *engine(void)
- QList<QQmlError> errors(void)
- QSize initialSize(void)
- QQuickWidget::ResizeMode resizeMode(void)
- QQmlContext *rootContext(void)
- QQuickItem *rootObject(void)
- void setFormat(QSurfaceFormat format)
- void setResizeMode(QQuickView::ResizeMode)
- QUrl source(void)
- QQuickWidget::Status status(void)
- void setSource(QUrl url)

91.247 QQuickWidget Class

C++ Reference : http://doc.qt.io/qt-5/qquickwidget.html

Parameters : QWidget *

Parent Class : QWidget

- QQmlEngine *engine(void)
- QList<QQmlError> errors(void)
- QSurfaceFormat format(void)
- QImage grabFramebuffer(void)
- QSize initialSize(void)
- QQuickWindow *quickWindow(void)
- QQuickWidget::ResizeMode resizeMode(void)
- QQmlContext *rootContext(void)
- QQuickItem *rootObject(void)
- void setClearColor(QColor color)
- void setFormat(QSurfaceFormat format)
- void setResizeMode(QQuickWidget::ResizeMode)
- QUrl source(void)
- QQuickWidget::Status status(void)
- void setSource(QUrl url)
- void setsceneGraphErrorEvent(const char *)
- void setstatusChangedEvent(const char *)
- const char *getsceneGraphErrorEvent(void)
- const char *getstatusChangedEvent(void)

91.248 QRadioButton Class

C++ Reference : http://doc.qt.io/qt-5/qradiobutton.html

Parameters : QWidget *parent

Parent Class : QAbstractButton

- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setclickedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- void settoggledEvent(const char *)
- const char *getclickedEvent(void)

- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)
- const char *gettoggledEvent(void)

91.249 QRect Class

C++ Reference : http://doc.qt.io/qt-5/qrect.html

Parameters : void

- void adjust(int dx1, int dy1, int dx2, int dy2)
- QRect adjusted(int dx1, int dy1, int dx2, int dy2)
- int bottom(void)
- QPoint bottomLeft(void)
- QPoint bottomRight(void)
- QPoint center(void)
- bool contains(int x, int y, bool proper)
- void getCoords(int *x1, int *y1, int *x2, int *y2)
- void getRect(int *x, int *y, int *width, int *height)
- int height(void)
- QRect intersected(QRect)
- bool intersects(QRect)
- bool isEmpty(void)
- bool isNull(void)
- bool isValid(void)
- int left(void)
- void moveBottom(int y)
- void moveBottomLeft(QPoint)
- void moveBottomRight(QPoint)
- void moveCenter(QPoint)
- void moveLeft(int x)
- void moveRight(int x)
- void moveTo(int x, int y)
- void moveTop(int y)
- void moveTopLeft(QPoint)
- void moveTopRight(QPoint)
- QRect normalized(void)
- int right(void)

- void setBottom(int y)
- void setBottomLeft(QPoint)
- void setBottomRight(QPoint)
- void setCoords(int x1, int y1, int x2, int y2)
- void setHeight(int height)
- void setLeft(int x)
- void setRect(int x, int y, int width, int height)
- void setRight(int x)
- void setSize(QSize)
- void setTop(int y)
- void setTopLeft(QPoint)
- void setTopRight(QPoint)
- void setWidth(int width)
- void setX(int x)
- void setY(int y)
- QSize size(void)
- int top(void)
- QPoint topLeft(void)
- QPoint topRight(void)
- void translate(int dx, int dy)
- QRect translated(int dx, int dy)
- QRect united(QRect)
- int width(void)
- int x(void)
- int y(void)

91.250 QRegion Class

C++ Reference : http://doc.qt.io/qt-5/qregion.html

Parameters : void

- QRect boundingRect(void)
- bool contains(QPoint p)
- bool contains_2(QRect r)
- QRegion intersected(QRegion r)
- QRegion intersected_2(QRect rect)
- bool intersects(QRegion region)

- bool intersects_2(QRect rect)
- bool isEmpty(void)
- bool isNull(void)
- int rectCount(void)
- void setRects(QRect *rects, int number)
- QRegion subtracted(QRegion r)
- void swap(QRegion other)
- void translate(int dx, int dy)
- void translate_2(QPoint point)
- QRegion translated(int dx, int dy)
- QRegion translated_2(QPoint p)
- QRegion united(QRegion r)
- QRegion united_2(QRect rect)
- QRegion xored(QRegion r)

91.251 QRegularExpression Class

C++ Reference : http://doc.qt.io/qt-5/qregularexpression.html

Parameters : void

- int captureCount(void)
- QString errorString(void)
- QRegularExpressionMatchIterator globalMatch(QString subject, int offset, QRegularExpression::MatchType matchType, QRegularExpression::MatchOptions matchOptions)
- bool isValid(void)
- QRegularExpressionMatch match(QString subject, int offset, QRegularExpression::MatchType matchType, QRegularExpression::MatchOptions matchOptions)
- QStringList namedCaptureGroups(void)
- QString pattern(void)
- int patternErrorOffset(void)
- QRegularExpression::PatternOptions patternOptions(void)
- void setPattern(QString pattern)
- void setPatternOptions(QRegularExpression::PatternOptions options)
- void swap(QRegularExpression other)

91.252 QRegularExpressionMatch Class

C++ Reference : http://doc.qt.io/qt-5/qregularexpressionmatch.html

Parameters : void

- QString captured(int nth)
- QString captured_2(const QString name)
- int capturedEnd(int nth)
- int capturedEnd_2(const QString name)
- int capturedLength(int nth)
- int capturedLength_2(const QString name)
- QStringRef capturedRef(int nth)
- QStringRef capturedRef_2(const QString name)
- int capturedStart(int nth)
- int capturedStart_2(const QString name)
- QStringList capturedTexts(void)
- bool hasMatch(void)
- bool hasPartialMatch(void)
- bool isValid(void)
- int lastCapturedIndex(void)
- QRegularExpression::MatchOptions matchOptions(void)
- QRegularExpression::MatchType matchType(void)
- QRegularExpression regularExpression(void)
- void swap(QRegularExpressionMatch other)

91.253 QRegularExpressionMatchIterator Class

C++ Reference : http://doc.qt.io/qt-5/qregularexpressionmatchiterator.html

Parameters : void

- bool hasNext(void)
- bool isValid(void)
- QRegularExpression::MatchOptions matchOptions(void)
- QRegularExpression::MatchType matchType(void)
- QRegularExpressionMatch next(void) # In RingQt use : QRegularExpressionMatch nextitem(void)
- QRegularExpressionMatch peekNext(void)
- QRegularExpression regularExpression(void)
- void swap(QRegularExpressionMatchIterator other)

91.254 QRenderAspect Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qrenderaspect.html

Parameters : QObject *

91.255 QRenderPass Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qrenderpass.html

Parameters : Qt3DCore::QNode *

- void addFilterKey(Qt3DRender::QFilterKey *filterKey)
- void addParameter(Qt3DRender::QParameter *parameter)
- void addRenderState(Qt3DRender::QRenderState *state)
- QVector<Qt3DRender::QFilterKey *> filterKeys(void)
- QVector<Qt3DRender::QParameter *> parameters(void)
- void removeFilterKey(Qt3DRender::QFilterKey *filterKey)
- void removeParameter(Qt3DRender::QParameter *parameter)
- void removeRenderState(Qt3DRender::QRenderState *state)
- QVector<Qt3DRender::QRenderState *> renderStates(void)
- Qt3DRender::QShaderProgram * shaderProgram(void)
- void setShaderProgram(Qt3DRender::QShaderProgram *shaderProgram)

91.256 QScatterSeries Class

C++ Reference : http://doc.qt.io/qt-5/qscatterseries.html

Parameters : QObject *

Parent Class : QXYSeries

- QBrush brush(void)
- QScatterSeries::MarkerShape markerShape(void)
- qreal markerSize(void)
- void setBorderColor(QColor color)
- void setMarkerShape(QScatterSeries::MarkerShape shape)
- void setMarkerSize(qreal size)
- void setborderColorChangedEvent(const char *)
- void setcolorChangedEvent(const char *)
- void setmarkerShapeChangedEvent(const char *)
- void setmarkerSizeChangedEvent(const char *)
- const char *getborderColorChangedEvent(void)

- const char *getcolorChangedEvent(void)
- const char *getmarkerShapeChangedEvent(void)
- const char *getmarkerSizeChangedEvent(void)

91.257 QSceneLoader Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qsceneloader.html

Parameters : Qt3DCore::QNode *

Parent Class : QComponent

- Qt3DCore::QComponent * component(QString entityName, Qt3DRender::QSceneLoader::ComponentType componentType)
- Qt3DCore::QEntity * entity(QString entityName)
- QStringList entityNames(void)
- QUrl source(void)
- Qt3DRender::QSceneLoader::Status status(void)
- void setSource(QUrl arg)

91.258 QScreen Class

C++ Reference : http://doc.qt.io/qt-5/qscreen.html

- int angleBetween(Qt::ScreenOrientation a, Qt::ScreenOrientation b)
- QRect availableGeometry(void)
- QSize availableSize(void)
- QRect availableVirtualGeometry(void)
- QSize availableVirtualSize(void)
- int depth(void)
- qreal devicePixelRatio(void)
- QRect geometry(void)
- QPixmap grabWindow(int window, int x, int y, int width, int height)
- QPixmap grabWindow_2(int window)
- QPlatformScreen * handle(void)
- bool isLandscape(Qt::ScreenOrientation o)
- bool isPortrait(Qt::ScreenOrientation o)
- qreal logicalDotsPerInch(void)
- qreal logicalDotsPerInchX(void)
- qreal logicalDotsPerInchY(void)
- QRect mapBetween(Qt::ScreenOrientation a, Qt::ScreenOrientation b, QRect rect)

- QString name(void)
- Qt::ScreenOrientation nativeOrientation(void)
- Qt::ScreenOrientation orientation(void)
- Qt::ScreenOrientations orientationUpdateMask(void)
- qreal physicalDotsPerInch(void)
- qreal physicalDotsPerInchX(void)
- qreal physicalDotsPerInchY(void)
- QSizeF physicalSize(void)
- Qt::ScreenOrientation primaryOrientation(void)
- qreal refreshRate(void)
- void setOrientationUpdateMask(Qt::ScreenOrientations mask)
- QSize size(void)
- QTransform transformBetween(Qt::ScreenOrientation a, Qt::ScreenOrientation b, QRect target)

91.259 QScrollArea Class

C++ Reference : http://doc.qt.io/qt-5/qscrollarea.html

Parameters : QWidget *parent

Parent Class : QAbstractScrollArea

- Qt::Alignment alignment(void)
- void ensureVisible(int x, int y, int xmargin , int ymargin)
- void ensureWidgetVisible(QWidget *childWidget, int xmargin , int ymargin)
- void setAlignment(Qt::Alignment)
- void setWidget(QWidget *widget)
- void setWidgetResizable(bool resizable)
- QWidget *takeWidget(void)
- QWidget *widget(void)
- bool widgetResizable(void)

91.260 QScrollBar Class

C++ Reference : http://doc.qt.io/qt-5/qscrollbar.html

Parameters : QWidget *parent

Parent Class : QAbstractSlider

91.261 QSerialPort Class

C++ Reference : http://doc.qt.io/qt-5/qserialport.html

Parameters : QObject *

Parent Class : QIODevice

- qint32 baudRate(QSerialPort::Directions directions)
- bool clear(QSerialPort::Directions directions)
- void clearError(void)
- QSerialPort::DataBits dataBits(void)
- QSerialPort::SerialPortError error(void)
- QSerialPort::FlowControl flowControl(void)
- bool flush(void)
- void *handle(void)
- bool isDataTerminalReady(void)
- bool isRequestToSend(void)
- QSerialPort::Parity parity(void)
- QSerialPort::PinoutSignals pinoutSignals(void)
- QString portName(void)
- qint64 readBufferSize(void)
- bool setBaudRate(qint32 baudRate, QSerialPort::Directions directions)
- bool setBreakEnabled(bool set)
- bool setDataBits(QSerialPort::DataBits dataBits)
- bool setDataTerminalReady(bool set)
- bool setFlowControl(QSerialPort::FlowControl flowControl)
- bool setParity(QSerialPort::Parity parity)
- void setPort(QSerialPortInfo serialPortInfo)
- void setPortName(QString name)
- void setReadBufferSize(qint64 size)
- bool setRequestToSend(bool set)
- bool setStopBits(QSerialPort::StopBits stopBits)
- QSerialPort::StopBits stopBits(void)
- void setbaudRateChangedEvent(const char *)
- void setbreakEnabledChangedEvent(const char *)
- void setdataBitsChangedEvent(const char *)
- void setdataTerminalReadyChangedEvent(const char *)
- void seterrorEvent(const char *)

- void setflowControlChangedEvent(const char *)
- void setparityChangedEvent(const char *)
- void setrequestToSendChangedEvent(const char *)
- void setstopBitsChangedEvent(const char *)
- const char *getbaudRateChangedEvent(void)
- const char *getbreakEnabledChangedEvent(void)
- const char *getdataBitsChangedEvent(void)
- const char *getdataTerminalReadyChangedEvent(void)
- const char *geterrorEvent(void)
- const char *getflowControlChangedEvent(void)
- const char *getparityChangedEvent(void)
- const char *getrequestToSendChangedEvent(void)
- const char *getstopBitsChangedEvent(void)

91.262 QSerialPortInfo Class

C++ Reference : http://doc.qt.io/qt-5/qserialportinfo.html

Parameters : void

- QString description(void)
- bool hasProductIdentifier(void)
- bool hasVendorIdentifier(void)
- bool isNull(void)
- QString manufacturer(void)
- QString portName(void)
- quint16 productIdentifier(void)
- void swap(QSerialPortInfo other)
- QString systemLocation(void)
- quint16 vendorIdentifier(void)

91.263 QSize Class

C++ Reference : http://doc.qt.io/qt-5/qsize.html

Parameters : int width, int height

91.264 QSkyboxEntity Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qskyboxentity.html

Parameters : Qt3DCore::QNode *

Parent Class : QEntity

- QString baseName(void)
- QString extension(void)
- bool isGammaCorrectEnabled(void)
- void setBaseName(QString baseName)
- void setExtension(QString extension)
- void setGammaCorrectEnabled(bool enabled)

91.265 QSlider Class

C++ Reference : http://doc.qt.io/qt-5/qslider.html

Parameters : QWidget *parent

Parent Class : QAbstractSlider

- void setTickInterval(int ti)
- void setTickPosition(QSlider::TickPosition position)
- int tickInterval(void)
- int tickPosition(void)
- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setactionTriggeredEvent(const char *)
- void setrangeChangedEvent(const char *)
- void setsliderMovedEvent(const char *)
- void setsliderPressedEvent(const char *)
- void setsliderReleasedEvent(const char *)
- void setvalueChangedEvent(const char *)
- const char *getactionTriggeredEvent(void)
- const char *getrangeChangedEvent(void)
- const char *getsliderMovedEvent(void)
- const char *getsliderPressedEvent(void)
- const char *getsliderReleasedEvent(void)
- const char *getvalueChangedEvent(void)

91.266 QSphereMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qspheremesh.html

Parameters : Qt3DCore::QNode *

- bool generateTangents(void)
- float radius(void)
- int rings(void)
- int slices(void)
- void setGenerateTangents(bool gen)
- void setRadius(float radius)
- void setRings(int rings)
- void setSlices(int slices)

91.267 QSpinBox Class

C++ Reference : http://doc.qt.io/qt-5/qspinbox.html

Parameters : QWidget *parent

Parent Class : QWidget

- QString cleanText(void)
- int displayIntegerBase(void)
- int maximum(void)
- int minimum(void)
- QString prefix(void)
- void setDisplayIntegerBase(int base)
- void setMaximum(int max)
- void setMinimum(int min)
- void setPrefix(QString)
- void setRange(int minimum, int maximum)
- void setSingleStep(int val)
- void setSuffix(QString)
- int singleStep(void)
- QString suffix(void)
- int value(void)
- void setValue(int val)
- void setvalueChangedEvent(const char *)
- const char *getvalueChangedEvent(void)

91.268 QSplashScreen Class

C++ Reference : http://doc.qt.io/qt-5/qsplashscreen.html

Parameters : QPixmap

Parent Class : QWidget

- void finish(QWidget *mainWin)
- QPixmap pixmap(void)
- void repaint(void)
- void setPixmap(QPixmap pixmap)
- void clearMessage(void)
- void showMessage(QString message, int alignment ,QColor color)

91.269 QSplineSeries Class

C++ Reference : http://doc.qt.io/qt-5/qsplineseries.html

Parameters : QObject *

Parent Class : QLineSeries

- QAbstractSeries::SeriesType type(void)

91.270 QSplitter Class

C++ Reference : http://doc.qt.io/qt-5/qsplitter.html

Parameters : QWidget *parent

Parent Class : QFrame

- void addWidget(QWidget *widget)
- bool childrenCollapsible(void)
- int count(void)
- void getRange(int index, int *min, int *max)
- QSplitterHandle * handle(int index)
- int handleWidth(void)
- int indexOf(QWidget *widget)
- void insertWidget(int index, QWidget *widget)
- bool isCollapsible(int index)
- bool opaqueResize(void)
- Qt::Orientation orientation(void)
- void refresh(void)
- bool restoreState(QByteArray state)

- QByteArray saveState(void)
- void setChildrenCollapsible(bool)
- void setCollapsible(int index, bool collapse)
- void setHandleWidth(int)
- void setOpaqueResize(bool opaque)
- void setOrientation(Qt::Orientation)
- void setSizes(QList<int> list)
- void setStretchFactor(int index, int stretch)
- QList<int> sizes(void)
- QWidget * widget(int index)

91.271 QSqlDatabase Class

C++ Reference : http://doc.qt.io/qt-5/qsqldatabase.html

Parameters : void

- void close(void)
- bool commit(void)
- QString connectOptions(void)
- QString connectionName(void)
- QString databaseName(void)
- QSqlDriver *driver(void)
- QString driverName(void)
- QSqlQuery exec(QString)
- QString hostName(void)
- bool isOpen(void)
- bool isOpenError(void)
- bool isValid(void)
- QSqlError lastError(void)
- QSql::NumericalPrecisionPolicy numericalPrecisionPolicy(void)
- bool open(void)
- QString password(void)
- int port(void)
- QSqlIndex primaryIndex(QString)
- QSqlRecord record(QString)
- bool rollback(void)
- void setConnectOptions(QString)

- void setDatabaseName(QString)
- void setHostName(QString)
- void setNumericalPrecisionPolicy(QSql::NumericalPrecisionPolicy precisionPolicy)
- void setPassword(QString)
- void setPort(int port)
- void setUserName(QString)
- QStringList tables(QSql::TableType type)
- bool transaction(void)
- QString userName(void)
- QSqlDatabase addDatabase(QString)
- QSqlDatabase cloneDatabase(QSqlDatabase, QString)
- QStringList connectionNames(void)
- bool contains(QString)
- QSqlDatabase database(QString , bool)
- QStringList drivers(void)
- bool isDriverAvailable(QString)
- void registerSqlDriver(QString, QSqlDriverCreatorBase *)
- void removeDatabase(QString)

91.272 QSqlDriver Class

C++ Reference : http://doc.qt.io/qt-5/qsqldriver.html

Parameters : void

- QSqlError lastError(void)
- QSql::NumericalPrecisionPolicy numericalPrecisionPolicy(void)
- void setNumericalPrecisionPolicy(QSql::NumericalPrecisionPolicy)

91.273 QSqlDriverCreatorBase Class

C++ Reference : http://doc.qt.io/qt-5/qsqldrivercreatorbase.html

Parameters : void

91.274 QSqlError Class

C++ Reference : http://doc.qt.io/qt-5/qsqlerror.html

Parameters : QString, QString, QSqlError::ErrorType

- QString databaseText(void)
- QString driverText(void)
- bool isValid(void)
- QString text(void)
- QSqlError::ErrorType type(void)

91.275 QSqlField Class

C++ Reference : http://doc.qt.io/qt-5/qsqlfield.html

Parameters : QString,QVariant::Type

- void clear(void)
- QVariant defaultValue(void)
- bool isAutoValue(void)
- bool isGenerated(void)
- bool isNull(void)
- bool isReadOnly(void)
- bool isValid(void)
- int length(void)
- QString name(void)
- int precision(void)
- RequiredStatus requiredStatus(void)
- void setAutoValue(bool autoVal)
- void setDefaultValue(QVariant)
- void setGenerated(bool gen)
- void setLength(int fieldLength)
- void setName(QString)
- void setPrecision(int precision)
- void setReadOnly(bool readOnly)
- void setRequired(bool required)
- void setRequiredStatus(QSqlField::RequiredStatus required)
- void setType(QVariant::Type type)
- void setValue(QVariant)
- QVariant::Type type(void)

- QVariant value(void)

91.276 QSqlIndex Class

C++ Reference : http://doc.qt.io/qt-5/qsqlindex.html

Parameters : QString, QString

Parent Class : QSqlRecord

- void append(QSqlField, bool)
- QString cursorName(void)
- bool isDescending(int i)
- QString name(void)
- void setCursorName(QString)
- void setDescending(int i, bool desc)
- void setName(QString)

91.277 QSqlQuery Class

C++ Reference : http://doc.qt.io/qt-5/qsqlquery.html

Parameters : void

- void addBindValue(QVariant, QSql::ParamType paramType)
- int at(void)
- void bindValue(QString, QVariant, QSql::ParamType paramType)
- QVariant boundValue(QString)
- void clear(void)
- QSqlDriver * driver(void)
- bool exec(QString)
- bool exec_2(void)
- bool execBatch(QSqlQuery::BatchExecutionMode mode)
- QString executedQuery(void)
- void finish(void)
- bool first(void)
- bool isActive(void)
- bool isForwardOnly(void)
- bool isNull(int field)
- bool isSelect(void)
- bool isValid(void)
- bool last(void)

- QSqlError lastError(void)
- QVariant lastInsertId(void)
- QString lastQuery(void)
- bool next(void) # In RingQt use : bool movenext(void)
- bool nextResult(void)
- int numRowsAffected(void)
- QSql::NumericalPrecisionPolicy numericalPrecisionPolicy(void)
- bool prepare(QString)
- bool previous(void)
- QSqlRecord record(void)
- QSqlResult *result(void)
- bool seek(int index, bool relative)
- void setForwardOnly(bool forward)
- void setNumericalPrecisionPolicy(QSql::NumericalPrecisionPolicy precisionPolicy)
- int size(void)
- QVariant value(int index)

91.278 QSqlRecord Class

C++ Reference : http://doc.qt.io/qt-5/qsqlrecord.html

Parameters : void

- void append(QSqlField)
- void clear(void)
- void clearValues(void)
- bool contains(QString)
- int count(void)
- QSqlField field(int index)
- QString fieldName(int index)
- int indexOf(QString)
- void insert(int pos, QSqlField)
- bool isEmpty(void)
- bool isGenerated(QString)
- bool isNull(QString)
- void remove(int pos)
- void replace(int pos, QSqlField)
- void setGenerated(QString, bool generated)

- void setNull(int index)
- void setValue(int index, QVariant)
- QVariant value(int index)

91.279 QStackedBarSeries Class

C++ Reference : http://doc.qt.io/qt-5/qstackedbarseries.html

Parameters : QObject *

Parent Class : QAbstractBarSeries

- QAbstractSeries::SeriesType type(void)

91.280 QStackedWidget Class

C++ Reference : http://doc.qt.io/qt-5/qstackedwidget.html

Parameters : QWidget *

Parent Class : QFrame

- int addWidget(QWidget *widget)
- int count(void)
- int currentIndex(void)
- QWidget * currentWidget(void)
- int indexOf(QWidget *widget)
- int insertWidget(int index, QWidget *widget)
- void removeWidget(QWidget *widget)
- QWidget * widget(int index)
- void setCurrentIndex(int index)
- void setCurrentWidget(QWidget *widget)
- void setcurrentChangedEvent(const char *)
- void setwidgetRemovedEvent(const char *)
- const char *getcurrentChangedEvent(void)
- const char *getwidgetRemovedEvent(void)

91.281 QStandardPaths Class

C++ Reference : http://doc.qt.io/qt-5/qstandardpaths.html

Parameters : void

- QString displayName(QStandardPaths::StandardLocation type)
- QString findExecutable(QString executableName, QStringList paths))
- QString locate(QStandardPaths::StandardLocation type, QString fileName, QStandardPaths::LocateOptions options)
- QStringList locateAll(QStandardPaths::StandardLocation type, QString fileName, QStandardPaths::LocateOptions options)
- void setTestModeEnabled(bool testMode)
- QStringList standardLocations(QStandardPaths::StandardLocation type)
- QString writableLocation(QStandardPaths::StandardLocation type)

91.282 QStatusBar Class

C++ Reference : http://doc.qt.io/qt-5/qstatusbar.html

Parameters : QWidget *

Parent Class : QWidget

- void addPermanentWidget(QWidget * widget, int stretch)
- void addWidget(QWidget * widget, int stretch)
- QString currentMessage(void)
- int insertPermanentWidget(int index, QWidget * widget, int stretch)
- int insertWidget(int index, QWidget * widget, int stretch)
- bool isSizeGripEnabled(void)
- void removeWidget(QWidget *widget)
- void setSizeGripEnabled(bool)
- void clearMessage(void)
- void showMessage(QString , int timeout)

91.283 QString2 Class

Parameters : void

- QStringList split(QString sep, QString::SplitBehavior behavior , Qt::CaseSensitivity cs)
- QStringList split_2(QChar sep, QString::SplitBehavior behavior , Qt::CaseSensitivity cs)
- QStringList split_3(QRegExp rx, QString::SplitBehavior behavior)
- QStringList split_4(QRegularExpression re, QString::SplitBehavior behavior)

- QString append(QString str)
- QString append_2(QChar ch)
- QByteArray toUtf8(void)
- QByteArray toLatin1(void)
- QByteArray toLocal8Bit(void)
- QChar * unicode(void)
- QString number(ulong n, int base)
- int count(void)
- QString left(int n)
- QString mid(int position, int n)
- QString right(int n)
- int compare(QString other, Qt::CaseSensitivity cs)
- bool contains(QString str, Qt::CaseSensitivity cs)
- int indexOf(QString str,int from,Qt::CaseSensitivity cs)
- int lastIndexOf(QString str,int from,Qt::CaseSensitivity cs)
- QString insert(int position, QString str)
- bool isRightToLeft(void)
- QString repeated(int times)
- QString replace(int position, int n, QString after)
- QString replace_2(QString before, QString after, Qt::CaseSensitivity)
- bool startsWith(QString s, Qt::CaseSensitivity cs)
- bool endsWith(QString s, Qt::CaseSensitivity cs)
- QString toHtmlEscaped(void)
- void clear(void)
- bool isNull(void)
- void resize(int size)
- QString fill(QChar ch, int size)
- int localeAwareCompare(QString other)
- QString leftJustified(int width, QChar fill, bool truncate)
- QString rightJustified(int width, QChar fill, bool truncate)
- QString section_1(QChar sep, int start, int end, QString::SectionFlags flags)
- QString section_2(QString sep, int start, int end, QString::SectionFlags flags)
- QString section_3(QRegExp reg, int start, int end, QString::SectionFlags flags)
- QString section_4(QRegularExpression re, int start, int end, QString::SectionFlags flags)
- QString simplified(void)
- QString toCaseFolded(void)

- QString trimmed(void)
- void truncate(int position)

91.284 QStringList Class

C++ Reference : http://doc.qt.io/qt-5/qstringlist.html

Parameters : void

- QString join(QString)
- void sort(void)
- int removeDuplicates(void)
- QStringList filter(QString, Qt::CaseSensitivity)
- QStringList replaceInStrings(QString,QString, Qt::CaseSensitivity)
- void append(QString)
- QString at(int)
- QString back(void)
- void clear(void)
- bool contains(QString)
- int count(void)
- bool empty(void)
- bool endsWith(QString)
- QString first(void)
- QString front(void)
- int indexOf(QString, int)
- void insert(int, QString)
- bool isEmpty(void)
- QString last(void)
- int lastIndexOf(QString,int)
- int length(void)
- void move(int,int)
- void pop_back(void)
- void pop_front(void)
- void prepend(QString)
- void push_back(QString)
- void push_front(QString)
- int removeAll(QString)
- void removeAt(int)

- void removeFirst(void)
- void removeLast(void)
- bool removeOne(QString)
- void replace(int,QString)
- void reserve(int)
- int size(void)
- bool startsWith(QString)
- QString takeAt(int)
- QString takeFirst(void)
- QString takeLast(void)
- QString value(int)

91.285 QStringRef Class

C++ Reference : http://doc.qt.io/qt-5/qstringref.html

Parameters : void

- QStringRef appendTo(QString * string)
- QChar at(int position)
- void clear(void)
- int compare_3(QLatin1String other, Qt::CaseSensitivity cs)
- QChar * constData(void)
- bool contains(QString str, Qt::CaseSensitivity cs)
- bool contains_2(QChar ch, Qt::CaseSensitivity cs)
- bool contains_3(QStringRef str, Qt::CaseSensitivity cs)
- bool contains_4(QLatin1String str, Qt::CaseSensitivity cs)
- int count(void)
- int count_2(QString str, Qt::CaseSensitivity cs)
- int count_3(QChar ch, Qt::CaseSensitivity cs)
- int count_4(QStringRef str, Qt::CaseSensitivity cs)
- QChar * data(void)
- bool endsWith(QString str, Qt::CaseSensitivity cs)
- bool endsWith_2(QChar ch, Qt::CaseSensitivity cs)
- bool endsWith_3(QLatin1String str, Qt::CaseSensitivity cs)
- bool endsWith_4(QStringRef str, Qt::CaseSensitivity cs)
- int indexOf(QString str, int from, Qt::CaseSensitivity cs)
- int indexOf_2(QLatin1String str, int from, Qt::CaseSensitivity cs)

- int indexOf_3(QChar ch, int from, Qt::CaseSensitivity cs)
- int indexOf_4(QStringRef str, int from, Qt::CaseSensitivity cs)
- bool isEmpty(void)
- bool isNull(void)
- int lastIndexOf(QString str, int from, Qt::CaseSensitivity cs)
- int lastIndexOf_2(QChar ch, int from, Qt::CaseSensitivity cs)
- int lastIndexOf_3(QLatin1String str, int from, Qt::CaseSensitivity cs)
- int lastIndexOf_4(QStringRef str, int from, Qt::CaseSensitivity cs)
- int length(void)
- int localeAwareCompare(QString other)
- int localeAwareCompare_2(QStringRef other)
- int position(void)
- int size(void)
- bool startsWith(QString str, Qt::CaseSensitivity cs)
- bool startsWith_2(QLatin1String str, Qt::CaseSensitivity cs)
- bool startsWith_3(QStringRef str, Qt::CaseSensitivity cs)
- bool startsWith_4(QChar ch, Qt::CaseSensitivity cs)
- QString * string(void)
- QByteArray toLatin1(void)
- QByteArray toLocal8Bit(void)
- QString toString(void)
- QVector<uint> toUcs4(void)
- QByteArray toUtf8(void)
- QChar * unicode(void)
- int compare_4(QStringRef s1, QString s2, Qt::CaseSensitivity cs)
- int compare_5(QStringRef s1, QStringRef s2, Qt::CaseSensitivity cs)
- int compare_6(QStringRef s1, QLatin1String s2, Qt::CaseSensitivity cs)
- int localeAwareCompare_3(QStringRef s1, QString s2)
- int localeAwareCompare_4(QStringRef s1, QStringRef s2)

91.286 QStyle Class

C++ Reference : http://doc.qt.io/qt-5/qstyle.html

Parameters : void

Parent Class : QObject

- int combinedLayoutSpacing(QSizePolicy::ControlTypes controls1, QSizePolicy::ControlTypes controls2, Qt::Orientation orientation, QStyleOption *option, QWidget *widget)
- void drawComplexControl(QStyle::ComplexControl control, QStyleOptionComplex *option, QPainter *painter, QWidget *widget)
- void drawControl(QStyle::ControlElement element, QStyleOption *option, QPainter *painter, QWidget *widget)
- void drawItemPixmap(QPainter *painter, QRect rectangle, int alignment, QPixmap pixmap)
- void drawItemText(QPainter *painter, QRect rectangle, int alignment, QPalette palette, bool enabled, QString text, QPalette::ColorRole textRole)
- void drawPrimitive(QStyle::PrimitiveElement element, QStyleOption *option, QPainter *painter, QWidget *widget)
- QPixmap generatedIconPixmap(QIcon::Mode iconMode, QPixmap pixmap, QStyleOption *option)
- QStyle::SubControl hitTestComplexControl(QStyle::ComplexControl control, QStyleOptionComplex *option, QPoint position, QWidget *widget)
- QRect itemPixmapRect(QRect rectangle, int alignment, QPixmap pixmap)
- QRect itemTextRect(QFontMetrics metrics, QRect rectangle, int alignment, bool enabled, QString text)
- int layoutSpacing(QSizePolicy::ControlType control1, QSizePolicy::ControlType control2, Qt::Orientation orientation, QStyleOption *option, QWidget *widget)
- int pixelMetric(QStyle::PixelMetric metric)
- int pixelMetric_2(QStyle::PixelMetric metric, QStyleOption *option, QWidget *widget)
- void polish(QWidget *widget)
- void polish_2(QApplication *application)
- void polish_3(QPalette palette)
- QStyle * proxy(void)
- QSize sizeFromContents(QStyle::ContentsType type, QStyleOption *option, QSize contentsSize, QWidget *widget)
- QIcon standardIcon(QStyle::StandardPixmap standardIcon, QStyleOption *option, QWidget *widget)
- QPalette standardPalette(void)
- int styleHint(QStyle::StyleHint hint, QStyleOption *option, QWidget *widget, QStyleHintReturn *returnData)
- QRect subControlRect(QStyle::ComplexControl control, QStyleOptionComplex *option, QStyle::SubControl subControl, QWidget *widget)
- QRect subElementRect(QStyle::SubElement element, QStyleOption *option, QWidget *widget)
- void unpolish(QWidget *widget)
- void unpolish_2(QApplication *application)

- QRect alignedRect(Qt::LayoutDirection direction, Qt::Alignment alignment, QSize size, QRect rectangle)
- int sliderPositionFromValue(int min, int max, int logicalValue, int span, bool upsideDown)
- int sliderValueFromPosition(int min, int max, int position, int span, bool upsideDown)
- Qt::Alignment visualAlignment(Qt::LayoutDirection direction, Qt::Alignment alignment)
- QPoint visualPos(Qt::LayoutDirection direction, QRect boundingRectangle, QPoint logicalPosition)
- QRect visualRect(Qt::LayoutDirection direction, QRect boundingRectangle, QRect logicalRectangle)

91.287 QSurfaceFormat Class

C++ Reference : http://doc.qt.io/qt-5/qsurfaceformat.html

Parameters : void

- int alphaBufferSize(void)
- int blueBufferSize(void)
- int greenBufferSize(void)
- bool hasAlpha(void)
- int majorVersion(void)
- int minorVersion(void)
- QSurfaceFormat::FormatOptions options(void)
- QSurfaceFormat::OpenGLContextProfile profile(void)
- int redBufferSize(void)
- QSurfaceFormat::RenderableType renderableType(void)
- int samples(void)
- void setAlphaBufferSize(int size)
- void setBlueBufferSize(int size)
- void setGreenBufferSize(int size)
- void setMajorVersion(int major)
- void setMinorVersion(int minor)
- void setOption(QSurfaceFormat::FormatOption option, bool on)
- void setOptions(QSurfaceFormat::FormatOptions options)
- void setProfile(QSurfaceFormat::OpenGLContextProfile profile)
- void setRedBufferSize(int size)
- void setRenderableType(QSurfaceFormat::RenderableType type)
- void setSamples(int numSamples)
- void setStencilBufferSize(int size)
- void setStereo(bool enable)
- void setSwapBehavior(QSurfaceFormat::SwapBehavior behavior)

- void setSwapInterval(int interval)
- void setVersion(int major, int minor)
- int stencilBufferSize(void)
- bool stereo(void)
- QSurfaceFormat::SwapBehavior swapBehavior(void)
- int swapInterval(void)
- bool testOption(QSurfaceFormat::FormatOption option)
- QSurfaceFormat defaultFormat(void)
- void setDefaultFormat(QSurfaceFormat format)

91.288 QSystemTrayIcon Class

C++ Reference : http://doc.qt.io/qt-5/qsystemtrayicon.html

Parameters : void

- QMenu *contextMenu(void)
- QRect geometry(void)
- QIcon icon(void)
- bool isVisible(void)
- void setContextMenu(QMenu *menu)
- void setIcon(QIcon)
- void setToolTip(QString)
- QString toolTip(void)
- void hide(void)
- void setVisible(bool visible)
- void show(void)
- void showMessage(QString, QString, QSystemTrayIcon::MessageIcon, int millisecondsTimeoutHint)
- bool isSystemTrayAvailable(void)
- bool supportsMessages(void)

91.289 QTabBar Class

C++ Reference : http://doc.qt.io/qt-5/qtabbar.html

Parameters : QWidget *

Parent Class : QWidget

- int addTab(QString text)
- int addTab_2(QIcon icon, QString text)
- int count(void)

- int currentIndex(void)
- bool documentMode(void)
- bool drawBase(void)
- Qt::TextElideMode elideMode(void)
- bool expanding(void)
- QSize iconSize(void)
- int insertTab(int index, QString text)
- int insertTab_2(int index, QIcon icon, QString text)
- bool isMovable(void)
- bool isTabEnabled(int index)
- void moveTab(int from, int to)
- void removeTab(int index)
- QTabBar::SelectionBehavior selectionBehaviorOnRemove(void)
- void setDocumentMode(bool set)
- void setDrawBase(bool drawTheBase)
- void setElideMode(Qt::TextElideMode)
- void setExpanding(bool enabled)
- void setIconSize(QSize size)
- void setMovable(bool movable)
- void setSelectionBehaviorOnRemove(QTabBar::SelectionBehavior behavior)
- void setShape(QTabBar::Shape shape)
- void setTabButton(int index, QTabBar::ButtonPosition position, QWidget * widget)
- void setTabData(int index, QVariant data)
- void setTabEnabled(int index, bool enabled)
- void setTabIcon(int index, QIcon icon)
- void setTabText(int index, QString text)
- void setTabTextColor(int index, QColor color)
- void setTabToolTip(int index, QString tip)
- void setTabWhatsThis(int index, QString text)
- void setTabsClosable(bool closable)
- void setUsesScrollButtons(bool useButtons)
- QTabBar::Shape shape(void)
- int tabAt(QPoint position)
- QWidget * tabButton(int index, QTabBar::ButtonPosition position)
- QVariant tabData(int index)
- QIcon tabIcon(int index)

- QRect tabRect(int index)
- QString tabText(int index)
- QColor tabTextColor(int index)
- QString tabToolTip(int index)
- QString tabWhatsThis(int index)
- bool tabsClosable(void)
- bool usesScrollButtons(void)
- void setCurrentIndex(int index)
- void setcurrentChangedEvent(const char *)
- void settabCloseRequestedEvent(const char *)
- void settabMovedEvent(const char *)
- const char *getcurrentChangedEvent(void)
- const char *gettabCloseRequestedEvent(void)
- const char *gettabMovedEvent(void)

91.290 QTabWidget Class

C++ Reference : http://doc.qt.io/qt-5/qtabwidget.html

Parameters : QWidget *parent

Parent Class : QWidget

- QTabBar *tabBar(void)
- int addTab(QWidget *page, QString)
- void clear(void)
- QWidget *cornerWidget(Qt::Corner corner)
- int count(void)
- int currentIndex(void)
- QWidget *currentWidget(void)
- bool documentMode(void)
- int elideMode(void)
- QSize iconSize(void)
- int indexOf(QWidget *w)
- int insertTab(int index, QWidget *page,QString)
- bool isMovable(void)
- bool isTabEnabled(int index)
- void removeTab(int index)
- void setCornerWidget(QWidget *widget, Qt::Corner corner)

- void setDocumentMode(bool set)
- void setElideMode(Qt::TextElideMode)
- void setIconSize(QSize)
- void setMovable(bool movable)
- void setTabEnabled(int index, bool enable)
- void setTabIcon(int index, QIcon)
- void setTabText(int index,QString)
- void setTabToolTip(int index, QString)
- void setTabWhatsThis(int index, QString)
- void setTabsClosable(bool closeable)
- void setUsesScrollButtons(bool useButtons)
- QIcon tabIcon(int index)
- QString tabText(int index)
- QString tabToolTip(int index)
- QString tabWhatsThis(int index)
- bool tabsClosable(void)
- bool usesScrollButtons(void)
- QWidget *widget(int index)
- int heightForWidth(int width)
- QSize minimumSizeHint(void)
- QSize sizeHint(void)
- void setCurrentIndex(int index)
- void setCurrentWidget(QWidget *widget)
- void setcurrentChangedEvent(const char *)
- void settabCloseRequestedEvent(const char *)
- const char *getcurrentChangedEvent(void)
- const char *gettabCloseRequestedEvent(void)
- void geteventparameters(void)

91.291 QTableView Class

C++ Reference : http://doc.qt.io/qt-5/qtableview.html

Parameters : QWidget *parent

Parent Class : QAbstractItemView

- void clearSpans(void)
- int columnAt(int x)

- int columnSpan(int row, int column)
- int columnViewportPosition(int column)
- int columnWidth(int column)
- Qt::PenStyle gridStyle(void)
- QHeaderView *horizontalHeader(void)
- bool isColumnHidden(int column)
- bool isCornerButtonEnabled(void)
- bool isRowHidden(int row)
- bool isSortingEnabled(void)
- int rowAt(int y)
- int rowHeight(int row)
- int rowSpan(int row, int column)
- int rowViewportPosition(int row)
- void setColumnHidden(int column, bool hide)
- void setColumnWidth(int column, int width)
- void setCornerButtonEnabled(bool enable)
- void setGridStyle(Qt::PenStyle style)
- void setHorizontalHeader(QHeaderView *header)
- void setRowHeight(int row, int height)
- void setRowHidden(int row, bool hide)
- void setSortingEnabled(bool enable)
- void setSpan(int row, int column, int rowSpanCount, int columnSpanCount)
- void setVerticalHeader(QHeaderView *header)
- void setWordWrap(bool on)
- bool showGrid(void)
- void sortByColumn(int column, Qt::SortOrder order)
- QHeaderView *verticalHeader(void)
- bool wordWrap(void)
- void hideColumn(int column)
- void hideRow(int row)
- void resizeColumnToContents(int column)
- void resizeColumnsToContents(void)
- void resizeRowToContents(int row)
- void resizeRowsToContents(void)
- void selectColumn(int column)
- void selectRow(int row)

- void setShowGrid(bool show)
- void showColumn(int column)
- void showRow(int row)

91.292 QTableWidget Class

C++ Reference : http://doc.qt.io/qt-5/qtablewidget.html

Parameters : QWidget *parent

Parent Class : QTableView

- QWidget *cellWidget(int row, int column)
- void closePersistentEditor(QTableWidgetItem *item)
- int column(QTableWidgetItem *item)
- int columnCount(void)
- int currentColumn(void)
- QTableWidgetItem *currentItem(void)
- int currentRow(void)
- void editItem(QTableWidgetItem *item)
- QTableWidgetItem *horizontalHeaderItem(int column)
- QTableWidgetItem *item(int row, int column)
- QTableWidgetItem *itemAt(int ax, int ay)
- QTableWidgetItem *itemPrototype(void)
- void openPersistentEditor(QTableWidgetItem *item)
- void removeCellWidget(int row, int column)
- int row(const QTableWidgetItem *item)
- int rowCount(void)
- QList<QTableWidgetItem *> selectedItems(void)
- QList<QTableWidgetSelectionRange> selectedRanges(void)
- void setCellWidget(int row, int column, QWidget *widget)
- void setColumnCount(int columns)
- void setCurrentCell(int row, int column)
- void setCurrentItem(QTableWidgetItem * item)
- void setHorizontalHeaderItem(int column, QTableWidgetItem *item)
- void setHorizontalHeaderLabels(QStringList)
- void setItem(int row, int column, QTableWidgetItem *item)
- void setItemPrototype(QTableWidgetItem *item)
- void setRowCount(int rows)

- void setVerticalHeaderItem(int row, QTableWidgetItem *item)
- void sortItems(int column, Qt::SortOrder order)
- QTableWidgetItem *takeHorizontalHeaderItem(int column)
- QTableWidgetItem *takeItem(int row, int column)
- QTableWidgetItem *takeVerticalHeaderItem(int row)
- QTableWidgetItem *verticalHeaderItem(int row)
- int visualColumn(int logicalColumn)
- QRect visualItemRect(QTableWidgetItem *)
- int visualRow(int logicalRow)
- void clear(void)
- void clearContents(void)
- void insertColumn(int column)
- void insertRow(int row)
- void removeColumn(int column)
- void removeRow(int row)
- void scrollToItem(QTableWidgetItem *item, QAbstractItemView::ScrollHint hint)
- void setcellActivatedEvent(const char *)
- void setcellChangedEvent(const char *)
- void setcellClickedEvent(const char *)
- void setcellDoubleClickedEvent(const char *)
- void setcellEnteredEvent(const char *)
- void setcellPressedEvent(const char *)
- void setcurrentCellChangedEvent(const char *)
- void setcurrentItemChangedEvent(const char *)
- void setitemActivatedEvent(const char *)
- void setitemChangedEvent(const char *)
- void setitemClickedEvent(const char *)
- void setitemDoubleClickedEvent(const char *)
- void setitemEnteredEvent(const char *)
- void setitemPressedEvent(const char *)
- void setitemSelectionChangedEvent(const char *)
- const char *getcellActivatedEvent(void)
- const char *getcellChangedEvent(void)
- const char *getcellClickedEvent(void)
- const char *getcellDoubleClickedEvent(void)
- const char *getcellEnteredEvent(void)

- const char *getcellPressedEvent(void)
- const char *getcurrentCellChangedEvent(void)
- const char *getcurrentItemChangedEvent(void)
- const char *getitemActivatedEvent(void)
- const char *getitemChangedEvent(void)
- const char *getitemClickedEvent(void)
- const char *getitemDoubleClickedEvent(void)
- const char *getitemEnteredEvent(void)
- const char *getitemPressedEvent(void)
- const char *getitemSelectionChangedEvent(void)

91.293 QTableWidgetItem Class

C++ Reference : http://doc.qt.io/qt-5/qtablewidgetitem.html

Parameters : QString

- QBrush background(void)
- int checkState(void)
- QTableWidgetItem *clone(void)
- int column(void)
- QVariant data(int role)
- int flags(void)
- QFont font(void)
- QBrush foreground(void)
- QIcon icon(void)
- bool isSelected(void)
- void read(QDataStream)
- int row(void)
- void setBackground(QBrush)
- void setCheckState(Qt::CheckState state)
- void setData(int role, QVariant)
- void setFlags(Qt::ItemFlag flags)
- void setFont(QFont)
- void setForeground(QBrush)
- void setIcon(QIcon)
- void setSelected(bool select)
- void setSizeHint(QSize)

- void setStatusTip(QString)
- void setText(QString)
- void setTextAlignment(int alignment)
- void setToolTip(QString)
- void setWhatsThis(QString)
- QSize sizeHint(void)
- QString statusTip(void)
- QTableWidget *tableWidget(void)
- QString text(void)
- int textAlignment(void)
- QString toolTip(void)
- int type(void)
- QString whatsThis(void)
- void write(QDataStream)

91.294 QTcpServer Class

C++ Reference : http://doc.qt.io/qt-5/qtcpserver.html

Parameters : QWidget *

- void close(void)
- QString errorString(void)
- bool hasPendingConnections(void)
- bool isListening(void)
- bool listen(QHostAddress, int port)
- int maxPendingConnections(void)
- QTcpSocket *nextPendingConnection(void)
- void pauseAccepting(void)
- QNetworkProxy proxy(void)
- void resumeAccepting(void)
- QHostAddress serverAddress(void)
- int serverError(void)
- int serverPort(void)
- void setMaxPendingConnections(int numConnections)
- void setProxy(QNetworkProxy)
- bool setSocketDescriptor(qintptr socketDescriptor)
- int *socketDescriptor(void)

- bool waitForNewConnection(int msec, bool *timedOut)
- void setacceptErrorEvent(const char *)
- void setnewConnectionEvent(const char *)
- const char *getacceptErrorEvent(void)
- const char *getnewConnectionEvent(void)

91.295 QTcpSocket Class

C++ Reference : http://doc.qt.io/qt-5/qtcpsocket.html

Parameters : QObject *

Parent Class : QAbstractSocket

- void setconnectedEvent(const char *)
- void setdisconnectedEvent(const char *)
- void seterrorEvent(const char *)
- void sethostFoundEvent(const char *)
- void setproxyAuthenticationRequiredEvent(const char *)
- void setstateChangedEvent(const char *)
- void setaboutToCloseEvent(const char *)
- void setbytesWrittenEvent(const char *)
- void setreadChannelFinishedEvent(const char *)
- void setreadyReadEvent(const char *)
- const char *getconnectedEvent(void)
- const char *getdisconnectedEvent(void)
- const char *geterrorEvent(void)
- const char *gethostFoundEvent(void)
- const char *getproxyAuthenticationRequiredEvent(void)
- const char *getstateChangedEvent(void)
- const char *getaboutToCloseEvent(void)
- const char *getbytesWrittenEvent(void)
- const char *getreadChannelFinishedEvent(void)
- const char *getreadyReadEvent(void)

91.296 QTechnique Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qtechnique.html

Parameters : Qt3DCore::QNode *

- void addFilterKey(Qt3DRender::QFilterKey *filterKey)
- void addParameter(Qt3DRender::QParameter *parameter)
- void addRenderPass(Qt3DRender::QRenderPass *pass)
- QVector<Qt3DRender::QFilterKey *> filterKeys(void)
- Qt3DRender::QGraphicsApiFilter * graphicsApiFilter(void)
- QVector<Qt3DRender::QParameter *> parameters(void)
- void removeFilterKey(Qt3DRender::QFilterKey *filterKey)
- void removeParameter(Qt3DRender::QParameter *parameter)
- void removeRenderPass(Qt3DRender::QRenderPass *pass)
- QVector<Qt3DRender::QRenderPass *> renderPasses(void)

91.297 QTest Class

C++ Reference : http://doc.qt.io/qt-5/qtest.html

- void qsleep(int)

91.298 QText2DEntity Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qtext2dentity.html

Parameters : Qt3DCore::QNode *

Parent Class : QEntity

- QColor color(void)
- QFont font(void)
- float height(void)
- void setColor(QColor color)
- void setFont(QFont font)
- void setHeight(float height)
- void setText(QString text)
- void setWidth(float width)
- QString text(void)
- float width(void)

91.299 QTextBlock Class

C++ Reference : http://doc.qt.io/qt-5/qtextblock.html

Parameters : void

- int blockFormatIndex(void)
- int blockNumber(void)
- QTextCharFormat charFormat(void)
- int charFormatIndex(void)
- void clearLayout(void)
- bool contains(int position)
- QTextDocument *document(void)
- bool isValid(void)
- bool isVisible(void)
- QTextLayout * layout(void)
- int length(void)
- int lineCount(void)
- QTextBlock next(void) # In RingQt use : QTextBlock nextblock(void)
- int position(void)
- QTextBlock previous(void)
- int revision(void)
- void setLineCount(int count)
- void setRevision(int rev)
- void setUserData(QTextBlockUserData * data)
- void setUserState(int state)
- void setVisible(bool visible)
- QString text(void)
- int textDirection(void)
- QTextList * textList(void)
- QTextBlockUserData * userData(void)
- int userState(void)

91.300 QTextBrowser Class

C++ Reference : http://doc.qt.io/qt-5/qtextbrowser.html

Parameters : QWidget *

Parent Class : QTextEdit

- int backwardHistoryCount(void)
- void clearHistory(void)
- int forwardHistoryCount(void)
- QString historyTitle(int i)
- QUrl historyUrl(int i)
- bool isBackwardAvailable(void)
- bool isForwardAvailable(void)
- bool openExternalLinks(void)
- bool openLinks(void)
- QStringList searchPaths(void)
- void setOpenExternalLinks(bool open)
- void setOpenLinks(bool open)
- void setSearchPaths(QStringList paths)
- QUrl source(void)
- void setanchorClickedEvent(const char *)
- void setbackwardAvailableEvent(const char *)
- void setforwardAvailableEvent(const char *)
- void sethighlightedEvent(const char *)
- void sethistoryChangedEvent(const char *)
- void setsourceChangedEvent(const char *)
- const char *getanchorClickedEvent(void)
- const char *getbackwardAvailableEvent(void)
- const char *getforwardAvailableEvent(void)
- const char *gethighlightedEvent(void)
- const char *gethistoryChangedEvent(void)
- const char *getsourceChangedEvent(void)

91.301 QTextCharFormat Class

C++ Reference : http://doc.qt.io/qt-5/qtextcharformat.html

Parameters : void

- QString anchorHref(void)
- QStringList anchorNames(void)
- QFont font(void)
- QFont::Capitalization fontCapitalization(void)
- QString fontFamily(void)
- bool fontFixedPitch(void)
- QFont::HintingPreference fontHintingPreference(void)
- bool fontItalic(void)
- bool fontKerning(void)
- qreal fontLetterSpacing(void)
- QFont::SpacingType fontLetterSpacingType(void)
- bool fontOverline(void)
- qreal fontPointSize(void)
- int fontStretch(void)
- bool fontStrikeOut(void)
- QFont::StyleHint fontStyleHint(void)
- QFont::StyleStrategy fontStyleStrategy(void)
- bool fontUnderline(void)
- int fontWeight(void)
- qreal fontWordSpacing(void)
- bool isAnchor(void)
- bool isValid(void)
- void setAnchor(bool anchor)
- void setAnchorHref(QString value)
- void setAnchorNames(QStringList names)
- void setFontCapitalization(QFont::Capitalization capitalization)
- void setFontFamily(QString family)
- void setFontFixedPitch(bool fixedPitch)
- void setFontHintingPreference(QFont::HintingPreference hintingPreference)
- void setFontItalic(bool italic)
- void setFontKerning(bool enable)
- void setFontLetterSpacing(qreal spacing)

- void setFontLetterSpacingType(QFont::SpacingType letterSpacingType)
- void setFontOverline(bool overline)
- void setFontPointSize(qreal size)
- void setFontStretch(int factor)
- void setFontStrikeOut(bool strikeOut)
- void setFontStyleHint(QFont::StyleHint hint, QFont::StyleStrategy strategy)
- void setFontStyleStrategy(QFont::StyleStrategy strategy)
- void setFontUnderline(bool underline)
- void setFontWeight(int weight)
- void setFontWordSpacing(qreal spacing)
- void setTextOutline(QPen pen)
- void setToolTip(QString text)
- void setUnderlineColor(QColor color)
- void setUnderlineStyle(QTextCharFormat::UnderlineStyle style)
- void setVerticalAlignment(QTextCharFormat::VerticalAlignment alignment)
- QPen textOutline(void)
- QString toolTip(void)
- QColor underlineColor(void)
- QTextCharFormat::UnderlineStyle underlineStyle(void)
- QTextCharFormat::VerticalAlignment verticalAlignment(void)

91.302 QTextCodec Class

C++ Reference : http://doc.qt.io/qt-5/qtextcodec.html

- QTextCodec *codecForName(const char *name)
- void setCodecForLocale(QTextCodec *c)

91.303 QTextCursor Class

C++ Reference : http://doc.qt.io/qt-5/qtextcursor.html

Parameters : void

- int anchor(void)
- bool atBlockEnd(void)
- bool atBlockStart(void)
- bool atEnd(void)
- bool atStart(void)
- void beginEditBlock(void)

- QTextBlock block(void)
- QTextCharFormat blockCharFormat(void)
- QTextBlockFormat blockFormat(void)
- int blockNumber(void)
- QTextCharFormat charFormat(void)
- void clearSelection(void)
- int columnNumber(void)
- QTextList *createList(QTextListFormat)
- QTextFrame *currentFrame(void)
- QTextList *currentList(void)
- QTextTable *currentTable(void)
- void deleteChar(void)
- void deletePreviousChar(void)
- QTextDocument *document(void)
- void endEditBlock(void)
- bool hasComplexSelection(void)
- bool hasSelection(void)
- void insertBlock(void)
- void insertFragment(QTextDocumentFragment)
- QTextFrame *insertFrame(QTextFrameFormat)
- void insertHtml(QString)
- void insertImage(QTextImageFormat)
- QTextList *insertList(QTextListFormat)
- QTextTable * insertTable(int rows, int columns, QTextTableFormat)
- void insertText(QString)
- void insertText_2(QString,QTextCharFormat)
- bool isCopyOf(QTextCursor)
- bool isNull(void)
- void joinPreviousEditBlock(void)
- bool keepPositionOnInsert(void)
- void mergeBlockCharFormat(QTextCharFormat)
- void mergeBlockFormat(QTextBlockFormat)
- void mergeCharFormat(QTextCharFormat)
- bool movePosition(QTextCursor::MoveOperation operation, QTextCursor::MoveMode mode, int n)
- int position(void)
- int positionInBlock(void)

- void removeSelectedText(void)
- void select(QTextCursor::SelectionType selection)
- void selectedTableCells(int *firstRow, int *numRows, int *firstColumn, int *numColumns)
- QString selectedText(void)
- QTextDocumentFragment selection(void)
- int selectionEnd(void)
- int selectionStart(void)
- void setBlockCharFormat(QTextCharFormat)
- void setBlockFormat(QTextBlockFormat)
- void setCharFormat(QTextCharFormat)
- void setKeepPositionOnInsert(bool b)
- void setPosition(int pos, QTextCursor::MoveMode m)
- void setVerticalMovementX(int x)
- void setVisualNavigation(bool b)
- int verticalMovementX(void)
- bool visualNavigation(void)

91.304 QTextDocument Class

C++ Reference : http://doc.qt.io/qt-5/qtextdocument.html

Parameters : void

Parent Class : QObject

- void addResource(int type,QUrl name, QVariant resource)
- void adjustSize(void)
- QVector<QTextFormat> allFormats(void)
- int availableRedoSteps(void)
- int availableUndoSteps(void)
- QTextBlock begin(void)
- int blockCount(void)
- QChar characterAt(int pos)
- int characterCount(void)
- void clearUndoRedoStacks(QTextDocument::Stacks stacksToClear)
- QTextDocument *clone(QObject *parent)
- int defaultCursorMoveStyle(void)
- QFont defaultFont(void)
- QString defaultStyleSheet(void)

- QTextOption defaultTextOption(void)
- QAbstractTextDocumentLayout *documentLayout(void)
- double documentMargin(void)
- void drawContents(QPainter *p, QRectF rect)
- QTextBlock end(void) # In RingQt use : QTextBlock enddoc(void)
- QTextCursor find(QString subString, QTextCursor cursor, QTextDocument::FindFlag options)
- QTextBlock findBlock(int pos)
- QTextBlock findBlockByLineNumber(int lineNumber)
- QTextBlock findBlockByNumber(int blockNumber)
- QTextBlock firstBlock(void)
- double idealWidth(void)
- double indentWidth(void)
- bool isEmpty(void)
- bool isModified(void)
- bool isRedoAvailable(void)
- bool isUndoAvailable(void)
- bool isUndoRedoEnabled(void)
- QTextBlock lastBlock(void)
- int lineCount(void)
- void markContentsDirty(int position, int length)
- int maximumBlockCount(void)
- QString metaInformation(QTextDocument::MetaInformation info)
- QTextObject *object(int objectIndex)
- QTextObject *objectForFormat(QTextFormat f)
- int pageCount(void)
- QSizeF pageSize(void)
- void print(QPrinter *printer)
- void redo(QTextCursor *cursor)
- QVariant resource(int type, QUrl name)
- int revision(void)
- QTextFrame *rootFrame(void)
- void setDefaultCursorMoveStyle(Qt::CursorMoveStyle style)
- void setDefaultFont(QFont font)
- void setDefaultStyleSheet(QString sheet)
- void setDefaultTextOption(QTextOption option)
- void setDocumentLayout(QAbstractTextDocumentLayout * layout)

- void setDocumentMargin(double margin)
- void setHtml(QString html)
- void setIndentWidth(double width)
- void setMaximumBlockCount(int maximum)
- void setMetaInformation(QTextDocument::MetaInformation info, QString string)
- void setPageSize(QSizeF size)
- void setPlainText(QString text)
- void setTextWidth(double width)
- void setUndoRedoEnabled(bool enable)
- void setUseDesignMetrics(bool b)
- QSizeF size(void)
- qreal textWidth(void)
- QString toHtml(QByteArray encoding)
- QString toPlainText(void)
- void undo(QTextCursor *cursor)
- bool useDesignMetrics(void)
- void setModified(bool m)

91.305 QTextEdit Class

C++ Reference : http://doc.qt.io/qt-5/qtextedit.html

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- void setTabStopDistance(qreal width)
- qreal tabStopDistance(void)
- bool acceptRichText(void)
- int alignment(void)
- QString anchorAt(QPoint)
- bool canPaste(void)
- QTextCharFormat currentCharFormat(void)
- QFont currentFont(void)
- QTextCursor cursorForPosition(QPoint)
- QRect cursorRect(void)
- int cursorWidth(void)
- QTextDocument *document(void)
- QString documentTitle(void)

- void ensureCursorVisible(void)
- bool find(QString, QTextDocument::FindFlag)
- QString fontFamily(void)
- bool fontItalic(void)
- double fontPointSize(void)
- bool fontUnderline(void)
- int fontWeight(void)
- bool isReadOnly(void)
- bool isUndoRedoEnabled(void)
- int lineWrapColumnOrWidth(void)
- QVariant loadResource(int, QUrl)
- void mergeCurrentCharFormat(QTextCharFormat)
- void moveCursor(QTextCursor::MoveOperation operation, QTextCursor::MoveMode mode)
- bool overwriteMode(void)
- void print(QPrinter * printer)
- void setAcceptRichText(bool accept)
- void setCurrentCharFormat(QTextCharFormat)
- void setCursorWidth(int width)
- void setDocument(QTextDocument *document)
- void setDocumentTitle(QString)
- void setLineWrapColumnOrWidth(int w)
- void setLineWrapMode(QTextEdit::LineWrapMode)
- void setOverwriteMode(bool overwrite)
- void setReadOnly(bool)
- void setTabChangesFocus(bool)
- void setTextCursor(QTextCursor)
- void setTextInteractionFlags(Qt::TextInteractionFlag flags)
- void setUndoRedoEnabled(bool enable)
- void setWordWrapMode(QTextOption::WrapMode policy)
- bool tabChangesFocus(void)
- QColor textBackgroundColor(void)
- QColor textColor(void)
- QTextCursor textCursor(void)
- int textInteractionFlags(void)
- QString toHtml(void)
- QString toPlainText(void)

- int wordWrapMode(void)
- void append(QString)
- void clear(void)
- void copy(void)
- void cut(void)
- void insertHtml(QString)
- void insertPlainText(QString)
- void paste(void)
- void redo(void)
- void scrollToAnchor(QString)
- void selectAll(void)
- void setAlignment(Qt::AlignmentFlag a)
- void setCurrentFont(QFont)
- void setFontFamily(QString)
- void setFontItalic(bool italic)
- void setFontPointSize(double s)
- void setFontUnderline(bool underline)
- void setFontWeight(int weight)
- void setHtml(QString)
- void setPlainText(QString)
- void setText(QString)
- void setTextBackgroundColor(QColor)
- void setTextColor(QColor)
- void undo(void)
- void zoomIn(int range)
- void zoomOut(int range)
- void setcopyAvailableEvent(const char *)
- void setcurrentCharFormatChangedEvent(const char *)
- void setcursorPositionChangedEvent(const char *)
- void setredoAvailableEvent(const char *)
- void setselectionChangedEvent(const char *)
- void settextChangedEvent(const char *)
- void setundoAvailableEvent(const char *)
- const char *getcopyAvailableEvent(void)
- const char *getcurrentCharFormatChangedEvent(void)
- const char *getcursorPositionChangedEvent(void)

- const char *getredoAvailableEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettextChangedEvent(void)
- const char *getundoAvailableEvent(void)
- void cyanline(void)
- void setactivelinecolor(QColor)

91.306 QTextOption Class

C++ Reference : http://doc.qt.io/qt-5/qtextoption.html

Parameters : Qt::Alignment

- Qt::Alignment alignment(void)
- QTextOption::Flags flags(void)
- void setAlignment(Qt::Alignment alignment)
- void setFlags(QTextOption::Flags flags)
- void setTextDirection(Qt::LayoutDirection direction)
- void setUseDesignMetrics(bool enable)
- void setWrapMode(QTextOption::WrapMode mode)
- qreal tabStopDistance(void)
- Qt::LayoutDirection textDirection(void)
- bool useDesignMetrics(void)
- QTextOption::WrapMode wrapMode(void)

91.307 QTextStream Class

C++ Reference : http://doc.qt.io/qt-5/qtextstream.html

Parameters : void

- bool atEnd(void)
- bool autoDetectUnicode(void)
- QTextCodec * codec(void)
- QIODevice * device(void)
- QTextStream::FieldAlignment fieldAlignment(void)
- int fieldWidth(void)
- void flush(void)
- bool generateByteOrderMark(void)
- int integerBase(void)
- QLocale locale(void)

- QTextStream::NumberFlags numberFlags(void)
- QChar padChar(void)
- qint64 pos(void)
- QString read(qint64 maxlen)
- QString readAll(void)
- QString readLine(qint64 maxlen)
- QTextStream::RealNumberNotation realNumberNotation(void)
- int realNumberPrecision(void)
- void reset(void)
- void resetStatus(void)
- bool seek(qint64 pos)
- void setAutoDetectUnicode(bool enabled)
- void setCodec(QTextCodec * codec)
- void setCodec_2(char * codecName)
- void setDevice(QIODevice * device)
- void setFieldAlignment(QTextStream::FieldAlignment mode)
- void setFieldWidth(int width)
- void setGenerateByteOrderMark(bool generate)
- void setIntegerBase(int base)
- void setLocale(QLocale locale)
- void setNumberFlags(QTextStream::NumberFlags flags)
- void setPadChar(QChar ch)
- void setRealNumberNotation(QTextStream::RealNumberNotation notation)
- void setRealNumberPrecision(int precision)
- void setStatus(QTextStream::Status status)
- void setString(QString * string, QIODevice::OpenMode openMode)
- void skipWhiteSpace(void)
- QTextStream::Status status(void)
- QString * string(void)

91.308 QTextStream2 Class

Parameters : QIODevice * device

Parent Class : QTextStream

91.309 QTextStream3 Class

Parameters : FILE * fileHandle, QIODevice::OpenMode

Parent Class : QTextStream

91.310 QTextStream4 Class

Parameters : QString *, QIODevice::OpenMode

Parent Class : QTextStream

91.311 QTextStream5 Class

Parameters : QByteArray *, QIODevice::OpenMode

Parent Class : QTextStream

91.312 QTextToSpeech Class

C++ Reference : http://doc.qt.io/qt-5/qtexttospeech.html

Parameters : QObject *

Parent Class : QObject

- QVector<QLocale> availableLocales(void)
- QVector<QVoice> availableVoices(void)
- QLocale locale(void)
- double pitch(void)
- double rate(void)
- QTextToSpeech::State state(void)
- QVoice voice(void)
- double volume(void)
- void pause(void)
- void resume(void)
- void say(QString text)
- void setLocale(QLocale locale)

- void setPitch(double pitch)
- void setRate(double rate)
- void setVoice(QVoice voice)
- void setVolume(double volume)
- void stop(void)
- QStringList availableEngines(void)
- void setlocaleChangedEvent(const char *)
- void setpitchChangedEvent(const char *)
- void setrateChangedEvent(const char *)
- void setstateChangedEvent(const char *)
- void setvoiceChangedEvent(const char *)
- void setvolumeChangedEvent(const char *)
- const char *getlocaleChangedEvent(void)
- const char *getpitchChangedEvent(void)
- const char *getrateChangedEvent(void)
- const char *getstateChangedEvent(void)
- const char *getvoiceChangedEvent(void)
- const char *getvolumeChangedEvent(void)

91.313 QTextureLoader Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qtextureloader.html

Parameters : Qt3DCore::QNode *

- bool isMirrored(void)
- QUrl source(void)
- void setMirrored(bool mirrored)
- void setSource(QUrl source)

91.314 QTextureMaterial Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qtexturematerial.html

Parameters : Qt3DCore::QNode *

- bool isAlphaBlendingEnabled(void)
- Qt3DRender::QAbstractTexture *texture(void)
- QVector2D textureOffset(void)
- QMatrix3x3 textureTransform(void)
- void setAlphaBlendingEnabled(bool enabled)

- void setTexture(Qt3DRender::QAbstractTexture *texture)
- void setTextureOffset(QVector2D textureOffset)
- void setTextureTransform(QMatrix3x3 matrix)

91.315 QThread Class

C++ Reference : http://doc.qt.io/qt-5/qthread.html

Parameters : QObject *

Parent Class : QObject

- QAbstractEventDispatcher *eventDispatcher(void)
- void exit(int returnCode) # In RingQt use : void exitfromthread(int returnCode)
- bool isFinished(void)
- bool isInterruptionRequested(void)
- bool isRunning(void)
- QThread::Priority priority(void)
- void requestInterruption(void)
- void setEventDispatcher(QAbstractEventDispatcher *eventDispatcher)
- void setPriority(QThread::Priority priority)
- void setStackSize(uint stackSize)
- uint stackSize(void)
- bool wait(unsigned long time)
- void quit(void)
- void start(QThread::Priority priority)
- void terminate(void)
- QThread *currentThread(void)
- Qt::HANDLE currentThreadId(void)
- int idealThreadCount(void)
- void msleep(unsigned long msecs)
- void sleep(unsigned long secs)
- void usleep(unsigned long usecs)
- void yieldCurrentThread(void)
- void setStartedEvent(const char *)
- void setFinishedEvent(const char *)
- const char *getStartedEvent(void)
- const char *getFinishedEvent(void)

91.316 QThreadPool Class

C++ Reference : http://doc.qt.io/qt-5/qthreadpool.html

Parameters : void

Parent Class : QObject

- int activeThreadCount(void)
- void clear(void)
- int expiryTimeout(void)
- int maxThreadCount(void)
- void releaseThread(void)
- void reserveThread(void)
- void setExpiryTimeout(int expiryTimeout)
- void setMaxThreadCount(int maxThreadCount)
- void start(QRunnable * runnable, int priority)
- bool tryStart(QRunnable * runnable)
- bool waitForDone(int msecs)
- QThreadPool *globalInstance(void)

91.317 QTime Class

C++ Reference : http://doc.qt.io/qt-5/qtime.html

Parameters : void

- QTime addMSecs(int ms)
- QTime addSecs(int s)
- int hour(void)
- bool isNull(void)
- bool isValid(void)
- int minute(void)
- int msec(void)
- int msecsSinceStartOfDay(void)
- int msecsTo(QTime)
- int second(void)
- int secsTo(QTime)
- bool setHMS(int h, int m, int s, int ms)
- QString toString(QString)
- QTime currentTime(void)
- QTime fromMSecsSinceStartOfDay(int msecs)

- QTime fromString(QString,QString)

91.318 QTimer Class

C++ Reference : http://doc.qt.io/qt-5/qtimer.html

Parameters : QObject *parent

- int interval(void)
- bool isActive(void)
- bool isSingleShot(void)
- void setInterval(int msec)
- void setSingleShot(bool singleShot)
- int timerId(void)
- void start(void)
- void stop(void)
- void settimeoutEvent(const char *)
- const char *gettimeoutEvent(void)

91.319 QToolBar Class

C++ Reference : http://doc.qt.io/qt-5/qtoolbar.html

Parameters : QWidget *

Parent Class : QWidget

- QAction *actionAt(int x, int y)
- QAction *addAction(QString)
- QAction *addSeparator(void)
- QAction *addWidget(QWidget *widget)
- int allowedAreas(void)
- void clear(void)
- QSize iconSize(void)
- QAction *insertSeparator(QAction *before)
- QAction *insertWidget(QAction *before, QWidget *widget)
- bool isAreaAllowed(Qt::ToolBarArea area)
- bool isFloatable(void)
- bool isFloating(void)
- bool isMovable(void)
- int orientation(void)
- void setAllowedAreas(Qt::ToolBarArea areas)

- void setFloatable(bool floatable)
- void setMovable(bool movable)
- void setOrientation(Qt::Orientation orientation)
- QAction *toggleViewAction(void)
- int toolButtonStyle(void)
- QWidget *widgetForAction(QAction *action)
- void setIconSize(QSize)
- void setToolButtonStyle(Qt::ToolButtonStyle toolButtonStyle)

91.320 QToolButton Class

C++ Reference : http://doc.qt.io/qt-5/qtoolbutton.html

Parameters : QWidget *

Parent Class : QAbstractButton

- Qt::ArrowType arrowType(void)
- bool autoRaise(void)
- QAction * defaultAction(void)
- QMenu * menu(void)
- QToolButton::ToolButtonPopupMode popupMode(void)
- void setArrowType(Qt::ArrowType type)
- void setAutoRaise(bool enable)
- void setMenu(QMenu * menu)
- void setPopupMode(QToolButton::ToolButtonPopupMode mode)
- Qt::ToolButtonStyle toolButtonStyle(void)
- void setDefaultAction(QAction * action)
- void setToolButtonStyle(Qt::ToolButtonStyle style)
- void showMenu(void)
- void settriggeredEvent(const char *)
- const char *gettriggeredEvent(void)
- void setClickEvent(const char *)
- const char *getClickEvent(void)

91.321 QTorusMesh Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qtorusmesh.html

Parameters : Qt3DCore::QNode *

- float minorRadius(void)
- float radius(void)
- int rings(void)
- int slices(void)
- void setMinorRadius(float minorRadius)
- void setRadius(float radius)
- void setRings(int rings)
- void setSlices(int slices)

91.322 QTransform Class

C++ Reference : http://doc.qt.io/qt-5/qt3dcore-qtransform.html

Parameters : Qt3DCore::QNode *

- QMatrix4x4 matrix(void)
- QQuaternion rotation(void)
- float rotationX(void)
- float rotationY(void)
- float rotationZ(void)
- float scale(void)
- QVector3D scale3D(void)
- QVector3D translation(void)
- void setMatrix(QMatrix4x4 matrix)
- void setRotation(QQuaternion rotation)
- void setRotationX(float rotationX)
- void setRotationY(float rotationY)
- void setRotationZ(float rotationZ)
- void setScale(float scale)
- void setScale3D(QVector3D scale)
- void setTranslation(QVector3D translation)
- QQuaternion fromAxes(QVector3D xAxis, QVector3D yAxis, QVector3D zAxis)
- QQuaternion fromAxesAndAngles(QVector3D axis1, float angle1, QVector3D axis2, float angle2)
- QQuaternion fromAxesAndAngles_2(QVector3D axis1, float angle1, QVector3D axis2, float angle2, QVector3D axis3, float angle3)

- QQuaternion fromAxisAndAngle(QVector3D axis, float angle)
- QQuaternion fromAxisAndAngle_2(float x, float y, float z, float angle)
- QQuaternion fromEulerAngles(QVector3D eulerAngles)
- QQuaternion fromEulerAngles_2(float pitch, float yaw, float roll)
- QMatrix4x4 rotateAround(QVector3D point, float angle, QVector3D axis)
- QMatrix4x4 rotateFromAxes(QVector3D xAxis, QVector3D yAxis, QVector3D zAxis)

91.323 QTreeView Class

C++ Reference : http://doc.qt.io/qt-5/qtreeview.html

Parameters : QWidget *

Parent Class : QAbstractItemView

- bool allColumnsShowFocus(void)
- int autoExpandDelay(void)
- int columnAt(int x)
- int columnViewportPosition(int column)
- int columnWidth(int column)
- bool expandsOnDoubleClick(void)
- QHeaderView *header(void)
- int indentation(void)
- QModelIndex indexAbove(QModelIndex)
- QModelIndex indexBelow(QModelIndex)
- bool isAnimated(void)
- bool isColumnHidden(int column)
- bool isExpanded(QModelIndex)
- bool isFirstColumnSpanned(int row, QModelIndex)
- bool isHeaderHidden(void)
- bool isRowHidden(int row,QModelIndex)
- bool isSortingEnabled(void)
- bool itemsExpandable(void)
- bool rootIsDecorated(void)
- void setAllColumnsShowFocus(bool enable)
- void setAnimated(bool enable)
- void setAutoExpandDelay(int delay)
- void setColumnHidden(int column, bool hide)
- void setColumnWidth(int column, int width)

- void setExpanded(QModelIndex, bool expanded)
- void setExpandsOnDoubleClick(bool enable)
- void setFirstColumnSpanned(int row, QModelIndex, bool span)
- void setHeader(QHeaderView * header)
- void setHeaderHidden(bool hide)
- void setIndentation(int i)
- void setItemsExpandable(bool enable)
- void setRootIsDecorated(bool show)
- void setRowHidden(int row,QModelIndex, bool hide)
- void setSortingEnabled(bool enable)
- void setUniformRowHeights(bool uniform)
- void setWordWrap(bool on)
- void sortByColumn(int column,Qt::SortOrder order)
- bool uniformRowHeights(void)
- bool wordWrap(void)
- void dataChanged(QModelIndex,QModelIndex)
- QModelIndex indexAt(QPoint)
- void keyboardSearch(QString)
- void reset(void)
- void scrollTo(QModelIndex, QAbstractItemView::ScrollHint)
- void selectAll(void)
- void setModel(QAbstractItemModel *model)
- void setRootIndex(QModelIndex)
- void setSelectionModel(QItemSelectionModel *selectionModel)
- QRect visualRect(QModelIndex)
- void collapse(QModelIndex)
- void collapseAll(void)
- void expand(QModelIndex)
- void expandAll(void)
- void expandToDepth(int depth)
- void hideColumn(int column)
- void resizeColumnToContents(int column)
- void showColumn(int column)
- void setcollapsedEvent(const char *)
- void setexpandedEvent(const char *)
- void setactivatedEvent(const char *)

- void setclickedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void setenteredEvent(const char *)
- void setpressedEvent(const char *)
- void setviewportEnteredEvent(const char *)
- const char *getcollapsedEvent(void)
- const char *getexpandedEvent(void)
- const char *getactivatedEvent(void)
- const char *getclickedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *getenteredEvent(void)
- const char *getpressedEvent(void)
- const char *getviewportEnteredEvent(void)

91.324 QTreeWidget Class

C++ Reference : http://doc.qt.io/qt-5/qtreewidget.html

Parameters : QWidget *

Parent Class : QTreeView

- void addTopLevelItem(QTreeWidgetItem *item)
- void closePersistentEditor(QTreeWidgetItem *item, int column)
- int columnCount(void)
- int currentColumn(void)
- QTreeWidgetItem *currentItem(void)
- void editItem(QTreeWidgetItem *item, int column)
- QTreeWidgetItem *headerItem(void)
- int indexOfTopLevelItem(QTreeWidgetItem *item)
- void insertTopLevelItem(int index, QTreeWidgetItem *item)
- QTreeWidgetItem *invisibleRootItem(void)
- QTreeWidgetItem *itemAbove(QTreeWidgetItem *item)
- QTreeWidgetItem *itemAt(int x, int y)
- QTreeWidgetItem *itemBelow(QTreeWidgetItem *item)
- QWidget *itemWidget(QTreeWidgetItem *item, int column)
- void openPersistentEditor(QTreeWidgetItem *item, int column)
- void removeItemWidget(QTreeWidgetItem *item, int column)
- void setColumnCount(int columns)

- void setCurrentItem(QTreeWidgetItem * item, QItemSelectionModel::SelectionFlag column)
- void setHeaderItem(QTreeWidgetItem *item)
- void setHeaderLabel(QString)
- void setHeaderLabels(QStringList)
- void setItemWidget(QTreeWidgetItem *item, int column, QWidget * widget)
- int sortColumn(void)
- void sortItems(int column, Qt::SortOrder order)
- QTreeWidgetItem *takeTopLevelItem(int index)
- QTreeWidgetItem *topLevelItem(int index)
- int topLevelItemCount(void)
- QRect visualItemRect(QTreeWidgetItem *item)
- void setSelectionModel(QItemSelectionModel *selectionModel)
- void clear(void)
- void collapseItem(QTreeWidgetItem *item)
- void expandItem(QTreeWidgetItem *item)
- void scrollToItem(QTreeWidgetItem *item, QAbstractItemView::ScrollHint hint)
- void setcollapsedEvent(const char *)
- void setexpandedEvent(const char *)
- void setactivatedEvent(const char *)
- void setclickedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void setenteredEvent(const char *)
- void setpressedEvent(const char *)
- void setviewportEnteredEvent(const char *)
- void setcurrentItemChangedEvent(const char *)
- void setitemActivatedEvent(const char *)
- void setitemChangedEvent(const char *)
- void setitemClickedEvent(const char *)
- void setitemCollapsedEvent(const char *)
- void setitemDoubleClickedEvent(const char *)
- void setitemEnteredEvent(const char *)
- void setitemExpandedEvent(const char *)
- void setitemPressedEvent(const char *)
- void setitemSelectionChangedEvent(const char *)
- const char *getcollapsedEvent(void)
- const char *getexpandedEvent(void)

- const char *getactivatedEvent(void)
- const char *getclickedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *getenteredEvent(void)
- const char *getpressedEvent(void)
- const char *getviewportEnteredEvent(void)
- const char *getcurrentItemChangedEvent(void)
- const char *getitemActivatedEvent(void)
- const char *getitemChangedEvent(void)
- const char *getitemClickedEvent(void)
- const char *getitemCollapsedEvent(void)
- const char *getitemDoubleClickedEvent(void)
- const char *getitemEnteredEvent(void)
- const char *getitemExpandedEvent(void)
- const char *getitemPressedEvent(void)
- const char *getitemSelectionChangedEvent(void)

91.325 QTreeWidgetItem Class

C++ Reference : http://doc.qt.io/qt-5/qtreewidgetitem.html

Parameters : void

- void addChild(QTreeWidgetItem *child)
- QBrush background(int column)
- int checkState(int column)
- QTreeWidgetItem *child(int index)
- int childCount(void)
- int childIndicatorPolicy(void)
- QTreeWidgetItem *clone(void)
- int columnCount(void)
- QVariant data(int column, int role)
- int flags(void)
- QFont font(int column)
- QBrush foreground(int column)
- QIcon icon(int column)
- int indexOfChild(QTreeWidgetItem *child)
- void insertChild(int index, QTreeWidgetItem *child)

- bool isDisabled(void)
- bool isExpanded(void)
- bool isFirstColumnSpanned(void)
- bool isHidden(void)
- bool isSelected(void)
- QTreeWidgetItem *parent(void)
- void read(QDataStream)
- void removeChild(QTreeWidgetItem *child)
- void setBackground(int column,QBrush)
- void setCheckState(int column, Qt::CheckState state)
- void setChildIndicatorPolicy(QTreeWidgetItem::ChildIndicatorPolicy policy)
- void setData(int column, int role,QVariant)
- void setDisabled(bool disabled)
- void setExpanded(bool expand)
- void setFirstColumnSpanned(bool span)
- void setFlags(Qt::ItemFlag flags)
- void setFont(int column, QFont)
- void setForeground(int column, QBrush)
- void setHidden(bool hide)
- void setIcon(int column, QIcon)
- void setSelected(bool select)
- void setSizeHint(int column, QSize)
- void setStatusTip(int column, QString)
- void setText(int column, QString)
- void setTextAlignment(int column, int alignment)
- void setToolTip(int column, QString)
- void setWhatsThis(int column, QString)
- QSize sizeHint(int column)
- void sortChildren(int column, Qt::SortOrder order)
- QString statusTip(int column)
- QTreeWidgetItem *takeChild(int index)
- QString text(int column)
- int textAlignment(int column)
- QString toolTip(int column)
- QTreeWidget *treeWidget(void)
- int type(void)

- QString whatsThis(int column)
- void write(QDataStream)

91.326 QUrl Class

C++ Reference : http://doc.qt.io/qt-5/qurl.html

Parameters : QString

- void clear(void)
- QString errorString(void)
- QString fileName(QUrl::ComponentFormattingOption options)
- QString fragment(QUrl::ComponentFormattingOption options)
- bool hasFragment(void)
- bool hasQuery(void)
- QString host(QUrl::ComponentFormattingOption options)
- bool isEmpty(void)
- bool isLocalFile(void)
- bool isParentOf(QUrl)
- bool isRelative(void)
- bool isValid(void)
- QString path(QUrl::ComponentFormattingOption options)
- int port(int defaultPort)
- QString query(QUrl::ComponentFormattingOption options)
- QUrl resolved(QUrl)
- QString scheme(void)
- void setAuthority(QString, QUrl::ParsingMode mode)
- void setFragment(QString, QUrl::ParsingMode mode)
- void setHost(QString, QUrl::ParsingMode mode)
- void setPassword(QString, QUrl::ParsingMode mode)
- void setPath(QString, QUrl::ParsingMode mode)
- void setPort(int port)
- void setQuery(QString, QUrl::ParsingMode mode)
- void setScheme(QString)
- void setUrl(QString, QUrl::ParsingMode parsingMode)
- void setUserInfo(QString, QUrl::ParsingMode mode)
- void setUserName(QString, QUrl::ParsingMode mode)
- void swap(QUrl)

- QString userInfo(QUrl::ComponentFormattingOption options)
- QString userName(QUrl::ComponentFormattingOption options)
- QUrl fromLocalFile(QString)

91.327 QUuid Class

C++ Reference : http://doc.qt.io/qt-5/quuid.html

Parameters : void

- QString toString(void)

91.328 QVBarModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qvbarmodelmapper.html

Parameters : QObject *

Parent Class : QObject

- int firstRow(void)
- int lastBarSetColumn(void)
- QAbstractItemModel * model(void)
- int rowCount(void)
- QAbstractBarSeries * series(void)
- void setFirstBarSetColumn(int firstBarSetColumn)
- void setFirstRow(int firstRow)
- void setLastBarSetColumn(int lastBarSetColumn)
- void setModel(QAbstractItemModel *model)
- void setRowCount(int rowCount)
- void setSeries(QAbstractBarSeries *series)
- void setfirstBarSetColumnChangedEvent(const char *)
- void setfirstRowChangedEvent(const char *)
- void setlastBarSetColumnChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setrowCountChangedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- const char *getfirstBarSetColumnChangedEvent(void)
- const char *getfirstRowChangedEvent(void)
- const char *getlastBarSetColumnChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getrowCountChangedEvent(void)

- const char *getseriesReplacedEvent(void)

91.329 QVBoxLayout Class

C++ Reference : http://doc.qt.io/qt-5/qvboxlayout.html

Parameters : void

Parent Class : QBoxLayout

- void addWidget(QWidget *)
- void addLayout(QLayout *)

91.330 QVBoxPlotModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qvboxplotmodelmapper.html

Parameters : QObject *

Parent Class : QObject

- int firstRow(void)
- int lastBoxSetColumn(void)
- QAbstractItemModel * model(void)
- int rowCount(void)
- QBoxPlotSeries * series(void)
- void setFirstBoxSetColumn(int firstBoxSetColumn)
- void setFirstRow(int firstRow)
- void setLastBoxSetColumn(int lastBoxSetColumn)
- void setModel(QAbstractItemModel *model)
- void setRowCount(int rowCount)
- void setSeries(QBoxPlotSeries *series)
- void setfirstBoxSetColumnChangedEvent(const char *)
- void setfirstRowChangedEvent(const char *)
- void setlastBoxSetColumnChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setrowCountChangedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- const char *getfirstBoxSetColumnChangedEvent(void)
- const char *getfirstRowChangedEvent(void)
- const char *getlastBoxSetColumnChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getrowCountChangedEvent(void)

- const char *getseriesReplacedEvent(void)

91.331 QVCandlestickModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qvcandlestickmodelmapper.html

Parameters : QObject *

Parent Class : QCandlestickModelMapper

- int closeRow(void)
- int firstSetColumn(void)
- int highRow(void)
- int lastSetColumn(void)
- int lowRow(void)
- int openRow(void)
- void setCloseRow(int closeRow)
- void setFirstSetColumn(int firstSetColumn)
- void setHighRow(int highRow)
- void setLastSetColumn(int lastSetColumn)
- void setLowRow(int lowRow)
- void setOpenRow(int openRow)
- void setTimestampRow(int timestampRow)
- int timestampRow(void)
- void setcloseRowChangedEvent(const char *)
- void setfirstSetColumnChangedEvent(const char *)
- void sethighRowChangedEvent(const char *)
- void setlastSetColumnChangedEvent(const char *)
- void setlowRowChangedEvent(const char *)
- void setopenRowChangedEvent(const char *)
- void settimestampRowChangedEvent(const char *)
- const char *getcloseRowChangedEvent(void)
- const char *getfirstSetColumnChangedEvent(void)
- const char *gethighRowChangedEvent(void)
- const char *getlastSetColumnChangedEvent(void)
- const char *getlowRowChangedEvent(void)
- const char *getopenRowChangedEvent(void)
- const char *gettimestampRowChangedEvent(void)

91.332 QVPieModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qvpiemodelmapper.html

Parameters : QObject *

Parent Class : QPieModelMapper

- int firstRow(void)
- int labelsColumn(void)
- QAbstractItemModel * model(void)
- int rowCount(void)
- QPieSeries * series(void)
- void setFirstRow(int firstRow)
- void setLabelsColumn(int labelsColumn)
- void setModel(QAbstractItemModel *model)
- void setRowCount(int rowCount)
- void setSeries(QPieSeries *series)
- void setValuesColumn(int valuesColumn)
- int valuesColumn(void)
- void setfirstRowChangedEvent(const char *)
- void setlabelsColumnChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setrowCountChangedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- void setvaluesColumnChangedEvent(const char *)
- const char *getfirstRowChangedEvent(void)
- const char *getlabelsColumnChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getrowCountChangedEvent(void)
- const char *getseriesReplacedEvent(void)
- const char *getvaluesColumnChangedEvent(void)

91.333 QVXYModelMapper Class

C++ Reference : http://doc.qt.io/qt-5/qvxymodelmapper.html

Parameters : QObject *

Parent Class : QXYModelMapper

- int firstRow(void)
- QAbstractItemModel * model(void)
- int rowCount(void)
- QXYSeries * series(void)
- void setFirstRow(int firstRow)
- void setModel(QAbstractItemModel *model)
- void setRowCount(int rowCount)
- void setSeries(QXYSeries *series)
- void setXColumn(int xColumn)
- void setYColumn(int yColumn)
- int xColumn(void)
- int yColumn(void)
- void setfirstRowChangedEvent(const char *)
- void setmodelReplacedEvent(const char *)
- void setrowCountChangedEvent(const char *)
- void setseriesReplacedEvent(const char *)
- void setxColumnChangedEvent(const char *)
- void setyColumnChangedEvent(const char *)
- const char *getfirstRowChangedEvent(void)
- const char *getmodelReplacedEvent(void)
- const char *getrowCountChangedEvent(void)
- const char *getseriesReplacedEvent(void)
- const char *getxColumnChangedEvent(void)
- const char *getyColumnChangedEvent(void)

91.334 QValueAxis Class

C++ Reference : http://doc.qt.io/qt-5/qvalueaxis.html

Parameters : QObject *

Parent Class : QAbstractAxis

- QString labelFormat(void)
- qreal max(void)
- qreal min(void)
- int minorTickCount(void)
- void setLabelFormat(QString format)
- void setMax(qreal max)
- void setMin(qreal min)
- void setMinorTickCount(int count)
- void setRange(qreal min, qreal max)
- void setTickCount(int count)
- void setTickAnchor(qreal anchor)
- void setTickInterval(qreal insterval)
- void setTickType(QValueAxis::TickType type)
- qreal tickAnchor(void)
- qreal tickInterval(void)
- QValueAxis::TickType tickType(void)
- int tickCount(void)
- void setlabelFormatChangedEvent(const char *)
- void setmaxChangedEvent(const char *)
- void setminChangedEvent(const char *)
- void setminorTickCountChangedEvent(const char *)
- void setrangeChangedEvent(const char *)
- void settickAnchorChangedEvent(const char *)
- void settickCountChangedEvent(const char *)
- void settickIntervalChangedEvent(const char *)
- void settickTypeChangedEvent(const char *)
- const char *getlabelFormatChangedEvent(void)
- const char *getmaxChangedEvent(void)
- const char *getminChangedEvent(void)
- const char *getminorTickCountChangedEvent(void)
- const char *getrangeChangedEvent(void)

- const char *gettickAnchorChangedEvent(void)
- const char *gettickCountChangedEvent(void)
- const char *gettickIntervalChangedEvent(void)
- const char *gettickTypeChangedEvent(void)

91.335 QVariant Class

C++ Reference : http://doc.qt.io/qt-5/qvariant.html

Parameters : void

- bool canConvert(int targetTypeId)
- void clear(void)
- bool convert(int targetTypeId)
- bool isNull(void)
- bool isValid(void)
- void swap(QVariant)
- QBitArray toBitArray(void)
- bool toBool(void)
- QByteArray toByteArray(void)
- QChar toChar(void)
- QDate toDate(void)
- QDateTime toDateTime(void)
- double toDouble(bool *ok)
- QEasingCurve toEasingCurve(void)
- float toFloat(bool *ok)
- int toInt(bool *ok)
- QJsonArray toJsonArray(void)
- QJsonDocument toJsonDocument(void)
- QJsonObject toJsonObject(void)
- QJsonValue toJsonValue(void)
- QLine toLine(void)
- QLineF toLineF(void)
- QLocale toLocale(void)
- qlonglong toLongLong(bool *ok)
- QModelIndex toModelIndex(void)
- QPointF toPointF(void)
- qreal toReal(bool *ok)

- QRect toRect(void)
- QRectF toRectF(void)
- QRegExp toRegExp(void)
- QRegularExpression toRegularExpression(void)
- QSize toSize(void)
- QSizeF toSizeF(void)
- QStringList toStringList(void)
- QTime toTime(void)
- uint toUInt(bool *ok)
- qulonglong toULongLong(bool *ok)
- QUrl toUrl(void)
- QUuid toUuid(void)
- QVariant::Type type(void)
- const char *typeName(void)
- int userType(void)
- QString toString(void)

91.336 QVariant2 Class

Parent Class : QVariant

Parameters : int

91.337 QVariant3 Class

Parent Class : QVariant

Parameters : float

91.338 QVariant4 Class

Parent Class : QVariant

Parameters : double

91.339 QVariant5 Class

Parent Class : QVariant

Parameters : QString

91.340 QVariantDouble Class

C++ Reference : http://doc.qt.io/qt-5/qvariantdouble.html

Parent Class : QVariant

Parameters : double

91.341 QVariantFloat Class

C++ Reference : http://doc.qt.io/qt-5/qvariantfloat.html

Parent Class : QVariant

Parameters : float

91.342 QVariantInt Class

C++ Reference : http://doc.qt.io/qt-5/qvariantint.html

Parent Class : QVariant

Parameters : int

91.343 QVariantString Class

C++ Reference : http://doc.qt.io/qt-5/qvariantstring.html

Parent Class : QVariant

Parameters : QString

91.344 QVector2D Class

C++ Reference : http://doc.qt.io/qt-5/qvector2d.html

Parameters : float,float

- float distanceToLine(QVector2D point, QVector2D direction)
- float distanceToPoint(QVector2D point)
- bool isNull(void)
- float length(void)

- float lengthSquared(void)
- void normalize(void)
- QVector2D normalized(void)
- void setX(float x)
- void setY(float y)
- QPoint toPoint(void)
- QPointF toPointF(void)
- QVector3D toVector3D(void)
- QVector4D toVector4D(void)
- float x(void)
- float y(void)
- float dotProduct(QVector2D v1, QVector2D v2)

91.345 QVector3D Class

C++ Reference : http://doc.qt.io/qt-5/qvector3d.html

Parameters : float,float,float

- float distanceToLine(QVector3D point, QVector3D direction)
- float distanceToPlane(QVector3D plane, QVector3D normal)
- float distanceToPlane_2(QVector3D plane1, QVector3D plane2, QVector3D plane3)
- float distanceToPoint(QVector3D point)
- bool isNull(void)
- float length(void)
- float lengthSquared(void)
- void normalize(void)
- QVector3D normalized(void)
- QVector3D project(QMatrix4x4 modelView, QMatrix4x4 projection, QRect viewport)
- void setX(float x)
- void setY(float y)
- void setZ(float z)
- QPoint toPoint(void)
- QPointF toPointF(void)
- QVector2D toVector2D(void)
- QVector4D toVector4D(void)
- QVector3D unproject(QMatrix4x4 modelView, QMatrix4x4 projection, QRect viewport)
- float x(void)

- float y(void)
- float z(void)
- QVector3D crossProduct(QVector3D v1, QVector3D v2)
- float dotProduct(QVector3D v1, QVector3D v2)
- QVector3D normal(QVector3D v1, QVector3D v2)
- QVector3D normal_2(QVector3D v1, QVector3D v2, QVector3D v3)

91.346 QVector4D Class

C++ Reference : http://doc.qt.io/qt-5/qvector4d.html

Parameters : float,float,float,float

- bool isNull(void)
- float length(void)
- float lengthSquared(void)
- void normalize(void)
- QVector4D normalized(void)
- void setW(float w)
- void setX(float x)
- void setY(float y)
- void setZ(float z)
- QPoint toPoint(void)
- QPointF toPointF(void)
- QVector2D toVector2D(void)
- QVector2D toVector2DAffine(void)
- QVector3D toVector3D(void)
- QVector3D toVector3DAffine(void)
- float w(void)
- float x(void)
- float y(void)
- float z(void)
- float dotProduct(QVector4D v1, QVector4D v2)

91.347 QVectorQVoice Class

C++ Reference : http://doc.qt.io/qt-5/qvectorqvoice.html

Parameters : void

- int count(void)
- QVoice value(int i)

91.348 QVideoWidget Class

C++ Reference : http://doc.qt.io/qt-5/qvideowidget.html

Parameters : QWidget *parent

Parent Class : QWidget

- int aspectRatioMode(void)
- int brightness(void)
- int contrast(void)
- int hue(void)
- bool isFullScreen(void)
- int saturation(void)
- void setAspectRatioMode(Qt::AspectRatioMode mode)
- void setBrightness(int brightness)
- void setContrast(int contrast)
- void setFullScreen(bool fullScreen)
- void setHue(int hue)
- void setSaturation(int saturation)
- void setbrightnessChangedEvent(const char *)
- void setcontrastChangedEvent(const char *)
- void setfullScreenChangedEvent(const char *)
- void sethueChangedEvent(const char *)
- void setsaturationChangedEvent(const char *)
- const char *getbrightnessChangedEvent(void)
- const char *getcontrastChangedEvent(void)
- const char *getfullScreenChangedEvent(void)
- const char *gethueChangedEvent(void)
- const char *getsaturationChangedEvent(void)

91.349 QVideoWidgetControl Class

C++ Reference : http://doc.qt.io/qt-5/qvideowidgetcontrol.html

Parent Class : QMediaControl

91.350 QViewport Class

C++ Reference : http://doc.qt.io/qt-5/qt3drender-qviewport.html

Parameters : Qt3DCore::QNode *

- float gamma(void)
- QRectF normalizedRect(void)
- void setGamma(float gamma)
- void setNormalizedRect(QRectF normalizedRect)

91.351 QVoice Class

C++ Reference : http://doc.qt.io/qt-5/qvoice.html

Parameters : void

- QVoice::Age age(void)
- QVoice::Gender gender(void)
- QString name(void)
- QString ageName(QVoice::Age age)
- QString genderName(QVoice::Gender gender)

91.352 QWebEnginePage Class

C++ Reference : http://doc.qt.io/qt-5/qwebenginepage.html

Parameters : QObject *

Parent Class : QObject

- QAction * action(QWebEnginePage::WebAction action)
- QColor backgroundColor(void)
- QSizeF contentsSize(void)
- QWebEnginePage * devToolsPage(void)
- void download(QUrl url, QString filename)
- void findText(QString subString, QWebEnginePage::FindFlags options, QWebEngineCallback<bool> resultCallback)
- bool hasSelection(void)

- QWebEngineHistory * history(void)
- QIcon icon(void)
- QUrl iconUrl(void)
- QWebEnginePage * inspectedPage(void)
- bool isAudioMuted(void)
- void load_3(QWebEngineHttpRequest request)
- void print(QPrinter *printer,QWebEngineCallback<bool>)
- bool recentlyAudible(void)
- QUrl requestedUrl(void)
- void runJavaScript(QString scriptSource)
- void runJavaScript_2(QString scriptSource, quint32 worldId)
- QPointF scrollPosition(void)
- QString selectedText(void)
- void setAudioMuted(bool muted)
- void setBackgroundColor(QColor color)
- void setContent(QByteArray data, QString mimeType, QUrl baseUrl)
- void setDevToolsPage(QWebEnginePage *devToolsPage)
- void setFeaturePermission(QUrl securityOrigin, QWebEnginePage::Feature feature, QWebEnginePage::PermissionPolicy policy)
- void setHtml(QString html, QUrl baseUrl)
- void setInspectedPage(QWebEnginePage *page)
- void setWebChannel_2(QWebChannel *channel)
- void setZoomFactor(qreal factor)
- QWebEngineSettings * settings(void)
- QString title(void)
- QUrl url(void)
- QWidget * view(void)
- QWebChannel * webChannel(void)
- qreal zoomFactor(void)
- void setaudioMutedChangedEvent(const char *)
- void setauthenticationRequiredEvent(const char *)
- void setcontentsSizeChangedEvent(const char *)
- void setfeaturePermissionRequestCanceledEvent(const char *)
- void setfeaturePermissionRequestedEvent(const char *)
- void setfindTextFinishedEvent(const char *)
- void setfullScreenRequestedEvent(const char *)

- void setgeometryChangeRequestedEvent(const char *)
- void seticonChangedEvent(const char *)
- void seticonUrlChangedEvent(const char *)
- void setloadFinishedEvent(const char *)
- void setloadProgressEvent(const char *)
- void setloadStartedEvent(const char *)
- void setpdfPrintingFinishedEvent(const char *)
- void setprintRequestedEvent(const char *)
- void setproxyAuthenticationRequiredEvent(const char *)
- void setquotaRequestedEvent(const char *)
- void setrecentlyAudibleChangedEvent(const char *)
- void setrecommendedStateChangedEvent(const char *)
- void setregisterProtocolHandlerRequestedEvent(const char *)
- void setselectClientCertificateEvent(const char *)
- void setselectionChangedEvent(const char *)
- void settitleChangedEvent(const char *)
- void seturlChangedEvent(const char *)
- void setvisibleChangedEvent(const char *)
- void setwindowCloseRequestedEvent(const char *)
- const char *getaudioMutedChangedEvent(void)
- const char *getauthenticationRequiredEvent(void)
- const char *getcontentsSizeChangedEvent(void)
- const char *getfeaturePermissionRequestCanceledEvent(void)
- const char *getfeaturePermissionRequestedEvent(void)
- const char *getfindTextFinishedEvent(void)
- const char *getfullScreenRequestedEvent(void)
- const char *getgeometryChangeRequestedEvent(void)
- const char *geticonChangedEvent(void)
- const char *geticonUrlChangedEvent(void)
- const char *getloadFinishedEvent(void)
- const char *getloadProgressEvent(void)
- const char *getloadStartedEvent(void)
- const char *getpdfPrintingFinishedEvent(void)
- const char *getprintRequestedEvent(void)
- const char *getproxyAuthenticationRequiredEvent(void)
- const char *getquotaRequestedEvent(void)

- const char *getrecentlyAudibleChangedEvent(void)
- const char *getrecommendedStateChangedEvent(void)
- const char *getregisterProtocolHandlerRequestedEvent(void)
- const char *getselectClientCertificateEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettitleChangedEvent(void)
- const char *geturlChangedEvent(void)
- const char *getvisibleChangedEvent(void)
- const char *getwindowCloseRequestedEvent(void)

91.353 QWebEngineView Class

C++ Reference : http://doc.qt.io/qt-5/qwebengineview.html

Parameters : QWidget *parent

Parent Class : QWidget

- bool hasSelection(void)
- QWebEngineHistory *history(void)
- void load(QUrl) # In RingQt use : void loadpage(QUrl)
- QWebEnginePage *page(void)
- QAction *pageAction(QWebEnginePage::WebAction action)
- QString selectedText(void)
- void setContent(QByteArray,QString,QUrl)
- void setHtml(QString,QUrl)
- void setPage(QWebEnginePage *page)
- void setUrl(QUrl)
- void setZoomFactor(qreal factor)
- QWebSettings *settings(void)
- QString title(void)
- void triggerPageAction(QWebEnginePage::WebAction action, bool checked)
- QUrl url(void)
- qreal zoomFactor(void)
- void back(void)
- void forward(void)
- void reload(void)
- void stop(void)
- void setloadFinishedEvent(const char *)

- void setloadProgressEvent(const char *)
- void setloadStartedEvent(const char *)
- void setselectionChangedEvent(const char *)
- void settitleChangedEvent(const char *)
- void seturlChangedEvent(const char *)
- const char *getloadFinishedEvent(void)
- const char *getloadProgressEvent(void)
- const char *getloadStartedEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettitleChangedEvent(void)
- const char *geturlChangedEvent(void)
- void print(QPrinter *printer,const char *cCode)

91.354 QWebView Class

C++ Reference : http://doc.qt.io/qt-5/qwebview.html

Parameters : QWidget *parent

Parent Class : QWidget

- QWebHistory *history(void)
- QAction *pageAction(QWebPage::WebAction action)
- void setContent(QByteArray,QString,QUrl)
- void setHtml(QString,QUrl)
- void setPage(QWebPage *page)
- void setZoomFactor(qreal factor)
- QWebSettings *settings(void)
- void triggerPageAction(QWebPage::WebAction action, bool checked)
- QUrl url(void)
- qreal zoomFactor(void)
- void back(void)
- void forward(void)
- void print(QPrinter *printer)
- void reload(void)
- void stop(void)
- void setloadProgressEvent(const char *)
- void setloadStartedEvent(const char *)
- void setselectionChangedEvent(const char *)

- void seturlChangedEvent(const char *)
- const char *getloadFinishedEvent(void)
- const char *getloadProgressEvent(void)
- const char *getloadStartedEvent(void)
- const char *getselectionChangedEvent(void)
- const char *gettitleChangedEvent(void)
- const char *geturlChangedEvent(void)

91.355 QWebView Class

C++ Reference : http://doc.qt.io/qt-5/qwebview.html

Parameters : QWidget *parent

Parent Class : QWebEngineView

91.356 QWidget Class

C++ Reference : http://doc.qt.io/qt-5/qwidget.html

Parameters : void

Parent Class : QObject

- bool acceptDrops(void)
- QString accessibleDescription(void)
- QString accessibleName(void)
- void activateWindow(void)
- void addAction(QAction *action)
- void adjustSize(void)
- bool autoFillBackground(void)
- int backgroundRole(void)
- QSize baseSize(void)
- QWidget *childAt(int x, int y)
- QRect childrenRect(void)
- QRegion childrenRegion(void)
- void clearFocus(void)
- void clearMask(void)
- QMargins contentsMargins(void)
- QRect contentsRect(void)
- int contextMenuPolicy(void)
- QCursor cursor(void)

- int effectiveWinId(void)
- void ensurePolished(void)
- int focusPolicy(void)
- QWidget *focusProxy(void)
- QWidget *focusWidget(void)
- QFont font(void)
- QFontInfo fontInfo(void)
- int foregroundRole(void)
- QRect frameGeometry(void)
- QSize frameSize(void)
- QRect geometry(void)
- QPixmap grab(QRect)
- void grabGesture(Qt::GestureType gesture, Qt::GestureFlag flags)
- void grabKeyboard(void)
- void grabMouse(void)
- int grabShortcut(QKeySequence , Qt::ShortcutContext context)
- QGraphicsEffect *graphicsEffect(void)
- QGraphicsProxyWidget *graphicsProxyWidget(void)
- bool hasFocus(void)
- bool hasMouseTracking(void)
- int height(void)
- int heightForWidth(int w)
- int inputMethodHints(void)
- QVariant inputMethodQuery(Qt::InputMethodQuery query)
- void insertAction(QAction *before, QAction *action)
- bool isActiveWindow(void)
- bool isAncestorOf(QWidget *child)
- bool isEnabled(void)
- bool isEnabledTo(QWidget *ancestor)
- bool isFullScreen(void)
- bool isHidden(void)
- bool isMaximized(void)
- bool isMinimized(void)
- bool isModal(void)
- bool isVisible(void)
- bool isVisibleTo(QWidget *ancestor)

- bool isWindow(void)
- bool isWindowModified(void)
- QLayout *layout(void)
- int layoutDirection(void)
- QLocale locale(void)
- QPoint mapFrom(QWidget *parent, QPoint)
- QPoint mapFromGlobal(QPoint)
- QPoint mapFromParent(QPoint)
- QPoint mapTo(QWidget *parent, QPoint)
- QPoint mapToGlobal(QPoint pos)
- QPoint mapToParent(QPoint pos)
- QRegion mask(void)
- int maximumHeight(void)
- QSize maximumSize(void)
- int maximumWidth(void)
- int minimumHeight(void)
- QSize minimumSize(void)
- int minimumWidth(void)
- void move(int x, int y)
- QWidget *nativeParentWidget(void)
- QWidget *nextInFocusChain(void)
- QRect normalGeometry(void)
- void overrideWindowFlags(Qt::WindowType flags)
- QPalette palette(void)
- QWidget *parentWidget(void)
- QPoint pos(void)
- QWidget *previousInFocusChain(void)
- QRect rect(void)
- void releaseKeyboard(void)
- void releaseMouse(void)
- void releaseShortcut(int id)
- void removeAction(QAction *action)
- void render(QPaintDevice *target, QPoint,QRegion, QWidget::RenderFlag)
- void repaint(void)
- void resize(int w, int h)
- bool restoreGeometry(QByteArray)

- QByteArray saveGeometry(void)
- void scroll(int dx, int dy)
- void setAcceptDrops(bool on)
- void setAccessibleDescription(QString)
- void setAccessibleName(QString)
- void setAttribute(Qt::WidgetAttribute attribute, bool on)
- void setAutoFillBackground(bool enabled)
- void setBackgroundRole(QPalette::ColorRole role)
- void setBaseSize(int basew, int baseh)
- void setContentsMargins(int left, int top, int right, int bottom)
- void setContextMenuPolicy(Qt::ContextMenuPolicy policy)
- void setCursor(QCursor)
- void setFixedHeight(int h)
- void setFixedSize(int w, int h)
- void setFixedWidth(int w)
- void setFocus(Qt::FocusReason reason)
- void setFocusPolicy(Qt::FocusPolicy policy)
- void setFocusProxy(QWidget *w)
- void setFont(QFont)
- void setForegroundRole(QPalette::ColorRole role)
- void setGeometry(int x, int y, int w, int h)
- void setGraphicsEffect(QGraphicsEffect *effect)
- void setInputMethodHints(Qt::InputMethodHint hints)
- void setLayout(QLayout *layout)
- void setLayoutDirection(Qt::LayoutDirection direction)
- void setLocale(QLocale)
- void setMask(QBitmap)
- void setMaximumHeight(int maxh)
- void setMaximumSize(int maxw, int maxh)
- void setMaximumWidth(int maxw)
- void setMinimumHeight(int minh)
- void setMinimumSize(int minw, int minh)
- void setMinimumWidth(int minw)
- void setMouseTracking(bool enable)
- void setPalette(QPalette)
- void setParent(QWidget *parent)

- void setShortcutAutoRepeat(int id, bool enable)
- void setShortcutEnabled(int id, bool enable)
- void setSizeIncrement(int w, int h)
- void setSizePolicy(QSizePolicy::Policy horizontal, QSizePolicy::Policy vertical)
- void setStatusTip(QString)
- void setStyle(QStyle *style)
- void setToolTip(QString)
- void setUpdatesEnabled(bool enable)
- void setWhatsThis(QString)
- void setWindowFilePath(QString)
- void setWindowFlags(Qt::WindowType type)
- void setWindowIcon(QIcon)
- void setWindowIconText(QString)
- void setWindowModality(Qt::WindowModality windowModality)
- void setWindowOpacity(double level)
- void setWindowRole(QString)
- void setWindowState(Qt::WindowState windowState)
- QSize size(void)
- QSize sizeIncrement(void)
- QSizePolicy sizePolicy(void)
- void stackUnder(QWidget *w)
- QString statusTip(void)
- QStyle *style(void)
- QString styleSheet(void)
- bool testAttribute(Qt::WidgetAttribute attribute)
- QString toolTip(void)
- bool underMouse(void)
- void ungrabGesture(Qt::GestureType gesture)
- void unsetCursor(void)
- void unsetLayoutDirection(void)
- void unsetLocale(void)
- void update(int x, int y, int w, int h)
- void updateGeometry(void)
- bool updatesEnabled(void)
- QRegion visibleRegion(void)
- QString whatsThis(void)

- int width(void)
- int winId(void)
- QWidget *window(void)
- QString windowFilePath(void)
- int windowFlags(void)
- QWindow *windowHandle(void)
- QIcon windowIcon(void)
- QString windowIconText(void)
- int windowModality(void)
- double windowOpacity(void)
- QString windowRole(void)
- int windowState(void)
- QString windowTitle(void)
- int windowType(void)
- int x(void)
- int y(void)
- bool close(void)
- void hide(void)
- void lower(void)
- void raise(void)
- void setDisabled(bool disable)
- void setEnabled(bool)
- void setHidden(bool hidden)
- void setStyleSheet(QString)
- void setWindowModified(bool)
- void setWindowTitle(QString)
- void show(void)
- void showFullScreen(void)
- void showMaximized(void)
- void showMinimized(void)
- void showNormal(void)
- QWidget *find(int id)
- QWidget *keyboardGrabber(void)
- QWidget *mouseGrabber(void)
- void setTabOrder(QWidget *first, QWidget *second)
- QWidget *createWindowContainer(QWindow *window, QWidget *parent, Qt::WindowFlags flags)

91.357 QWindow Class

C++ Reference : http://doc.qt.io/qt-5/qwindow.html

Parameters : QScreen *

Parent Class : QObject

- QSize baseSize(void)
- Qt::ScreenOrientation contentOrientation(void)
- void create(void)
- QCursor cursor(void)
- void destroy(void)
- qreal devicePixelRatio(void)
- QString filePath(void)
- Qt::WindowFlags flags(void)
- QObject * focusObject(void)
- QRect frameGeometry(void)
- QMargins frameMargins(void)
- QPoint framePosition(void)
- QRect geometry(void)
- int height(void)
- QIcon icon(void)
- bool isActive(void)
- bool isAncestorOf(QWindow *child, QWindow::AncestorMode mode)
- bool isExposed(void)
- bool isModal(void)
- bool isTopLevel(void)
- bool isVisible(void)
- QPoint mapFromGlobal(QPoint pos)
- QPoint mapToGlobal(QPoint pos)
- QRegion mask(void)
- int maximumHeight(void)
- QSize maximumSize(void)
- int maximumWidth(void)
- int minimumHeight(void)
- QSize minimumSize(void)
- int minimumWidth(void)
- Qt::WindowModality modality(void)

- qreal opacity(void)
- QPoint position(void)
- void reportContentOrientationChange(Qt::ScreenOrientation orientation)
- QSurfaceFormat requestedFormat(void)
- void resize(QSize newSize)
- void resize_2(int w, int h)
- QScreen * screen(void)
- void setBaseSize(QSize size)
- void setCursor(QCursor cursor)
- void setFilePath(QString filePath)
- void setFlags(Qt::WindowFlags flags)
- void setFormat(QSurfaceFormat format)
- void setFramePosition(QPoint point)
- void setGeometry(int posx, int posy, int w, int h)
- void setGeometry_2(QRect rect)
- void setIcon(QIcon icon)
- bool setKeyboardGrabEnabled(bool grab)
- void setMask(QRegion region)
- void setMaximumSize(QSize size)
- void setMinimumSize(QSize size)
- void setModality(Qt::WindowModality modality)
- bool setMouseGrabEnabled(bool grab)
- void setOpacity(qreal level)
- void setParent(QWindow *parent)
- void setPosition(QPoint pt)
- void setPosition_2(int posx, int posy)
- void setScreen(QScreen *newScreen)
- void setSizeIncrement(QSize size)
- void setTransientParent(QWindow *parent)
- void setVisibility(QWindow::Visibility v)
- void setWindowState(Qt::WindowState state)
- QSize sizeIncrement(void)
- QString title(void)
- QWindow * transientParent(void)
- Qt::WindowType type(void)
- void unsetCursor(void)

- QWindow::Visibility visibility(void)
- int width(void)
- WId winId(void)
- Qt::WindowState windowState(void)
- int x(void)
- int y(void)
- void alert(int msec)
- bool close(void)
- void hide(void)
- void lower(void)
- void raise(void)
- void requestActivate(void)
- void setHeight(int arg)
- void setMaximumHeight(int h)
- void setMaximumWidth(int w)
- void setMinimumHeight(int h)
- void setMinimumWidth(int w)
- void setTitle(QString)
- void setVisible(bool visible)
- void setWidth(int arg)
- void setX(int arg)
- void setY(int arg)
- void show(void)
- void showFullScreen(void)
- void showMaximized(void)
- void showMinimized(void)
- void showNormal(void)
- QWindow * fromWinId(WId id)
- void setactiveChangedEvent(const char *)
- void setcontentOrientationChangedEvent(const char *)
- void setfocusObjectChangedEvent(const char *)
- void setheightChangedEvent(const char *)
- void setmaximumHeightChangedEvent(const char *)
- void setmaximumWidthChangedEvent(const char *)
- void setminimumHeightChangedEvent(const char *)
- void setminimumWidthChangedEvent(const char *)

- void setmodalityChangedEvent(const char *)
- void setopacityChangedEvent(const char *)
- void setscreenChangedEvent(const char *)
- void setvisibilityChangedEvent(const char *)
- void setvisibleChangedEvent(const char *)
- void setwidthChangedEvent(const char *)
- void setwindowStateChangedEvent(const char *)
- void setwindowTitleChangedEvent(const char *)
- void setxChangedEvent(const char *)
- void setyChangedEvent(const char *)
- const char *getactiveChangedEvent(void)
- const char *getcontentOrientationChangedEvent(void)
- const char *getfocusObjectChangedEvent(void)
- const char *getheightChangedEvent(void)
- const char *getmaximumHeightChangedEvent(void)
- const char *getmaximumWidthChangedEvent(void)
- const char *getminimumHeightChangedEvent(void)
- const char *getminimumWidthChangedEvent(void)
- const char *getmodalityChangedEvent(void)
- const char *getopacityChangedEvent(void)
- const char *getscreenChangedEvent(void)
- const char *getvisibilityChangedEvent(void)
- const char *getvisibleChangedEvent(void)
- const char *getwidthChangedEvent(void)
- const char *getwindowStateChangedEvent(void)
- const char *getwindowTitleChangedEvent(void)
- const char *getxChangedEvent(void)
- const char *getyChangedEvent(void)

91.358 QXYLegendMarker Class

C++ Reference : http://doc.qt.io/qt-5/qxylegendmarker.html

Parent Class : QLegendMarker

- QXYSeries * series(void)
- QLegendMarker::LegendMarkerType type(void)

91.359 QXYSeries Class

C++ Reference : http://doc.qt.io/qt-5/qxyseries.html

Parent Class : QAbstractSeries

- void append(qreal x, qreal y)
- void append_2(QPointF point)
- void append_3(QList<QPointF> points)
- QPointF at(int index)
- QBrush brush(void)
- void clear(void)
- QColor color(void)
- int count(void)
- void insert(int index, QPointF point)
- QPen pen(void)
- bool pointLabelsClipping(void)
- QColor pointLabelsColor(void)
- QFont pointLabelsFont(void)
- QString pointLabelsFormat(void)
- bool pointLabelsVisible(void)
- QList<QPointF> points(void)
- QVector<QPointF> pointsVector(void)
- bool pointsVisible(void)
- void remove(qreal x, qreal y)
- void remove_2(QPointF point)
- void remove_3(int index)
- void removePoints(int index, int count)
- void replace(qreal oldX, qreal oldY, qreal newX, qreal newY)
- void replace_2(QPointF oldPoint, QPointF newPoint)
- void replace_3(int index, qreal newX, qreal newY)
- void replace_4(int index, QPointF newPoint)
- void replace_5(QList<QPointF> points)
- void replace_6(QVector<QPointF> points)
- void setBrush(QBrush brush)
- void setColor(QColor color)
- void setPen(QPen pen)
- void setPointLabelsClipping(bool enabled)

- void setPointLabelsColor(QColor color)
- void setPointLabelsFont(QFont font)
- void setPointLabelsFormat(QString format)
- void setPointLabelsVisible(bool visible)
- void setPointsVisible(bool visible)
- void setclickedEvent(const char *)
- void setcolorChangedEvent(const char *)
- void setdoubleClickedEvent(const char *)
- void sethoveredEvent(const char *)
- void setpenChangedEvent(const char *)
- void setpointAddedEvent(const char *)
- void setpointLabelsClippingChangedEvent(const char *)
- void setpointLabelsColorChangedEvent(const char *)
- void setpointLabelsFontChangedEvent(const char *)
- void setpointLabelsFormatChangedEvent(const char *)
- void setpointLabelsVisibilityChangedEvent(const char *)
- void setpointRemovedEvent(const char *)
- void setpointReplacedEvent(const char *)
- void setpointsRemovedEvent(const char *)
- void setpointsReplacedEvent(const char *)
- void setpressedEvent(const char *)
- void setreleasedEvent(const char *)
- const char *getclickedEvent(void)
- const char *getcolorChangedEvent(void)
- const char *getdoubleClickedEvent(void)
- const char *gethoveredEvent(void)
- const char *getpenChangedEvent(void)
- const char *getpointAddedEvent(void)
- const char *getpointLabelsClippingChangedEvent(void)
- const char *getpointLabelsColorChangedEvent(void)
- const char *getpointLabelsFontChangedEvent(void)
- const char *getpointLabelsFormatChangedEvent(void)
- const char *getpointLabelsVisibilityChangedEvent(void)
- const char *getpointRemovedEvent(void)
- const char *getpointReplacedEvent(void)
- const char *getpointsRemovedEvent(void)

- const char *getpointsReplacedEvent(void)
- const char *getpressedEvent(void)
- const char *getreleasedEvent(void)

91.360 QXmlStreamAttribute Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamattribute.html

Parameters : void

- bool isDefault(void)
- QStringRef name(void)
- QStringRef namespaceUri(void)
- QStringRef prefix(void)
- QStringRef qualifiedName(void)
- QStringRef value(void)

91.361 QXmlStreamAttributes Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamattributes.html

Parameters : void

- void append(QString namespaceUri, QString name, QString value)
- void append_2(QString qualifiedName, QString value)
- bool hasAttribute(QString qualifiedName)
- bool hasAttribute_2(QLatin1String qualifiedName)
- bool hasAttribute_3(QString namespaceUri, QString name)
- QStringRef value(QString namespaceUri, QString name)
- QStringRef value_2(QString namespaceUri, QLatin1String name)
- QStringRef value_3(QLatin1String namespaceUri, QLatin1String name)
- QStringRef value_4(QString qualifiedName)
- QStringRef value_5(QLatin1String qualifiedName)

91.362 QXmlStreamEntityDeclaration Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamentitydeclaration.html

Parameters : void

- QStringRef name(void)
- QStringRef notationName(void)
- QStringRef publicId(void)
- QStringRef systemId(void)
- QStringRef value(void)

91.363 QXmlStreamEntityResolver Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamentityresolver.html

Parameters : void

91.364 QXmlStreamNamespaceDeclaration Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamnamespacedeclaration.html

Parameters : void

- QStringRef namespaceUri(void)
- QStringRef prefix(void)

91.365 QXmlStreamNotationDeclaration Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamnotationdeclaration.html

Parameters : void

- QStringRef name(void)
- QStringRef publicId(void)
- QStringRef systemId(void)

91.366 QXmlStreamReader Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamreader.html

Parameters : void

- void addData(QByteArray)
- void addData_2(QString)
- void addData_3(const char * data)

- void addExtraNamespaceDeclaration(QXmlStreamNamespaceDeclaration)
- void addExtraNamespaceDeclarations(QXmlStreamNamespaceDeclarations)
- bool atEnd(void)
- QXmlStreamAttributes attributes(void)
- qint64 characterOffset(void)
- void clear(void)
- qint64 columnNumber(void)
- QIODevice *device(void)
- QStringRef documentEncoding(void)
- QStringRef documentVersion(void)
- QStringRef dtdName(void)
- QStringRef dtdPublicId(void)
- QStringRef dtdSystemId(void)
- QXmlStreamEntityDeclarations entityDeclarations(void)
- QXmlStreamEntityResolver *entityResolver(void)
- Error error(void)
- QString errorString(void)
- bool hasError(void)
- bool isCDATA(void)
- bool isCharacters(void)
- bool isComment(void)
- bool isDTD(void)
- bool isEndDocument(void)
- bool isEndElement(void)
- bool isEntityReference(void)
- bool isProcessingInstruction(void)
- bool isStandaloneDocument(void)
- bool isStartDocument(void)
- bool isStartElement(void)
- bool isWhitespace(void)
- qint64 lineNumber(void)
- QStringRef name(void)
- QXmlStreamNamespaceDeclarations namespaceDeclarations(void)
- bool namespaceProcessing(void)
- QStringRef namespaceUri(void)
- QXmlStreamNotationDeclarations notationDeclarations(void)

- QStringRef prefix(void)
- QStringRef processingInstructionData(void)
- QStringRef processingInstructionTarget(void)
- QStringRef qualifiedName(void)
- void raiseError(QString)
- QString readElementText(QXmlStreamReader::ReadElementTextBehaviour)
- TokenType readNext(void)
- bool readNextStartElement(void)
- void setDevice(QIODevice *device)
- void setEntityResolver(QXmlStreamEntityResolver *resolver)
- void setNamespaceProcessing(bool)
- void skipCurrentElement(void)
- QStringRef text(void)
- QString tokenString(void)
- TokenType tokenType(void)

91.367 QXmlStreamWriter Class

C++ Reference : http://doc.qt.io/qt-5/qxmlstreamwriter.html

Parameters : void

- bool autoFormatting(void)
- int autoFormattingIndent(void)
- QTextCodec *codec(void)
- QIODevice *device(void)
- bool hasError(void)
- void setAutoFormatting(bool enable)
- void setAutoFormattingIndent(int spacesOrTabs)
- void setCodec(QTextCodec *codec)
- void setCodec_2(const char *codecName)
- void setDevice(QIODevice *device)
- void writeAttribute(QString, QString,QString)
- void writeAttribute_2(QString, QString)
- void writeAttribute_3(QXmlStreamAttribute)
- void writeAttributes(QXmlStreamAttributes)
- void writeCDATA(QString text)
- void writeCharacters(QString text)

- void writeComment(QString text)
- void writeCurrentToken(QXmlStreamReader reader)
- void writeDTD(QString dtd)
- void writeDefaultNamespace(QString namespaceUri)
- void writeEmptyElement(QString namespaceUri, QString name)
- void writeEmptyElement_2(QString qualifiedName)
- void writeEndDocument(void)
- void writeEndElement(void)
- void writeEntityReference(QString name)
- void writeNamespace(QString namespaceUri, QString prefix)
- void writeProcessingInstruction(QString target, QString data)
- void writeStartDocument(QString version)
- void writeStartDocument_2(QString version, bool standalone)
- void writeStartDocument_3(void)
- void writeStartElement(QString namespaceUri, QString name)
- void writeStartElement_2(QString qualifiedName)
- void writeTextElement(QString namespaceUri, QString name, QString text)
- void writeTextElement_2(QString qualifiedName, QString text)

91.368 Qt3DCamera Class

C++ Reference : http://doc.qt.io/qt-5/qt3dcamera.html

Parameters : Qt3DCore::QNode *

Parent Class : QEntity

- float aspectRatio(void)
- float bottom(void)
- float exposure(void)
- float farPlane(void)
- float fieldOfView(void)
- float left(void)
- QCameraLens *lens(void)
- float nearPlane(void)
- void pan(float angle)
- void pan_2(float angle, QVector3D axis)
- void panAboutViewCenter(float angle)
- void panAboutViewCenter_2(float angle, QVector3D axis)

- QQuaternion panRotation(float angle)
- QVector3D position(void)
- QMatrix4x4 projectionMatrix(void)
- Qt3DRender::QCameraLens::ProjectionType projectionType(void)
- float right(void)
- void roll(float angle)
- void rollAboutViewCenter(float angle)
- QQuaternion rollRotation(float angle)
- void rotate(QQuaternion q)
- void rotateAboutViewCenter(QQuaternion q)
- QQuaternion rotation(float angle, QVector3D axis)
- void tilt(float angle)
- void tiltAboutViewCenter(float angle)
- QQuaternion tiltRotation(float angle)
- float top(void)
- Qt3DCore::QTransform * transform(void)
- void translate(QVector3D vLocal, Qt3DRender::QCamera::CameraTranslationOption option)
- void translateWorld(QVector3D vWorld, Qt3DRender::QCamera::CameraTranslationOption option)
- QVector3D upVector(void)
- QVector3D viewCenter(void)
- QVector3D viewVector(void)
- void setAspectRatio(float aspectRatio)
- void setBottom(float bottom)
- void setExposure(float exposure)
- void setFarPlane(float farPlane)
- void setFieldOfView(float fieldOfView)
- void setLeft(float left)
- void setNearPlane(float nearPlane)
- void setPosition(QVector3D position)
- void setProjectionMatrix(QMatrix4x4 projectionMatrix)
- void setProjectionType(Qt3DRender::QCameraLens::ProjectionType type)
- void setRight(float right)
- void setTop(float top)
- void setUpVector(QVector3D upVector)
- void setViewCenter(QVector3D viewCenter)
- void viewAll(void)

- void viewEntity(Qt3DCore::QEntity *entity)
- void viewSphere(QVector3D center, float radius)

91.369 Qt3DWindow Class

C++ Reference : http://doc.qt.io/qt-5/qt3dextras-qt3dwindow.html

Parameters : void

Parent Class : QWindow

- Qt3DRender::QFrameGraphNode * activeFrameGraph(void)
- QForwardRenderer * defaultFrameGraph(void)
- void registerAspect(Qt3DCore::QAbstractAspect *aspect)
- void registerAspect_2(QString name)
- Qt3DRender::QRenderSettings * renderSettings(void)
- void setActiveFrameGraph(Qt3DRender::QFrameGraphNode *activeFrameGraph)
- void setRootEntity(Qt3DCore::QEntity *root)
- Qt3DCamera *camera(void)

91.370 RingCodeHighlighter Class

Parameters : QTextDocument *parent

- void setColors(QColor c1,QColor c2,QColor c3,QColor c4,QColor c5)
- void setKeywordsBold(int nStatus)
- void setUseDefaultKeywords(int nStatus)
- void setCustomKeywords(QStringList aKeywords)

CHAPTER

NINETYTWO

USING FASTPRO

In this chapter we will learn about Using the FastPro extension.

This extension is added to the Ring language starting from Ring 1.19.

Contents:

- Bytes2List() function
- List2Bytes() function
- UpdateList() function
- UpdateColumn() function

92.1 Bytes2List() function

Syntax:

```
Bytes2List(cBytes,nWidth,nHeight,nChannels) -> aList // [[x,y,R,G,B,A],...]
```

92.2 List2Bytes() function

Syntax:

```
List2Bytes(aList,nChannels) -> cBytes // "RGBA...."
```

92.3 UpdateList() function

A function that could update 1D and 2D lists and provide very high performance.

This function support using numbers only.

Syntax:

```
updateList(<aList>,<cCommand>,<cSelection>,<nPara1>,[<nPara2>],[nPara3])
```

cCommand could be :set, :add, :sub, :mul, :div, :merge and :copy

cSelection could be :col, :row, :manycols, :manyrows and :items

Example:

```
updateList(<aList>,:set,:row,<nRow>,<nValue>)
updateList(<aList>,:set,:col,<nCol>,<nValue>)
updateList(<aList>,:set,:manyrows,<nRowStart>,<nRowEnd>,<nValue>)
updateList(<aList>,:set,:manycols,<nColStart>,<nColEnd>,<nValue>)
updateList(<aList>,:set,:items,<nValue>)
updateList(<aList>,:copy,:row,<nRowSrc>,<nRowDest>)
updateList(<aList>,:copy,:col,<nColSrc>,<nColDest>)
updateList(<aList>,:merge,:row,<nRowDest>,<nRow>)
updateList(<aList>,:merge,:col,<nColDest>,<nCol>)
```

The required parameters depend on the cCommand/cSelection.

The parameters could be columns/rows numbers.

Also, some commands requires a value like the set command.

Example:

```
load "fastpro.ring"

aList = [ [1,2,3],
          [4,5,6],
          7:9 ]

# Set the values of the first row to 10
        updateList(aList,:set,:row,1,10)
# Add 10 to each value in the first row
        updateList(aList,:add,:row,1,10)
# Sub 5 from each value in the first row
        updateList(aList,:sub,:row,1,5)
# Multiply each value in the first row by 10
        updateList(aList,:mul,:row,1,10)
# Divide each value in the first row by 2
        updateList(aList,:div,:row,1,2)
# Copy the first row values to the second row
        updateList(aList,:copy,:row,1,2)
# Sum the third row and the second row
# And the result will be in the third row
        updateList(aList,:merge,:row,3,2)

? aList
```

Output:

The list will be [[75,75,75], [75,75,75], [82,83,84]]

```
75
75
75
75
75
75
82
83
84
```

Using :col as the third parameter we can do operations on the list columns.

Example:

```
load "fastpro.ring"

# Store [ [1,2,3], [4,5,6], [7,8,9] ] in aList
aList = [ 1:3,
          4:6,
          7:9 ]

updateList(aList,:set,:col,1,100)
updateList(aList,:mul,:col,2,10)
updateList(aList,:div,:col,3,3)

? aList
```

Output:

The list will be [[100,20,1], [100,50,2], [100,80,3]]

```
100
20
1
100
50
2
100
80
3
```

Using :manyrows or :manycols we can do operations on many rows/columns

Example:

```
load "fastpro.ring"

aList = [
        1:3,
        4:6,
        7:9
]

# Starting from row 1 to row 2, set each value to 100
updateList(aList,:set,:manyrows,1,2,100)
# Starting from row 2 to row 3, multiply each value by 10
updateList(aList,:mul,:manyrows,2,3,10)

? aList
```

Output:

```
100
100
100
1000
```

(continues on next page)

(continued from previous page)

```
1000
1000
70
80
90
```

Using :items as the third parameter we can do operations on 1D lists.

Example:

```
load "fastpro.ring"

aList = 1:5

updateList(aList,:set,:items,1000)
updateList(aList,:mul,:items,2)
? aList
```

Output:

The list will be [2000,2000,2000,2000,2000]

```
2000
2000
2000
2000
2000
```

92.4 UpdateColumn() function

Syntax:

```
updateColumn(<aList>, [<cCommand>,<nPara1>,[<nPara2>],[nPara3]],...)
```

Using this function we can execute many commands on the list columns.

Instead of using updateList() many times and each time we pass :col as the third parameter, we can use updateColumn().

This function support using numbers only.

Example from the ImagePixel application that convert the image to Gray.

```
    //=====================================================================
    // GRAY SCALE -- Display Color RBG in GRAY Scale
    // Average looks better brighter than Gamma Corrected
    // Color corrected is for eye sensitivity Red 30%, Green 59% Blue 11%.

    elseif lGray

        updateColumn(MCOrig,:mul,RVALUE,0.3,            # R *= 0.3
                            :mul,GVALUE,0.59,           # G *= 0.59
                            :mul,BVALUE,0.11,           # B *= 0.11
                            :merge,RVALUE,GVALUE,       # R += G
                            :merge,RVALUE,BVALUE,       # R += B
                            :copy,RVALUE,GVALUE,        # G = R
                            :copy,RVALUE,BVALUE)        # B = R

    ok

```

92.5 Generate Image

The next example generate an image using a string that contain bytes

Then uses Bytes2List() to convert the generated image to Ring List

Using updateColumn() the list is updated and the colors are converted to Gray

Then using List2Bytes() we get another string contains bytes that represent the Gray Image

Then using the drawBytes() method in RingQt - QPainter class we draw the generated images

Note: We can update the string bytes directly without conversion to a Ring list

Tip: Converting the string to a Ring list enable us to use the updateColumn() function

```
load "stbimage.ring"
load "fastpro.ring"
load "lightguilib.ring"

width    = 640
height   = 480
channels = 3
cImage   = space(width*height*channels)

RVALUE   = 3
GVALUE   = 4
BVALUE   = 5

WindowWidth = Width*2 + 100

nIndex=0
for x=1 to height
        for y=1 to width
```

(continues on next page)

(continued from previous page)

```
                cImage[nIndex++] = x*x
                cImage[nIndex++] = x*y
                cImage[nIndex++] = x*2
        next
next

aList = Bytes2List(cImage,Width,Height,Channels,255)
updateColumn(aList,:mul,RVALUE,0.3,              # R *= 0.3
                   :mul,GVALUE,0.59,            # G *= 0.59
                   :mul,BVALUE,0.11,            # B *= 0.11
                   :merge,RVALUE,GVALUE,        # R += G
                   :merge,RVALUE,BVALUE,        # R += B
                   :copy,RVALUE,GVALUE,         # G = R
                   :copy,RVALUE,BVALUE)         # B = R
cGrayImage = list2Bytes(aList,Channels,255)

MyApp = new QApp
{
        win1 = new QWidget()
        {
                setwindowtitle("Generate Image & Convert it to Gray")
                setgeometry(0,0,WindowWidth,Height)
                Canvas = new QLabel(win1)
                {
                        MonaLisa  = new QPixMap2( WindowWidth, Height)
                        daVinci = new QPainter()
                        {
                                begin(MonaLisa)
                                drawbytes(0,0,cImage,width,Height,channels)
                                drawbytes(width+10,0,cGrayImage,width,Height,channels)
                                endpaint()
                        }
                        setPixMap(MonaLisa)
                }
                show()
        }
        exec()
}
```

Output:

CHAPTER

NINETYTHREE

USING REFERENCES

In this chapter we will learn about using references.

This feature is added to the Ring language starting from Ring 1.18.

Contents:

- Introduction
- Ref()/Reference() function
- Refcount() function
- Circular references
- The Tree class
- Linked list
- Dependency injection
- Passing lists to functions
- Ref() and temp. lists
- Nested Ref()
- Ref() in left side
- Ref() and sub lists

93.1 Introduction

In Ring, Using the Assignment (=) operator copy variables by value. Also, Adding a List/Object to another List create a new copy. To change this behavior, We can use the Ref()/Reference() function. This function doesn't create the reference directly. It's a flag setter, And Ring VM will decide when to create the reference.

Once we have more than one reference, No need to use the Ref() again with the same list/object Because Ring will use Copy by Reference with this list/object. If the reference count drop to one again then Ring will change the behavior and will use the default rule (copy lists/objects by value).

In general Ring as a language is designed to reduce references usage. This feature is added to be used in special cases like teaching data structures and implementing specific design patterns.

93.2 ref()/reference() function

Syntax:

```
ref(aList|oObject)       ---> aList|oObject (Enable Reference Flag) # Short name
reference(aList|oObject) ---> aList|oObject (Enable Reference Flag) # Long name
```

Example:

```
aList     = [1,2,3]
aListCopy = aList       # Copy by Value
aList2    = ref(aList)  # Copy by Reference  (RC: 2)
aList3    = aList       # Copy by Reference  (RC: 3)
for item in aList
        item *= 10
next
? aList3                # 10 20 30
? aListCopy             # 1 2 3
aList2    = NULL
aList3    = NULL
aList2    = aList       # Copy by Value
for item in aList2
        item /= 10
next
? aList                 # 10 20 30
? aList2                # 1  2  3
```

Output:

```
10
20
30

1
2
3

10
20
30

1
2
3
```

93.3 refcount() function

Using the refcount() function we can know how many references exist.

Syntax:

```
refcount(variable) ---> Number (References Count)
```

Example:

```
aList  = 1:10
aList2 = ref(aList)
? refcount(aList)       # 2
? refcount(aList2)      # 2
aList3 = aList2
aList4 = aList
aList5 = aList4
? refcount(aList)       # 5
? refcount(aList2)      # 5
? refcount(aList3)      # 5
? refcount(aList4)      # 5
? refcount(aList5)      # 5
aList5 = NULL
aList4 = []
aList3 = [10]
? refcount(aList)       # 2
? refcount(aList2)      # 2
```

Output:

```
2
2
5
5
5
5
5
2
2
```

93.4 Circular References

Using Ref() we can create circular references

Ring VM can detect them and free memory when the variable is deleted.

Example:

```
aList = [ 10,20,30, ref(aList) ]
? aList[4][1]
? aList[4][4][4][4][4][2]
? refcount(aList)
```

Output:

```
10
20
2
```

93.5 The Tree Class

The Tree class is a good example about using the Ref() function.

In this class each object contains a group of objects and these objects have a reference to the parent object.

Example:

```
Class tree

        data parent
        children = []

        func set x
                data = x

        func value
                return data

        func add x
                children + new tree
                nMax = len(children)
                children[nMax].parent = ref(self)
                children[nMax].data = x
                return children[nMax]

        func parent
                if ! isObject(parent)
                        raise("This node is the root!")
                        return
                ok
                return parent

        func print
                for x in children
                        ? x.data
                        x.print()
                next
```

Tip: The Tree class already exist in the StdLib

93.6 Linked list

The next example demonstrates how to create a linked list using the Ref() function.

Tip: In practice we don't need to do this since Ring comes with lists

Example:

```
func main

        n1 = new node(:one)
        n2 = new node(:two)
        n3 = new node(:three)
        n4 = new node(:four)
        n5 = new node(:five)

        n1 { pNext = ref(n2) "No Previous Node" }
        n2 { pNext = ref(n3) pPrev = ref(n1)    }
        n3 { pNext = ref(n4) pPrev = ref(n2)    }
        n4 { pNext = ref(n5) pPrev = ref(n3)    }
        n5 { "No Next Node"  pPrev = ref(n4)    }

        n3 {
                toTheEnd()
                ? Copy("=",20)
                toTheStart()
        }

class node

        pPrev pNext

        func init value
                data = value

        func toTheEnd

                print()
                pCurrent = pNext
                while isObject(pCurrent)
                        pCurrent.print()
                        pCurrent = pCurrent.pNext
                end

        func toTheStart

                print()
                pCurrent = pPrev
                while isObject(pCurrent)
                        pCurrent.print()
                        pCurrent = pCurrent.pPrev
                end
```

(continues on next page)

(continued from previous page)

```
        func print

                ? data

        private

                data  pCurrent
```

Output:

```
three
four
five
====================
three
two
one
```

93.7 Dependency injection

The next example demonstrates how to apply dependency injection using the Ref() function

Example:

```
func main

        v1 = new MyClass1  v2 = new MyClass2

        oCont = new Controller(v1,v2)

        v1.value = "one"  v2.value = "two"

        oCont.test()

        v1 = 10 v2 = 20  ? v1 ? v2

        oCont.test()

class MyClass1 value func test ? :myclass1

class MyClass2 value func test ? :myclass2

class Controller o1 o2

        func init myo1,myo2

                o1=ref(myo1) o2=ref(myo2)

        func test
```

(continues on next page)

(continued from previous page)

```
                o1.test()   o2.test()
                ? o1.value  ? o2.value
```

Output:

```
myclass1
myclass2
one
two
10
20
myclass1
myclass2
one
two
```

93.8 Passing lists to functions

In Ring, when we pass a list/object to a function, This function will have full ownership on the list/object variable.

This means it can delete it and change the variable type.

Example:

```
func main
        ? "Hello from Main() function"
        aList = [1,2,3]
        ? aList
        sub(aList)
        ? "Hello from Main() function (Again)"
        if ! isList(aList)
                ? "We don't have a list!"
        ok

func sub aList
        ? "Hello from Sub() function"
        aList = NULL
```

Output:

```
Hello from Main() function
1
2
3

Hello from Sub() function
Hello from Main() function (Again)
We don't have a list!
```

Using Ref() function we can change this behavior and pass a reference to the list/object instead of sharing it.

Example:

```
func main
        ? "Hello from Main() function"
        aList = [1,2,3]
        ? aList
        sub(Ref(aList))
        ? "Hello from Main() function (Again)"
        if ! isList(aList)
                ? "We don't have a list!"
        else
                ? "We still have our list!"
                ? aList
        ok

func sub aList
        ? "Hello from Sub() function"
        aList = NULL
```

Output:

```
Hello from Main() function
1
2
3

Hello from Sub() function
Hello from Main() function (Again)
We still have our list!
1
2
3
```

93.9 Ref() and temp. lists

Using Ref() and temp. lists/objects does nothing.

Example:

```
aList = ref([1,2,3])     # The same as aList = [1,2,3]
? refcount(aList)        # 1
aList = ref(1:10)        # The same as aList = 1:10
? refcount(aList)        # 1
aList = ref(list(10))    # The same as aList = list(10)
? refcount(aList)        # 1
myobj = ref(new point)   # The same as myobj = new point
? refcount(myobj)        # 1
class point x y z
```

93.10 Nested Ref()

Since Ref() function is a flag setter, nested Ref() usage is not useful.

Example:

```
aList  = [1,2,3]
aList2 = ref(ref(ref(ref(aList))))  # The same as aList2 = ref(aList)
? refcount(aList)                   # 2
```

93.11 Ref() in left side

Using Ref() in left side of an assignment is not useful and will disable the assignment i.e. will not change the value.

Example:

```
aList = [1,2,3]
ref(aList) = [4,5]
? aList
```

Output:

```
1
2
3
```

93.12 Ref() and sub lists

Using Ref() with a sub list will create a strong reference to this sub list

Tip: if the sub list contains other references, we will get weak references to them.

Example:

```
a = [[ref(a),ref(a),3],[4,5,6]]
? a
b = ref(a[1])   # Get Weak references to References inside a[1]
? b
b = NULL
? a
c = a[1]        # Get Strong references to References inside a[1]
? c
c = NULL
? a
? :done
```

Output:

```
[...] (RC:3)
[...] (RC:3)
3
4
5
6

[...] (RC:3)
[...] (RC:3)
3

[...] (RC:3)
[...] (RC:3)
3
4
5
6

[...] (RC:5)
[...] (RC:5)
3

[...] (RC:3)
[...] (RC:3)
3
4
5
6

done
```

CHAPTER

NINETYFOUR

LOW LEVEL FUNCTIONS

In this chapter we will learn about the low level functions provided by Ring

It's not recommended to use these functions in your application code

These functions exist for C/C++ developers who are developing Ring libraries/tools

We expect from those developers to know about pointers and dynamic memory management

```
* callgarbagecollector()| callgc()
* variablepointer()     | varptr()
* space()
* nullpointer()         | nullptr()
* object2pointer()      | obj2ptr()
* pointer2object()      | ptr2obj()
* ispointer()
* pointercompare()      | ptrcmp()
* setpointer()          | setptr()
* getpointer()          | getptr()
* pointer2string()      | ptr2str()
* memorycopy()          | memcpy()
* ringvm_cfunctionslist()
* ringvm_functionslist()
* ringvm_classeslist()
* ringvm_packageslist()
* ringvm_memorylist()
* ringvm_calllist()
* ringvm_fileslist()
* ringvm_settrace()
* ringvm_tracedata()
* ringvm_traceevent()
* ringvm_tracefunc()
* ringvm_scopescount()
* ringvm_evalinscope()
* ringvm_passerror()
* ringvm_hideerrorMsg()
* ringvm_callfunc()
* ringvm_see()
* ringvm_give()
* ringvm_info()
* ringvm_ismempool()
* ringvm_codelist()
```

94.1 callgc() function

Syntax:

```
callgc()                        # Short name
callgarbagecollector()          # Long name
```

Use this function to force calling the garbage collector during function execution when you use a loop that create temp. variables that you don't free using the assignment operation.

It's very rare to need this function but it's useful when you create something like event-loop for your game engine and start creating lists on the fly when you call functions.

Example

```
While True

        # process events
        # call functions using temp. lists like myfunc(["temp list"])

        # call the garbage collector
        callgc()
End
```

Tip: In Ring the garbage collector works automatically in the end of function execution or when you use the assignment statement.

94.2 varptr() function

Use the varptr() function when you need to pass a pointer to a C/C++ function.

Syntax:

```
varptr(cVariableName,cPointerType) ---> Low Level Object (C Pointer)
variablepointer(cVariableName,cPointerType) ---> Low Level Object (C Pointer)
```

example:

```
r = 10
z = 20
see r + nl
see varptr("r","int")
see varptr("z","int")
```

Output:

```
10
00E3C740
int
2
00E3BEC0
```

(continues on next page)

(continued from previous page)

```
int
2
```

Note: the low level object is a list contains three items (The Pointer, The Type, The Status)

94.3 space() function

Use the space function to allocate a specific number of bytes in Memory.

Syntax:

```
Space(nBytesCount) ---> String
```

Example:

```
mystring = space(200)
See "String Size : " + len(mystring) + nl
See "String : " + mystring + nl
See "String Pointer : "
See varptr("mystring",:char)
```

Output:

```
String Size : 200
String :
String Pointer : 00FF8FE8
char
2
```

Note: You may need the space() and VarPtr() functions to pass buffers to C functions.

Tip: To free the memory allocated using the space() function, use the Assignment operator

```
mystring = space(1000)  # Allocate memory (1000 bytes)
mystring = NULL         # Free memory stored in mystring
```

Note: We don't need to free the memory if it's a local variable that will be deleted after the function execution.

94.4 nullpointer() function

Syntax:

```
nullptr()                # Short name
nullpointer()            # Long name
```

You may need to pass the NULL pointer to a C function that may expect a pointer as parameter and accept NULL pointers for optional parameters.

Example:

The next example uses the SDL_BlitSurface() function from the LibSDL Library through RingSDL The function accept SDL_Rect pointers in the second and the last parameter. Also the function accept NULL pointers, so we can pass them using the NULLPointer() Function.

```
SDL_BlitSurface(text, nullpointer(), surface, nullpointer())
```

Note: The previous code doesn't work alone, you need to learn how to use RingSDL first.

Tip: We can pass NULL as parameter instead of using the NULLPointer() function

```
SDL_BlitSurface(text, NULL, surface, NULL)
```

94.5 object2pointer() function

Use this function to get a C pointer for Ring lists and objects

Syntax:

```
obj2ptr(List|Object) --> Low Level Object ( C Pointer )         # Short name
object2pointer(List|Object) --> Low Level Object ( C Pointer )  # Long name
```

Note: You have to be sure that the Pointer still valid (Doesn't point to deallocated memory)

94.6 pointer2object() function

Use this function to get the Ring list and/or object from the low level object (C Pointer)

Syntax:

```
ptr2obj(Low Level Object) ---> ListReference|ObjectReference            # Short name
pointer2object(Low Level Object) ---> ListReference|ObjectReference     # Long name
```

Note: Before using Pointer2Object() to create a reference, Be sure that the Pointer is valid (Doesn't point to deallocated memory, i.e. deleted list/object)

Tip: After using Pointer2Object() to create a reference, and if the original List/Object is deleted, Ring will uses reference counting and will keep the List/Object until the latest reference is removed

Example:

```
# Create the list
mylist = 1:5

# Create pointer to the list
x = object2pointer(mylist)
see x

see nl

# Add items to the list
mylist + "welcome"

# Get a copy from the list
y = pointer2object(x)
# print the new list items
see y
```

Output:

```
0069A5D8
OBJECTPOINTER
0

1
2
3
4
5
welcome
```

Note: In Ring the assignment operator copy lists and objects by value, to copy lists and objects by reference Just use the object2pointer() and pointer2object() functions.

The functions Object2Pointer() and Pointer2Object() are low level functions

We have to be careful when using them to avoid memory problems

If we created a Pointer to a (Local Variable)

This local variable will be deleted from the memory after the end of the function/method execution

This means that the pointer created with Object2Pointer() will becomes a dangling pointer

i.e. A pointer that points to the memory location of the deallocated memory

Using this invalid pointer could lead to (CRASH or Memory Corruption).

If you will use pointers (Using Object2Pointer() or Pointer2Object()) then never use pointers that point to the memory that are deallocated.

In simple words, Keep the memory (Don't delete it if you still need it)

i.e. instead of using (Local Variables) that will be deleted, You can use Class Attributes or Global Variables.

94.7 ispointer() function

Check if the parameter is a pointer (C Object) or not.

Syntax:

```
IsPointer(vPara) ---> True|False        # Long name
```

Example :

```
fp = fopen(filename(),"r")

? type(fp)

? ispointer(fp)
```

Output :

```
file
1
```

94.8 ptrcmp() function

We can compare between two pointers (C Objects) using the ptrcmp() function.

Syntax:

```
ptrcmp(oObject1,oObject2) ---> value = 1 if oObject1 = oObject2
                               value = 0 if oObject1 != oObject2
pointercompare(oObject1,oObject2) ---> value = 1 if oObject1 = oObject2
                                       value = 0 if oObject1 != oObject2
```

Example:

```
fp = fopen("ptrcmp.ring","r")
fp2 = fp
fp3 = fopen("ptrcmp.ring","r")

see ptrcmp(fp,fp2) + nl
see ptrcmp(fp,fp3) + nl

fclose(fp)
fclose(fp3)
```

Output:

```
1
0
```

94.9 setpointer() function

Set the pointer address to another address

Syntax:

```
setptr(pointer,nNewAddress)             # Short name
setpointer(pointer,nNewAddress)         # Long name
```

Note: Using setPointer() and getPointer() functions we can change the Memory Address

94.10 getpointer() function

Get the pointer address

Syntax:

```
getptr(pointer) ---> nAddress           # Short name
getpointer(pointer) ---> nAddress       # Long name
```

Example:

```
? "Sample about using setPointer() and getPointer() functions"
? copy("=",50)
pointer = NULLPOINTER()
? pointer
? "Type: " + type(pointer)
? "Address: " + Upper(hex(getpointer(pointer)))
? copy("=",50)
name = "ring"
pointer = varptr(:name,:char)
? pointer
? "Type: " + type(pointer)
? "Address: " + Upper(hex(getpointer(pointer)))
? copy("=",50)
setpointer(pointer, getpointer(pointer) + 1 )
? "After Update"
? "Address: " + Upper(hex(getpointer(pointer)))
? copy("=",50)
```

Output:

```
==================================================
00000000
NULLPOINTER
0
```

(continues on next page)

(continued from previous page)

```
Type: NULLPOINTER
Address: 0
==================================================
026E2BA8
char
0

Type: char
Address: 26E2BA8
==================================================
After Update
Address: 26E2BA9
==================================================
```

94.11 pointer2string() function

Convert a pointer to a string of binary data

If you want to convert the string to a pointer again use VarPtr() function

Syntax:

```
ptr2str(pointer,nStart,nCount) ---> cString             # Short name
pointer2string(pointer,nStart,nCount) ---> cString      # Long name
```

Note: pointer2String() return another copy of the data

Note: if nStart is Zero, this means starting from the first character

Example:

```
name = "ring"
pointer = varptr(:name,:char)
? pointer
? "Type: " + type(pointer)
? "Address: " + Upper(hex(getpointer(pointer)))

? "Get 4 bytes starting from the pointer address"
mystring = Pointer2String(pointer,0,4)
? mystring

? "Get 2 bytes starting from the pointer address + 1"
mystring2 = Pointer2String(pointer,1,2)
? mystring2
```

Output:

```
01E03380
char
0

Type: char
Address: 1E03380
Get 4 bytes starting from the pointer address
ring
Get 2 bytes starting from the pointer address + 1
in
```

94.12 memcpy() function

Syntax:

```
memcpy(pDestinationPointer,cSourceString,nSize)         # Short name
memorycopy(pDestinationPointer,cSourceString,nSize)     # Long name
```

Example:

```
str = space(9)
pointer = varptr(:str,"char")
memcpy(pointer,"one",3)
? str
setPointer(pointer,getPointer(pointer)+3)
memcpy(pointer,"one",3)
? str
setPointer(pointer,getPointer(pointer)+3)
memcpy(pointer,"one",3)
? str
```

Output:

```
one
oneone
oneoneone
```

94.13 ringvm_cfunctionslist() function

The Function return a list of functions written in C.

Syntax:

```
RingVM_CFunctionsList() ---> List
```

Example:

```
See RingVM_CFunctionsList()
```

94.14 ringvm_functionslist() function

The Function return a list of functions written in Ring.

Each List Member is a list contains the next items

- Function Name
- Program Counter (PC) - Function Position in Byte Code.
- Source Code File Name
- Private Flag (For Private Methods in Classes)

Syntax:

```
RingVM_FunctionsList() ---> List
```

Example:

```
test()

func test
        see ringvm_functionslist()
```

Output:

```
test
8
B:/ring/tests/scripts/functionslist.ring
0
```

94.15 ringvm_classeslist() function

The Function return a list of Classes.

Each List Member is a list contains the next items

- Class Name
- Program Counter (PC) - Class Position in Byte Code.
- Parent Class Name
- Methods List
- Flag (Is parent class information collected)
- Pointer to the package (or NULL if no package is used)

Syntax:

```
RingVM_ClassesList() ---> List
```

Example:

```
see ringvm_classeslist()

class class1
        func f1
class class2 from class1
class class3 from class1
```

Output:

```
class1
9

f1
13
B:/ring/tests/scripts/classeslist.ring
0
0
00000000
class2
16
class1
0
00000000
class3
20
class1
0
00000000
```

94.16 ringvm_packageslist() function

The Function return a list of Packages.

Each List Member is a list contains the next items

- Package Name
- Classes List

Syntax:

```
RingVM_PackagesList() ---> List
```

Example:

```
see ringvm_packageslist()

package package1
        class class1

package package2
        class class1
```

(continues on next page)

(continued from previous page)

```
package package3
        class class1
```

Output:

```
package1
class1
11

0
00FEF838
package2
class1
17

0
00FEF978
package3
class1
23

0
00FEFF68
```

94.17 ringvm_memorylist() function

The Function return a list of Memory Scopes and Variables.

Each List Member is a list contains variables in a different scope.

Each Item in the scope list is a list contains the next items

- Variable Name
- Variable Type
- Variable Value
- Pointer Type (List/Item) if the value is a list
- Private Flag (if the variable is an attribute in a Class)

Syntax:

```
RingVM_MemoryList() ---> List
```

Example:

```
x = 10
test()
func test
        y = 20
        see ringvm_memorylist()
```

Output:

```
true
2
1
0
0
false
2
0
0
0
nl
1


0
0
null
1

0
0
ring_gettemp_var
4
00000000
0
0
ccatcherror
1
NULL
0
0
ring_settemp_var
4
00000000
0
0
ring_tempflag_var
2
0
0
0
stdin
3
50512DB8
file
0
0
0
stdout
3
50512DD8
file
0
```

(continues on next page)

(continued from previous page)

```
0
0
stderr
3
50512DF8
file
0
0
0
this
4
00000000
0
0
sysargv
3
B:\ring\bin/ring
B:/ring/tests/scripts/memorylist.ring
0
0
x
2
10
0
0
y
2
20
0
0
```

94.18 ringvm_calllist() function

The Function return a list of the functions call list.

Each List Member is a list contains the next items

- Function Type
- Function Name
- Program Counter (PC)
- Stack Pointer (SP)
- Method or Function Flag
- Caller PC

Syntax:

```
RingVM_CallList() ---> List
```

Example:

```
hello()
func hello
        test()

func test
        mylist = ringvm_calllist()
        for t in mylist see t[2] + nl next
```

Output:

```
function hello() in file B:/ring/tests/scripts/calllist.ring
called from line 1
function test() in file B:/ring/tests/scripts/calllist.ring
called from line 3
ringvm_calllist
```

94.19 ringvm_fileslist() function

Function return a list of the Ring Files.

Syntax:

```
RingVM_FilesList() ---> List
```

Example:

```
load "stdlib.ring"
see ringvm_fileslist()
```

Output:

```
B:/ring/tests/scripts/fileslist.ring
B:\ring\bin\stdlib.ring
eval
stdlib.ring
stdlib.rh
stdclasses.ring
stdfunctions.ring
stdbase.ring
stdstring.ring
stdlist.ring
stdstack.ring
stdqueue.ring
stdmath.ring
stddatetime.ring
stdfile.ring
stdsystem.ring
stddebug.ring
stddatatype.ring
stdconversion.ring
stdodbc.ring
stdmysql.ring
```

(continues on next page)

(continued from previous page)

```
stdsecurity.ring
stdinternet.ring
stdhashtable.ring
stdtree.ring
```

94.20 ringvm_settrace()

The function ringvm_settrace() determine the Trace function name

The trace function is a Ring function that will be called for each event

Syntax:

```
RingVM_SetTrace(cCode)
```

94.21 ringvm_tracedata()

Inside the function that we will use for tracing events

We can use the ringvm_tracedata() function to get the event data.

The event data is a list contains the next items

- The Source Code Line Number
- The Source File Name
- The Function/Method Name
- Method or Function (Bool : True=Method, False=Function/File)

Syntax:

```
RingVM_TraceData() ---> aDataList
```

94.22 ringvm_traceevent()

Inside the function that we will use for tracing events

We can use ringvm_traceevent() to know the event type

- New Line
- Before Function
- After Function
- Runtime Error
- Before C Function
- After C Function

Syntax:

```
RingVM_TraceEvent() ---> nTraceEvent
```

94.23 ringvm_tracefunc()

The function return the name of the function that we are using for tracing events.

Syntax:

```
RingVM_TraceEvent() ---> cCode
```

94.24 ringvm_scopescount()

We can use the RingVM_ScopesCount() function to know the number of scopes used in the application.

In the start of the program, We have the (global scope only)

When we call a function, A new scope is created.

When the function execution is done, the function scope is deleted.

Syntax:

```
RingVM_ScopesCount() ---> nScopes
```

94.25 ringvm_evalinscope()

The function ringvm_evalinscope() is similar to the eval() function

Unlike eval() which execute the code in the current scope

Using RingVM_EvalInScope() we can execute the code in a specific scope.

Syntax:

```
RingVM_EvalInScope(nScope,cCode)
```

94.26 ringvm_passerror()

When we have runtime error, After printing the Error message, Ring will end the execution of the program.

Using ringvm_passerror() we can avoid that, and continue the execution of our program.

Syntax:

```
RingVM_PassError()
```

94.27 ringvm_hideerrormsg()

We can disable/enable displaying the runtime error messages using the RingVM_HideErrorMsg() function.

Syntax:

```
RingVM_HideErrorMsg(lStatus)
```

94.28 ringvm_callfunc()

We can call a function from a string without using eval() using the ringvm_callfunc()

Syntax:

```
RingVM_CallFunc(cFuncName)
```

94.29 Example - Using the Trace Functions

The next example use the Trace Functions to trace the program Events!

In practical, We will use the Trace Library instead of these low level functions!

```
load "tracelib.ring"

ringvm_settrace("mytrace()")

see "Hello, world!" + nl
see "Welcome" + nl
see "How are you?" +nl
mytest()
new myclass { mymethod() }

func mytest
        see "Message from mytest" + nl

func mytrace
        see "====== The Trace function is Active ======" + nl +
                "Trace Function Name : " + ringvm_TraceFunc() + nl +
                "Trace Event : "
        switch ringvm_TraceEvent()
                on TRACEEVENT_NEWLINE           see "New Line"
                on TRACEEVENT_NEWFUNC           see "New Function"
                on TRACEEVENT_RETURN            see "Return"
                on TRACEEVENT_ERROR             see "Error"
                on TRACEEVENT_BEFORECFUNC       see "Before C Function"
                on TRACEEVENT_AFTERCFUNC        see "After C Function"
        off
        see nl +
                "Line Number : " + ringvm_tracedata()[TRACEDATA_LINENUMBER] + nl +
                "File Name   : " + ringvm_tracedata()[TRACEDATA_FILENAME] + nl +
```

(continues on next page)

(continued from previous page)

```
                "Function Name : " + ringvm_tracedata()[TRACEDATA_FUNCNAME] + nl +
                "Method or Function : "
                if ringvm_tracedata()[TRACEDATA_METHODORFUNC] =
                                TRACEDATA_METHODORFUNC_METHOD
                        see "Method"
                else
                        if ringvm_tracedata()[TRACEDATA_FUNCNAME] = NULL
                                see "Command"
                        else
                                see "Function"
                        ok
                ok
                see nl + Copy("=",42) + nl

class myclass
        func mymethod
                see "Message from mymethod" + nl
```

Output:

```
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 3
File Name   : test1.ring
Function Name : ringvm_settrace
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 5
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
Hello, world!
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 6
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
Welcome
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 7
File Name   : test1.ring
Function Name :
```

(continues on next page)

(continued from previous page)

```
Method or Function : Command
==========================================
How are you?
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 8
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 8
File Name   : test1.ring
Function Name : mytest
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 12
File Name   : test1.ring
Function Name : mytest
Method or Function : Function
==========================================
Message from mytest
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 14
File Name   : test1.ring
Function Name : mytest
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 8
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 9
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
```

(continues on next page)

(continued from previous page)

```
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 43
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 9
File Name   : test1.ring
Function Name : mymethod
Method or Function : Method
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 44
File Name   : test1.ring
Function Name : mymethod
Method or Function : Method
==========================================
Message from mymethod
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 9
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Before C Function
```

(continues on next page)

(continued from previous page)

```
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 11
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
```

94.30 Example - The Trace Library

The next example uses the Trace functions provided by the Ring language to create the Trace library.

Using the Trace library we have nice Tracing tools and Interaction debugger too.

```
# Trace Events
TRACEEVENT_NEWLINE      = 1
TRACEEVENT_NEWFUNC      = 2
TRACEEVENT_RETURN       = 3
TRACEEVENT_ERROR        = 4
TRACEEVENT_BEFORECFUNC  = 5
TRACEEVENT_AFTERCFUNC   = 6
```

(continues on next page)

(continued from previous page)

```
# Trace Data
TRACEDATA_LINENUMBER    = 1
TRACEDATA_FILENAME      = 2
TRACEDATA_FUNCNAME      = 3
TRACEDATA_METHODORFUNC  = 4

# Method of Function
TRACEDATA_METHODORFUNC_METHOD           = TRUE
TRACEDATA_METHODORFUNC_NOTMETHOD        = FALSE

TRACE_BREAKPOINTS = TRUE

TRACE_TEMPLIST = []

func Trace cType
        switch trim(lower(cType))
        on :AllEvents
                ringvm_settrace("TraceLib_AllEvents()")
        on :Functions
                ringvm_settrace("TraceLib_Functions()")
        on :PassError
                ringvm_settrace("TraceLib_PassError()")
        on :Debugger
                ringvm_settrace("TraceLib_Debugger()")
        on :LineByLine
                ringvm_settrace("TraceLib_LineByLine()")
        off

func TraceLib_AllEvents
        if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
                return
        ok
        see "====== The Trace function is Active ======" + nl +
                "Trace Function Name : " + ringvm_TraceFunc() + nl +
                "Trace Event : "
        switch ringvm_TraceEvent()
                on TRACEEVENT_NEWLINE           see "New Line"
                on TRACEEVENT_NEWFUNC           see "New Function"
                on TRACEEVENT_RETURN            see "Return"
                on TRACEEVENT_ERROR             see "Error"
                on TRACEEVENT_BEFORECFUNC       see "Before C Function"
                on TRACEEVENT_AFTERCFUNC        see "After C Function"
        off
        see nl +
                "Line Number : " + ringvm_tracedata()[TRACEDATA_LINENUMBER] + nl +
                "File Name   : " + ringvm_tracedata()[TRACEDATA_FILENAME] + nl +
                "Function Name : " + ringvm_tracedata()[TRACEDATA_FUNCNAME] + nl +
                "Method or Function : "
                if ringvm_tracedata()[TRACEDATA_METHODORFUNC] =
                                TRACEDATA_METHODORFUNC_METHOD
                        see "Method"
```

(continues on next page)

(continued from previous page)

```
                else
                        if ringvm_tracedata()[TRACEDATA_FUNCNAME] = NULL
                                see "Command"
                        else
                                see "Function"
                        ok
                ok
                see nl + Copy("=",42) + nl

func TraceLib_Functions
        if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
                return
        ok
        switch ringvm_TraceEvent()
                on TRACEEVENT_NEWFUNC
                        see "Open Func : " +
                        ringvm_TraceData()[TRACEDATA_FUNCNAME] + nl
                on TRACEEVENT_RETURN
                        see "Return to Func : " +
                        ringvm_TraceData()[TRACEDATA_FUNCNAME] + nl
        off

func TraceLib_PassError
        if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
                return
        ok
        switch ringvm_TraceEvent()
                on  TRACEEVENT_ERROR
                        see nl
                        see "TraceLib : After Error !" + nl
                        ringvm_passerror()
        off

func TraceLib_Debugger
        if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring"
                return
        ok
        switch ringvm_TraceEvent()
                on  TRACEEVENT_ERROR
                        _BreakPoint()
        off

func TraceLib_LineByLine
        if right(ringvm_tracedata()[TRACEDATA_FILENAME],13) = "tracelib.ring" or
                ringvm_TraceEvent() != TRACEEVENT_NEWLINE
                return
        ok
        aList = ringvm_tracedata()
        see "Before Line : " + aList[TRACEDATA_LINENUMBER] + nl
        _BreakPoint()

func BreakPoint
```

(continues on next page)

(continued from previous page)

```
        if not TRACE_BREAKPOINTS
                return
        ok
        _BreakPoint()

func _BreakPoint
        see nl+nl+Copy("=",60) + nl +
        Copy(" ",20)+"Interactive Debugger" + nl +
        Copy("=",60) + nl +
        "Command (Exit)        : End Program" + nl +
        "Command (Cont)        : Continue Execution" + nl +
        "Command (Locals)      : Print local variables names" + nl +
        "Command (LocalsData)  : Print local variables data" + nl +
        "Command (Globals)     : Print global variables names" + nl +
        "We can execute Ring code" + nl +
        Copy("=",60) + nl
        while true
                        see nl + "code:> "
                        give cCode
                cmd = trim(lower(cCode))
                if cmd = "exit" or cmd = "bye"
                        shutdown()
                ok
                nScope = ringvm_scopescount()-2
                switch cmd
                        on "locals"
                                ringvm_EvalInScope(nScope,"see locals() callgc()")
                                loop
                        on "localsdata"
                                PrintLocalsData(nScope)
                                loop
                        on "globals"
                                ringvm_EvalInScope(nScope,"see globals() callgc()")
                                loop
                        on "cont"
                                ringvm_passerror()
                                exit
                off
                Try
                        ringvm_EvalInScope(nScope,cCode)
                        catch
                                        see cCatchError
                        done
        end

func NoBreakPoints
        TRACE_BREAKPOINTS = FALSE


func PrintLocalsData nScope
        if nScope = 1   # Global
                ringvm_Evalinscope(nScope,'TRACE_TEMPLIST = globals()')
```

(continues on next page)

(continued from previous page)

```
        else
                ringvm_Evalinscope(nScope,'TRACE_TEMPLIST = locals() callgc()')
        ok
        see nl
        aTempList = TRACE_TEMPLIST
        TRACE_TEMPLIST = []
        nSpaces = 5
        for TRACE_ITEM in aTempList
                if len(TRACE_ITEM) + 5 > nSpaces
                        nSpaces = len(TRACE_ITEM) + 5
                ok
        next
        for TRACE_ITEM in aTempList
                see "Variable : " +  TRACE_ITEM
                cVarName = TRACE_ITEM
                see copy(" ",nSpaces-len(cVarName)) + " Type : "
                ringvm_Evalinscope(nScope,"see type(" +  TRACE_ITEM +")")
                ringvm_Evalinscope(nScope,"see Copy(' ',fabs(15-len(type(" +
                                        TRACE_ITEM +"))))")
                see " Value : "
                ringvm_Evalinscope(nScope,"see " +  TRACE_ITEM)
                see nl
        next
```

94.31 ringvm_see() function

Using the ringvm_see() function we can redefine the behavior of the See command

Also we can use ring_see() to have the original behavior

Example:

```
see "Hello world" + nl
see 123 + nl
see ["one","two","three"]
see new point {x=10 y=20 z=30}

func ringvm_see t
        ring_see("We want to print: ")
        ring_See(t)

class point x y z
```

Output:

```
We want to print: Hello world
We want to print: 123
We want to print: one
two
three
We want to print: x: 10.000000
```

(continues on next page)

(continued from previous page)

```
y: 20.000000
z: 30.000000
```

94.32 ringvm_give() function

Using the ringvm_give() function we can redefine the behavior of the Give command

Example:

```
see "Name: " give name
see "Hello " + name

func ringvm_give
        see "Mahmoud" + nl
        return "Mahmoud"
```

Output:

```
Name: Mahmoud
Hello Mahmoud
```

94.33 ringvm_codelist() function

The Function return a list contains the Byte Code of the current program.

Each item is a sub list that represent an instruction

This sub list starts with the operation code (A Number) then the parameters

94.34 ringvm_info() function

The ringvm_info() is an internal function that return a list of information about the Ring VM structure.

It's used only by the Ring Team in advanced tests to check the VM status.

Syntax:

```
ringvm_info() ---> List of information about the VM structure
```

94.35 ringvm_ismempool() function

Check if we still have items in the memory pool or not

This function is used to write tests that could detect a memory leak

Syntax:

```
ringvm_ismempool() ---> lStatus
```

94.36 ringvm_runcode() function

Similar to the Eval() function

1 - Used for GUI events like RingQt applications

2 - Execute the Main Loop (i.e. Eval + MainLoop in one function)

3 - Maximum nested events is 255 events

This function is used to write tests that contains events

Syntax:

```
ringvm_runcode(cCode)
```

CHAPTER

NINETYFIVE

TUTORIAL: RING EXTENSIONS IN C/C++

In this chapter we will see simple examples about using C code in Ring programs

95.1 Hello World

Sample : ring/extensions/tutorial/helloworld

The file mylib.c contains

```
#include "ring.h"

#include "stdlib.h"

RING_FUNC(ring_myfunction)
{
        printf("Hello, World!");
}

RING_LIBINIT
{
        RING_API_REGISTER("myfunction",ring_myfunction);
}
```

As we see in the source code, we start with including the ring.h file which contains the definitions for Ring API

Then we use the RING_FUNC macro to define new functions, it's a good idea to start the function definition with **ring_** to make these definitions unique and different than normal C functions

Then we have the function RING_LIBINIT that will be called when the extension is loaded by the Ring VM

In this function we use the RING_API_REGISTER() function to register the new functions in Ring VM

Then we build the extension using :

```
buildvc.bat     // Ring for Windows 32-bit
buildvc_x64.bat // Ring for windows 64-bit
```

The file buildvc_x64.bat contains the next commands to build the extension using Visual C/C++ (64bit)

```
cls
call ../../../src/locatevc.bat x64
cl /c /DEBUG mylib.c -I"..\..\..\include"
```

(continues on next page)

(continued from previous page)

```
link /DEBUG mylib.obj  ..\..\..\lib\ring.lib /DLL /OUT:mylib.dll
del mylib.obj
```

Then we test the function using

```
ring test.ring
```

The file test.ring contains

```
? "Loading Library"
loadlib("mylib.dll")

? "Calling a C Function"
myfunction()
```

Output

```
Loading Library
Calling a C Function
Hello, World!
```

95.2 Build the extension on different platforms

Sample : ring/extensions/tutorial/helloworld2

This extension is the same as the first one but in this time, we support Windows, Linux and macOS.

We will use the next files

```
buildvc.bat
buildvc_x64.bat
buildgcc.sh
buildclang.sh
```

Where we use buildvc.bat/buildvc_x64.bat in Windows, buildgcc.sh in Linux and buildclang.sh in macOS

The file buildgcc.sh contains the next commands

```
gcc -c -fpic mylib.c -I $PWD/../../../include
gcc -shared -o libmylib.so mylib.o -L $PWD/../../../lib -lring
sudo cp libmylib.so /usr/lib
sudo cp libmylib.so /usr/lib64
```

The file buildclang.sh contains

```
clang -c -fpic mylib.c -I $PWD/../../../include
clang -dynamiclib -o libmylib.dylib mylib.o  -L $PWD/../../../lib  -lring
cp libmylib.dylib /usr/local/lib
```

This time we use mylib.ring instead of using LoadLib() function directly

In mylib.ring we have the next code

```
if iswindows()
        LoadLib("mylib.dll")
but ismacosx()
        LoadLib("libmylib.dylib")
else
        LoadLib("libmylib.so")
ok
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
```

95.3 Sum Two Numbers

Sample : ring/extensions/tutorial/sumtwonumbers

In this extension we learn how to create a C function to sum two numbers

This extension is an update to the (Hello World 2) extension in : ring/extensions/tutorial/helloworld2 folder

In mylib.c we update the file to add

```
RING_FUNC(ring_sumtwonumbers)
{
        double nNum1,nNum2,nSum;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 2) {
                        RING_API_ERROR(RING_API_MISS2PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! (RING_API_ISNUMBER(1) && RING_API_ISNUMBER(2)) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Sum Numbers
                nNum1 = RING_API_GETNUMBER(1);
                nNum2 = RING_API_GETNUMBER(2);
                nSum  = nNum1 + nNum2 ;
```

(continues on next page)

(continued from previous page)

```
        // Return Output
                RING_API_RETNUMBER(nSum);
}
```

Then we register the new function

```
RING_API_REGISTER("sumtwonumbers",ring_sumtwonumbers);
```

The previous code is written to check errors, and to be easy to understand

We can write short code like

```
RING_API_RETNUMBER(RING_API_GETNUMBER(1) + RING_API_GETNUMBER(2));
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
```

95.4 Say Hello

Sample : ring/extensions/tutorial/sayhello

In this extension we learn how to create a C function that get a name as string then say hello.

This extension is an update to the (sumtwonumbers) extension in : ring/extensions/tutorial/sumtwonumbers folder

In mylib.c we update the file to add

```
RING_FUNC(ring_sayhello)
{
        // Check Parameters Count
                if (RING_API_PARACOUNT != 1) {
                        RING_API_ERROR(RING_API_MISS1PARA);
                        return;
```

(continues on next page)

(continued from previous page)

```
                }
        // Check Parameters Type
                if ( ! RING_API_ISSTRING(1) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        printf("Hello %s\n",RING_API_GETSTRING(1));
}
```

Then we register the new function

```
RING_API_REGISTER("sayhello",ring_sayhello);
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
```

95.5 Sum List of Numbers

Sample : ring/extensions/tutorial/sumlist

In this extension we learn how to create a C function that sum list of numbers.

This extension is an update to the (sayhello) extension in : ring/extensions/tutorial/sayhello folder

In mylib.c we update the file to add

```
RING_FUNC(ring_sumlist)
{
        List *pList;
        int x,nSum;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 1) {
                        RING_API_ERROR(RING_API_MISS1PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! RING_API_ISLIST(1) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Sum List Numbers
                nSum = 0;
                pList = RING_API_GETLIST(1);
                for(x=1 ; x <= ring_list_getsize(pList) ; x++) {
                        if ( ring_list_isdouble(pList,x) ) {
                                nSum += (int) ring_list_getdouble(pList,x) ;
                        }
                }
        // Return Output
                RING_API_RETNUMBER(nSum);
}
```

Then we register the new function

```
RING_API_REGISTER("sumlist",ring_sumlist);
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
? SumList(aList)
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
Sum List contains numbers from 1 to 10
55
```

95.6 Increment List Items

Sample : ring/extensions/tutorial/incrementlist

In this extension we learn how to create a C function that increment the list items.

This extension is an update to the (sumlist) extension in : ring/extensions/tutorial/sumlist folder

In mylib.c we update the file to add

```
RING_FUNC(ring_inclist)
{
        List *pList;
        int x,nSum;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 2) {
                        RING_API_ERROR(RING_API_MISS2PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! ( RING_API_ISLIST(1) && RING_API_ISNUMBER(2) ) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Increment List Items
                nSum = 0;
                pList = RING_API_GETLIST(1);
                for(x=1 ; x <= ring_list_getsize(pList) ; x++) {
                        if ( ring_list_isdouble(pList,x) ) {
                                ring_list_setdouble_gc(RING_API_STATE,pList,x,
                                        ring_list_getdouble(pList,x)+
                                        RING_API_GETNUMBER(2)) ;
                        }
                }
        // Return Output
                RING_API_RETLIST(pList);
}
```

Then we register the new function

```
RING_API_REGISTER("inclist",ring_inclist);
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
? SumList(aList)

? "Increment List Items"
? inclist(aList,10)
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
Sum List contains numbers from 1 to 10
55
Increment List Items
11
12
13
14
15
16
17
18
19
20
```

95.7 Filter List Items

Sample : ring/extensions/tutorial/filterlist

In this extension we learn how to create a C function that filter the list items.

This extension is an update to the (incrementlist) extension in : ring/extensions/tutorial/incrementlist folder

In mylib.c we update the file to add

```
RING_FUNC(ring_filterlist)
{
        List *pList;
        int x;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 2) {
                        RING_API_ERROR(RING_API_MISS2PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! ( RING_API_ISLIST(1) && RING_API_ISNUMBER(2) ) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Filter List Items
                pList = RING_API_GETLIST(1);
                for(x = ring_list_getsize(pList) ; x >= 1 ; x--)
                        if ( ring_list_isdouble(pList,x) )
                                if ( ! (ring_list_getdouble(pList,x) >
                                                RING_API_GETNUMBER(2)) )
                                        ring_list_deleteitem_gc(RING_API_STATE,pList,x) ;
        // Return Output
                RING_API_RETLIST(pList);
}
```

Then we register the new function

```
RING_API_REGISTER("filterlist",ring_filterlist);
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
```

(continues on next page)

(continued from previous page)

```
? SumList(aList)

? "Increment List Items"
? inclist(aList,10)

? "Filter List Items (Items > 15)"
? filterlist(aList,15)
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
Sum List contains numbers from 1 to 10
55
Increment List Items
11
12
13
14
15
16
17
18
19
20


Filter List Items (Items > 15)
16
17
18
19
20
```

95.8 Replicate List Items

Sample : ring/extensions/tutorial/replicatelist

In this extension we learn how to create a C function that add more items to the list.

This extension is an update to the (filterlist) extension in : ring/extensions/tutorial/filterlist folder

In mylib.c we update the file to add

```
RING_FUNC(ring_replicatelist)
{
        List *pList;
        int x,y,nTimes,nSize;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 2) {
                        RING_API_ERROR(RING_API_MISS2PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! ( RING_API_ISLIST(1) && RING_API_ISNUMBER(2) ) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Replicate List Items
                pList = RING_API_GETLIST(1);
                nSize = ring_list_getsize(pList);
                nTimes = (int) RING_API_GETNUMBER(2);
                if (nTimes < 1) {
                        RING_API_ERROR("Error: The second parameter must be >= 1 \n");
                        return;
                }
                for(x = 1 ; x <= nTimes ; x++)
                        for(y = 1 ; y <= nSize ; y++)
                                if ( ring_list_isdouble(pList,y) )
                                        ring_list_adddouble_gc(RING_API_STATE,pList,
                                                ring_list_getdouble(pList,y));
        // Return Output
                RING_API_RETLIST(pList);
}
```

Then we register the new function

```
RING_API_REGISTER("replicatelist",ring_replicatelist);
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)
```

(continues on next page)

(continued from previous page)

```
? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
? SumList(aList)

? "Increment List Items"
? inclist(aList,10)

? "Filter List Items (Items > 15)"
? filterlist(aList,15)

aList = 1:3
? "Replicate list (1:3) three times then print the items (We expect 12 items)"
? replicatelist(aList,3)
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
Sum List contains numbers from 1 to 10
55
Increment List Items
11
12
13
14
15
16
17
18
19
20


Filter List Items (Items > 15)
16
17
18
19
20
```

(continues on next page)

(continued from previous page)

```
Replicate list (1:3) three times then print the items (We expect 12 items)
1
2
3
1
2
3
1
2
3
1
2
3
```

95.9 Generate List

Sample : ring/extensions/tutorial/generatelist

In this extension we learn how to create a C function that create new list and add items to the list.

This extension is an update to the (replicatelist) extension in : ring/extensions/tutorial/replicatelist folder

In mylib.c we update the file to add

```
RING_FUNC(ring_generatelist)
{
        List *pList;
        int x,nSize;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 1) {
                        RING_API_ERROR(RING_API_MISS1PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! RING_API_ISNUMBER(1) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Create the List
                pList = RING_API_NEWLIST;
        // Generate List Items
                nSize = (int) RING_API_GETNUMBER(1);
                if (nSize < 1) {
                        RING_API_ERROR("Error: The list size must be >= 1 \n");
                        return;
                }
                for(x = 1 ; x <= nSize ; x++)
                        ring_list_adddouble_gc(RING_API_STATE,pList,(double) x);
        // Return Output
                RING_API_RETLIST(pList);
}
```

Then we register the new function

```
RING_API_REGISTER("generatelist",ring_generatelist);
```

The file test.ring contains

```
? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
? SumList(aList)

? "Increment List Items"
? inclist(aList,10)

? "Filter List Items (Items > 15)"
? filterlist(aList,15)

aList = 1:3
? "Replicate list (1:3) three times then print the items (We expect 12 items)"
? replicatelist(aList,3)

? "Create list contains 5 items using C code"
aList = GenerateList(5)
? aList
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
Sum List contains numbers from 1 to 10
55
Increment List Items
11
12
```

(continues on next page)

(continued from previous page)

```
13
14
15
16
17
18
19
20


Filter List Items (Items > 15)
16
17
18
19
20

Replicate list (1:3) three times then print the items (We expect 12 items)
1
2
3
1
2
3
1
2
3
1
2
3

Create list contains 5 items using C code
1
2
3
4
5
```

95.10 Display List

Sample : ring/extensions/tutorial/displaylist

In this extension we learn how to create a C function that display all of the list items including nested lists

In mylib.c we update the file to add

```
void mylib_displaylist(List *pList);

RING_FUNC(ring_displaylist)
{
        List *pList;
```

(continues on next page)

(continued from previous page)

```
        // Check Parameters Count
                if (RING_API_PARACOUNT != 1) {
                        RING_API_ERROR(RING_API_MISS1PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! RING_API_ISLIST(1) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Get the List
                pList = RING_API_GETLIST(1);
        // Display the List Items including Nested Lists
                mylib_displaylist(pList);
}

void mylib_displaylist(List *pList) {
        int x;
        for (x=1; x <= ring_list_getsize(pList); x++) {
                if ( ring_list_isdouble(pList,x) ) {
                        printf("Number : %f \n", ring_list_getdouble(pList,x) ) ;
                } else if ( ring_list_isstring(pList,x) ) {
                        printf("String : %s \n", ring_list_getstring(pList,x) ) ;
                } else if ( ring_list_islist(pList,x) ) {
                        printf("Sub List..\n");
                        mylib_displaylist( ring_list_getlist(pList,x) );
                }
        }
}
```

Then we register the new function

```
RING_API_REGISTER("displaylist",ring_displaylist);
```

The file test.ring contains

```
load "stdlib.ring"

? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
```

(continues on next page)

(continued from previous page)

```
? SumList(aList)

? "Increment List Items"
? inclist(aList,10)

? "Filter List Items (Items > 15)"
? filterlist(aList,15)

aList = 1:3
? "Replicate list (1:3) three times then print the items (We expect 12 items)"
? replicatelist(aList,3)

? "Create list contains 5 items using C code"
aList = GenerateList(5)
? aList

? "Create List (3,2)"
aList = newList(3,2)
aList[1][1] = "R 1 C 1"
aList[1][2] = "R 1 C 2"
aList[2][1] = "R 2 C 1"
aList[2][2] = "R 2 C 2"
aList[3][1] = "R 3 C 1"
aList[3][2] = "R 3 C 2"
? "Print the List using Ring"
? aList
? "Print the List by calling C Code"
displayList(aList)
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
Sum List contains numbers from 1 to 10
55
Increment List Items
11
12
13
14
15
16
17
```

(continues on next page)

(continued from previous page)

```
18
19
20


Filter List Items (Items > 15)
16
17
18
19
20

Replicate list (1:3) three times then print the items (We expect 12 items)
1
2
3
1
2
3
1
2
3
1
2
3

Create list contains 5 items using C code
1
2
3
4
5

Create List (3,2)
Print the List using Ring
R 1 C 1
R 1 C 2
R 2 C 1
R 2 C 2
R 3 C 1
R 3 C 2

Print the List by calling C Code
Sub List..
String : R 1 C 1
String : R 1 C 2
Sub List..
String : R 2 C 1
String : R 2 C 2
Sub List..
String : R 3 C 1
String : R 3 C 2
```

95.11 Update Table

Sample : ring/extensions/tutorial/updatetable

In this extension we learn how to create a C function that update a table contains rows and columns

In mylib.c we update the file to add

```
RING_FUNC(ring_updatetable)
{
        List *pList, *pRow;
        int nRow,nCol;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 2) {
                        RING_API_ERROR(RING_API_MISS2PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! ( RING_API_ISLIST(1) && RING_API_ISNUMBER(2) ) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Get the List (Represent a Table)
                pList = RING_API_GETLIST(1);
        // Update the Table Rows and Columns
        for (nRow = 1 ; nRow <= ring_list_getsize(pList) ; nRow++ ) {
                if ( ring_list_islist(pList,nRow) ) {
                pRow = ring_list_getlist(pList,nRow);
                for (nCol = 1 ; nCol <= ring_list_getsize(pRow) ; nCol++ ) {
                        if ( ring_list_isdouble(pRow,nCol) ) {
                                ring_list_setdouble_gc(RING_API_STATE,pRow,nCol,RING_API_
↪GETNUMBER(2));
                        } else {
                                RING_API_ERROR("Error : We expect numbers!\n");
                                return ;
                        }
                }
                } else {
                        RING_API_ERROR("Error : The parameter is not a table! \n");
                        return ;
                }
        }
}
```

Then we register the new function

```
RING_API_REGISTER("updatetable",ring_updatetable);
```

The file test.ring contains

```
load "stdlib.ring"

? "Loading Library"
load "mylib.ring"
```

(continues on next page)

(continued from previous page)

```
? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
? SumList(aList)

? "Increment List Items"
? inclist(aList,10)

? "Filter List Items (Items > 15)"
? filterlist(aList,15)

aList = 1:3
? "Replicate list (1:3) three times then print the items (We expect 12 items)"
? replicatelist(aList,3)

? "Create list contains 5 items using C code"
aList = GenerateList(5)
? aList

? "Create List (3,2)"
aList = newList(3,2)
aList[1][1] = "R 1 C 1"
aList[1][2] = "R 1 C 2"
aList[2][1] = "R 2 C 1"
aList[2][2] = "R 2 C 2"
aList[3][1] = "R 3 C 1"
aList[3][2] = "R 3 C 2"
? "Print the List using Ring"
? aList
? "Print the List by calling C Code"
displayList(aList)
? ""

? "Create List (2,2)"
aList = newList(2,2)
? "Update the list using C code - set all cells to 10"
UpdateTable(aList,10)
? "aList[1][1] : " + aList[1][1]
? "aList[1][2] : " + aList[1][2]
? "aList[2][1] : " + aList[2][1]
? "aList[2][2] : " + aList[2][2]
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
Sum List contains numbers from 1 to 10
55
Increment List Items
11
12
13
14
15
16
17
18
19
20


Filter List Items (Items > 15)
16
17
18
19
20

Replicate list (1:3) three times then print the items (We expect 12 items)
1
2
3
1
2
3
1
2
3
1
2
3

Create list contains 5 items using C code
1
2
3
4
5
```

(continues on next page)

(continued from previous page)

```
Create List (3,2)
Print the List using Ring
R 1 C 1
R 1 C 2
R 2 C 1
R 2 C 2
R 3 C 1
R 3 C 2

Print the List by calling C Code
Sub List..
String : R 1 C 1
String : R 1 C 2
Sub List..
String : R 2 C 1
String : R 2 C 2
Sub List..
String : R 3 C 1
String : R 3 C 2

Create List (2,2)
Update the list using C code - set all cells to 10
aList[1][1] : 10
aList[1][2] : 10
aList[2][1] : 10
aList[2][2] : 10
```

95.12 Create Table

Sample : ring/extensions/tutorial/createtable

In this extension we learn how to create a C function that create a table contains rows and columns

In mylib.c we update the file to add

```
RING_FUNC(ring_createtable)
{
        List *pList, *pRow;
        int x,y,nRows,nCols;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 2) {
                        RING_API_ERROR(RING_API_MISS2PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! ( RING_API_ISNUMBER(1) && RING_API_ISNUMBER(2) ) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Create the List
```

(continues on next page)

(continued from previous page)

```
                pList = RING_API_NEWLIST;
        // Create the table items
                nRows = (int) RING_API_GETNUMBER(1);
                nCols = (int) RING_API_GETNUMBER(2);
                if ( (nRows < 1) || (nCols < 1) ) {
                RING_API_ERROR("Error: The table rows and columns must be >= 1 \n");
                return;
                }
                for(x = 1 ; x <= nRows ; x++) {
                        pRow = ring_list_newlist(pList);
                        for(y = 1 ; y <= nCols ; y++)
                                ring_list_adddouble_gc(RING_API_STATE,pRow,0.0);
                }
        // Return Output
                RING_API_RETLIST(pList);
}
```

Then we register the new function

```
RING_API_REGISTER("createtable",ring_createtable);
```

The file test.ring contains

```
load "stdlib.ring"

? "Loading Library"
load "mylib.ring"

? "Calling a C Function"
myfunction()

? "Sum Two Numbers (3,5)"
? SumTwoNumbers(3,5)

? "Say Hello"
SayHello("Mahmoud")

? "Sum List contains numbers from 1 to 10"
aList = 1:10
? SumList(aList)

? "Increment List Items"
? inclist(aList,10)

? "Filter List Items (Items > 15)"
? filterlist(aList,15)

aList = 1:3
? "Replicate list (1:3) three times then print the items (We expect 12 items)"
? replicatelist(aList,3)

? "Create list contains 5 items using C code"
```

(continues on next page)

(continued from previous page)

```
aList = GenerateList(5)
? aList

? "Create List (3,2)"
aList = newList(3,2)
aList[1][1] = "R 1 C 1"
aList[1][2] = "R 1 C 2"
aList[2][1] = "R 2 C 1"
aList[2][2] = "R 2 C 2"
aList[3][1] = "R 3 C 1"
aList[3][2] = "R 3 C 2"
? "Print the List using Ring"
? aList
? "Print the List by calling C Code"
displayList(aList)
? ""

? "Create List (2,2)"
aList = newList(2,2)
? "Update the list using C code - set all cells to 10"
UpdateTable(aList,10)
? "aList[1][1] : " + aList[1][1]
? "aList[1][2] : " + aList[1][2]
? "aList[2][1] : " + aList[2][1]
? "aList[2][2] : " + aList[2][2]
? ""

? "Create List (3,3) using C code"
aList = CreateTable(3,3)
? "aList[1][1] : " + aList[1][1]
? "aList[1][2] : " + aList[1][2]
? "aList[1][3] : " + aList[1][3]
? "aList[2][1] : " + aList[2][1]
? "aList[2][2] : " + aList[2][2]
? "aList[2][3] : " + aList[2][3]
? "aList[3][1] : " + aList[3][1]
? "aList[3][2] : " + aList[3][2]
? "aList[3][3] : " + aList[3][3]
```

Then we test the function using

```
ring test.ring
```

Output

```
Loading Library
Calling a C Function
Hello, World!
Sum Two Numbers (3,5)
8
Say Hello
Hello Mahmoud
```

(continues on next page)

(continued from previous page)

```
Sum List contains numbers from 1 to 10
55
Increment List Items
11
12
13
14
15
16
17
18
19
20


Filter List Items (Items > 15)
16
17
18
19
20

Replicate list (1:3) three times then print the items (We expect 12 items)
1
2
3
1
2
3
1
2
3
1
2
3

Create list contains 5 items using C code
1
2
3
4
5

Create List (3,2)
Print the List using Ring
R 1 C 1
R 1 C 2
R 2 C 1
R 2 C 2
R 3 C 1
R 3 C 2
```

(continues on next page)

(continued from previous page)

```
Print the List by calling C Code
Sub List..
String : R 1 C 1
String : R 1 C 2
Sub List..
String : R 2 C 1
String : R 2 C 2
Sub List..
String : R 3 C 1
String : R 3 C 2

Create List (2,2)
Update the list using C code - set all cells to 10
aList[1][1] : 10
aList[1][2] : 10
aList[2][1] : 10
aList[2][2] : 10

Create List (3,3) using C code
aList[1][1] : 0
aList[1][2] : 0
aList[1][3] : 0
aList[2][1] : 0
aList[2][2] : 0
aList[2][3] : 0
aList[3][1] : 0
aList[3][2] : 0
aList[3][3] : 0
```

CHAPTER

NINETYSIX

EXTENSION USING THE C/C++ LANGUAGES

We can extend the Ring Virtual Machine (RingVM) by adding new functions written in the C programming language or C++. The RingVM comes with many functions written in C that we can call like any Ring function.

We can extend the language by writing new functions then rebuilding the RingVM again, or we can create shared library (DLL/So/Dylib) file to extend the RingVM without the need to rebuild it.

The Ring language source code comes with two files to add new modules to the RingVM, ring_ext.h and ring_ext.c

96.1 ring_ext.h

The file ring_ext.h contains constants that we can change to include/exclude modules during the build process.

```
#ifndef ringext_h
#define ringext_h
/* Constants */
#define RING_VM_LISTFUNCS       1
#define RING_VM_REFMETA         1
#define RING_VM_MATH            1
#define RING_VM_FILE            1
#define RING_VM_OS              1
#define RING_VM_MYSQL           1
#define RING_VM_ODBC            1
#define RING_VM_OPENSSL         1
#define RING_VM_CURL            1
#define RING_VM_DLL             1
#endif
```

96.2 ring_ext.c

The file ring_ext.c check constants defined in ring_ext.h before calling the start-up function in each module.

Each module contains a function that register the module functions in the RingVM.

```
#include "ring.h"

void ring_vm_extension ( RingState *pRingState )
{
        /* Reflection and Meta-programming */
```

(continues on next page)

(continued from previous page)

```
        #if RING_VM_REFMETA
                ring_vm_refmeta_loadfunctions(pRingState);
        #endif
        /* List Functions */
        #if RING_VM_LISTFUNCS
                ring_vm_listfuncs_loadfunctions(pRingState);
        #endif
        /* Math */
        #if RING_VM_MATH
                ring_vm_math_loadfunctions(pRingState);
        #endif
        /* File */
        #if RING_VM_FILE
                ring_vm_file_loadfunctions(pRingState);
        #endif
        /* OS */
        #if RING_VM_OS
                ring_vm_os_loadfunctions(pRingState);
        #endif
        /* MySQL */
        #if RING_VM_MYSQL
                ring_vm_mysql_loadfunctions(pRingState);
        #endif
        /* ODBC */
        #if RING_VM_ODBC
                ring_vm_odbc_loadfunctions(pRingState);
        #endif
        /* OPENSSL */
        #if RING_VM_OPENSSL
                ring_vm_openssl_loadfunctions(pRingState);
        #endif
        /* CURL */
        #if RING_VM_CURL
                ring_vm_curl_loadfunctions(pRingState);
        #endif
        /* DLL */
        #if RING_VM_DLL
                ring_vm_dll_loadfunctions(pRingState);
        #endif
}
```

96.3 Module Organization

Each module starts by include the ring header file (ring.h). This files contains the Ring API that we can use to extend the RingVM.

Each module comes with a function to register the module functions in the RingVM The registration is done by using RING_API_REGISTER() function.

The RING_API_REGISTER() function takes two parameters, the first is the function name that will be used by Ring programs to call the function. The second parameter is the function pointer in the C program.

for example, the ring_vmmath.c module contains the next code to register the module functions

```
#include "ring.h"

void ring_vm_math_loadfunctions ( RingState *pRingState )
{
        RING_API_REGISTER("sin",ring_vm_math_sin);
        RING_API_REGISTER("cos",ring_vm_math_cos);
        RING_API_REGISTER("tan",ring_vm_math_tan);
        RING_API_REGISTER("asin",ring_vm_math_asin);
        RING_API_REGISTER("acos",ring_vm_math_acos);
        RING_API_REGISTER("atan",ring_vm_math_atan);
        RING_API_REGISTER("atan2",ring_vm_math_atan2);
        RING_API_REGISTER("sinh",ring_vm_math_sinh);
        RING_API_REGISTER("cosh",ring_vm_math_cosh);
        RING_API_REGISTER("tanh",ring_vm_math_tanh);
        RING_API_REGISTER("exp",ring_vm_math_exp);
        RING_API_REGISTER("log",ring_vm_math_log);
        RING_API_REGISTER("log10",ring_vm_math_log10);
        RING_API_REGISTER("ceil",ring_vm_math_ceil);
        RING_API_REGISTER("floor",ring_vm_math_floor);
        RING_API_REGISTER("fabs",ring_vm_math_fabs);
        RING_API_REGISTER("pow",ring_vm_math_pow);
        RING_API_REGISTER("sqrt",ring_vm_math_sqrt);
        RING_API_REGISTER("unsigned",ring_vm_math_unsigned);
        RING_API_REGISTER("decimals",ring_vm_math_decimals);
        RING_API_REGISTER("murmur3hash",ring_vm_math_murmur3hash);
}
```

Tip: Remember that the function ring_vm_math_loadfunctions() will be called by the ring_vm_extension() function (in the ring_ext.c file).

96.4 Function Structure

Each module function may contains the next steps

1 - Check Parameters Count

2 - Check Parameters Type

3 - Get Parameters Values

4 - Execute Code/Call Functions

5 - Return Value

The structure is very similar to any function (Input - Process - Output) But here we will use the Ring API for the steps 1,2,3 and 5.

96.5 Check Parameters Count

We can check the parameters count using the RING_API_PARACOUNT macro.

We can compare RING_API_PARACOUNT with any numeric value using == or != operators.

Example:

```
if ( RING_API_PARACOUNT != 1 ) {
        /* code */
}
```

Example:

```
if ( RING_API_PARACOUNT == 1 ) {
        /* code */
}
```

96.6 Display Error Message

We can display error messages using the RING_API_ERROR() function.

The function will display the error and end the execution of the program.

Note: the behaviour of this function can be changed by the Ring code using Try/Catch/Done statements, so in your C code, use Return after this function.

Syntax:

```
RING_API_ERROR(const char *cErrorMsg);
```

The Ring API comes with some of predefined error messages that we can use

```
#define RING_API_MISS1PARA "Bad parameters count, the function expect one parameter"
#define RING_API_MISS2PARA "Bad parameters count, the function expect two parameters"
#define RING_API_MISS3PARA "Bad parameters count, the function expect three parameters"
#define RING_API_MISS4PARA "Bad parameters count, the function expect four parameters"
#define RING_API_BADPARATYPE    "Bad parameter type!"
#define RING_API_BADPARACOUNT   "Bad parameters count!"
#define RING_API_BADPARARANGE   "Bad parameters value, error in range!"
#define RING_API_NOTPOINTER     "Error in parameter, not pointer!"
#define RING_API_NULLPOINTER    "Error in parameter, NULL pointer!"
#define RING_API_EMPTYLIST      "Bad parameter, empty list!"
```

96.7 Check Parameters Type

We can check the parameter type using the next functions

```
int RING_API_ISNUMBER(int nParameterNumber);
int RING_API_ISSTRING(int nParameterNumber);
int RING_API_ISLIST(int nParameterNumber);
int RING_API_ISCPOINTER(int nParameterNumber);
int RING_API_ISPOINTER(int nParameterNumber);  // List or C Pointer
```

The output of these functions will be 1 (True) or 0 (False).

96.8 Get Parameters Values

We can get parameters values using the next functions

```
double RING_API_GETNUMBER(int nParameterNumber);
const char *RING_API_GETSTRING(int nParameterNumber);
int RING_API_GETSTRINGSIZE(int nParameterNumber);
List *RING_API_GETLIST(int nParameterNumber);
void *RING_API_GETCPOINTER(int nParameterNumber, const char *cPoinerType);
int RING_API_GETPOINTERTYPE(int nParameterNumber);
```

If we would like to get pointers to char, int, float & double

We can use the next functions

```
char *RING_API_GETCHARPOINTER(int nParameterNumber);
int *RING_API_GETINTPOINTER(int nParameterNumber);
void RING_API_ACCEPTINTVALUE(int nParameterNumber);
float *RING_API_GETFLOATPOINTER(int nParameterNumber);
void RING_API_ACCEPTFLOATVALUE(int nParameterNumber);
double *RING_API_GETDOUBLEPOINTER(int nParameterNumber);
```

96.9 Return Value

We can return values from our function using the next functions.

```
RING_API_RETNUMBER(double nValue);
RING_API_RETSTRING(const char *cString);
RING_API_RETSTRING2(const char *cString,int nStringSize);
RING_API_RETLIST(List *pList);
RING_API_RETCPOINTER(void *pValue,const char *cPointerType);
RING_API_RETMANAGEDCPOINTER(void *pValue,const char *cPointerType,
                                void (* pFreeFunc)(void *,void *))
```

96.10 Function Prototype

When we define new function to be used for RingVM extension, we use the next prototype

```
void my_function_name( void *pPointer );
```

or we can use the RING_FUNC() Macro

```
RING_FUNC(my_function_name);
```

96.11 Sin() Function Implementation

The next code represents the sin() function implementation using the Ring API and the sin() C function.

```
void ring_vm_math_sin ( void *pPointer )
{
        if ( RING_API_PARACOUNT != 1 ) {
                RING_API_ERROR(RING_API_MISS1PARA);
                return ;
        }
        if ( RING_API_ISNUMBER(1) ) {
                RING_API_RETNUMBER(sin(RING_API_GETNUMBER(1)));
        } else {
                RING_API_ERROR(RING_API_BADPARATYPE);
        }
}
```

96.12 Fopen() and Fclose() Functions Implementation

The next code represents the fopen() function implementation using the Ring API and the fopen() C Function.

The function takes two parameters, the first parameter is the file name as string. The second parameter is the mode as string.

In the file ring_vmfile.h we have some constants to use as the pointer type like

```
#define RING_VM_POINTER_FILE    "file"
#define RING_VM_POINTER_FILEPOS "filepos"
```

The function implementation in ring_vmfile.c

```
void ring_vm_file_fopen ( void *pPointer )
{
        FILE *fp  ;
        if ( RING_API_PARACOUNT != 2 ) {
                RING_API_ERROR(RING_API_MISS2PARA);
                return ;
        }
        if ( RING_API_ISSTRING(1) && RING_API_ISSTRING(2) ) {
                fp = fopen(RING_API_GETSTRING(1),RING_API_GETSTRING(2));
```

(continues on next page)

(continued from previous page)

```
                RING_API_RETCPOINTER(fp,RING_VM_POINTER_FILE);
        } else {
                RING_API_ERROR(RING_API_BADPARATYPE);
        }
}
```

The next code represents the fclose() function implementation

```
void ring_vm_file_fclose ( void *pPointer )
{
        FILE *fp  ;
        if ( RING_API_PARACOUNT != 1 ) {
                RING_API_ERROR(RING_API_MISS1PARA);
                return ;
        }
        if ( RING_API_ISCPOINTER(1) ) {
                fp = (FILE *) RING_API_GETCPOINTER(1,RING_VM_POINTER_FILE) ;
                if ( fp != NULL ) {
                        RING_API_RETNUMBER(fclose(fp));
                        RING_API_SETNULLPOINTER(1);
                }
        } else {
                RING_API_ERROR(RING_API_BADPARATYPE);
        }
}
```

From fopen() and fclose() implementation we learned

1 - how to return C pointer using RING_API_RETCPOINTER() function

2 - how to check if the parameter is a pointer using the RING_API_ISCPOINTER() function

3 - how to get C pointer value using the RING_API_GETCPOINTER() function

4 - how to set the C pointer variable (in RingVM) to NULL using the RING_API_SETNULLPOINTER() function

96.13 Ring API - List Functions

In this section we will learn about the list functions provided by the Ring API to create new lists and manipulate the list items.

This section contains the most important functions required for common use-cases

For a complete list of function, including functions that store and process integers, pointers & objects, see this file: ring/language/include/rlist.h

If your list will be modified by Ring code then it's required to use the _gc version of the functions

Lists created using ring_list_new()/ring_list_new_gc() must be deleted using ring_list_delete()/ring_list_delete_gc().

If we created a list using RING_API_NEWLIST then deleting the list is not required and may cause problems, because Ring VM will delete it automatically after the end of the caller scope.

Using RING_API_NEWLISTUSINGBLOCKS1D and RING_API_NEWLISTUSINGBLOCKS2D is similar to RING_API_NEWLIST but we have better performance when we create large lists.

```
// Create and delete lists

List * RING_API_NEWLIST
List * RING_API_NEWLISTUSINGBLOCKS1D(int nItems)
List * RING_API_NEWLISTUSINGBLOCKS2D(int nRows,int nColumns)
List * ring_list_new_gc ( void *pState,unsigned int nSize ) ;
List * ring_list_delete_gc ( void *pState,List *pList ) ;

List * ring_list_new ( unsigned int nSize ) ;
List * ring_list_delete ( List *pList ) ;

// Add Items

void ring_list_adddouble_gc ( void *pState,List *pList,double x ) ;
void ring_list_addstring_gc ( void *pState,List *pList,const char *str ) ;
void ring_list_addstring2_gc ( void *pState,List *pList,const char *str,int nStrSize ) ;
List * ring_list_newlist_gc ( void *pState,List *pList ) ;

void ring_list_adddouble ( List *pList,double x ) ;
void ring_list_addstring ( List *pList,const char *str ) ;
void ring_list_addstring2 ( List *pList,const char *str,int nStrSize ) ;
List * ring_list_newlist ( List *pList ) ;

// Get List Size

int ring_list_getsize( List *pList );

// Check Item Type

int ring_list_isdouble ( List *pList, unsigned int index ) ;
int ring_list_isstring ( List *pList, unsigned int index ) ;
int ring_list_islist ( List *pList, unsigned int index ) ;
int ring_list_isobject ( List *pList ) ;

// Get Items

double ring_list_getdouble ( List *pList, unsigned int index ) ;
char * ring_list_getstring ( List *pList, unsigned int index ) ;
int ring_list_getstringsize ( List *pList, unsigned int index ) ;
String * ring_list_getstringobject ( List *pList, unsigned int index ) ;
List * ring_list_getlist ( List *pList, unsigned int index ) ;

// Insert Items

void ring_list_insertdouble_gc ( void *pState,List *pList,unsigned int nPos,double x ) ;
void ring_list_insertstring_gc ( void *pState,List *pList,unsigned int nPos,const char␣
↪*str ) ;
void ring_list_insertstring2_gc ( void *pState,List *pList,unsigned int nPos,const char␣
↪*str,int nStrSize ) ;
List * ring_list_insertlist_gc ( void *pState,List *pList,unsigned int nPos ) ;

void ring_list_insertdouble ( List *pList,unsigned int nPos,double x ) ;
void ring_list_insertstring ( List *pList,unsigned int nPos,const char *str ) ;
```

(continues on next page)

(continued from previous page)

```
void ring_list_insertstring2 ( List *pList,unsigned int nPos,const char *str,int␣
↪nStrSize ) ;
List * ring_list_insertlist ( List *pList,unsigned int nPos ) ;

// Set Items

void ring_list_setdouble_gc ( void *pState,List *pList, unsigned int index ,double␣
↪number ) ;
void ring_list_setstring_gc ( void *pState,List *pList, unsigned int index ,const char␣
↪*str ) ;
void ring_list_setstring2_gc ( void *pState,List *pList, unsigned int index ,const char␣
↪*str,int nStrSize ) ;
void ring_list_setlist_gc ( void *pState,List *pList, unsigned int index ) ;

void ring_list_setdouble ( List *pList, unsigned int index ,double number ) ;
void ring_list_setstring ( List *pList, unsigned int index ,const char *str ) ;
void ring_list_setstring2 ( List *pList, unsigned int index ,const char *str,int␣
↪nStrSize ) ;
void ring_list_setlist ( List *pList, unsigned int index ) ;

// Delete Items

void ring_list_deleteitem_gc ( void *pState,List *pList,unsigned int index ) ;
void ring_list_deleteallitems_gc ( void *pState,List *pList ) ;

void ring_list_deleteitem ( List *pList,unsigned int index ) ;
void ring_list_deleteallitems ( List *pList ) ;

// Copy Lists

void ring_list_copy_gc ( void *pState,List *pNewList, List *pList ) ;
void ring_list_copy ( List *pNewList, List *pList ) ;

// Find Items

int ring_list_findstring ( List *pList,const char *str,unsigned int nColumn ) ;
int ring_list_finddouble ( List *pList,double nNum1,unsigned int nColumn ) ;

// Swap

void ring_list_swap ( List *pList,unsigned int x,unsigned int y ) ;
void ring_list_swaptwolists ( List *pList1, List *pList2 ) ;

// Print

void ring_list_print ( List *pList ) ;
void ring_list_print2 ( List *pList,unsigned int nDecimals ) ;
void ring_list_printobj ( List *pList, unsigned int nDecimals ) ;
```

96.14 RING_API_STATE

To modify or delete the Lists and Strings created using Ring code, We have to use specific Lists/Strings functions. These functions requires that the Ring State to be passed as the first parameter.

Also, these functions add _gc to the function name which means that it uses the Ring Garbage Collector.

Some of these functions are

```
void ring_list_setdouble_gc ( void *pState, List *pList, int index , double number ) ;
void ring_list_adddouble_gc ( void *pState, List *pList, double x ) ;
void ring_list_deleteitem_gc ( void *pState, List *pList, int index ) ;
```

Example:

The next C code implement the filterList() function

We pass a list and a number to this function and it will check each item in the list.

If the item value is not greater than the passed number, then the item will be deleted.

In the implementation we uses ring_list_deleteitem_gc() function to delete the item and we pass Ring state as the first parameter using RING_API_STATE.

```
RING_FUNC(ring_filterlist)
{
        List *pList;
        int x;
        // Check Parameters Count
                if (RING_API_PARACOUNT != 2) {
                        RING_API_ERROR(RING_API_MISS2PARA);
                        return;
                }
        // Check Parameters Type
                if ( ! ( RING_API_ISLIST(1) && RING_API_ISNUMBER(2) ) ) {
                        RING_API_ERROR(RING_API_BADPARATYPE);
                        return;
                }
        // Filter List Items
                pList = RING_API_GETLIST(1);
                for(x = ring_list_getsize(pList) ; x >= 1 ; x--)
                        if ( ring_list_isdouble(pList,x) )
                                if ( ! (ring_list_getdouble(pList,x) >
                                                RING_API_GETNUMBER(2)) )
                                        ring_list_deleteitem_gc(RING_API_STATE,pList,x) ;
        // Return Output
                RING_API_RETLIST(pList);
}
```

96.15 Ring API - String Functions

In this section we will learn about the string functions provided by the Ring API to create new string and manipulate the string content.

As in List functions, the GC version of these functions (which add _gc to the name) exist.

Here, Using the GC version of the functions is not necessary for stability but it could be helpful for performance.

Why it's not necessary for stability as we have in the List functions?

Because Ring API uses List * to pass lists from Ring VM to C code and also uses List * to return lists from C code to Ring. These lists are passed by pointers, They are the same lists, and we need the same memory functions to process the list items.

While when using Strings, Ring API create new strings when using RING_API_GETSTRING() or RING_API_RETSTRING()

For a complete list of function, including functions that uses the GC, see this file: ring/language/include/rstring.h

```
// Create and delete strings

String * ring_string_new ( const char *str ) ;
String * ring_string_new2 ( const char *str,int nStrSize ) ;
String * ring_string_delete ( String *pString ) ;

// Get string

char * ring_string_get ( String *pString ) ;
int ring_string_size ( String *pString ) ;

// Add text to the string

void ring_string_add ( String *pString,const char *str ) ;
void ring_string_add2 ( String *pString,const char *str,int nStrSize ) ;

// Change the string

void ring_string_set ( String *pString,const char *str ) ;
void ring_string_set2 ( String *pString,const char *str,int nStrSize ) ;
void ring_string_setfromint ( String *pString,int x ) ;

// Print the string

void ring_string_print ( String *pString ) ;

// Convert to lower/UPPER case

void ring_string_tolower(String *pString)
void ring_string_toupper(String *pString)
char * ring_string_lower ( char *cStr ) ;
char * ring_string_upper ( char *cStr ) ;
char * ring_string_lower2 ( char *cStr,int nStrSize ) ;
char * ring_string_upper2 ( char *cStr,int nStrSize ) ;

// Find
```

(continues on next page)

(continued from previous page)

```
char * ring_string_find ( char *cStr1,char *cStr2 ) ;
char * ring_string_find2 ( char *cStr1,int nStrSize1,char *cStr2,int nStrSize2 ) ;
```

96.16 MySQL_Columns() Function Implementation

The next code presents the MySQL_Columns() function implementation.

This function returns table columns information.

```
void ring_vm_mysql_columns ( void *pPointer )
{
        MYSQL *con  ;
        MYSQL_RES *result  ;
        int nColumns,x  ;
        MYSQL_ROW row  ;
        MYSQL_FIELD *field  ;
        List *pList, *pList2  ;
        if ( RING_API_PARACOUNT != 1 ) {
                RING_API_ERROR(RING_API_MISS1PARA);
                return ;
        }
        if ( RING_API_ISCPOINTER(1) ) {
                con = (MYSQL *) RING_API_GETCPOINTER(1,RING_VM_POINTER_MYSQL) ;
                if ( con == NULL ) {
                        return ;
                }
                result = mysql_store_result(con);
                if ( result == NULL ) {
                        RING_API_RETNUMBER(0);
                        return ;
                }
                pList = RING_API_NEWLIST ;
                nColumns = mysql_num_fields(result);
                if ( row = mysql_fetch_row(result) ) {
                        while ( field = mysql_fetch_field(result) ) {
                                pList2 = ring_list_newlist_gc(RING_API_STATE,pList);
                                ring_list_addstring(pList2,field->name);
                                ring_list_adddouble(pList2,field->length);
                                ring_list_adddouble(pList2,field->type);
                                ring_list_adddouble(pList2,field->flags);
                        }
                }
                mysql_free_result(result);
                RING_API_RETLIST(pList);
        } else {
                RING_API_ERROR(RING_API_BADPARATYPE);
        }
}
```

Lists are of type List, in the previous function we declared two pointers of type List using List *pList, *pList2;

Note: The function uses RING_API_NEWLIST to create new list instead of ring_list_new() to create the list in Temp. Memory related to the function scope. This way we can return the list from the function. Also we don't delete the list, if it's stored in a variable by Ring Code it will be saved, if not it will be automatically deleted by RingVM.

The list can contains sub lists, we used the function ring_list_newlist_gc() to create a sublist.

The function ring_list_addstring() is used to add string items to the list/sublist.

The function ring_list_adddouble() is used to add numeric items to the list/sublist.

Note: All numeric items in lists returned from RingVM extension functions must be of type double and added to the list using ring_list_adddouble() function.

We return the list from the extension function using the RING_API_RETLIST() function.

96.17 Dynamic/Shared Libraries (DLL/So/Dylib) and LoadLib() function

Instead of rebuilding the RingVM after writing new functions using C/C++ and the Ring API, we can create a DLL/So/Dylib file and dynamically use the functions provided by this file in the runtime using the LoadLib() function.

Dynamic library example in C

```
#include "ring.h"

RING_DLL __declspec(dllexport)

RING_FUNC(ring_ringlib_dlfunc)
{
        printf("Message from dlfunc");
}

RING_LIBINIT
{
        RING_API_REGISTER("dlfunc",ring_ringlib_dlfunc);
}
```

the idea is to create the RING_LIBINIT function, this function will be called by the RingVM when we use the generated DLL file through the LoadLib() function.

Inside the RING_LIBINIT function we can register the module function or call a function that do the registration process for all of the module functions.

The next Ring code demonstrates how to use the DLL library during the runtime.

```
See "Dynamic DLL" + NL
LoadLib("ringlib.dll")
dlfunc()
```

Output:

```
Dynamic DLL
Message from dlfunc
```

96.18 Using RING_API_RETMANAGEDCPOINTER()

Using RING_API_RETMANAGEDCPOINTER() the Ring extensions written in C/C++ languages can return a managed pointer to Ring. This pointer can be controlled by the Ring VM using reference counting.

This is important to avoid the need to write code that free the unmanaged resources like QPixMap objects in RingQt.

Also the Code Generator for extensions is updated to automatically use RING_API_RETMANAGEDCPOINTER() based on need.

Syntax:

```
RING_API_RETMANAGEDCPOINTER(void *pValue,const char *cPointerType,
                                void (* pFreeFunc)(void *,void *))
```

Example:

The next example from ring_qt.cpp - QPixMap Class - Scaled() Method.

```
RING_FUNC(ring_QPixmap_scaled)
{
        QPixmap *pObject ;
        if ( RING_API_PARACOUNT != 5 ) {
                RING_API_ERROR(RING_API_BADPARACOUNT);
                return ;
        }
        RING_API_IGNORECPOINTERTYPE ;
        if ( ! RING_API_ISCPOINTER(1) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        pObject = (QPixmap *) RING_API_GETCPOINTER(1,"QPixmap");
        if ( ! RING_API_ISNUMBER(2) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        if ( ! RING_API_ISNUMBER(3) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        if ( ! RING_API_ISNUMBER(4) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        if ( ! RING_API_ISNUMBER(5) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        {
                QPixmap *pValue ;
```

(continues on next page)

(continued from previous page)

```
                pValue = new QPixmap() ;
                *pValue = pObject->scaled( (int ) RING_API_GETNUMBER(2),
                         (int ) RING_API_GETNUMBER(3),
                         (Qt::AspectRatioMode )  (int) RING_API_GETNUMBER(4),
                         (Qt::TransformationMode )  (int) RING_API_GETNUMBER(5));
                RING_API_RETMANAGEDCPOINTER(pValue,"QPixmap",ring_QPixmap_freefunc);
        }
}
```

The function that will free the memory takes two parameters (Ring State and the allocated Memory Pointer)

Example:

```
void ring_QPixmap_freefunc(void *pState,void *pPointer)
{
        QPixmap *pObject ;
        pObject = (QPixmap *) pPointer;
        delete pObject ;
}
```

96.19 Memory Functions

Ring API provides the next functions for Memory Management

These functions uses the Memory pool provided by Ring VM

```
RING_API_MALLOC(nSize)
RING_API_CALLOC(nItems,nItemSize)
RING_API_REALLOC(pPointer,nSize)
RING_API_FREE(pPointer)
RING_API_FREEFUNC
```

- Use RING_API_MALLOC() instead of the malloc() function.
- Use RING_API_CALLOC() instead of the calloc() function.
- Use RING_API_REALLOC() instead of the realloc() function.
- Use RING_API_FREE() instead of the free() function.
- RING_API_FREEFUNC provides a function pointer to the ring_state_free() function

CHAPTER

NINETYSEVEN

EMBEDDING RING LANGUAGE IN C/C++ PROGRAMS

We can use the Ring language from C/C++ programs using the next functions

```
RingState *ring_state_init();
ring_state_runcode(RingState *pState,const char *cCode);
ring_state_delete(RingState *pState);
```

97.1 Ring State

The idea is to use the ring_state_init() to create new state for the Ring Language then call the ring_state_runcode() function to execute Ring code using the same state. When we are done, we call the ring_state_delete() to free the memory.

Example:

```
#include "ring.h"
#include "stdlib.h"
int main(int argc, char *argv[])
{
  RingState *pState = ring_state_init();
  printf("welcome\n");
  ring_state_runcode(pState,"see 'hello world from the ring programming language'+nl");
  ring_state_delete(pState);
}
```

Output:

```
welcome
hello world from the ring programming language
```

97.2 Ring State Functions

The Ring API comes with the next functions to create and delete the state. Also we have functions to create new variables and get variables values.

```
RingState * ring_state_init ( void ) ;
RingState * ring_state_delete ( RingState *pRingState ) ;
void ring_state_runcode ( RingState *pRingState,const char *cStr ) ;
```

(continues on next page)

(continued from previous page)

```
List * ring_state_findvar ( RingState *pRingState,const char *cStr ) ;
List * ring_state_newvar ( RingState *pRingState,const char *cStr ) ;
void ring_state_main ( int argc, char *argv[] ) ;
int  ring_state_runfile ( RingState *pRingState,const char *cFileName ) ;
void ring_state_runobjectfile ( RingState *pRingState,const char *cFileName ) ;
void ring_state_runobjectstring ( RingState *pRingState,char *cString,const char␣
↪*cFileName ) ;
```

97.3 Ring State Variables

We can create more than one ring state in the same program and we can create and modify variable values.

To get the variable list we can use the ring_state_findvar() function.

To create new variable we can use the ring_state_newvar() function.

Example:

```
#include "ring.h"
#include "stdlib.h"

int main(int argc, char *argv[])
{
  List *pList;

  RingState *pState = ring_state_init();
  RingState *pState2 = ring_state_init();

  printf("welcome\n");
  ring_state_runcode(pState,"see 'hello world from the ring programming language'+nl");

  printf("Again from C we will call ring code\n");
  ring_state_runcode(pState,"for x = 1 to 10 see x + nl next");

  ring_state_runcode(pState2,"for x = 1 to 5 see x + nl next");

  printf("Now we will display the x variable value from ring code\n");
  ring_state_runcode(pState,"see 'x value : ' + x + nl ");
  ring_state_runcode(pState2,"see 'x value : ' + x + nl ");

  pList = ring_state_findvar(pState,"x");

  printf("Printing Ring variable value from C , %.0f\n",
               ring_list_getdouble(pList,RING_VAR_VALUE));

  printf("now we will set the ring variable value from C\n");
  ring_list_setdouble(pList,RING_VAR_VALUE,20);

  ring_state_runcode(pState,"see 'x value after update : ' + x + nl ");

  pList = ring_state_newvar(pState,"v1");
```

(continues on next page)

(continued from previous page)

```
  ring_list_setdouble(pList,RING_VAR_VALUE,10);

  pList = ring_state_newvar(pState,"v2");
  ring_list_setdouble(pList,RING_VAR_VALUE,20);

  ring_state_runcode(pState,"see 'v1 + v2 = ' see v1+v2 see nl");

  ring_state_runcode(pState,"see 'end of test' + nl");

  ring_state_delete(pState);
  ring_state_delete(pState2);
}
```

Output:

```
welcome
hello world from the ring programming language
Again from C we will call ring code
1
2
3
4
5
6
7
8
9
10
1
2
3
4
5
Now we will display the x variable value from ring code
x value : 11
x value : 6
Printing Ring variable value from C , 11
now we will set the ring variable value from C
x value after update : 20
v1 + v2 = 30
end of test
```

CHAPTER

NINETYEIGHT

CODE GENERATOR FOR WRAPPING C/C++ LIBRARIES

In this chapter we will learn how to use the code generator to wrap C/C++ Libraries to use it in our Ring applications.

98.1 Using the tool

The code generator program is parsec.ring that can be executed as any ring code using the ring language.

URL : https://github.com/ring-lang/ring/tree/master/extensions/codegen

for example to read a configuration file called test.cf to generate the source code file test.c run parsec.ring as in the next command

```
ring parsec.ring test.cf test.c
```

98.2 Configuration file

The configuration file (*.cf) is the input file that we pass to the code generator. This file determine the functions prototypes that we need to use from a C/C++ library.

Writing configuration files is simple according to the next rules

98.3 Using the function prototype

- To generate code that wraps a C function, we just write the C function prototype

Example:

```
ALLEGRO_DISPLAY *al_create_display(int w, int h)
void al_destroy_display(ALLEGRO_DISPLAY *display)
int al_get_new_display_flags(void)
void al_set_new_display_flags(int flags)
int al_get_new_display_option(int option, int *importance)
```

The previous example will guide the code generator to generate 5 functions that wraps the al_create_display(), al_destroy_display(), al_get_new_display_flags(), al_set_new_diplay_flas() and al_get_new_display_option() functions.

The generated code will be as in the next example

```
RING_FUNC(ring_al_create_display)
{
        if ( RING_API_PARACOUNT != 2 ) {
                RING_API_ERROR(RING_API_MISS2PARA);
                return ;
        }
        if ( ! RING_API_ISNUMBER(1) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        if ( ! RING_API_ISNUMBER(2) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        RING_API_RETCPOINTER(al_create_display( (int ) RING_API_GETNUMBER(1),
                                (int ) RING_API_GETNUMBER(2)),"ALLEGRO_DISPLAY");
}


RING_FUNC(ring_al_destroy_display)
{
  if ( RING_API_PARACOUNT != 1 ) {
        RING_API_ERROR(RING_API_MISS1PARA);
        return ;
  }
  if ( ! RING_API_ISPOINTER(1) ) {
        RING_API_ERROR(RING_API_BADPARATYPE);
        return ;
  }
  al_destroy_display((ALLEGRO_DISPLAY *) RING_API_GETCPOINTER(1,"ALLEGRO_DISPLAY"));
}


RING_FUNC(ring_al_get_new_display_flags)
{
        if ( RING_API_PARACOUNT != 0 ) {
                RING_API_ERROR(RING_API_BADPARACOUNT);
                return ;
        }
        RING_API_RETNUMBER(al_get_new_display_flags());
}


RING_FUNC(ring_al_set_new_display_flags)
{
        if ( RING_API_PARACOUNT != 1 ) {
                RING_API_ERROR(RING_API_MISS1PARA);
                return ;
        }
        if ( ! RING_API_ISNUMBER(1) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
```

(continues on next page)

(continued from previous page)

```
        al_set_new_display_flags( (int ) RING_API_GETNUMBER(1));
}


RING_FUNC(ring_al_get_new_display_option)
{
        if ( RING_API_PARACOUNT != 2 ) {
                RING_API_ERROR(RING_API_MISS2PARA);
                return ;
        }
        if ( ! RING_API_ISNUMBER(1) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        if ( ! RING_API_ISSTRING(2) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        RING_API_RETNUMBER(al_get_new_display_option( (int ) RING_API_GETNUMBER(1),
                                RING_API_GETINTPOINTER(2)));
        RING_API_ACCEPTINTVALUE(2) ;
}
```

from the previous example we can see how much of time and effort is saved using the Code Generator.

98.4 Adding code to the generated code

- To generate code directly type it between <code> and </code>

Example :

```
<code>
        /* some C code will be written here */
</code>
```

We use this feature when we need to do something without the help of the code generator. for example including header files and defining constants using Macro.

98.5 Prefix for Functions Names

- **To determine a prefix in all of the functions names type it between <funcstart> and </funcstart>**
  for example when we wrap the Allegro game programming library and we need all of the library functions to start with "al" we type the next code in the configuration file

```
<funcstart>
al
</funcstart>
```

98.6 Generate function to wrap structures

- To generate functions that wrap structures (create/delete/get structure members)

just type the structures names between <struct> and </struct> also after the structure name you can type the structure members between { } separated by comma.

Example

```
<struct>
ALLEGRO_COLOR
ALLEGRO_EVENT { type , keyboard.keycode , mouse.x , mouse.y }
</struct>
```

from the previous example we will generate two function to create/delete the structure ALLEGRO_COLOR Also we will generate two functions to create/delete the structure ALLEGRO_EVENT and four functions to get the structure ALLEGRO_EVENT members (type, keyboard.keycode, mouse.x, mouse.y).

98.7 Determine Structure Members Types

You can determine the pointer name before the structure member name.

Example:

```
SDL_Surface {flags,SDL_PixelFormat *format,w,h,pitch,void *pixels}
```

98.8 Defining Constants

You can define constants using <constant> and </constant>

The generator will generate the required functions to get the constant values

And will define the constants to be used with the same name in Ring code using *.rh file that will be generated too.

rh = Ring Header

Example:

```
<constant>
MIX_DEFAULT_FORMAT
SDL_QUIT
SDL_BUTTON_LEFT
SDL_BUTTON_MIDDLE
SDL_BUTTON_RIGHT
</constant>
```

Note: You will need to pass the *.rh file name to parsec.ring after the generated source file name.

Example:

```
ring ..\codegen\parsec.ring libsdl.cf ring_libsdl.c ring_libsdl.rh
```

98.9 Register New Functions

We can register functions by typing the function prototype between <register> and </register> We need this feature only when we don't provide the function prototype as input directly where we need to write the code of this function.

Example:

```
<register>
void al_exit(void)
</register>

<code>
RING_FUNC(ring_al_exit)
{
        if ( RING_API_PARACOUNT != 0 ) {
                RING_API_ERROR(RING_API_BADPARACOUNT);
                return ;
}
exit(0);
}
</code>
```

In the previous example we register the al_exit() function. This function is not part of the Allegro Library, it's just an extra function that we need to add. Then the code if this function is written inside <code> and </code>. This function call the exit() function from the C language library.

98.10 Writing comments in the configuration file

- To type comments just type it between <comment> and </comment>

Example:

```
<comment>
configuration files
</comment>
```

98.11 Executing code during code generation

- To ask from the code generator to execute Ring code during reading the configuration file, just

write the code between <runcode> and </runcode>

Example:

```
<runcode>
aNumberTypes + "al_fixed"
</runcode>
```

The previous line of code add the string "al_fixed" to the list aNumberTypes, This list contains types that can be considered as numbers when the code generator find it in the function prototype.

98.12 Enum and Numbers

We have the list aEnumTypes to use for adding each Enumeration we uses in the functions prototype.

Example:

```
<runcode>
aNumberTypes + "qreal"
aNumberTypes + "qint64"
aEnumTypes + "Qt::GestureType"
aEnumTypes + "Qt::GestureFlag"
</runcode>
```

98.13 Filtering using Expressions

using <filter> and </filter> we can include/exclude parts of the configuration file based on a condition, for example

```
<filter> iswindows()
        ... functions related to windows
</filter>
```

98.14 Constants Type

The default type for constant is Number But Some constants may be another type, for example (pointer : void *)

before using <constant> and </constant> we can use <runcode> and </runcode> to determine the constant type using two global variables used by the code generator.

The first variable is $nDefaultConstantType which can be * C_CONSTANT_TYPE_NUMBER * C_CONSTANT_TYPE_STRING * C_CONSTANT_TYPE_POINTER

if we are using C_CONSTANT_TYPE_POINTER then we will need the second global variable which is $cDefaultConstantPointerType to determine the pointer type.

Example :

The next example uses this feature to define constants in the FreeGLUT library

```
<runcode>
$nDefaultConstantType = C_CONSTANT_TYPE_POINTER
$cDefaultConstantPointerType = "void"
</runcode>
<constant>
        GLUT_STROKE_ROMAN
        GLUT_STROKE_MONO_ROMAN
        GLUT_BITMAP_9_BY_15
        GLUT_BITMAP_8_BY_13
        GLUT_BITMAP_TIMES_ROMAN_10
        GLUT_BITMAP_TIMES_ROMAN_24
        GLUT_BITMAP_HELVETICA_10
        GLUT_BITMAP_HELVETICA_12
```

(continues on next page)

(continued from previous page)

```
        GLUT_BITMAP_HELVETICA_18
</constant>
```

98.15 Configuration file for the Allegro Library

The next configuration file enable us to use the Allegro library functions. The configuration file size is less than 1000 lines. when the code generator take this file as input the generated source code file in the C language will be 12000 lines of code!

We can see this configuration file as a complete example about using the code generator Also we can use it to know the functions that can be used from RingAllegro when you use it to create 2D games!

```
<code>
#define ALLEGRO_NO_MAGIC_MAIN

#include <allegro5/allegro.h>
#include "allegro5/allegro_image.h"
#include <allegro5/allegro_font.h>
#include <allegro5/allegro_ttf.h>
#include <allegro5/allegro_audio.h>
#include <allegro5/allegro_acodec.h>
#include <allegro5/allegro_opengl.h>
#include <allegro5/allegro_direct3d.h>
#include <allegro5/allegro_color.h>
#include <allegro5/allegro_memfile.h>
#include "allegro5/allegro_native_dialog.h"
#include <allegro5/allegro_physfs.h>
#include <allegro5/allegro_primitives.h>
</code>

<funcstart>
al
</funcstart>

<struct>
ALLEGRO_EVENT { type , keyboard.keycode , mouse.x , mouse.y }
ALLEGRO_TIMEOUT
ALLEGRO_SAMPLE_ID
ALLEGRO_COLOR
</struct>

<register>
void al_exit(void)
</register>

<code>
RING_FUNC(ring_al_exit)
{
        if ( RING_API_PARACOUNT != 0 ) {
                RING_API_ERROR(RING_API_BADPARACOUNT);
                return ;
```

(continues on next page)

(continued from previous page)

```
        }
        exit(0);
}
</code>

int al_init(void)

<comment>
configuration files
</comment>

<runcode>
aNumberTypes + "al_fixed"
</runcode>

ALLEGRO_CONFIG *al_create_config(void)
void al_destroy_config(ALLEGRO_CONFIG *config)
ALLEGRO_CONFIG *al_load_config_file(const char *filename)
ALLEGRO_CONFIG *al_load_config_file_f(ALLEGRO_FILE *file)
bool al_save_config_file(const char *filename, const ALLEGRO_CONFIG *config)
bool al_save_config_file_f(ALLEGRO_FILE *file, const ALLEGRO_CONFIG *config)
void al_add_config_section(ALLEGRO_CONFIG *config, const char *name)
```

Note: we just provided part of the configuration file, for complete copy check the Ring source code distribution.

98.16 Threads Support

Next, another part of the configuration file, it's important because we can learn from it how to add threads to our Ring applications by using a threads library.

The idea is using ring_vm_mutexfunctions() and ring_vm_runcodefromthread() to execute Ring code.

```
<comment>
Threads
</comment>

<code>
void *al_func_thread(ALLEGRO_THREAD *thread, void *pPointer)
{
        List *pList;
        VM *pVM;
        const char *cStr;
        pList = (List *) pPointer ;
        pVM = (VM *) ring_list_getpointer(pList,2);
        cStr = ring_list_getstring(pList,1);
        ring_vm_runcodefromthread(pVM,cStr);
        ring_list_delete(pList);
        return NULL;
}
```

(continues on next page)

(continued from previous page)

```
RING_FUNC(ring_al_create_thread)
{
        ALLEGRO_THREAD *pThread;
        List *pList;
        if ( RING_API_PARACOUNT != 1 ) {
                RING_API_ERROR(RING_API_MISS1PARA);
                return ;
        }
        if ( ! RING_API_ISSTRING(1) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        pList = ring_list_new(0);
        ring_list_addstring(pList,RING_API_GETSTRING(1));
        ring_list_addpointer(pList,pPointer);
        ring_vm_mutexfunctions((VM *) pPointer,al_create_mutex,
                al_lock_mutex,al_unlock_mutex,al_destroy_mutex);
        pThread = al_create_thread(al_func_thread, pList);
        al_start_thread(pThread);
        RING_API_RETCPOINTER(pThread,"ALLEGRO_THREAD");
}

RING_FUNC(ring_al_run_detached_thread)
{
        List *pList;
        if ( RING_API_PARACOUNT != 1 ) {
                RING_API_ERROR(RING_API_MISS1PARA);
                return ;
        }
        if ( ! RING_API_ISSTRING(1) ) {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return ;
        }
        pList = ring_list_new(0);
        ring_list_addstring(pList,RING_API_GETSTRING(1));
        ring_list_addpointer(pList,pPointer);
        ring_vm_mutexfunctions((VM *) pPointer,al_create_mutex,
                al_lock_mutex,al_unlock_mutex,al_destroy_mutex);
        al_run_detached_thread(al_func_thread, pList);
}
</code>

<register>
ALLEGRO_THREAD *al_create_thread(void)
void al_run_detached_thread(void)
</register>

void al_start_thread(ALLEGRO_THREAD *thread)
void al_join_thread(ALLEGRO_THREAD *thread, void **ret_value)
void al_set_thread_should_stop(ALLEGRO_THREAD *thread)
bool al_get_thread_should_stop(ALLEGRO_THREAD *thread)
```

(continues on next page)

(continued from previous page)

```
void al_destroy_thread(ALLEGRO_THREAD *thread)
ALLEGRO_MUTEX *al_create_mutex(void)
ALLEGRO_MUTEX *al_create_mutex_recursive(void)
void al_lock_mutex(ALLEGRO_MUTEX *mutex)
void al_unlock_mutex(ALLEGRO_MUTEX *mutex)
void al_destroy_mutex(ALLEGRO_MUTEX *mutex)
ALLEGRO_COND *al_create_cond(void)
void al_destroy_cond(ALLEGRO_COND *cond)
void al_wait_cond(ALLEGRO_COND *cond, ALLEGRO_MUTEX *mutex)
```

98.17 Code Generator Rules for Wrapping C++ Classes

- We can define classes between <class> and </class>
- Between <class> and <class> we set attributes like "name, nonew, para, parent, codename, passvmpointer, abstract and staticmethods"
- we set the attributes using the style attributename:value or attributename only if no values are required
- The "name" attribute determine the class name in C++ code and this name will be the default name in the Ring code
- The nonew instruction means that we don't need new/delete methods
- The parent attribute determine the parent class name
- The codename attribute determine another class name in C++ code
- The passvmpoint instruction means passing the Ring VM pointer to the class constructor when we create new objects, this happens when we set the codename attribute to a class that we will define and this class need the Virtual Machine pointer (for example to use it to execute Ring code from C++ code).
- The abstract instruction means that no new method is required for this class "no objects will be created".
- The staticmethods instruction means that method doesn't need an object to be called.
- Using <nodllstartup> we can avoid #include "ring.h", We need this to write our startup code.
- Using <libinitfunc> we can change the function name that register the library functions
- Using <ignorecpointertype> we can ignore pointer type check
- Using the aStringTypes list when can defined new types that treated like const char *
- Using the aBeforeReturn list when can define code that is inserted after the variable name when we return that variable from a function
- Using the aNewMethodName list we can define another method name to be used in Ring code when we call the C++ method. this feature is required because some C++ method may be identical to Ring Keywords like "load","next","end" and "done".
- in method prototype - when we use @ in the method name, we mean that we have the same method with different parameters (As in C++)

98.18 Using configuration file that wrap C++ Library

To run the code generator to generate code for using C++ library in the Ring application, we can do that as we did with using C libraries but here we will generate *.cpp file instead of *.c file. Also we will determine another file to be generated (*.ring). This file will contains classes in Ring code that wraps C++ functions for using C++ classes and objects.

```
ring parsec.ring generator\qt.cf ring_qt.cpp ring_qt.ring
```

98.19 Configuration file for the Qt Framework

The next configuration file is used to wrap many Qt classes The configuration file is around 3500 lines and generate C++ code around 56000 lines and generate also Ring code around 9000 lines.

```
<nodllstartup>

<libinitfunc> ring_qt_start

<ignorecpointertype>

<code>

extern "C" {
        #include "ring.h"
}

#include "ring_qt.h"
#include "gpushbutton.h"
#include "gaction.h"
#include "glineedit.h"
#include "gtextedit.h"
#include "glistwidget.h"
#include "gtreeview.h"
#include "gtreewidget.h"
#include "gcombobox.h"
#include "gtabwidget.h"
#include "gtablewidget.h"
#include "gprogressbar.h"
#include "gspinbox.h"
#include "gslider.h"
#include "gdial.h"
#include "gwebview.h"
#include "gcheckbox.h"
#include "gradiobutton.h"
#include "gbuttongroup.h"
#include "gvideowidget.h"
#include "gtimer.h"
#include "gtcpserver.h"
#include "giodevice.h"
#include "gabstractsocket.h"
#include "gtcpsocket.h"
```

(continues on next page)

(continued from previous page)

```
#include "gcolordialog.h"
#include "gallevents.h"
#include <QApplication>
#include <QObject>
#include <QWidget>
#include <QLabel>
#include <QPixmap>
#include <QIcon>
#include <QSize>
#include <QPushButton>
#include <QMainWindow>
#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QLineEdit>
#include <QTextEdit>
#include <QListWidget>
#include <QTreeView>
#include <QDir>
#include <QFileSystemModel>
#include <QTreeWidget>
#include <QTreeWidgetItem>
#include <QComboBox>
#include <QVariant>
#include <QMenuBar>
#include <QMenu>
#include <QToolBar>
#include <QMainWindow>
#include <QStatusBar>
#include <QDockWidget>
#include <QTabWidget>
#include <QTableWidget>
#include <QTableWidgetItem>
#include <QSizePolicy>
#include <QFrame>
#include <QAbstractScrollArea>
#include <QAbstractItemView>
#include <QProgressBar>
#include <QSpinBox>
#include <QSlider>
#include <QAbstractSlider>
#include <QDateEdit>
#include <QDateTimeEdit>
#include <QAbstractSpinBox>
#include <QDial>
#include <QWebView>
#include <QUrl>
#include <QCheckBox>
#include <QRadioButton>
#include <QButtonGroup>
#include <QMediaPlayer>
#include <QMediaPlaylist>
#include <QVideoWidget>
```

(continues on next page)

(continued from previous page)

```
#include <QPrinter>
#include <QAction>
#include <QEvent>
#include <QMessageBox>
#include <QTimer>
#include <QFileDialog>
#include <QPainter>
#include <QPicture>
#include <QPen>
#include <QColor>
#include <QPrinter>
#include <QFont>
#include <QWebSettings>
#include <QBrush>
#include <QByteArray>
#include <QIODevice>
#include <QAbstractSocket>
#include <QTcpSocket>
#include <QTcpServer>
#include <QNetworkProxy>
#include <QHostAddress>
#include <QHostInfo>
#include <QList>
#include <QFileInfo>
#include <QDirModel>
#include <QModelIndex>
#include <QFontDialog>
#include <QDialog>
#include <QTextCursor>
#include <QTextBlock>
#include <QTextDocumentFragment>
#include <QColorDialog>
#include <QHeaderView>
#include <QStringList>
#include <QKeySequence>
#include <QLCDNumber>
#include <QInputDialog>
#include <QDesktopWidget>
#include <QRect>
#include <QTextDocument>

extern "C" {

        #define RING_DLL __declspec(dllexport)

        RING_DLL void ringlib_init(RingState *pRingState)
        {

                new QApplication(pRingState->argc,pRingState->argv);
                ring_qt_start(pRingState) ;
        }
```

(continues on next page)

(continued from previous page)

```
}
</code>


<runcode>
aStringTypes + "QString"
aBeforeReturn + ["QString",".toStdString().c_str()"]
aNewMethodName + ["QWebView","load","loadpage"]
aNewMethodName + ["QMediaPlaylist","load","loadfile"]
aNewMethodName + ["QMediaPlaylist","next","movenext"]
aNewMethodName + ["QPainter","end","endpaint"]
aNewMethodName + ["QPicture","load","loadfile"]
aNewMethodName + ["QLineEdit","end","endtext"]
aNewMethodName + ["QDialog","done","donedialog"]
aNewMethodName + ["QTextDocument","end","enddoc"]
aNewMethodName + ["QTextBlock","next","nextblock"]
</runcode>

<class>
name: qApp
nonew
</class>

<register>
void exec(void)
void quit(void)
void processEvents(void)
</register>

<code>

RING_FUNC(ring_qApp_quit)
{
        qApp->quit();
}

RING_FUNC(ring_qApp_exec)
{
        qApp->exec();
}

RING_FUNC(ring_qApp_processEvents)
{
        qApp->processEvents();
}

</code>

<class>
name: QObject
para: void
</class>
```

(continues on next page)

(continued from previous page)

```
bool blockSignals(bool block)
QObjectList children(void)
void dumpObjectInfo(void)
void dumpObjectTree(void)
bool inherits(const char *className)
void installEventFilter(QObject *filterObj)
bool isWidgetType(void)
void killTimer(int id)
void moveToThread(QThread *targetThread)
QString objectName(void)
QObject *parent(void)
QVariant property(const char *name)
void removeEventFilter(QObject *obj)
void setObjectName(QString)
void setParent(QObject *parent)
bool setProperty(const char *name, QVariant)
bool signalsBlocked(void)
int startTimer(int interval)
QThread *thread(void)
void deleteLater(void)

<class>
name: QWidget
para: void
parent: QObject
</class>

bool acceptDrops(void)
QString accessibleDescription(void)
QString accessibleName(void)
void activateWindow(void)
void addAction(QAction *action)
void adjustSize(void)
bool autoFillBackground(void)
int backgroundRole(void)
QSize baseSize(void)
QWidget *childAt(int x, int y)
QRect childrenRect(void)
QRegion childrenRegion(void)
void clearFocus(void)
void clearMask(void)
QMargins contentsMargins(void)
QRect contentsRect(void)
int contextMenuPolicy(void)
QCursor cursor(void)
int effectiveWinId(void)
void ensurePolished(void)
int focusPolicy(void)
QWidget *focusProxy(void)
QWidget *focusWidget(void)
QFont font(void)
```

(continues on next page)

(continued from previous page)

```
QFontInfo fontInfo(void)
QFontMetrics fontMetrics(void)
int foregroundRole(void)
QRect frameGeometry(void)
QSize frameSize(void)
QRect geometry(void)
void getContentsMargins(int *left, int *top, int *right, int *bottom)
void grabGesture(int gesture, int flags)
void grabKeyboard(void)
void grabMouse(void)
int grabShortcut(QKeySequence , int context)
QGraphicsEffect *graphicsEffect(void)
QGraphicsProxyWidget *graphicsProxyWidget(void)
bool hasFocus(void)
bool hasMouseTracking(void)
int height(void)
int heightForWidth(int w)
int inputMethodHints(void)
QVariant inputMethodQuery(int query)
void insertAction(QAction *before, QAction *action)
bool isActiveWindow(void)
bool isAncestorOf(QWidget *child)
bool isEnabled(void)
bool isEnabledTo(QWidget *ancestor)
bool isFullScreen(void)
bool isHidden(void)
bool isMaximized(void)
bool isMinimized(void)
bool isModal(void)
bool isVisible(void)
bool isVisibleTo(QWidget *ancestor)
bool isWindow(void)
bool isWindowModified(void)
QLayout *layout(void)
int layoutDirection(void)
QLocale locale(void)
QPoint mapFrom(QWidget *parent, QPoint)
QPoint mapFromGlobal(QPoint)
QPoint mapFromParent(QPoint)
QPoint mapTo(QWidget *parent, QPoint)
QPoint mapToGlobal(QPoint pos)
QPoint mapToParent(QPoint pos)
QRegion mask(void)
int maximumHeight(void)
QSize maximumSize(void)
int maximumWidth(void)
int minimumHeight(void)
QSize minimumSize(void)
int minimumWidth(void)
void move(int x, int y)
QWidget *nativeParentWidget(void)
QWidget *nextInFocusChain(void)
```

(continues on next page)

(continued from previous page)

```
QRect normalGeometry(void)
void overrideWindowFlags(int flags)
QPalette palette(void)
QWidget *parentWidget(void)
QPoint pos(void)
QWidget *previousInFocusChain(void)
QRect rect(void)
void releaseKeyboard(void)
void releaseMouse(void)
void releaseShortcut(int id)
void removeAction(QAction *action)
void render(QPaintDevice *target, QPoint,QRegion, int)
void repaint(int x, int y, int w, int h)
void resize(int w, int h)
bool restoreGeometry(QByteArray)
QByteArray saveGeometry(void)
void scroll(int dx, int dy)
void setAcceptDrops(bool on)
void setAccessibleDescription(QString)
void setAccessibleName(QString)
void setAttribute(int attribute, bool on)
void setAutoFillBackground(bool enabled)
void setBackgroundRole(int role)
void setBaseSize(int basew, int baseh)
void setContentsMargins(int left, int top, int right, int bottom)
void setContextMenuPolicy(int policy)
void setCursor(QCursor)
void setFixedHeight(int h)
void setFixedSize(int w, int h)
void setFixedWidth(int w)
void setFocus(int reason)
void setFocusPolicy(int policy)
void setFocusProxy(QWidget *w)
void setFont(QFont)
void setForegroundRole(int role)
void setGeometry(int x, int y, int w, int h)
void setGraphicsEffect(QGraphicsEffect *effect)
void setInputMethodHints(int hints)
void setLayout(QLayout *layout)
void setLayoutDirection(int direction)
void setLocale(QLocale)
void setMask(QBitmap)
void setMaximumHeight(int maxh)
void setMaximumSize(int maxw, int maxh)
void setMaximumWidth(int maxw)
void setMinimumHeight(int minh)
void setMinimumSize(int minw, int minh)
void setMinimumWidth(int minw)
void setMouseTracking(bool enable)
void setPalette(QPalette)
void setParent(QWidget *parent)
void setShortcutAutoRepeat(int id, bool enable)
```

(continues on next page)

(continued from previous page)

```
void setShortcutEnabled(int id, bool enable)
void setSizeIncrement(int w, int h)
void setSizePolicy(int horizontal, int vertical)
void setStatusTip(QString)
void setStyle(QStyle *style)
void setToolTip(QString)
void setUpdatesEnabled(bool enable)
void setWhatsThis(QString)
void setWindowFilePath(QString)
void setWindowFlags(int type)
void setWindowIcon(QIcon)
void setWindowIconText(QString)
void setWindowModality(int windowModality)
void setWindowOpacity(double level)
void setWindowRole(QString)
void setWindowState(int windowState)
QSize size(void)
QSize sizeIncrement(void)
QSizePolicy sizePolicy(void)
void stackUnder(QWidget *w)
QString statusTip(void)
QStyle *style(void)
QString styleSheet(void)
bool testAttribute(int attribute)
QString toolTip(void)
bool underMouse(void)
void ungrabGesture(int gesture)
void unsetCursor(void)
void unsetLayoutDirection(void)
void unsetLocale(void)
void update(int x, int y, int w, int h)
void updateGeometry(void)
bool updatesEnabled(void)
QRegion visibleRegion(void)
QString whatsThis(void)
int width(void)
int winId(void)
QWidget *window(void)
QString windowFilePath(void)
int windowFlags(void)
QIcon windowIcon(void)
QString windowIconText(void)
int windowModality(void)
double windowOpacity(void)
QString windowRole(void)
int windowState(void)
QString windowTitle(void)
int windowType(void)
int x(void)
int y(void)
bool close(void)
void hide(void)
```

(continues on next page)

(continued from previous page)

```
void lower(void)
void raise(void)
void setDisabled(bool disable)
void setEnabled(bool)
void setHidden(bool hidden)
void setStyleSheet(QString)
void setWindowModified(bool)
void setWindowTitle(QString)
void show(void)
void showFullScreen(void)
void showMaximized(void)
void showMinimized(void)
void showNormal(void)
QWidget *find(int id)
QWidget *keyboardGrabber(void)
QWidget *mouseGrabber(void)
void setTabOrder(QWidget *first, QWidget *second)

<class>
name: QLabel
para: QWidget *
parent: QWidget
</class>

int alignment(void)
QWidget *buddy(void)
bool hasScaledContents(void)
bool hasSelectedText(void)
int indent(void)
int margin(void)
QMovie *movie(void)
bool openExternalLinks(void)
QPicture *picture(void)
QPixmap *pixmap(void)
QString selectedText(void)
int selectionStart(void)
void setAlignment(int)
void setBuddy(QWidget *buddy)
void setIndent(int)
void setMargin(int)
void setOpenExternalLinks(bool open)
void setScaledContents(bool)
void setSelection(int start, int length)
void setTextFormat(int)
void setTextInteractionFlags(int flags)
void setWordWrap(bool on)
QString text(void)
int textFormat(void)
int textInteractionFlags(void)
bool wordWrap(void)
void clear(void)
void setMovie(QMovie *movie)
```

(continues on next page)

(continued from previous page)

```
void setNum(double num)
void setPicture(QPicture)
void setPixmap(QPixmap)
void setText(QString)

<class>
name: QPushButton
para: QWidget *
parent: QWidget
codename: GPushButton
passvmpointer
</class>

void setText(const char *)
void setClickEvent(const char *)
void setIcon(QIcon)
void setIconSize(QSize)

<class>
name: QLineEdit
para: QWidget *
parent: QWidget
codename: GLineEdit
passvmpointer
</class>

int alignment(void)
void backspace(void)
QCompleter *completer(void)
QMenu *createStandardContextMenu(void)
void cursorBackward(bool mark, int steps)
void cursorForward(bool mark, int steps)
int cursorMoveStyle(void)
int cursorPosition(void)
int cursorPositionAt(QPoint)
void cursorWordBackward(bool mark)
void cursorWordForward(bool mark)
void del(void)
void deselect(void)
QString displayText(void)
bool dragEnabled(void)
int echoMode(void)
void end(bool mark)
void getTextMargins(int *left, int *top, int *right, int *bottom)
bool hasAcceptableInput(void)
bool hasFrame(void)
bool hasSelectedText(void)
void home(bool mark)
QString inputMask(void)
void insert(QString)
bool isModified(void)
bool isReadOnly(void)
```

(continues on next page)

(continued from previous page)

```
bool isRedoAvailable(void)
bool isUndoAvailable(void)
int maxLength(void)
QString placeholderText(void)
QString selectedText(void)
int selectionStart(void)
void setAlignment(int flag)
void setCompleter(QCompleter *c)
void setCursorMoveStyle(int style)
void setCursorPosition(int)
void setDragEnabled(bool b)
void setEchoMode(int)
void setFrame(bool)
void setInputMask(QString)
void setMaxLength(int)
void setModified(bool)
void setPlaceholderText(QString)
void setReadOnly(bool)
void setSelection(int start, int length)
void setTextMargins(int left, int top, int right, int bottom)
void setValidator(QValidator *v)
QString text(void)
QMargins textMargins(void)
QValidator *validator(void)

void clear(void)
void copy(void)
void cut(void)
void paste(void)
void redo(void)
void selectAll(void)
void setText(QString)
void undo(void)

void setTextChangedEvent(const char *)
void setcursorPositionChangedEvent(const char *)
void seteditingFinishedEvent(const char *)
void setreturnPressedEvent(const char *)
void setselectionChangedEvent(const char *)
void settextEditedEvent(const char *)
```

Note: Most of the content of the previous configuration file is removed from this documentation, for a complete version see the Ring source code distribution.

98.20 Static Methods

Starting from Ring 1.8 the code generator support the staticmethods option.

So the code generator can know that the class doesn't need an object to call the methods.

Example:

```
<class>
name: QStandardPaths
para: void
nonew
staticmethods
</class>

QString displayName(QStandardPaths::StandardLocation type)
QString findExecutable(QString executableName, QStringList paths))
```

98.21 Loading Files

Starting from Ring 1.9 the code generator support the <loadfile> command.

```
<loadfile> filename.cf
```

This is useful to separate the extension configuration file to many files

Example:

The file : qt_module_network.cf in the RingQt Extension

```
<comment>
                                Module (network)
</comment>

<loadfile> qabstractsocket.cf
<loadfile> qnetworkproxy.cf
<loadfile> qtcpsocket.cf
<loadfile> qtcpserver.cf
<loadfile> qhostaddress.cf
<loadfile> qhostinfo.cf
<loadfile> qnetworkrequest.cf
<loadfile> qnetworkaccessmanager.cf
<loadfile> qnetworkreply.cf
```

98.22 Managed Classes

Starting from Ring 1.9 the code generator support the <managed> option when defining classes.

Using this option, the generator will use RING_API_RETMANAGEDCPOINTER() to return the C pointer.

So the Garbage Collector will manage these C pointers.

Example

```
<class>
name: QFont
para: QString, int, int, bool
managed
</class>
```

98.23 Extra names for functions

Starting from Ring 1.19

The code generator is updated and support defining extra names for functions

Using the predefined list (aExtraFunctionName) we can define the extra names

Example:

To define vec2() as another name for raylib_new_managed_vector2()

```
<runcode>
aExtraFunctionName + ["vec2","raylib_new_managed_vector2"]
aExtraFunctionName + ["vec2setx","raylib_set_vector2_x"]
aExtraFunctionName + ["vec2sety","raylib_set_vector2_y"]
</runcode>
```

98.24 Configuration Files Examples

You can learn from the next examples

- RingAllegro : https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/allegro.cf
- RingQt : https://github.com/ring-lang/ring/blob/master/extensions/ringqt/classes/qt.cf
- RingLibSDL : https://github.com/ring-lang/ring/blob/master/extensions/ringsdl/libsdl.cf

After modifying the configuration file, You will need to generate the code, You can learn from the next examples

- RingAllegro : https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/gencode.bat
- RingQt : https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencode.bat
- RingLibSDL : https://github.com/ring-lang/ring/blob/master/extensions/ringsdl/gencode.bat

After generating the code, You will need to build the library, You can learn from the next examples

- RingAllegro : https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/buildvc.bat
- RingQt : https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildmingw32.bat

- RingLibSDL : https://github.com/ring-lang/ring/blob/master/extensions/ringsdl/buildvc.bat

CHAPTER

NINETYNINE

CREATE YOUR FIRST EXTENSION

In this chapter we will create RingBeep

RingBeep is a simple extension for the beep() function in Windows API

Just a simple example about creating extensions

99.1 Location

You will find this extension in the ring/extensions/ringbeep folder

99.2 Steps to create the extension

At first we write the configuration file in : ringbeep.cf

```
notepad ringbeep.cf
```

The file ringbeep.cf contains

```
<code>
#include "windows.h"
</code>

int Beep(int dwFreq,int dwDuration)
```

Then we run : gencode.bat to generate ringbeep.c

```
gencode.bat
```

The gencode.bat contains the next command to call Ring Extensions Generator

```
ring ..\codegen\parsec.ring ringbeep.cf ringbeep.c
```

Then we build the extension using :

```
buildvc.bat
```

The file buildvc.bat contains the next commands to build the extension using Visual C/C++

```
cls
call ../../src/locatevc.bat
cl /c /DEBUG ringbeep.c -I"..\..\include"
link /DEBUG ringbeep.obj  ..\..\lib\ring.lib kernel32.lib /DLL /OUT:ringbeep.dll ^
                                                        /SUBSYSTEM:CONSOLE,"5.01"
del ringbeep.obj
```

99.3 Testing the extension

Then we test the function using

```
ring test.ring
```

The file test.ring contains

```
loadlib("ringbeep.dll")

for f = 750 to 1000 step 50
        beep(f,300)
next
```

CHAPTER

LANGUAGE DESIGN

In this chapter we will learn about the basic concepts behind the language design.

100.1 Why Ring?

The language is simple, trying to be natural, encourage organization and comes with transparent and visual implementation. It comes with compact syntax and a group of features that enable the programmer to create natural interfaces and declarative domain-specific languages in a fraction of time. It is very small, fast and comes with smart garbage collector that puts the memory under the programmer control. It supports many programming paradigms, comes with useful and practical libraries. The language is designed for productivity and developing high quality solutions that can scale.

100.2 Designed for a Clear Goal

- Applications programming language.
- Productivity and developing high quality solutions that can scale.
- Small and fast language that can be embedded in C/C++ projects.
- Simple language that can be used in education and introducing Compiler/VM concepts.
- General-Purpose language that can be used for creating domain-specific libraries, frameworks and tools.
- Practical language designed for creating the next version of the Programming Without Coding Technology software.

100.3 Simple

Ring is a very simple language, and has a very straightforward syntax. It encourages programmers to program without boilerplate code

```
See "Hello, World!"
```

The Main function is optional and will be executed after the statements, and is useful for using the local scope.

```
Func Main
        See "Hello, World!"
```

Uses Dynamic Typing and Lexical scoping. No $ is required before the variable name! You can use the '+' operator for string concatenation and the language is weakly typed and will convert automatically between numbers and strings based on the context.

```
nCount = 10     # Global variable
Func Main
        nID = 1 # Local variable
        See "Count = " + nCount + nl + " ID = " + nID
```

100.4 Trying to be natural

Ring is not case-sensitive

```
See "Enter your name ? "
Give name
See "Hello " + Name     # Name is the same as name
```

The list index starts from 1

```
aList = ["one","two","three"]
See aList[1]    # print one
```

Call functions before definition

```
one()
two()
three()
Func one
        See "One" + nl
Func two
        See "two" + nl
Func three
        See "three" + nl
```

The assignment operator uses Deep copy (no references in this operation)

```
aList = ["one","two","three"]
aList2 = aList
aList[1] = 1
see alist[1]    # print 1
see aList2[1]   # print one
```

Pass numbers and strings by value, but pass lists and objects by reference. The for in loop can update the list items.

```
Func Main
        aList = [1,2,3]
        update(aList)
        see aList       # print one two three

Func update aList
        for x in aList
                switch x
                on 1 x = "one"
                on 2 x = "two"
                on 3 x = "three"
```

(continues on next page)

(continued from previous page)

```
                off
        next
```

Using Lists during definition

```
aList = [ [1,2,3,4,5] , aList[1] , aList[1] ]
see aList       # print 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

Exit from more than one loop

```
for x = 1 to 10
                for y = 1 to 10
                                see "x=" + x + " y=" + y + nl
                                if x = 3 and y = 5
                                                exit 2     # exit from 2 loops
                                ok
                next
next
```

100.5 Encourage Organization

The language encourage organization, Forget bad days using languages where the programmer start with function then class then function and a strange mix between things!

Each source file follow the next structure

- Load Files
- Statements and Global Variables
- Functions
- Packages and Classes

This enable us to use Packages, Classes and Functions without the need to use a keyword to end these components.

We can write one line comments and multi-line comments The comment starts with # or // Multi-line comments are written between /* and */

```
/*
        Program Name : My first program using Ring
        Date         : 2015.05.08
*/

See "What is your name? "       # print message on screen
give cName                      # get input from the user
see "Hello " + cName            # say hello!

// See "Bye!"
```

100.6 Data Representation

Ring contains only 4 types that represent the program data

These types are (String, Number, List & Object)

The idea is to have many use cases for each type which increase the flexibility and the ability to write functions that are more usable in different situations.

The String type is used to represent: * One character * A string of many characters * Multi-line string * Binary Data

```
cStr1 = "a"                     # One character
cStr2 = "Hello, World!"         # A string of many characters
cStr3 = "Hello
Welcome to the Ring language!
"                               # Multi-line string
cStr4 = read(EXEFileName())     # Read executable file (Binary Data)
```

The Number type is used to represent * Boolean Values * Signed/Unsigned Integers * Float/Double

```
nNum1 = True            # Boolean Value (1)
nNum2 = False           # Boolean Value (0)
nNum3 = 10              # Integer
nNum4 = -10             # Signed Integer
nNum5 = 1250.11         # Float/Double
```

The List type is used instead of * One Dimension Arrays * Multi-Dimension Arrays * Lists of multiple types * Nested Lists * Hash Tables (Key & Value) * Tree * Wrapper around a C Pointer

```
aList1 = ["one","two","three"]                          # Strings
aList2 = [1,2,3,4,5,6,7,8,9,10]                         # Numbers
aList3 = ["Ring",1234]                                  # Multiple types
aList4 = [["Fayed","Egypt"],["Mansour","Tunisia"]]      # Nested Lists
aList5 = [ :name = "Fayed", :country = "Egypt"]         # Hash Tables
```

The Object type is used to represent objects created from classes

Using classes and operator overloading we can create custom types

100.7 Compact Syntax

The language is not line sensitive, you don't need to write ; after statements, also you don't need to press ENTER or TAB, so we can write the next code

```
See "The First Message" See " Another message in the same line! " + nl
See "Enter your name?" Give Name See "Hello " + Name
```

The next code create a class called Point contains three attributes X,Y and Z. No keywords is used to end the package/class/function definition. Also, we can write the attributes names directly below the class name.

```
Class Point X Y Z
```

We can use classes and functions before their definition, In this example we will create new object, set the object attributes then print the object values.

```
o1 = New point  o1.x=10    o1.y=20   o1.z=30    See O1  Class Point X Y Z
```

Instead of using the dot '.' operator to access the object attributes and methods we can use braces { } to access the object, then we can use the object attributes and methods.

```
o1 = New point { x=10 y=20 z=30 } See O1  Class Point X Y Z
```

Now we will call a method after accessing the object using { }

```
oPerson = new Person
{
        Name = "Somebody"
        Address = "Somewhere"
        Phone = "0000000"
        Print()                 # here we call the Print() method
}
Class Person Name Address Phone
        Func Print
                See "Name :" + name + nl +
                        "Address :" + Address + nl +
                        "Phone : " + phone + nl
```

When we use { } to access the object then write any attribute name, the language will check the class for any setter/getter methods that will be called automatically.

```
New Number {
                See one         # Execute GetOne()
                See two         # Execute GetTwo()
                See three       # Execute GetThree()
}
Class Number one two three
        Func GetOne
                See "Number : One" + nl
                return 1
        Func GetTwo
                See "Number : Two" + nl
                return 2
        Func GetThree
                See "Number : Three" + nl
                return 3
```

100.8 Define Natural Statements

After the object access using { } if the class contains a method called BraceEnd() it will be executed!

```
TimeForFun = new journey
# The first surprise!
TimeForFun {
        Hello it is me          # What a beautiful programming world!
}
# Our Class
```

(continues on next page)

(continued from previous page)

```
Class journey
        hello=0 it=0 is=0 me=0
        func GetHello
                See "Hello" + nl
        func braceEnd
                See "Goodbye!" + nl
```

We can execute code written in strings using the Eval() function

```
cCode = "See 'Code that will be executed later!' "
Eval(cCode)     # execute the code to print the message
```

We can create a list then execute code generated from that list

```
aWords = ["hello","it","is","me"]
for word in aWords cCode=word+"=0" eval(cCode) next
```

We can read text files using the Read(cFileName) function and we can write files using the Write(cFileName,cString) function.

```
See "Enter File Name:" Give cFileName See Read(cFileName) # Print the file content
```

The next example presents how to create a class that defines two instructions The first instruction is : I want window The second instruction is : Window title = Expression Also keywords that can be ignored like the 'the' keyword

```
New App
{
        I want window
        The window title = "hello world"
}

Class App

        # Attributes for the instruction I want window
                i want window
                nIwantwindow = 0
        # Attributes for the instruction Window title
        # Here we don't define the window attribute again
                title
                nWindowTitle = 0
        # Keywords to ignore, just give them any value
                the=0

        func geti
                        if nIwantwindow = 0
                                nIwantwindow++
                        ok

        func getwant
                        if nIwantwindow = 1
                                nIwantwindow++
                        ok
```

(continues on next page)

(continued from previous page)

```
        func getwindow
                        if nIwantwindow = 2
                                nIwantwindow= 0
                                see "Instruction : I want window" + nl
                        ok
                        if nWindowTitle = 0
                                nWindowTitle++
                        ok

        func settitle cValue
                        if nWindowTitle = 1
                                nWindowTitle=0
                                see "Instruction : Window Title = " + cValue + nl
                        ok
```

To complete the previous example, use read() to get the content of a file that contains

```
I want window
The window title = "hello world"
```

Then use eval() to execute the content of that file!. Also, you can update the methods GetWindow() and SetTitle() to create Real windows using the GUI Library

100.9 Define Declarative Languages

We learned how to use Natural statements to execute our code and using the same features we can use nested structures to execute our code.

The next example from the Web library, generate HTML document using the Bootstrap library. No HTML code is written directly in this example, we created a similar language (just as example) Then using this declarative language that uses nested structures, we generated the HTML Document.. The idea in this example is that the GetDiv() and GetH1() methods return an object that we can access using { } and after each object access the method BraceEnd() will be executed to send the generated HTML to the parent object until we reach to the root where BraceEnd() will print the output.

```
Load "weblib.ring"
Import System.Web

Func Main

  BootStrapWebPage()
  {
        div
        {
          classname = :container
          div
          {
                classname = :jumbotron
                H1 {   text("Bootstrap Page")   }
          }
          div
```

(continues on next page)

(continued from previous page)

```
        {
                classname = :row
                for x = 1 to 3
                  div
                  {
                    classname = "col-sm-4"
                    H3 { html("Welcome to the Ring programming language") }
                    P  { html("Using a scripting language is very fun!") }
                  }
                next
        }
      }
  }
```

The classes that power the declarative interface looks like this

```
Class Link from ObjsBase
        title  link
        Func braceend
                cOutput = nl+GetTabs() + "<a href='" +
                          Link + "'> "+ Title + " </a> " + nl

Class Div from ObjsBase
        Func braceend
                cOutput += nl+'<div'
                addattributes()
                AddStyle()
                getobjsdata()
                cOutput += nl+"</div>" + nl
                cOutput = TabMLString(cOutput)
```

100.10 Transparent Implementation

Ring comes with transparent implementation. We can know what is happening in each compiler stage and what is going on during the run-time by the Virtual Machine Example : ring helloworld.ring -tokens -rules -ic

```
See "Hello, World!"
```

Output

```
=================================================================
Tokens - Generated by the Scanner
=================================================================

   Keyword : SEE
   Literal : Hello, World!
   EndLine

=================================================================
```

(continues on next page)

(continued from previous page)

```
=================================================================
Grammar Rules Used by The Parser
=================================================================

Rule : Program --> {Statement}

Line 1
Rule : Factor --> Literal
Rule : Range --> Factor
Rule : Term --> Range
Rule : Arithmetic --> Term
Rule : BitShift --> Arithmetic
Rule : BitAnd --> BitShift
Rule : BitOrXOR -->  BitAnd
Rule : Compare --> BitOrXOR
Rule : EqualOrNot --> Compare
Rule : LogicNot -> EqualOrNot
Rule : Expr --> LogicNot
Rule : Statement  --> 'See' Expr

=================================================================




=================================================================
Byte Code - Before Execution by the VM
=================================================================

         PC      OPCode        Data

          1     FuncExE
          2       PushC   Hello, World!
          3       Print
          4  ReturnNull

=================================================================

Hello, World!
```

100.11 Visual Implementation

The Ring programming language is designed using the PWCT visual programming tool and you will find the visual source of the language in the folder "language/visualsrc" - *.ssf files and the generated source code (In the C Language) in the language/src folder and the language/include folder.

The next screen shot from the ring_vm.ssf file (Generate ring_vm.c and ring_vm.h)

The next screen shot from the ring_list.ssf file (Generate ring_list.c and ring_list.h)

100.12 Smart Garbage Collector

Avoid memory problems :-

- Invalid Memory Access
- Memory leaks
- Uninitialized Memory Access
- Dangling pointer

Rules :-

- Global variables always stay in the memory, until you delete these variables using the assignment statement.
- Local variables always deleted after the end of the function.
- The programmer have full control on when to delete the variable from the memory using the Assignment statement.

Example:

```
aList = [1,2,3,4,5]
aList = "nice"
```

After the second line directly, The list [1,2,3,4,5] will be deleted from the memory and we will have a string "nice"

- The programmer can call the function callgc() to force running the garbage collector.
- If we have a reference to a variable (when we pass objects and lists to functions), then deleting variables will be based on reference counting, if no references everything will be deleted, but if we have a reference, the data will stay in memory.

100.13 No Global Interpreter (VM) Lock - No GIL

When we use threads in Ring applications, We don't have global interpreter (VM) lock (No GIL)

So threads can work in parallel and execute Ring instructions at the same time

This is better for threads and concurrency (More Faster!)

100.14 Fast Enough For Many Applications

Ring is designed to be a simple, small and flexible language in the first place, but also it is fast enough for many applications.

Also when we need more speed we can use C/C++ extensions!

CHAPTER

ONE

WHAT IS NEW IN RING 1.1?

In this chapter we will learn about the changes and new features in Ring 1.1 release.

101.1 List of changes and new features

Ring 1.1 comes with many new features

- Better Natural Language Programming Support
- Generate/Execute Ring Object Files (*.ringo)
- Syntax Flexibility and different styles for I/O and Control Structures
- New Functions and Changes
- StdLib functions and classes written in Ring
- RingLibSDL
- Demo Project - Game Engine for 2D Games
- RingSQLite
- Better Code Generator for Extensions
- Using Self.Attribute in the Class Region to define new attributes
- Using This.Attribute in nested Braces inside the Class Methods
- Better Documentation

101.2 Better Natural Language Programming Support

Ring is an innovative language because of it's compact syntax, smart implementation (small, transparent & visual) and it's ability to create declarative and natural domain specific languages in a fraction of time.

This release add support for calling methods when an expression is evaluated

check this example:

```
# Natural Code
new program {
        Accept 2 numbers then print the sum
}
```

(continues on next page)

(continued from previous page)

```
# Natural Code Implementation
class program
        # Keywords
                Accept=0 numbers=0 then=0 print=0 the=0 sum=0

        # Execution
        func braceexpreval x
                value = x
        func getnumbers
                for x=1 to value
                        see "Enter Number ("+x+") :" give nNumber
                        aNumbers + nNumber
                next
        func getsum
                nSUm = 0
                for x in aNumbers nSum+= x next
                see "The Sum : " + nSum
        private
                value=0 aNumbers=[]
```

Output:

```
Enter Number (1) :3
Enter Number (2) :4
The Sum : 7
```

for more information see the "Natural Language Programming" chapter.

101.3 Generate/Execute Ring Object Files (*.ringo)

This feature enable you to distribute your applications without distributing the source code. Also it makes application distribution a simple process where you get one Ring object file for the complete project (many source code files). Also using Ring object file remove the loading time required for compiling the application.

Check the "command line options" chapter to know more about this feature.

101.4 Syntax Flexibility and different styles for I/O and Control Structures

Programmers are sensitive to the programming language syntax. Great programmers know how to work using many different styles but each programmer may have his/her favorite style.

Each programming language comes with a style that you may like or not. Ring is just one of these languages, but as a response to many programmers asking for a better syntax we decided to provide more options.

Also some of these features are very necessary for Natural Language Programming.

Example :

We have two commands to change language keywords and operators.

```
ChangeRingOperator + plus
ChangeRingKeyword see print

Print 5 plus 5

ChangeRingOperator plus +
ChangeRingKeyword print see
```

We have new styles (Optional) for Input/Output.

Example :

```
Put "What is your name? "
Get cName
Put "Hello " + cName
```

Example :

```
Load "stdlib.ring"

Print("What is your name? ")     # print message on screen
cName=GetString()                # get input from the user
print("Hello #{cName}")          # say hello!
```

We have new styles (optional) for control structures.

Example :

```
While True

        Put "
                Main Menu
                ---------
                (1) Say Hello
                (2) About
                (3) Exit

            " Get nOption

        Switch nOption
        Case 1
                Put "Enter your name : "
                Get name
                Put "Hello " + name + nl
        Case 2
                Put "Sample : using while loop" + nl
        Case 3
                Bye
        Else
                Put "bad option..." + nl
        End
End
```

Example :

```
Load "stdlib.ring"

While True {

        print("
                Main Menu
                ---------
                (1) Say Hello
                (2) About
                (3) Exit

                  ")

        nOption = GetString()

        switch nOption {
        case 1
                print("Enter your name : ")
                name = getstring()
                print("Hello #{name}\n")
        case 2
                print("Sample : using switch statement\n")
        case 3
                Bye
        else
                print("bad option...\n")
        }

}
```

Check the next chapters:-

- Getting Started - Second Style
- Getting Started - Third Style
- Control Structures - Second Style - May looks like Lua and Ruby
- Control Structures - Third Style - May looks like C (uses braces)
- Syntax Flexibility

Note: All of these styles are provided automatically by the compiler at the same time, It's better to select one style for the same project (you can create your style as a mix from these styles) for example you can use Put/Get and Braces.

101.5 New Functions and Changes

Changed:

- get() function : changed to sysget()
- sort() function : can now work on list of objects
- find() function : can now work on list of objects

Added:

- clockspersecond()
- CurrentDir()
- ExeFileName()
- ChDir()
- ExeFolder()
- varptr()
- space()
- nullpointer()
- object2pointer()
- pointer2object()

Check the next chapters

- System Functions
- Object Oriented Programming (OOP)
- Low Level Functions

101.6 StdLib functions and classes written in Ring

Ring 1.1 comes with a library called StdLib, it's written in Ring by the help of Ring Team

The library provide a useful group of new functions and classes

Example:

```
Load "stdlib.ring"

Puts("Test Times()")
Times ( 3 , func { see "Hello, World!" + nl } )
```

Example:

```
Load "stdlib.ring"

Puts("Test Map()")
See Map( 1:10, func x { return x*x } )
```

Example:

```
Load "stdlib.ring"

Puts("Test Filter()")
See Filter( 1:10 , func x { if x <= 5 return true else return false ok } )
```

Example:

```
Load "stdlib.ring"

See "Testing the String Class" + nl
oString = new string("Hello, World!")
oString.println()
oString.upper().println()
oString.lower().println()
oString.left(5).println()
oString.right(6).println()
```

Example:

```
Load "stdlib.ring"

oList = new list ( [1,2,3] )
oList.Add(4)
oList.print()
```

Example:

```
Load "stdlib.ring"

oStack = new Stack
oStack.push(1)
oStack.push(2)
oStack.push(3)
see oStack.pop() + nl
```

Example:

```
Load "stdlib.ring"

oQueue = new Queue
oQueue.add(1)
oQueue.add(2)
oQueue.add(3)
see oQueue.remove() + nl
```

Example:

```
Load "stdlib.ring"

ohashtable = new hashtable
See "Test the hashtable Class Methods" + nl
ohashtable {
        Add("Egypt","Cairo")
        Add("KSA","Riyadh")
```

(continues on next page)

(continued from previous page)

```
        see self["Egypt"] + nl
        see self["KSA"] + nl
        see contains("Egypt") + nl
        see contains("USA") + nl
        see index("KSA")  + NL
        print()
        delete(index("KSA"))
        see copy("*",60) + nl
        print()
}
```

Example:

```
Load "stdlib.ring"

otree = new tree
See "Test the tree Class Methods" + nl
otree {
        set("The first step")   # set the root node value
        see value() + nl
        Add("one")
        Add("two")
        Add("three") {
                Add("3.1")
                Add("3.2")
                Add("3.3")
                see children
        }
        see children
        oTree.children[2] {
                Add("2.1") Add("2.2") Add("2.3") {
                        Add("2.3.1") Add("2.3.2") Add("test")
                }
        }
        oTree.children[2].children[3].children[3].set("2.3.3")
}
see copy("*",60) + nl
oTree.print()
```

Check the next chapters:

- StdLib Functions
- StdLib Classes

101.7 RingLibSDL

Ring 1.0 provided RingAllegro to be able to create games using the Allegro game programming library

Now Ring 1.1 provide RingLibSDL also so we can have the choice between Allegro or LibSDL

Example:

```
Load "libsdl.ring"

SDL_Init(SDL_INIT_EVERYTHING)
win = SDL_CreateWindow("Hello World!", 100, 100, 640, 480, SDL_WINDOW_SHOWN)
SDL_Delay(2000)
SDL_DestroyWindow(win)
SDL_Quit()
```

See the RingLibSDL Chapter.

101.8 Demo Project - Game Engine for 2D Games

In practice we would create a game engine in a language like C/C++ to get the best performance then provide Ring classes to use the engine.

But many 2D Games are simple and creating a game engine in Ring will be fast enough in many cases

Also this would be a good demo project to learn about the language concepts where we build things using Object Oriented Programming (OOP) then access the power that we have using declarative programming using nested structures or using natural programming.

In this project we selected the first way (declarative programming using nested structures)

Example:

```
Load "gameengine.ring"  # Give Control to the Game Engine

func main               # Called by the Game Engine

        oGame = New Game        # Create the Game Object
        {
                title = "My First Game"
                text {
                        x = 10  y=50
                        animate = false
                        size = 20
                        file = "fonts/pirulen.ttf"
                        text = "game development using ring is very fun!"
                        color = rgb(0,0,0)      # Color = black
                }
                text {
                        x = 10  y=150
                        # Animation Part ======================================
                        animate = true                          # Use Animation
                        direction = GE_DIRECTION_INCVERTICAL    # Increase y
                        point = 400             # Continue until y=400
```

(continues on next page)

(continued from previous page)

```
                        nStep = 3               # Each time y+= 3
                        #=====================================================
                        size = 20
                        file = "fonts/pirulen.ttf"
                        text = "welcome to the real world!"
                        color = rgb(0,0,255)    # Color = Blue
                }
                Sound {                                 # Play Sound
                        file = "sound/music1.wav"       # Sound File Name
                }
        }                                       # Start the Events Loop
```

See the "Demo Project - Game Engine for 2D Games" chapter.

101.9 RingSQLite

Ring 1.0 provided support for ODBC to use any database and provided native support for MySQL.

Now Ring 1.1 provide native support for SQLite database too.

Example:

```
oSQLite = sqlite_init()

sqlite_open(oSQLite,"mytest.db")

sql = "CREATE TABLE COMPANY("  +
         "ID INT PRIMARY KEY     NOT NULL," +
         "NAME           TEXT    NOT NULL," +
         "AGE            INT     NOT NULL," +
         "ADDRESS        CHAR(50)," +
         "SALARY         REAL );"

sqlite_execute(oSQLite,sql)

sql = "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "  +
         "VALUES (1, 'Mahmoud', 29, 'Jeddah', 20000.00 ); " +
         "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) "  +
         "VALUES (2, 'Ahmed', 27, 'Jeddah', 15000.00 ); "     +
         "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
         "VALUES (3, 'Mohammed', 31, 'Egypt', 20000.00 );" +
         "INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)" +
         "VALUES (4, 'Ibrahim', 24, 'Egypt ', 65000.00 );"

sqlite_execute(oSQLite,sql)

aResult =  sqlite_execute(oSQLite,"select * from COMPANY")
for x in aResult
        for t in x
                see t[2] + nl
        next
```

(continues on next page)

(continued from previous page)

```
next
see copy("*",50)  + nl
for x in aResult
        see x["name"] + nl
next
sqlite_close(oSQLite)
```

101.10 Better Code Generator for Extensions

We are using the code generator (written in Ring) every day to add new libraries to Ring.

The generator is used to create RingQt and RingAllegro

Also in Ring 1.1 it's used to create RingLibSDL.

more features are added like

- Set/Get structure members (numbers & pointers)
- Using constants
- Better Generated Code

See the Code Generator chapter.

101.11 Using Self.Attribute in the Class Region to define new attributes

We can use Self.Attribute in the Class Region (after the class name and before any methods) to define new attributes.

```
class Person
        name            # Define name as attribute if it's not a global variable
        address
        phone

class person2
        self.name       # Must Define the attribute
        self.address
        self.phone
```

101.12 Using This.Attribute in nested Braces inside the Class Methods

We can use nested braces { } while we are inside methods to access another objects, In this case the current object scope will be changed while we are inside the brace and Self will point to the object that we access using braces { }. In this case we can use This.Attribute and This.Method() to access the object that will be created from the current class.

Check the Object Oriented Programming chapter for more information.

Also Check the Weight History Application in GUI Development using RingQt chapter.

101.13 Better Documentation

Ring 1.1 documentation (800 pages) is better than Ring 1.0 documentation (340 pages)

Many chapters are added for providing better information about the language like:

- Language Reference
- Scope Rules
- FAQ

And more!

CHAPTER

TWO

WHAT IS NEW IN RING 1.2?

In this chapter we will learn about the changes and new features in Ring 1.2 release.

102.1 List of changes and new features

Ring 1.2 comes with many new features

- New Functions
- Better Functions
- Better Ring Notepad
- Better RingQt
- Objects Library for RingQt
- RingLibCurl
- Better Call Command
- Using NULL instead of NULLPointer()
- Display Warnings Option
- Better Quality

102.2 New Functions

- PtrCmp() Function is a new function that compare between C pointers like the GUI objects.
- PrevFileName() Function is added to return the previous active source file name.
- RingVM_CFunctionsList() Function is added to return a list of functions written in C.
- RingVM_FunctionsList() Function is added to return a list of functions written in Ring.
- RingVM_ClassesList() Function is added to return a list of Classes.
- RingVM_PackagesList() Function is added to return a list of Packages.
- RingVM_MemoryList() Function is added to return a list of Memory Scopes and Variables.
- RingVM_CallList() Function is added to return a list of the functions call list.
- RingVM_FilesList() Function is added to return a list of the Ring Files.

Example:

```
fp = fopen("ptrcmp.ring","r")
fp2 = fp
fp3 = fopen("ptrcmp.ring","r")

see ptrcmp(fp,fp2) + nl
see ptrcmp(fp,fp3) + nl

fclose(fp)
fclose(fp3)
```

Output:

```
1
0
```

Also we can compare between them using the '=' operator

Example:

```
fp = fopen("ptrcmp2.ring","r")
fp2 = fopen("ptrcmp2.ring","r")
fp3 = fp
see fp = fp2
see nl
see fp = fp3
fclose(fp)
fclose(fp2)
```

Output:

```
0
1
```

Example:

The next function in stdlib.ring uses the PrevFileName() to know if the file of the caller function is the main source file of the program or not.

```
Func IsMainSourceFile
        if PrevFileName() = sysargv[2]
                return true
        ok
        return false
```

102.3 Better Functions

The find() function is updated to support searching in lists using C pointers like GUI Objects.

The type() function is updated to display the C pointers types (like the GUI Object Class Name).

102.4 Better Ring Notepad

The Ring Notepad will save the current line number of opened files to be restored when we switch between files.

Also Ring Notepad will ask the user to save the file if the file content is changed when the user switch between files.

102.5 Better RingQt

RingQt classes are updated to include methods to get events (The code that will be executed when an event is fired). This is necessary to enable/disable events for some time or to get the events information.

For example the next code disable an event then call a method then enable the event again.

```
cEvent = oView.oListResult.getCurrentItemChangedEvent()
oView.oListResult.setCurrentItemChangedEvent("")
FindValueAction()       # Call Method while an event is disabled
oView.oListResult.setCurrentItemChangedEvent(cEvent)
```

Also the QAllEvents class is updated where we can set the output from the event function to be true or false using a new method added to the class called setEventOutput.

```
Load "guilib.ring"

MyApp = New qApp {
                win = new qWidget() {
                                setwindowtitle("Hello World")
                                setGeometry(100,100,370,250)
                                lineedit1 = new qlineedit(win) {
                                        setGeometry(10,100,350,30)
                                        setinputmask("9999;_")
                                        oFilter = new qallevents(lineedit1)
                                        oFilter.setfocusoutEvent("pMove()")
                                        installeventfilter(oFilter)
                                }
                                lineedit2 = new qlineedit(win) {
                                                setGeometry(10,150,350,30)
                                }
                                show()
                }
                exec()
}

func pMove
            win.setWindowTitle("xxxx")
            oFilter.setEventOutput(False)
```

102.6 Objects Library for RingQt

Ring 1.2 comes with the Objects library for RingQt applications. Instead of using global variables for windows objects and connecting events to objects using the object name, the Objects Library will manage the GUI objects and will provide a more natural API to quickly create one or many windows from the same class and the library provide a way to quickly set methods to be executed when an event is fired. Also the library provide a natural interface to quickly use the parent or the caller windows from the child or sub windows.

The Objects Library is designed to be used with the MVC Design Pattern.

The Objects Library is merged in RingQt so you can use it directly when you use RingQt

Example :

```
load "guilib.ring"

new qApp {
        open_window( :MainWindowController )
        exec()
}

class MainWindowController from WindowsControllerParent
        oView = new MainWindowView
        func SubWindowAction
                Open_window( :SubWindowController )
                Last_Window().SetParentObject(self)

class MainWindowView from WindowsViewParent
        win = new qWidget() {
                SetWindowTitle("Main Window")
                btnSub = new qPushButton(win) {
                        setText("Sub Window")
                        setClickEvent( Method( :SubWindowAction ) )
                }
                resize(400,400)
        }

class SubWindowController from WindowsControllerParent
        oView = new SubWindowView
        func SetMainWindowTitleAction
                Parent().oView.win.SetWindowTitle("Message from the Sub Window")
                oView.win.SetWindowTitle("Click Event Done!")

class SubWindowView from WindowsViewParent
        win = new qWidget() {
                SetWindowTitle("Sub Window")
                btnMsg = new qPushButton(win) {
                        setText("Set Main Window Title")
                        setClickEvent( Method( :SetMainWindowTitleAction ) )
                }
                btnClose = new qPushButton(win) {
                        Move(200,0)
                        setText("Close")
                        setClickEvent( Method( :CloseAction ) )
```

(continues on next page)

(continued from previous page)

```
                }
                resize(400,400)
        }
```

102.7 RingLibCurl

The LibCurl library is used starting from Ring 1.0 for the Download() and SendEmail() functions implementation. In Ring 1.2 more functions are added to provide a powerful library (RingLibCurl) around LibCurl.

Example:

```
load "libcurl.ring"

curl = curl_easy_init()

cPostThis = "page=4&Number1=4&Number2=5"
curl_easy_setopt(curl, CURLOPT_URL, "http://localhost/ringapp/index.ring?page=3")
curl_easy_setopt(curl, CURLOPT_POSTFIELDS, cPostThis)

curl_easy_perform(curl)

curl_easy_cleanup(curl)
```

102.8 Better Call Command

The Call command is updated to support calling functions from object attributes also (not only variables).

For example the next code from the Stars Fighter Game

```
cFunc = oself.keypress
call cFunc(oGame,oSelf,Key_Space)
```

Can be written in one line

```
call oself.keypress(oGame,oSelf,Key_Space)
```

102.9 Using NULL instead of NULLPointer()

We can pass NULL to functions instead of using NULLPointer()

For example the next code from RingLibSDL

```
SDL_RenderCopy(SDL_ren,tex,NULLPointer(),rect)
```

Can be written as in the next line

```
SDL_RenderCopy(SDL_ren,tex,NULL,rect)
```

102.10 Display Warnings Option

In Ring 1.2 the Ring compiler is updated to include the Display Warnings option (-w)

Example:

```
load "stdlib.ring"
load "stdlib.ring"
```

compiling the program using the Display Warnings option will display the file duplication warning, While without that option the error will pass silent.

This is a warning (not an error) because in large projects you may use the same file more than one time. For example it's common to start each file with the next code. where the function IsMainSourceFile() is part from the stdlib.ring

```
load "stdlib.ring"
if IsMainSourceFile()
        // Testing
ok
```

102.11 Better Quality

Ring 1.2 is more stable, We discovered and fixed more bugs during Ring usage everyday in practical projects. Some functions are optimized to be faster like the SubStr() function. Also the documentation is more better.

CHAPTER

THREE

WHAT IS NEW IN RING 1.3?

In this chapter we will learn about the changes and new features in Ring 1.3 release.

103.1 List of changes and new features

Ring 1.3 comes with many new features

- Better RingQt
- Better Ring Notepad
- Ring mode for Emacs Editor
- Better StdLib
- Better Loop/Exit Command
- New Functions
- Return Self by Reference
- Using '<' and ':' operators as 'from' keyword
- Embedding Ring in Ring without sharing the State
- RingZip Library
- Form Designer

103.2 Better RingQt

(1) Another version of QPixMap class is added (QPixMap2) which takes (int width,int height) during object init.

Example:

```
Load "guilib.ring"
New qapp
{
        win1 =  new qwidget()
        {
                setwindowtitle("Drawing using QPixMap")
                setgeometry(100,100,500,500)
                label1 = new qlabel(win1)
                {
```

(continues on next page)

(continued from previous page)

```
                        setgeometry(10,10,400,400)
                        settext("")
                }
                imageStock = new qlabel(win1)
                {
                        image = new qPixMap2(200,200)
                        color = new qcolor() {
                                setrgb(255,255,255,255)
                        }
                        pen = new qpen() {
                                setcolor(color)
                                setwidth(10)
                        }
                        new qpainter() {
                                begin(image)
                                        setpen(pen)
                                        drawline(0,0,200,200)
                                        drawline(200,0,0,200)
                                endpaint()
                        }
                        setpixmap(image)
                }
                show()
        }
        exec()
}
```

Screen Shot:

(2) The Objects Library is updated to include the next functions

- Last_WindowID()
- Open_WindowNoShow()
- Open_WindowAndLink()

Also the class name (WindowViewBase) is changed to (WindowsViewParent).

In The next code for example the Open_WindowAndLink() will create an object from the SecondWindowController Class Then will add the Method SecondWindow() to the FirstWindowController Class Also will add the Method FirstWindow() to the SecondWindowController Class

So the SendMessage() method in FirstWindowController class can use the SecondWindow() method to access the object.

```
class firstwindowController from windowsControllerParent

    oView = new firstwindowView

    func OpenSecondWindow
```

(continues on next page)

(continued from previous page)

```
        Open_WindowAndLink(:SecondWindowController,self)

    func SendMessage
        if IsSecondWindow()
            SecondWindow().setMessage("Message from the first window")
        ok

    func setMessage cMessage
        oView.Label1.setText(cMessage)
```

(3) The next classes are added to RingQt

- QPixMap2
- QScrollArea
- QSplitter
- QCompleter
- QCompleter2
- QCompleter3
- QProcess
- QMdiArea
- QMdiSubWindow
- QCursor
- QListView
- QDesktopServices

(4) Many constants are defined in qt.rh (loaded by guilib.ring)

(5) New Classes names - Index Start from 1

We added new classes to RingQt - another version of classes where the class names doesn't start with the "q" letter Also updated methods so the index start from 1 when we deal with the GUI controls like

- ComboBox
- ListWidget
- TableWidget
- TreeWidget

These classes are inside guilib.ring under the package name : System.GUI

To use it

```
load "guilib.ring"

import System.GUI
```

This doesn't have any effect on our previous code, It's just another choice for better code that is consistent with Ring rules.

Also the form designer is updated to provide us the choice between using classes where (index start from 0) or (index start from 1)

Example (Uses the Form Designer)

(1) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/indexstart/indexstartView.ring

(2) https://github.com/ring-lang/ring/blob/master/samples/UsingFormDesigner/indexstart/indexstartController.ring

103.3 Better Ring Notepad

(1) Using QPlainTextEdit instead of QTextEdit

(2) Displaying the line number for each line in the source code file.

Screen Shot:

(3) Auto-Complete for Ring functions names, classes and words in the opened file.

(4) Functions and Methods List

(5) Output Window

(6) Classes List

(7) Change the Current Style

103.4 Ring mode for Emacs Editor

Ring 1.3 comes with Ring mode for Emacs Editor

Screen Shot:

103.5 Better StdLib

The StdLib is updated to include the next functions

- SplitMany()
- JustFilePath()
- JustFileName()

103.6 Better Loop|Exit Command

The Loop|Exit command is updated to accept Expressions after the command (not only numbers).

The syntax:

```
Loop|Exit [Number]
```

Changed to

```
Loop|Exit [Expression]
```

Example

```
XLoop = 2       # The outer loop
YLoop = 1       # The first inner loop
for x = 1 to 10
        for y = 1 to 10
                see "x=" + x + " y=" + y + nl
                if x = 3 and y = 5
                        exit XLoop
                ok
        next
next
```

103.7 New Functions

- PackageName() function
- Swap() function

Example:

```
aList = [:one,:two,:four,:three]
see aList
see copy("*",50) + nl
swap(aList,3,4)
see aList
```

Output

```
one
two
four
three
**************************************************
one
two
three
four
```

103.8 Return Self by Reference

In this release, using Return Self in class methods will return the object by reference.

Example:

```
mylist = [new mytest() {
        see self
        x = 20
        see self
}]

see mylist
```

(continues on next page)

(continued from previous page)

```
class mytest
        x = 15
        func init
                return self     # Return by reference
```

Output

```
x: 15.000000
x: 20.000000
x: 20.000000
```

103.9 Using '<' and ':' operators as 'from' keyword

In this release of the Ring language we can use the '<' and ':' operators as the 'from' keyword

Syntax (1):

```
class Cat from Animal
```

Syntax (2):

```
class Cat < Animal
```

Syntax (3):

```
class Cat : Animal
```

103.10 Embedding Ring in Ring without sharing the State

From Ring 1.0 we already have functions for embedding Ring in the C language. Also we can execute Ring code inside Ring programs using the eval() function. In this release we provide functions for embedding Ring in Ring programs without sharing the state.

Advantages:

(1) Quick integration for Ring programs and applications together without conflicts.

(2) Execute and run Ring code in safe environments that we can trace.

Example:

```
pState = ring_state_init()
ring_state_runcode(pState,"See 'Hello, World!'+nl")
ring_state_runcode(pState,"x = 10")

pState2 = ring_state_init()
ring_state_runcode(pState2,"See 'Hello, World!'+nl")
ring_state_runcode(pState2,"x = 20")

ring_state_runcode(pState,"see x +nl")
```

(continues on next page)

(continued from previous page)

```
ring_state_runcode(pState2,"see x +nl")

v1 = ring_state_findvar(pState,"x")
v2 = ring_state_findvar(pState2,"x")

see v1[3] + nl
see V2[3] + nl

ring_state_delete(pState)
ring_state_delete(pState2)
```

Output:

```
Hello, World!
Hello, World!
10
20
10
20
```

103.11 RingZip Library

Ring 1.3 comes with the RingZip library for creating, modifying and extracting *.zip files.

Example (1): Create myfile.zip contains 4 files

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip",'w')
zip_addfile(oZip,"test.c")
zip_addfile(oZip,"zip.c")
zip_addfile(oZip,"zip.h")
zip_addfile(oZip,"miniz.h")
zip_close(oZip)
```

Example (2): Extract myfile.zip to myfolder folder.

```
load "ziplib.ring"
zip_extract_allfiles("myfile.zip","myfolder")
```

Example (3): Print file names in the myfile.zip

```
load "ziplib.ring"
oZip = zip_openfile("myfile.zip",'r')
for x=1 to zip_filescount(oZip)
       see zip_getfilenamebyindex(oZip,x) + nl
next
zip_close(oZip)
```

Example (4) : Using Classes instead of Functions

```
load "ziplib.ring"

new Zip {
        SetFileName("myfile.zip")
        Open("w")
        AddFile("test.c")
        AddFile("zip.c")
        AddFile("zip.h")
        AddFile("miniz.h")
        Close()
}
```

103.12 Form Designer

Ring 1.3 comes with the Form Designer to quickly design your GUI application windows/forms and generate the Ring source code.

It's written in Ring (Around 8000 Lines of code) using Object-Oriented Programming and Meta-Programming.

We can run the From Designer from Ring Notepad

Also we can run the Form Designer in another window.

Form Designer
File Edit View Help
ToolBox
Lock
Select
Label
Button
LineEdit
TextEdit
ListWidget
Checkbox
Image
Slider
ProgressBar
SpinBox
ComboBox
DateTimeEdit
TableWidget
TreeWidget
RadioButton
WebView
DialSlider
VideoWidget
Frame
LCDNumber
HyperLink
Timer
Form1
Ready!
Properties
Object Window
Property Value
1 X 20
2 Y 20
3 Width 400
4 Height 400
5 Title Form1
6 Back Color
7 Window Flags
8 Set Layout
9 Window Icon
10 Menubar
11 Index Type Start from 1

CHAPTER

FOUR

WHAT IS NEW IN RING 1.4?

In this chapter we will learn about the changes and new features in Ring 1.4 release.

104.1 List of changes and new features

Ring 1.4 comes with many new features

- Change: Basic Extensions are separated from RingVM
- The Natural Library
- New Style is added to Ring Notepad
- RingREPL
- Convert between Numbers and Bytes
- Better StdLib
- Better WebLib
- Better RingQt
- Qt Class Converter

104.2 Change: Basic Extensions are separated from RingVM

In Ring 1.4 the next libraries are separated from RingVM

- RingODBC
- RingMySQL
- RingSQLite
- RingOpenSSL
- RingInternet

To use these libraries, Use the Load command.

```
load "odbclib.ring"
# use ODBC Functions
```

```
load "mysqllib.ring"
# use MySQL Functions
```

```
load "sqlitelib.ring"
# use SQLite Functions
```

```
load "openssllib.ring"
# use OpenSSL Functions  ( Hash and Security functions)
```

```
load "internetlib.ring"
# use Internet Functions ( Download() and SendEmail() )
```

If you will use all of these libraries, You can just use stdlib.ring And the stdlib.ring will load odbclib.ring, mysqllib.ring, sqlitelib.ring, opensslib.ring and internetlib.ring files.

```
load "stdlib.ring"
```

104.3 The Natural Library

Ring 1.4 comes with the Natural Library to quickly define a language that contains a group of commands.

We will write the natural code in a Text file, for example program.txt

File: program.txt

```
Welcome to the Ring programming language!
What you are reading now is not comments, I swear!

After many years of programming I decided to think different about
programming and solve the problems in a better way.

We are writing commands or code and the Ring language is reading
it to understand us! Sure, What you are seeing now is
just ***part of the code - Not the Complete Program***
You have to write little things before and after this
part to be able to run it!

It is the natural part of our code where we can write in English,
Arabic or any Natural Language Then we will tell the computer
through the Ring language what must happens! in a way that we can scale
for large frameworks and programs.

Just imagine what will happens to the world of programming once
we create many powerful frameworks using the Ring language that
uses this way (Natural Programming).

For example When we say Hello to the Machine, It can reply! and when we
say count from 1 to 5 it will understand us, Also if
we said count from 5 to 1 it will
understand us too! You can see the Output window!

This Goal is not new, but the Ring language comes
with an innovative solution to this problem.
```

Output:

```
Hello, Sir!


The Numbers!

1

2

3

4

5

I will count Again!

5

4

3

2

1
```

To execute the natural code, We have start.ring

In start.ring we define the language and the commands.

File: start.ring

```
load "stdlib.ring"
load "naturallib.ring"

New NaturalLanguage {
        SetLanguageName(:MyLanguage)
        SetCommandsPath(CurrentDir()+"/../command")
        SetPackageName("MyLanguage.Natural")
        UseCommand(:Hello)
        UseCommand(:Count)
        RunFile("program.txt")
}
```

We defined a language called MyLanguage, We have folder for the language commands.

Each command will define a class that belong to the MyLanguage.Natural package.

We will define two commands, Hello and Count.

So we must have two files for defining the commands in the CurrentDir()+"/../command" folder

File: hello.ring

```
DefineNaturalCommand.SyntaxIsKeyword([
        :Package = "MyLanguage.Natural",
        :Keyword = :hello,
        :Function = func {
                See  "Hello, Sir!" + nl + nl
        }
])
```

File: count.ring

```
DefineNaturalCommand.SyntaxIsKeywordNumberNumber([
        :Package = "MyLanguage.Natural",
        :Keyword = :count,
        :Function = func {
                if not isattribute(self,:count_times) {
                        AddAttribute(self,:count_times)
                        Count_Times = 0
                }
                if Expr(1) > Expr(2) {
                        nStep = -1
                else
                        nStep = 1
                }
                if Count_Times = 0 {
                        see nl+"The Numbers!" + nl
                        Count_Times++
                else
                        see nl + "I will count Again!" +nl
                }
                for x = Expr(1) to Expr(2) step nStep {
                        see nl+x+nl
                }
                CommandReturn(fabs(Expr(1)-Expr(2))+1)
        }
])
```

104.4 New Style is added to Ring Notepad

In Ring Notepad - From View - Styles - Select the (Modern) Style

Screen Shot:

104.5 RingREPL

In the application folder, You will find RingREPL (Read-Eval-Print-Loop)

Also you can run it from Ring Notepad (Menubar - Tools)

Screen Shot:

104.6 Convert between Numbers and Bytes

Ring 1.4 comes with the next functions to convert between Numbers and Bytes.

- Int2Bytes()
- Float2Bytes()
- Double2Bytes()
- Bytes2Int()
- Bytes2Float()
- Bytes2Double()

Example:

```
see "Test Int2Bytes() and Bytes2Int() - Value : 77" + nl
r = Int2Bytes(77)
see "Int Size : " + len(r) + nl
```

(continues on next page)

(continued from previous page)

```
see r + nl
see Bytes2Int(r) + nl
see "Test Float2Bytes() and Bytes2Float() - Value 77.12" + nl
r = Float2Bytes(77.12)
see "Float Size : " + len(r) + nl
see r + nl
see Bytes2Float(r) + nl
see "Test Double2Bytes() and Bytes2Double() - Value 9999977.12345" + nl
r = Double2Bytes(9999977.12345)
see "Double Size : " + len(r) + nl
see r + nl
decimals(5)
see Bytes2Double(r) + nl
```

104.7 Better StdLib

The StdLib is updated to include the next functions

- FSize()

The print() function is updated to accept local variables.

```
load "stdlib.ring"

func main
        print("Enter your name : ")     ;
        Name = getString()              ;
        print( "Hello : #{Name} ")      ;
        return                          ;
```

104.8 Better WebLib

The web library is updated

- Provide better error message

(1) Error (WebLib-1) : REQUEST_METHOD is empty ! - Run this script from the browser

(2) Error (DataLib-1) : Can't connect to the database server!

- Better Template() function - can accept NULL instead of object as the second parameter.

```
html(template("main.rhtml",NULL))
```

- The Form Class is updated to support the "target" attribute.

```
BootStrapWebPage()
{
        Title = "The Ring Programming Language"
        html(template("main.rhtml",NULL))
        div {
```

(continues on next page)

(continued from previous page)

```
                classname = :container
                div
                {
                        id = "div3"
                        color = "black"
                        backgroundcolor = "white"
                        width = "100%"
                        form
                        {
                                method = "POST"
                                Action = website
                                Target = "codeoutput"
                                input { type="hidden" name="page" value=1 }
                                Table
                                {
                                        style = stylewidth("100%") +
                                                stylegradient(3)
                                        TR
                                        {

                                                TD { align="center"
                                                        WIDTH="10%"
                                                         text("Code :")
                                                }
                                                TD {
                                                        html(`
                                                        <textarea name = "cCode"
                                                        rows="5"
                                                        style="width : 100%; ">
                                                        See "Hello, World!" + nl
                                                        </textarea>`)
                                                }
                                        }
                                }
                                Input { type = "submit"
                                        classname="btn btn-primary btn-block"
                                                value = "Execute" }
                                Table
                                {
                                        style = stylewidth("100%") +
                                                stylegradient(34)
                                        TR
                                        {

                                                TD { align="center"
                                                        WIDTH="10%"
                                                        text("Output :")
                                                }
                                                TD {
                                                html(`
                                                <iframe name="codeoutput"
                                                width="100%"
```

(continues on next page)

(continued from previous page)

```
                                                    style="background-color:white;">
                                                    </iframe>`)
                                                    }
                                            }
                                    }

                            }
                    }

            }
            html(template("footer.rhtml",NULL))
}
```

104.9 Better RingQt

The next functions are added to RingQt

- SetDialogIcon(cIconFile)
- MsgInfo(cTitle,cMessage)
- ConfirmMsg(cTitle,cMessage)
- InputBox(cTitle,cMessage)
- InputBoxInt(cTitle,cMessage)
- InputBoxNum(cTitle,cMessage)
- InputBoxPass(cTitle,cMessage)

The next classes are added to RingQt

- QToolButton
- QSerialPort
- QSerialPortInfo

104.10 Qt Class Converter

Ring 1.4 comes with a simple tool that help in porting Qt classes to RingQt.

You will find it in ring/extensions/ringqt/converter

Online : https://github.com/ring-lang/ring/tree/master/extensions/ringqt/converter

Screen Shot:

104.11 What is new in Ring 1.4.1?

Ring 1.4.1 comes with the next changes

- Better Scripts for Building from Source Code
- Better Colors for the Modern Style in Ring Notepad
- Better StdLib
- Better RingQt
- New Sample : Sixteen Puzzle

The scripts are updated for building from source code.

Tested using Windows, Ubuntu Linux, Linux Mint and MacOS X.

Screen Shot:

In Ring Notepad - the (Modern) Style colors are updated

Screen Shot:

The StdLib is updated to include the next functions

- TrimLeft()
- TrimRight()
- TrimAll()
- EpochTime()

The next functions are updated to display the dialogs on the top of other windows.

- SetDialogIcon(cIconFile)
- MsgInfo(cTitle,cMessage)
- ConfirmMsg(cTitle,cMessage)
- InputBox(cTitle,cMessage)
- InputBoxInt(cTitle,cMessage)
- InputBoxNum(cTitle,cMessage)
- InputBoxPass(cTitle,cMessage)

The Sixteen Puzzle is added to the Applications folder.

Screen Shot:

Calmosoft Sixteen Puzzle ...
up
up
up
up
<<<
9
10
3
4
>>>
<<<
13
14
7
8
>>>
<<<
11
12
1
2
>>>
<<<
15
16
5
6
>>>
down
down
down
down
Scarmble
Reset
Save Game
Replay Game
Moves : 32
Elapsed Time : 32.53 s

CHAPTER

FIVE

WHAT IS NEW IN RING 1.5?

In this chapter we will learn about the changes and new features in Ring 1.5 release.

105.1 List of changes and new features

Ring 1.5 comes with many new features!

- Video-Music-Player Application
- Windows StartUp Manager Application
- Calculator Application
- Better Ring Notepad
- Better StdLib
- Better WebLib
- Better RingQt
- Better Objects Library
- RingFreeGLUT Extension
- RingOpenGL Extension
- Better Code Generator for Extensions
- Better Documentation Generator for Extensions
- Ring VM - Tracing Functions
- Trace Library and Interactive Debugger
- More Syntax Flexibility
- Type Hints Library
- Better Quality

105.2 Video-Music-Player Application

The Video-Music-Player application is added to the Applications folder.

Screen Shot:

105.3 Windows StartUp Manager Application

The Windows StartUp Manager

URL : https://github.com/ring-lang/WinStartupManager

Screen Shot:

105.4 Calculator Application

The Calculator application is added to the Applications folder.

Screen Shot:

105.5 Better Ring Notepad

(1) Ring Notepad is updated to include some new styles and the Main File ToolBar

The idea of the Main File ToolBar is to determine the main file in the project When the project contains many source code files

This way you can run the project (Main File) at any time while opening other files in the project without the need to switch to the Main File to run the project.

To quickly use this feature

(Open the project main file)

Press Ctrl+Shift+M to set the current source code file as the main file

Open and modify other source code files in the project

To run the project (Main File) at any time press Ctrl+Shift+F5 (GUI) or Ctrl+Shift+D (Console)

Screen Shots:

Ring Notepad
File
Edit
View
Program
Browser
Tools
Help
Project Files
Ctrl+J
Source Code
Ctrl+U
Web Browser
Ctrl+W
Functions List
Alt+F
Classes List
Alt+C
Output Window
Alt+O
Form Designer Window
Ctrl+Shift+F
Style
Windows
Fusion : White
Fusion : Blue
Fusion : Black
Modern
Modern : Black
Modern : Black 2
Notepad : White
Notepad : Purple
Notepad : DarkBlue
Notepad : Black
controls
qallevents.ring
qcheckbox.ring
qcombobox.ri...
qdatetimeedit...
qdial.ring
qframe3.ring
qhyperlink.ring
qimage.ring
qlabel.ring
qlayout.ring
qlcdnumber.ri...
qlineedit.ring
qlistwidget.ring
qprogressbar.r...
qpushbutton.r...

Ring Notepad
File
Edit
View
Program
Browser
Tools
Help
Project Files
ring
Debug (Run then wait!)
Ctrl+D
Run
Ctrl+R
Run GUI Application (No Console)
Ctrl+F5
Set the Main File to be the current source file
Ctrl+Shift+M
Main File : Debug (Run then wait!)
Ctrl+Shift+D
Main File : Run
Ctrl+Shift+R
Main File : Run GUI Application (No Console)
Ctrl+Shift+F5

(2) The output window is updated to display the new lines correctly and contains the "Clear" button.

Screen Shot:

(3) The Ring Notepad is updated to quickly open and switch between large files while preparing the functions/classes lists in the background.

Screen Shot:

105.6 Better StdLib

New Functions

- Print2Str()
- ListAllFiles()
- SystemCmd()

(1) The Print2Str() is a new function added to the StdLib

Example:

```
load "stdlib.ring"

world = "World!"
mystring = print2str("Hello, #{world} \nIn Year \n#{2000+17} \n")

see mystring + nl
```

Output:

```
Hello, World!
In Year
2017
```

(2) The ListAllFiles() is a new function added to the StdLib

Using this function we can quickly do a process on a group of files in a folder and it's sub folders.

Example:

```
load "stdlib.ring"
aList = ListAllFiles("c:/ring/ringlibs","ring") # *.ring only
aList = sort(aList)
see aList
```

Example:

```
load "stdlib.ring"
see listallfiles("b:/ring/ringlibs/weblib","") # All Files
```

(3) The SystemCmd() is a new function added to the StdLib

The function will execute a system command like the System() function but will return the output in a string.

Example:

```
cYou = SystemCmd("whoami")
See "SystemCmd: whoami =====" + nl + cYou +nl
```

Output:

```
SystemCmd: whoami =====
desktop-umberto\umberto
```

105.7 Better WebLib

The WebLib is updated to include the HTMLPage class

Using this class we can create HTML documents without printing the output to the standard output

So instead of using the WebLib in Web Applications only

We can use it in Console/GUI/Mobile Applications too

Example:

```
load "stdlib.ring"
load "weblib.ring"

import System.Web

func main

        mypage = new HtmlPage {
                h1 { text("Customers Report") }
                Table
                {
                        style = stylewidth("100%") + stylegradient(4)
                        TR
                        {
                              TD { WIDTH="10%" text("Customers Count : " )  }
                              TD { text (100) }
                        }
                }
```

(continues on next page)

(continued from previous page)

```
                Table
                {
                        style = stylewidth("100%") + stylegradient(26)
                        TR
                        {
                                style = stylewidth("100%") + stylegradient(24)
                                TD { text("Name " )  }
                                TD { text("Age" ) }
                                TD { text("Country" ) }
                                TD { text("Job" ) }
                                TD { text("Company" ) }
                        }
                        for x =  1 to 100
                                TR
                                {
                                        TD { text("Test" )  }
                                        TD { text("30" ) }
                                        TD { text("Egypt" ) }
                                        TD { text("Sales" ) }
                                        TD { text("Future" ) }
                                }
                        next
                }

        }

        write("report.html",mypage.output())
```

Using this feature we can create reports quickly using WebLib & GUILib together

Example:

```
load "stdlib.ring"
load "weblib.ring"
load "guilib.ring"

import System.Web
import System.GUI

new qApp {
        open_window(:CustomersReportController)
        exec()
}

class CustomersReportController

        oView = new CustomersReportView

        func Start
                CreateReport()

        func CreateReport
```

(continues on next page)

(continued from previous page)

```
                mypage = new HtmlPage {
                        h1 { text("Customers Report") }
                        Table
                        {
                                style = stylewidth("100%") + stylegradient(4)
                                TR
                                {
                                        TD { WIDTH="10%"
                                                text("Customers Count : " )  }
                                        TD { text (100) }
                                }
                        }
                        Table
                        {
                                style = stylewidth("100%") + stylegradient(26)
                                TR
                                {
                                        style = stylewidth("100%") +
                                                stylegradient(24)
                                        TD { text("Name " )  }
                                        TD { text("Age" ) }
                                        TD { text("Country" ) }
                                        TD { text("Job" ) }
                                        TD { text("Company" ) }
                                }
                                for x =  1 to 100
                                        TR
                                        {
                                                TD { text("Test" )  }
                                                TD { text("30" ) }
                                                TD { text("Egypt" ) }
                                                TD { text("Sales" ) }
                                                TD { text("Future" ) }
                                        }
                                next
                        }
                }
                write("report.html",mypage.output())

        func PrintEvent
                printer1 = new qPrinter(0) {
                        setoutputformat(1)
                        setoutputfilename("report.pdf")
                }
                oView {
                        web.print(printer1)
                        web.show()
                }
                system ("report.pdf")

class CustomersReportView
```

(continues on next page)

(continued from previous page)

```
win = new window() {
                setwindowtitle("Report Window")
                setgeometry(100,100,500,500)
                web = new webview(win) {
                        setgeometry(100,100,1000,500)
                        loadpage(new qurl("file:///"+
                        currentdir()+"/report.html"))
                }
                new pushbutton(win) {
                                setGeometry(100,20,100,30)
                                settext("Print")
                                setclickevent(Method(:PrintEvent))
                }
                showMaximized()
        }
```

Screen Shot:

105.8 Better RingQt

New classes added to RingQt :

- QStringRef
- QMutex
- QMutexLocker
- QBuffer

- QBluetoothAddress
- QBluetoothDeviceDiscoveryAgent
- QBluetoothDeviceInfo
- QBluetoothHostInfo
- QBluetoothLocalDevice
- QBluetoothServer
- QBluetoothServiceDiscoveryAgent
- QBluetoothServiceInfo
- QBluetoothSocket
- QBluetoothTransferManager
- QBluetoothTransferReply
- QBluetoothTransferRequest
- QBluetoothUuid

Example:

```
### Submits your car VIN - Vehicle Id Number - to the Web Site - vpic.nhtsa.dot.gov -
### Parses XML data returned
### Prints out the car info result

load "libcurl.ring"
load "guilib.ring"
load "stdlib.ring"

curl = curl_easy_init()

# request = "3G1JC5248YS251015?format=xml"  ### VIN - Chevrolet
  request = "3GYFK62847G247323?format=xml"  ### VIN - Cadillac

call_type   = "decodevinvalues/"
url         = "https://vpic.nhtsa.dot.gov/api/vehicles/"
url_request = url + call_type + request

        See "URL Request: "+ url_request +nl

curl_easy_setopt(curl, curlopt_url, url_request)
response = curl_easy_perform_silent(curl);

        See nl +"Response Raw: "+ response +nl +nl

curl_easy_cleanup(curl)

xml = new qxmlstreamreader()
xml.adddata_2(response)

x = new qstringref()
while not xml.atend()
           if xml.error()
```

(continues on next page)

(continued from previous page)

```
                            see xml.errorstring() see nl
                            exit loop
            ok

            x = xml.text()
            if not x.length() = 0
                            see "Length: " see x.length() +" --- "
                            see "Value: " see x.tostring() see nl
            ok

            xml.readnext()
end

get x

###-----------------------------------------
### Results
#
# ==>Value: 115
# ==>Value: Results returned successfully
# ==>Value: VIN(s): 3G1JC5248YS251015
# ==>Value: 3G1JC5248YS251015
# ==>Value: Sedan/Saloon
# ==>Value: 4
# ==>Value: 2200.0
# ==>Value: 134.25223700841
# ==>Value: 2.2
# ==>Value: 4
# ==>Value: LN2
# ==>Value: CHEVROLET
# ==>Value: GENERAL MOTORS LLC
# ==>Value: Cavalier
# ==>Value: 2000
# ==>Value: Ramos Arzipe
# ==>Value: PASSENGER CAR
# ==>Value: 4
# ==>Value: In-Line
# ==>Value: 1st Row (Driver & Passenger)
# ==>Value: Sequential Fuel Injection (SFI)
# ==>Value: Mexico
# ==>Value: NA
# ==>Value: Manual
# ==>Value: Body Type: Sedan, 4-6 Window, Notchback (GM codes: 19, 69)
# ==>Value:  Name Plate: Chevrolet, Pontiac
# ==>Value: 0 - VIN decoded clean. Check Digit (9th position) is correct
# ==>Value: LAN
# ==>Value: 984
#
###-----------------------------------------
```

105.9 Better Objects Library

The function Open_WindowInPackages() is added to the Objects library.

The Open_WindowInPackages() function is the same as Open_Window() but takes an extra list that determine the packages to import before opening the window.

Syntax:

```
Open_WindowInPackages(cClassName,aPackagesList)
```

Example:

The next example from the Form Designer source code, Open the Window Flags window using the open_windowInPackages() function.

We determine the class name "WindowFlagsController" and the packages name.

The Window Flags window uses the FormDesigner and System.GUI packages.

```
open_windowInPackages(:WindowFlagsController,[
        "formdesigner",
        "System.GUI"
])
```

105.10 RingFreeGLUT Extension

Ring 1.5 comes with RingFreeGLUT extension to support the FreeGLUT library

Example:

```
/*
        This sample is based on C Tutorials
        from : http://www.lighthouse3d.com/tutorials/glut-tutorial/
*/

load "freeglut.ring"
load "opengl21lib.ring"


// angle of rotation for the camera direction
angle = 0.0

// actual vector representing the camera's direction
lx=0.0 lz=-1.0

// XZ position of the camera
x=0.0  z=5.0

// the key states. These variables will be zero
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1
```

(continues on next page)

(continued from previous page)

```
// Constant definitions for Menus
C_RED  = 1
C_GREEN = 2
C_BLUE = 3
C_ORANGE = 4

C_FILL = 5
C_LINE = 6

// Pop up menu identifiers
fillMenu=NULL
fontMenu=NULL
mainMenu=NULL
colorMenu=NULL

// color for the nose
red = 1.0
blue=0.5
green=0.5

// scale of snowman
scale = 1.0

// menu status
menuFlag = 0

// default font
font = GLUT_BITMAP_TIMES_ROMAN_24

C_INT_GLUT_BITMAP_8_BY_13 = 7
C_INT_GLUT_BITMAP_9_BY_15 = 8
C_INT_GLUT_BITMAP_TIMES_ROMAN_10  = 9
C_INT_GLUT_BITMAP_TIMES_ROMAN_24  = 10
C_INT_GLUT_BITMAP_HELVETICA_10  = 11
C_INT_GLUT_BITMAP_HELVETICA_12  = 12
C_INT_GLUT_BITMAP_HELVETICA_18  = 13

// width and height of the window
h = 0
w = 0

// variables to compute frames per second
frame=0
time=0
timebase=0
s = ""

func changeSize
        w = glutEventWidth()
        h = glutEventHeight()
```

(continues on next page)

(continued from previous page)

```
        // Prevent a divide by zero, when window is too short
        // (you cant make a window of zero width).
        if h = 0
                h = 1
        ok

        ratio =  w * 1.0 / h

        // Use the Projection Matrix
        glMatrixMode(GL_PROJECTION)

        // Reset Matrix
        glLoadIdentity()

        // Set the viewport to be the entire window
        glViewport(0, 0, w, h)

        // Set the correct perspective.
        gluPerspective(45.0, ratio, 0.1, 100.0)

        // Get Back to the Modelview
        glMatrixMode(GL_MODELVIEW)

func drawSnowMan

        glScalef(scale, scale, scale)
        glColor3f(1.0, 1.0, 1.0)

// Draw Body
        glTranslatef(0.0 ,0.75, 0.0)
        glutSolidSphere(0.75,20,20)

// Draw Head
        glTranslatef(0.0, 1.0, 0.0)
        glutSolidSphere(0.25,20,20)

// Draw Eyes
        glPushMatrix()
        glColor3f(0.0,0.0,0.0)
        glTranslatef(0.05, 0.10, 0.18)
        glutSolidSphere(0.05,10,10)
        glTranslatef(-0.1, 0.0, 0.0)
        glutSolidSphere(0.05,10,10)
        glPopMatrix()

// Draw Nose
        glColor3f(red, green, blue)
        glRotatef(0.0,1.0, 0.0, 0.0)
        glutSolidCone(0.08,0.5,10,2)

        glColor3f(1.0, 1.0, 1.0)
```

(continues on next page)

(continued from previous page)

```
func renderBitmapString x,y,z,font,string
        glRasterPos3f(x, y,z)
        for c in string
                glutBitmapCharacter(font,ascii(c))
        next

func renderStrokeFontString x,y,z,font,string
        glPushMatrix()
        glTranslatef(x, y,z)
        glScalef(0.002, 0.002, 0.002)
        for c in string
                glutStrokeCharacter(font, Ascii(c));
        next
        glPopMatrix()


func restorePerspectiveProjection

        glMatrixMode(GL_PROJECTION)
        // restore previous projection matrix
        glPopMatrix()

        // get back to modelview mode
        glMatrixMode(GL_MODELVIEW)


func setOrthographicProjection

        // switch to projection mode
        glMatrixMode(GL_PROJECTION)

        // save previous matrix which contains the
        //settings for the perspective projection
        glPushMatrix()

        // reset matrix
        glLoadIdentity()

        // set a 2D orthographic projection
        gluOrtho2D(0, w, h, 0)

        // switch back to modelview mode
        glMatrixMode(GL_MODELVIEW)



func computePos deltaMove

        x += deltaMove * lx * 0.1
        z += deltaMove * lz * 0.1
```

(continues on next page)

(continued from previous page)

```
func renderScene

        if  deltaMove
                computePos(deltaMove)
        ok

        // Clear Color and Depth Buffers
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)

        // Reset transformations
        glLoadIdentity()

        // Set the camera
        gluLookAt(      x, 1.0, z,
                        x+lx, 1.0,  z+lz,
                        0.0, 1.0,  0.0)

        // Draw ground

        glColor3f(0.9, 0.9, 0.9)
        glBegin(GL_QUADS)
                glVertex3f(-100.0, 0.0, -100.0)
                glVertex3f(-100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0,  100.0)
                glVertex3f( 100.0, 0.0, -100.0)
        glEnd()

// Draw 9 SnowMen
        for i = -3 to -1
                for j = -3 to -1
                        glPushMatrix()
                        glTranslatef(i*10.0, 0.0, j * 10.0)
                        drawSnowMan()
                        number = (i+3)*3+(j+3)
                        renderBitmapString(0.0, 0.5, 0.0,font ,""+number)
                        glPopMatrix()
                next
        next

        // Code to compute frames per second
        frame++

        time=glutGet(GLUT_ELAPSED_TIME)
        if time - timebase > 1000
                s = "RingFreeGLUT - FPS: " + (frame*1000.0/(time-timebase))
                timebase = time
                frame = 0
        ok

        // Code to display a string (fps) with bitmap fonts
        setOrthographicProjection()
```

(continues on next page)

(continued from previous page)

```
        glPushMatrix()
        glLoadIdentity()
        renderBitmapString(5,30,0,GLUT_BITMAP_HELVETICA_18,s)
        glPopMatrix()

        restorePerspectiveProjection()

        glutSwapBuffers()


// -----------------------------------
//             KEYBOARD
// -----------------------------------

func processNormalKeys
        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on 27
                        glutDestroyMenu(mainMenu)
                        glutDestroyMenu(fillMenu)
                        glutDestroyMenu(colorMenu)
                        glutDestroyMenu(fontMenu)
                        Shutdown()
        off


func pressKey

        key = glutEventKey()
        xx = glutEventX()
        yy = glutEventY()

        switch key
                on GLUT_KEY_UP
                         deltaMove = 0.5
                on GLUT_KEY_DOWN
                        deltaMove = -0.5
        off


func releaseKey

        key = glutEventKey()

        switch key
                on GLUT_KEY_UP
                        deltaMove = 0
                on GLUT_KEY_DOWN
                        deltaMove = 0
```

(continues on next page)

(continued from previous page)

```
        off


// -----------------------------------
//             MOUSE
// -----------------------------------

func mouseMove
        xx = glutEventX()
        yy = glutEventY()

        // this will only be true when the left button is down
        if xOrigin >= 0

                // update deltaAngle
                deltaAngle = (xx - xOrigin) * 0.001

                // update camera's direction
                lx = sin(angle + deltaAngle)
                lz = -cos(angle + deltaAngle)
        ok


func mouseButton

        button = glutEventButton()
        state = glutEventState()
        xx = glutEventX()
        yy = glutEventY()

        // only start motion if the left button is pressed
        if button = GLUT_LEFT_BUTTON
                // when the button is released
                if state = GLUT_UP
                        angle += deltaAngle
                        xOrigin = -1
                else
                        // state = GLUT_DOWN
                        xOrigin = xx
                ok
        ok


// -----------------------------------
//             MENUS
// -----------------------------------

func processMenuStatus

        status = glutEventStatus()

        if status = GLUT_MENU_IN_USE
```

(continues on next page)

(continued from previous page)

```
                menuFlag = 1
        else
                menuFlag = 0
        ok


func processMainMenu

        // nothing to do in here
        // all actions are for submenus


func processFillMenu

        option = glutEventValue()

        switch option

                on C_FILL
                        glPolygonMode(GL_FRONT, GL_FILL)
                on C_LINE
                         glPolygonMode(GL_FRONT, GL_LINE)
        off


func processFontMenu

        option = glutEventValue()

        switch (option) {
                on C_INT_GLUT_BITMAP_8_BY_13
                        font = GLUT_BITMAP_8_BY_13
                on C_INT_GLUT_BITMAP_9_BY_15
                        font = GLUT_BITMAP_9_BY_15
                on C_INT_GLUT_BITMAP_TIMES_ROMAN_10
                        font = GLUT_BITMAP_TIMES_ROMAN_10
                on C_INT_GLUT_BITMAP_TIMES_ROMAN_24
                        font = GLUT_BITMAP_TIMES_ROMAN_24
                on C_INT_GLUT_BITMAP_HELVETICA_10
                        font = GLUT_BITMAP_HELVETICA_10
                on C_INT_GLUT_BITMAP_HELVETICA_12
                        font = GLUT_BITMAP_HELVETICA_12
                on C_INT_GLUT_BITMAP_HELVETICA_18
                        font = GLUT_BITMAP_HELVETICA_18
        off

func processColorMenu

        option = glutEventValue()

        switch option
                on C_RED
```

(continues on next page)

(continued from previous page)

```
                        red = 1.0
                        green = 0.0
                        blue = 0.0
                on C_GREEN
                        red = 0.0
                        green = 1.0
                        blue = 0.0
                on C_BLUE
                        red = 0.0
                        green = 0.0
                        blue = 1.0
                on C_ORANGE
                        red = 1.0
                        green = 0.5
                        blue = 0.5
        off


func createPopupMenus

        fontMenu = glutCreateMenu(:processFontMenu)

        glutAddMenuEntry("BITMAP_8_BY_13 ",C_INT_GLUT_BITMAP_8_BY_13 )
        glutAddMenuEntry("BITMAP_9_BY_15",C_INT_GLUT_BITMAP_9_BY_15 )
        glutAddMenuEntry("BITMAP_TIMES_ROMAN_10 ",C_INT_GLUT_BITMAP_TIMES_ROMAN_10  )
        glutAddMenuEntry("BITMAP_TIMES_ROMAN_24",C_INT_GLUT_BITMAP_TIMES_ROMAN_24  )
        glutAddMenuEntry("BITMAP_HELVETICA_10 ",C_INT_GLUT_BITMAP_HELVETICA_10  )
        glutAddMenuEntry("BITMAP_HELVETICA_12",C_INT_GLUT_BITMAP_HELVETICA_12  )
        glutAddMenuEntry("BITMAP_HELVETICA_18",C_INT_GLUT_BITMAP_HELVETICA_18  )

        fillMenu = glutCreateMenu(:processFillMenu)

        glutAddMenuEntry("Fill",C_FILL)
        glutAddMenuEntry("Line",C_LINE)

        colorMenu = glutCreateMenu(:processColorMenu)
        glutAddMenuEntry("Red",C_RED);
        glutAddMenuEntry("Blue",C_BLUE);
        glutAddMenuEntry("Green",C_GREEN);
        glutAddMenuEntry("Orange",C_ORANGE);

        mainMenu = glutCreateMenu(:processMainMenu)

        glutAddSubMenu("Polygon Mode", fillMenu)
        glutAddSubMenu("Color", colorMenu)
        glutAddSubMenu("Font",fontMenu)
        // attach the menu to the right button
        glutAttachMenu(GLUT_RIGHT_BUTTON)

        // this will allow us to know if the menu is active
        glutMenuStatusFunc(:processMenuStatus)
```

(continues on next page)

(continued from previous page)

```
// -----------------------------------
//             MAIN
// -----------------------------------

func main

        // init GLUT and create window
        glutInit()
        glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA)
        glutInitWindowPosition(100,100)
        glutInitWindowSize(320,320)
        glutCreateWindow("RingFreeGLUT - Test - 9 SnowMan")

        // register callbacks
        glutDisplayFunc(:renderScene)
        glutReshapeFunc(:changeSize)
        glutIdleFunc(:renderScene)

        glutIgnoreKeyRepeat(1)
        glutKeyboardFunc(:processNormalKeys)
        glutSpecialFunc(:pressKey)
        glutSpecialUpFunc(:releaseKey)

        // here are the two new functions
        glutMouseFunc(:mouseButton)
        glutMotionFunc(:mouseMove)

        // OpenGL init
        glEnable(GL_DEPTH_TEST)
        glEnable(GL_CULL_FACE)

        // init Menus
        createPopupMenus()

        // enter GLUT event processing cycle
        glutMainLoop()
```

Screen Shots:

Ring Notepad
File Edit View Program Browser Tools Help
Main File : B:/pwct2/pwct.ring
Project Files
Source Code : B:/ring/extensions/ringfreeglut/test16.ring
Functions (20)
/*
This sample is based on C Tutorials
from : http://www.lighthouse3d.com/tutorials/glut-tutorial/
*/
load "freeglut.ring"
// angle of rotation for the camera
angle = 0.0
// actual vector representing the c
lx=0.0 lz=-1.0
// XZ position of the camera
x=0.0 z=5.0
// the key states. These variables
//when no key is being presses
deltaAngle = 0.0
deltaMove = 0
xOrigin = -1
// Constant definitions for Menus
C_RED = 1
RingFreeGLUT - Test - 9 SnowMan
RingFreeGLUT - FPS: 100.30
changesize()
computepos()
createpopupmenus()
drawsnowman()
main()
mousebutton()
Ready!

Ring Notepad
File Edit View Program Browser Tools Help
Main File :
Project Files
Source Code : B:/ring/extensions/ringfreeglut/tests/test16.ring
Functions (20)
/*
This sample is based
from : http://www.lig
*/
load "freeglut.ring"
load "opengl21lib.ring"
// angle of rotation for t
angle = 0.0
// actual vector represen
lx=0.0 lz=-1.0
// XZ position of the car
RingFreeGLUT - Test - 9 SnowMan
RingFreeGLUT - FPS: 111.89

105.11 RingOpenGL Extension

Ring 1.5 comes with RingOpenGL and support for the next versions

- OpenGL 1.1
- OpenGL 1.2
- OpenGL 1.3
- OpenGL 1.4
- OpenGL 1.5
- OpenGL 2.0
- OpenGL 2.1
- OpenGL 3.0
- OpenGL 3.2
- OpenGL 3.3
- OpenGL 4.0
- OpenGL 4.1
- OpenGL 4.2
- OpenGL 4.3
- OpenGL 4.4
- OpenGL 4.5
- OpenGL 4.6

Example:

```
/*
        This sample is based on C Tutorials
        from :
                http://www.wikihow.com/Make-a-Cube-in-OpenGL
*/

load "freeglut.ring"
load "opengl21lib.ring"

// -------------------------------------------------------
// Global Variables
// -------------------------------------------------------
rotate_y=0
rotate_x=0

// -------------------------------------------------------
// display() Callback function
// -------------------------------------------------------
func display

  //  Clear screen and Z-buffer
  glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
```

(continues on next page)

(continued from previous page)

```
// Reset transformations
glLoadIdentity()

// Rotate when user changes rotate_x and rotate_y
glRotatef( rotate_x, 1.0, 0.0, 0.0 )
glRotatef( rotate_y, 0.0, 1.0, 0.0 )

//Multi-colored side - FRONT
glBegin(GL_POLYGON)

glColor3f( 1.0, 0.0, 0.0 )     glVertex3f(  0.5, -0.5, -0.5 )      # P1 is red
glColor3f( 0.0, 1.0, 0.0 )     glVertex3f(  0.5,  0.5, -0.5 )      # P2 is green
glColor3f( 0.0, 0.0, 1.0 )     glVertex3f( -0.5,  0.5, -0.5 )      # P3 is blue
glColor3f( 1.0, 0.0, 1.0 )     glVertex3f( -0.5, -0.5, -0.5 )      # P4 is purple

glEnd()

// White side - BACK
glBegin(GL_POLYGON)
glColor3f(   1.0,  1.0, 1.0 )
glVertex3f(  0.5, -0.5, 0.5 )
glVertex3f(  0.5,  0.5, 0.5 )
glVertex3f( -0.5,  0.5, 0.5 )
glVertex3f( -0.5, -0.5, 0.5 )
glEnd()

// Purple side - RIGHT
glBegin(GL_POLYGON)
glColor3f(  1.0,  0.0,  1.0 )
glVertex3f( 0.5, -0.5, -0.5 )
glVertex3f( 0.5,  0.5, -0.5 )
glVertex3f( 0.5,  0.5,  0.5 )
glVertex3f( 0.5, -0.5,  0.5 )
glEnd()

// Green side - LEFT
glBegin(GL_POLYGON)
glColor3f(   0.0,  1.0,  0.0 )
glVertex3f( -0.5, -0.5,  0.5 )
glVertex3f( -0.5,  0.5,  0.5 )
glVertex3f( -0.5,  0.5, -0.5 )
glVertex3f( -0.5, -0.5, -0.5 )
glEnd()

// Blue side - TOP
glBegin(GL_POLYGON)
glColor3f(   0.0,  0.0,  1.0 )
glVertex3f(  0.5,  0.5,  0.5 )
glVertex3f(  0.5,  0.5, -0.5 )
glVertex3f( -0.5,  0.5, -0.5 )
glVertex3f( -0.5,  0.5,  0.5 )
```

(continues on next page)

(continued from previous page)

```
  glEnd()

  // Red side - BOTTOM
  glBegin(GL_POLYGON)
  glColor3f(   1.0,  0.0,  0.0 )
  glVertex3f(  0.5, -0.5, -0.5 )
  glVertex3f(  0.5, -0.5,  0.5 )
  glVertex3f( -0.5, -0.5,  0.5 )
  glVertex3f( -0.5, -0.5, -0.5 )
  glEnd()

  glFlush()
  glutSwapBuffers()


// -----------------------------------------------------------
// specialKeys() Callback Function
// -----------------------------------------------------------
func specialKeys

        key = glutEventKey()

  //  Right arrow - increase rotation by 5 degree
        switch Key

        on GLUT_KEY_RIGHT
                rotate_y += 5

        //  Left arrow - decrease rotation by 5 degree
        on GLUT_KEY_LEFT
                rotate_y -= 5

        on GLUT_KEY_UP
                rotate_x += 5

        on GLUT_KEY_DOWN
                rotate_x -= 5

        off

  //  Request display update
  glutPostRedisplay()



// -----------------------------------------------------------
// main() function
// -----------------------------------------------------------
func main

  //  Initialize GLUT and process user parameters
  glutInit()
```

(continues on next page)

(continued from previous page)

```
//  Request double buffered true color window with Z-buffer
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)

// Create window
glutCreateWindow("Awesome Cube")

//  Enable Z-buffer depth test
glEnable(GL_DEPTH_TEST)

// Callback functions
glutDisplayFunc(:display)
glutSpecialFunc(:specialKeys)

//  Pass control to GLUT for events
glutMainLoop()

//  Return to OS
```

Screen Shot:

105.12 Better Code Generator for Extensions

The Code Generator is updated to support <constant> type, So we can have constants other than numbers, for example : Strings and Pointers.

When we have pointers we can determine the pointer type. To use this feature, before <constant> and </constant> we can use

```
$nDefaultConstantType = C_CONSTANT_TYPE_POINTER
$cDefaultConstantPointerType = "void *"
```

The next example from the RingFreeGLUT extension

```
<runcode>
        $nDefaultConstantType = C_CONSTANT_TYPE_POINTER
        $cDefaultConstantPointerType = "void"
</runcode>
<constant>
        GLUT_STROKE_ROMAN
        GLUT_STROKE_MONO_ROMAN
        GLUT_BITMAP_9_BY_15
        GLUT_BITMAP_8_BY_13
        GLUT_BITMAP_TIMES_ROMAN_10
        GLUT_BITMAP_TIMES_ROMAN_24
        GLUT_BITMAP_HELVETICA_10
        GLUT_BITMAP_HELVETICA_12
        GLUT_BITMAP_HELVETICA_18
</constant>
```

105.13 Better Documentation Generator for Extensions

The documentation generator for extensions is updated to generate a list of constants in the generated documentation

The previous versions provides the functions prototype only, Now we have the list of constants too.

105.14 Ring VM - Tracing Functions

In Ring 1.5 the next functions are added to Ring VM

- RingVM_SetTrace(cCode)
- RingVM_TraceData() –> aDataList
- RingVM_TraceEvent() –> nTraceEvent
- RingVM_TraceFunc() –> cCode
- RingVM_ScopesCount() –> nScopes
- RingVM_EvalInScope(nScope,cCode)
- RingVM_PassError()
- RingVM_HideErrorMsg(lStatus)

- RingVM_CallFunc(cFuncName)

Example:

```
load "tracelib.ring"

ringvm_settrace("mytrace()")

see "Hello, world!" + nl
see "Welcome" + nl
see "How are you?" +nl
mytest()
new myclass { mymethod() }

func mytest
        see "Message from mytest" + nl

func mytrace
        see "====== The Trace function is Active ======" + nl +
                "Trace Function Name : " + ringvm_TraceFunc() + nl +
                "Trace Event : "
        switch ringvm_TraceEvent()
                on TRACEEVENT_NEWLINE           see "New Line"
                on TRACEEVENT_NEWFUNC           see "New Function"
                on TRACEEVENT_RETURN            see "Return"
                on TRACEEVENT_ERROR             see "Error"
                on TRACEEVENT_BEFORECFUNC       see "Before C Function"
                on TRACEEVENT_AFTERCFUNC        see "After C Function"
        off
        see nl +
                "Line Number : " + ringvm_tracedata()[TRACEDATA_LINENUMBER] + nl +
                "File Name   : " + ringvm_tracedata()[TRACEDATA_FILENAME] + nl +
                "Function Name : " + ringvm_tracedata()[TRACEDATA_FUNCNAME] + nl +
                "Method or Function : "
                if ringvm_tracedata()[TRACEDATA_METHODORFUNC] =
                                 TRACEDATA_METHODORFUNC_METHOD
                        see "Method"
                else
                        if ringvm_tracedata()[TRACEDATA_FUNCNAME] = NULL
                                see "Command"
                        else
                                see "Function"
                        ok
                ok
                see nl + Copy("=",42) + nl

class myclass
        func mymethod
                see "Message from mymethod" + nl
```

Output:

```
====== The Trace function is Active ======
Trace Function Name : mytrace()
```

(continues on next page)

(continued from previous page)

```
Trace Event : After C Function
Line Number : 3
File Name   : test1.ring
Function Name : ringvm_settrace
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 5
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
Hello, world!
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 6
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
Welcome
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 7
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
How are you?
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 8
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 8
File Name   : test1.ring
Function Name : mytest
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
```

(continues on next page)

(continued from previous page)

```
Line Number : 12
File Name   : test1.ring
Function Name : mytest
Method or Function : Function
==========================================
Message from mytest
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 14
File Name   : test1.ring
Function Name : mytest
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 8
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 9
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 43
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
```

(continues on next page)

(continued from previous page)

```
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Function
Line Number : 9
File Name   : test1.ring
Function Name : mymethod
Method or Function : Method
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 44
File Name   : test1.ring
Function Name : mymethod
Method or Function : Method
==========================================
Message from mymethod
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Return
Line Number : 9
File Name   : test1.ring
Function Name :
Method or Function : Command
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : Before C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
==========================================
====== The Trace function is Active ======
```

(continues on next page)

(continued from previous page)

```
Trace Function Name : mytrace()
Trace Event : After C Function
Line Number : 9
File Name   : test1.ring
Function Name : ismethod
Method or Function : Function
=========================================
====== The Trace function is Active ======
Trace Function Name : mytrace()
Trace Event : New Line
Line Number : 11
File Name   : test1.ring
Function Name :
Method or Function : Command
=========================================
```

105.15 Trace Library and Interactive Debugger

Ring 1.5 comes with the Trace Library and the Interactive Debugger

Using this library we can trace events, execute programs line by line, open the Interactive Debugger when an error happens or at breakpoints.

Example:

The next example uses a Breakpoint to open the Interactive Debugger!

```
load "tracelib.ring"

test1()

func test1
        x = 10
        see :test1 + nl
        t = 12
        BreakPoint()
        see "After breakpoint!" +nl
        see "t = " + t + nl
        see "End of program!" + nl
```

Screen Shots:

We have the Interactive Debugger at the Breakpoint!

```
5 # BreakPoint
6
7 load "tracelib.ring"
8
9 test1()
10
11 func test1
12         x = 10
13         see :test1 + nl
14         t = 12
15         BreakPoint()
16         see "After breakpoint!" +nl
17         see "t = " + t + nl
18         see "End of program!" + nl
19
20
```

```
C:\WINDOWS\system32\cmd.exe - B:\ring\applications\rnote/batch\run "B:/ring/ringlibs/tracelib/samples/sample6.ring"
test1


============================================================
                    Interactive Debugger
============================================================
Command (Exit)        : End Program
Command (Cont)        : Continue Execution
Command (Locals)      : Print local variables names
Command (LocalsData)  : Print local variables data
Command (Globals)     : Print global variables names
Command (CallStack)   : Print call stack
We can execute Ring code
============================================================

code:> _
```

We can print the variables values

```
6
7 load "tracelib.ring"
8
9 test1()
10
11 func test1
12         x = 10
13         see :test1 + nl
14         t = 12
15         BreakPoint()
16         see "After breakpoint!" +nl
17         see "t = " + t + nl
18         see "End of program!" + nl
19
20
```

```
C:\WINDOWS\system32\cmd.exe - B:\ring\applications\rnote/batch\run "B:/ring/ringlibs/tracelib/samples/sample6.ring"
============================================================
Command (Exit)        : End Program
Command (Cont)        : Continue Execution
Command (Locals)      : Print local variables names
Command (LocalsData)  : Print local variables data
Command (Globals)     : Print global variables names
Command (CallStack)   : Print call stack
We can execute Ring code
============================================================

code:> localsdata

Variable : x      Type : NUMBER          Value : 10
Variable : t      Type : NUMBER          Value : 12

code:> _
```

We can change the variables values then continue execution

```
5 # BreakPoint
6
7 load "tracelib.ring"
8
9 test1()
10
11 func test1
12         x = 10
13         see :test1 + nl
14         t = 12
15         BreakPoint()
16         see "After breakpoint!" +nl
17         see "t = " + t + nl
18         see "End of program!" + nl
19
20
```

```
C:\WINDOWS\system32\cmd.exe - B:\ring\applications\rnote/batch\run "B:/ring/ringlibs/tracelib/samples/sample6.ring"

code:> localsdata

Variable : x      Type : NUMBER          Value : 10
Variable : t      Type : NUMBER          Value : 12

code:> x = 100

code:> t = 200

code:> cont
After breakpoint!
t = 200
End of program!
```

We can run the Interactive Debugger in the Output Window

105.16 More Syntax Flexibility

- Using braces { } in Packages/Classes/Functions

Example:

```
load "stdlib.ring"

import mypackage

new myclass {
        myfunc()
}

package mypackage
{
        class myclass
        {
                func myfunc
                {
                        print("Hello, World!\n")
                }
        }
}
```

- Using 'end' keyword after Packages/Classes/Functions

Example:

```
import mypackage
```

(continues on next page)

(continued from previous page)

```
new myclass {
        myfunc()
}

package mypackage
        class myclass
                def myfunc
                        put "Hello, World!"
                end
        end
end
```

- Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions

Example:

```
import mypackage

new myclass { myfunc() }

package mypackage
        class myclass
                func myfunc
                        see "welcome"  + nl
                endfunc
        endclass
endpackage
```

105.17 Type Hints Library

Ring 1.5 comes with the Type Hints library

Using this library we can add the type information to the source code which will be very useful for tools like

- Code Editors
- Static-Analysis

Example:

```
load "typehints.ring"

see sum(3,4) + nl ;
see sayHello("Mahmoud");

int func sum(int x,int y) {
        return x+y ;
}

string func sayHello(string name) {
        return "Hello " + name ;
}
```

The library is very powerful and support the User types (Classes) automatically!

Example:

```
load "typehints.ring"

import mypackage

test()  { main([:one,:two,:three]) }

myclass func test() {
        see "Testing User Types!" + nl
        return new myclass
}

package mypackage {
        public class myclass {
                public static void func main(list args) {
                        see "welcome" + nl
                        see args
                }
        }
}
```

Also You can use the types inside the code (not only the function prototype)

Example:

```
load "typehints.ring"

int     sum = sum(3,4)
string  msg = sayHello("Mahmoud")

see "Sum = " + sum + nl + msg + nl


int func sum(int x,int y) {
        return x+y ;
}

string func sayHello(string name) {
        return "Hello " + name ;
}
```

Rules:

- To use the types in the function prototype, You must use '(' and ')' around parameters
- To use the types in the function code, You must set the variable value (Assignment).

Note: Ring is a dynamic language, No type checking will be done by the compiler.

105.18 Better Quality

Based on Ring usage every day in practical projects

Ring 1.5 is more stable and more productive!

We are adding features based on clear vision and practical needs.

Also the documentation is better.

105.19 What is new in Ring 1.5.1?

- Better Documentation
- StdLib - Factorial() function update
- RingVM - Better code for clearing the stack in the Class Region.
- Sample : 3D Cube (OpenGL) + Texture Image using GameLib (RingAllegro)

Source Code:

```
load "gamelib.ring"
load "opengl21lib.ring"

func main

        new GraphicsApp {
                start()
        }


class GraphicsApp from GraphicsAppBase

        TITLE = "Ring Cube"

        bitmap texture

        xrot = 0.0
        yrot = 0.0
        zrot = 0.0

        func loadresources

                bitmap = al_load_bitmap("ring.bmp")
                texture = al_get_opengl_texture(bitmap)

        func destroyResources

                al_destroy_bitmap(bitmap)

        func drawScene

                w = 800 h = 600
                ratio =  w / h
```

(continues on next page)

(continued from previous page)

```
glViewport(0, 0, w, h)
glMatrixMode(GL_PROJECTION)
glLoadIdentity()

gluPerspective(45,ratio,1,100)
glMatrixMode(GL_MODELVIEW)
glLoadIdentity()

glEnable(GL_TEXTURE_2D)
glShadeModel(GL_SMOOTH)
glClearColor(0.0, 0.0, 0.0, 0.5)
glClearDepth(1.0)
glEnable(GL_DEPTH_TEST)
glEnable(GL_CULL_FACE)
glDepthFunc(GL_LEQUAL)
glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST)

glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
glLoadIdentity();
glTranslatef(0.0,0.0,-5.0);

glRotatef(xrot,1.0,0.0,0.0);
glRotatef(yrot,0.0,1.0,0.0);
glRotatef(zrot,0.0,0.0,1.0);

glBindTexture(GL_TEXTURE_2D, texture)

glBegin(GL_QUADS)
        // Front Face
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
        // Back Face
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
        // Top Face
        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0,  1.0,  1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
        // Bottom Face
        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0, -1.0, -1.0)
        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0, -1.0, -1.0)
        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
        // Right face
        glTexCoord2f(1.0, 0.0) glVertex3f( 1.0, -1.0, -1.0)
        glTexCoord2f(1.0, 1.0) glVertex3f( 1.0,  1.0, -1.0)
```

(continues on next page)

(continued from previous page)

```
                        glTexCoord2f(0.0, 1.0) glVertex3f( 1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 0.0) glVertex3f( 1.0, -1.0,  1.0)
                        // Left Face
                        glTexCoord2f(0.0, 0.0) glVertex3f(-1.0, -1.0, -1.0)
                        glTexCoord2f(1.0, 0.0) glVertex3f(-1.0, -1.0,  1.0)
                        glTexCoord2f(1.0, 1.0) glVertex3f(-1.0,  1.0,  1.0)
                        glTexCoord2f(0.0, 1.0) glVertex3f(-1.0,  1.0, -1.0)
                glEnd()

                xrot += 0.3
                yrot += 0.2
                zrot += 0.4


class GraphicsAppBase

        display event_queue ev timeout
        timer  redraw   = true

        FPS             = 60

        SCREEN_W        = 800
        SCREEN_H        = 600

        KEY_UP          = 1
        KEY_DOWN        = 2
        KEY_LEFT        = 3
        KEY_RIGHT       = 4

        Key = [false,false,false,false]

        TITLE = "Graphics Application"

        func start

                SetUp()
                loadResources()
                eventsLoop()
                destroy()

        func setup

                al_init()
                al_init_image_addon()
                al_set_new_display_flags(ALLEGRO_OPENGL)
                display = al_create_display(SCREEN_W,SCREEN_H)
                al_set_Window_title(display,TITLE)
                al_clear_to_color(al_map_rgb(0,0,0))
                event_queue = al_create_event_queue()
                al_register_event_source(event_queue,
                        al_get_display_event_source(display))
                ev = al_new_allegro_event()
```

(continues on next page)

(continued from previous page)

```
            timeout = al_new_allegro_timeout()
            al_init_timeout(timeout, 0.06)
            timer = al_create_timer(1.0 / FPS)
            al_register_event_source(event_queue,
                    al_get_timer_event_source(timer))
            al_start_timer(timer)
            al_install_mouse()
            al_register_event_source(event_queue,
                    al_get_mouse_event_source())
            al_install_keyboard()
            al_register_event_source(event_queue,
                    al_get_keyboard_event_source())

    func eventsLoop

            while true
                    al_wait_for_event_until(event_queue, ev, timeout)
                    switch al_get_allegro_event_type(ev)
                    on ALLEGRO_EVENT_DISPLAY_CLOSE
                            exit
                    on ALLEGRO_EVENT_TIMER
                            redraw = true
                    on ALLEGRO_EVENT_MOUSE_AXES
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                    on ALLEGRO_EVENT_MOUSE_ENTER_DISPLAY
                            mouse_x = al_get_allegro_event_mouse_x(ev)
                            mouse_y = al_get_allegro_event_mouse_y(ev)
                    on ALLEGRO_EVENT_MOUSE_BUTTON_UP
                            exit
                    on ALLEGRO_EVENT_KEY_DOWN
                            switch al_get_allegro_event_keyboard_keycode(ev)
                                    on ALLEGRO_KEY_UP
                                            key[KEY_UP] = true
                                    on ALLEGRO_KEY_DOWN
                                            key[KEY_DOWN] = true
                                    on ALLEGRO_KEY_LEFT
                                            key[KEY_LEFT] = true
                                    on ALLEGRO_KEY_RIGHT
                                            key[KEY_RIGHT] = true
                            off
                    on ALLEGRO_EVENT_KEY_UP
                            switch al_get_allegro_event_keyboard_keycode(ev)
                                    on ALLEGRO_KEY_UP
                                            key[KEY_UP] = false
                                    on ALLEGRO_KEY_DOWN
                                            key[KEY_DOWN] = false
                                    on ALLEGRO_KEY_LEFT
                                            key[KEY_LEFT] = false
                                    on ALLEGRO_KEY_RIGHT
                                            key[KEY_RIGHT] = false
                                    on ALLEGRO_KEY_ESCAPE
```

(continues on next page)

(continued from previous page)

```
                                        exit
                                off
                        off
                        if redraw and al_is_event_queue_empty(event_queue)
                                redraw = false
                                drawScene()
                                al_flip_display()
                        ok
                        callgc()
                end

        func destroy

                destroyResources()
                al_destroy_timer(timer)
                al_destroy_allegro_event(ev)
                al_destroy_allegro_timeout(timeout)
                al_destroy_event_queue(event_queue)
                al_destroy_display(display)

        func loadresources

        func drawScene

        func destroyResources
```

Screen Shot:

105.20 What is new in Ring 1.5.2?

- Documentation - Chapter "Applications developed in little hours" is updated
- Ring Notepad - Display programs output in the output window on all platforms
- Form Designer - Help Menu - Open CHM/PDF files without displaying the console window
- Form Designer - Better response to Resize/Move Events when moving the Mouse quickly
- Form Designer - New/Open/Save As, will open the Controller class in Ring Notepad
- Form Designer - Added "Close Form" option to the file menu
- Ring Notepad - Run, will save the current file (Also the opened Form) automatically
- GetQuotesHistory Application - Updated to work on MacOS X and Qt 5.2
- Calculator Application - Updated to include more features!
- RingVM - Classification for Environment Errors (Check Chapter : Language Reference)
- RingQt - New methods added to QAllEvents for faster Events execution
- RingQt - Fusion Black Style - Better colors for disabled controls
- Scripts - For building Ring on Fedora Linux (Check Chapter : Building From Source Code)

Screen Shot:

105.21 What is new in Ring 1.5.3?

- Form Designer : Close Action will notify Ring Notepad to be able to open the Form again
- Form Designer : Save Action will open the controller class in Ring Notepad
- Form Designer : Keep current control selected when selecting many controls using CTRL Key
- Form Designer : Nice form back color when used in Ring Notepad (Style: Modern Black)
- RingOpenSSL : Updated to support newer versions like OpenSSL 1.1
- Building Scripts : Updated to work on Fedora 26 (64bit)
- OpenGL : New Sample - Many Cubes (samples/3D/manycubes)

Screen Shot:

- RingQt : Add QDateTime Class
- RingQt : New methods added to QMenu and QCursor Classes

Example:

```
load "guilib.ring"
```

(continues on next page)

(continued from previous page)

```
new qApp {
        win = new qwidget() {
                setwindowtitle("Context Menu")
                resize(400,400)
                myfilter = new qAllEvents(win) {
                        setContextmenuEvent("mymenu()")
                }
                installeventfilter(myfilter)
                show()
        }
        exec()
}


func mymenu

        new qMenu(win) {
                oAction = new qAction(win) {
                        settext("new")
                        setClickEvent("See :New")
                }
                addaction(oAction)
                oAction = new qAction(win) {
                        settext("open")
                        setClickEvent("See :Open")
                }
                addaction(oAction)
                oAction = new qAction(win) {
                        settext("save")
                        setClickEvent("See :Save")
                }
                addaction(oAction)
                oAction = new qAction(win) {
                        settext("close")
                        setClickEvent("See :Close")
                }
                addaction(oAction)
                oCursor  = new qCursor()
                exec(oCursor.pos())
        }
```

- Compiler : Support using _ in numbers

Example:

```
x = 1_000_000
see type(x)+nl
see x+1+nl
```

Output:

```
NUMBER
100000001
```

- Compiler : Support using f after numbers

Example:

```
x = 19.99f
see type(x) + nl
```

Output:

```
NUMBER
```

- Google API Shortener Application

Screen Shots:

- TicTacToe 3D Game

Screen Shot:

105.22 What is new in Ring 1.5.4?

- CalmoSoft Fifteen Puzzle Game 3D
- Ring Notepad - New Styles
- Ring Notepad - Better Toolbar Style
- Ring Notepad - View Modes
- Ring Notepad - QPlainTextEdit - don't set back color for the scroll bars
- Ring Notepad - Style Fusion (White) - use Silver color for comments
- Ring Notepad - Tab and Shift-Tab - Indent multiple lines
- Form Designer - Better Toolbar Style
- Form Designer - Nice backcolor for Window Flags and Menubar Designer
- Form Designer - Default back color for controls
- RingQt - Added grab() and windowHandle() methods to QWidget class
- RingQt - Added new methods to QPixmap Class
- **RingQt - Added Classes :-**
  - QScreen
  - QWindow

  - QGuiApplication
  - QTextBrowser
- Code Generator for Extensions - Nonew Option - Support Parent Class
- Ring VM - Internal Implementation - Pass state to Strings and Lists objects
- Ring VM - Garbage Collector - Memory Pool for Small Objects
- Ring VM - Better code for Saving/Restoring the State

CHAPTER

SIX

WHAT IS NEW IN RING 1.6?

In this chapter we will learn about the changes and new features in Ring 1.6 release.

106.1 List of changes and new features

Ring 1.6 comes with many new features!

- Employee Application
- New Tool: Ring2EXE
- Better Ring For Android
- New Tool : Folder2qrc
- Better Scripts for building Ring
- RingConsoleColors Extension
- RingMurmurHash Extension
- Better Ring Notepad
- Better RingQt
- Better StdLib
- Better RingVM
- Better RingREPL
- Using Tab instead of char(9)
- Using CR as Carriage return
- Using the ! operator as not
- Using && and || operators
- Using ? to print expression then new line

106.2 Employee Application

The Employee application is added to ring/applications

Developer: Ahmed Hassouna

106.3 New Tool: Ring2EXE

In Ring 1.6 we have a nice tool called Ring2EXE (Written in Ring itself)

Using Ring2EXE we can distribute applications quickly for Windows, Linux, macOS and Mobile devices

Read the chapter "Distributing Ring Applications using Ring2EXE" for more information!

106.4 Better Ring For Android

Ring For Android (using RingQt) is updated to use the Ring Object File (*.ringo) instead of using many source code files (*.ring)

The next screen shot is an example of building the cards game for Android

We are using cards.ringo instead of cards.ring

If you have large project (many *.ring files) it will use only one *.ringo file.

To prepare Qt project to distribute RingQt application for Mobile use Ring2EXE

Example

```
ring2exe cards.ring -dist -mobileqt
```

Example (2)

```
ring2exe formdesigner.ring -dist -mobileqt
```

106.5 New Tool: Folder2qrc

When we have large RingQt project that contains a lot of images and files, We need to add these files to the resource file (*.qrc) when distributing applications for Mobile devices.

Instead of adding these files one by one, Ring 1.6 comes with a simple tool that save our time, It's called Folder2qrc.

Example:

```
folder2qrc formdesigner.ring
```

We determine the main source file while we are in the application folder, and Folder2qrc will check all of the files in the current folder and sub folders, Then add them to the resource file after the mainfile.ringo (In our example this will be formdesigner.ringo)

The output file will be : project.qrc

You can open it and remove the files that you don't need in the resources!

106.6 Better Scripts for building Ring

Ring 1.6 comes with better scripts for building Ring from source code.

The updates are tested on 32bit and 64bit systems on Windows, Linux (Ubuntu,Fedora) and macOS.

The scripts for Windows are updated to find the Visual C/C++ compiler based on your Visual Studio version.

106.7 RingConsoleColors Extension

Using the RingConsoleColors extension we can easily change the colors used in our console applications

For more information check the RingConsoleColors chapter in the documentation.

106.8 RingMurmurHash Extension

Ring 1.6 comes with the RingMurmurHash extension!

Developer: Hassan Ahmed

Example:

```
load "murmurhashlib.ring"

key = "Ring Language"

see murmurhash3_x86_32(key, 0, 0) + nl // Output: 1894444853
see murmurhash3_x86_32(key, 0, 1) + nl // Output: 70eaef35
```

For more information check the RingMurmurHash chapter in the documentation.

106.9 Better Ring Notepad

Ring Notepad comes with the next updates

(1) Automatic setting for the Main File when we Run the application (using the Main file buttons).

(2) Main File - Automatic save before running.

(3) When we run GUI application - don't change the focus to the text box used for the input in the Output Window.

(4) A button and option to run web applications

For Windows users, Ring 1.6 comes with Apache Web server!

We can run any web application from any folder directly without doing any configuration.

Source Code : C:/Users/samir/Desktop/mytest.ring

```
#!ring -cgi

Load "weblib.ring"

Import System.Web

func main
        New Page
        {
                Text("Hello, World!") newline()
                Text("Welcome to web development using Ring!")
                newline()
                for x = 1 to 5 {
                        TEXT("Number: " + x) newline()
                }
                text("Enjoy!")
        }
```

Test

localhost:8080/mytest.ring

Hello, World!
Welcome to web development using Ring!
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
Enjoy!

(5) Tools - Operating System - Terminal (Command Prompt) & Files Manager (Explorer).

So we can quickly open the Command Prompt or the Explorer at the application folder.

(6) Support *.sh & *.bat extensions.

(7) New Menu: Distribute

106.10 Better RingQt

RingQt comes with the next updates

(1) QAllEvents - getkeytext() Method

(2) QSQLQuery - exec_2() Method

(3) QDockWidget Events

(4) AppFile() Function

(5) IsMobile() Function

(6) QRegion Class

(7) QCoreApplication class

106.11 Better StdLib

StdLib comes with the next updates

(1) Factors() function is updated (Return the output instead of printing it)

(2) Palindrome() function is updated (Return the output instead of printing it)

(3) Using stdlibcore.ring we can use the StdLib functions (Without Classes)

Also this is useful when developing standalone console applications

Because using stdlib.ring (functions & classes) will load libraries like RingLibCurl, RingOpenSSL, etc.

(4) New Functions

- SystemSilent(cCommand) Function : Execute system commands without displaying the output.
- OSCreateOpenFolder(cFolder) : Create folder then change the current folder to this new folder
- OSCopyFolder(cParentFolder,cFolderNameToCopy) : Copy folder to the current directory
- OSDeleteFolder(cFolder) : Delete Folder
- OSCopyFile(cFileName) : Copy File to the current directory
- OSDeleteFile(cFileName) : Delete File
- OSRenameFile(cOldFileName,cNewFileName) : Rename file

106.12 Better RingVM

RingVM comes with the next updates

(1) Support using many getter methods in Expressions

(2) Support using this & self in setter/getter/normal methods without calling setter/getter methods.

(3) TempName() function is updated (Better Code)

(4) ExeFileName() function is updated (Better Code)

(5) Private Attributes - Support re-usage in the class region (After the keyword private)

(6) Ring API : ring_scanner_runobjstring()

(7) ring_state_setvar() function

106.13 Better RingREPL

RingREPL comes with the next updates

(1) RingREPL will get command line parameters to determine the style.

(2) Setting RingREPL Style based on Ring Notepad Style.

106.14 Using Tab instead of char(9)

The variable Tab is defined to be used instead of char(9)

Example (1):

```
see :one + nl + tab + :two + nl + tab + tab + :three
```

Output:

```
one
        two
                three
```

You can change the variable to another value

Example (2):

```
tab = "  "
see :one + nl + tab + :two + nl + tab + tab + :three
```

Output:

```
one
  two
    three
```

106.15 Using CR as Carriage return

The next example count from 1 to 10 in the same line during 10 seconds

```
load "stdlibcore.ring"
for x = 1 to 10 see x sleep(1) see cr next
```

106.16 Using the ! operator as not

We have = and != in the Ring language

But for the logical operators we have and, or & not

Now we can use the ! operator as not

Example:

```
if ! false
        see "True!" + nl
ok
```

Output

```
True!
```

106.17 Using && and || operators

In Ring we have the next keywords for the logical operations

- and
- or
- not

Now we have also the next operators

- &&
- ||
- !

Example:

```
if one() and two()
        see "Test1 - Fail" + nl
else
        see "Test1 - Pass" + nl
ok

if two() or one()
        see "Test2 - Pass" + nl
else
        see "Test2 - Fail" + nl
ok

if one() && two()
        see "Test3 - Fail" + nl
else
        see "Test3 - Pass" + nl
ok
```

(continues on next page)

(continued from previous page)

```
if two() || one()
        see "Test4 - Pass" + nl
else
        see "Test4 - Fail" + nl
ok

func one return True

func two return False
```

Output:

```
Test1 - Pass
Test2 - Pass
Test3 - Pass
Test4 - Pass
```

106.18 Using ? to print expression then new line

It's common to print new line after printing an expression, We can use the ? operator to do that!

Example:

```
? "Hello, World!"
for x = 1 to 10
        ? x
next
```

Output:

```
Hello, World!
1
2
3
4
5
6
7
8
9
10
```

CHAPTER

SEVEN

WHAT IS NEW IN RING 1.7?

In this chapter we will learn about the changes and new features in Ring 1.7 release.

107.1 List of changes and new features

Ring 1.7 comes with many new features!

- New Command: Load Package
- ringvm_see() and ringvm_give() functions
- ring_state_new() and ring_state_mainfile() functions
- Better Trace Library
- Better Ring Notepad
- Better RingQt
- Better Ring2EXE
- Better RingZip
- Better Documentation
- Better Ring VM
- RingLibuv Extension

107.2 New Command: Load Package

Using the 'load' command we can use many ring source files in the same project

But all of these files will share the same global scope

Now we have the "Load Package" command too

Using "Load Package" we can load a library (*.ring file) in new global scope

This is very useful to create libraries that avoid conflicts in global variables

Example:

File: loadpackage.ring

```
x = 100
? "Hello, World!"
load package "testloadpackage.ring"

? x
test()
```

File: testloadpackage.ring

```
? "Hello from testloadpackage.ring"

x = 1000

test()

func test
        ? x
```

Output:

```
Hello, World!
Hello from testloadpackage.ring
1000
100
1000
```

107.3 ringvm_see() and ringvm_give() functions

Using the ringvm_see() function we can redefine the behavior of the See command

Also we can use ring_see() to have the original behavior

Example:

```
see "Hello world" + nl
see 123 + nl
see ["one","two","three"]
see new point {x=10 y=20 z=30}

func ringvm_see t
        ring_see("We want to print: ")
        ring_See(t)

class point x y z
```

Output:

```
We want to print: Hello world
We want to print: 123
We want to print: one
two
three
```

(continues on next page)

(continued from previous page)

```
We want to print: x: 10.000000
y: 20.000000
z: 30.000000
```

Using the ringvm_give() function we can redefine the behavior of the Give command

Also we can use ring_give() to have the original behavior

Example:

```
see "Name: " give name
see "Hello " + name

func ringvm_give
        see "Mahmoud" + nl
        return "Mahmoud"
```

Output:

```
Name: Mahmoud
Hello Mahmoud
```

107.4 ring_state_new() and ring_state_mainfile() functions

Using ring_state_new() and ring_state_mainfile() we can run Ring programs from Ring programs

But unlike ring_state_main(), Here we can control when to delete the Ring state!

This is important when we run GUI programs from GUI programs

Because they will share the GUI Library (RingQt), And In this case the caller will call

qApp.Exec()

So the sub program, will not stop and will return to the Main program

Here deleting the State of the sub programs will lead to a problem when we run the sub program events

So keeping the state is important for sub GUI programs hosted in GUI programs.

107.5 Better Trace Library

The Trace library is updated, In the Debugger at break points we have now the "callstack" command

This command will print the functions call stack.

Example:

```
load "tracelib.ring"

func main
        ? "Hello from main!"
        test1()
```

(continues on next page)

(continued from previous page)

```
func test1
        ? "Hello from test1!"
        test2()

func test2
        ? "Hello from test2!"
        test3()

func test3
        ? "Hello from test3!"
        breakpoint()
```

107.6 Better Ring Notepad

Ring Notepad comes with the next updates

(1) Support *.cf extension

(2) Using Hash function (SHA256) for better "Save Changes?" message

(3) Ring Notepad - X Button - Ask for saving changes?

107.7 Better RingQt

The next classes are added to RingQt

(1) QStackedWidget
(2) QCalendarWidget
(3) QOpenGLFunctions
(4) QOpenGLContext
(5) QSurfaceFormat
(6) QOpenGLWidget
(7) QOpenGLVersionProfile
(8) QOpenGLFunctions_3_2_Core
(9) QVector2D
(10) QVector3D
(11) QVector4D
(12) QQuaternion
(13) QMatrix4x4
(14) QOpenGLPaintDevice
(15) QPaintDevice
(16) QOpenGLTimerQuery
(17) QOpenGLDebugLogger
(18) QOpenGLFramebufferObject
(19) QOpenGLVertexArrayObject
(20) QOpenGLBuffer
(21) QOpenGLShaderProgram
(22) QOpenGLShader
(23) QOpenGLTexture

107.8 Better Ring2EXE

Ring2EXE is updated to works as expected when we don't have a C/C++ compiler

Where we can distribute applications and get (exe file and ringo file) in this case.

107.9 Better RingZip

The library is updated to support extracting files contains sub folders!

107.10 Better Documentation

(1) RingQt Classes Chapter - The classes are sorted.

107.11 Better Ring VM

(1) Better Error Message
(2) List2Str() function support lists contains numbers
(3) Correct support for numbers contains _ as separator
(4) Creating lists without variables (statement –> Expression –> List)
(5) IsNULL() - Not case sensitive - treat Null and null like NULL
(6) Support adding the Self object to an attribute in this object
(7) Using ':' operator then keyword will create lower case literal
(8) Printing objects - respect decimals() function
(9) When literal is not closed - determine the start of the literal
(10) Better message when printing objects contains lists
(11) VarPtr() - Support getting a pointer to variables in the local scope
(12) replace performance instructions with normal instructions when creating new threads

107.12 RingLibuv Extension

Ring 1.7 comes with the RingLibuv extension

Libuv is a multi-platform support library with a focus on asynchronous I/O.

Example (Events Loop):

```
load "libuv.ring"

counter = 0
idler = NULL

func main
        idler = new_uv_idle_t()
        uv_idle_init(uv_default_loop(), idler)
        uv_idle_start(idler, "wait()")
        ? "Idling..."
        uv_run(uv_default_loop(), UV_RUN_DEFAULT);
        uv_loop_close(uv_default_loop());
```

(continues on next page)

(continued from previous page)

```
        destroy_uv_idle_t(idler)

func wait
        counter++
        if counter >= 100000
                uv_idle_stop(idler)
        ok
```

Output:

```
Idling...
```

Example (Server):

```
load "libuv.ring"
load "objectslib.ring"

? "Testing RingLibuv - Server Side - Using Classes"

open_object(:MyServer)

class MyServer from ObjectControllerParent

        DEFAULT_PORT    = 13370
        DEFAULT_BACKLOG = 1024

        addr    = new_sockaddr_in()
        server  = NULL
        client  = NULL
        myloop  = NULL

        func start
                myloop = uv_default_loop()
                server = new_uv_tcp_t()
                uv_tcp_init(myloop, server)
                uv_ip4_addr("127.0.0.1", DEFAULT_PORT, addr)
                uv_tcp_bind(server, addr, 0)
                r = uv_listen(server, DEFAULT_BACKLOG, Method(:newconnection) )
                if r
                        ? "Listen error " + uv_strerror(r)
                        return 1
                ok
                uv_run(myloop, UV_RUN_DEFAULT)
                destroy_uv_tcp_t(server)
                destroy_uv_sockaddr_in(addr)

        func newconnection
                ? "New Connection"
                aPara   = uv_Eventpara(server,:connect)
                nStatus = aPara[2]
                if nStatus < 0
                        ? "New connection error : " + nStatus
```

(continues on next page)

(continued from previous page)

```
                        return
                ok
                client = new_uv_tcp_t()
                uv_tcp_init(myloop, client)
                if uv_accept(server, client) = 0
                                uv_read_start(client, uv_myalloccallback(),
                                                        Method(:echo_read))
                ok

        func echo_read
                aPara = uv_Eventpara(client,:read)
                nRead = aPara[2]
                buf   = aPara[3]
                if nRead > 0
                        req = new_uv_write_t()
                                wrbuf = uv_buf_init(get_uv_buf_t_base(buf), nread)
                        uv_write(req, client, wrbuf, 1, Method(:echo_write))
                        ? uv_buf2str(wrbuf)
                        message = "message from the server to the client"
                        buf = new_uv_buf_t()
                        set_uv_buf_t_len(buf,len(message))
                        set_uv_buf_t_base(buf,varptr("message","char *"))
                        uv_write(req, client, buf, 1, Method(:echo_write))
                ok

        func echo_write
                aPara = uv_Eventpara(client,:read)
                req   = aPara[1]
```

Output:

When we run the client, We will see the message "New Connection"

Then the message "hello from the client"

```
Testing RingLibuv - Server Side - Using Classes
New Connection
hello from the client
```

Example (Using Threads):

```
load "libuv.ring"
load "objectslib.ring"

? "Testing RingLibuv - Threads - Using Classes"

open_object(:MyThreads)

class MyThreads from ObjectControllerParent

        func Start
                one_id = new_uv_thread_t()
                two_id = new_uv_thread_t()
```

(continues on next page)

(continued from previous page)

```
                uv_thread_create(one_id, Method(:One))
                uv_thread_create(two_id, Method(:Two))
                uv_thread_join(one_id)
                uv_thread_join(two_id)
                destroy_uv_thread_t(one_id)
                destroy_uv_thread_t(two_id)

        func one
                ? "Message from the First Thread!"

        func Two
                ? "Message from the Second Thread!"
```

Output:

```
Testing RingLibuv - Threads - Using Classes
Message from the First Thread!
Message from the Second Thread!
```

For more information about this extension (RingLibuv) check the chapter: Using RingLibuv

CHAPTER

EIGHT

WHAT IS NEW IN RING 1.8?

In this chapter we will learn about the changes and new features in Ring 1.8 release.

108.1 List of changes and new features

Ring 1.8 comes with the next features!

- Better Performance
- Find in files Application
- String2Constant Application
- StopWatch Application
- More 3D Samples
- Compiling on Manjaro Linux
- Using This in the class region as Self
- Default value for object attributes is NULL
- The For Loops uses the local scope
- Merge binary characters
- FoxRing Library
- Better Form Designer
- Better Cards Game
- Better RingQt
- Better Code Generator For Extensions
- Better Ring Compiler and VM
- Notes to extensions creators

108.2 Better Performance

Ring 1.8 is faster than Ring 1.7

The performance gain is between 10% and 100% based on the application.

Check the 3D samples in this release to get an idea about the current performance.

For more information check the Performance Tips chapter.

108.3 Find in files Application

Ring 1.8 comes with Find in files application

108.4 String2Constant Application

Ring 1.8 comes with String2Constant application

Using this tool we can convert the source code to be based on constants instead of string literals

Then we can store constants in separate source code files that we can translate to different languages

Where we can have special file for each language, like (English.ring, Arabic.ring and so on)

Using this simple tool, the Form Designer is translated to the Arabic language.

For more information check the Multi-language Applications chapter.

108.5 StopWatch Application

Ring 1.8 comes with StopWatch application

108.6 More 3D Samples

Ring 1.8 comes with more 3D Samples

The next screen shot for the Top-Down view - Many levels of cubes sample

The next screen shot for the Camera Sample

The next screen shot for the Camera and background sample

Developer : Azzeddine Remmal

108.7 Compiling on Manjaro Linux

Ring 1.8 is tested on Manjaro Linux too

Tests by : Iip Rifai

108.8 Using This in the class region as Self

The class region is the region that comes after the class name and before any method.

Now we can use This in the class region as Self.

Example:

```
func main

        o1 = new program {
                test()
        }

        ? o1

class program

        this.name = "My Application"
        this.version = "1.0"
        ? name ? version

        func test
                ? "Name    = " + name
                ? "Version = " + version
```

Output

```
My Application
1.0
Name    = My Application
Version = 1.0
name: My Application
version: 1.0
```

Note: When we use braces to change the current active object, Using This we can still point to the class.

Tip: The difference between This and Self is that Self point to the current active object that we can change using braces.

Remember that in most cases we don't need to use This or Self in the class region

We can write

```
class program name version
```

Or

```
class program name="My Application" version="1.0"
```

Note: We use This or Self in the class region just to avoid conflict with global variables that are defined with the same

name.

108.9 Default value for object attributes is NULL

Starting from Ring 1.8 the default value for object attributes is NULL

In Ring, the NULL value is just an empty string or a string that contains "NULL"

We can check for NULL values using the isNULL() function

Example:

```
oProgram = new Program
? oProgram.name
? oProgram.version
? isNULL(oProgram.name)
? isNULL(oProgram.version)
oProgram { name="My Application" version="1.0" }
? isNULL(oProgram.name)
? isNULL(oProgram.version)
? oProgram

class program
        name
        version
```

Output:

```
NULL
NULL
1
1
0
0
name: My Application
version: 1.0
```

In previous versions of Ring, trying to access the object attribute before assigning a value to it

Will lead to runtime error and you can't check it using isnull()

The only way was assigning a value or using try/catch/end

We changed this behavior so we can have full control in seamless way.

108.10 The For Loops uses the local scope

In Ring 1.8, when the For Loop defines new identifier (variable) it will define it in the local scope.

Example:

```
x = 10
? x             # Print 10
test1()
? x             # Print 10
test2()
? x             # Print 10

func test1
        for x = 1 to 5
        next
        ? x     # Print 6

func test2
        list = 1:5
        for x in list
        next
        ? x     # Print NULL (The "For In" loop will kill the reference after the loop)
```

Output:

```
10
6
10
NULL
10
```

108.11 Merge binary characters

From Ring 1.0 we can create binary strings and do operations on these strings.

Now in Ring 1.8, we can get individual characters from these strings and merge them together using the '+' operator.

Example:

```
cStr = "Welcome"
? cstr[1] + cstr[2] + cStr[5]
v = cstr[1] + cstr[2] + cStr[5]
? v
? len(v)
c1 = cStr[1]
? c1
aList = [1,2,3]
cStr = ""
for item in aList
        cStr += int2bytes(item)
next
```

(continues on next page)

(continued from previous page)

```
? "All String"
? len(cStr)
? "First Part"
n1 = cStr[1] + cStr[2] + cStr[3] + cStr[4]
? len(n1)
? "Second Part"
n2 = cStr[5] + cStr[6] + cStr[7] + cStr[8]
? len(n2)
? "Third Part"
n3 = cStr[9] + cStr[10] + cStr[11] + cStr[12]
? len(n3)
? "All String"
cString = cStr[1] + cStr[2] + cStr[3] + cStr[4] +
          cStr[5] + cStr[6] + cStr[7] + cStr[8] +
          cStr[9] + cStr[10] + cStr[11] + cStr[12]
? len(cString)
? ascii(cStr[1])
? len(cStr[2])
```

Output:

```
Weo
Weo
3
W
All String
12
First Part
4
Second Part
4       }
Third Part
4
All String
12
1
1
```

108.12 FoxRing Library

Developer: Jose Rosado

A class with some of the functions I used in FoxPro

Example:

```
Load "foxring.ring"

mf = new frFunctions
? mf.frAbs(-45)
? mf.frAbs(10-30)
```

(continues on next page)

(continued from previous page)

```
? mf.frAbs(30-10)

? mf.frTransform("    Ring is a good language    ",
                 "@! !!!!
```

108.15 Better RingQt

- The next classes are added to RingQt

(1) QTabBar

(2) QFile

(3) QFileDevice

(4) QStandardPaths

(5) QDir

(6) QQuickWidget

(7) QQmlError

(8) QScrollBar

- RingQt for Android is updated to support modern versions of Qt

Tested using

(1) Qt 5.5.1

(2) Qt 5.9.5

(3) Qt 5.11.0

- In RingQt for Android, The Ring Object File (ringo) will be executed directly from resources.

108.16 Better Code Generator For Extensions

New Option: StaticMethods

Starting from Ring 1.8 the code generator support the staticmethods option.

So the code generator can know that the class doesn't need an object to call the methods.

Example:

```
<class>
name: QStandardPaths
para: void
nonew
staticmethods
</class>

QString displayName(QStandardPaths::StandardLocation type)
QString findExecutable(QString executableName, QStringList paths))
```

108.17 Better Ring Compiler and VM

(1) Better loading for files in relative paths

(2) Code Optimization for eval() function

(3) Better Memory Pool

(4) When embedding Ring in Ring, the error in the hosted environment will not close the host

Example:

```
? "Start the test!"

pState = ring_state_init()

ring_state_runcode(pState," ? 'Let us try having an error' ? x")

ring_state_delete(pState)

? ""
? "End of test!"
```

Output:

```
Start the test!
Let us try having an error

Line 1 Error (R24) : Using uninitialized variable : x
in file Ring_EmbeddedCode
End of test!
```

(5) The compiler will ignore new lines after keywords that expect tokens after it

Example:

```
see
"
        Hello, World!
"
test()

func
#=====================#
        Test
#=====================#

        ?
        "

        Hello from the Test function

        "
```

Output:

```
Hello, World!


Hello from the Test function
```

(6) Better code (faster) for the main loop, special loop for eval() function.

(7) Better code (faster) for tracking C pointers to avoid using NULL pointers.

(8) Better code (faster) for getting the self object using braces.

108.18 Notes to extensions creators

If you have created new extensions for Ring in the C/C++ languages.

You have to rebuild your extension (Generate the DLL file again using Ring 1.8 header files) before usage with Ring 1.8

Because we changed the internal structure of the VM, but no changes to the code are required. just rebuild.

CHAPTER

NINE

WHAT IS NEW IN RING 1.9?

In this chapter we will learn about the changes and new features in Ring 1.9 release.

109.1 List of changes and new features

Ring 1.9 comes with the next features!

- New Game : Gold Magic 800
- More Games
- Better Ring Notepad
- Better StdLib
- BigNumber Library
- RingPostgreSQL Extension
- Deploying Web applications in the Cloud
- Better RingQt
- Better Memory Management
- Better Code Generator for Extensions
- More Improvements

109.2 New Game : Gold Magic 800

The Gold Magic 800 is a new puzzle game.

Ring 1.9 comes with the Demo Version (18 Levels) and the game source code.

Steam Page (44 Levels) : https://store.steampowered.com/app/939200/Gold_Magic_800/

We can select the level

The next screen shot for level (1)

The Gold Magic 800 Level Editor

109.3 More Games

The next games are added to Ring Applications

(1) The 2048 Game

(2) The Memory Game

(3) The Wise Quadrat Game

(4) The Tessera Game

(5) The Othello Game

The next screen shot for the 2048 game on Android

The next screen shot for the Tessera game

The next screen shot for the Othello game

109.4 Better Ring Notepad

(1) New Style

(2) New Mode

(3) Tabs for opened files

(4) Support executing batch files

109.5 Better StdLib

(1) The List2Code() function is added to the StdLib

This function convert a Ring list during the runtime to Ring source code that we can save to source files.

The list may contains strings, numbers or sub lists.

Example:

```
load "stdlibcore.ring"
aList = 1:10
? list2Code(aList)
```

Output:

```
[
        1,2,3,4,5,6,7,8,9,10
]
```

(2) The Str2ASCIIList() and ASCIIList2Str() are added to the StdLib

Using these functions we can convert between string of bytes and a list of numbers where each item represent the ASCII code of one byte in the string.

So we may convert a string of bytes to ASCII List then do some operations on the list numbers (Like XOR)

Example:

```
load "stdlibcore.ring"

cStr = "MmMm"
```

(continues on next page)

(continued from previous page)

```
aList = Str2ASCIILIST(cStr)
? aList

cStr2 = ASCIIList2Str(aList)
? cStr2
? len(cStr2)
```

Output:

```
77
109
77
109

MmMm
4
```

109.6 BigNumber Library

Using the BigNumber library we can do arithmetic operations on huge numbers.

Example:

```
load "bignumber.ring"

num1 = "6234567890123456789167834512345678 9"    ### Big
num2 =  "123789456789012341987123654 5"          ### Small
num3 =     "64"                                  ### Divide Small
num4 = "765432"
num5 =      "3"                                  ### Power

? "Add big numbers:"
a1 = new BigNumber(num1)        a1.Print()
a2 = new BigNumber(num2)        a2.Print()
a3 = a1 + a2                    a3.Print() ? nl

? "Substract big numbers:"
a1 = new BigNumber(num1)        a1.Print()
a2 = new BigNumber(num2)        a2.Print()
a3 = a1 - a2                    a3.Print() ? nl

? "Multiply big numbers:"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num2)        a2.print()
a3 = a1 * a2                    a3.print() ? nl

? "Divide big numbers:"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num2)        a2.print()
a3 = a1 / a2                    a3.print() ? nl
```

(continues on next page)

(continued from previous page)

```
? "Divide big numbers: by very small number"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num3)        a2.print()
a3 = a1 / a2                    a3.print() ? nl

? "Power of big number:"
a1 = new BigNumber(num1)        a1.print()
a2 = new BigNumber(num5)        a2.print()
a3 = a1 ^ a2                    a3.print() ? nl
```

Output:

```
Add big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
6234568013912913578180176499469333 4


Substract big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
5234568766334000000155492525222024 4


Multiply big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
7717737724326015010346217871419745473643247278011968230515400 5


Divide big numbers:
6234567890123456789167834512345678 9
1237894567890123419871236545
50364288


Divide big numbers: by very small number
6234567890123456789167834512345678 9
64
9741512328317901233074741425540 12


Power of big number:
6234567890123456789167834512345678 9
3
2423366362614711720923471460317270 04 (Output continue in next line)
3716981956283439342389882561522895 08 (Output continue in next line)
4939646110432289716923898608970 69
```

For more information check the BigNumber Library Chapter in the documentation

109.7 RingPostgreSQL Extension

Ring 1.9 provide native support for PostgreSQL database using the RingPostgreSQL Extension.

Example:

```
load "postgresqllib.ring"

conninfo = "user=postgres password=sa dbname = mahdb"

exit_nicely = func conn {
        PQfinish(conn)
        shutdown(1)
}

conn = PQconnectdb(conninfo)

if (PQstatus(conn) != CONNECTION_OK)
        fputs(stderr, "Connection to database failed: "+PQerrorMessage(conn))
                call exit_nicely(conn)
ok

res = PQexec(conn, "
        DROP DATABASE mahdb;
")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Remove failed: " + PQerrorMessage(conn))
        PQclear(res)
ok
PQclear(res)


res = PQexec(conn, "CREATE DATABASE mahdb;")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Create database failed: " + PQerrorMessage(conn))
        PQclear(res)
ok


res = PQexec(conn, "
CREATE TABLE COMPANY (
                ID INT PRIMARY KEY     NOT NULL,
                NAME           TEXT    NOT NULL,
                AGE            INT     NOT NULL,
                ADDRESS        CHAR(50),
                SALARY         REAL );
")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Create Table failed: " + PQerrorMessage(conn))
        PQclear(res)
ok
PQclear(res)
```

(continues on next page)

(continued from previous page)

```
res = PQexec(conn, "
                INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
                VALUES  (1, 'Mahmoud' , 31, 'Jeddah', 10.00 ),
                                (2, 'Ahmed'   , 27, 'Jeddah', 20.00 ),
                                (3, 'Mohammed', 33, 'Egypt' , 30.00 ),
                                (4, 'Ibrahim' , 24, 'Egypt ', 40.00 );
")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Insert Table failed: " + PQerrorMessage(conn))
        PQclear(res)
ok
PQclear(res)

res = PQexec(conn, "
           select * from COMPANY
")
if PQresultStatus(res) != PGRES_TUPLES_OK
        fputs(stderr, "Select failed: " + PQerrorMessage(conn))
        PQclear(res)
        call exit_nicely(conn)
ok


nFields = PQnfields(res)
for i = 1 to nFields
                ? PQfname(res, i-1)
next

? copy("*",60)

for i = 1 to PQntuples(res)
        for j=1 to nFields
                see PQgetvalue(res, i-1, j-1) + " "
        next
        see nl
next

PQclear(res)

PQfinish(conn)
```

Output:

```
id
name
age
address
salary
************************************************************
1 Mahmoud  31 Jeddah  10
2 Ahmed    27 Jeddah  20
3 Mohammed 31 Egypt   30
```

(continues on next page)

(continued from previous page)

```
4 Ibrahim  24 Egypt   40
```

For more information check the PostgreSQL Chapter in the documentation

109.8 Deploying Web applications in the Cloud

We created a new project and tutorial to explain how to deploy Ring web applications in the Cloud using Heroku

Project : https://github.com/ring-lang/RingWebAppOnHeroku

Heroku Website : https://www.heroku.com/

For more information check the Deploying Web Applications In The Cloud chapter in the documentation.

109.9 Better RingQt

(1) The next classes are added to RingQt

- QDrag
- QMimeData
- QDropEvent
- QDragMoveEvent
- QDragEnterEvent
- QDragLeaveEvent

- QClipboard
- QChildEvent
- QGeoPositionInfo
- QGeoCoordinate
- QGeoAddress
- QGeoAreaMonitorInfo
- QGeoAreaMonitorSource
- QGeoCircle
- QGeoPositionInfoSource
- QGeoRectangle
- QGeoShape
- QGeoSatelliteInfo
- QGeoSatelliteInfoSource
- QNmeaPositionInfoSource
- QAxWidget
- QTextStream
- QPrinterInfo
- QPrintPreviewWidget
- QPrintPreviewDialog
- QPageSetupDialog
- QAbstractPrintDialog
- QPrintDialog

(2) The next classes are updated

- QAllEvents Class : New Events (ChildAdded, ChildPolished, ChildRemoved).
- QPainter Class : Updated Methods (drawConvexPloygon, drawPoints, drawPolyline) Accept Ring list of points.
- QVariant : More versions that accept different parameters when creating the object.
- QAxBase : Different versions for the dynamicCall() and querySubObject() methods.

The next example for using the QPrintPreviewDialog class

Example:

```
load "guilib.ring"

new qApp {
        win1 = new qwidget() {
                setwindowtitle("Printer Preview Dialog")
                setgeometry(100,100,800,880)
                printer1 = new qPrinter(0)
                show()
                oPreview = new qPrintPreviewDialog(printer1) {
```

(continues on next page)

(continued from previous page)

```
                        setParent(win1)
                        move(10,10)
                        setPaintrequestedevent("printPreview()")
                        exec()
                }
        }
        exec()
}

func printPreview
        printer1  {
                painter = new qpainter() {
                        begin(printer1)
                        myfont = new qfont("Times",50,-1,0)
                        setfont(myfont)
                        drawtext(100,100,"Test - Page (1)")
                        printer1.newpage()
                        drawtext(100,100,"Test - Page (2)")
                        printer1.newpage()
                        myfont2 = new qfont("Times",14,-1,0)
                        setfont(myfont2)
                        for x = 1 to 30
                                drawtext(100,100+(20*x),"Number : " + x)
                        next
                        endpaint()
                }
        }
```

Screen Shot:

Printer Preview Dialog
56.5%
3 / 3
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
Number : 6
Number : 7
Number : 8
Number : 9
Number : 10
Number : 11
Number : 12
Number : 13
Number : 14
Number : 15
Number : 16
Number : 17
Number : 18
Number : 19
Number : 20
Number : 21
Number : 22
Number : 23
Number : 24
Number : 25
Number : 26
Number : 27
Number : 28
Number : 29
Number : 30

109.10 Better Memory Management

The Ring API is updated to include RING_API_RETMANAGEDCPOINTER()

Using RING_API_RETMANAGEDCPOINTER() the Ring extensions written in C/C++ languages can return a managed pointer to Ring. This pointer can be controlled by the Ring VM using reference counting.

This is important to avoid the need to write code that free the unmanaged resources like QPixMap objects in RingQt.

Also the Code Generator for extensions is updated to automatically use RING_API_RETMANAGEDCPOINTER() based on need.

Syntax:

```
RING_API_RETMANAGEDCPOINTER(void *pValue,const char *cPointerType,
                                void (* pFreeFunc)(void *,void *))
```

For more information about RING_API_RETMANAGEDCPOINTER()

See the "Extension using the C/C++ languages" Chapter in the documentation

109.11 Better Code Generator for Extensions

(1) The code generator for extensions is updated to support the <loadfile> command

```
<loadfile> filename.cf
```

This is useful to separate the extension configuration file to many files

Example:

The file : qt_module_network.cf in the RingQt Extension

```
<comment>
                                Module (network)
</comment>

<loadfile> qabstractsocket.cf
<loadfile> qnetworkproxy.cf
<loadfile> qtcpsocket.cf
<loadfile> qtcpserver.cf
<loadfile> qhostaddress.cf
<loadfile> qhostinfo.cf
<loadfile> qnetworkrequest.cf
<loadfile> qnetworkaccessmanager.cf
<loadfile> qnetworkreply.cf
```

(2) The code generator support the <managed> option when defining classes.

Using this option, the generator will use RING_API_RETMANAGEDCPOINTER() to return the C pointer.

So the Garbage Collector will manage these C pointers.

Example

```
<class>
name: QFont
para: QString, int, int, bool
managed
</class>
```

109.12 More Improvements

(1) Ring Compiler - The Rule (Factor -> '-' Expr) changed to (Factor -> '-' Factor).

(2) Ring VM - Better Error Message.

(3) Better code for IsNULL() function - updated to check pointers too.

(4) Better code for ringvm_evalinscope() function - used by the Trace Library.

(5) Better code for Space() function.

(6) Better code for Hex() and Dec() functions.

(7) Better code for Download() function.

(8) Better code for SubStr() function.

(9) Better code for the Unsigned() function.

(10) Better code for Chdir() function.

(11) Better code for Tempname() function.

(12) Better code for HashTable - New Key (using ring_strdup() instead of strdup() function).

(13) New Function : SRandom() - Initialize random number generator.

(14) New Function : IsPointer().

(15) IsList() will not return True for C Pointers, we have now the IsPointer() function.

(16) The ? Operator is updated to respect the ringvm_see() function.

(17) Scripts to run Ring tests on Linux and macOS (Not only Windows).

(18) RingAllegro is Updated from Allegro 5.1 to Allegro 5.2.

(19) Shader Functions are added to RingAllegro.

(20) Joystick Functions are added to RingAllegro.

(21) Network functions are added to RingLibSDL.

(22) Game Engine for 2D Games - Text Class - Check the font object before usage.

(23) Game Engine for 2D Games - Automatic support for Joystick.

(24) RingLibCurl is updated to automatically use CURLOPT_COPYPOSTFIELDS when needed.

(25) Ring Notepad - Find Previous Feature.

(26) Ring Notepad - Default Style.

(27) Ring Notepad - Support using Non-English language (Like Arabic) in file names.

(28) Form Designer - Nice Alignment for Toolbox Icons.

(29) Form Designer - QAllEvents Class - Mouse Double Click Event.

(30) Find in Files - Replace and Replace All options.

(31) Qt Class Converter is updated for better conversion results.

(32) More samples are added to ring/samples/other folder.

(33) Code Refactoring for Ring Notepad, RingQt, Game Engine for 2D Games.

(34) Better Documentation - Many images are updated to reflect the current state of Ring Environment.

(35) Better Documentation - More chapters are added to the documentation.

CHAPTER

WHAT IS NEW IN RING 1.10?

In this chapter we will learn about the changes and new features in Ring 1.10 release.

110.1 List of changes and new features

Ring 1.10 comes with the next features!

- Chess Game
- Minesweeper Game
- Knight Tour Game
- Game of Life Game
- Pong Game
- Snakes and Ladder Game
- More Games
- Ring Extension for Visual Studio Code
- The Ring Package Manager (RingPM)
- Better Tests
- More Improvements

110.2 Chess Game

Chess Game (2 Players)

110.3 Minesweeper Game

The objective of the game is to clear a rectangular board containing hidden "mines" or bombs without detonating any of them.

110.4 Knight Tour Game

Move to every square on the chess board, using only the moves of a knight.

Can you visit every square in just 63 moves?

110.5 Game of Life Game

The game is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input.

One interacts with the Game of Life by creating an initial configuration and observing how it evolves, or, for advanced players, by creating patterns with particular properties.

110.6 Pong Game

Simple "tennis like" game features two paddles and a ball.

The goal is to defeat your opponent once the opponent misses a ball.

110.7 Snakes And Ladder Game

The Snakes and Ladder Game using Ring Game Engine for 2D Games

110.8 More Games

The next games are added to the application folder

- Lights Out Game
- Dots and Boxes Game
- Magic Four Game
- Sum Puzzle Game

The next screen shot for the Sum Puzzle Game

110.9 Ring Extension for Visual Studio Code

This extension support the Ring programming language in Visual Studio Code

110.10 The Ring Package Manager (RingPM)

Ring comes with a package manager (RingPM) that we can use to install, update and remove packages.

The Package Manager uses Semantic Versioning to check compatibility between packages

The Package Manager comes with the next options

```
==========================================================================
Usage   : ringpm [command]
Command : search  [keywords...]
Command : refresh : Update the Registry (Packages List)
Command : install [ <packagename> [from <UserName>] [branch <branchname>] ]
Command : list    [-u : Check updates]
Command : run     [packagename]
Command : update  <packagename>
Command : remove  <packagename>
Command : format  : Delete All Packages
Command : new     <packagename>
Command : package : Create package in the current folder
==========================================================================
```

110.11 Better Tests

New framework for Ring programming language tests that test the language.

```
Command Prompt

B:\ring\tests>test
==========================================================================================
Tests Count : 307
==========================================================================================

Test (1)    : Hello World Program (Using See)                                  --- PASS
Test (2)    : Hello World Program (Using See and nl)                           --- PASS
Test (3)    : Hello World Program (Using Put)                                  --- PASS
Test (4)    : Hello World Program (Using Put and nl)                           --- PASS
Test (5)    : Testing Procedural Programming (Part 1)                          --- PASS
Test (6)    : Testing Procedural Programming (Part 2)                          --- PASS
Test (7)    : Testing Procedural Programming (Part 3)                          --- PASS
Test (8)    : Testing Procedural Programming (Part 4)                          --- PASS
Test (9)    : Testing Procedural Programming (Part 5)                          --- PASS
Test (10)   : Testing Procedural Programming (Part 6)                          --- PASS
Test (11)   : Testing Procedural Programming (Part 7)                          --- PASS
Test (12)   : Testing Procedural Programming (Part 8)                          --- PASS
Test (13)   : Testing Object Oriented Programming (Part 1)                     --- PASS
Test (14)   : Testing Object Oriented Programming (Part 2)                     --- PASS
Test (15)   : Testing Object Oriented Programming (Part 3)                     --- PASS
Test (16)   : Testing Object Oriented Programming (Part 4)                     --- PASS
Test (17)   : Testing Object Oriented Programming (Part 5)                     --- PASS
Test (18)   : Testing Object Oriented Programming (Part 6)                     --- PASS
Test (19)   : Testing Object Oriented Programming (Part 7)                     --- PASS
Test (20)   : Testing Object Oriented Programming (Part 8)                     --- PASS
```

110.12 More Improvements

(1) Ring Compiler : Better support for (Operator Operator) to avoid checking the first operator when it's a literal.

(2) Ring Compiler : When we load a file that doesn't exist, display the caller file name in the error message.

(3) Ring Compiler : Support source code files with one line of comment without end of line.

(4) Ring Compiler : change nNoAssignment attribute in Parser Structure to nNewObject.

(5) Ring VM : Better support for the (Return) command inside braces that access new objects.

(6) Ring VM : Dir() Function - Don't add "." and ".." to the output

(7) Ring VM : Dir() Function - Correct output for the item type (file|directory) on Linux and macOS.

(8) Ring VM : ICO_LISTSTART - Clean pVM->aSetProperty when setting an object attribute.

(9) Ring VM : ICO_NEWOBJECT - Clean pVM->aSetProperty when setting an object attribute.

(10) Ring VM : Better code for Setter and Getter methods support.

(11) Ring2EXE: The libraries information are stored in separated files in ring/ring2exe/libs folder.

(12) WebLib : Separate the library to many source code files.

(13) StdLib : IsVowel() function - Better Code.

(14) RingQt : Count(), Left(), Mid() and Right() methods are added to QString class.

(15) Better Read Me File

CHAPTER

ONE

WHAT IS NEW IN RING 1.11?

In this chapter we will learn about the changes and new features in Ring 1.11 release.

111.1 List of changes and new features

Ring 1.11 comes with the next features!

- More 3D Samples
- Checkers Game
- Sokoban Game
- Maze Game
- Snake Game
- Sudoku Game
- Desktop Screen Shot Application
- Text To Speech Application
- RingRayLib Extension
- ZeroLib Library
- StdLib - More Functions
- Better RingQt
- Better Performance
- Better Documentation
- More Improvements

111.2 More 3D Samples

Ring 1.11 comes with more 3D samples based on Qt3D

- Folder : ring/samples/other/UsingQt3D (Contains 18 samples)

111.3 Checkers Game

It plays the - Must Jump - version of Checkers, The International Rules

It handles various invalid moves, invalid jumps, and must jumps.

The squares are colored to reflect errors.

The squares are colored to show source and destination of the move or jump.

111.4 Sokoban Game

A quick implementation for the Sokoban Game

Developed using Ring Game Engine for 2D Games in 2 hours (Less than 300 lines of code)

111.5 Maze Game

A quick implementation for the Maze Game

Developed using Ring Game Engine for 2D Games (Around 100 lines of code)

Also the game comes with a level designer (Developed in 10 minutes, 37 Lines of code)

111.6 Snake Game

A quick implementation for the Snake Game

Developed using Ring Game Engine for 2D Games (Around 200 lines of code)

111.7 Sudoku Game

Fill a 9 x 9 grid with digits so that each column, each row, and each of the nine 3 x 3 subgrids that compose the grid contain all of the digits from 1 to 9.

Filled: 40 Left: 41
Hint: 9
P001
Solve
New Game
Left-Mouse Button
Right-Mouse Button Select Nbr
1
2
3
4
5
6
7
8
9
0
Black Original number
Blue User solved number

111.8 Desktop Screen Shot Application

111.9 Text To Speech Application

111.10 RingRayLib Extension

Ring 1.11 comes with new extension to support the RayLib game programming library

Example:

```
load "raylib.ring"

screenWidth     = 800
screenHeight    = 450

InitWindow(screenWidth, screenHeight, "raylib [core] example - basic window")

SetTargetFPS(60)

while !WindowShouldClose()
        BeginDrawing()
        ClearBackground(RED)
        DrawText("Congrats! You created your first window!", 190, 200, 20, WHITE)
        EndDrawing()
end

CloseWindow()
```

Output:

Example:

```
load "raylib.ring"
```

(continues on next page)

(continued from previous page)

```
screenWidth = 800
screenHeight = 450

InitWindow(screenWidth, screenHeight, "raylib [shapes] example - basic shapes drawing")

SetTargetFPS(60)

while !WindowShouldClose()

        BeginDrawing()

        ClearBackground(RAYWHITE)

        DrawText("some basic shapes available on raylib", 20, 20, 20, DARKGRAY)
        DrawCircle(screenWidth/4, 120, 35, DARKBLUE)
        DrawRectangle(screenWidth/4*2 - 60, 100, 120, 60, RED)
        DrawRectangleLines(screenWidth/4*2 - 40, 320, 80, 60, ORANGE)
        DrawRectangleGradientH(screenWidth/4*2 - 90, 170, 180, 130, MAROON, GOLD)

        DrawTriangle(Vector2(screenWidth/4*3, 80),
                         Vector2(screenWidth/4*3 - 60, 150),
                         Vector2(screenWidth/4*3 + 60, 150), VIOLET)

        DrawPoly(Vector2(screenWidth/4*3, 320), 6, 80, 0, BROWN)

        DrawCircleGradient(screenWidth/4, 220, 60, GREEN, SKYBLUE)

        DrawLine(18, 42, screenWidth - 18, 42, BLACK)
        DrawCircleLines(screenWidth/4, 340, 80, DARKBLUE)
        DrawTriangleLines(Vector2(screenWidth/4*3, 160),
                          Vector2(screenWidth/4*3 - 20, 230),
                          Vector2(screenWidth/4*3 + 20, 230), DARKBLUE)
        EndDrawing()

end

CloseWindow()
```

Output:

111.11 ZeroLib Library

Ring 1.11 comes with the ZeroLib library that contains classes for Lists and Strings where the index starts from 0.

Example:

```
load "zerolib.ring"

? "Using List - Index start from 0"
List = Z( [1,2,3] )
List.Add(4)
List.Add(5)
? List[0]
? List[1]
? List[2]
? List[3]
? List[4]
nIndex = List.find(2)
? "Find(2) = " + nIndex
List.delete(0)
? "After deleting the first item : List[0]"
? "Now List[0] = " + List[0]

? "Using String - Index start from 0"
String = Z( "Welcome" )
? String[0]
? String[1]
? String[2]
```

(continues on next page)

(continued from previous page)

```
? String[3]
? String[4]
? String[5]
? String[6]
```

Output:

```
Using List - Index start from 0
1
2
3
4
5
Find(2) = 1
After deleting the first item : List[0]
Now List[0] = 2
Using String - Index start from 0
W
e
l
c
o
m
e
```

111.12 StdLib - More Functions

The next functions are added to the StdLib

- IsListContainsItems(aParent,aChild)
- IsBetween(nNumber,nMin,nMax)
- TimeInfo(cInformation)

Example:

```
load "stdlibcore.ring"
? "Using the IsListContainsItems() function"
aList1 = "a":"z"
aList2 = [:h,:l,:p,:u]
? IsListContainsItems(aList1,aList2)
? "Using the IsBetween() function"
? isBetween(1,3,4)
? isBetween(4,1,6)
? "Using the TimeInfo() function"
? timeInfo(:date)
? timeInfo(:year)
? timeInfo(:time)
? timeInfo(:hour_12)
```

Output:

```
Using the IsListContainsItems() function
1
Using the IsBetween() function
0
1
Using the TimeInfo() function
05/24/19
2019
15:30:33
03
```

For more information about these functions, see the StdLib functions chapter.

111.13 Better RingQt

- Building RingQt using Qt 5.12.3
- RingQt for Android - project file is updated to include webview module when we have it
- QString Class - New Methods
  - compare()
  - contains()
  - indexOf()
  - insert()
  - isRightToLeft()
  - remove()
  - repeated()
  - replace()
  - startsWith()
  - endsWith()
  - toHtmlEscaped()
  - clear()
  - isnull()
  - resize()
  - fill()
- QAxBase & QVariant - Better API
- The next classes are added to RingQt
  - QQuickView Class
  - QPrintDialog Class
  - QAxWidget2 Class
  - QTextToSpeech Class
  - QGraphicsView Class

- QAbstractAspect Class
- QNode Class
- QEntity Class
- QTransform Class
- QAspectEngine Class
- QTorusMesh Class
- QConeMesh Class
- QCylinderMesh Class
- QCuboidMesh Class
- QPlaneMesh Class
- QSphereMesh Class
- QPhongMaterial Class
- QForwardRenderer Class
- Qt3DWindow Class
- QAbstractCameraController Class
- QFirstPersonCameraController Class
- QTextureMaterial Class
- QExtrudedTextMesh Class
- QText2DEntity Class
- QSkyBoxEntity Class
- QConeGeometry Class
- QOrbitCameraController Class
- QDiffuseSpecularMaterial Class
- QGoochMaterial Class
- QMetalroughMaterial Class
- MorphPhongMaterial Class
- QPervertexColorMaterial Class
- QInputAspect Class
- QFrameAction Class
- QLogicAspect Class
- QCamera Class
- QCameraLens Class
- QMesh Class
- QTechnique Class
- QMaterial Class
- QEffect Class

  - QRenderPass Class
  - QSceneLoader Class
  - QPointLight Class
  - QRenderAspect Class
  - QTextureLoader Class
  - QObjectPicker Class
  - QCameraSelector Class
  - QCullFace Class
  - QDepthTest Class
  - QViewPort Class

111.14 Better Performance

Ring 1.11 is faster than Ring 1.10

The performance gain is between 10% and 30% based on the application.

111.15 Better Documentation

The next chapters are added to the documentation

- Tutorial : Ring Extensions in C/C++
- Using Qt3D
- Using ZeroLib
- Using RingRayLib

111.16 More Improvements

- **New Samples**
  - samples/other/ModuloTimesTableCircle folder
  - samples/other/saveimage folder
  - samples/other/UsingQML folder
  - samples/other/myguicontrol.ring
  - samples/other/qcalendarwidget.ring
  - samples/other/qcalendarwidget2.ring
  - samples/other/sudoku-KL02.ring
  - samples/other/sudoku-KL02-longproblem.ring
  - samples/other/zerobasedlist.ring
  - ringlibs/gameengine/lesson17.ring (Using Buttons)

  - samples/other/SQLTutorial/SQL-Tutorial.ring
  - samples/other/DrawFourier/AA-Draw-Fourier.ring
  - samples/other/SmartPhoneEmulator/ejemploKey.ring
  - samples/other/DiscreteFourierTransform/DiscreteFourierTransform.ring
  - samples/other/phonedatabase/PhoneDatabase.ring
- Gold Magic 800 - More levels (44 Levels)
- Fifteen Puzzle Game 3D - Better Code (Animation Speed)
- Flappy Bird 3000 - change the Bird direction to be looking down when we have game over
- Ring Notepad - Keyboard shortcuts for the Dockable Windows Mode
- Ring Notepad - When displaying functions list, don't avoid functions that contains the "_" character
- Ring Notepad - Output Window - Send Data - Better Code
- Ring Notepad - Find and Replace Window - Better Code
- Ring Notepad - Edit Menu - Insert Text Window
- Ring Notepad - Edit Menu - Lower Case and Upper Case options
- Ring Notepad - Edit Menu - Capitalize option
- Ring Notepad - Edit Menu - Comment Lines and Comment Block of lines
- Ring Notepad - File Tabs - Context Menu (Close other files, Close Active file and Close All files)
- RingPM - When updating a package - don't reinstall the dependency again
- Ring Game Engine for 2D Games - Added : GE_FULLSCREEN, GE_SCREEN_W and GE_SCREEN_H
- Ring Game Engine for 2D Games - Added : name property to game objects
- Ring Game Engine for 2D Games - Added : find() method to game class (find an object by name)
- Ring Game Engine for 2D Games - Support oGame[:ObjectName] to access an object
- Natural Library - Better Performance
- FoxRing - Added: frCTOD() function
- Code generator for extensions - generate functions that use managed pointers to new structures
- Ring VM - Error codes for Ring Object File errors
- Ring VM - Eval() function - Better Code (Better Performance)
- Ring VM - State Management - Better Code
- Ring VM - the ">" operator and operator overloading - Better Code
- Ring VM - Assignment and calling object methods - Better Code
- Ring VM - OOP - Getter Methods - Better Performance
- Ring API - Supporting RING_API_ISLIST() in C Extensions
- Ring Compiler - Supporting new lines after numbers and literals when writing conditions
- Ring Compiler - Supporting semi-colon (;) in the start of the line
- Ring Compiler - Prevent using Loop and Exit commands from outside loops

CHAPTER

TWO

WHAT IS NEW IN RING 1.12?

In this chapter we will learn about the changes and new features in Ring 1.12 release.

112.1 List of changes and new features

Ring 1.12 comes with the next features!

- Go Game
- ASCII Table application
- BMI Calculator application
- Calendar application
- Julian Day Calendar application
- Tutorial: Number to Words
- Load Again Command
- ring_state_filetokens() function
- Embedded Ring Object File
- Better RingRayLib
- More Improvements

112.2 Go Game

An implementation for the Go Game

112.3 ASCII Table application

Simple application for displaying the ASCII table

112.4 BMI Calculator application

Simple application for calculating the BMI

112.5 Calendar application

The Calendar for Year 2020

112.6 Julian Day Calendar application

The Julian Day Calendar application

112.7 Tutorial: Number to Words

Folder : ring/samples/other/number2words

112.8 Load Again Command

Ring 1.12 comes with the Load Again command

Using this command we can load the Ring source file which contains constants more than one time.

This is useful when using Ring source files for translations through global constants.

Example:

The next function is part from a project which support Arabic and English languages

The files english.ring and arabic.ring contains constants for translation

One of these files is loaded in the start of the program

Loading the same file again using the (Load) command is not possible

Because the (Load) command load the same source file only for the first time and ignore next times.

So we have to use the (Load Again) command.

Where we can use these files again during the runtime as in the next code

```
func setLang nLanguage
        if C_ENV_DEFAULT_LANG = nLanguage
                return
        ok
        C_ENV_DEFAULT_LANG = nLanguage
        # Change the language
                switch nLanguage
                        on C_TRANSLATION_ENGLISH
                                load again "translation/english.ring"
                        on C_TRANSLATION_ARABIC
                                load again "translation/arabic.ring"
                off
```

112.9 ring_state_filetokens() function

Using the ring_state_filetokens() function we can get all the tokens in the ring source code file.

```
C_FILENAME      = "test_tokens.ring"
C_WIDTH         = 12

# write the file
        write(C_FILENAME,'
                        see "Hello, World!"
                        ? 3*2+3
                        Name = "Ring"
                        ? Name
')

# Token Type
        C_KEYWORD       = 0
        C_OPERATOR      = 1
        C_LITERAL       = 2
        C_NUMBER        = 3
        C_IDENTIFIER    = 4
        C_ENDLINE       = 5

# Keywords List
aKEYWORDS = ["IF","TO","OR","AND","NOT","FOR","NEW","FUNC",
"FROM","NEXT","LOAD","ELSE","SEE","WHILE","OK","CLASS","RETURN","BUT",
"END","GIVE","BYE","EXIT","TRY","CATCH","DONE","SWITCH","ON","OTHER","OFF",
"IN","LOOP","PACKAGE","IMPORT","PRIVATE","STEP","DO","AGAIN","CALL","ELSEIF",
"PUT","GET","CASE","DEF","ENDFUNC","ENDCLASS","ENDPACKAGE",
"CHANGERINGKEYWORD","CHANGERINGOPERATOR","LOADSYNTAX"]

pState = ring_state_new()
aList = ring_state_filetokens(pState,C_FILENAME)
PrintTokens(aList)
ring_state_delete(pState)

func PrintTokens aList
        for aToken in aList
```

(continues on next page)

(continued from previous page)

```
                switch aToken[1]
                on C_KEYWORD
                        ? Width("Keyword",C_WIDTH) + ": "  + aKeywords[0+aToken[2]]
                on C_OPERATOR
                        ? Width("Operator",C_WIDTH)  + ": " + aToken[2]
                on C_LITERAL
                        ? Width("Literal",C_WIDTH)  + ": " + aToken[2]
                on C_NUMBER
                        ? Width("Number",C_WIDTH)  + ": " + aToken[2]
                on C_IDENTIFIER
                        ? Width("Identifier",C_WIDTH)  + ": " + aToken[2]
                on C_ENDLINE
                        ? "EndLine"
                off
        next

func Width cText,nWidth
        return cText+copy(" ",nWidth-len(cText))
```

Output:

```
EndLine
Keyword     : SEE
Literal     : Hello, World!
EndLine
Operator    : ?
Number      : 3
Operator    : *
Number      : 2
Operator    : +
Number      : 3
EndLine
Identifier  : name
Operator    : =
Literal     : Ring
EndLine
Operator    : ?
Identifier  : name
EndLine
```

112.10 Generate Embedded Ring Object File

We can generate embedded object file (C source code) from the source code file (*.ring) using -geo option

Command:

```
ring test.ring -geo
```

This command will generate at least three files

```
test.c
ringappcode.c
ringappcode.h
```

More files could be generated based on the project size

The generated files will pass the byte code to Ring VM to be executed

112.11 Better RingRayLib

More Samples are added to RingRayLib

- Sound Loading Playing
- Texture Source
- Music Playing Streaming
- Rectangle scaling
- Colors Palette
- Following Eyes
- Collision Area
- Bezier Lines
- Images Generation
- Fifteen Puzzle Game
- Cubic Map

Screen Shot:

112.12 More Improvements

- **New Samples**
  - ring/samples/other/Hex2UTF8.ring
  - ring/samples/other/CalmoSoftPrimesTable.ring
  - ring/samples/other/CalmoSoftTicTacToeGame.ring
  - ring/samples/other/CalmoSoftSimpleGoGame.ring
  - ring/samples/other/arabicmysql.ring
  - ring/samples/other/CalmoSoftExtraCube.ring
  - ring/samples/other/DynamicCode/anonfunc.ring
  - ring/samples/other/DynamicCode/deletethisfile.ring
  - ring/samples/other/DynamicCode/modifythisfile.ring
  - ring/samples/other/changesyntax/ArabicDemo.ring
  - ring/samples/other/changesyntax/EnglishDemo.ring
  - ring/samples/other/changesyntax/ChangeKeywordsArabic.ring
  - ring/samples/other/changesyntax/ChangeKeywordsEnglish.ring
  - ring/samples/other/changesyntax/pascal.ring
  - ring/samples/other/hijridate.ring
- Ring Notepad - Project Files - set minimum width based on desktop screen width

- Ring Notepad - Output Window - Move the Cursor to the end of text
- Ring Notepad - Output Window - Correct displaying for line breaks
- Form Designer - Better Style - Controls colors and size
- VideoMusicPlayer is updated to work as expected after RingQt update
- FlappyBird3000 - Fast response on Android
- Snake Game : Change the default window size (800x600)
- Maze Game : Change the default window size (800x600)
- Maze Game : Move the camera with the player
- Maze Game : Restarting the game will hide the (You Win) message
- Game Engine : display error message when we can't create the game window
- Ring Tests : Added File build.sh for building on Linux and macOS
- RingQt : Updated to Qt 5.12.6
- RingQt : Added QQMLEngine class
- RingQt : Added files for building RingQt without Bluetooth support
- RingQt : The size of the events code is changed from 100 characters to 200 characters
- RingQt : Correct links for Qt documentation in RingQt classes chapter
- RingQt for Android : Better code for executing the ring object file (ringo)
- Ring2EXE configuration files are updated for RingQt to correctly distribute RingQt apps
- Code Generator : Convert function names to lower case when generating the functions for structures
- OSCopyFolder() function is updated to copy the files in sub folders too
- fgetpos() function is updated to work as expected
- IsFunction() function is updated to be not case sensitive
- Space() function is updated to clear the output string with spaces
- Ring Compiler : Added file buildclang.bat for building on Windows using Clang compiler
- Ring VM - Internal hash function is updated
- Ring VM - Better Code for setting pVM->aSetProperty when creating new objects
- Ring VM - Better Code for state management

CHAPTER

THREE

WHAT IS NEW IN RING 1.13?

In this chapter we will learn about the changes and new features in Ring 1.13 release.

113.1 List of changes and new features

Ring 1.13 comes with the next features!

- New Book by Apress
- Ring For WebAssembly
- Better Threads Support
- Laser Game
- Magic Balls Game
- MoneyBoxes Game
- Matching Game
- Pairs Game
- Word Game
- Tetris Game
- Escape Game
- Hassouna Course Samples
- Ring support in SpaceVim
- Better RingQt
- Better RingRayLib
- RingStbImage Extension
- More Low Level Functions
- Better Organization
- More Improvements

113.2 New Book by Apress

Apress: Beginning Ring Programming (From Novice to Professional)

URL: https://www.apress.com/gp/book/9781484258323

Gain a gentle introduction to the world of Ring programming with clarity as a first concern using a lot of practical examples.

What You Will Learn

- Get started with Ring and master its data types, I/O, functions, and classes
- Carry out structural, object-oriented, functional, declarative, natural, and meta programming in Ring
- Use the full power of Ring to refactor program code and develop clean program architectures
- Quickly design professional-grade video games on top of the Ring game engine

113.3 Ring for WebAssembly

Ring support distributing applications for WebAssembly (Using Qt for WebAssembly)

- Hello World : https://ring-lang.github.io/web/helloworld/project.html
- Matching Game : https://ring-lang.github.io/web/matching/project.html
- Pairs Game : https://ring-lang.github.io/web/pairs/project.html
- Othello Game : https://ring-lang.github.io/web/othello/project.html
- Game of Life : https://ring-lang.github.io/web/gameoflife/project.html
- Online Form Designer : https://ring-lang.github.io/web/formdesigner/project.html

113.4 Better Threads Support

Ring 1.13 provides better support for threads

(1) Ring VM is updated to use a memory pool for each thread (Faster)

(2) More tests for RingAllegro threads functions

(3) RingLibUV is updated to be thread safe when using callback functions

113.5 Laser Game

An implementation for the Laser Game

113.6 Magic Balls Game

An implementation for the Magic Balls Game

113.7 Money Boxes Game

See if you can collect the required amount of money by opening boxes!

113.8 Matching Game

An implementation for the Matching Game

113.9 Pairs Game

An implementation for the Pairs Game

113.10 Word Game

An implementation for the Word Game

113.11 Tetris Game

An implementation for the Tetris Game

Features

- New Shapes are selected in random way (Different Shape, Color & Position)
- You can move and rotate the shapes, increase the speed.
- Score: You get 10 points when completing a row (The game check for nested rows completion as expected)
- The game is designed to work forever without user interaction (After Game Over, the Game restarts automatically)

113.12 Escape Game

Escape from the Snake and collect the Stars to prevent it from growing!

113.13 Hassouna Course Samples

URL (YouTube): https://www.youtube.com/watch?v=6VIHMyrEilw&list=PLHIfW1KZRIfl6KzfLziFl650MmThnQ0jT

The course samples are added to ring/samples/UsingArabic/HassounaCourse folder

The Rockets sample

113.14 Ring support in SpaceVim

URL: https://github.com/SpaceVim/SpaceVim

Screen Shot:

113.15 Better RingQt

(1) The next classes are added to RingQt

- QAbstractAxis
- QAbstractBarSeries
- QAbstractSeries
- QAreaLegendMarker
- QAreaSeries
- QBarCategoryAxis
- QBarLegendMarker
- QBarSeries
- QBarSet
- QBoxPlotLegendMarker
- QBoxPlotSeries
- QBoxSet
- QCandlestickLegendMarker
- QCandlestickModelMapper
- QCandlestickSeries
- QCandlestickSet
- QCategoryAxis
- QChart
- QChartView
- QDateTimeAxis
- QHBarModelMapper
- QHBoxPlotModelMapper
- QHCandlestickModelMapper
- QHPieModelMapper
- QHXYModelMapper
- QHorizontalBarSeries
- QHorizontalPercentBarSeries
- QHorizontalStackedBarSeries
- QLegend
- QLegendMarker.
- QLineSeries
- QLogValueAxis
- QPercentBarSeries

- QPieLegendMarker
- QPieSeries
- QPieSlice
- QPolarChart
- QScatterSeries
- QSplineSeries
- QStackedBarSeries
- QVBarModelMapper
- QVBoxPlotModelMapper
- QVCandleStickModelMapper
- QVPieModelMapper
- QVXYModelMapper
- QValueAxis
- QXYLegendMarker
- QXYSeries
- QGraphicsScene
- QMovie

(2) Better QtConverter application

This application is used for preparing Qt classes for the Ring Code Generator

Then using the Ring Code Generator we generate RingQt classes

The application user interface is updated for better productivity!

(3) RingQt for Android - Qt Project - Special folders for Ring and RingQt

(4) Using Qt 5.15.0

113.16 Better RingRayLib

(1) All functions return objects instead of pointers

(2) Support RayMath libary functions

(3) More Samples are ported from RayLib to RingRayLib

- Scissor Test
- Input Gestures Detection
- Bouncing Ball
- Rectangle Rounded
- Draw Circle Sector
- RayLib Logo Animation
- First Person Maze

- Plane Rotations

113.17 RingStbImage Extension

New extension support the stb_image library

Example:

```
# Load the library
        load "stbimage.ring"
# Image Information
        width=0 height=0 channels=0
# Ring will Free cData automatically in the end of the program
        cData = stbi_load("ring.jpg",:width,:height,:channels,STBI_rgb)
# Display the output
        ? "Size (bytes): " + len(cData)
        ? "Width : " + width
        ? "Height: " + height
        ? "Channels: " + channels
```

Output:

```
Size (bytes): 557371
Width : 563
Height: 330
Channels: 3
```

113.18 More Low Level Functions

The next functions are added to the Low Level functions

For more information see the Low Level Functions chapter in the documentation

```
setpointer(pointer,nNewAddress)
getpointer(pointer) ---> nAddress
pointer2string(pointer,nStart,nCount) ---> cString
memcpy(pDestinationPointer,cSourceString,nSize)
```

113.19 Better Organization

We have better organization for the project folders and source code files

(1) A folder called (Language) contains the source code and the visual source of the Compiler and the Virtual Machine

(2) The (Extensions) folder (Bindings for C/C++ libraries) - contains also the (libdepwin, android & webassembly folders)

(3) The (Libraries) folder - contains Ring libraries written in Ring itself, contains now (GuiLib & ObjectsLib) too

(4) The (Tools) folder - contains development tools - contains now (Editors, RingNotepad, FormDesigner, etc)

(5) The (Samples) folder - contains Ring samples - a lot of organization is done in this folder

113.20 More Improvements

- **More Applications**
  - Towers of Hanoi
  - Questions Game
  - Money Case Game
  - Rock Paper Scissors Game
  - Lottery Game
  - Nim Game
  - Eight Queens Game
  - Typing Quiz
  - Pong 2 Game
  - Space Shooter Game
- **More Samples**
  - ring/samples/Language/HelloWorld folder
  - ring/samples/Language/ChangeIsNULL folder
  - ring/samples/Language/Console/customsee.ring
  - ring/samples/Language/Numbers/decimalscount.ring

  - ring/samples/Language/EvalInScope/swap.ring
  - ring/samples/Language/EvalInScope/global.ring
  - ring/samples/Language/EvalInScope/enum.ring
  - ring/samples/Language/Endian/endian.ring
  - ring/samples/Language/VariablePointer/varptr4.ring
  - ring/samples/Language/VariablePointer/varptr5.ring
  - ring/samples/Language/DebugCode folder
  - ring/samples/Language/ClassMethods folder
  - ring/samples/Language/Lists/CheckHashTableAttribute.ring
  - ring/samples/Language/RingFileTokens folder (Ring Compiler - Scanner Output)
  - ring/samples/ProblemSolving/Fib folder
  - ring/samples/ProblemSolving/ArrayPathDest/solveArrayPathDest.ring
  - ring/samples/ProblemSolving/PegSolitaire/peg-soli.ring
  - ring/samples/General/RosettaCode/uniquecharacters.ring
  - ring/samples/General/RosettaCode/similarcharacters.ring
  - ring/samples/General/RandomLatinSquares folder
  - ring/samples/General/FactorialRecursion folder
  - ring/samples/UsingArabic/RightToLeft folder (Set Layout Direction)
  - ring/samples/UsingWebLib/Unicode folder
  - ring/samples/UsingQt/InputMask folder
  - ring/samples/UsingQt/PlayGif folder
  - ring/samples/UsingQt/TableWidget folder
  - ring/samples/UsingQt/ButtonSizeInLayout folder
  - ring/samples/UsingQt/DateTimeEditFormat folder (Date Picker Control)
  - ring/samples/UsingQML/sample10 folder (Charts Samples)
  - ring/samples/UsingQML/sample11 folder (Data Visualization Samples)
  - ring/samples/UsingQtWASM/colordialog folder
  - ring/samples/UsingQtWASM/fontdialog folder
  - ring/samples/UsingQtWASM/filedialog folder
  - ring/samples/UsingQtWASM/filecontent folder (Download/Upload Files)
  - ring/samples/UsingRayLib/more/ex4_levelsofcubes.ring
  - ring/samples/UsingOpenGL/cubeongpu/cubeongpu.ring
  - ring/samples/UsingOpenGL/cubeongpu2/cubeongpu.ring
- Ring Notepad - Output Window - set the buffer size to 1 MB
- Ring Notepad - View Menu - Source Code (Full Screen)
- Ring Notepad - Keyboard shortcuts for different styles

- Ring Notepad - Support saving files in folders contains the dot character
- Ring Notepad - Browser Window - Set colors based on the current style
- Ring Notepad - Functions List - Display functions defined using "def"
- Ring Notepad - Distribute Menu - Distribute for Web Browser using WebAssembly (RingQt)
- Form Designer - ToolBox - Larger width in Windows style
- Form Designer - ToolBar - New icon for the (Select Objects) button
- Gold Magic 800 - Level Editor - Decrease the window size
- Super Man 2016 - Increase speed of (Game Over) message animation
- Super Man 2016 - Better code for collision detection between SuperMan and Walls
- Game Engine - Game Class - Icon property (set the window icon)
- Set the window icon for many games developed using the Ring game engine
- Type Hints library - Better Code
- StdLib - IsMainSourceFile() function - Better Code
- StdLib - TimeInfo() function - All TimeList() information are now available
- StdLib - Map() & Filter() functions - Support accessing the global scope
- StdLib - NewList() function is no longer required - Use the List() function
- WebLib - Template() function - Support accessing the global scope
- Objects Library - Better API
- Natural Library - Better Code
- RingLibSDL - Update LibSDL version from 2.0.10 to 2.0.12
- RingOpenGL - Better Code (Added GLEW functions)
- RingFreeGLUT - Better Code (Added many functions)
- RingLibUV - Better Code - Samples that uses the VarPtr() function
- RingPM - Support terminals that pass the executable name using UPPER case
- Ring2EXE - Distribute for Web Browser using WebAssembly (RingQt)
- Ring2EXE - Delete the executable file if we have it in the target folder
- Ring2EXE - Always copy files listed in the resources file to the target folder
- Ring2EXE - Distribute for Android - Copy Ring and RingQt folders
- Ring Tests - Display report summary after running all of the tests
- CodeGen - Add Option: PassNullBeforeVMPointer (For C++ Classes)
- CodeGen - Using RING_API_ISCPOINTER() instead of RING_API_ISPOINTER()
- Ring Compiler - ChangeRingKeyword - Support comments and many commands in the same line
- Ring Compiler - ChangeRingOperator - Support comments and many commands in the same line
- Ring Compiler - ring_parser.c - Better Code
- Ring Compiler - ring_stmt.c - Better Code
- Ring Compiler - ring_expr.c - Better Code

- Ring Compiler - ring_state.c - Flag for the (Not Case Sensitive) feature
- Ring Compiler - Load Command - Support loading libraries from ring/bin/load folder
- Ring Compiler - LoadSyntax Command - Support loading libraries from ring/bin/load folder
- Ring Compiler - Command: ? <expr> - Clear error message when the expression is missing
- Ring Compiler - Better Error Messages
- Ring VM - Using lists during definition - Support using the list itself (not only items)
- Ring VM - List() Function - Support List(nRow,nCols) to create 2D lists
- Ring VM - List() Function - Better Performance
- Ring VM - Object File - Save/Restore the files list
- Ring VM - ring_vmexpr.c - Better Code - Avoid magic numbers
- Ring VM - ring_state_filetokens() - Optional parameter for the (Not Case Sensitive) feature.
- Ring VM - ring_state_setvar() - Better Code
- Ring VM - int2bytes(), float2bytes() & double2bytes() uses sizeof() function
- Ring VM - fclose() function - Display error message if the FILE pointer is NULL
- Ring VM - Extensions API - Support local scope of the caller when getting integer pointer

CHAPTER

FOUR

WHAT IS NEW IN RING 1.14?

In this chapter we will learn about the changes and new features in Ring 1.14 release.

114.1 List of changes and new features

Ring 1.14 comes with the next features!

- MyCoffee (Web Application)
- Web Development Samples
- More Samples
- Erlang B Formula
- Customers Form
- RingTilengine Extension
- RingLibui Extension
- RingSockets Extension
- RingThreads Extension
- Better RingOpenSSL
- More Functions
- Better Functions
- Better Performance For Strings
- Better Handling For Numbers
- Using CLOC (Count Lines of Code)
- More Improvements

114.2 MyCoffee (Web Application)

A web application that uses the WebLib library and PostgreSQL database

Screen Shots:

114.3 Web Development Samples

Starting from lesson 301, the Hassouna Course provides lessons about web development.

These lessons uses the WebLib library for Back-end web development.

Special YouTube playlist for these lessons:

- https://www.youtube.com/watch?v=a3ZYjssAvbI&list=PLHIfW1KZRIfn1cs2BupsdJ7dzCQ7wmHFk

The samples are added to this folder:

- https://github.com/ring-lang/ring/tree/master/samples/UsingArabic/HassounaCourse/WebDevelopment

Screen Shot:

- Calculator Sample

114.4 More Samples

The next samples are added to the samples folder

- ring/samples/UsingQML/sample12 folder
- ring/samples/General/TimeServer folder
- ring/samples/General/BinarokuGame folder
- ring/samples/Language/EnumGenerator folder
- ring/samples/Language/DynamicObject folder
- ring/samples/Drawing/MatrixMultiply3DRotationCube

Screen Shots:

Matrix Multiplication 3D Cube Rotation
DrawMatrix
ShapePI
ROTATON
X: 1.0
Y: 1.0
Z: 1.0
SCALE
X: 1.0
Y: 1.0
Z: 1.0
TRANSLATE
X: 0.0
Y: 0.0
Z: 0.0
SHEAR
X: 0.0
Y: 0.0
Z: 0.0
REFLECTION
X: 1
Y: 1
Z: 1
SPEED
Delay: 0.00
Inc: 0.05
TimeCycle:
MSec: 403
Count 55

114.5 Erlang B Formula

An application that uses the Erlang B Formula (Circuit Switching Systems)

	A	B	J	K	Q
1	20	0.00	0.07	19.93	0.62
2	21	0.01	0.13	20.87	0.65
3	22	0.01	0.22	21.78	0.68
4	23	0.02	0.33	22.67	0.71
5	24	0.02	0.49	23.51	0.73
6	25	0.03	0.68	24.32	0.76
7	26	0.04	0.91	25.09	0.78
8	27	0.05	1.17	25.83	0.81
9	28	0.07	1.46	26.54	0.83
10	29	0.08	1.78	27.22	0.85
11	30	0.10	2.12	27.88	0.87
12	31	0.11	2.47	28.53	0.89
13	32	0.13	2.83	29.17	0.91
14	33	0.15	3.20	29.80	0.93
15	34	0.16	3.56	30.44	0.95
16	35	0.18	3.93	31.07	0.97
17	36	0.20	4.29	31.71	0.99
18	37	0.21	4.65	32.35	1.01
19	38	0.23	5.00	33.00	1.03
20	39	0.24	5.35	33.65	1.05
21	40	0.26	5.68	34.32	1.07

114.6 Customers Form

An application that uses SQLite and TableWidget

The Use Interface is provided in Arabic, English & French

114.7 RingTilengine Extension

This extension provides support for Tilengine - The 2D retro graphics engine with raster effects

Example:

```
load "tilengine.ring"

TLN_Init(400, 240, 1, 0, 0)
TLN_SetLoadPath("assets\sonic")
foreground = TLN_LoadTilemap ("Sonic_md_fg1.tmx", NULL)
TLN_SetLayerTilemap(0, foreground)

TLN_CreateWindow(NULL, 0)
while TLN_ProcessWindow()
        TLN_DrawFrame(0)
end
```

(continues on next page)

(continued from previous page)

```
TLN_DeleteTilemap(foreground)
TLN_Deinit()
```

Screen Shots:

114.8 RingLibui Extension

This extension provides complete support for Libui

Using this extension we can develop and distribute lightweight GUI Applications using Ring (Less than 1 MB)

Runtime files and their size

- Ring.dll (448 KB)
- Libui.dll (210 KB)
- Ring_Libui.dll (633 KB)
- Total : 1,291 KB without compressing the files
- After compressing the files (To ZIP file) - Total : 504 KB

Example:

```
load "libui.ring"

oWindow = uiNewWindow( "Say Hello", 500, 80, True)
uiWindowOnClosing(oWindow,"closeApp()")

lbl1 = uiNewLabel("Name: ")
text1 = uiNewEntry()

btn1 = uiNewButton("SayHello")
uiButtonOnClicked(btn1,"sayHello()")

btn2 = uiNewButton("Close")
uiButtonOnClicked(btn2,"closeApp()")

lbl2 = uiNewLabel("")

g = uiNewGrid() uiGridSetPadded(g, 1) uiWindowSetChild(oWindow, g)
uiGridAppend(g, lbl1, 0, 0, 2, 1, 1, uiAlignCenter, 0, uiAlignCenter)
uiGridAppend(g, text1, 1, 0, 2, 1, 1, uiAlignFill, 0, uiAlignFill)
uiGridAppend(g, btn1, 0, 1, 1, 2, 1, uiAlignFill, 0, uiAlignFill)
uiGridAppend(g, btn2, 2, 1, 1, 1, 1, uiAlignFill, 0, uiAlignFill)
uiGridAppend(g, lbl2, 0, 3, 2, 1, 1, uiAlignCenter, 0, uiAlignCenter)

uiControlShow( oWindow )
uiMain()

func sayHello
        uiLabelSetText(lbl2,"Hello " + uiEntryText(text1))

func closeApp
        uiQuit()
```

Screen Shots:

libui Control Gallery
Basic Controls
Numbers and Lists
Data Choosers
Button
Checkbox
This is a label. Right now, labels can only span one line.
Entries
Entry
Password Entry
Search Entry
Multiline Entry
Multiline Entry No Wrap

Histogram Sample
48
57
34
95
5
42
27
95
36
65

libui Text-Drawing Example
Arial Regular 16
Alignment
Left
Drawing strings with libui is done with the uiAttributedString and uiDrawTextLayout objects. uiAttributedString lets you have a variety of attributes: font family, font size, **font weight**, *font italicness*, font stretch, text color, text background color, underline style, and underline color. Furthermore, there are attributes allowing for special underlines for indicating spelling errors (and other types of errors) and control over OpenType features such as ligatures (for instance, afford vs. afford).\nUse the controls opposite to the text to control properties of the text.

Drawing Sample
Close Application

114.9 RingSockets Extension

In Ring, We have sockets using different extensions like RingQt, RingLibuv and RingLibSDL

In this release we provide a special extension for sockets

This will be useful if your application doesn't use the previous libraries

Example (Server Code)

```
# TCP SERVER

load "sockets.ring"

sock = socket(AF_INET,SOCK_STREAM,0)
bind(sock,"127.0.0.1",5050)
listen(sock,5)
ns = accept(sock)
send(ns,"Hello Client")
msg = recv(ns,1024)
? "Client Say >> " + msg
close(sock)
? "Socket connection closed"
```

Example (Client Code)

```
# TCP Client

load "sockets.ring"

sock = socket(AF_INET,SOCK_STREAM)
connect(sock,"127.0.0.1",5050)

send(sock,"Hello Server")
msg = recv(sock,1024)
? "Server Say >> " + msg

close(sock)
? "Socket connection closed"
```

114.10 RingThreads Extension

In Ring, We have threads using different extensions like RingQt, RingLibuv and RingAllegro

In this release we provide a special extension for threads

This will be useful if your application doesn't use the previous libraries

Example:

```
load "threads.ring"

func main
```

(continues on next page)

(continued from previous page)

```
        nThreads = 2
        aList = list(nThreads)

        for x=1 to nThreads
                aList[x] = new_thrd_t()
                thrd_create(aList[x],"Hello("+x+")")
        next

        for x=1 to nThreads
                res= 0
                thrd_join(aList[x],:res)
        next

        ? :Done
        shutdown()

func Hello x

        for r=1 to 100
                ? "Message from the Hello("+x+") function"
        next
```

114.11 Better RingOpenSSL

The next functions are added to the RingOpenSSL extension

These functions compute the hash of large files/data without the need to load all of the content in a single string.

```
md5init() -> MD5_CTX
md5update (MD5_CTX, string) -> 1 for success or 0 for failure
md5final (MD5_CTX) -> string

sha1init() -> SHA_CTX
sha1update (SHA_CTX, string) -> 1 for success or 0 for failure
sha1final (SHA_CTX) -> string

sha224init() -> SHA224_CTX
sha224update (SHA224_CTX, string) -> 1 for success or 0 for failure
sha224final (SHA224_CTX) -> string

sha256init() -> SHA256_CTX
sha256update (SHA256_CTX, string) -> 1 for success or 0 for failure
sha256final (SHA256_CTX) -> string

sha384init() -> SHA384_CTX
sha384update (SHA384_CTX, string) -> 1 for success or 0 for failure
sha384final (SHA384_CTX) -> string

sha512init() -> SHA512_CTX
sha512update (SHA512_CTX, string) -> 1 for success or 0 for failure
sha512final (SHA512_CTX) -> string
```

114.12 More Functions

- DirExists() Function

```
DirExists(cDirPath) ---> returns 1 if the directory exists
```

- GetPathType() Function

```
GetPathType(cPath) ---> 0 if the path doesn't exists
                        1 if it corresponds to existing file
                        2 if it corresponds to existing directory
                        -1 if the path exists but has
                           an unknown type (e.g. a pipe)
```

- SysSet() Function

We can set environment variables using the SysSet() function

```
SysSet(cVariable, cValue) ---> Returns 1 for success and return 0 for failure
```

- SysUnset() Function

We can delete an environment variables using the SysUnset() function

```
SysUnset(cVariable) ---> Returns 1 for success and return 0 for failure
```

- GetArch() Function

We can detect the architecture of the Ring executable using the GetArch() function

Syntax:

```
GetArch() ---> cString (The name of the architecture of the Ring executable)
```

Example:

```
switch getarch()
on "x86"
        ? "x86 32bit architecture"
on "x64"
        ? "x64 64bit architecture"
on "arm64"
        ? "ARM64 64bit architecture"
on "arm"
        ? "ARM 32bit architecture"
other
        ? "Unknown architecture"
off
```

- NofProcessors() Function

We can detect the number of processors using the NofProcessors() Function

Syntax:

```
NofProcessors() ---> nProcessors
```

Example:

```
? NofProcessors()
```

114.13 Better Functions

- Log() Function

The functions is updated to support calculating the logarithm to any base

Syntax:

```
Log(nNumber)       --> logarithm of nNumber to the base of e
Log(nNumber,nBase) --> logarithm of nNumber to the base of nBase
```

Example:

```
? log(10)       # 2.30
? log(2)        # 0.69
? log(10,10)    # 1
? log(2,2)      # 1
? log(100,10)   # 2
```

114.14 Better Performance For Strings

- Ring 1.14 is 3x Faster in programs that have strings with long and fixed size

Example:

```
load "openssllib.ring"
f = fopen(exefilename(),"rb")
h = SHA256Init()
while true
  s = fread(f, 4096)
  if isstring(s)
    SHA256Update(h, s)
  else
    exit
  ok
end
digest = SHA256Final(h)
fclose(f)
? digest
```

Output:

```
4e677154639dae3baa048ce5ae0b04b63bcd33316e2d2041297dcee85604d778
```

- Ring 1.14 is 60x Faster when adding strings to other strings

Example:

```
t1 = clock()
test = "My Ring"
for x = 1 to 20000
        test += x
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"
```

Output:

```
Time : 0.01 seconds
```

- These improvements increased the performance of Ring Code Generator for Extensions (4x Faster)

114.15 Better Handling For Numbers

(1) Using Hexadecimal Values

Example:

```
x = 0x10
? x              # 16
x = 0xff
? x              # 255
x = 0x0A
? x              # 10
? 0xFFFF         # 65535
? 0x0A + 1       # 10+1 = 11
```

(2) Printing large double value

Example:

```
c = 999999999999999
for i = 1 to 13
        c *= 999999999999999
next
? "c = " + c
```

Output:

```
c = 9.
↪999999999999862032924046117813879019544782068185773946275755888189234614925384380788550958e+209
```

(3) Using String() and Number() with large double values

Example:

```
c1 = 999999999999999
for i = 1 to 13
        c1 *= 999999999999999
next

s = string(c1)  ? "c1 = " + s
```

(continues on next page)

(continued from previous page)

```
c2 = number(s)  ? "c2 = " + c2

? "c2 - c1 = " + (c2 - c1)

str1    = "-2222044646462"
c       = Number(str1)
str2    = String(c)

if str1 = str2
        ? "Strings Identical"
else
        ? "Strings Mismatch!"
ok
```

Output:

```
c1 = 9.
↪999999999998620329240461178138790195447820681857739462757558881892346149253843807885509 58e+209
c2 = 9.
↪999999999998620329240461178138790195447820681857739462757558881892346149253843807885509 58e+209
c2 - c1 = 0
Strings Identical
```

114.16 Using CLOC (Count Lines of Code)

Usage:

```
ringcloc <application_folder_path>
```

Example(1):

```
ringcloc c:\ring\tools\ringnotepad
```

Output:

```
      47 text files.
      47 unique files.
      11 files ignored.

github.com/AlDanial/cloc v 1.88  T=1.18 s (39.0 files/s, 3212.2 lines/s)
-------------------------------------------------------------------------------
Language                         files          blank        comment           code
-------------------------------------------------------------------------------
Ring                                33            273            139           3186
Bourne Shell                         4             22              9             88
DOS Batch                            4              3              7             20
INI                                  1              0              0             15
C                                    1              0              1             10
Markdown                             2              4              0              8
Windows Resource File                1              0              0              1
-------------------------------------------------------------------------------
```

(continues on next page)

(continued from previous page)

```
SUM:                                46            302            156           3328
-------------------------------------------------------------------------------
```

Example(2):

```
ringcloc c:\ring\tools\formdesigner
```

Output:

```
      54 text files.
      54 unique files.
      13 files ignored.

github.com/AlDanial/cloc v 1.88  T=1.54 s (34.5 files/s, 7122.4 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Ring                            52           1306            567           9071
Markdown                         1              5              0              7
-------------------------------------------------------------------------------
SUM:                            53           1311            567           9078
-------------------------------------------------------------------------------
```

114.17 More Improvements

- Ring Compiler - Pass new lines after the FUNC keyword in anonymous functions
- Ring Compiler - Pass new lines after the function parameters in anonymous functions
- Ring Compiler - Pass new lines after the access of the list items
- Ring Compiler - Pass new lines after the access of the object attribute
- Ring Compiler - Support numbers of 97 digits
- Ring Compiler - Support Statement –> Expression –> '(' ')'
- Ring Compiler - Support Statement –> Expression –> '(' Expression ')' '.' Attribute|Method
- Ring Compiler - Better error message when literal is not closed
- Ring VM - Str2Hex() function - Better Performance
- Ring VM - Hex2Str() function - Better Performance
- Ring VM - SubStr() function - Better Code
- Ring VM - Dec() function - Better Code
- Ring VM - String() function - Better Code
- Ring VM - Number() function - Better Code
- Ring VM - Decimals() function - Allows a range between 0 and 90
- Ring VM - Comparing between binary strings - Better Code
- Ring VM - IsWindows64() function - Better Code

- Ring VM - Unsigned() function - Better Code
- Ring VM - Function Call - Better Management for Lists & Nested Lists state
- Ring VM - AddAttribute() - The default type of the new attribute will be a STRING
- Ring VM - Dir() function - Better Code
- Ring VM - Copy() function - Better Code (Ignore Negative Numbers)
- Ring VM - Restore the step value when we use exit from more than one loop
- Ring VM - After (For-In Loop) - Delete the reference but keep a copy of the value
- Ring VM - Better Code when working with many threads
- Ring VM - Better Code when restoring the state after Try/Catch/Done
- Ring VM - Better Code when using Raise() in nested calls after Try/Catch/Done
- Ring VM - Better format when printing instructions during execution
- Ring VM - Support using the & Operator to return the String Character Reference
- Ring VM - Call command - function name - not case sensitive
- Ring Compiler/VM Source Code - Better Format
- Ring Compiler/VM Source Code - Code Refactoring
- Ring Compiler/VM - New instruction: ICO_FREETEMPLISTS
- RingPM - New Package: GUILib (Separate package for the GUILib classes)
- RingPM - New Package: ObjectsLib (Separate package for the ObjectsLib classes)
- RingPM - New Package: CLOC (Count Lines of Code)
- RingQt - QWebEngineView Class - Added: Print(QPrinter,cEventCode) Method
- RingQt - QCombobox Class - Added: setCurrentText() and addItems() methods
- RingQt - Added: QLocale Class
- RingQtRuntime Package: Using GUILib & ObjectsLib packages
- GameEngine - Add the delay function to the Graphics Library: gl_delay() function
- GameEngine - Using the Resources Class when loading the Window Icon
- RingAllegro - Better Code
- RingMurmurHash - Better Code
- RingOpenSSL - RandBytes() function - Better Code
- RingZip - GetFileNameByIndex() function - Better Code
- RingLibSDL - Building using LibSDL version 2.0.14

CHAPTER

FIVE

WHAT IS NEW IN RING 1.15?

In this chapter we will learn about the changes and new features in Ring 1.15 release.

115.1 List of changes and new features

Ring 1.15 comes with the next features!

- New Arabic Book
- Chess Endgame Application
- Chess 3D (Qt3D Sample)
- Better Ring Notepad
- Better StdLib
- Better RingQt
- Better RingLibCurl
- Heroku (Better support)
- Using ':' then Numbers
- More Samples
- More Improvements

115.2 New Arabic Book

We have a new printed book (Written in Arabic Language)

The Book is a about building a project using Ring to analysis the Arabic Poetry

115.3 Chess Endgame Application

Chess Endgame data set visualization

Also the application provides game result prediction using Microsoft Azure Machine Learning

Screen Shots:

	WhiteKingFile	WhiteKingRank	WhiteRookFile	WhiteRookRank	BlackKingFile	BlackKingRank	GameResult
1	a	1	b	3	c	2	draw
2	a	1	c	1	c	2	draw
3	a	1	c	1	d	1	draw
4	a	1	c	1	d	2	draw
5	a	1	c	2	c	1	draw
6	a	1	c	2	c	3	draw
7	a	1	c	2	d	1	draw
8	a	1	c	2	d	2	draw
9	a	1	c	2	d	3	draw
10	a	1	c	3	c	2	draw
11	a	1	c	3	d	2	draw
12	a	1	c	3	d	3	draw
13	a	1	c	3	d	4	draw
14	a	1	c	4	d	3	draw
15	a	1	d	1	c	1	draw
16	a	1	d	1	c	2	draw
17	a	1	d	1	d	2	draw

	Game Result	Count	Percentage
1	draw	2796	9.97%
2	zero	27	0.10%
3	one	78	0.28%
4	two	246	0.88%
5	three	81	0.29%
6	four	198	0.71%
7	five	471	1.68%
8	six	592	2.11%
9	seven	683	2.43%
10	eight	1433	5.11%
11	nine	1712	6.10%
12	ten	1985	7.07%
13	eleven	2854	10.17%
14	twelve	3597	12.82%
15	thirteen	4194	14.95%
16	fourteen	4553	16.23%
17	fifteen	2166	7.72%
18	sixteen	390	1.39%

115.4 Chess 3D (Qt3D Sample)

It's an example about using Qt3D

- We can move the board
- Select a Piece and move it
- Castle - King Rook side
- En-Passant capture
- Pawn promotion to Queen

To Install the sample using the Ring Package Manager

```
ringpm install Chess3D-Qt3D
ringpm run Chess3D-Qt3D
```

Screen Shots:

115.5 Better Ring Notepad

(1) Automatic generation for the Event Function

In the Form Designer, When we determine the name of the Event Function

Ring Notepad will generate the function code automatically

Source Code : C:/Users/samir/Desktop/test1Controller.ring

test1Controller.ring

```
# Form/Window Controller - Source Code File

load "test1View.ring"

import System.GUI

if IsMainSourceFile() {
        new App {
                StyleFusion()
                openWindow(:test1Controller)
                exec()
        }
}

class test1Controller from windowsControllerParent

        oView = new test1View

        func sayhello
                oView {

                }

```

Source Code : C:/Users/samir/Desktop/test1Controller.ring Form Designer

115.6 Better StdLib

(1) RandomList() function

Syntax:

```
RandomList(aList) --> List contains the same items using Random order
```

Example:

```
load "stdlib.ring"
aList = 1:5
? RandomList(aList)
```

(2) RandomItem() function

Pick an item from a list (Random Choice)

Syntax:

```
RandomItem(aList) --> Item
```

Example:

```
load "stdlib.ring"
aList = 1:5
? RandomItem(aList)
```

(3) List2Code() function

- Support double quotation in list strings
- Better Format - Add tabs before the items

(4) StdLib - Split() function - Support using an empty string as delimiter

(5) StdLib - StdFile Class - FTell() function - Better Code

115.7 Better RingQt

- QApp Class - Clipboard() method
- QString2 Class - More Methods
- QMessageBox Class - setIcon() Method
- QByteArray Class - Append() Method - Support Binary Data
- QByteArray Class - Replace() Method - Multiple versions and better code
- QTabWidget Class - TabBar() Method
- QVector3D Class - Project() and UnProject() Methods
- QPainter Class - DrawHSVFList() and DrawRGBFList() Methods
- RingQt For WebAssembly - Download File - Support Binary File
- QCompleter4 Class
- Event Code size changed from 99 characters to 255 characters
- Clear Error Message if the Event Code is larger than the allowed size

Example (Using Clipboard):

```
load "guilib.ring"

oApp = new qapp {

        w = new qWidget() {
                setWindowTitle("Using Clipboard - Press CTRL+V")
                resize(400,100)
                new qlineedit(w) {
                        move(10,10)
                        resize(200,30)
                }
                show()
        }

        oApp.clipboard().setText("I Love Ring Programming!",0)

        exec()
}
```

Screen Shot:

Example (Using Strings):

```
load "guilib.ring"

o1 = new QString2()
o1.append("Ring")

oChar = new QChar(61)
? o1.leftJustified(20,oChar,False)
? o1.rightJustified(20,oChar,False)
```

Output:

```
Ring================
================Ring
```

115.8 Better RingLibCurl

- curl_easy_setopt() support passing (CURLLIST *) as the third parameter

Example (From ChessEndGame Application)

URL: https://github.com/ring-lang/ring/blob/master/applications/chessendgame/predictionController.ring#L67

```
# Header
mylist = curl_slist_append(null,'accept-encoding: identity')
mylist = curl_slist_append(mylist,'authorization: Bearer ' + my_api_key )
mylist = curl_slist_append(mylist,'connection: close')
mylist = curl_slist_append(mylist,'content-length: ' + len(my_body) )
mylist = curl_slist_append(mylist,'content-type: application/json')
curl_easy_setopt(curl, CURLOPT_HTTPHEADER, mylist)
```

115.9 Heroku (Better support)

From Ring 1.8 We can run Ring Web Applications in the Cloud using Heroku

In Ring 1.15 We updated the packages to use a modern Ring version

Also to avoid all of the reported problems during installation

URL: https://github.com/ring-lang/RingWebAppOnHeroku

115.10 Using ':' then Numbers

Ring 1.15 support using ':' then Numbers to define literals

Example:

```
? Type( :1234 )

aList = [ :1 = "One",
          :2 = "Two"]

? aList[:1]
? aList[:2]
```

Output:

```
STRING
One
Two
```

115.11 More Samples

The next samples are added to the samples folder

- samples/Language/AnonFunctions/anonfunc1.ring
- samples/Language/AnonFunctions/anonfunc2.ring
- samples/Language/AnonFunctions/anonfunc3.ring
- samples/Language/AnonFunctions/anonfunc4.ring
- samples/Language/AnonFunctions/anonfunclib.ring
- samples/Language/Numbers/powfunc.ring
- samples/Language/Strings/trim.ring
- samples/Language/Strings/trim2.ring
- samples/Language/Loops/loopcommand.ring
- samples/General/NumberTripletsGame/NumberTripletsGame.ring
- samples/General/Elevator/elevator.ring
- samples/General/SmallExamples/ParametersOrder/parametersorder.ring
- samples/UsingArabic/WriteArabicFile/using_QFile_write.ring
- samples/UsingArabic/WriteArabicFile/using_QFile_read.ring
- samples/UsingRayLib/more/ex6_jump_player2D.ring
- samples/UsingRayLib/more/ex7_jump_player3D.ring
- samples/UsingRayLib/more/ex8_Aiming_at_3D_moving_targets.ring
- samples/UsingRayLib/more/ex9_jump_player3D_collision_detection.ring
- samples/UsingRayLib/more/ex10_jump_player3D_long_way.ring

- samples/UsingQt/String/QCharUnicodeValue.ring
- samples/UsingQt/String/QCharUnicodeValue2.ring
- samples/UsingQt/String/QStringJustified.ring
- samples/UsingQt/Clipboard/QTextEditClipboard.ring
- samples/UsingQt/Clipboard/UsingQClipboard.ring
- samples/UsingQt/Time/timemillisecond.ring
- samples/UsingQt/ByteArray/Base64.ring
- samples/UsingWebLib/PathInfo/pathinfo.ring
- samples/Drawing/ColorWheel/ColorWheel-Animate.ring
- samples/Drawing/ColorWheel/ColorWheel-FastDraw.ring

Screen Shot:

115.12 More Improvements

- Form Designer - Better Translation
- Customers Application - New translation file (Portuguese Language)
- Code Generator for Extensions - Function prototype - Trim spaces in the output type
- Code Generator for Extensions - Return objects of types that have classes without init
- Code Generator for Extensions - Correct parameter number when freeing unassigned pointers

- RingLibUV - Better Code (When using threads)
- Ring VM - Better Code (When terminating threads)
- Ring VM - Support converting NULL to Zero when we compare between Number and NULL
- Ring VM - ring_state_stringtokens() function
- Ring VM - ring_state_scannererror() function
- Ring VM - Clean error message when we have infinite recursions
- Ring VM - TempName() Function - Better Code!
- Ring VM - SubStr() Function - Better Code!
- Ring VM - Random() & Randomize() Functions - Better Code!
- Ring VM - ICO_FREETEMPLISTS Instruction - Better Implementation
- Ring VM - ICO_NEWOBJ/ICO_SETSCOPE - State Management - Better Code!
- Ring VM - RING_API_RETLISTBYREF - Better Code!
- Ring VM - Respect Decimals() function when printing numbers inside Lists & Objects
- Ring VM - Convert the String to a Number when using (=) or (!=) operators
- Ring VM - ICO_INCPJUMPSTEP1 and ICO_JUMPVARPLENUMSTEP1 - Better Code
- Ring VM - ring_vm_newfunc() function - Better Code
- Ring Compiler - For-In Loop - Always access string characters by reference
- Ring Compiler - For-In Loop - The iterator value after the loop is processed
- Ring Compiler - Do-Again Loop - Avoid unnecessary JUMP instruction when using (Loop) command
- Ring Compiler - When Embedding Ring in Ring - Always keep Ring keywords (Not Case Sensitive)
- Ring Compiler - Scanner - Option (Comments as Tokens)
- Ring Compiler - Hexadecimal Numbers - Support many zeros in the start of the number
- Ring Compiler - Support Setter/Getter when using (++) and (–) operators
- Ring Compiler - Display Warning (unrecognized option) when we pass wrong options to the Compiler

CHAPTER

SIX

WHAT IS NEW IN RING 1.16?

In this chapter we will learn about the changes and new features in Ring 1.16 release.

116.1 List of changes and new features

Ring 1.16 comes with the next features!

- Light GUILib
- UTF-8 File Names in Microsoft Windows
- Nested Methods Call in Separate Lines
- Code Runner Extension support Ring
- Zero and Strings
- Better Installation Scripts
- Better Documentation
- MDI Windows Sample
- More Improvements

116.2 Light GUILib

A lot of RingQt applications uses only QtCore, QtGui & QtWidget modules

These applications could use

```
load "lightguilib.ring"
```

Instead of

```
load "guilib.ring"
```

Also Ring Notepad, Distribute menu comes with a new option : Distribute light RingQt application

Using this option we can distribute lightguilib applications

For example, Distributing (Game of Life) using this option provides : target/windows folder

- Size : 35 MB (Uncompressed)
- Size : 13 MB (zip)
- Size : 9 MB (exe) compressed using 7zip

So we can distribute these GUI applications using an installer less than 10 MB

Tip: if you need something smaller than that (1 MB) then switch to other libraries like LibUI

116.3 UTF-8 File Names in Microsoft Windows

In Ring 1.16, The Load command support using UTF-8 in the file name.

For example, We can write Arabic letters in the File Name!

Also the next functions support this feature

```
read()
write()
fopen()
```

116.4 Nested Methods Call in Separate Lines

In Ring 1.16, the Compiler support adding new lines after the method call and before the dot operator

Example:

```
myobj = new Start
myobj.one()
     .two()
     .three()
     .go()

class start
        func one
                ? "One"
                return new one


class one
        func two
                ? "Two"
                return new two

class two
        func three
                ? "Three"
                return new three

class three
        func go
                ? "Go!"
```

Output:

```
One
Two
Three
Go!
```

116.5 Code Runner Extension support Ring

If you are using Microsoft Visual Studio Code, We have good news for you!

The Code Runner Extension added support for the Ring programming language

After installing Code Runner

It's recommended to modify this file :

C:/Users/YOURUSERNAME/.vscode/extensions/formulahendry.code-runner-0.11.6/package.json

Set the property (code-runner.fileDirectoryAsCwd) to (True)

So Code Runner can move to the file directory when we run it using (Ctrl+Alt+N)

```
"code-runner.fileDirectoryAsCwd": {
        "type": "boolean",
        "default": true,
        "description": "Whether to use the directory of the file to be executed as the␣
↪working directory.",
        "scope": "resource"
},
```

Tip: Check ring/tools/editors/vscode folder to support Ring in VSCode

116.6 Zero and Strings

From Ring 1.0, the language do implicit conversion between numbers and strings

This is useful when we mix them in some situations like printing something on the screen

```
x = 10          # Number
? "x = " + x    # x converted from Number to String
```

Also we can do arithmetic operations

```
x = "10"        # String
? 5 + x         # x converted from String to Number
```

The question is What happens if x content is not a number?

The answer : The result of the conversion will be (Zero)

```
x = "Test"      # String - The content is not a number
? 5 + x         # print (5) - x converted from String to Number (Zero)
```

The other operators like "=" and "!=" do the conversion too

Starting from Ring 1.16, They will be careful when we compare things to Zero

Example:

```
x = "Test"
? 0 = x         # The result will be FALSE
? 0 != x        # The result will be TRUE
```

This is useful when we compare between values inside Empty Lists and Strings

```
aList = list(10)        # 10 items - Each item is Zero
? aList[1] + 5          # print (5)
? aList[1] = "Test"     # False
? aList[1] = 0          # True
```

The other values (Not Zero) will follow the normal conversion rules

```
x = "5"
? 5  = x        # True
? 6 != x        # True
```

116.7 Better Installation Scripts

Ring 1.16 comes with better installation scripts on Linux and macOS

- install.sh - Force creation of symlinks in case they already exist
- install.sh - Remove quarantine flag
- uninstall.sh - Delete symlinks from /usr/local accordingly

116.8 Better Documentation

All of the documentation chapters are revised and improved

116.9 MDI Windows Sample

The next sample is added to the samples folder

- samples/UsingQt/MDIWindows/mdi_windows.ring

116.10 More Improvements

- Sample: samples/AQuickStart/GUILib/gui1.ring - Better Code
- Form Designer - Set the Button Event (If it's Empty) from the Button Text
- Form Designer - Order the controls based on the position (Not the Creation Order)
- Form Designer - File System - File Name Encoding
- Form Designer - Properties Window - Property Name Column - Better Colors

- StdLib - IsPrime() Function - Better Code
- RingQt - QListWidget Class - AddItem() Method - Convert Number to String
- RingQt - More Qt constants are defined
- Ring2EXE - Display usage information
- Ring2EXE - Support Creating folders when copying files
- Ring VM - ring_vm_error() - Better Code
- Ring Compiler - Don't display (Unrecognized Option) if we have a source code file

CHAPTER

SEVEN

WHAT IS NEW IN RING 1.17

In this chapter we will learn about the changes and new features in Ring 1.17 release.

117.1 List of changes and new features

Ring 1.17 comes with the next features!

- Merge Two Files
- Poetry Analysis
- Citations Prediction
- TokensLib Library
- CSVLib Library
- JSONLib Library
- HTTPLib Library
- Better GUILib
- Better RingOpenSSL
- Better RingWinAPI
- Better Syntax Flexibility
- Better I/O functions
- Better Ring API
- Better Code Generator for Extensions
- More Improvements

117.2 Merge Two Files

An application to merge two text files line by line

Development Time: 30 minutes (Includes user interface design, coding, testing and distribution)

117.3 Poetry Analysis

An application to analysis the Arabic poetry

The application is over 4000 lines of Ring code

The source code is explained in a printed book about the subject

117.4 Citations Prediction

An applicaiton to predict the research paper citations using title, abstract or authors.

The applications uses Microsoft Azure Machine Learning for the Models

The Models are trained using a Dataset for a specific domain and specific journal

This application comes with 3 samples to see how it works!

It's an example about using Ring, RingQt, RingLibCurl and JSONLib

117.5 TokensLib Library

Using this library we can get Ring tokens from source code files or strings.

Example:

```
load "tokenslib.ring"

func main
        oTokens = new RingTokens {
                fromFile("hello.ring")
                PrintTokens()
                ? Copy("=",50)
                fromString("? 1+1")
                PrintTokens()
        }
```

Output:

```
Keyword     : SEE
Literal     : Hello, World!
EndLine
==================================================
Operator    : ? (23)
Number      : 1
Operator    : + (1)
```

(continues on next page)

(continued from previous page)

```
Number      : 1
EndLine
```

117.6 CSVLib Library

Using this library we can generate and read CSV files

Functions:

```
List2CSV(aList) --> cCSVString
CSV2List(cCSVString) --> aList
```

Example:

```
load "csvlib.ring"

aList = [ ["number", "square" ] ]

for t=1 to 10
        aList + [ t, t*t ]
next

write( "squares.csv", list2CSV(aList) )
```

Output:

Example (2)

```
load "csvlib.ring"

if ! fexists("squares.csv")
        ? "The file squares.csv doesn't exist! - Run writeSquaresTable.ring to create it"
        return
ok

aList = CSV2List( read("squares.csv") )

for subList in aList
        ? "" + subList[1] + " - " + subList[2]
next
```

Output:

```
number - square
1 - 1
2 - 4
3 - 9
4 - 16
```

(continues on next page)

(continued from previous page)

```
5 - 25
6 - 36
7 - 49
8 - 64
9 - 81
10 - 100
```

117.7 JSONLib Library

Using this library we can generate and read JSON files

Functions:

```
List2JSON(aList) --> cJSONString
JSON2List(cJSONString) --> aList
```

Example (1):

File: sample.json

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 20,
  "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
  },
  "phoneNumbers": [
        { "type": "home", "number": "212 555-1234" },
        { "type": "fax", "number": "646 555-4567" }
  ]
}
```

Ring Code:

```
load "jsonlib.ring"

func main

        aList = JSON2List( read("sample.json") )

        ? aList[:FirstName]
        ? aList[:LastName]
        ? aList[:Age]
        ? aList[:Address][:city]
        ? aList[:phoneNumbers][1][:Type]
        ? aList[:phoneNumbers][1][:Number]
        ? aList[:phoneNumbers][2][:Type]
        ? aList[:phoneNumbers][2][:Number]
```

Output:

```
John
Smith
20
New York
home
212 555-1234
fax
646 555-4567
```

Example (2):

```
load "jsonlib.ring"

func main

        aList = [
                :name = "Ring",
                :year = 2016
        ]

        ? List2JSON(aList)
```

Output:

```
{
        "name": "Ring",
        "year": 2016
}
```

JSONLib uses RingCJSON to read JSON files at high-performance

This RingCJSON extension support the CJSON library

CJSON URL: https://github.com/DaveGamble/cJSON

117.8 HTTPLib Library

This library provides support for HTTP based on cpp-httplib

URL: https://github.com/yhirose/cpp-httplib

Example:

```
load "httplib.ring"

oServer = new Server {

        ? "Try localhost:8080/hi"
        route(:Get,"/hi",:mytest)

        ? "Listen to port 8080"
        listen("0.0.0.0", 8080)
```

(continues on next page)

(continued from previous page)

```
}

func mytest
        oServer.setContent("Hello World!", "text/plain")
```

The other examples exist in this folder: ring/samples/UsingHTTPLib

See the (Using HTTPLib) chapter for more information.

117.9 Better GUILib

(1) Many deprecated methods are removed from RingQt

(2) Added: AddList() method to the TableWidget class

Tip: TableWidget class is a subclass of QTableWidget class

Note: To use TableWidget class, import system.gui after loading guilib.ring or lightguilib.ring

Example:

Source code: https://github.com/ring-lang/ring/tree/master/samples/UsingQt/TableWidget/AddRingList

```
class addRingListController from windowsControllerParent

        oView = new addRingListView

        aList = [["one","two"],
                 ["three","four"],
                 ["five","six"],
                 [7,8],
                 ["I","Love","Ring","Programming"]]

        oView.tablewidget1.addList(aList)

        aList = [["Number","Square"]]
        for t = 1 to 10
                aList + [ t, t*t]
        next

        oView.tablewidget1.addList(aList)
```

Screen Shot:

(3) Added: toList() method to the tableWidget class

Example:

```
aList = oView.tablewidget1.toList()
? aList
```

(4) Added: AddList() method to the ListWidget class

Tip: ListWidget class is a subclass of QListWidget class

Note: To use ListWidget class, import system.gui after loading guilib.ring or lightguilib.ring

Example:

```
aList = [ 1:10, "one", "two", "three"]
oView.listWidget1.addList(aList)
```

(5) Added: toList() method to the ListWidget class

Example:

```
aList = oView.listwidget1.toList()
? aList
```

117.10 Better RingOpenSSL

- Enhancements to support all cipher algorithms
- New Functions:

```
rsa_generate(nBits[,nPublicExponent]) ---> a random RSA key pair
rsa_export_params(pRsaKey) ---> list of the key parameters
rsa_import_params(pParamsList) ---> a new RSA key
rsa_export_pem(pRsaKey) ---> string encoding of the key in PEM format
rsa_import_pem(cStrPEM) ---> a new RSA key
rsa_is_privatekey(pRsaKey) ---> Bool
rsa_encrypt_pkcs(pRsaKey,cPlainData) ---> String
rsa_decrypt_pkcs(pRsaKey,cEncryptedData) ---> String
rsa_encrypt_oaep(pRsaKey,cPlainData[,nHashAlgorithm]) ---> String
rsa_decrypt_oaep(pRsaKey,cEncryptedData[,nHashAlgorithm]) ---> String
rsa_encrypt_raw(pRsaKey,cPlainData) ---> String
rsa_decrypt_raw(pRsaKey,cEncryptedData) ---> String
rsa_sign_pkcs(pRsaKey,cData) ---> String
rsa_signhash_pkcs(pRsaKey,cHashValue) ---> String
rsa_verify_pkcs(pRsaKey,cData,cSignature) ---> Bool
rsa_verifyhash_pkcs(pRsaKey,cHashValue,cSignature) ---> Bool
rsa_sign_pss(pRsaKey,cData,nHashAlgorithm[,nSaltLength]) ---> String
rsa_signhash_pss(pRsaKey,cHashValue[,nSaltLength]) ---> String
rsa_verify_pss(pRsaKey,cData,cSignature,nHashAlgorithm[,nSaltLength]) ---> Bool
rsa_verifyhash_pss(pRsaKey,cHashValue,cSignature[,nSaltLength]) ---> Bool
openssl_versiontext() ---> String
openssl_version() ---> List
```

117.11 Better RingWinAPI

- Better Code
- New functions for resources, temp files and uuid

```
rGetTempFileName(LPCSTR lpPathName, LPCSTR lpPrefixString, UINT   uUnique)
rwaCreateUUID() ---> a string containing the value of the generated UUID
rwaReadBinaryResource(modulePath, resourceName, resourceType) ---> String (Bytes)
```

117.12 Better Syntax Flexibility

The next keywords are added to the language

- endif
- endfor
- endwhile
- endswitch
- endtry
- function
- endfunction
- break
- continue

Example:

```
for t=1 to 10
        ? t
        if t=3
                ? :three
        endif
endfor
```

117.13 Better I/O Functions

The next functions can be used without the need to load stdlib.ring

- print(cString)
- print2str(cString) -> cString
- puts(cString)
- getstring() -> cString
- getnumber() -> nNumber

They are written in the C language to provide better performance

Example:

```
puts("Hello, World!")
puts(2022)
puts("one\ntwo\nthree")
puts("one\n\ttwo\n\t\tthree")
age = 6
puts("Ring is #{age} years old!")
puts("I know that 2+2=#{2+2} and 3+3=#{3+3}")
happy()

func happy
```

(continues on next page)

(continued from previous page)

```
        new myclass {x=10 y=20 z=30 test()}
        ? :done

class myclass
        name = "Ring"
        x y z
        func test
                puts("Language Name = #{name}")
                puts("x=#{x}\ny=#{y}\nz=#{z}")
```

Output:

```
Hello, World!
2022
one
two
three
one
                two
                                three
Ring is 6 years old!
I know that 2+2=4 and 3+3=6
Language Name = Ring
x=10
y=20
z=30
done
```

117.14 Better Ring API

The next functions are added to Ring API

- RING_LIBINIT
- RING_API_REGISTER(cFunctionName,pFunction)
- RING_API_MALLOC(nSize)
- RING_API_CALLOC(nItems,nItemSize)
- RING_API_REALLOC(pPointer,nSize)
- RING_API_FREE(pPointer)
- RING_API_FREEFUNC

The next example demonstrates using RING_LIBINIT and RING_API_REGISTER

```
#include "ring.h"
#include "stdlib.h"

RING_FUNC(ring_myfunction)
{
        printf("Hello, World!");
```

(continues on next page)

(continued from previous page)

```
}

RING_LIBINIT
{
        RING_API_REGISTER("myfunction",ring_myfunction);
}
```

117.15 Better Code Generator for Extensions

The code generator is updated to include the next features

- Better output messages
- Avoid generating the delete method twice
- Using codegenlib.ring for common functions
- Use casting in functions that destroy the structures
- Clean error message when we can't parse a line
- Support C/C++ strings in structures

Example from RingHTTPLib extension

```
<struct>
MultipartFormData {string name,string content,string filename,string content_type}
</struct>
```

117.16 More Improvements

- Ring Notepad (Web Browser) - Support URLs without "http" in the start
- Form Designer - Default TextColor is Empty (Better when changing styles)
- Form Designer - Generating Code - Form Height - Better Code
- Many documents and samples are revised (Better English)
- Documentation - Language Reference - Functions - Parameters & Output
- Many applications are updated to use LightGUILib
- Calendar application - Better code to detect the current year
- JulianDayCalendar application - Better code to detect the current year
- 2048 Game - Added window icon
- Added: samples/Drawing/Mandelbrot/Mandelbrot_Fast.ring
- Added: samples/AQuickStart/MySQL/mysql10.ring
- Added: samples/UsingQt/Process/processoutput.ring
- Added: samples/UsingObjectsLib/test1.ring
- Added: samples/UsingObjectsLib/test2.ring

- Added: samples/UsingObjectsLib/test3.ring
- StdLib - AppPath() function - Better Code
- StdLib - New Function: IsAppCompiled()
- StdLib - New Function: AppArguments()
- StdLib - New Function: CheckEquality()
- ObjectsLib - New Function: GetWindowByID()
- RingLibSDL - LibSDL version is updated from 2.0.14 to 2.0.20
- RingAllegro - Allegro version is updated from 5.2.4 to 5.2.7.1
- RingLibCurl - Better support for Ring2EXE
- RingLibUI - The function uiFreeText() is removed from the library API
- RingLibUI - Better implementation for functions that require uiFreeText()
- RingSockets - Better Code
- RingWinCReg - Better Code
- RingQt - QString Class - Another version for the Replace() Method
- RingQt - QLabel Class inherits the QFrame Class
- RingQt - Added: QStyle class
- RingQt - QApp Class - Added more methods
- Ring Object File - ring_objfile_writeCfile() - Better Code
- Eval() function - Return NULL if the Return command is not used
- SetPointer()/GetPointer() functions - Better Code
- Fexists() function - Support UTF-8 file names
- GetFileSize() function - Get the file size without requiring any I/O
- ringvm_codelist() function - Return the Byte Code of the current program
- Ring VM - Better code when copying lists
- Ring VM - ICO_JUMPZ Instruction - Treat empty lists as Zero
- Ring VM - ICO_JUMPZ Instruction - Treat empty strings (NULL) as Zero
- Ring VM - The result of using the Not (!) operator then List will be False
- Ring VM - ring_vm_catch() function - Better Code
- Ring VM - ring_vm_showerrormessage() function - Better Code
- Ring VM - State Management - Better Code - ring_vm_newobjectstackpointer()
- Ring VM - State management - Better Performance (when creating new objects)
- Ring VM - Adding binary strings to the Stack - Better Code
- Ring VM - ring_vm_loadindexaddress() function - Better Code
- Ring VM - ring_vm_range() function - Better Code
- Ring VM - ring_vm_string_assignment() function - Better Code
- Ring VM - ring_vm_notequal() function - Better Code

- Ring VM - ring_item_getnumber() - support different types (int/double)
- Ring Compiler - Better Code - Using ring_general_addosfileseparator()
- Ring Compiler - Better Code - Using ring_general_folderexistinfilename()
- Ring Compiler - ring_scanner_checktoken() - Faster & Better Code
- Ring Source Code - Better Format
- Ring Source Code - Check out of memory at the Callee side instead of the Caller
- Update: language/src/locatevc.bat - Support VS 2022

CHAPTER

EIGHT

WHAT IS NEW IN RING 1.18

In this chapter we will learn about the changes and new features in Ring 1.18 release.

118.1 List of changes and new features

Ring 1.18 comes with the next features!

- Read Files application
- Random User application
- Better CitPre application
- Better Tools
- More Samples
- Syntax Files
- The Power Operator
- Using References
- Ring for MS-DOS
- Faster (For-In) Loop
- Faster BraceError()
- Better RingRayLib
- Better RingSockets
- Heroku (Better support)
- Command: New From
- ImportPackage() function
- More Low Level functions
- Better WebLib
- Better Ring API
- Better Documentation
- MatrixLib Library
- More Improvements

118.2 Read Files application

Using this application we can display text/binary files using Hexadecimal

118.3 Random User application

A simple application as an example about using GUILib, InternetLib & JSONLib

Random User Data
Samuel Johnston
Email: samuel.johnston@example.com
New User Data
Close Application

Random User Data
Jennifer Fuller
Email: jennifer.fuller@example.com
New User Data
Close Application

118.4 Better CitPre application

In Ring 1.18 the Citations Prediction application is updated

(1) The application comes with 500 samples

(2) Journal Paper (HTML): https://www.hindawi.com/journals/bmri/2022/2239152/

(3) Journal Paper (PDF): https://downloads.hindawi.com/journals/bmri/2022/2239152.pdf

DataSet Rows

	Title	Authors	Abstract	Total Citations
49	Blamey, Peter Artieres, ...	Factors Affecting Auditor...	Objective: To update a 15-year-old study of 800 ...	189
50	Snik, AFM Mylanus, EAM ...	Consensus statements...	After more than 25 years of clinical experience, the BAHA...	190
51	Nadol, JB Shiao, JY Burgess,...	Histopathology of ...	The insertion of an intrascalar electrode array during cochle...	190
52	Papsin, Blake C	Cochlear implantation i...	OBJECTIVES/HYPOTHESIS: To evaluate outcomes after ...	181
53	Pisoni, DB Cleary, M	Measures of working ...	Large individual differences in spoken word recognition ...	183
54	Stakhovskaya, Olga Sridhar, ...	Frequency map for the ...	The goals of this study were to derive a frequency-position ...	186
55	Gstoettner, W Kiefer, J ...	Hearing preservation in ...	Objective-To evaluate the possibility of preservation of low-...	178
56	Robbins, AM Koch, DB ...	Effect of age at cochlear...	Objectives: To investigate the effect of age at cochlear ...	173
57	Geers, Ann E. Moog, Jean S. ...	Spoken Language Scor...	This study investigated three questions: Is it realistic to expe...	181
58	Kirk, KI Miyamoto, RT Lento, ...	Effects of age at ...	This study examined the effects of age at implantation on the...	183
59	Brown, CJ Hughes, ML Luk, ...	The relationship betwe...	Objective: The objective of this study was to determine the ...	166
60	Kral, Andrej Eggermont, Jos J.	What's to lose and what...	Sensory and environmental manipulations affect the ...	175

Select Close

118.5 Better Tools

- Ring2EXE: Better code when distributing GUI applications
- Form Designer: Correct drawing for new controls after moving Ring Notepad
- Ring Notepad: Output Window - Correct cursor position while using unicode
- Ring Notepad: Set the default font when opening the Font dialog
- Ring Notepad: Preserve editing history after text insertion
- Ring Notepad: New button to clear the output and stop the running program
- Ring Notepad: Goto Line - Support files contains UTF-8 characters
- Ring Notepad: Find/Replace - Support files contains UTF-8 characters
- Ring Notepad: Avoid hiding the cursor when pressing TAB
- Ring Notepad: Reset the dockable window title when closing all opened files
- Ring Notepad: Translation files (English/Arabic)
- Ring Notepad: New option to build GUI application using Ring2EXE
- Ring Notepad: View menu - Navigate options

- Ring Notepad: View menu - Apperance options (Checkboxes)
- Ring Notepad: Check if we have web application before running
- Ring Notepad: Better icon for closing the application
- Ring Notepad: Select file window (Quick navigation between opened files)

118.6 More Samples

- Added: samples/General/OrbitalMechanics
- Added: samples/UsingBingChat (Code generated by AI)
- Added: samples/General/QuizMarks/QuizMarks.ring
- Added: samples/UsingQt/TabActivate/simpleExample.ring
- Added: samples/UsingQt/TabActivate/TabActivateController.ring
- Added: samples/UsingFormDesigner/centerwindow
- Added: samples/AQuickStart/MySQL/mysql11.ring
- Added: samples/UsingQt3D/ex19_sceneKeyboard.ring
- Added: samples/Algorithms/path_finding_bfs_usingref.ring
- Added: samples/General/SmallExamples/Random/random100.ring
- Added: samples/General/SmallExamples/Conjecture/knuthconjecture.ring
- Added: samples/General/SmallExamples/SendMoreMoney/SendMoreMoneyMonteCarlo.ring
- Added: samples/General/SmallExamples/MatrixMulti/Matrix-Func.ring
- Added: samples/UsingFreeGLUT/test17.ring, test18.ring & test19.ring
- Added: samples/ProblemSolving/Cryptogram/Cryptogram.ring

- Added: samples/UsingArabic/ArabicSyntax
- Added: samples/UsingQt/Painter/test.ring & test2.ring
- Added: samples/UsingJSONLib - From test5.ring to test9.ring
- File: samples/AQuickStart/GUILib/gui44.ring - Better Code
- File: samples/UsingWebLib/Demo/datalib.ring - Better Code
- File: samples/UsingWebLib/Demo/bootstrap.ring - Better Code
- File: samples/UsingStbImage/test2.ring - Better Code
- Folder: samples/UsingLibSDL - Better Code

	-	Planet	Mass	Distance	L1	L2	L3	L4-L5-x	L4-L5-y
1	-	Sun	1.9891^30	0	1	2	3	4	5
2	-	Earth	5.9736^24	149597871	1481012	1510945	149597933	74798486	-129555557
3	-	Moon	7.3476^22	384400	3806	3882	384400	192199	-332900
4	-	Mercury	3.3000^23	57894376	576738	581150	57894377	28947178	-50138000
5	-	Venus	4.8700^24	108159261	1071484	1091701	108159298	54079366	-93668668
6	-	Mars	6.4170^23	227987155	2269029	2290714	227987165	113993504	-197442668
7	-	Ceres	9.1000^20	413937309	4137161	4141585	413937309	206968654	-358480225

118.7 Syntax Files

Ring 1.18 provide another step towards better syntax flexibility

- Support running source code files with any extension
- Automatic loading for (ringsyntax.ring) file that exist in the current folder

For example in the the next screen shot

We have (ringsyntax.ring) that translate some of the Ring keywords to Arabic language

When we execute the file with Arabic name which means in English (hello.ring)

Ring will automatically execute (ringsyntax.ring) using Load Syntax command

Each Folder in the program could have it's optional (ringsyntax.ring) file

We can mix styles in the same project

118.8 The Power Operator

Ring 1.18 Support (**) and (^^) as the power operator

Example:

```
? 3**4                  # 81.00
? 3^^4                  # 81.00
? 256^^(1/4)            # 4
? 3.4^^4.5              # 246.41
? 256^^(1/3)            # 6.35
? 129.6^^(1/5)          # 2.65
? 5.9736*(10^^6)        # 5973600.00
? 7.3476*(10^^(-5))     # 0.0000734760
```

118.9 Using References

Ring as a language is designed to reduce references usage

The assignment operator (=) copy lists/objects by value

This release comes with the next functions to create and use managed references

```
ref(aList|oObject)       ---> aList|oObject (Reference) # Short name
reference(aList|oObject) ---> aList|oObject (Reference) # Long name
refcount(variable) ---> Number (References Count)
```

Example:

```
aList = [ 10,20,30, ref(aList) ]        # Circular Reference
? aList[4][1]                           # Print 10
```

(continues on next page)

(continued from previous page)

```
? aList[4][4][4][4][4][2]                # Print 20
? refcount(aList)                        # Print 2
```

We added this feature to use Ring in teaching Data Structures & Design Patterns.

Check the chapter (Using References) to learn more about using this feature!

118.10 Ring for MS-DOS

Starting from Ring 1.18, We can build Ring Compiler/VM on MS-DOS

Tested using

- Watcom C 386 9.01 (1992)
- Borland C 3.1 (1992)
- DOSBox-x 0.83.22 (2022)
- DOSBox 0.74

Tip: We use Watcom C 386 as our formal compiler on this platform

Note: Ring uses DOS/4GW (32-bit DOS extender) to eliminate the 640 KB conventional memory limit

118.11 Faster (For-In) Loop

The (For-In) Loop implementation is revised in Ring 1.18

It's six times (6x) faster than Ring 1.17

Example:

```
aList = list(1_000_000)
t1 = clock()
for x in aList
next
t2 = clock()
? (t2-t1)/clockspersecond()
```

Time using Ring 1.18: 0.4 second

Time using Ring 1.17: 2.6 seconds

Tip: The normal (For Loop) is faster than (For-In Loop) by 25%

Example:

```
aList = list(1_000_000)
t1 = clock()
for t=1 to 1_000_000
        x = aList[t]
next
t2 = clock()
? (t2-t1)/clockspersecond()
```

Time using Ring 1.18: 0.28 second

Time using Ring 1.17: 0.28 second

118.12 Faster BraceError()

Using BraceError() we can handle errors that happens inside braces

In Ring 1.18 - Calling BraceError() is faster and we can return values from this method

Example:

```
new points {
        first  { x=10   y=20   z=30   }
        second { x=100  y=200  z=300  }
        third  { x=1000 y=2000 z=3000 }
        print()
}

class points

        aPoints = []
```

(continues on next page)

(continued from previous page)

```
        func braceerror
                aPoints + new point
                return aPoints[len(aPoints)]

        func print
                ? aPoints

class point x y z
```

Output:

```
x: 10
y: 20
z: 30
x: 100
y: 200
z: 300
x: 1000
y: 2000
z: 3000
```

118.13 Better RingRayLib

The next functions are added to the extension

- SetShaderInt(shader, Location, nValue)
- SetShaderFloat(shader, Location, nValue)
- SetShaderVec2(shader, Location, aValue)
- SetShaderVec3(shader, Location, aValue)
- SetShaderVec4(shader, Location, aValue)
- SetShaderIVec2(shader, Location, aValue)
- SetShaderIVec3(shader, Location, aValue)
- SetShaderIVec4(shader, Location, aValue)
- SetModelMaterialShader(model, nID, shader)

The next samples are ported from RayLib to RingRayLib

- ring/samples/UsingRayLib/shader/custom_uniform.ring
- ring/samples/UsingRayLib/shader/julia_set.ring
- ring/samples/UsingRayLib/shader/model_shader.ring
- ring/samples/UsingRayLib/shader/postprocessing.ring
- ring/samples/UsingRayLib/shader/shapes_textures.ring
- ring/samples/UsingRayLib/shader/waves.ring

raylib [shaders] example - julia sets
Press Mouse buttons right/left to zoom in/out and move
Press KEY_F1 to toggle these controls
Press KEYS [1 - 6] to change point of interest
Press KEY_LEFT | KEY_RIGHT to change speed
Press KEY_SPACE to pause movement animation

raylib [shaders] example - postprocessing shader
CURRENT POSTPRO SHADER: PREDATOR_VIEW
61 FPS
(c)Church-3D model by Alberto Cano

118.14 Better RingSockets

The next functions are added to the extension

- inet_pton(AdressFamily, IP) -> packed_address
- inet_ntop(AdressFamily, packed_address) -> IP
- socketsCleanup()

118.15 Heroku (Better support)

In Ring 1.18 We updated the packages to use a modern Ring version

The project is tested using Heroku-22 (Ubuntu 22.04)

URL: https://github.com/ring-lang/RingWebAppOnHeroku

118.16 Command: New From

Using (new) we can create a new object from a specific class

In Ring 1.18 we have (new from) where we provide a variable which contains the class name

Using this command we can avoid using Eval() for this common case

Example:

```
cClassName = "myclass2"
myobj = new from cClassName

cClassName = "myclass"
myobj = new from cClassName

class myclass
        ? :hello

class myclass2
        ? :wow
```

Output:

```
wow
hello
```

118.17 ImportPackage() Function

Instead of using the import command we can use the importpackage() function

This function get the package name through a string or variable

This is useful if the package name will be known only during the runtime

Syntax:

```
importpackage(cPackageName)
```

Example:

```
importpackage(:mypackage)
new myclass { myfunction() }

package mypackage
        class myclass
                function myfunction
                        ? "Hello, World!"
```

118.18 More Low Level Functions

The next functions are added to the Low Level functions

- ringvm_ismempool()
- ringvm_runcode(cCode)

These functions could be helpful for writing tests!

The ringvm_ismempool() tell us if the memory pool still active or not.

The ringvm_runcode(cCode) execute code using Eval() & the Main Loop

See the chapter (Low Level Functions) for more information.

118.19 Better WebLib

(1) HTTP Post - Better Code

(2) **Application Class - More Methods:**

- Redirect()
- NoJavaScript()

(3) **Application Class - Attributes for JavaScript Libraries**

- cJSJQueryPath
- cJSBootstrapPath
- cCSSBootstrapPath
- lNoJavaScript

(4) **Page Class - More Methods:**

- theadStart(aOptions)
- theadEnd()
- tbodyStart(aOptions)
- tbodyEnd()
- tfootStart(aOptions)
- tfootEnd()

(5) WebPage Class - Support (THead, TBody & TFoot)

118.20 Better Ring API

- Added: RING_API_GETFLOATPOINTER(nParameter)
- Added: RING_API_ACCEPTFLOATVALUE(nParameter)
- Added: RING_API_GETCHARPOINTER(nParameter)
- RING_API_GETINTPOINTER - Better code when passing integer pointer
- RING_API_ISLIST - Don't accept empty strings (NULL) from Ring code
- Sample: extensions/tutorial/integerpointer
- Sample: extensions/tutorial/stringpointer

Example (C Code):

```
#include "ring.h"
#include "stdlib.h"

void changeValue(char *var)
{
        strcpy(var,"Hello from C Code");
}

RING_FUNC(ring_changevalue)
{

        if (RING_API_PARACOUNT != 1)
        {
                RING_API_ERROR(RING_API_BADPARACOUNT);
                return;
        }

        if (!RING_API_ISSTRING(1))
        {
                RING_API_ERROR(RING_API_BADPARATYPE);
                return;
        }

        char *p1 = RING_API_GETCHARPOINTER(1);
        changeValue(p1);
```

(continues on next page)

(continued from previous page)

```
}

RING_LIBINIT
{
        RING_API_REGISTER("changevalue",ring_changevalue);
}
```

Example (Ring Code):

```
? "Loading Library"
loadlib("mylib.dll")

cString = space(100)
changevalue(:cString)
? cString
```

Output:

```
Loading Library
Hello from C Code
```

118.21 Better Documentation

The next chapters are improved!

(1) Building RingQt applications for Mobile

(2) Building RingQt applications for WebAssembly

(3) How to compile Ring from source code?

118.22 MatrixLib Library

Ring 1.18 comes with the MatrixLib library (Contains the Matrix functions).

The source code exist in the ring/libraries/matrixlib folder

The samples exist in the ring/samples/UsingMatrixLib folder

The library comes with the next functions

```
MatrixMulti(A,B)             // Exists, aka MatrixMultiply(U,V)
MatrixTrans(A)               // Exists, aka MatrixTranspose(U)
MatrixPrint(U)
MatrixFlatPrint(U)
DotProduct(U,V)
MatrixAdd(U,V)
MatrixSub(U,V)
ScalarMultiply(k,U)
VectorAngle(U,V)
CrossProduct(U,V)
MatrixCofactor(U)
```

(continues on next page)

(continued from previous page)

```
12 MatrixAdjoint(U)
13 MatrixInverse(U)
14 MatrixDetCalc(U)              // Calls MatrixDeterminantReduce(U)
15 MatrixDeterminantReduce(U)    // Any 2x2 to 10x10 Recursive
16 Determinant(U)                // Calls 16a-16e Determinant2x2,3x3,4x4,5x5,6x6
17 MatrixProjection(U,V)         // ProjvU = (U.V)/(V.V)xV
18 MatrixOrthoDistance(U,V)
19 VectorLength(U)
20 VectorNorm(U)                 // Same as VectorLength - different name
21 VectorUnit(U)                 // Vector Norm to a Unit Matrix
22 VectorDistance(U,V)
23 MatricOrthoNormal2(U,V)       // Gram-Schmidt method for 2 Vectors in R2 Space
24 MatrixOrthoNormal3(V1,V2,V3) // Gram-Schmidt method for 3 Vectors in R3 Space
25 MatrixLuDecompose(U)          // LU Decompose Matrix to Lower and Upper Matrix
26 SyntheticFactor(Eq)           // Find Factors of Quartric Equation
27 SyntheticDiv(Eq,Factors)      // Synthetic Division using Factors
28 QuadSolve(Eq)                 // Quadratric equation solve format x^2 + x + c
29 CubicSolve(Eq)                // Cubic equation solve format x^3 + x^2 + x + c
30 QuarticSolve(Eq)              // Solve Quartic equation format x^4 + x^3 + x^2 + x + c
31 QuinticSolve(Eq)              // Solve Quintic equation format x^5 x^4 + x^3 + x^2 + x␣
↪+ c
31 PolyMultiply(A,B)             // Polynomial Multiple [A]*[B] ascending powers left to␣
↪right
32 PolyAdd(A,B)                  // Polynomial Add [A]+[B] ascending powers left to right
33 PolySub(A,B)                  // Polynomial Subtract [A]-[B] ascending powers left to␣
↪right
34 CharEquation(N)               // Characteristic Polynomial of 4x4 Matrix
34 EigenValueN)                  // Find EigenValues 2x2, 3x3, 4x4 Matrix
35 EigenVectors(A)               // Find EigenVectors 2x2, 3x3 Matrix
```

118.23 More Improvements

- NaturalLib - RunString() Method - Ignore braces inside commands
- FindInFiles Application: Better Code (Check Read() function output)
- GetQuotesHistory Application: Better Style
- Chess/Checkers Games: Looks more nice on small screen resolutions
- KnightTour Game: Avoid unnecessary invalid move message
- Tessera Game: Add the window icon
- SpaceShooter Game - Added: applications/spaceshooter/Resources.ring
- StdLib - Tree Class - Use the ref() function
- StdLib - Matrixmulti() function - Better Code
- StdLib - Added: Reduce() function
- RingAllegro: Use Allegro 5.2.8 instead of Allegro 5.2.7.1
- RingSQLite: Use SQLite 3.39.2.
- RingHTTPLib: Use cpp-httplib 0.10.9

- RingZip: Use zip 0.2.4
- RingCJSON: Update the CJSON version
- RingCJSON: Better support for arrays and nested arrays
- RingQt: Added QTextOption class
- RingQt: Set the environment variable QT_QPA_PLATFORM_PLUGIN_PATH
- RingQt: QMediaPlayer Class - Remove currentNetworkConfiguration() Method
- RingQt: QString2 Class - Added lastIndexOf() method
- RingQt: RingCodeHighlighter Class - Added setUseDefaultKeywords() Method
- RingQt: RingCodeHighlighter Class - Added setCustomKeywords() Method
- Ring Functions - SetEnv()/UnSetEnv() - Better Code
- Ring Functions - Print()/Puts() - Better Code
- Ring Functions - Eval() - Better Code
- Ring Compiler/VM Source Code - Better File Names
- Ring Compiler/VM Source Code - Define RING_MSDOS
- Ring Compiler/VM Source Code - Many structures updated to use unsigned int/char
- Ring Compiler/VM Source Code - Added: vmerror.c, vmeval.c & vmthreads.c
- Ring Compiler/VM Tests - Better names for output files
- Support building Ring on windows using Tiny C Compiler
- Ring Object File - Better support for classes that exist inside packages
- Ring Compiler - Support larger number of function/method parameters
- Ring Compiler - ring_parser_epsilon() function - Better Code
- Ring Compiler - Support running source code files with any extension
- Ring VM - Memory Functions - Display the size when the memory allocation fail
- Ring VM - Support calling private methods from the class region
- Ring VM - Fixed count for ByteCode items for the ICO_FUNC instruction
- Ring VM - Better code for dynamic change between function/method calls inside braces
- Ring VM - Check string range when we access strings after calling functions directly
- Ring VM - Get pointer as string when we access the pointer address
- Ring VM - Don't create more than one temp. list per VM instruction
- Ring VM - Using RING_VM_IR_GETLINENUMBER & RING_VM_IR_SETLINENUMBER(x)
- Ring VM - Faster code when creating new threads
- Ring VM - Better numbers format when printing lists and objects
- Ring VM - Range Operator - return list reference instead of a copy
- Ring VM - Pointer2Object() function - return list/object reference instead of a copy
- Ring VM - ObjectID() function - Support Lists too
- Ring VM - Increase the Stack size to 1K

- Ring VM - Error System - Display recursion depth
- Ring VM - ICO_FREESTACK instruction - Don't delete temp. lists in global scope
- Ring VM - ICO_FREETEMPLISTS instruction - Delete temp. lists in global scope (Thread Safe)
- Ring VM - ICO_SETSCOPE instruction - Better implementation
- Ring VM - List Functions - Using (unsigned int) for the list index & size
- Ring VM - Using Braces - Save/Restore the State - Better Code
- Ring VM - Divide Operator - Display error message - Better Code
- Ring VM - Try/Catch/Done - Better code for memory management
- Ring VM - Using lists during definition - Protect the list (Prevent deleting it)
- Ring VM - The default RING_POOLMANAGER_ITEMSIZE value is 40 (instead of 64)
- Ring VM - Better code when loading the VM instructions (Uses less memory)
- Ring VM - Max. parameters for each VM instruction is three (3)
- Ring VM - New instruction: ICO_EXTRAPARA (For instructions that need more parameters)
- Ring VM - Delete the Intermediate Code list after having the Byte Code
- Ring VM - Better format when displaying the Byte Code
- Ring VM - Memory defragmentation before Byte Code execution

CHAPTER

NINE

WHAT IS NEW IN RING 1.19

In this chapter we will learn about the changes and new features in Ring 1.19 release.

119.1 List of changes and new features

Ring 1.19 comes with the next features!

- Image Pixel application
- Hours Counter application
- Planetary Orbits application
- More Samples
- Ring for Windows 64bit
- Better batch files for building Ring
- For Loop - Better Performance
- Calling Functions - Better Performance
- Nothing() function
- OptionalFunc() function
- FastPro Extension
- Better TypeHints
- Better RingRayLib
- Better RingStbImage
- Better Extensions Generator
- Better Documentation
- More Improvements

119.2 Image Pixel application

An example about processing an image pixels.

119.3 Hours Counter application

This is a simple calculator for teaching hours

119.4 Planetary Orbits application

An example about drawing and using the Matrix library

119.5 More Samples

- Added: samples/UsingFastPro
- Added: samples/Language/Constants
- Added: samples/UsingTypeHints/test6.ring
- Added: samples/UsingQt/CustomDragAndDrop/customdraganddrop.ring
- Added: samples/UsingQt/PrintPreview/simplereport.ring
- Added: samples/UsingArabic/ArabicInCommandPrompt/DisplayArabic.ring
- Added: samples/UsingMatrixLib/59-Matrix-Transform-Draw-Rotation.ring
- Added: samples/UsingRayLib/more/ex5_wavingcubes_withoutthreads.ring
- Added: samples/Language/OptionalFunc/Question.ring
- Added: samples/Language/OptionalFunc/Answer.ring
- Added: samples/General/Performance/emptyloop.ring
- Added: samples/General/Performance/print.ring
- Added: samples/General/Performance/math.ring
- Added: samples/General/Performance/createlists.ring
- Added: samples/General/Performance/len.ring
- Added: samples/General/Performance/methods.ring

- Added: samples/General/Performance/manylistitems.ring
- Added: samples/General/Performance/search.ring

119.6 Ring for Windows 64bit

Starting from Ring 1.19 the default Ring for Windows version is 64bit

This provide access to more memory. Also, most of our tests indicates an increase of performance from 15% to 25%

This increase of performance is measured when we compare Ring 1.19 (32bit) and Ring 1.19 (64bit)

This is different from comparing Ring 1.19 (32bit) and Ring 1.18 (32bit) to see how much the new release is faster

119.7 Better batch files for building Ring

Using ring/buildvc.bat and ring/buildvc_x64.bat we can build everything (Ring Compiler/VM, Extensions, Tools, etc.) for Windows 32bit or 64bit

Ring comes with all of the dependencies except Qt

These batch files assume that we have Qt 5.15.16 (We can change the used Qt 5.15 version using the environment variables).

119.8 For Loop - Better Performance

The For Loop in Ring 1.19 is three times (3x) faster than Ring 1.18.

Tested using Victus Laptop [13th Gen Intel(R) Core(TM) i7-13700H, Windows 11]

Example:

```
t1= clock()
for t=1 to 100_000_000 next
t2 = clock()
? (t2-t1)/clockspersecond()
```

Time using Ring 1.18 (32bit) : 3.78 seconds

Time using Ring 1.19 (32bit) : 1.31 seconds

Time using Ring 1.19 (64bit) : 1.12 seconds

119.9 Calling Functions - Better Performance

Calling functions written in Ring code in Ring 1.19 is three times (3x) faster than Ring 1.18.

While calling functions written in C code in Ring 1.19 is four times (4x) faster than Ring 1.18.

Example:

```
t1=clock()
for t=1 to 1_000_000
        result = max(t,t*2)
next
? result
t2 = clock()
? (t2-t1)/clockspersecond()
```

Time using Ring 1.18 (32bit) : 1.45 seconds

Time using Ring 1.19 (32bit) : 0.32 seconds

Time using Ring 1.19 (64bit) : 0.25 seconds

These improvements let the Stars animation sample works at 2350 FPS in Ring 1.19 instead of 500FPS in Ring 1.18

119.10 Nothing() function

This function does nothing and can accept any number/type of parameters. The output will be Zero.

Some of the Use Cases

(1) Performance measurements, where we can test the performance of calling functions written in C and we can change the number of parameters during tests.

(2) In places of code that you want to write a function name, and it's not defined yet.

(3) To disable some feature/code by just changing the function name to nothing without changing the parameters or commenting the code.

(4) In small programs, where you want to write a function that you can override from a Test program.

119.11 OptionalFunc() function

Using this function we can define functions similar to Nothing() but with a different name.

Syntax:

```
OptionalFunc(cFunctionName)
```

Example:

File: Question.ring

```
optionalFunc(:reply)

? "I love Programming, What about you?"
reply()
? "Ok, Thanks!"
```

Output:

```
I love Programming, What about you?
Ok, Thanks!
```

File: Answer.ring

```
load "Question.ring"

func reply
        ? "Me too!"
```

Output:

```
I love Programming, What about you?
Me too!
Ok, Thanks!
```

119.12 FastPro Extension

This new extension comes with the next functions

- Bytes2List(cBytes,nWidth,nHeight,nChannels) —> aList // [[x,y,R,G,B,A],...]
- List2Bytes(aList,nChannels) —> cBytes // "RGBA...."
- updateList(<aList>,<cCommand>,<cSelection>,<nPara1>,[<nPara2>],[nPara3])
- updateColumn(<aList>, [<cCommand>,<nPara1>,[<nPara2>],[nPara3]],...)

Using the updateColumn() function we can update the list columns in one function call

Example from the ImagePixel application that convert the image to Gray

```
    //=====================================================================
    // GRAY SCALE -- Display Color RBG in GRAY Scale
    // Average looks better brighter than Gamma Corrected
    // Color corrected is for eye sensitivity Red 30%, Green 59% Blue 11%.

    elseif lGray

        updateColumn(MCOrig,:mul,RVALUE,0.3,            # R *= 0.3
                            :mul,GVALUE,0.59,           # G *= 0.59
                            :mul,BVALUE,0.11,           # B *= 0.11
                            :merge,RVALUE,GVALUE,       # R += G
                            :merge,RVALUE,BVALUE,       # R += B
                            :copy,RVALUE,GVALUE,        # G = R
                            :copy,RVALUE,BVALUE)        # B = R

    ok

```

This also provides better performance compared to calling updateList() many times.

119.13 Better TypeHints

Added the next definitions:

- Byte
- Boolean
- @override

Example:

```
load "typehints.ring"

o = new MyNewLib {
        ? isGreaterThanTwo(10)
        ? isGreaterThanTwo(1)
}

class MyLib {
```

(continues on next page)

(continued from previous page)

```
        boolean func isGreaterThanTwo(int x) {
                if x > 2
                        return true
                else
                        return false
                ok
        }
}

class MyNewLib < MyLib {
        @override
        boolean func isGreaterThanTwo(int x) {
                ? "Using override"
                return x > 2
        }
}
```

Output:

```
Using override
1
Using override
0
```

119.14 Better RingRayLib

The next functions are added to the RingRayLib extension

```
* vec2() --> Vector2
* vec2getx() --> double
* vec2setx(double)
* vec2gety() --> double
* vec2sety(double)
* vec3() --> Vector3
* vec3getx() --> double
* vec3setx(double)
* vec3gety() --> double
* vec3sety(double)
* vec3getz() --> double
* vec3setz(double)
* vec4() --> Vector4
* vec4getx() --> double
* vec4setx(double)
* vec4gety() --> double
* vec4sety(double)
* vec4getz() --> double
* vec4setz(double)
* vec4getw() --> double
* vec4setw(double)
* getcamera3dposx() --> double
```

(continues on next page)

(continued from previous page)

```
* setcamera3dposx(double)
* getcamera3dposy() --> double
* setcamera3dposy(double)
* getcamera3dposz() --> double
* setcamera3dposz(double)
* getcamera3dtarx() --> double
* setcamera3dtarx(double)
* getcamera3dtary() --> double
* setcamera3dtary(double)
* getcamera3dtarz() --> double
* setcamera3dtarz(double)
* getcamera3dupx() --> double
* setcamera3dupx(double)
* getcamera3dupy() --> double
* setcamera3dupy(double)
* getcamera3dupz() --> double
* setcamera3dupz(double)
* getcamera3fovy() --> double
* setcamera3fovy(double)
* getcamera3type() --> double
* setcamera3type(double)
```

119.15 Better RingStbImage

The extension is improved to support the next features

- Support UTF-8 file names on Windows
- Added functions for writing images

```
int stbi_write_png(char const *filename, int w, int h, int comp, const void *data, int␣
↪stride_in_bytes)
int stbi_write_bmp(char const *filename, int w, int h, int comp, const void *data)
int stbi_write_tga(char const *filename, int w, int h, int comp, const void *data)
int stbi_write_jpg(char const *filename, int w, int h, int comp, const void *data, int␣
↪quality)
void stbi_flip_vertically_on_write(int flag)
```

Also, When using Strings that contains bytes, we supported updating a character using a numeric value which will be converted to (char) without the need to use the Ring char() function.

This provide better performance (3x faster, i.e. 300%) when generating images.

This feature could be used to update the images quickly while treating them as bytes without the need to convert them to Lists.

Example:

```
load "stbimage.ring"

width    = 640
height   = 480
channels = 3
```

(continues on next page)

(continued from previous page)

```
cData    = space(width*height*channels)

? "Creating the image..."
t1 = clock()
nIndex=0
for x=1 to height
        for y=1 to width
                cData[nIndex++] = x*x
                cData[nIndex++] = x*y
                cData[nIndex++] = x*2
        next
next
t2 = clock()

# Write the image
? "Writing mynewimage.bmp"
stbi_write_bmp("mynewimage.bmp", width, height, channels, cData)
t3 = clock()

? "Time to create the image : " + ((t2-t1)/clockspersecond()) + " seconds"
? "Time to save the image   : " + ((t3-t2)/clockspersecond()) + " seconds"

system("mynewimage.bmp")
```

Output:

```
Creating the image...
Writing mynewimage.bmp
Time to create the image : 0.12 seconds
Time to save the image   : 0.00 seconds
```

Screen Shot:

119.16 Better Extensions Generator

The code generator for C/C++ extensions is updated and support defining extra names for functions

Using the predefined list (aExtraFunctionName) we can define the extra names

Example:

To define vec2() as another name for raylib_new_managed_vector2()

```
<runcode>
aExtraFunctionName + ["vec2","raylib_new_managed_vector2"]
aExtraFunctionName + ["vec2setx","raylib_set_vector2_x"]
aExtraFunctionName + ["vec2sety","raylib_set_vector2_y"]
</runcode>
```

119.17 Better Documentation

The CHM file is updated to support search

We added a new chapter: Using FastPro Extension

Also, The next chapters are revised and improved

- Introduction
- Performance Tips
- Syntax Flexibility
- The Type Hints Library
- Building From Source Code
- Extension using C/C++ languages
- Tutorial: Ring Extensions in C/C++
- Frequently Asked Questions (FAQ)

119.18 More Improvements

- Ring Notepad - Close Button - Force closing the web browser
- FindInFiles Application - Replace Button - Check line selection
- WinStartupManager application - Use extensions that comes with Ring
- EightPuzzleGame3D game - Better Code
- SuperMan2016 game - Reset FPS, energy value and player speed
- GoldMagic800 game - Reset FPS and player speed
- Sokoban game - higher speed
- TowersOfHanoi simulation - higher speed
- samples/AQuickStart/ODBC - Better Code
- samples/UsingOpenGL/LevelsOfCubes - Reset FPS
- samples/UsingOpenGL - Set the window icon
- samples/General/TimeServer - Replace RingThreads with GameLib
- samples/UsingQt/ListWidget/listwidgetitemvalue.ring - Better Code
- samples/UsingOpenGL/stars - Using al_exit() function
- samples/Drawing - Many samples are revised to use the Quit() method
- samples/UsingQt/PrintPreview/printpreivewdialog.ring - Better Code
- extensions/tutorial - filterlist() updated to use ring_list_deleteitem_gc()
- Better Qt3D samples - Use Quit() method when closing the application
- Better RingPostgreSQL samples
- Better RingMySQL sample - samples/AQuickStart/MySQL/mysql7.ring
- Form Designer - Better behavior when displaying the MenuBar editor

- StdLib - OSCopyFolder() - Better Implementation
- StdLib - OSCopyFile() - Better Implementation
- MatrixLib - Added: RowReduceEchelonForm() function
- MatrixLib - Added: MatrixTransform() function
- BigNumber - Better code for the power function
- ObjectsLib - Better Code when opening new windows
- GameEngine - Support playing many sound files at the same time
- RingQt - QPainter class - Added: drawRGBFListAtXY() & drawHSVFListAtXY()
- RingQt - Qt version is updated to Qt 5.15.16
- Number() function - Don't produce an error when the input is already a number
- String() function - Don't produce an error when the input is already a string
- Low Level Functions - ringvm_calllist() - Better Output
- Ring2EXE - Better Output
- Batch Files - locatevc.bat - Support working from the "C:/Program Files (x86)" folder
- Ring API - Added: RING_API_STATE to get a pointer to Ring State
- Ring API - Added: RING_API_NEWLISTUSINGBLOCKS1D(int nItems) –> List
- Ring API - Added: RING_API_NEWLISTUSINGBLOCKS2D(int nRows,int nColumns) –> List
- Ring Compiler - Remove unnecessary ICO_FREESTACK after For-Loops
- Ring Compiler - When we run ringsyntax.ring don't load the same file again
- Ring Compiler/VM - Check the path limit (4096 characters) before execution
- Ring Compiler/VM - Items structure functions - Better Performance
- Ring Compiler/VM - HashTable - Better default parameters
- Ring VM - For-Loop will produce a runtime error if used with wrong data-type
- Ring VM - Function Call - Replace some Ring lists with C Structures
- Ring VM - Use ICO_LOADFUNCP when loading C functions
- Ring VM - Memory Pool - Allocate 1,000,000 items at startup
- Ring VM - Byte Code - Added: Flag Register
- Ring VM - Byte Code - Added: Integer Register
- Ring VM - Use ring_list_genarray() with global scope variables When using threads
- Ring VM - Remove aAddressScope from VM structure
- Ring VM - New instructions (ICO_PUSH2N, ICO_PUSH3N & ICO_PUSH4N)
- Ring VM - Byte Code items count is changed to 4
- Ring VM - ICO_LOADFUNCP instruction support replacing C function with Ring function
- Ring VM - Better performance when deleting a scope - Avoid: ring_vm_gc_checkreferences()
- Ring VM - Better format when printing the byte code
- Ring VM - Variables - Reduce the required memory to store a variable

- Ring VM - Arguments - Reduce the required memory to pass a value
- Ring VM - ring_state_calloc() - Use the memory pool if possible
- Ring VM - Better implementation for using (Return This)

CHAPTER

USING OTHER CODE EDITORS

In the Ring/Tools/Editors folder you will find extensions for the next editors

- Notepad++
- Geany
- Atom
- Sublime Text 2
- Visual Studio IDE
- Emacs
- Visual Studio Code (VSCode)
- SpaceVim

120.1 Using Notepad++

Folder : ring/tools/editors/notepad_plus_plus

- Open Notepad++
- Open the "Language" menu
- Select "Define your language…"
- Click "Import…"
- select *Ring.xml*
- Select "OK" on the "Import successful" dialog and close the "User Defined Language" dialog/panel
- You may need to restart notepad++

120.2 Using Geany

Folder : ring/tools/editors/geany

- Run Geany editor
- Click on “Tools -> configuration files -> filetypes_extensions.conf” menu
- Add this line “Ring=*.ring;” without quotes after [Extensions]
- In Ubuntu copy file “filetypes.Ring.conf” to folder “/home/USERNAME/filetypes.Ring.conf”
- You can run your files by pressing F5 button

120.3 Using Atom

Folder : ring/tools/editors/atom

Just Copy the folder atom-language-ring to the next path

```
"C:\Users\{UserName}\.atom\Packages"
```

120.4 Using Sublime Text 2

Folder : ring/tools/editors/sublime text 2

In the folder Sublime_Text_2 you will find the next three files

1 - ring.json-tmlanguage

2 - ring.sublime-build

3 - ring.tmlanguage

Just Copy the files to the next path

```
"C:\Users\{UserName}\AppData\Roaming\Sublime Text 2\Packages\User\"
```

The file ring.sublime-build includes the next line

```
"cmd": ["B:\\ring\\bin\\ring.exe","$file"],
```

You can modify it according to the ring.exe path in your machine

120.5 Using Visual Studio IDE

Folder : ring/tools/editors/visualstudio

Check the ReadMe file for installation instructions.

120.6 Using Emacs Editor

Folder : ring/tools/editors/emacs

Check the ReadMe file for installation instructions.

Screen Shot:

120.7 Visual Studio Code

Folder : ring/tools/editors/vscode

Check the ReadMe file for installation instructions.

Screen Shot:

120.8 SpaceVim

URL: https://github.com/SpaceVim/SpaceVim

Screen Shot:

CHAPTER

ONE

FREQUENTLY ASKED QUESTIONS (FAQ)

121.1 Why do we need Yet Another Programming Language (YAPL)?

The language comes with better support for natural language programming and declarative programming. The innovation comes in supporting these paradigms with new practical techniques on the top of object-oriented programming and functional programming. Ring provides the programmers with the tools required to build a natural language like Supernova or a declarative language like REBOL and QML without the need to know anything about (compilers and parsing). You get the language constructs ready for use to create domain-specific languages in a fraction of time.

Take a look at the Supernova programming language, in this language you can type: (I want window and the window title is hello world.) and it will create a GUI window with "Hello, World!" as the window title. When I created Supernova language in 2010, i discovered that using the natural code can be (similar to English and without limits and we can use the power of human language in programming) but to implement that you need a new language that has:

(1) General Purpose

(2) Practical

(3) Can create natural languages very quickly.

So we can get a system that can increase ease of use and productivity to the maximum level.

So I created Ring because it was the best way to achieve this goal.

Supernova was just a test of the idea, it helped getting a better view of the advantages and the disadvantages of the idea. And After testing the new ideas you are provided with something practical. So now we have Ring after Supernova. A story that is maybe similar to having Python after ABC. Where Python avoids the problems of ABC, but keeps the advantages of ABC. Also, Ring learns from Ruby and ROR's story. The language power could appear in frameworks better than the direct usage as a general purpose language. Also Ring comes with a clear goal/motivation; (Creating a new version of the PWCT Software) something that was learned from the design the C language in a certain way to create the Unix Operating System. In other words, you have a goal that directs you in each design decision.

You will understand the value of our decisions once you start trying to solve the problem that we will use Ring to solve. The questions is: could you enable any one in the world without knowledge about computer programming concepts to create very powerful software? Scientifically the answer is (visual Programming) and (natural Programming). In practice we are still away from switching to these paradigms without introducing other problems. Ring is designed to solve this problem. It is designed to provide natural programming in a practical way. And to create a powerful visual programming tool. Ring is designed to be a new world of programming after 10 years of research in visual programming and natural languages.

The Ring Programming Language (Compiler+VM) is developed 100% using visual programming without writing a single line of code. I used my tool (Programming Without Coding Technology) to design everything and get the C code generated for me.

Advantages ?

(1) Faster

(2) No Syntax Errors

(3) Easy to understand and manage the code because the abstraction level is higher

(4) No critical disadvantages because you can control everything as writing your code.

Using my experience in using visual programming for 10 years and natural programming for 5 years, I designed Ring to move the knowledge to mainstream programmers by providing a practical language that supports these ideas.

I agree that each programmer/developer has the freedom to form his opinions about any software including programming languages. Ring is not an exception but you may miss the idea behind the language. It is innovative and may help you to think differently about how to solve your problems. Maybe this is not clear to many programmers because It is a practical language and includes many features known to programmers and when a programmer looks at the language they might think that nothing new because it's familiar. I created Ring to solve problems in a different way. Where I will start programming just by describing the software using new natural interfaces that I will implement later when I move from the design stage to the implementation stage. (I don't determine the time to switch between stages, You are free to use Agile methods). Since Ring is a new language you have 3 options:

(1) To not care at all for now.

(2) Think of the future of the language and help us if you understand the idea and want to contribute.

(3) Wait and come back again in the future to use it.

Summary:

- Ring is designed based on a need to develop a new version of the PWCT software.

Once we finish PWCT 2.0 we will have good and large software developed using Ring.

- We will push declarative and natural paradigms many steps forward. Also in next versions

we have a plan to present a new paradigm for network programming and concurrency. We tested this new paradigm through simple prototypes during the last years and we will integrate it with Ring in future releases.

121.2 Which of 3 coding styles are commonly used or recommended by the community?

(1) **Just select any style of them but don't mix between the different styles in the same project**
or at least in the same context (Implementation, Tests, Scripts, etc)

Note: State the rules in the start of each project and follow it.

(2) You can create your style (by changing keywords) - The idea is about customization and freedom.

Note: It's better to change keywords and create new style only for a clear reason like using another natural language (Arabic, French, etc.)

(3) **The First style is better (IMHO) for questions, tutorials and small applications/programs (Less than 5,000 LOC)**
Example : Ring Book, Most of Ring Samples and Applications.

(4) The Third style is better(IMHO) for large applications and mainstream programmers

Example (Form Designer) : https://github.com/ring-lang/ring/tree/master/tools/formdesigner

121.3 What is the Ring Architecture?

We have the next architecture

(1) Ring Applications (Your Code) - Written in Ring - See folder : ring/applications

(2) Ring Libraries (StdLib, WebLib, GameEngine, etc) - Written in Ring - See folder : ring/libraries

(3) Ring Extensions (RingAllegro, RingQt, etc) - Written in C/C++ (may include Ring code) - See folder : ring/extensions

(4) Ring Virtual Machine (Ring VM) - Written in C language

(5) Operating System (Current Platform) - (Windows, Linux, macOS, Android, etc)

The extensions are just dynamic libraries (DLL, So, Dylib) You can update the extensions without the need to update your code.

Folder (ring/extensions/libdepwin) ====> C libraries used for building Ring Extensions (written in C) on Windows platform

Folder (ring/libraries) ====> Ring libraries written in Ring itself (StdLib, WebLib, GameEngine, etc)

Folder (ring/language/visualsrc) ====> The Visual Source Code of the Ring Compiler & Ring VM developed using Programming Without Coding Technology (PWCT)

We use the term Ring Library —> When the library code is written in Ring We use the term Ring Extension —> When the library code is Written in C or C++

121.4 What about Memory Management in Ring?

(1) When we call function, we get a new scope for this function, inside this scope we store variables.

Also we get Temp. memory for this function. Inside this temp. memory we store temp. lists

All of this will be deleted directly after the end of the function call (end of scope)

(2) In Ring to delete memory, you have the next options

2.1 Wait until the end of the function scope

2.2 Use the Assignment operator

2.3 Use callgc() function (delete temp. lists only)

(3) Ring Garbage Collector uses Escape Analysis and Reference Counting

In 90% of cases, the Escape Analysis is used so we don't waste time in running the garbage collector We directly know what will be deleted and what will remain in the memory.

https://en.wikipedia.org/wiki/Escape_analysis

In 10% of cases (or less than that) Ring may use Reference Counting.

For example when we pass a list and sub list to a function Ring will pass the lists by reference, but what will happens if we deleted the Parent List? In this case, the Sub List will use reference counting, and when deleting the Parent List, it will stay in memory until the end of the function.

Remember that Ring encourage us to avoid using references, and the Assignment Operator will copy lists by value, so Ring usage of reference counting is very limited to special cases and in most cases the Escape Analysis is enough which is very fast.

Starting from Ring 1.9 we extended the Reference Counting support to Ring Extensions and low level C pointers. So we don't have to care about using fclose() when we use fopen() for example. and the same for other extensions like RingODBC, RingSQLite, RingMySQL, RingQt, etc.

All of the allocated resources will be cleaned by the Garbage Collector when we finish using it (when we lost the last reference).

121.5 What about Data Representation in Ring?

(1) In Ring, The String is just (Array of bytes)

Ring is 8-bit clean, Each character in the string is 8 bits (1 byte)

So these functions (Int2Bytes(), Float2Bytes() and Double2Bytes()) just return a string.

Also we can store binary data in strings

```
mystring = read("myfile.exe")
```

(2) Remember this, when you think about variables

- Value —> What we have (What we are storing in the computer memory as data) - Low Level Concept
- Type —> What we can do with what we have or how we do things with what we have (Just a Logical Concept)

Computer memory —-> Just store [Bytes] - Each byte is 8-bit - (Here we avoid the memory word concept)

These bytes could be grouped together when moved between the memory and the processor registers. Here we have (The register size) and things like 32-bit and 64-bit for example. Also we have the bytes order.

Programming Languages —-> Add Types (Just as a concept) to these bytes so we can determine what to do with them and how operations should be done.

And programming language could allow (Type Conversion) —> Because the Type is a logical concept in most cases, What we really have is just data (Bytes, Bytes Count, Bytes Order, etc)

Ring Strings —-> You have these bytes (each byte is 8-bit) and Ring know the string size (stored as number in the String structure)

So we don't check the NULL character or add it to the end of the string (Not Required)

All operations inside Ring VM, will check the Ring size and deal with the string as binary data (each character is 8-bit)

In the C language —> The normal case is adding NULL character (0) to the end of each string

And the string functions check this character, This is not suitable for binary data.

Signed vs Unsigned —> Is a logical concept which is important when you do arithmetic operations on the data, but when storing the data, if you will include all of the (8-bits) and will not ignore any of them —> Then don't care.

In Ring, don't think about these details, we are hiding it from you, so you can focus on your application and what you will do.

Think in C when you write C code which could be (based on need) low level code to have control on everything. —-> Good for performance and memory management

Think in Ring when you write Ring code which let you ignore a lot of details and concentrate only on the result —-> Good for productivity and delivering software quickly

The good news (We can mix between Ring and C in our projects)

(3) The functions Int2Bytes(), Float2Bytes() and Double2Bytes()

These function take input as (Number) —> Convert it to group of bytes based on the number type (int|float|double) —> Then return a Ring string that contains these bytes

Int2Bytes() —> Ring string (Group of bytes) and the string size = sizeof(int)

Float2Bytes() —> Ring string (Group of bytes) and the string size = sizeof(float)

Double2Bytes() —> Ring string (Group of bytes) and the string size = sizeof(double)

Example:

```
? len( int2bytes(1) )
? len( float2bytes(1) )
? len( double2bytes(1) )
```

Output:

```
4
4
8
```

(4) Storing Numbers

When we use a number, Ring always use the (Double) data type for representing these numbers in memory. This is important to know when we do arithmetic operations on numbers.

But when we convert the number to a String using "" + number or using string(number) we get a string where each digit is represented in 1 byte (Not good idea for storage, but useful for string processing)

If you need the number to be represented in specific size (int|float|double) for storage then use bytes2int() , bytes2float() and bytes2double() when writing the data to binary files.

Ring Number (double) —-> int2bytes() - will cast the number from double to int then return the bytes —-> 4 bytes (Ring String)

Ring Number (double) —-> float2bytes() - will cast the number from double to float then return the bytes —-> 4 bytes (Ring String)

Ring Number (double) —-> double2bytes() - will use the number (double) to return the bytes —-> 8 bytes (Ring String)

The (int) type is used only for internal Ring operations, but Ring applications|code will use only the (double) type for numbers.

(5) The Unsigned() Function

The function unsigned() expect the first and the second parameters as numbers

```
unsigned(nNumber1,nNumber2,cOperator)
```

We can use the bytes2int() function to convert the bytes to a number

Example:

```
B = list(4)

for k=1 to 4
{
        B[k]= Space(4)
        for kk=1 to 4 { B[k][kk]= char(60+4*k +kk) }
        ? " B" +k +": " +B[k]
}
```

(continues on next page)

(continued from previous page)

```
A12= Space(4)     A12= bytes2int(B[1]) ^ bytes2int(B[2])
? "A12: " +string(A12)
A34= Space(4)     A34= bytes2int(B[3]) ^ bytes2int(B[4])
? "A34: " +string(A34)
A12= space(4)     A12= Unsigned(bytes2int(B[1]),bytes2int(B[2]),"^")
? "unsigned A12: " +A12
A34= space(4)     A34= Unsigned(bytes2int(B[3]),bytes2int(B[4]),"^")
? "unsigned A34: " +A34
```

Output:

```
B1: ABCD
B2: EFGH
B3: IJKL
B4: MNOP
A12: 201589764
A34: 470025220
unsigned A12: 201589764
unsigned A34: 470025220
```

121.6 Why is Ring weakly typed?

Because it's faster and more natural, and this is important for the language's goals. One of the rules is: the data type at the beginning affects the final result. For example, when you type "Print : " + 5 , The String comes first, so 5 will be converted to a String. While when you type 5 + "10" The number comes first so "10" will be converted to 10. This helps a lot to quickly convert between numbers and strings using the same operator. If you want to prevent conversion (Write code that prevent conversion) In these cases you will notice that what you are writing is less code (And can be removed).

Weakly typed = automatic conversion and *automatic* is *good thing* and is better than *manual* if you know how to use it correctly.

121.7 What are the advantages to using Ring over Lisp or Smalltalk?

Smalltalk and Lisp are GREAT languages. I like many of the concepts behind them but I'm sure that selecting the right programming language is based on the problem and comes after the problem's definition. I have a problem that I want to solve and these GREAT languages are not ideal for this problem so I designed Ring.

When you design a new language, You can learn from the past but you must look forward and live in the future. What you know about natural programming maybe based on the *old knowledge* about the power of these paradigms in the practical world and I agree with you but I see other techniques that can be applied to get this to work in practice. What you miss about *natural language* is that they are *context sensitive* and this means we can use it and think differently about how we can express our ideas.

Example : I want window contains 3 buttons.

In one sentence I created 4 objects (The window and the three buttons) and added the buttons to the window. The idea of natural programming is to get many things done like that.

121.8 Why is Ring largely focused on UI creation?

Yes UI creation is one of the important things in the language features because it is designed to create a visual programming tool, But the language is a multi-paradigm language where we can select the programming paradigm based on the problem.

121.9 Is Ring some sort of an improvement of PHP?

Ring is not designed to replace PHP, Lua or Smalltalk. Ring's support for declarative programming and natural language programming is very innovative and much better than staying with procedural, object-oriented and functional languages. Ring see the future in programming without code (using natural languages) and is designed to support that.

121.10 What are the advantages of using Ring over native C or C++?

Ring provides a better way to mix between different programming paradigms in solving problems.

The different programming paradigms play well together in the same language.

(1) It's easy to switch from one programming paradigm to another one because the language constructs use similar syntax for similar concepts.

(2) The paradigms are provided to interact and used together in different layers in the software.

for example you can create a game engine using object-oriented programming but write the game code using declarative programming or natural programming and behind the scenes your declarative or natural code will use the object-oriented classes.

(3) Ring is more productive and natural than C/C++.

(4) Ring is a dynamic language. We can generate and execute code during the runtime. Ring is dynamically typed and weakly typed for flexibility.

(5) The Garbage collector is generational (escape analysis) and also uses reference counting. it's very fast and still provides control to the programmer who can delete memory at any time.

(6) Ring's compiler and virtual machine are just 20,000 lines of ANSI C code that can be compiled and used in any platform.

(7) You can use C/C++ libraries and Ring comes with code generator to create wrappers from C functions or C++ classes. so when you need more performance or when you need to use more libraries you can easily do that.

121.11 What is the difference between Ring and Python? And is Ring Open Source?

Yes the language is Open Source (MIT license)

In general I like Python and Ruby but I was looking for a language more suitable for creating the next version of the Programming Without Coding Technology (PWCT) software so I started the Ring design.

Some simple changes that matters for my goal are

(1) Not case sensitive

(2) The list index start from 1

(3) You can call functions before definition

(4) Don't use Python syntax like (indentation, using self, :, pass & _)

(5) Weakly typed (convert automatically between types based on context)

(6) The programs follow simple and constant structure (Statements then functions then packages and classes)

(7) Using the '=' operator for assignment and for testing values

Critical changes are

(1) Small Language : The Ring compiler + Virtual Machine = 15K lines of C code , the other 500K lines are related to libraries and are optional when we go for using the language in C/C++ programs.

(2) The Garbage collector : Uses Escape Analysis/Reference counting and give the programmer the ability to determine when to delete memory using the assignment operator

(3) Compact Syntax : Ring is not line sensitive, you don't need to write ; or press ENTER to separate between statements

(4) Using { } to access the object then using the object attributes and methods directly

(5) Natural Programming : It's very easy to create natural interfaces using Ring based on OOP

(6) Declarative Programming using Nested Structure

The Ring programming language is designed based on my experience from using many other languages like C, C++, C#, Lua, PHP, Python, Ruby, Harbour, Basic and Supernova And the language comes with innovative features added to achieve the language goal

- Applications programming language.
- Productivity and developing high quality solutions that can scale.
- Small and fast language that can be embedded in C/C++ projects.
- Simple language that can be used in education and introducing Compiler/VM concepts.
- General-Purpose language that can be used for creating domain-specific libraries, frameworks and tools.
- Practical language designed for creating the next version of the Programming Without Coding Technology software.

121.12 What are the advantages to using Ring over Perl, PHP, Python or Ruby?

(1) Ring is New and Innovative. The language will let you think different about programming.

(2) Ring is Smaller. (Lessons learned from the Lua language)

(3) Ring is Simple. (Lessons learned from the BASIC and Clipper/Harbour languages)

(4) Ring is more Natural. (Lessons learned from the Supernova language)

(5) Ring is more Declarative. (Lessons learned from REBOL and QML languages)

(6) Ring Implementation is Transparent, Visual and comes with Rich Features.

121.13 What are the advantages to using Ring over Tcl or Lua?

(1) Clean Code (More Natural)

(2) More Features (A lot of useful programming paradigms)

121.14 What are the advantages to using Ring over C# or Java?

(1) Compact Code (Clean and Natural), More Productivity and Flexibility.

(2) Better support for Declarative Programming and Natural Programming

121.15 The documentation says functional programming is supported, but then this happens?

The question was about this code

```
f = func {
    a = 42
    return func { return a }
}

innerF = call f()
call innerF()
```

Output:

```
Using uninitialized variable : a In function _ring_anonymous_func_16601()
```

The Answer:

- It's Anonymous Functions, i.e. Not Closures.
- Many developers asked about supporting Closures and during language development we may add new features that doesn't go against the language goals or spirit.
- You can use classes and objects when you want to merge between the state and functions to provide a clear solution.
- You can use Lists and put the anonymous function inside the List then return the list that contains the state and the function. Pass the list to the function when you use it.
- You can use eval() and substr() to add the variable value directly to the anonymous function before return.
- We protect you from other scopes when you define the function. In Ring we provided the Three Scopes Rule where at each point you have only at maximum three scopes (Global, Object Scope and Local Scope).
- We don't get everything from everywhere to be like others! We don't need to do that. If we will think like that then we will create a very complex language or we will save our time and use other languages.
- When you think about learning or studying a new language concentrate about (What is new?) and (What is better in this language?) to know when to use it. Don't compare a new language just released little months ago with languages started many years ago and expect to find everything that you used to have.

- Each programming language miss features in other languages. The idea is not the Features. it's the spirit and ability behind all of the features together.

121.16 Why the ability to define your own languages Instead of parsing?

It's innovation - You create natural statements without the need to learn about parsing. You just use Classes which is intelligent decision (where later we can mix between classes to support more statements based on the context - We can change and translate the defined statements and many more!). Also the statements are added in Ring World where you can use any Ring statement.

121.17 Why you can specify the number of loops you want to break out of?

The language supports programming in the small and programming in the large. The selection of what features to use is based on what are you going to do. Any programmer can write poorly code in any language if he/she wants to do that. The idea is what must be done in the language design to prevent errors without causing other problems like killing flexibility.

Read some source code in the Linux Kernel and Ruby Implementation for example, You will find good usage for GOTO as a practical example that General Rules are not for All Use Cases and great programmers know when to break the rules. I'm not saying go and use GOTO or saying Ring add things like that. But the ability to break more than one loop and/or the ability to break the loop from sub functions is practical for small programs.

Anyway these are some of the small new things added by the language (Not the big idea).

121.18 Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?

See and Give are selected not to be "opposite actions" but to reflect what I want to do as a programmer.

When I want to see something on the screen I use 'See'.

When I want to give some input to the program I use 'Give'.

My selection of "but" and "ok" is based on selecting keywords that can be written quickly.

Also using "but" is easy to remember than elseif/elif/elsif where each language select a different keyword.

In Ring 1.1 and later versions All of this is just an option.

You can use 'Put' and 'Get' instead of 'See' and 'Give'

You can use 'elseif' and 'end' instead of 'But' and 'Ok'

It's your choice. In Ring we have syntax flexibility where we provide more than one style.

Also you can change the language keywords and operators.

Also you can define new natural languages too.

121.19 What is the philosophy behind data types in Ring?

The Ring programming language is designed to be SMALL. The language provides the basic constructs that you need to do anything! One of the goals is to keep the basic constructs simple and small as possible.

Using Lists in Ring you can

- Create Arrays (one data type)
- Create Lists (Mix of data types)
- Create Tree (Nested arrays)
- Use String Index (Looks like Dictionary/Hash Table)

The same principle is applied to Numbers

- You can use the number for int value
- You can use the number for double value
- You can use the number for Boolean value (True/False)

The sample principle is applied for Strings

- You can use the string for storing one character
- You can use the string for storing text (one or many lines)
- You can use the string for storing binary data
- You can use the string for storing date
- You can use the string for storing time
- You can use the string for storing NULL values (empty strings)

And we have Object Oriented Support + Operator Overloading where the programmer can define new data types and use them as default types defined by the language

So We have

- A small and simple language that someone can pick in little days
- A fast language that provide primitive types (String - Number - List - Object)
- A flexible language that can be extended using OOP to add new types according to the application domain

121.20 What about the Boolean values in Ring?

You can use true for 1 and false for 0

when you test the result of Boolean expressions in your code.

Just when you print the value using the see command you will see 1 for (true) and 0 for (false)

Why ?

Because Ring contains only 4 types of variables

(1) Number

(2) String

(3) List

(4) Object

The first type (Number) is used to represent int, double and Boolean values.

The second type (String) is used to represent char, array of characters, date and time.

The third type (List) is used to represent Arrays of one type, Arrays of more than one type, Hash (Dictionary), Tree, etc.

The object can be an object created from a Ring class (Any Class) or just a C Pointer that we get from calling a C/C++ function/method.

Why ?

The Ring is designed to give the programmer/developer the most simple constructs that can be used to do everything. The programmer/developer can customize the language by creating new classes (and use operator overloading) to get more types that he care about according to the problem domain.

Why ?

Because simple is better, and easy to learn and remember! And this provide flexibility to convert between high level types that can be represented using the same basic type

121.21 What is the goal of including the "Main" function in Ring?

The main function is very important, you need it when you want to write statements that uses local variables instead of the Global scope.

Example:

```
x = 10
myfunc()
See "X value = " + X  # here I expect that x will be (10)
                      # but I will get another value (6) because myfunc() uses x !
Func myfunc
   for x = 1 to 5
       See x + nl
   next
```

Output:

```
1
2
3
4
5
X value = 6
```

Now using the Main function

```
Func Main
   x = 10
   myfunc()
   See "X value = " + X

Func myfunc
   for x = 1 to 5
```

(continues on next page)

(continued from previous page)

```
        See x + nl
    next
```

Output

```
1
2
3
4
5
X value = 10
```

121.22 Why the list index start from 1 in Ring?

It's about how we count in the real world, when we have three apples in our hand

we say 1 2 3

We don't start from 0

The question must be why the other languages start from 0 ?

The answer is, because this is related to the machine and how we deal with values and memory address.

Example

we have array called myarray[5]

In memory : myarray will have an address

The first item will be stored in that address

The second item will come after that address and so on

Now when we need to point to the first item we need the address of myarray

So we type myarray[0] because myarray + 0 result will still point to the first item

for the second item myarray[1] because myarray + 1 result will point to the second item and so on

In Low Level languages or languages near to the machine it's good to be like this

But for high level language designed for applications it's better to be natural

Example

```
mylist = [1,2,3,4,5]
for x = 1 to len(mylist)
        see x + nl
next
```

In the previous example we start from 1 to the length of the array if the index starts from 0 we will write

```
for x = 0 to len(mylist)-1
```

or remember the for loop in other languages

```
for(x=0 ; x<nMax ; x++ )
```

You will use the < operator !

121.23 Why Ring is not case-sensitive?

(1) To be more human-friendly

(2) Like Ada, SQL, Pascal, Delphi, Visual Basic, Visual FoxPro, etc.

(3) To help in supporting Natural Language Programming.

(4) To be able to select your favorite style when writing the language keywords

```
see "lower case!"
```

```
SEE "UPPER case!"
```

```
See "First Letter is UPPER case!"
```

(5) To avoid getting error message when writing quick tests then type "variable" instead of "Variable".

(6) To avoid getting error message when you type "Dosomething()" instead of "doSomething()"

(7) In Ring, No conflict between Variables, Method Names & Classes Names

We can write person as variable name and Person as class name.

```
person = new Person
class Person
        name address phone
```

121.24 Why the Assignment operator uses Deep Copy?

"Because it's a poor tradeoff to add complexity for dubious performance gains, a good approach to deep vs. shallow copies is to prefer deep copies until proven otherwise."

, Steve McConnell, Code Complete

(1) It's more natural, When you use the assignment operator, You expect a deep copy.

(2) If you don't need a deep copy, Just don't use it!

(3) The Ring language is designed to reduce references usage as much as possible.

(4) The Ring language is designed to make using references simple and possible in special cases where this make sense.

(5) **We have references when this is natural, like passing lists and objects to functions,**
creating objects (Like GUI Objects) from a C/C++ library, returning an object stored inside a list.

(6) **It is a feature, We can use it to create pure functions. The Value() function in the**
stdlib uses this feature to pass lists & objects by value when we need this.

(7) When we need references, It's recommended to create a class that manage sharing lists and objects.

(8) It's more safe at the application level to avoid many logical errors.

(9) **In Ring, we start without thinking about the little details and concentrate on the application, You** don't have to write the type (Dynamic Typing), You don't have to write explicit conversions between numbers and strings (Weakly Typed) and you don't have to select between using values or references, You don't have to write the scope (Lexical Scoping).

(10) **In Ring, we have smart garbage collector (Simple & Fast), We can delete the memory directly** at any time using the Assignment operator too. Reducing references usage or using them through managers helps a lot to achieve this goal. by doing this we have full control.

(11) **If you want to create unmanaged pointers and avoid creating a manager,** You can use Object2Pointer() and Pointer2Object() functions But It's not the Ring way "Spirit" to do things.

(12) Starting from Ring 1.18, Ring comes with the Ref() function that can be used to create safe references.

121.25 Is there constructor methods in Ring?

When you create new object for example

```
new point
```

1 - Ring will allocate dynamic memory space to be used for the new object attributes that Ring doesn't know anything about them.

2 - Ring will change the current local scope and the current object scope to use the object state created in step (1)

3 - Ring will move the execution to the class Region (After the class name and before any methods)

4 - Any Instructions/Code in the class region will be executed as any Ring code

5 - Control is moved from the class region to the location of (new point) once we reach the end of the class region or we uses a Return command.

So All attributes that added to the object are dynamic attributes, this mean that you can control what attributes will be added through the runtime.

Example:

```
$3D = False
see  new point
$3D = True
see new point

class point
        x y
        if not $3D return ok
        z
```

Output:

```
x: NULL
y: NULL
x: NULL
y: NULL
z: NULL
```

You have an option to call init() method directly when you create a new object

This method can do anything with the object attributes as it will be called after creating the object and executing the class region code.

```
p1 = new point3d(100,200,300)
see p1

class point3d
        x y z
        func init p1,p2,p3
                x=p1 y=p2 z=p3
```

121.26 What happens when we create a new object?

1- When you create an object, the class region code will be executed and you will have the object attributes based on the code in that region

2- Ring don't care about the object methods until you start calling a method

3- When you call a method, Ring will check the object class and the class parent (if you are using inheritance) and will collect the methods for you to be used now or later from any object that belong to the same class.

4- Since methods are dynamic and each object get the method from the class, you can after creating objects, add new methods and use it with the object or any object created or will be created from the same class.

Example:

```
o1 = new point {x=10 y=20 z=30}
o2 = new point {x=100 y=200 z =300}

addmethod(o1,"print", func { see x + nl + y + nl + z + nl } )

o1.print()
o2.print()

class point x y z
```

Output:

```
10
20
30
100
200
300
```

121.27 Can we use the attributes by accessing the Getter and Setter methods?

Yes we can, The setter/getter methods are called automatically when you start using the attributes from outside the class Also you can call the methods instead of using the attributes. It's your choice.

Example:

```
o1 = new Developer
o1.name = "Mahmoud"  see o1.name + nl
o1 { name = "Gal"  see name }
o1 { name = "Bert"  see name }

o1.setname("Marino")
see o1.getname()

Class Developer

                name language = "Ring Programming Language"

                func setname value
                                see "Message from SetName() Function!" + nl
                                name = value + " - " + language

                func getname
                                see "Message from GetName() Function!" + nl + nl
                                return "Mr. " + name + nl
```

Output

```
Message from SetName() Function!
Message from GetName() Function!

Mr. Mahmoud - Ring Programming Language

Message from SetName() Function!
Message from GetName() Function!

Mr. Gal - Ring Programming Language
Message from SetName() Function!
Message from GetName() Function!

Mr. Bert - Ring Programming Language
Message from SetName() Function!
Message from GetName() Function!

Mr. Marino - Ring Programming Language
```

121.28 Why should a search of global names be made while defining the class attributes?

The question is why we don't avoid conflicts with global variable names when we define the class attributes ?

At first remember that using the optional $ mark in the global variables names solve the problem. Also using the Main function and avoiding global variables may help.

The Answer:

Ring is a dynamic language

We can in the run-time determine the class attributes (Add/Remove)

We can execute (any code) while defining the class attributes

Example (1)

```
oPerson = new Person
Class Person
   See "Welcome to the Ring language"
```

Example (2)

Customize attributes based on global variable value

```
$debug = true
oPerson = new Person
see oPerson
Class Person
    if $debug  date=date()  time=time() ok
```

In the previous example when we have the $debug flag set to true, we will add the Date and Time attributes to the object state.

Example (3)

Store the object index based on global variable

```
$ObjectsCount = 0
oPerson = new Person
see oPerson
oPerson2 = new Person
see oPerson2
Class Person
      $ObjectsCount++
      nIndex = $ObjectsCount
```

Output:

```
nindex: 1.000000
nindex: 2.000000
```

Common Example:

- Connect to the database then get table columns (Using global Variable/Object).
- Create class attributes based on the column names.
- Later when you modify the database - you may don't need to modify your code.

It's flexibility but remember that power comes with great responsibility.

121.29 Why Ring doesn't avoid the conflict between Global Variables and Class Attributes Names?

In this use case we have

1 - Global Variable defined without a special mark like $

2 - Class contains Attributes defined using a special syntax (where we type the attribute name directly after the class)

3 - The Attributes are defined in the class region that allows writing code and using global variables

If I will accepted your proposal about changing how Ring find variables in the class region I must break one of the previous three features which will lead to more problems.

I don't like changing the feature number (1) because I would like to keep Ring code more clean and let the programmer decide when to use $ or not.

I don't like changing the feature number (2) because I like this feature and I don't like forcing the programmer to type self.attribute

I don't like changing the feature number (3) because it's very important in many applications to access global variables in the class region.

So what was my decision ?

I decided to leave this case for the programmer who will decide what to do to avoid this special case

1 - The programmer can avoid using global variables (Better) and can use the Main function (Optional)

2 - The programmer can use $ before the variable name or any mark like **global_** or **g_**

3 - The programmer can use self.attribute after the class name to define the attributes

4 - Ring have a nice feature through the (load package) command which load a Ring source file in a new global scope different from the global scope used by the caller.

In general, for small programs you can use global variables and functions. For large programs, use classes and objects and small number of global variables or avoid them at all.

121.30 Where can I write a program and execute it?

Run the Ring Notepad where you can write/execute programs.

If you want to run programs using the command line

Add Ring/bin folder to the path then

```
ring <filename.ring>|<filename.ringo>
```

121.31 How to get the file size using ftell() and fseek() functions?

The next function can be used to get the file size without reading the file!

```
func getFileSize fp
       C_FILESTART = 0
       C_FILEEND = 2
       fseek(fp,0,C_FILEEND)
       nFileSize = ftell(fp)
       fseek(fp,0,C_FILESTART)
       return nFileSize
```

Note: The previous function take the fp (file pointer) as parameter, We can get the fp from opening the file using fopen() function.

```
fp = fopen("filename","r")

see  "File Size : " + getFileSize(fp) + nl
```

Another solution (Read the file)

```
see len(read("filename"))
```

121.32 How to get the current source file path?

We can use the next function to get the current source file path then we can add the path variable to the file name

```
cPath = CurrentPath()
func currentpath
        cFileName = filename()
        for x = len(cFileName) to 1 step -1
                if cFileName[x] = "/"
                        return left(cFileName,x-1)
                ok
        next
        return cFileName
```

Note: We can use the function JustFilePath() from the stdlibcore.ring

121.33 What about predefined parameters or optional parameters in functions?

if you want to use predefined parameters or optional parameters Just accept a list that works like hash/dictionary

Example

```
sum([ :a = 1, :b = 2])
sum([ :a = 1 ])
sum([ :b = 2 ])
func sum pList
        if plist[:a] = NULL pList[:a] = 4 ok
        if plist[:b] = NULL pList[:b] = 5 ok
        see pList[:a] + pList[:b] + nl
```

Output

```
3
6
6
```

121.34 How to print keys or values only in List/Dictionary?

If you want to print keys only or values only just select the index of the item (one or two).

Example

```
C_COUNTRY = 1
C_CITY = 2
mylist = [
        :KSA = "Riyadh" ,
        :Egypt = "Cairo"
]

for x in mylist
        see x[C_COUNTRY] + nl
next

for x in mylist
        see x[C_CITY] + nl
next
```

Output

```
ksa
egypt
Riyadh
Cairo
```

121.35 Why I get a strange result when printing nl with lists?

In the next code

```
list = 1:5        # list = [1,2,3,4,5]
see list + nl
```

New Line will be added to the list then the list will be printed, the default print of the lists will print a newline at the end, You added new newline and You have now 2 newlines to be printed.

```
See <Expr>
```

The see command just print the final result of the expression, the expression will be evaluated as it

```
nl = char(13) + char(10) # just a variable that you can change to anything !
```

The + is an operator

```
string + string ---> new string
string + number ---> new string
number + number ---> new number
number + string ---> new number
```

list + item —> nothing new will be created but the item will be added to the same list

Exception

number + nl -> New String

This exception is added to easily print numbers then new line.

No need for this with printing lists because after printing the last item we already get a new line.

121.36 Could you explain the output of the StrCmp() function?

At first remember that you can check strings using '=' operator directly.

```
see strcmp("hello","hello") + nl +
strcmp("abc","bcd") + nl +
strcmp("bcd","abc") + nl
```

if the two strings are the same then it returns 0

abc and bcd aren't the same. in the second line it returns -1 and in the third line it returns 1

In the second line we compare between "abc" and "bcd"

Not equal because the first letter in "abc" = "a" and the first letter in "bcd" = "b"

So we have "a" != "b" and "a" < "b"

So we get output = -1

In the third line we have "bcd" and "abc"

the first letter in "bcd" is "b" and the first letter in "abc" is "a"

So we have "b" != "a" and "b" > "a"

So we get output = 1

Note: ASCII("a") = 97 and ASCII("b") = 98 So "a" < "b" because 97 < 98

121.37 How to use many source code files in the project?

Example:

I have the next folder

```
C:\LRing
```

Contains the next files

```
C:\LRing\t1.ring
C:\LRing\mylib.ring
C:\LRing\libs\mylib2.ring
```

The file t1.ring contains the next code

```
load "mylib.ring"
load "libs\mylib2.ring"
myfunc()
test()
```

The file mylib.ring contains the next code

```
func myfunc
        see "message from myfunc"+nl
```

The file libsmylib2.ring contains the next code

```
func test
        see "message from test" + nl
```

from the folder C:LRing

If Ring is not added to the path you can add it or use the next command

```
set path=%path%;c:\ring\bin;
```

Where c:ring is the Ring folder

Now run

```
Ring t1.ring
```

Output

```
message from myfunc
message from test
```

121.38 Why this example use the GetChar() twice?

The GetChar() function accept one character from the keyboard buffer

In this example

```
While True
        See "
                Main Menu
                (1) Say Hello
                (2) Exit
        "
        Option = GetChar()
        GetChar() GetChar()  # End of line
        # the previous two lines can be replaced with the next line
        # Give Option

        if Option = 1
                see "Enter your name : " give cName
                see "Hello " + cName
        else
                bye
        ok
End
```

We uses GetChar() Three times

The first time we get the user option

```
Option = GetChar()
```

But in the second and the third times (We accept the new line characters from the buffer)

```
GetChar() GetChar()  # End of line
```

Example : when the user select the option number 1 then press ENTER

We have Three Characters

- The first character is : Number 1
- The second character is : CHAR(13)
- The third character is : CHAR(10)

Because Windows uses CHAR(13) and CHAR(10) for each new line (i.e. CR+LF)

121.39 How to use NULL and ISNULL() function?

when we try to use uninitialized variable in the Ring programming language, we get a clear runtime error message

Example

```
See x
```

Output

```
Line 1 Error (R24) : Using uninitialized variable : x
in file tests\seeuninit.ring
```

The same happens when you try to access uninitialized attributes

Example

```
o1 = new point
see o1
see o1.x
class point x y z
```

Output

```
x: NULL
y: NULL
z: NULL

Line 3 Error (R24) : Using uninitialized variable : x
in file tests\seeuninit2.ring
```

if you want to check for the error, just use Try/Catch/End

```
Try
        see x
Catch
        See "Sorry, We can't use x!" + nl
Done
```

Output

```
Sorry, We can't use x!
```

Now we will talk about NULL and ISNULL()

Since we get error message when we deal with uninitialized variables

We can check these errors using Try/Catch/Done, So we uses NULL and ISNULL() for dealing with Strings.

NULL is a variable contains an empty string

ISNULL() is a function that returns true (1) if the input is an empty string or just a string contains "NULL"

This because we need to test these values (empty strings) and strings contains "NULL" that sometimes come from external resource like DBMS.

Example

```
See IsNull(5) + nl +        # print 0
IsNull("hello") + nl +      # print 0
IsNull([1,3,5]) + nl +      # print 0
IsNull("") + nl +           # print 1
IsNull("NULL")              # print 1
```

121.40 How to print lists that contains objects?

In this example we will see how we can print a list contains objects.

```
aList = [[1,2,3] , new point(1,2,3), new point(1,2,3)]
see "print the list" + nl
see alist
see "print the item (object)" + nl
see alist[2]
class point x y z
        func init p1,p2,p3 x=p1 y=p2 z=p3
```

Output

```
print the list
1
2
3
x: 1.000000
y: 2.000000
z: 3.000000
x: 1.000000
y: 2.000000
z: 3.000000
print the item (object)
x: 1.000000
y: 2.000000
z: 3.000000
```

121.41 How to insert an item to the first position in the list?

To insert an item we can use the insert(aList,nIndex,Value) function.

```
aList = 1:5
insert(aList,0,0)
See aList # print numbers from 0 to 5
```

121.42 How to print new lines and other characters?

To print new line we can use the nl variable.

```
See "Hello" + nl
```

or we can use multi-line literal as in the next example

```
See "Hello

"
```

if we want to print other characters we can use the char(nASCII) function

```
See char(109) + nl +    # print m
    char(77)            # print M
```

121.43 Why I get Calling Function without definition Error?

Each program follow the next order

1 - Loading Files 2 - Global Variables and Statements 3 - Functions 4 - Packages, Classes and Methods

So what does that mean ?

(1) **** No Functions comes After Classes ****

(2) **** No command is required to end functions/methods/classes/packages ****

Look at this example

```
See "Hello"
test()
func test
    see "message from the test function!" + nl
class test
```

In the previous example we have a function called test() so we can call it directly using test()

In the next example, test() will become a method

```
See"Hello"
test()    # runtime error message
class test
        func test # Test() now is a method (not a function)
                see "message from the test method!" + nl
```

The errors comes when you define a method then try calling it directly as a function.

The previous program must be

```
See"Hello"
new test { test() }   # now will call the method
class test
        func test # Test() now is a method (not a function)
                see "message from the test method!" + nl
```

121.44 Can Ring work on Windows XP?

Ring can work on Windows XP and load extensions without problems.

Just be sure that the extension can work on Windows XP and your compiler version support that (modern compilers requires some flags to support XP)

Check this topic https://blogs.msdn.microsoft.com/vcblog/2012/10/08/windows-xp-targeting-with-c-in-visual-studio-2012/

For example, We added

```
/link /SUBSYSTEM:CONSOLE,"5.01"
```

To the batch file to support Windows XP

See : https://github.com/ring-lang/ring/blob/master/language/src/buildvc.bat

121.45 Why we don't use () after the QApp class name?

When we use RingQt to create GUI application, we uses () after the class name when we create new objects for example.

```
new qWidget() { setWindowTitle("Hello World") resize(400,400) show() }
```

but before doing that we create an object from the qApp class and we don't use () after that

```
Load "guilib.ring"
app = new qApp
{
    win=new qWidget()
    {
        setwindowtitle(:test)
        show()
    }
    exec()
}
```

Using () after the class name means calling the init() method in the class and passing parameters to this method.

If we used () while no init() method in the class we get the expected error message.

The class qApp don't have this method while the other classes have it because they need it to create an object using a function that return a pointer to that object and this pointer will be stored in an attribute called pObject, for more information see ring_qt.ring file which contains the classes.

121.46 Why the window title bar is going outside the screen?

When we write the next code

```
Load "guilib.ring"
app = new qApp
{
        win=new qWidget()
        {
```

(continues on next page)

(continued from previous page)

```
                setwindowtitle(:test)
                setGeometry(0,0,200,200)
                show()
        }
        exec()
}
```

I would expect that the window will run at the point (0,0) with (200,200) size but the actual result is that the window title bar is going outside the screen.

This is related to the behavior of Qt framework.

The next code will avoid the problem

```
load "guilib.ring"
new qApp {
        new qWidget() {
                move(0,0)
                resize(200,200)
                show()
        }
        exec()
}
```

121.47 How to create an array of buttons in GUI applications?

Check the next example:

```
Load "guilib.ring"

App1 = new qApp {

        win1 = new qWidget() {
                move(0,0)
                resize(500,500)
                new qPushButton(win1)
                {
                        settext("OK")
                        setclickevent("click()")
                }
                btn1 = new qPushButton(win1)
                {
                        setgeometry(100,100,100,30)
                        settext("Button1")
                }

                btn2 = new qPushButton(win1)
                {
                        setgeometry(200,100,100,30)
                        settext("Button2")
                }
```

(continues on next page)

(continued from previous page)

```
                button = [btn1, btn2]
                show()
        }

        exec()

}

func click

        button[1] { settext ("Button3") }
        button[2] { settext ("Button4") }
```

121.48 How to Close a window then displaying another one?

This example demonstrates how to close a window and show another one

```
Load "guilib.ring"

app=new qApp
{
        frmBefore=new Qwidget()
        {
                setWindowTitle("before!")
                resize(300,320)
                move(200,200)

                button=new qPushButton(frmBefore)
                {
                        setText("Close")
                        setClickEvent("frmBefore.close() frmMain.show()")
                }

                show()
        }

        frmMain=new Qwidget()
        {
                setWindowTitle("After!")
                resize(300,320)
                move(200,200)
        }

        exec()

}
```

121.49 How to create a Modal Window?

This example demonstrates how to create a modal window

```
load "guilib.ring"
app=new qApp
{
        frmStart=new Qwidget()
        {
                setWindowTitle("The First Window")
                resize(300,320)
                move(200,200)

                button=new qPushButton(frmStart)
                {
                        setText("Show Modal Window")
                        resize(200,30)
                        setClickEvent("frmModal.show()")
                }

                new qPushButton(frmStart)
                {
                        setText("Close Window")
                        move(0,50)
                        resize(200,30)
                        setClickEvent("frmStart.Close()")
                }

                show()
        }

        frmModal =new Qwidget()
        {
                setWindowTitle("Modal Window")
                resize(300,320)
                move(200,200)
                setparent(frmStart)
                setwindowmodality(true)
                setwindowflags(Qt_Dialog)
        }

        exec()

}
```

Related Documents

- http://doc.qt.io/qt-5/qtwidgets-widgets-windowflags-example.html
- http://doc.qt.io/qt-5/qt.html#WindowType-enum
- http://doc.qt.io/qt-5/qwindow.html#setParent
- http://doc.qt.io/qt-5/qt.html#WindowModality-enum

121.50 How can I disable maximize button and resize window?

Use the method setWindowFlags()

```
Load "guilib.ring"
app1 = new qapp {
                win1 = new qwidget() {
                                setwindowtitle("First")
                                setgeometry(100,100,500,500)

                                new qpushbutton(win1) {
                                                setgeometry(100,100,100,30)
                                                settext("close")
                                                setclickevent("app1.quit()")
                                }

                                new qpushbutton(win1) {
                                                setgeometry(250,100,100,30)
                                                settext("Second")
                                                setclickevent("second()")
                                }

                                showmaximized()
                }
                exec()
}

func second
                win2 = new qwidget() {
                                setwindowtitle("Second")
                                setgeometry(100,100,500,500)
                                setwindowflags(Qt_dialog)
                                show()
                }
```

121.51 Why setClickEvent() doesn't see the object methods directly?

setClickEvent(cCode) take a string contains code. The code will be executed when the event happens.

Ring support Many Programming Paradigms like Procedural, OOP, Functional and others.

But when you support many paradigms at the language level you can't know which paradigm will be used so you have two options

(1) Provide General Solutions that works with many programming paradigms.

(2) Provide Many Specific solutions where each one match a specific paradigm.

setClickEvent() and others belong to (General Solutions that works with many programming paradigms).

You just pass a string of code that will be executed without any care about classes and objects.

This code could be anything like calling a function, calling a method and setting variable value.

Some other languages force you to use OOP and call methods for events. Also some other languages uses anonymous functions that may get parameters like the current object.

Now we have the general solution (not restricted with any paradigm), In the future we may add specific solutions that match specific paradigms (OOP, Functional, Declarative and Natural).

121.52 How to extend RingQt and add more classes?

You have many options

In general you can extend Ring using C or C++ code

Ring from Ring code you can call C Functions or use C++ Classes & Methods

This chapter in the documentation explains this part in the language https://ring-lang.github.io/doc1.16/extension.html

For example the next code in *.c file can be compiled to a DLL file using the Ring library (*.lib)

```
#include "ring.h"

RING_FUNC(ring_ringlib_dlfunc)
{
        printf("Message from dlfunc");
}

RING_API void ringlib_init(RingState *pRingState)
{
        ring_vm_funcregister("dlfunc",ring_ringlib_dlfunc);
}
```

Then from Ring you can load the DLL file using LoadLib() function then call the C function that called dlfunc() as any Ring function.

```
See "Dynamic DLL" + NL
LoadLib("ringlib.dll")
dlfunc()
```

Output

```
Dynamic DLL
Message from dlfunc
```

When you read the documentation you will know about how to get parameters like (strings, numbers, lists and objects)

And how to return a value (any type) from your function.

From experience, when we support a C library or C++ Library

We discovered that a lot of functions share a lot of code

To save our time, and to quickly generate wrappers for C/C++ Libraries to be used in Ring

We have this code generator

https://github.com/ring-lang/ring/blob/master/extensions/codegen/parsec.ring

The code generator is just a Ring program < 1200 lines of Ring code

The generator take as input a configuration file contains the C/C++ library information

like Functions Prototype, Classes and Methods, Constants, Enum, Structures and members , etc.

Then the generator will generate

*.C File for C libraries (to be able to use the library functions)

*.CPP File for C++ libraries (to be able to use C++ classes and methods)

*.Ring File (to be able to use C++ classes as Ring classes)

*.RH file (Constants)

To understand how the generator work check this extension for the Allegro game programming library

https://github.com/ring-lang/ring/tree/master/extensions/ringallegro

At first we have the configuration file

https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/allegro.cf

To write this file, i just used the Allegro documentation + the Ring code generator rules

Then after executing the generator using this batch file

https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/gencode.bat

or using this script

https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/gencode.sh

I get the generated source code file

https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/ring_allegro.c

The generated source code file (ring_allegro.c) is around 12,000 Lines of code (12 KLOC)

While the configuration file is less than 1 KLOC

To build the library (create the DLL files)

https://github.com/ring-lang/ring/blob/master/extensions/ringallegro/buildvc.bat

Also you can check this extension for the LibSDL Library

https://github.com/ring-lang/ring/tree/master/extensions/ringsdl

After this know you should know about

1 - Writing the configuration file

2 - Using the Code Generator

3 - Building your library/extension

4 - Using your library/extension from Ring code

Let us move now to you question about Qt

We have RingQt which is just an extension to ring (ringqt.dll)

You don't need to modify Ring.

(1) You just need to modify RingQt

(2) Or extend Ring with another extension based on Qt (but the same Qt version)

For the first option see the RingQt extension

https://github.com/ring-lang/ring/tree/master/extensions/ringqt

Configuration file

https://github.com/ring-lang/ring/blob/master/extensions/ringqt/classes/qt.cf

To generate the source code

https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencode.bat

https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencode.sh

https://github.com/ring-lang/ring/blob/master/extensions/ringqt/gencodeandroid.bat

To build the DLL/so/Dylib files

https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildmingw32.bat

https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildgcc.sh

https://github.com/ring-lang/ring/blob/master/extensions/ringqt/buildclang.sh

Study RingQt

Learn about the options that you have

(1) wrapping a Qt class directly

(2) Creating a new class then wrapping your new class

For the second option (in the previous two points or in the two points before that)

You will create new classes in C++ code

Then you merge these classes to RingQt or provide special DLL for them (your decision)

If your work is general (will help others) just put it to RingQt.

if your work is special (to specific application) just put it in another extension.

121.53 How to add Combobox and other elements to the cells of a QTableWidget?

Check the next code

```
Load "guilib.ring"
New qApp
{
        win1 = new qMainWindow() {
                setGeometry(100,100,1100,370)
                setwindowtitle("Using QTableWidget")

                Table1 = new qTableWidget(win1) {
                        setrowcount(10) setcolumncount(10)
                        setGeometry(0,0,800,400)
                        setselectionbehavior(QAbstractItemView_SelectRows)

                        for x = 1 to 10
                                for y = 1 to 10
                                        item1 = new qtablewidgetitem("R"+X+"C"+Y)
                                        setitem(x-1,y-1, item1)
                                next
                        next
```

(continues on next page)

(continued from previous page)

```
                        cmb = new QComboBox(Table1) {
                                alist = ["one","two","three","four","five"]
                                for x in aList additem(x,0) next
                        }
                                setCellWidget(5, 5, cmb)
                }

                setcentralwidget(table1)
                show()
        }
        exec()
}
```

121.54 How to perform some manipulations on selected cells in QTableWidget?

Check the next sample

```
Load "guilib.ring"

New qApp {
        win1 = new qMainWindow() {
                setGeometry(100,100,800,600)
                setwindowtitle("Using QTableWidget")
                Table1 = new qTableWidget(win1) {
                        setrowcount(10) setcolumncount(10)
                        setGeometry(10,10,400,400)
                        for x = 1 to 10
                                for y = 1 to 10
                                        item1 = new qtablewidgetitem("10")
                                        setitem(x-1,y-1,item1)
                                next
                        next
                }
                btn1 = new qPushButton(win1) {
                        setText("Increase")
                        setGeometry(510,10,100,30)
                        setClickEvent("pClick()")
                }
                show()
        }
        exec()
}

func pClick
        for nRow = 0 to Table1.rowcount() - 1
                for nCol = 0 to Table1.columncount() - 1
                        Table1.item(nRow,nCol)  {
                                if isSelected()
                                        setText( "" + ( 10 + text()) )
```

(continues on next page)

(continued from previous page)

```
                                ok
                        }
                next
        next
```

121.55 How to use SQLite using ODBC?

In Ring 1.1 and later versions we have native support for SQLite, so you don't need to use it through ODBC.

Also we can access SQLite through RingQt.

The answer to your question

```
pODBC = odbc_init()
odbc_connect(pODBC,"DRIVER=SQLite3 ODBC Driver;Database=mydb.db;LongNames=0;"+
                   "Timeout=1000;NoTXN=0;SyncPragma=NORMAL;StepAPI=0;")
odbc_execute(pODBC,"create table 'tel' ('ID','NAME','PHONE');")
odbc_execute(pODBC,"insert into 'tel' values ('1','Mahmoud','123456');")
odbc_execute(pODBC,"insert into 'tel' values ('2','Ahmed','123456');")
odbc_execute(pODBC,"insert into 'tel' values ('3','Ibrahim','123456');")
odbc_execute(pODBC,"select * from tel") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
        See nl
        for x = 1 to nMax
                see odbc_getdata(pODBC,x)
                if x != nMax see " - " ok
        next
end
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

Output:

```
Columns Count : 3

1 - Mahmoud - 123456
2 - Ahmed - 123456
3 - Ibrahim - 123456
```

The program will create the file : mydb.db

Note : when I print the odbc drivers I see the long list that includes

```
SQLite3 ODBC Driver - UsageCount=1
SQLite ODBC Driver - UsageCount=1
SQLite ODBC (UTF-8) Driver - UsageCount=1
```

And I'm using "SQLite3 ODBC Driver".

121.56 Can I connect to dbase/harbour database?

You can connect to any database using ODBC

To connect to xbase files (*.DBF)

```
See "Using DBF Files using ODBC" + nl
pODBC = odbc_init()
See "Connect to database" + nl
odbc_connect(pODBC,"Driver={Microsoft dBase Driver (*.dbf)};"+
                   "datasource=dBase Files;DriverID=277")
See "Select data" + nl
odbc_execute(pODBC,"select * from tel.dbf")
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
           See "Row data:" + nl
           for x = 1 to nMax
                           see odbc_getdata(pODBC,x) + " - "
           next
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

Output

```
Using DBF Files using ODBC
Connect to database
Select data
Columns Count : 3
Row data:
Ahmad - Egypt - 234567 - Row data:
Fady - Egypt - 345678 - Row data:
Shady - Egypt - 456789 - Row data:
Mahmoud - Egypt - 123456 - Close database...
```

Also you can connect to a Visual FoxPro database (requires installing Visual FoxPro driver)

```
See "ODBC test 6" + nl
pODBC = odbc_init()
See "Connect to database" + nl
odbc_connect(pODBC,"Driver={Microsoft Visual FoxPro Driver};"+
        "SourceType=DBC;SourceDB=C:\PWCT19\ssbuild\PWCTDATA\CH1\Data\mydata.dbc;")
See "Select data" + nl
see odbc_execute(pODBC,"select * from t38") + nl
nMax = odbc_colcount(pODBC)
See "Columns Count : " + nMax + nl
while odbc_fetch(pODBC)
        See "Row data:" + nl
        for x = 1 to nMax
                see odbc_getdata(pODBC,x) + " - "
        next
end
```

(continues on next page)

(continued from previous page)

```
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

CHAPTER

TWO

BUILDING FROM SOURCE CODE

The Ring programming language is a free open source product (MIT License).

You can build Ring using CMake or using Scripts (Batch Files or Shell Scripts).

The next steps explains building using scripts.

The steps demonstrates building Ring and some of the Ring extensions.

There are more extensions in ring/extensions folder like RingCJSON, RingHTTPLib, etc.

122.1 Building using Microsoft Windows

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

122.2 Complete Build (Compiler/VM, Extensions, Tools, etc.)

Install Qt 5.15.16 (in C:/Qt) so we have the folder (C:/Qt/5.15.16)

If you have another version from Qt 5.15 (like Qt 5.15.2) - Or Qt is installed in another folder The next batch files uses environment variables (RING_QT_DIR & RING_QT_VERSION) to set the path

To build 32bit version: ring/buildvc.bat To build 64bit version: ring/buildvc_x64.bat

122.3 64bit version

The next steps demonstrates how to build the 32bit version

To build the 64bit version add _x64 to the batch file name (i.e. use buildvc_x64.bat instead of buildvc.bat)

Also, use ring/bin/install_x64.bat instead of ring/bin/install.bat

122.4 Custom Build

Build Ring (Compiler/VM)

```
cd ring/language/src
buildvc.bat
buildvcw.bat
```

Build Ring2EXE

```
cd ring/tools/ring2exe
build.bat
```

Generate RingConsoleColors Source Code and Build

```
cd ring/extensions/ringconsolecolors
gencode.bat
buildvc.bat
```

Build RingInternet

```
cd ring/extensions/ringinternet
buildvc.bat
```

Generate RingLibCurl Source Code and Build

```
cd ring/extensions/ringcurl
gencode.bat
buildvc.bat
```

Generate RingZip Source Code and Build

```
cd ring/extensions/ringzip
gencode.bat
buildvc.bat
```

Build RingPM

```
cd ring/tools/ringpm
build.bat
```

Build RingREPL

```
cd ring/tools/ringrepl
build.bat
```

Build Folder2QRC

```
cd ring/tools/folder2qrc
build.bat
```

Build RingODBC

```
cd ring/extensions/ringodbc
buildvc.bat
```

Build RingMySQL

```
cd ring/extensions/ringmysql
buildvc.bat
```

Build RingSQLite

```
cd ring/extensions/ringsqlite
buildvc.bat
```

Build RingPostgreSQL

```
cd ring/extensions/ringpostgresql
gencode.bat
buildvc.bat
```

Build RingOpenSSL

```
cd ring/extensions/ringopenssl
buildvc.bat
```

Build RingMurmurHash

```
cd ring/extensions/ringmurmurhash
buildvc.bat
```

Generate RingAllegro Source Code and Build

```
cd ring/extensions/ringallegro
gencode.bat
buildvc.bat
```

Generate RingLibuv Source Code and Build

```
cd ring/extensions/ringlibuv
gencode.bat
buildvc.bat
```

Generate RingFreeGLUT Source Code and Build

```
cd ring/extensions/ringfreeglut
gencode.bat
buildvc.bat
```

Generate RingOpenGL Source Code and Build

The ringopengl folder contains many sub folders for different OpenGL versions

Starting from OpenGL 1.1 to OpenGL 4.6

```
cd ring/extensions/ringopengl/opengl21
gencode.bat
buildvc.bat
```

Generate RingQt Source Code and Build

Tested using Qt 5.15

Install Qt 5.15 : https://www.qt.io/blog/qt-5.15-released

```
cd ring/extensions/ringqt
gencode_light.bat
buildvc_light.bat
gencode_nobluetooth.bat
buildvc_nobluetooth.bat
```

To Copy the Qt runtime files to ring/bin folder

```
cd ring/extensions/ringqt/binupdate
installqt515.bat
```

To be able to call ring from any folder

```
cd ring/bin
install.bat
```

Add Ring/bin to System path

```
Hit "windows key".
Type "Edit the System environment variables"
Select "Advanced" tab.
Click on "Environment Variables..."
Double click on "Path"
Add at the end the new path separated by semicolon.
    ;C:\Ring\Bin
```

Run Ring Notepad

```
cd ring/tools/ringnotepad
ring rnote.ring
```

We can run Ring Notepad using Ring Package Manager

```
ringpm run ringnotepad
```

122.5 Building using Ubuntu Linux

This version is tested using Ubuntu 22.04 LTS

Install Git

```
sudo apt-get install git
```

Upgrade the packages

```
sudo apt-get update && sudo apt-get upgrade
```

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Install Libraries

```
cd ring/language/src
./installdep.sh
```

Build Ring (Compiler/VM)

```
sudo ./buildgcc.sh
```

Build Ring2EXE

```
cd ring/tools/ring2exe
sudo ./build.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ring/extensions/ringconsolecolors
./gencode.sh
./buildgcc.sh
```

Build RingInternet

```
cd ring/extensions/ringinternet
./buildgcc.sh
```

Generate RingLibCurl Source Code and Build

```
cd ring/extensions/ringcurl
./gencode.sh
./buildgcc.sh
```

Generate RingZip Source Code and Build

```
cd ring/extensions/ringzip
./gencode.sh
./buildgcc.sh
```

Build RingPM

```
cd ring/tools/ringpm
sudo ./build.sh
```

Build RingREPL

```
cd ring/tools/ringrepl
sudo ./build.sh
```

Build Folder2QRC

```
cd ring/tools/folder2qrc
sudo ./build.sh
```

Build RingODBC

```
cd ring/extensions/ringodbc
./buildgcc.sh
```

Build RingMySQL

```
cd ring/extensions/ringmysql
./buildgcc.sh
```

Build RingSQLite

```
cd ring/extensions/ringsqlite
./buildgcc.sh
```

Build RingPostgreSQL

```
cd ring/extensions/ringpostgresql
gencode.sh
buildgcc.sh
```

Build RingOpenSSL

```
cd ring/extensions/ringopenssl
./buildgcc.sh
```

Build RingMurmurHash

```
cd ring/extensions/ringmurmurhash
./buildgcc.sh
```

Generate RingAllegro Source Code and Build

```
cd ring/extensions/ringallegro
./gencode.sh
./buildgcc.sh
```

Generate RingLibuv Source Code and Build

We will build Libuv first

```
cd ring/extensions/ringlibuv/libuv
sudo apt-get install libtool m4 automake
sh autogen.sh
./configure
make
make check
sudo make install
```

Then we will build RingLibuv

```
cd ring/extensions/ringlibuv
./gencode.sh
./buildgcc.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ring/extensions/ringfreeglut
./gencode.sh
./buildgcc.sh
```

Generate RingOpenGL Source Code and Build

The ringopengl folder contains many sub folders for different OpenGL versions

Starting from OpenGL 1.1 to OpenGL 4.6

```
cd ring/extensions/ringopengl/opengl21
gencode.sh
buildgcc.sh
```

Generate RingQt Source Code and Build

```
cd ring/extensions/ringqt
./gencode_light.sh
./buildgcc_light.sh
./gencode.sh
./buildgcc.sh
```

To be able to call ring from any folder

```
cd ring/bin
sudo ./install.sh
```

Run Ring Notepad

```
cd ring/tools/ringnotepad
ring rnote.ring
```

We can run Ring Notepad using Ring Package Manager

```
ringpm run ringnotepad
```

122.6 Building using MacOS X

This version is tested using macOS Catalina (version 10.15)

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Install homebrew (follow the directions on homebrew's homepage). Install Libraries

```
cd ring/language/src
./installdepmac.sh
```

Build Ring (Compiler/VM)

```
./buildclang.sh
```

Build Ring2EXE

```
cd ring/tools/ring2exe
sudo ./build.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ring/extensions/ringconsolecolors
./gencode.sh
./buildclang.sh
```

Build RingInternet

```
cd ring/extensions/ringinternet
./buildclang.sh
```

Generate RingLibCurl Source Code and Build

```
cd ring/extensions/ringcurl
./gencode.sh
./buildclang.sh
```

Generate RingZip Source Code and Build

```
cd ring/extensions/ringzip
./gencode.sh
./buildclang.sh
```

Build RingPM

```
cd ring/tools/ringpm
sudo ./build.sh
```

Build RingREPL

```
cd ring/tools/ringrepl
sudo ./build.sh
```

Build Folder2QRC

```
cd ring/tools/folder2qrc
sudo ./build.sh
```

Build RingODBC

```
cd ring/extensions/ringodbc
./buildclang.sh
```

Build RingMySQL

```
cd ring/extensions/ringmysql
./buildclang.sh
```

Build RingSQLite

```
cd ring/extensions/ringsqlite
./buildclang.sh
```

Build RingPostgreSQL

```
cd ring/extensions/ringpostgresql
gencode.sh
buildclang.sh
```

Build RingOpenSSL

```
cd ring/extensions/ringopenssl
./buildclang.sh
```

Build RingMurmurHash

```
cd ring/extensions/ringmurmurhash
./buildclang.sh
```

Generate RingAllegro Source Code and Build

```
cd ring/extensions/ringallegro
./gencode.sh
./buildclang.sh
```

Generate RingLibuv Source Code and Build

```
cd ring/extensions/ringlibuv
./gencode.sh
./buildclang.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ring/extensions/ringfreeglut
./gencode.sh
./buildclang.sh
```

Generate RingOpenGL Source Code and Build

The ringopengl folder contains many sub folders for different OpenGL versions Starting from OpenGL 1.1 to OpenGL 4.6

```
cd ring/extensions/ringopengl/opengl21
./gencode.sh
./buildclang.sh
```

Generate RingQt Source Code and Build

```
cd ring/extensions/ringqt
./gencode_light.sh
./buildclang_light.sh
./gencode.sh
./buildclang.sh
```

To be able to call ring from any folder

```
cd ring/bin
sudo ./install.sh
```

Run Ring Notepad

```
cd ring/tools/ringnotepad
sudo ring rnote.ring
```

We can run Ring Notepad using Ring Package Manager

```
ringpm run ringnotepad
```

122.7 Building using CMake

This will build the Ring compiler and Ring Virtual Machine

```
cmake .
make
```

CHAPTER

THREE

HOW TO CONTRIBUTE?

Ring is a free-open source project, Everyone is welcome to contribute to Ring.

Project Home : https://github.com/ring-lang/ring

To editing on web browser without Git client, when login GitHub then click pencil icon in target file. Then, sends pull request.

You can help in many parts in the project

- Documentation
- Testing
- Samples
- Applications
- Editors Support
- Libraries in Ring
- Extensions in C/C++
- Compiler and Virtual Machine (VM)
- Ideas and suggestions

123.1 Special thanks to contributors

Throughout the creation of this project, Ring relied heavily on contributions from experts along with college students. Their input was invaluable, and we want to take a moment to thank them and recognize them for all of their hard work.

Ring Team: https://ring-lang.github.io/team.html

123.2 Documentation

You can modify anything in the documentation, by updating the text files (*.txt) in this folder : https://github.com/ring-lang/ring/tree/master/documents/source

The documentation is created using Sphinx : http://www.sphinx-doc.org/en/stable/

123.3 Testing

You can write new tests in this folder

https://github.com/ring-lang/ring/tree/master/language/tests/scripts

123.4 Samples

You can add new samples to this folder

https://github.com/ring-lang/ring/tree/master/samples

123.5 Applications

You can add new applications to this folder

https://github.com/ring-lang/ring/tree/master/applications

123.6 Editors Support

You can help in supporting Ring in different code editors

Check the next folder

https://github.com/ring-lang/ring/tree/master/tools/editors

123.7 Libraries in Ring

You can update and add libraries to this folder

https://github.com/ring-lang/ring/tree/master/libraries

123.8 Extensions in C/C++

You can add and update extensions in this folder

https://github.com/ring-lang/ring/tree/master/extensions

123.9 Compiler and Virtual Machine (VM)

- Source Code (C Language) : https://github.com/ring-lang/ring/tree/master/language/src
- Visual Source (PWCT) : https://github.com/ring-lang/ring/tree/master/language/visualsrc

CHAPTER

FOUR

LANGUAGE SPECIFICATION

In this chapter we will learn about

- Language keywords
- Language Functions
- Compiler Errors
- Runtime Errors
- Environment Errors
- Language Grammar
- Virtual Machine (VM) Instructions

124.1 Language Keywords

Keywords Count : 58

- again
- and
- but
- bye
- call
- case
- catch
- changeringkeyword
- changeringoperator
- class
- def
- do
- done
- else
- elseif
- end

- exit
- for
- from
- func
- get
- give
- if
- import
- in
- load
- loadsyntax
- loop
- new
- next
- not
- off
- ok
- on
- or
- other
- package
- private
- put
- return
- see
- step
- switch
- to
- try
- while
- endfunc
- endclass
- endpackage
- endif
- endfor
- endwhile

- endswitch
- endtry
- function
- endfunction
- break
- continue

124.2 Language Functions

Functions Count : 253

```
acos(x) ---> The principal value of the arc cosine of x, expressed in radians
add(List,Item)
addattribute(Object,cAttributeName|aAttributesList)
adddays(cDate,nDays) ---> Date from cDate and after nDays
addmethod(Object,cNewMethodName,cMethodName|AnonymousFunction)
addsublistsbyfastcopy(lFlag)
addsublistsbymove(lFlag)
ascii(character) ---> ASCII Code
asin(x) ---> The principal value of the arc sine of x, expressed in radians
assert(condition)
atan(x) ---> The principal value of the arc tangent of x, expressed in radians
atan2(y,x) ---> The principal arc tangent of y/x, in the interval [-pi,+pi] radians
attributes(object) ---> Returns a list contains the object attributes
binarysearch(List,ItemValue) ---> Item Index
binarysearch(List,ItemValue,nColumn) ---> Search in nColumn, returns the Item Index
bytes2double(cBytes) ---> nNumber
bytes2float(cBytes) ---> nNumber
bytes2int(cBytes) ---> nNumber
callgarbagecollector()
callgc()
ceil(x) ---> The smallest integer value greater than or equal to x
cfunctions() ---> a list contains functions names
char(ASCII Code) ---> character
chdir(cNewPath)
checkoverflow(lFlag)
classes() ---> a list contains classes names
classname(object) ---> Returns the object class name
clearerr(FileHandle)
clock() ---> The number of clock ticks from program start
clockspersecond() ---> Number of clocks in one second
closelib(pDLL)
copy(string,nCount) ---> string replicated nCount times
cos(x) ---> The cosine of an angle of x radians
cosh(x) ---> The hyperbolic cosine of x radians
currentdir() ---> String contains the path of the current directory
date() ---> String represent the date "dd/mm/yyyy"
dec(hexadecimal) ---> decimal
decimals(n) ---> Determine the decimals digits after the point in float/double numbers
```

(continues on next page)

(continued from previous page)

```
del(list,index)
diffdays(cDate1,cDate2) ---> number of days (Date1 - Date2)
dir(cFolderPath) ---> List contains files & sub folders.
direxists(cDirPath) ---> returns 1 if the directory exists
double2bytes(nNumber) ---> cBytes
eval(cCode)
exefilename() ---> String contains the Ring executable file name
exefolder() ---> String contains the Ring executable path
exp(x) ---> The value of e raised to the xth power
fabs(x) ---> The absolute value of x
fclose(FileHandle)
feof(FileHandle) ---> returns 1 if EOF and 0 if not
ferror(FileHandle) ---> returns 1 if error and 0 if not
fexists(cFileName) ---> returns 1 if the file exists
fflush(FileHandle)
fgetc(FileHandle) ---> returns character or EOF
fgetpos(FileHandle) ---> position handle
fgets(FileHandle,nSize) ---> string
filename() ---> String contains the active source file name
find(List,ItemValue) ---> Item Index
find(List,ItemValue,nColumn) ---> Search in nColumn, returns the Item Index
find(List,ItemValue,nColumn,cAttribute) ---> Item Index
float2bytes(nNumber) ---> cBytes
floor(x) ---> The largest integer value less than or equal to x
fopen(cFileName,cMode) ---> FileHandle
fputc(FileHandle,cChar)
fputs(FileHandle,cString)
fread(FileHandle,nSize)
freopen(cFileName,cMode,file handle) ---> FileHandle
fseek(FileHandle,nOffset,nWhence) ---> zero if successful
fsetpos(FileHandle,PositionHandle)
ftell(FileHandle) ---> file position as number
functions() ---> a list contains functions names
fwrite(FileHandle,cString)
getarch() ---> cString (The name of the architecture of the Ring executable)
getattribute(oObject,cAttributeName) ---> Attribute Value
getchar() ---> Character
getfilesize(cFilePath) ---> nSize
getnumber() ---> Number
getpathtype(cPath) ---> nStatus
getpointer(pointer) ---> nAddress
getptr(pointer) ---> nAddress
getstring() ---> String
globals() ---> a list contains variables names in the global scope
hex(decimal) ---> hexadecimal
hex2str(Hexadecimal string) ---> string
importpackage(cPackageName)
input(nCount) ---> string
insert(List,Index,Item)
int2bytes(nNumber) ---> cBytes
intvalue(cVariableName)
isalnum(value) ---> 1 if the value is digit/letter or 0 if not
```

(continues on next page)

(continued from previous page)

```
isalpha(value) ---> 1 if the value is a letter or 0 if not
isandroid() ---> Returns 1 if the operating system is Android, Returns 0 if it's not
isattribute(object,cAttributeName) ---> Bool
iscfunction(cFunctionName) ---> returns 1 if the C function is defined
isclass(cClassName) ---> returns 1 if the Class is defined
iscntrl(value) ---> 1 if the value is a control character or 0 if not
isdigit(value) ---> 1 if the value is a digit or 0 if not
isfreebsd() ---> Returns 1 if the operating system is FreeBSD, Returns 0 if it's not
isfunction(cFunctionName) ---> returns 1 if the Ring function is defined
isglobal(cVariableName) ---> returns 1 if the variable is defined in the global scope
isgraph(value) ---> 1 if the value can be printed (Except space) or 0 if not
islinux() ---> Returns 1 if the operating system is Linux, Returns 0 if it's not
islist(value) ---> 1 if the value is a list or 0 if not
islocal(cVariableName) ---> returns 1 if the variable is defined in the local scope
islower(value) ---> 1 if the value is lowercase letter or 0 if not
ismacosx() ---> Returns 1 if the operating system is Mac OS X, Returns 0 if it's not
ismethod(object,cMethodName) ---> Returns True if the object class contains the method
ismsdos() ---> Returns 1 if the operating system is MS-DOS, Returns 0 if it's not
isnull(value) ---> 1 if the value is NULL or 0 if not
isnumber(value) ---> 1 if the value is a number or 0 if not
isobject(variable) ---> Returns True if it's an object, False if it's not
ispackage(cPackageName) ---> returns 1 if the Package is defined
ispackageclass(cPackageName,cClassName) ---> returns 1 if the Class is defined
ispointer(vPara) ---> True|False
isprint(value) ---> 1 if the value occupies a printing position or 0 if not
isprivateattribute(object,cAttributeName) ---> lResult
isprivatemethod(object,cMethodName) ---> lResult
ispunct(value) ---> 1 if the value is a punctuation character or 0 if not
isspace(value) ---> 1 if the value is a white-space or 0 if not
isstring(value) ---> 1 if the value is a string or 0 if not
isunix() ---> Returns 1 if the operating system is Unix, Returns 0 if it's not
isupper(value) ---> 1 if the value is an uppercase alphabetic letter or 0 if not
iswindows() ---> Returns 1 if the operating system is Windows
iswindows64() ---> Returns 1 if the operating system is Windows64
isxdigit(value) ---> 1 if the value is a hexadecimal digit character or 0 if not
left(string,count) ---> Get characters starting from the left
len(string) ---> String length
len(List) ---> The list size
lines(string) ---> Number of lines inside the string
list(nSize) ---> aList
list(nRows,nCols) ---> aList
list2str(list) ---> string contains the list items
loadlib(cDLLFileName) ---> pDLL
locals() ---> a list contains the variables names in the current scope
log(x) ---> The natural logarithm of x (the base of e)
log(x,b) ---> The logarithm of x to the base of b
log10(x) ---> The common logarithm (base-10 logarithm) of x
lower(string) ---> convert string letters to lower c
max(nNumber1,nNumber) ---> Maximum number
max(aList) ---> Maximum number inside the list
memcpy(pDestinationPointer,cSourceString,nSize)
memorycopy(pDestinationPointer,cSourceString,nSize)
```

(continues on next page)

(continued from previous page)

```
mergemethods(cClassNameDestination,cClassNameSource)
methods(object) ---> Returns a list contains the object methods
min(nNumber,nNumber2) ---> Minimum number
min(aList) ---> Minimum number inside the list
murmur3hash(cString,nNumber) ---> nNumber
newlist(nSize) ---> aList
newlist(nRows,nCols) ---> aList
nofprocessors() ---> nProcessors
nothing(Any number/type of parameters) ---> Zero (0)
nullpointer() ---> pPointer
nullptr() ---> pPointer
number(string) ---> Number
obj2ptr(List|Object) --> Low Level Object ( C Pointer )
object2pointer(List|Object) --> Low Level Object ( C Pointer )
objectid(object) ---> Returns the object id
optionalfunc(cFuncName)
packageclasses() cPackageName) ---> a list contains classes names inside the package
packagename() ---> Returns the package name of the latest successful import
packages() ---> a list contains packages names
perror(cErrorMessage)
pointer2object(Low Level Object) ---> List|Object
pointer2string(pointer,nStart,nCount) ---> cString
pointercompare(oObject1,oObject2) ---> lResult
pow(x,y) ---> x raised to the power of y
prevfilename() ---> String contains the previous source file name.
print(cString)
print2str(cString) ---> String
ptr2obj(Low Level Object) ---> List|Object
ptr2str(pointer,nStart,nCount) ---> cString
ptrcmp(oObject1,oObject2) ---> lResult
puts(cString)
raise(cErrorMessage)
random(x) ---> A random number in the range [0,x]
randomize(nNumber) ---> nNumber
read(cFileName) ---> String contains the file content
ref(aList|oObject) ---> List/Object reference
reference(aList|oObject) ---> List/Object reference
refcount(aList|oObject) ---> References Count
remove(cFileName)
rename(cOldFileName,cNewFileName) ---> Zero for Success or -1 for Error
reverse(List) ---> Reversed List
rewind(FileHandle)
right(string,count) ---> get characters starting from the right
ring_give(cVariable)
ring_see(cMessage)
ring_state_delete(oState)
ring_state_filetokens(oState,cRingFileName) ---> aTokens
ring_state_findvar(oState,cVariableName) ---> aVariableList
ring_state_init() ---> oState
ring_state_main(cRingFileName)
ring_state_mainfile(oState,cRingFileName)
ring_state_new() ---> oState
```

(continues on next page)

(continued from previous page)

```
ring_state_newvar(oState,cVariableName) ---> aVariableList
ring_state_runcode(oState,cCode)
ring_state_runfile(oState,cRingFileName)
ring_state_runobjectfile(oState,cRingObjectFileName)
ring_state_scannererror(oState) ---> nStatus
ring_state_setvar(oState,cVariableName,Value)
ring_state_stringtokens(oState,cRingFileName)
ring_state_stringtokens(oState,cRingFileName,lCaseSensitive)
ringvm_callfunc(cFuncName)
ringvm_calllist() ---> List
ringvm_cfunctionslist() ---> List
ringvm_classeslist() ---> List
ringvm_codelist() ---> List
ringvm_evalinscope(nScope,cCode)
ringvm_fileslist() ---> List
ringvm_functionslist() ---> List
ringvm_genarray(aList)
ringvm_give(cVariableName)
ringvm_hideerrormsg(lStatus)
ringvm_info() ---> List of information about the VM structure
ringvm_ismempool() ---> lStatus (Can provide memory or not)
ringvm_memorylist() ---> List
ringvm_packageslist() ---> List
ringvm_passerror()
ringvm_runcode(cCode)
ringvm_scopescount() ---> nScopes
ringvm_see(cMessage)
ringvm_settrace(cCode)
ringvm_tracedata() ---> aDataList
ringvm_traceevent() ---> nTraceEvent
ringvm_tracefunc() ---> cCode
setattribute(oObject,cAttributeName,Value)
setpointer(pointer,nNewAddress)
setptr(pointer,nNewAddress)
shutdown(nStatus)
sin(x) ---> The sine of an angle of x radians
sinh(x) ---> The hyperbolic sine of x radians
sort(List) ---> Sorted List
sort(List,nColumn) ---> Sorted List based on nColumn
sort(List,nColumn,cAttribute) ---> Sorted List based on Object Attribute
space(nBytesCount) ---> String
sqrt(x) ---> The square root of x
srandom(x) ---> Initialize random number generator
str2hex(string) ---> hexadecimal string
str2hexcstyle(string) ---> hexadecimal string
str2list(string) ---> list contains the string lines
strcmp(cString1,cString2) ---> value = 0 if cString1 = cString2
string(number) ---> String
substr(string,substring) ---> the starting position of substring in string
substr(string,position)  ---> Get substring starting from position to end
substr(string,position,count)  ---> Get characters starting from position
substr(string,substring,newsubstring)  ---> Transformed string (Match case)
```

(continues on next page)

(continued from previous page)

```
substr(string,substring,newsubstring,1) ---> Transformed string (Ignore case)
swap(aList,nItem1,nItem2)
sysget(cVariable)
sysset(cVariable,cValue) ---> Returns 1 for success and return 0 for failure
system(cCommand)
sysunset(cVariable) ---> Returns 1 for success and return 0 for failure
tan(x) ---> Tangent of an angle of x radians
tanh(x) ---> The hyperbolic tangent of x radians
tempfile() ---> FileHandle
tempname() ---> generated file name as string
time() ---> The system time as String
timelist() ---> List contains the time and date information.
trim(string) ---> Remove spaces from right and left
type(value) ---> The Type as String
ungetc(FileHandle,character)
unsigned(nNum1,nNum2,cOperator) ---> Perform operation using unsigned numbers
upper(string) ---> convert string letters to UPPER case
uptime() ---> nTime
variablepointer(cVariableName,cPointerType) ---> Low Level Object (C Pointer)
varptr(cVariableName,cPointerType) ---> Low Level Object (C Pointer)
version() ---> String contains the Ring version
windowsnl() ---> Returns a string contains CR+LF = CHAR(13) + CHAR(10)
write(cFileName,cString)
```

124.3 Compiler Errors

- Error (C1) : Error in parameters list, expected identifier
- Error (C2) : Error in class name
- Error (C3) : Unclosed control structure, 'ok' is missing
- Error (C4) : Unclosed control structure, 'end' is missing
- Error (C5) : Unclosed control structure, next is missing
- Error (C6) : Error in function name
- Error (C7) : Error in list items
- Error (C8) : Parentheses ')' is missing
- Error (C9) : Brackets ']' is missing
- Error (C10) : Error in parent class name
- Error (C11) : Error in expression operator
- Error (C12) : No class definition
- Error (C13) : Error in variable name
- Error (C14) : Try/Catch miss the Catch keyword!
- Error (C15) : Try/Catch miss the Done keyword!
- Error (C16) : Error in Switch statement expression!
- Error (C17) : Switch statement without OFF

- Error (C18) : Missing closing brace for the block opened!
- Error (C19) : Numeric Overflow!
- Error (C20) : Error in package name
- Error (C21) : Unclosed control structure, 'again' is missing
- Error (C22) : Function redefinition, function is already defined!
- Error (C23) : Using '(' after number!
- Error (C24) : The parent class name is identical to the subclass name
- Error (C25) : Trying to access the self reference after the object name
- Error (C26) : Class redefinition, class is already defined!
- Error (C27) : Syntax Error!
- Error (C28) : Expression is expected!
- Error (C29) : Braces are missing to define anonymous function!
- Error (S1) : Literal not closed!
- Warning (W1) : Duplication in file name
- Warning (W2) : The Compiler command ChangeRingKeyword requires two words as parameters
- Warning (W3) : Compiler command ChangeRingKeyword - Keyword not found !
- Warning (W4) : The Compiler command ChangeRingOperator requires two words as parameters
- Warning (W5) : Compiler command ChangeRingOperator - Operator not found !
- Warning (W6) : Using the EXIT command outside loop!"
- Warning (W7) : Using the LOOP command outside loop!"

124.4 Runtime Errors

- Error (R1) : Can't divide by zero
- Error (R2) : Array Access (Index out of range)
- Error (R3) : Calling Function without definition
- Error (R4) : Stack Overflow
- Error (R5) : Can't access the list item, Object is not list
- Error (R6) : Variable is required
- Error (R7) : Can't assign to a string letter more than one character
- Error (R8) : Variable is not a string
- Error (R9) : Using exit command outside loops
- Error (R10) : Using exit command with number outside the range
- Error (R11) : Error in class name, class not found
- Error (R12) : Error in property name, property not found
- Error (R13) : Object is required

- Error (R14) : Calling Method without definition
- Error (R15) : Error in parent class name, class not found
- Error (R16) : Using braces to access unknown object
- Error (R17) : Error, using 'Super' without parent class
- Error (R18) : Numeric Overflow
- Error (R19) : Calling function with less number of parameters
- Error (R20) : Calling function with extra number of parameters
- Error (R21) : Using operator with values of incorrect type
- Error (R22) : Using loop command outside loops
- Error (R23) : Using loop command with number outside the range
- Error (R24) : Using uninitialized variable
- Error (R25) : Error in package name, Package not found
- Error (R26) : Calling private method from outside the class
- Error (R27) : Using private attribute from outside the class
- Error (R28) : Using bad data type as step value
- Error (R29) : Using bad data type in for loop
- Error (R30) : Parent class name is identical to child class name
- Error (R31) : Trying to destroy the object using the self reference
- Error (R32) : The CALL command expect a variable contains string
- Error (R33) : Bad decimals number (correct range >= 0 and <= 90)
- Error (R34) : Variable is required for the assignment operation
- Error (R35) : Can't create/open the file
- Error (R36) : The column number is not correct! It's greater than the number of columns in the list
- Error (R37) : Sorry, The command is not supported in this context
- Error (R38) : Runtime Error in loading the dynamic library
- Error (R39) : Error occurred creating unique filename
- Error (R40) : Numeric underflow
- Error (R41) : Invalid numeric string
- Error (R42) : Error in eval() function
- Error (R43) : The variable contains a protected value
- Error (R44) : Maximum nested Eval()
- Error (R45) : Variable is not a list

124.5 Environment Errors

- Error (E1) : Caught SegFault
- Error (E2) : Out of Memory
- Error (E3) : Deleting scope while no scope!
- Error (E4) : Long VM Instruction!
- Error (E5) : The file type is not correct - the VM expect a ring object file
- Error (E6) : The Ring Object File version is not correct!
- Error (E7) : Internal error in using sscanf() function!
- Error (E8) : Internal error in using fscanf() function!

124.6 Language Grammar

Program —> {statement}

Statement —> 'package' <Identifier> { '.' <Identifier> } ['{' {statement} '}'] ['end'|'endpackage']

Statement —> 'class' <Identifier> ['from'|':'|'<' <Identifier>] ['{' {statement} '}'['end'|'endclass']

Statement —> 'func'|'def'|'function' <Identifier> [ParaList] ['{' {statement} '}']['end'|'endfunc'|'endfunction']

Statement —> 'import' <Identifier> { '.' <Identifier> }

Statement —> 'private'

Statement —> 'load' ['package'|'again'] <Literal>

Statement —> 'loadsyntax' <Literal>

Statement —> 'changeringkeyword' <OldKeyword> <NewKeyword>

Statement —> 'changeringoperator' <OldOperator> <NewOperator>

Statement —> 'see'|'put' <Expr>

Statement —> 'give'|'get' <Identifier>

Statement —> 'if' <Expr> ['{'] {statement} [{'but'|'elseif' <Expr> {Statement} }] ['else' {Statement}] 'ok'|'end'|'}'|'endif'

Statement —> 'Switch' <Expr> ['{'] { 'on'|'case' <Expr> {statement} } ['other' {Statement}] 'off'|'end'|'}'|'endswitch'

Statement —> 'for' <Identifier> '=' <Expr> 'to' <Expr> ['step' <Expr>] ['{'] {Statement} 'next'|'end'|'}'|'endfor'

Statement —> 'for' <Identifier> 'in' <Expr> ['step' <Expr>] ['{'] {statement} 'next'|'end'|'}'|'endfor'

Statement —> 'while' <Expr> ['{'] {statement} 'end'|'}'|'endwhile'

Statement —> 'do' {statement} 'again' <Expr>

Statement —> 'try' {statement} ['{'] 'catch' {statement} 'done'|'end'|'}'|'endtry'

Statement —> 'return' <Expr>

Statement —> 'bye'

Statement —> 'exit'|'break'

Statement —> ‘loop’|’continue’

Statement —> <Expr>

Statement —> epsilon

ParaList —> epsilon

ParaList —> [‘(’] <Identifier> [{ ‘,’ <Identifier> }] [‘)’]

Expr —> <LogicNot> [{ ‘and’|’or’ <LogicNot> }]

LogicNot –> [‘not’] <EqualOrNot>

EqualOrNot –> [‘=’|’!=’] <Compare>

Compare —> <BitOrXor> [{ ‘<’ | ‘>’ | ‘<=’ | ‘>=’ <BitOrXor> }]

BitOrXor —> <BitAnd> [{ ‘|’ | ‘^’ <BitAnd> }]

BitAnd —> <BitShift> [{ ‘&’ <BitShift> }]

BitShift —> <Arithmetic> [{ ‘<<’ | ‘>>’ <Arithmetic> }]

Arithmetic —> <Term> [{ ‘+’ | ‘-’ <Term> }]

Term —> <Range> [{ ‘*’ | ‘/’ | ‘%’ | ‘**’ | ‘^^’ <Range> }]

Range —> <Factor> [‘:’ <Factor>]

Factor —> <Identifier> [{Mixer}] [‘=’ <Expr>]

Factor —> <Number>

Factor —> <Literal>

Factor —> ‘:’ <Identifier>

Factor —> ‘-’ <Factor>

Factor —> ‘(’ <Expr> ‘)’

Factor —> <List>

Factor —> ‘new’ [‘from’] <Identifier>

Factor —> <AnonymousFunction>

Factor —> ‘call’ <identifier> { ‘.’ <Identifier> } ‘(’ <Parameters> ‘)’

List —> ‘[’ [<Expr> { ‘,’ <Expr> }] ‘]’

Mixer —> { ‘.’ <Identifier> }

Mixer —> ‘[’ <Expr> ‘]’

Mixer —> ‘(’ [<Expr> [{ ‘,’ <Expr> }]] ‘)’

Mixer —> ‘{’ {Statement} ‘}’

AnonymousFunction —> ‘func’|’def’|’function’ [<ParaList>] ‘{’ {Statement} ‘}’

124.7 Virtual Machine (VM) Instructions

Definitions :-

- VM : Virtual Machine
- Stack : VM Stack
- IR : Instruction Register
- PC : Program Counter
- VP : Variable Pointer
- Stack[nSize] : Last Item in the Stack (Last In - First Out)
- VV : Variable Value (We have a Pointer to a variable, And we access this variable value)

(Stack and Variables)

Operation	Description
ICO_PUSHC	Add string from the IR to the stack
ICO_PUSHN	Add number from the IR to the stack
ICO_PUSH2N	Add two numbers from the IR to the stack
ICO_PUSH3N	Add three numbers from the IR to the stack
ICO_PUSH4N	Add four numbers from the IR to the stack
ICO_PUSHV	Replace VP in the stack[nSize] with the variable value
ICO_LOADADDRESS	Read variable name from the IR, push VP to the stack
ICO_ASSIGNMENT	Stack[nSize-1] VV = Stack[nSize] VV , POP Stack[nSize]
ICO_INC	Increment Number in Stack[nSize] by 1
ICO_LOADAPUSHV	The same as ICO_LOADADDRESS then ICO_PUSHV
ICO_NEWLINE	Store new line number (debug info)
ICO_FREESTACK	Remove all items from the stack , nSize = 0
ICO_FILENAME	Store the source code file name (debug info)
ICO_FREELOADASCOPE	Free the Scope List of the current Expression

(Jump)

Operation	Description
ICO_JUMP	Set PC to new value from the IR
ICO_JUMPZERO	If Stack[nSize] is a number = 0 then Set PC to new value from the IR
ICO_JUMPFOR	End of for loop
ICO_JUMPONE	If Stack[nSize] is a number = 1 then Set PC to new value from the IR
ICO_JUMPZERO2	As ICO_JUMPZERO but add 1 to the stack (required for many 'AND' conditions)
ICO_JUMPONE2	As ICO_JUMPONE but add 1 to the stack (required for many 'OR' conditions)

(Compare)

Operation	Description
ICO_LESSEQUAL	If stack[nSize-1] <= stack[nSize] , POP stack[nSize], set Stack[nSize-1] = 1 else set Stack[nSize-1] = 0
ICO_EQUAL	If stack[nSize-1] = stack[nSize] , POP stack[nSize], set Stack[nSize-1] = 1 else set Stack[nSize-1] = 0
ICO_LESS	If stack[nSize-1] < stack[nSize] , POP stack[nSize], set Stack[nSize-1] = 1 else set Stack[nSize-1] = 0
ICO_GREATER	If stack[nSize-1] > stack[nSize] , POP stack[nSize], set Stack[nSize-1] = 1 else set Stack[nSize-1] = 0
ICO_GREATEREQUA	If stack[nSize-1] >= stack[nSize] , POP stack[nSize], set Stack[nSize-1] = 1 else set Stack[nSize-1] = 0
ICO_NOTEQUAL	If stack[nSize-1] != stack[nSize] , POP stack[nSize], set Stack[nSize-1] = 1 else set Stack[nSize-1] = 0

(Math)

Operation	Description
ICO_SUM	Stack[nSize-1] = Stack[nSize-1] + Stack[nSize] , POP stack[nSize]
ICO_SUB	Stack[nSize-1] = Stack[nSize-1] - Stack[nSize] , POP stack[nSize]
ICO_MUL	Stack[nSize-1] = Stack[nSize-1] * Stack[nSize] , POP stack[nSize]
ICO_DIV	Stack[nSize-1] = Stack[nSize-1] / Stack[nSize] , POP stack[nSize]
ICO_MOD	Stack[nSize-1] = Stack[nSize-1] % Stack[nSize] , POP stack[nSize]
ICO_POW	Stack[nSize-1] = Stack[nSize-1] ^ Stack[nSize] , POP stack[nSize]
ICO_NEG	Stack[nSize] = - Stack[nSize-1]
ICO_PLUSPLUS	Stack[nSize] = Stack[nSize] + 1
ICO_MINUSMINUS	Stack[nSize] = Stack[nSize] - 1

(Logic)

Operation	Description
ICO_AND	Stack[nSize-1] = Stack[nSize-1] && Stack[nSize] , POP stack[nSize]
ICO_OR	Stack[nSize-1] = Stack[nSize-1] || Stack[nSize] , POP stack[nSize]
ICO_NOT	Stack[nSize] = ! Stack[nSize]

(Lists)

Operation	Description
ICO_LISTSTART	Start New List in Temp. Memory
ICO_LISTITEM	Add List Item
ICO_LISTEND	End List
ICO_LOADINDEXADDRESS	Stack[nSize-1] = Stack[nSize-1] VV [Stack[nSize]] , POP stack[nSize]

(Functions)

Operation	Description
ICO_LOADFUNC	Find function
ICO_CALL	Call function
ICO_RETURN	Return from function
ICO_RETNULL	Return NULL from function
ICO_RETFROMEVAL	Return after eval()
ICO_RETITEMREF	Return the list item reference - not the value
ICO_NEWFUNC	Start new function
ICO_BLOCKFLAG	Flag to determine where to jump later (after ICO_RETURN)
ICO_FUNCEXE	Start executing function
ICO_ENDFUNCEXE	End function execution
ICO_ANONYMOUS	Anonymous function
ICO_FREETEMPLISTS	Delete temp. lists

(User Interface)

Operation	Description
ICO_PRINT	Print value to the standard output
ICO_GIVE	Get input from the keyboard

(End Program/Loop)

Operation	Description
ICO_BYE	End execution of VM
ICO_EXITMARK	Place to exit to from a loop
ICO_POPEXITMARK	Remove exit mark
ICO_EXIT	Break from one loop or more
ICO_LOOP	Continue to next loop

(For Better Performance)

Operation	Description
ICO_PUSHP	Push pointer to the stack
ICO_INCP	Increment variable value using pointer
ICO_PUSHPV	Push value of variable using variable pointer
ICO_INCJUMP	Increment then jump
ICO_INCPJUMP	Increment using pointer then jump
ICO_LOADFUNCP	Push function pointer
ICO_PUSHPLOCAL	Push pointer to local variable
ICO_INCLPJUMP	Increment value using pointer to local variable then jump
ICO_INCPJUMPSTEP1	Increment value using variable pointer then jump (for loop step = 1)
ICO_LEN	Remove list from the stack and push the list size

(Try-Catch-Done)

Operation	Description
ICO_TRY	Start try region
ICO_DONE	End try region

(Duplicate and Range)

Operation	Description
ICO_DUPLICATE	Duplicate stack value
ICO_RANGE	Create list from value to value

(OOP)

Operation	Description
ICO_NEWOBJ	Create new object, get class name from the IR, push object pointer to the stack.
ICO_SETSCOPE	Called after creating new object, set the active scope to be the object scope.
ICO_LOADSUBADDRESS	Get object attribute, push the pointer to the stack.
ICO_LOADMETHOD	Find object method
ICO_AFTERCALLMETHOD	Used after calling a method - normal case
ICO_AFTERCALLMETHOD2	Used after calling a method - second case
ICO_NEWCLASS	Start new class region
ICO_BRACESTART	Open brace
ICO_BRACEEND	End brace
ICO_IMPORT	Import package
ICO_PRIVATE	start private attributes region
ICO_SETPROPERTY	set attribute value - check for setter.
ICO_CALLCLASSINIT	call class init() method.

(Other)

Operation	Description
ICO_SETREFERENCE	Copy by reference
ICO_KILLREFERENCE	Remove reference
ICO_ASSIGNMENTPOINTER	Determine the left side variable
ICO_BEFOREEQUAL	Determine operators like += , -= , . . . etc

(Bitwise Operators)

Operation	Description
ICO_BITAND	Stack[nSize-1] = Stack[nSize-1] & Stack[nSize] , POP stack[nSize]
ICO_BITOR	Stack[nSize-1] = Stack[nSize-1] | Stack[nSize] , POP stack[nSize]
ICO_BITXOR	Stack[nSize-1] = Stack[nSize-1] ^ Stack[nSize] , POP stack[nSize]
ICO_BITNOT	Stack[nSize] = ! Stack[nSize]
ICO_BITSHL	Stack[nSize-1] = Stack[nSize-1] << Stack[nSize] , POP stack[nSize]
ICO_BITSHR	Stack[nSize-1] = Stack[nSize-1] >> Stack[nSize] , POP stack[nSize]

(For Step)

Operation	Description
ICO_STEPNUMBER	Determine step number in for loop
ICO_POPSTEP	POP step number from steps stack
ICO_LOADAFIRST	Load the first address of variable name

(Custom Global Scope)

Operation	Description
ICO_NEWGLOBALSCOPE	Start new custom global scope - used by 'load package' command
ICO_ENDGLOBALSCOPE	End of custom global scope - used by 'load package' command
ICO_SETGLOBALSCOPE	Set the current global scope

CHAPTER

FIVE

RESOURCES

In this section you will find resources about the language

125.1 Ring Language Website

For news about the language check the website

URL: http://ring-lang.net

125.2 Source Code

Ring is Free-Open Source (MIT License)

URL: https://github.com/ring-lang/ring

125.3 Ring Group

If you have any question or would like to send a bug report

URL: https://groups.google.com/g/ring-lang

125.4 Ring Team

URL: http://ring-lang.net/team.html

INDEX

Symbols

3D Camera
    Using RingRayLib, 733
3D Camera Free
    Using RingRayLib, 735
3D Cube and Texture
    Using RingOpenGL and RingAllegro for 3D Graphics, 841
3D Picking
    Using RingRayLib, 742
64bit version
    Building From Source Code, 1964

A

AbstractAxis Class
    RingQt Classes Reference, 1203
AbstractBarSeries Class
    RingQt Classes Reference, 1206
Access List Items by String Index
    Lists, 103
Access Objects Using Braces
    Object Oriented Programming, 226
Access String Letters
    Strings, 108
Accessing the class attributes from braces inside class methods
    Scope Rules, 302
Add Items
    Lists, 96
addattribute()
    Reflection and Meta-programming, 261
AddDays()
    Date and Time, 119
Adding code to the generated code
    Code Generator, 1590
Adding Hyperlink to QLabel
    Desktop, WebAssembly and Mobile Development, 427
addmethod()
    Reflection and Meta-programming, 262
Analog Clock
    Applications developed in a few hours, 13
Animate Class
    Game Engine for 2D Games, 676
Animate Events
    Game Engine for 2D Games, 689
Animation
    Game Engine for 2D Games, 682
Animation and Functions
    Game Engine for 2D Games, 683
Anonymous and Nested Functions
    Functional Programming, 245
apparguments()
    Stdlib Functions, 345
Application Class
    Web Development (*CGI Library*), 647
Application Database
    Deploying Web Applications in the Cloud, 655
Applications
    How to contribute?, 1975
Applications developed in a few hours
    Analog Clock, 13
    FetchStockData Application, 10
    Fifteen Puzzle Game, 12
    Google API Shortener Application, 10
    Innovative, 22
    Introduction, 1
    Practical, 23
    Quotes about Ring, 2
    Samples in this book, 15
    Squares Puzzle Game, 14
    Video-Music-Player Application, 11
apppath()
    Stdlib Functions, 346
Arithmetic Operations
    Performance Tips, 908
Arithmetic Operators
    Operators, 59
ASCII Table application
    What is new in Ring 1.12?, 1798
Ascii()
    Data Type, 127
ASCIIList2Str()

Stdlib Functions, 364
Assert()
Eval() and Debugging, 161
Assignment Operators
Operators, 60
attributes()
Reflection and Meta-programming, 258
Automatic loading for syntax files
Syntax Flexibility, 327

B

Basic Shapes
Using RingRayLib, 748
Basic Window
Using RingRayLib, 730
Better batch files for building Ring
What is new in Ring 1.19, 1906
Better Call Command
What is new in Ring 1.2?, 1640
Better Cards Game
What is new in Ring 1.8?, 1748
Better CitPre application
What is new in Ring 1.18, 1887
Better Code Generator For Extensions
What is new in Ring 1.8?, 1749
Better Code Generator for Extensions
What is new in Ring 1.1?, 1634
What is new in Ring 1.5?, 1696
What is new in Ring 1.9?, 1767
Better Documentation
What is new in Ring 1.18, 1899
What is new in Ring 1.19, 1914
What is new in Ring 1.11?, 1795
What is new in Ring 1.16?, 1867
What is new in Ring 1.1?, 1634
What is new in Ring 1.7?, 1734
Better Documentation Generator for Extensions
What is new in Ring 1.5?, 1697
Better Extensions Generator
What is new in Ring 1.19, 1914
Better Form Designer
What is new in Ring 1.8?, 1748
Better Functions
What is new in Ring 1.14?, 1845
What is new in Ring 1.2?, 1637
Better GUILib
What is new in Ring 1.17, 1877
Better Handling For Numbers
What is new in Ring 1.14?, 1846
Better I/O Functions
What is new in Ring 1.17, 1880
Better Installation Scripts
What is new in Ring 1.16?, 1867
Better Loop|Exit Command
What is new in Ring 1.3?, 1650
Better Memory Management
What is new in Ring 1.9?, 1766
Better Natural Language Programming Support
What is new in Ring 1.1?, 1625
Better Objects Library
What is new in Ring 1.5?, 1681
Better Organization
What is new in Ring 1.13?, 1824
Better Performance
What is new in Ring 1.11?, 1795
What is new in Ring 1.8?, 1738
Better Performance For Strings
What is new in Ring 1.14?, 1845
Better Quality
What is new in Ring 1.2?, 1641
What is new in Ring 1.5?, 1706
Better Ring API
What is new in Ring 1.17, 1881
What is new in Ring 1.18, 1898
Better Ring Compiler and VM
What is new in Ring 1.8?, 1750
Better Ring For Android
What is new in Ring 1.6?, 1719
Better Ring Notepad
What is new in Ring 1.15?, 1854
What is new in Ring 1.2?, 1637
What is new in Ring 1.3?, 1646
What is new in Ring 1.5?, 1672
What is new in Ring 1.6?, 1722
What is new in Ring 1.7?, 1732
What is new in Ring 1.9?, 1757
Better Ring VM
What is new in Ring 1.7?, 1734
Better Ring2EXE
What is new in Ring 1.7?, 1733
Better RingLibCurl
What is new in Ring 1.15?, 1858
Better RingOpenSSL
What is new in Ring 1.17, 1879
What is new in Ring 1.14?, 1843
Better RingQt
What is new in Ring 1.11?, 1793
What is new in Ring 1.13?, 1819
What is new in Ring 1.15?, 1857
What is new in Ring 1.2?, 1638
What is new in Ring 1.3?, 1642
What is new in Ring 1.4?, 1664
What is new in Ring 1.5?, 1679
What is new in Ring 1.6?, 1724
What is new in Ring 1.7?, 1732
What is new in Ring 1.8?, 1749
What is new in Ring 1.9?, 1763

Better RingRayLib
What is new in Ring 1.18, 1894
What is new in Ring 1.19, 1911
What is new in Ring 1.12?, 1805
What is new in Ring 1.13?, 1822
Better RingREPL
What is new in Ring 1.6?, 1725
Better RingSockets
What is new in Ring 1.18, 1895
Better RingStbImage
What is new in Ring 1.19, 1912
Better RingVM
What is new in Ring 1.6?, 1725
Better RingWinAPI
What is new in Ring 1.17, 1879
Better RingZip
What is new in Ring 1.7?, 1733
Better Scripts for building Ring
What is new in Ring 1.6?, 1721
Better StdLib
What is new in Ring 1.15?, 1856
What is new in Ring 1.3?, 1650
What is new in Ring 1.4?, 1662
What is new in Ring 1.5?, 1675
What is new in Ring 1.6?, 1725
What is new in Ring 1.9?, 1758
Better Syntax Flexibility
What is new in Ring 1.17, 1879
Better Tests
What is new in Ring 1.10?, 1778
Better Threads Support
What is new in Ring 1.13?, 1810
Better Tools
What is new in Ring 1.18, 1888
Better Trace Library
What is new in Ring 1.7?, 1731
Better TypeHints
What is new in Ring 1.19, 1910
Better WebLib
What is new in Ring 1.18, 1897
What is new in Ring 1.4?, 1662
What is new in Ring 1.5?, 1676
Bezier Lines
Using RingRayLib, 752
bignumber
BigNumber Class, 973
BigNumber Functions, 973
Examples, 971
Introduction, 970
Library Source Code, 973
Loading the library, 971
BigNumber Class
bignumber, 973
BigNumber Functions
bignumber, 973
BigNumber Library
What is new in Ring 1.9?, 1759
binarydigits()
Stdlib Functions, 356
Bitwise Operators
Operators, 60
BMI Calculator application
What is new in Ring 1.12?, 1798
BraceError() Method
Natural Language Programming, 284
BraceExprEval Method
Natural Language Programming, 282
BraceStart and BraceEnd Methods
Natural Language Programming, 282
Branching
Control Structures, 63
Control Structures - Second Style, 70
Control Structures - Third Style, 74
BreakPoint
The Trace Library and the Interactive Debugger, 336
Build the extension on different platforms
Tutorial: Ring Extensions in C/C++, 1545
Building From Source Code
64bit version, 1964
Building using CMake, 1973
Building using MacOS X, 1970
Building using Microsoft Windows, 1964
Building using Ubuntu Linux, 1967
Complete Build, 1964
Custom Build, 1964
Introduction, 1963
Building Games For Android
Building the project, 728
Download Requirements and Update the Android SDK, 727
Introduction, 726
Project Folder, 727
Building RingQt Applications for Mobile
Comments about developing for Android using RingQt, 583
Download Requirements, 578
Install Qt for Android, 578
Introduction, 577
The Qt project for your Ring application, 579
Using Ring2EXE, 579
Building RingQt Applications for WebAssembly
Comments about developing for WebAssembly using RingQt, 591
Dialogs, 592
Download Requirements, 586

Introduction, 585
Online Applications, 593
The Qt project for your Ring application, 588
Using Ring2EXE, 587
Building standalone console application
Distributing Ring Application using Ring2EXE, 933
Building the Cards Game for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 935
Building the Form Designer for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 937
Building the project
Building Games For Android, 728
Building the Weight History Application for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 936
Building using CMake
Building From Source Code, 1973
Building using MacOS X
Building From Source Code, 1970
Building using Microsoft Windows
Building From Source Code, 1964
Building using Ubuntu Linux
Building From Source Code, 1967
Bytes2List() function
Using FastPro, 1499

C

Calculator Application
What is new in Ring 1.5?, 1671
Calendar application
What is new in Ring 1.12?, 1799
Call Functions
Functions - First Style, 81
Functions - Second Style, 85
Functions - Third Style, 89
callgc()
Low Level Functions, 1516
Calling a function sharing the name with a method in the current class
Scope Rules for Functions and Methods, 314
Calling Functions and Methods
Performance Tips, 909
Camera
Using Qt3D, 556
Camera First Person
Using RingRayLib, 740
Can I connect to dbase/harbour database?
Frequently Asked Questions, 1961
Can Ring work on Windows XP?
Frequently Asked Questions, 1951
capitalized()
Stdlib Functions, 349
cfunctions()
Reflection and Meta-programming, 250
CGI Support
Command Line Options, 925
Change Focus
Desktop, WebAssembly and Mobile Development, 466
Change Language Keywords
Syntax Flexibility, 316
Change Language Operators
Syntax Flexibility, 317
Change the '=' operator to 'is'
Natural Language Programming, 279
Change the Ring Keyword 'And'
Natural Language Programming, 277
Change the Ring Operator '+'
Natural Language Programming, 279
Change: Basic Extensions are separated from RingVM
What is new in Ring 1.4?, 1656
changestring()
Stdlib Functions, 359
Char()
Data Type, 128
ChDir() Function
System Functions, 157
Check Character
Data Type, 123
Check Data Type
Data Type, 121
Check Parameters Count
Extension, 1572
Check Parameters Type
Extension, 1573
CheckEquality()
Stdlib Functions, 366
Checkers Game
What is new in Ring 1.11?, 1782
Chess 3D (*Qt3D Sample*)
What is new in Ring 1.15?, 1853
Chess Endgame Application
What is new in Ring 1.15?, 1851
Chess Game
What is new in Ring 1.10?, 1770
Circular References
Using References, 1508
Citations Prediction
What is new in Ring 1.17, 1871

Classes and Objects
    Object Oriented Programming, 224
Classes and their Methods to use the default events
    Desktop, WebAssembly and Mobile Development, 499
classes()
    Reflection and Meta-programming, 254
classname()
    Reflection and Meta-programming, 257
Clean Natural Code
    Natural Language Programming, 285
Clearerr()
    Files, 143
Client Example
    RingLibuv, 1126
Client Example - Using Classes
    RingLibuv, 1128
clock()
    Date and Time, 116
Close Window Event
    RingLibSDL, 1095
ClosPerSecond()
    Date and Time, 116
Code Generator
    Adding code to the generated code, 1590
    Comments in configuration file, 1592
    Configuration file, 1588
    Configuration file for the Allegro library, 1594
    Configuration Files Examples, 1610
    Constants Type, 1593
    Defining Constants, 1591
    Determine Structure Members Types, 1591
    Enum and Numbers, 1592
    Executing code during code generation, 1592
    Extra names for functions, 1610
    Filtering using Expressions, 1593
    function prototype, 1588
    Introduction, 1587
    Loading Files, 1609
    Managed Classes, 1609
    Prefix for Functions Names, 1590
    Qt configuration file, 1598
    Register New Functions, 1591
    Static Methods, 1608
    Threads Support, 1595
    Using configuration file that wrap C++ library, 1597
    Using the tool, 1588
    Wrap structures, 1590
    Wrapping C++ Classes, 1597
Code Generator for Extensions
    What is new in Ring 1.17, 1882
Code Runner Extension support Ring
    What is new in Ring 1.16?, 1864
CodeEditor Class
    RingQt Classes Reference, 1207
Collision Area
    Using RingRayLib, 753
Colors Palette
    Using RingRayLib, 758
Command Line Options
    CGI Support, 925
    Generate Embedded Object File, 927
    Generate Object File, 927
    Introduction, 910
    No Run, 925
    Performance, 926
    Printing Final Intermediate Code, 920
    Printing Instruction Operation Code, 926
    Printing Intermediate Code, 918
    Printing Rules, 913
    Printing Tokens, 911
Comments about developing for Android using RingQt
    Building RingQt Applications for Mobile, 583
Comments about developing for WebAssembly using RingQt
    Building RingQt Applications for WebAssembly, 591
Comments about evaluation
    Control Structures - First Style, 69
Comments in configuration file
    Code Generator, 1592
Compact Syntax
    Language Design, 1617
Compiler and Virtual Machine (*VM*)
    How to contribute?, 1975
Compiler Errors
    Language Specification, 1984
Compiling on Manjaro Linux
    What is new in Ring 1.8?, 1743
Complete Build
    Building From Source Code, 1964
Composition
    Object Oriented Programming, 227
Configuration file
    Code Generator, 1588
Configuration file for the Allegro library
    Code Generator, 1594
Configuration Files Examples
    Code Generator, 1610
Configure the Apache web server
    Using XAMPP Apache web server, 594
    Web Development (*CGI Library*), 594

Conflict between Class Attributes and Local Variables
Scope Rules, 298
Conflict between Global Variables and Class Attributes
Frequently Asked Questions, 1943
Scope Rules, 297
Conflict between self inside braces and self in the class region
Scope Rules, 305
Constants
RingStbImage Functions Reference, 1153
socket, 1013
Using TokensLib, 996
Constants Type
Code Generator, 1593
Constructor methods in Ring
Frequently Asked Questions, 1939
Context Menu
Desktop, WebAssembly and Mobile Development, 408
Control Gallery
libui, 1021
Control Structures
Branching, 63
Looping, 75
Control Structures - First Style
Comments about evaluation, 69
Do Again Loop, 66
Exit, 67
Exit from two loops, 67
For In Loop, 65
for in to modify lists, 66
For Loop, 65
Introduction, 62
Loop Command, 67
Looping, 64
Short-circuit evaluation, 68
Step Option, 66
Switch Statement, 63
While Loop, 64
Control Structures - Second Style
Branching, 70
Exceptions, 73
For In Loop, 72
For Loop, 72
If Statement, 70
Introduction, 69
Looping, 71
Switch Statement, 70
Control Structures - Third Style
Branching, 74
Exceptions, 77
For In Loop, 77
For Loop, 76
If Statement, 74
Introduction, 73
Switch Statement, 74
While Loop, 75
Conversion
Data Type, 126
Conversion Class
Stdlib Classes, 385
Convert between Numbers and Bytes
What is new in Ring 1.4?, 1661
Convert Letters Case
Strings, 108
Cookies
Web Development (*CGI Library*), 607
Copy Lists
Lists, 101
Copy()
Strings, 110
Could you explain the output of the StrCmp() function?
Frequently Asked Questions, 1946
Create Database
MySQL Functions, 178
Create Executable File
Getting Started - First Style, 47
Getting Started - Second Style, 50
Getting Started - Third Style, 53
Create Lists
Lists, 96
Create Multi Dimension Array Using List and Recursion
Lists, 105
Create Package in the Current Folder
Ring Package Manager (*RingPM*), 947
Create Table
Tutorial: Ring Extensions in C/C++, 1565
Create Table and Insert Data
MySQL Functions, 178
Create Window
RingLibSDL, 1091
Create your first extension
Introduction, 1611
Location, 1612
Steps to create the extension, 1612
Testing the extension, 1613
Create Zip File
RingZip, 1063
Creating a Class for each Window in GUI applications
Scope Rules, 304
Creating Lists
Performance Tips, 907
Creating Menubar

Desktop, WebAssembly and Mobile Development, 406
Creating more than one window
Desktop, WebAssembly and Mobile Development, 448
Creating New Package
Ring Package Manager (*RingPM*), 944
Creating Reports using the WebLib and the GUILib
Desktop, WebAssembly and Mobile Development, 506
Creating StatusBar
Desktop, WebAssembly and Mobile Development, 411
Creating the Game Window
Game Engine for 2D Games, 677
Creating the Qt resource file using Folder2qrc
Distributing Ring Application using Ring2EXE, 939
Creating Threads
Threads, 1014
Creating Toolbar
Desktop, WebAssembly and Mobile Development, 409
Creating Windows Installer
Distributing Ring Application, 929
CRUD Example using MVC
Web Development (*CGI Library*), 631
CSVLib Library
What is new in Ring 1.17, 1873
Cubic Map
Using RingRayLib, 773
CurrentDir() Function
System Functions, 156
Custom Build
Building From Source Code, 1964
Customers Form
What is new in Ring 1.14?, 1834

D

Data Representation
Language Design, 1616
Data Type
Ascii(), 127
Char(), 128
Check Character, 123
Check Data Type, 121
Conversion, 126
Dec(), 128
Hex(), 128
Hex2str(), 129
Introduction, 120
IsAlNum(), 123
IsAlpha(), 123
IsCntrl(), 124
IsDigit(), 124
IsGraph(), 124
IsList(), 122
IsLower(), 125
IsNULL(), 122
IsNumber(), 121
IsPrint(), 125
IsPunct(), 125
IsSpace(), 126
IsString(), 121
IsUpper(), 126
IsXdigit(), 126
Number(), 127
Str2Hex(), 129
String(), 127
Type(), 122
Database, ModelBase & ControllerBase classes
Web Development (*CGI Library*), 640
DataType Class
Stdlib Classes, 384
Date and Time
AddDays(), 119
clock(), 116
ClosPerSecond(), 116
Date(), 117
DiffDays(), 119
EpochTime(), 119
Introduction, 115
Time(), 116
TimeList(), 117
Date()
Date and Time, 117
DateTime Class
Stdlib Classes, 380
dayofweek()
Stdlib Functions, 357
Debug Class
Stdlib Classes, 383
Dec()
Data Type, 128
Decimals()
Mathematical Functions, 134
Declarative Programming
Declarative programming on the top of Object-Oriented, 273
executing code after the end of object access, 273
Introduction, 269
More Beautiful Code, 275
Objects inside lists, 270
Return object by reference, 271

Declarative programming on the top of Object-Oriented
    Declarative Programming, 273
Declare parameters
    Functions - First Style, 82
    Functions - Second Style, 86
    Functions - Third Style, 90
Decrypt()
    Security and Internet Functions, 201
Deep Copy
    Variables, 57
Default value for object attributes
    Object Oriented Programming, 241
Default value for object attributes is NULL
    What is new in Ring 1.8?, 1745
Define Declarative Languages
    Language Design, 1620
Define Functions
    Functions - First Style, 81
    Functions - Second Style, 85
    Functions - Third Style, 89
Define Natural Statements
    Language Design, 1618
Defining Commands
    Using the Natural Library, 289
Defining commands using classes
    Using the Natural Library, 293
Defining Constants
    Code Generator, 1591
Defining Variables and Variables Access
    Scope Rules, 295
Delete Item From List
    Lists, 97
Deleting All Packages
    Ring Package Manager (*RingPM*), 944
Demo Programs
    Introduction, 162
    Language Shell, 163
    Main Menu, 164
    The Cards Game, 493
Dependency injection
    Using References, 1511
Deploying after updates
    Deploying Web Applications in the Cloud, 655
Deploying Web Applications in the Cloud
    Application Database, 655
    Deploying after updates, 655
    Hello World program, 655
    Introduction, 652, 653
    Local Tests, 655
    Ring source code files and permissions, 654
    Usage, 653
Deploying Web applications in the Cloud
    What is new in Ring 1.9?, 1763
Designed for a clear goal
    Language Design, 1614
Desktop Screen Shot Application
    What is new in Ring 1.11?, 1787
Desktop, WebAssembly and Mobile Development
    Adding Hyperlink to QLabel, 427
    Change Focus, 466
    Classes and their Methods to use the default events, 499
    Context Menu, 408
    Creating Menubar, 406
    Creating more than one window, 448
    Creating Reports using the WebLib and the GUILib, 506
    Creating StatusBar, 411
    Creating Toolbar, 409
    Dialog Functions, 456
    Display Image using QLabel, 431
    Display Scaled Image using QLabel, 441
    Drawing using QPainter, 443
    Dynamic Objects, 470
    Inheritance from GUI Classes, 462
    KeyPress and Mouse Move Events, 457
    Menubar and StyleSheet Example, 432
    Methods to use Events with Events Filter, 502
    Movable Label Example, 452
    Moving Objects using the Mouse, 458
    New Classes names - Index Start from 1, 504
    Notepad Application, 476
    Other Widgets Events, 436
    Playing Sound, 450
    Printing using QPrinter, 445
    QLineEdit Events and QMessageBox, 434
    QMessagebox Example, 453
    QVideoWidget and QMediaPlayer, 428
    Regular Expressions, 467
    RingQt Classes and their Qt Documentation, 504
    Rotate Text, 464
    Simple Client and Server Example, 468
    TableWidget - AddList() Method, 505
    The Cards Game, 493
    The Difference between Qt and RingQt, 503
    The Events Loop, 396
    The First GUI Application, 394
    Using Layout, 396
    Using QCheckBox, 424
    Using QComboBox Class, 405
    Using QDateEdit, 419
    Using QDesktopWidget Class, 463

Using QDial, 420
Using QDockWidget, 412
Using QFrame, 430
Using QInputDialog Class, 455
Using qLCDNumber, 451
Using QPrintPreviewDialog, 446
Using QProgressBar, 416
Using QProgressBar and Timer, 440
Using QRadioButton and QButtonGroup, 425
Using QSlider, 417
Using QSpinBox, 417
Using QTableWidget, 415
Using QTabWidget, 413
Using QTreeView and QFileSystemModel, 402
Using QTreeWidget and QTreeWidgetItem, 403
Using QWebView, 423
Using the QColorDialog Class, 450
Using the QFileDialog Class, 442
Using the QListWidget Class, 399
Using the QTextEdit Class, 398
Using the QTimer Class, 439
Weight History Application, 472
Determine Structure Members Types
Code Generator, 1591
Dialog Functions
Desktop, WebAssembly and Mobile Development, 456
Dialogs
Building RingQt Applications for WebAssembly, 592
DiffDays()
Date and Time, 119
Dir()
Files, 138
Direxists()
Files, 145
Disable BreakPoints
The Trace Library and the Interactive Debugger, 336
Discovering Packages
Ring Package Manager (*RingPM*), 941
Display Error Message
Extension, 1573
Display Image
RingLibSDL, 1091
Display Image using QLabel
Desktop, WebAssembly and Mobile Development, 431
Display List
Tutorial: Ring Extensions in C/C++, 1558
Display PNG Images
RingLibSDL, 1093
Display Scaled Image using QLabel
Desktop, WebAssembly and Mobile Development, 441
Display Transparent Images
RingLibSDL, 1094
Display Warnings Option
What is new in Ring 1.2?, 1640
Distributing Applications and Games for Mobile
Distributing Ring Application, 929
Distributing Applications for Microsoft Windows
Distributing Ring Application, 928
Distributing Applications for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 935
Distributing Applications for WebAssembly using RingQt
Distributing Ring Application using Ring2EXE, 935
Distributing Ring Application
Creating Windows Installer, 929
Distributing Applications and Games for Mobile, 929
Distributing Applications for Microsoft Windows, 928
Introduction, 927
Protecting the Source Code, 928
The files ring.ring and ring.ringo, 929
Using C/C++ Compiler and Linker, 929
Distributing Ring Application using Ring2EXE
Building standalone console application, 933
Building the Cards Game for Mobile using RingQt, 935
Building the Form Designer for Mobile using RingQt, 937
Building the Weight History Application for Mobile using RingQt, 936
Creating the Qt resource file using Folder2qrc, 939
Distributing Applications for Mobile using RingQt, 935
Distributing Applications for WebAssembly using RingQt, 935
Distributing RingAllegro Applications, 933, 934
Example, 932
How Ring2EXE works?, 931
Important Information about Ring2EXE, 939
Introduction, 930
Options, 932
Using Ring2EXE, 931

Distributing RingAllegro Applications
    Distributing Ring Application using Ring2EXE, 933, 934
Do Again Loop
    Control Structures - First Style, 66
Documentation
    How to contribute?, 1974
Download Requirements
    Building RingQt Applications for Mobile, 578
    Building RingQt Applications for WebAssembly, 586
Download Requirements and Update the Android SDK
    Building Games For Android, 727
Download()
    Security and Internet Functions, 222
Draw Gradient
    libui, 1035
Draw Office
    Using Qt3D, 551
Draw Rectangle
    RingLibSDL, 1092
Draw Ring
    Using RingRayLib, 749
Drawing Cone
    Using Qt3D, 529
Drawing Cube
    Using Qt3D, 521
Drawing Cylinder
    Using Qt3D, 527
Drawing Plane
    Using Qt3D, 531
Drawing Sample
    libui, 1034
Drawing Sphere
    Using Qt3D, 525
Drawing Text
    Game Engine for 2D Games, 678
Drawing Torus
    Using Qt3D, 523
Drawing using QPainter
    Desktop, WebAssembly and Mobile Development, 443
Drawing using RingOpenGL
    Using RingOpenGL and RingFreeGLUT for 3D Graphics, 789
Drawing, Animation and Input
    Graphics and Game Programming, 657
Dynamic Attributes
    Object Oriented Programming, 234
Dynamic Objects
    Desktop, WebAssembly and Mobile Development, 470
Dynamic Typing
    Variables, 57

E

Editors Support
    How to contribute?, 1975
Embedding Ring in Ring
    Embedding Ring in Ring, 338
    Embedding Ring in Ring without sharing the State, 339
    ring_state_filetokens() function, 343
    ring_state_new() and ring_state_mainfile(), 341
    ring_state_setvar(), 340
    Runtime Errors when Embedding Ring in Ring, 342
    Serial Execution of Programs, 340
Embedding Ring in Ring without sharing the State
    Embedding Ring in Ring, 339
    What is new in Ring 1.3?, 1652
Embedding Ring Language in C/C++ Programs
    Introduction, 1584
    Ring State, 1585
    Ring State Functions, 1585
    Ring State Variables, 1586
Employee Application
    What is new in Ring 1.6?, 1718
Encourage Organization
    Language Design, 1616
Encrypt()
    Security and Internet Functions, 201
endswith()
    Stdlib Functions, 353
Entering Items
    Form Designer, 516
Enum and Numbers
    Code Generator, 1592
Environment Errors
    Language Specification, 1986
EpochTime()
    Date and Time, 119
    Stdlib Functions, 361
Equality of functions
    Functional Programming, 247
Erlang B Formula
    What is new in Ring 1.14?, 1833
Escape Game
    What is new in Ring 1.13?, 1817
Eval()
    Eval() and Debugging, 159
Eval() and Debugging
    Assert(), 161
    Eval(), 159

Introduction, 158
Raise(), 160
Try/Carch/Done, 159
evenorodd()
Stdlib Functions, 354
Events Code
Form Designer, 512
Example
Distributing Ring Application using Ring2EXE, 932
Files, 146
FoxRing Functions Reference, 968
Mathematical Functions, 130
Natural Language Programming, 276
Objects Library for RingQt Application, 565
RingMurmurHash Functions Reference, 1067
Security and Internet Functions, 201
System Functions, 153
The Type Hints Library, 330
Using HTTPLib, 979
Using TokensLib, 995
Example - The Trace Library
Low Level Functions, 1537
Example - Using the Trace Functions
Low Level Functions, 1533
Example about Sharing Names between Functions and Methods
Scope Rules for Functions and Methods, 313
Examples
bignumber, 971
PostgreSQL, 189
Using CSVLib, 974
Using JSONLib, 977
Exceptions
Control Structures - Second Style, 73
Control Structures - Third Style, 77
Execute Program Line by Line
The Trace Library and the Interactive Debugger, 336
Execute Query and Print Result
ODBC Functions, 171
executing code after the end of object access
Declarative Programming, 273
Executing code during code generation
Code Generator, 1592
ExeFileName() Function
System Functions, 156
ExeFolder() Function
System Functions, 157
Exit
Control Structures - First Style, 67
Exit from two loops
Control Structures - First Style, 67
Extension
Check Parameters Count, 1572
Check Parameters Type, 1573
Display Error Message, 1573
Fopen() and Fclose() Implementation, 1575
Function Prototype, 1574
Function Structure, 1572
Get Parameters Values, 1574
Introduction, 1569
Memory Functions, 1584
Module Organization, 1571
MySQL_Columns() Implementation, 1581
Return Value, 1574
RING API - list Functions, 1576
RING API - String Functions, 1579
RING_API_STATE, 1578
ring_ext.c, 1570
ring_ext.h, 1570
Shared Libraries, 1582
Sin() Implementation, 1575
Using RING_API_RETMANAGEDCPOINTER(), 1583
Extensions in C/C++
How to contribute?, 1975
Extra names for functions
Code Generator, 1610
Extract Zip File
RingZip, 1063
Extruded Text
Using Qt3D, 545

F

Facebook Login
RingLibCurl, 999
factorial()
Stdlib Functions, 350
factors()
Stdlib Functions, 355
Fast Enough For Many Applications
Language Design, 1624
Faster (For-In) Loop
What is new in Ring 1.18, 1892
Faster BraceError()
What is new in Ring 1.18, 1893
FastPro Extension
What is new in Ring 1.19, 1909
Fclose()
Files, 140
Features
Introduction, 28
Ring Package Manager (*RingPM*), 941
Feof()
Files, 143

Ferror()
    Files, 143
FetchStockData Application
    Applications developed in a few hours, 10
Fexists()
    Files, 145
Fflush()
    Files, 140
Fgetc()
    Files, 143
Fgetpos()
    Files, 142
Fgets()
    Files, 144
fibonacci()
    Stdlib Functions, 351
Fifteen Puzzle Game
    Applications developed in a few hours, 12
File Class
    Stdlib Classes, 381
File Hash
    Security and Internet Functions, 202
file2list()
    Stdlib Functions, 352
Files
    Clearerr(), 143
    Dir(), 138
    Direxists(), 145
    Example, 146
    Fclose(), 140
    Feof(), 143
    Ferror(), 143
    Fexists(), 145
    Fflush(), 140
    Fgetc(), 143
    Fgetpos(), 142
    Fgets(), 144
    Fopen(), 139
    Fputc(), 144
    Fputs(), 144
    Fread(), 144
    Freopen(), 140
    Fseek(), 141
    Fsetpos(), 142
    Ftell(), 142
    Fwrite(), 145
    Getfilesize(), 146
    Getpathtype(), 145
    Introduction, 136
    Numbers and Bytes, 148
    Perror(), 143
    Read File using Read(), 138
    Remove(), 139
    Rename(), 139
    Rewind(), 142
    Tempfile(), 141
    Tempname(), 141
    Ungetc(), 144
    Write file using Write(), 138
Filter List Items
    Tutorial: Ring Extensions in C/C++, 1551
filter()
    Stdlib Functions, 348
Filtering using Expressions
    Code Generator, 1593
Find in files Application
    What is new in Ring 1.8?, 1739
Find SubString
    Strings, 111
Find() and List of Objects
    Object Oriented Programming, 235
First Application using RingLibuv
    RingLibuv, 1123
First-Class Functions
    Functional Programming, 244
First-Class Lists
    Lists, 101
Flappy Bird 3000 Game
    Game Engine for 2D Games, 706
Following Eyes
    Using RingRayLib, 756
Fopen()
    Files, 139
Fopen() and Fclose() Implementation
    Extension, 1575
For In Loop
    Control Structures - First Style, 65
    Control Structures - Second Style, 72
    Control Structures - Third Style, 77
for in to modify lists
    Control Structures - First Style, 66
For Loop
    Control Structures - First Style, 65
    Control Structures - Second Style, 72
    Control Structures - Third Style, 76
For Loop - Better Performance
    What is new in Ring 1.19, 1907
Form Designer
    Entering Items, 516
    Events Code, 512
    Introduction, 509
    Keyboard Shortcuts, 515
    Menubar Designer, 515
    More Samples and Tests, 520
    Objects Order and Layouts, 517
    Running Forms, 512
    The Designer Windows, 511
    The Properties, 511

Using Layouts, 517
What is new in Ring 1.3?, 1654
Window Flags, 516
Form Designer Translation
Multi-language applications, 575
Forms Translation
Multi-language applications, 576
FoxRing functions
FoxRing Functions Reference, 955
FoxRing Functions Reference
Example, 968
FoxRing functions, 955
frAbs() function, 956
frAddBs() function, 957
frALines() function, 963
frAllTrim() function, 962
frAsc() function, 956
frAt() function, 957
frAtC() function, 957
frBetween() function, 966
frChr() function, 958
frEmpty() function, 958
frFile() function, 959
frFileToStr() function, 959
frForceExt() function, 963
frForcePath() function, 961
frIIf() function, 965
frInList() function, 961
frInt() function, 965
frJustDrive() function, 962
frJustExt() function, 962
frJustFName() function, 963
frJustPath() function, 963
frJustStem() function, 963
frLen() function, 964
frListToString() function, 965
frLTrim() function, 962
frPadL() function, 963
frPadR() function, 964
frProper() function, 964
frReplicate() function, 964
frRTrim() function, 963
frSetIfEmpty() function, 960
frSetSeparatorTo() function, 966
frSpace() function, 961
frStr() function, 959
frStringToList() function, 965
frStrTran() function, 965
frStuff() function, 964
frSubStr() function, 965
frTransform() function, 967
frVal() function, 966
frVarType() function, 967
Introduction, 954
FoxRing Library
What is new in Ring 1.8?, 1747
Fputc()
Files, 144
Fputs()
Files, 144
frAbs() function
FoxRing Functions Reference, 956
frAddBs() function
FoxRing Functions Reference, 957
frALines() function
FoxRing Functions Reference, 963
frAllTrim() function
FoxRing Functions Reference, 962
Frame Action
Using Qt3D, 540
Frames Per Second
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 826
frAsc() function
FoxRing Functions Reference, 956
frAt() function
FoxRing Functions Reference, 957
frAtC() function
FoxRing Functions Reference, 957
frBetween() function
FoxRing Functions Reference, 966
frChr() function
FoxRing Functions Reference, 958
Fread()
Files, 144
frEmpty() function
FoxRing Functions Reference, 958
Freopen()
Files, 140
Frequently Asked Questions
Can I connect to dbase/harbour database?, 1961
Can Ring work on Windows XP?, 1951
Conflict between Global Variables and Class Attributes, 1943
Constructor methods in Ring, 1939
Could you explain the output of the StrCmp() function?, 1946
Getter and Setter Methods, 1940
Goal of including the "Main" function in Ring, 1936
How can I disable maximize button and resize window?, 1955
How to add Combobox and other elements to the cells of a QTableWidget?, 1959
How to Close a window then displaying another one?, 1954
How to create a Modal Window?, 1954

How to create an array of buttons in GUI applications?, 1953
How to extend RingQt and add more classes?, 1957
How to get the current source file path?, 1944
How to get the file size using ftell() and fseek() functions?, 1943
How to insert an item to the first position in the list?, 1950
How to perform some manipulations on selected cells in QTableWidget?, 1960
How to print keys or values only in List/Dictionary?, 1945
How to print lists that contains objects?, 1950
How to print new lines and other characters?, 1950
How to use many source code files in the project?, 1947
How to use NULL and ISNULL() function?, 1948
How to use SQLite using ODBC?, 1961
Introduction, 1924
Is Ring some sort of an improvement of PHP?, 1931
List index start from 1, 1937
Philosophy behind data types in Ring, 1934
Search of global names while defining the class attributes, 1941
The documentation says functional programming is supported, but then this happens?, 1933
What about Data Representation in Ring?, 1928
What about Memory Management in Ring?, 1927
What about predefined parameters or optional parameters in functions?, 1944
What about the Boolean values in Ring?, 1935
What are the advantages of using Ring over native C or C++?, 1931
What are the advantages to using Ring over C# or Java?, 1933
What are the advantages to using Ring over Lisp or Smalltalk?, 1930
What are the advantages to using Ring over Python and Ruby?, 1932
What are the advantages to using Ring over Tcl and Lua?, 1932
What happens when we create a new object?, 1940
What is the difference between Ring and Python? And is Ring Open Source?, 1931
What is the Ring Architecture?, 1926
Where can I write a program and execute it?, 1943
Which of 3 coding styles are commonly used or recommended by the community?, 1926
Why do we need Yet Another Programming Language (YAPL)?, 1925
Why I get a strange result when printing nl with lists?, 1945
Why I get Calling Function without definition Error?, 1951
Why is Ring largely focused on UI creation?, 1930
Why is Ring weakly typed?, 1930
Why Ring is not case-sensitive, 1938
Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?, 1934
Why setClickEvent() doesn't see the object methods directly?, 1956
Why the ability to define your own languages Instead of parsing?, 1934
Why the Assignment operator uses Deep copy?, 1938
Why the window title bar is going outside the screen?, 1952
Why this example use the GetChar() twice?, 1947
Why we don't use () after the QApp class name?, 1952
Why you can specify the number of loops you want to break out of?, 1934
frFile() function
FoxRing Functions Reference, 959
frFileToStr() function
FoxRing Functions Reference, 959
frForceExt() function
FoxRing Functions Reference, 963
frForcePath() function
FoxRing Functions Reference, 961
frIIf() function
FoxRing Functions Reference, 965
frInList() function
FoxRing Functions Reference, 961
frInt() function
FoxRing Functions Reference, 965
frJustDrive() function
FoxRing Functions Reference, 962
frJustExt() function
FoxRing Functions Reference, 962

frJustFName() function
    FoxRing Functions Reference, 963
frJustPath() function
    FoxRing Functions Reference, 963
frJustStem() function
    FoxRing Functions Reference, 963
frLen() function
    FoxRing Functions Reference, 964
frListToString() function
    FoxRing Functions Reference, 965
frLTrim() function
    FoxRing Functions Reference, 962
frPadL() function
    FoxRing Functions Reference, 963
frPadR() function
    FoxRing Functions Reference, 964
frProper() function
    FoxRing Functions Reference, 964
frReplicate() function
    FoxRing Functions Reference, 964
frRTrim() function
    FoxRing Functions Reference, 963
frSetIfEmpty() function
    FoxRing Functions Reference, 960
frSetSeparatorTo() function
    FoxRing Functions Reference, 966
frSpace() function
    FoxRing Functions Reference, 961
frStr() function
    FoxRing Functions Reference, 959
frStringToList() function
    FoxRing Functions Reference, 965
frStrTran() function
    FoxRing Functions Reference, 965
frStuff() function
    FoxRing Functions Reference, 964
frSubStr() function
    FoxRing Functions Reference, 965
frTransform() function
    FoxRing Functions Reference, 967
frVal() function
    FoxRing Functions Reference, 966
frVarType() function
    FoxRing Functions Reference, 967
Fseek()
    Files, 141
Fsetpos()
    Files, 142
FSize()
    Stdlib Functions, 360
Ftell()
    Files, 142
Full Screen
    Using RingRayLib, 744
Function Prototype
    Extension, 1574
function prototype
    Code Generator, 1588
Function Structure
    Extension, 1572
Functional Programming
    Anonymous and Nested Functions, 245
    Equality of functions, 247
    First-Class Functions, 244
    Higher-order Functions, 244
    Introduction, 242
    Pure Functions, 243
Functions
    RingStbImage Functions Reference, 1154
    sockets, 1012
    Using CSVLib, 974
    Using JSONLib, 977
    Using RingRayLib, 775, 785
Functions - First Style
    Call Functions, 81
    Declare parameters, 82
    Define Functions, 81
    Introduction, 80
    Main Function, 82
    Recursion, 84
    Return Value, 83
    Send Parameters, 82
    Variables Scope, 83
Functions - Second Style
    Call Functions, 85
    Declare parameters, 86
    Define Functions, 85
    Introduction, 84
    Main Function, 87
    Recursion, 88
    Return Value, 87
    Send Parameters, 86
    Variables Scope, 87
Functions - Third Style
    Call Functions, 89
    Declare parameters, 90
    Define Functions, 89
    Introduction, 88
    Main Function, 91
    Recursion, 92
    Return Value, 91
    Send Parameters, 90
    Variables Scope, 91
functions()
    Reflection and Meta-programming, 250
Fwrite()
    Files, 145

G

Game Class
    Game Engine for 2D Games, 674
Game Engine Classes
    Game Engine for 2D Games, 673
Game Engine for 2D Games
    Animate Class, 676
    Animate Events, 689
    Animation, 682
    Animation and Functions, 683
    Creating the Game Window, 677
    Drawing Text, 678
    Flappy Bird 3000 Game, 706
    Game Class, 674
    Game Engine Classes, 673
    GameObject Class, 675
    Games Layer, 673
    Graphics Library Bindings, 672
    Interface to graphics library, 672
    Introduction, 671
    Map, 691
    Map Class, 677
    Map Events, 692
    Moving Text, 679
    Object and Drawing, 694
    Playing Sound, 681
    Project Layers, 672
    Sound Class, 677
    Sprite Automatic Movement, 685
    Sprite Class, 675
    Sprite Keypress Event, 686
    Sprite Mouse Event, 687
    Sprite State Event, 688
    Stars Fighter Game, 698
    Super Man 2016 Game, 714
    Text Class, 676
    What is new in Ring 1.1?, 1632
Game of Life Game
    What is new in Ring 1.10?, 1773
GameObject Class
    Game Engine for 2D Games, 675
Games Layer
    Game Engine for 2D Games, 673
gcd()
    Stdlib Functions, 353
Generate Embedded Object File
    Command Line Options, 927
    What is new in Ring 1.12?, 1804
Generate List
    Tutorial: Ring Extensions in C/C++, 1556
Generate Object File
    Command Line Options, 927
Generate/Execute Ring Object Files (*.ringo)
    What is new in Ring 1.1?, 1626
Generating Pages using Objects
    Web Development (*CGI Library*), 620
Geometric Shapes
    Using RingRayLib, 771
Get Active Source File Name
    System Functions, 155
Get Command Line Arguments
    System Functions, 154
Get List Item
    Lists, 97
Get List Size
    Lists, 97
Get Number of Characters from position
    Strings, 111
Get Parameters Values
    Extension, 1574
Get Request
    RingLibCurl, 999
Get Stock Data From Yahoo
    RingLibCurl, 1001
Get String Length
    Strings, 108
Get Substring from position to end
    Strings, 111
GetArch() Function
    System Functions, 153
getattribute()
    Reflection and Meta-programming, 263
GetChar()
    Getting Input, 79
Getfilesize()
    Files, 146
getnumber()
    Getting Started - Third Style, 56
Getpathtype()
    Files, 145
getpointer()
    Low Level Functions, 1522
getstring()
    Getting Started - Third Style, 55
Getter and Setter Methods
    Frequently Asked Questions, 1940
Getting Input
    GetChar(), 79
    Getting Started - First Style, 48
    Getting Started - Second Style, 51
    Getting Started - Third Style, 54
    Give Command, 79
    Input(), 80
    Introduction, 78
Getting Started
    tilengine, 870
Getting Started - First Style
    Create Executable File, 47

Getting Input, 48
Hello World, 47
Introduction, 46
Multi-Line literals, 47
No Explicit End For Statements, 48
Not Case-Sensitive, 47
Run the program, 47
Using ? to print expression then new line, 48
Writing Comments, 49
Getting Started - Second Style
Create Executable File, 50
Getting Input, 51
Hello World, 50
Introduction, 49
Multi-Line literals, 50
No Explicit End For Statements, 51
Not Case-Sensitive, 50
Run the program, 50
Writing Comments, 51
Getting Started - Third Style
Create Executable File, 53
getnumber(), 56
getstring(), 55
Getting Input, 54
Hello World, 53
Introduction, 52
Multi-Line literals, 53
No Explicit End For Statements, 54
Not Case-Sensitive, 53
print(), 55
Print2Str() Function, 55
Run the program, 53
Writing Comments, 54
Give Command
Getting Input, 79
globals()
Reflection and Meta-programming, 249
Go Game
What is new in Ring 1.12?, 1797
Goal of including the "Main" function in Ring
Frequently Asked Questions, 1936
Google API Shortener Application
Applications developed in a few hours, 10
Gradient
Web Development (*CGI Library*), 619
Grant Ring files execution permission
Web Development (*CGI Library*), 595
Graphics and Game Programming
Drawing, Animation and Input, 657
Introduction, 656
Playing Sound, 665
Scaling and Rotating Images, 666
Threads, 668
Transparent Image, 667
TrueType Fonts, 664
Graphics Library Bindings
Game Engine for 2D Games, 672

H

Hash Functions
Web Development (*CGI Library*), 613
HashTable Class
Stdlib Classes, 374
Hassouna Course Samples
What is new in Ring 1.13?, 1818
Hello World
Getting Started - First Style, 47
Getting Started - Second Style, 50
Getting Started - Third Style, 53
libui, 1018
Tutorial: Ring Extensions in C/C++, 1544
Hello World program
Deploying Web Applications in the Cloud, 655
Hello World Program using the Web Library
Web Development (*CGI Library*), 596
Heroku (*Better support*)
What is new in Ring 1.18, 1896
What is new in Ring 1.15?, 1858
Hex()
Data Type, 128
Hex2str()
Data Type, 129
Higher-order Functions
Functional Programming, 244
Histogram
libui, 1037
History
Introduction, 28
Natural Language Programming, 276
Hours Counter application
What is new in Ring 1.19, 1904
How can I disable maximize button and resize window?
Frequently Asked Questions, 1955
How Ring find a functions and methods?
Scope Rules for Functions and Methods, 312
How Ring find the Variable?
Scope Rules, 296
How Ring2EXE works?
Distributing Ring Application using Ring2EXE, 931
How to add Combobox and other elements to the cells of a QTableWidget?
Frequently Asked Questions, 1959

How to Close a window then displaying another one?
Frequently Asked Questions, 1954
How to contribute?
Applications, 1975
Compiler and Virtual Machine (*VM*), 1975
Documentation, 1974
Editors Support, 1975
Extensions in C/C++, 1975
Ideas and suggestions, 1976
Introduction, 1973
Libraries in Ring, 1975
Samples, 1975
Special thanks to contributors, 1974
Testing, 1974
How to create a Modal Window?
Frequently Asked Questions, 1954
How to create an array of buttons in GUI applications?
Frequently Asked Questions, 1953
How to extend RingQt and add more classes?
Frequently Asked Questions, 1957
How to get the current source file path?
Frequently Asked Questions, 1944
How to get the file size using ftell() and fseek() functions?
Frequently Asked Questions, 1943
How to insert an item to the first position in the list?
Frequently Asked Questions, 1950
How to perform some manipulations on selected cells in QTableWidget?
Frequently Asked Questions, 1960
How to play?
The Gold Magic 800 Game, 866
How to print keys or values only in List/Dictionary?
Frequently Asked Questions, 1945
How to print lists that contains objects?
Frequently Asked Questions, 1950
How to print new lines and other characters?
Frequently Asked Questions, 1950
How to use many source code files in the project?
Frequently Asked Questions, 1947
How to use NULL and ISNULL() function?
Frequently Asked Questions, 1948
How to use SQLite using ODBC?
Frequently Asked Questions, 1961
HTML Lists
Web Development (*CGI Library*), 616
HTML Special Characters
Web Development (*CGI Library*), 612
HTML Tables
Web Development (*CGI Library*), 618
HtmlPage Class
Web Development (*CGI Library*), 625, 652
HTTP Get Example
Web Development (*CGI Library*), 597
HTTP POST Example
Web Development (*CGI Library*), 602
HTTPLib Library
What is new in Ring 1.17, 1876

I

Ideas and suggestions
How to contribute?, 1976
If Statement
Control Structures - Second Style, 70
Control Structures - Third Style, 74
Ignore new lines after keywords
Syntax Flexibility, 327
Image Drawing
Using RingRayLib, 765
Image Generation
Using RingRayLib, 767
Image Pixel application
What is new in Ring 1.19, 1903
Implicit Conversion
Variables, 58
Important Information about Ring2EXE
Distributing Ring Application using Ring2EXE, 939
ImportPackage() Function
What is new in Ring 1.18, 1896
importpackage() Function
Reflection and Meta-programming, 267
Increment List Items
Tutorial: Ring Extensions in C/C++, 1550
Inheritance
Object Oriented Programming, 233
Inheritance from GUI Classes
Desktop, WebAssembly and Mobile Development, 462
Innovative
Applications developed in a few hours, 22
Input Keys
Using RingRayLib, 731
Input Mouse
Using RingRayLib, 732
Input Multi-touch
Using RingRayLib, 738
Input()
Getting Input, 80
Insert()
Lists, 100
Install Qt for Android

Building RingQt Applications for Mobile, 578
Installing Packages
Ring Package Manager (*RingPM*), 942
Interactive Debugger
The Trace Library and the Interactive Debugger, 335
Interface to graphics library
Game Engine for 2D Games, 672
Internet Class
Stdlib Classes, 392
Introduction
Deploying Web Applications in the Cloud, 653
Features, 28
History, 28
Introduction, 24
Motivation (*1*), 25
Motivation (*2*), 26
Motivation (*3*), 26
Motivation (*4*), 26
Motivation (*5*), 27
Performance Tips, 905
Ring and other languages, 27
RingStbImage Functions Reference, 1152
Using CSVLib, 974
Using HTTPLib, 979
Using JSONLib, 977
Using References, 1506
Using RingRayLib, 730
Using TokensLib, 995
Using ZeroLib, 951
Is Ring some sort of an improvement of PHP?
Frequently Asked Questions, 1931
IsAlNum()
Data Type, 123
IsAlpha()
Data Type, 123
IsAndroid() Function
System Functions, 153
isappcompiled()
Stdlib Functions, 345
isattribute()
Reflection and Meta-programming, 259
IsBetween()
Stdlib Functions, 364
iscfunction()
Reflection and Meta-programming, 252
isclass()
Reflection and Meta-programming, 255
IsCntrl()
Data Type, 124
IsDigit()
Data Type, 124
IsFreeBSD() Function
System Functions, 152
isfunction()
Reflection and Meta-programming, 252
isglobal()
Reflection and Meta-programming, 251
IsGraph()
Data Type, 124
isleapyear()
Stdlib Functions, 356
IsLinux() Function
System Functions, 152
IsList()
Data Type, 122
IsListContainsItems()
Stdlib Functions, 364
islocal()
Reflection and Meta-programming, 251
IsLower()
Data Type, 125
IsMacOSX() Function
System Functions, 152
ismainsourcefile()
Stdlib Functions, 359
ismethod()
Reflection and Meta-programming, 260
IsMSDOS() Function
System Functions, 151
IsNULL()
Data Type, 122
IsNumber()
Data Type, 121
isobject()
Reflection and Meta-programming, 258
ispackage()
Reflection and Meta-programming, 254
ispackagesclass()
Reflection and Meta-programming, 256
ispalindrome()
Stdlib Functions, 355
ispointer()
Low Level Functions, 1521
isprime()
Stdlib Functions, 351
IsPrint()
Data Type, 125
isprivateattribute()
Reflection and Meta-programming, 259
isprivatemethod()
Reflection and Meta-programming, 261
IsPunct()
Data Type, 125
IsSpace()
Data Type, 126

isspecial()
    Stdlib Functions, 349
IsString()
    Data Type, 121
IsUnix() Function
    System Functions, 152
IsUpper()
    Data Type, 126
isvowel()
    Stdlib Functions, 350
IsWindows() Function
    System Functions, 151
IsWindows64() Function
    System Functions, 152
IsXdigit()
    Data Type, 126

J

JSONLib Library
    What is new in Ring 1.17, 1875
Julian Day Calendar application
    What is new in Ring 1.12?, 1800
JustFileName()
    Stdlib Functions, 346
JustFilePath()
    Stdlib Functions, 346

K

Key Press
    Using Qt3D, 535
Keyboard Events and Colors
    Using RingOpenGL and RingFreeGLUT for 3D Graphics, 795
Keyboard Shortcuts
    Form Designer, 515
KeyPress and Mouse Move Events
    Desktop, WebAssembly and Mobile Development, 457
Knight Tour Game
    What is new in Ring 1.10?, 1772

L

Language Design
    Compact Syntax, 1617
    Data Representation, 1616
    Define Declarative Languages, 1620
    Define Natural Statements, 1618
    Designed for a clear goal, 1614
    Encourage Organization, 1616
    Fast Enough For Many Applications, 1624
    Introduction, 1613
    No Global Interpreter (VM) Lock - No GIL, 1624
    Simple, 1614
    Smart Garbage Collector, 1623
    Transparent Implementation, 1621
    Trying to be natural, 1615
    Visual Implementation, 1622
    Why Ring?, 1614
Language Functions
    Language Specification, 1979
Language Grammar
    Language Specification, 1987
Language Keywords
    Language Specification, 1977
Language Shell
    Demo Programs, 163
Language Specification
    Compiler Errors, 1984
    Environment Errors, 1986
    Introduction, 1976
    Language Functions, 1979
    Language Grammar, 1987
    Language Keywords, 1977
    Runtime Errors, 1985
    Virtual Machine Instructions, 1988
Large Files Hash
    Security and Internet Functions, 221
Laser Game
    What is new in Ring 1.13?, 1810
lcm()
    Stdlib Functions, 353
Left()
    Strings, 109
Libraries in Ring
    How to contribute?, 1975
Library Source Code
    bignumber, 973
Library Usage with GUI Applications
    Objects Library for RingQt Application, 564
libui
    Control Gallery, 1021
    Draw Gradient, 1035
    Drawing Sample, 1034
    Hello World, 1018
    Histogram, 1037
    Introduction, 1017
    More Drawing Samples, 1048
    Say Hello, 1020
    Say Something, 1028
    Text Drawing, 1042
    Using the Menubar, 1030
Light GUILib
    What is new in Ring 1.16?, 1862
linecount()
    Stdlib Functions, 350
Lines()

Strings, 110
Linked list
Using References, 1509
List Class
Stdlib Classes, 370
List index start from 1
Frequently Asked Questions, 1937
List of changes and new features
What is new in Ring 1.10?, 1770
What is new in Ring 1.11?, 1781
What is new in Ring 1.12?, 1797
What is new in Ring 1.13?, 1808
What is new in Ring 1.14?, 1828
What is new in Ring 1.15?, 1850
What is new in Ring 1.16?, 1862
What is new in Ring 1.17?, 1869
What is new in Ring 1.18?, 1885
What is new in Ring 1.19?, 1903
What is new in Ring 1.1?, 1625
What is new in Ring 1.2?, 1636
What is new in Ring 1.3?, 1642
What is new in Ring 1.4?, 1656
What is new in Ring 1.5?, 1669
What is new in Ring 1.6?, 1718
What is new in Ring 1.7?, 1729
What is new in Ring 1.8?, 1738
What is new in Ring 1.9?, 1752
list of functions
Mathematical Functions, 130
List2Bytes()
Using FastPro, 1499
List2Code() Function
Stdlib Functions, 363
list2file()
Stdlib Functions, 352
ListAllFiles() Function
Stdlib Functions, 361
Lists
Access List Items by String Index, 103
Add Items, 96
Copy Lists, 101
Create Lists, 96
Create Multi Dimension Array Using List and Recursion, 105
Delete Item From List, 97
First-Class Lists, 101
Get List Item, 97
Get List Size, 97
Insert(), 100
Introduction, 95
Nested Lists, 100
Passing Lists to Functions, 102
Passing Parameters Arguments Using List Array, 104
Passing Parameters or Arguments Using List, 103
Return as List or Hash Table, 104
Reverse(), 100
Search, 98
Set List Item, 98
Sort(), 99
Swap Items, 107
Using Lists during definition, 102
Load Again
Program Structure, 94
Load Again Command
What is new in Ring 1.12?, 1802
Load Package
Program Structure, 94
Load Syntax Files
Syntax Flexibility, 318
Loading Files
Code Generator, 1609
Loading the library
bignumber, 971
PostgreSQL, 189
Loading the Trace library
The Trace Library and the Interactive Debugger, 334
Local Tests
Deploying Web Applications in the Cloud, 655
locals()
Reflection and Meta-programming, 249
Location
Create your first extension, 1612
Logical Operators
Operators, 59
Loop Command
Control Structures - First Style, 67
Looping
Control Structures, 75
Control Structures - First Style, 64
Control Structures - Second Style, 71
Low Level Functions
callgc(), 1516
Example - The Trace Library, 1537
Example - Using the Trace Functions, 1533
getpointer(), 1522
Introduction, 1515
ispointer(), 1521
memcpy(), 1524
nullpointer(), 1518
object2pointer(), 1519
pointer2object(), 1519
pointer2string(), 1523
ptrcmp(), 1521
ringvm_callfunc(), 1533

RingVM_CallList(), 1529
RingVM_CFunctionsList(), 1524
RingVM_ClassesList(), 1525
RingVM_CodeList(), 1542
ringvm_evalinscope(), 1532
RingVM_FilesList(), 1530
RingVM_FunctionsList(), 1524
ringvm_give() function, 1542
ringvm_hideerrorMsg(), 1532
ringvm_info(), 1542
ringvm_ismempool(), 1542
RingVM_MemoryList(), 1527
RingVM_PackagesList(), 1526
ringvm_passerror(), 1532
ringvm_runcode(), 1542
ringvm_scopescount(), 1532
ringvm_see() function, 1541
ringvm_settrace(), 1531
ringvm_tracedata(), 1531
ringvm_traceevent(), 1531
ringvm_tracefunc(), 1532
setpointer(), 1522
space(), 1518
varptr(), 1517

M

Magic Balls Game
What is new in Ring 1.13?, 1811
Main Function
Functions - First Style, 82
Functions - Second Style, 87
Functions - Third Style, 91
Main Menu
Demo Programs, 164
Make a Cube using RingOpenGL and RingFreeGLUT
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 836
makedir()
Stdlib Functions, 360
Managed Classes
Code Generator, 1609
Many Cubes
Using RingOpenGL and RingAllegro for 3D Graphics, 846
Many Objects
Using Qt3D, 553
Map
Game Engine for 2D Games, 691
Map Class
Game Engine for 2D Games, 677
Map Events
Game Engine for 2D Games, 692
map()
Stdlib Functions, 347
Matching Game
What is new in Ring 1.13?, 1813
Math Class
Stdlib Classes, 376
Mathematical Functions
Decimals(), 134
Example, 130
Introduction, 129
list of functions, 130
Random(), 132
SRandom(), 133
Unsigned(), 133
Using _ in numbers, 135
Using f after numbers, 135
MatrixLib Library
What is new in Ring 1.18, 1899
matrixmulti()
Stdlib Functions, 356
matrixtrans()
Stdlib Functions, 357
Maze Game
What is new in Ring 1.11?, 1784
MD5()
Security and Internet Functions, 198
MDI Windows Sample
What is new in Ring 1.16?, 1867
memcpy()
Low Level Functions, 1524
Memory Functions
Extension, 1584
Menu Events
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 810
Menubar and StyleSheet Example
Desktop, WebAssembly and Mobile Development, 432
Menubar Designer
Form Designer, 515
Merge binary characters
Strings, 113
What is new in Ring 1.8?, 1746
Merge Two Files
What is new in Ring 1.17, 1869
mergemethods()
Reflection and Meta-programming, 266
Methods to use Events with Events Filter
Desktop, WebAssembly and Mobile Development, 502
methods()
Reflection and Meta-programming, 258
Minesweeper Game
What is new in Ring 1.10?, 1771
Misc Operators

Operators, 61
Model
Using Qt3D, 547
Model Texture
Using Qt3D, 549
Module Organization
Extension, 1571
MoneyBoxes Game
What is new in Ring 1.13?, 1812
More 3D Samples
Using RingOpenGL and RingAllegro for 3D Graphics, 863
What is new in Ring 1.11?, 1781
What is new in Ring 1.8?, 1741
More Beautiful Code
Declarative Programming, 275
More Drawing Samples
libui, 1048
More Examples
Threads, 1014
More Functions
What is new in Ring 1.14?, 1843
More Games
What is new in Ring 1.10?, 1776
What is new in Ring 1.9?, 1754
More Improvements
What is new in Ring 1.17, 1882
What is new in Ring 1.18, 1900
What is new in Ring 1.19, 1915
What is new in Ring 1.10?, 1779
What is new in Ring 1.11?, 1795
What is new in Ring 1.12?, 1806
What is new in Ring 1.13?, 1824
What is new in Ring 1.14?, 1848
What is new in Ring 1.15?, 1860
What is new in Ring 1.16?, 1867
What is new in Ring 1.9?, 1768
More Low Level Functions
What is new in Ring 1.18, 1897
What is new in Ring 1.13?, 1823
More Samples
tilengine, 871
Using HTTPLib, 986
What is new in Ring 1.18, 1889
What is new in Ring 1.19, 1905
What is new in Ring 1.14?, 1831
What is new in Ring 1.15?, 1859
More Samples and Tests
Form Designer, 520
More Syntax Flexibility
What is new in Ring 1.5?, 1704
Motivation (*1*)
Introduction, 25
Motivation (*2*)
Introduction, 26
Motivation (*3*)
Introduction, 26
Motivation (*4*)
Introduction, 26
Motivation (*5*)
Introduction, 27
Mouse Events
RingLibSDL, 1095
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 805
Mouse Wheel
Using RingRayLib, 737
Movable Label Example
Desktop, WebAssembly and Mobile Development, 452
Moving Objects using the Mouse
Desktop, WebAssembly and Mobile Development, 458
Moving Text
Game Engine for 2D Games, 679
Multi-language applications
Form Designer Translation, 575
Forms Translation, 576
Introduction, 573
Using String2Constant, 574
Multi-Line literals
Getting Started - First Style, 47
Getting Started - Second Style, 50
Getting Started - Third Style, 53
MurmurHash1 functions
RingMurmurHash Functions Reference, 1067
MurmurHash2 functions
RingMurmurHash Functions Reference, 1067
MurmurHash3 functions
RingMurmurHash Functions Reference, 1067
Music Playing
Using RingRayLib, 762
MyCoffee (*Web Application*)
What is new in Ring 1.14?, 1828
MySQL Class
Stdlib Classes, 387
MySQL Functions
Create Database, 178
Create Table and Insert Data, 178
Introduction, 175
MySQL_AutoCommit(), 184
MySQL_Close(), 177
MySQL_Columns(), 181
MySQL_Commit(), 184
MySQL_Connect(), 177
MySQL_Error(), 177
MySQL_Escape_String(), 183
MySQL_Info(), 176

MySQL_Init(), 177
MySQL_Inser_ID(), 179
MySQL_Next_Result(), 180
MySQL_Query(), 177
MySQL_Result(), 180
MySQL_Result2(), 182
MySQL_Rollback(), 184
Print Query Result, 180
Restore Image From The Database, 183
Save Image Inside the Database, 183
Transaction Example, 184
MySQL_AutoCommit()
MySQL Functions, 184
MySQL_Close()
MySQL Functions, 177
MySQL_Columns()
MySQL Functions, 181
MySQL_Columns() Implementation
Extension, 1581
MySQL_Commit()
MySQL Functions, 184
MySQL_Connect()
MySQL Functions, 177
MySQL_Error()
MySQL Functions, 177
MySQL_Escape_String()
MySQL Functions, 183
MySQL_Info()
MySQL Functions, 176
MySQL_Init()
MySQL Functions, 177
MySQL_Inser_ID()
MySQL Functions, 179
MySQL_Next_Result()
MySQL Functions, 180
MySQL_Query()
MySQL Functions, 177
MySQL_Result()
MySQL Functions, 180
MySQL_Result2()
MySQL Functions, 182
MySQL_Rollback()
MySQL Functions, 184

N

Natural Language Programming
BraceError() Method, 284
BraceExprEval Method, 282
BraceStart and BraceEnd Methods, 282
Change the '=' operator to 'is', 279
Change the Ring Keyword 'And', 277
Change the Ring Operator '+', 279
Clean Natural Code, 285
Example, 276
History, 276
Introduction, 275
Real Natural Code, 283
Using Eval() with our Natural Code, 280
Natural Library - Demo Program
Using the Natural Library, 287
Nested Lists
Lists, 100
Nested Methods Call in Separate Lines
What is new in Ring 1.16?, 1863
Nested Ref()
Using References, 1513
New Arabic Book
What is new in Ring 1.15?, 1850
New Book by Apress
What is new in Ring 1.13?, 1808
New Classes names - Index Start from 1
Desktop, WebAssembly and Mobile
Development, 504
New Command: Load Package
What is new in Ring 1.7?, 1729
New From
Object Oriented Programming, 241
What is new in Ring 1.18, 1896
New Functions
What is new in Ring 1.2?, 1636
What is new in Ring 1.3?, 1651
New Functions and Changes
What is new in Ring 1.1?, 1628
New Game : Gold Magic 800
What is new in Ring 1.9?, 1752
New Style to Ring Notepad
What is new in Ring 1.4?, 1659
New Tool: Folder2qrc
What is new in Ring 1.6?, 1720
New Tool: Ring2EXE
What is new in Ring 1.6?, 1719
No Explicit End For Statements
Getting Started - First Style, 48
Getting Started - Second Style, 51
Getting Started - Third Style, 54
No Global Interpreter (VM) Lock - No GIL
Language Design, 1624
No Run
Command Line Options, 925
NofProcessors() Function
System Functions, 157
Not Case-Sensitive
Getting Started - First Style, 47
Getting Started - Second Style, 50
Getting Started - Third Style, 53
Notepad Application
Desktop, WebAssembly and Mobile
Development, 476

Notes to extensions creators
What is new in Ring 1.8?, 1751
Nothing() function
Reflection and Meta-programming, 268
What is new in Ring 1.19, 1908
nullpointer()
Low Level Functions, 1518
Number()
Data Type, 127
Numbers and Bytes
Files, 148

O

Object and Drawing
Game Engine for 2D Games, 694
Object Library Source Code
Objects Library for RingQt Application, 573
Object Oriented Programming
Access Objects Using Braces, 226
Classes and Objects, 224
Composition, 227
Default value for object attributes, 241
Dynamic Attributes, 234
Find() and List of Objects, 235
Inheritance, 233
Introduction, 223
New From, 241
Operator Overloading, 230
Packages, 234
Printing Objects, 235
Private Attributes and Methods, 229
Setter and Getter, 228
Sort() and List of Objects, 236
Using Self.Attribute, 238
Using This in the class region as Self, 239
Using This.Attribute and This.Method(), 239
Object Picker
Using Qt3D, 537
object2pointer()
Low Level Functions, 1519
objectid()
Reflection and Meta-programming, 257
Objects inside lists
Declarative Programming, 270
Objects Library for RingQt
What is new in Ring 1.2?, 1638
Objects Library for RingQt Application
Example, 565
Introduction, 563
Library Usage with GUI Applications, 564
Object Library Source Code, 573
openWindowAndLink() Function, 567, 573
openWindowInPackages() Function, 568
Using ObjectID() and GetWindowByID(), 568
Using ObjectID() and OpenWindowAndLink(), 571
Objects Order and Layouts
Form Designer, 517
ODBC Class
Stdlib Classes, 386
ODBC Functions
Execute Query and Print Result, 171
Introduction, 166
odbc_autocommit(), 173
odbc_close(), 168
odbc_colcount(), 171
odbc_columns(), 172
odbc_commit(), 173
odbc_connect(), 170
odbc_datasources(), 168
odbc_disconnect(), 170
odbc_drivers(), 167
odbc_execute(), 170
odbc_fetch(), 171
odbc_getdata(), 171
odbc_init(), 167
odbc_rollback(), 174
odbc_tables(), 172
Open and Close Connection, 170
Print List of ODBC Data Sources, 169
Print List of ODBC Drivers, 168
Save and Restore Images, 175
Transactions and Using Commit and Rollback, 174
odbc_autocommit()
ODBC Functions, 173
odbc_close()
ODBC Functions, 168
odbc_colcount()
ODBC Functions, 171
odbc_columns()
ODBC Functions, 172
odbc_commit()
ODBC Functions, 173
odbc_connect()
ODBC Functions, 170
odbc_datasources()
ODBC Functions, 168
odbc_disconnect()
ODBC Functions, 170
odbc_drivers()
ODBC Functions, 167
odbc_execute()
ODBC Functions, 170
odbc_fetch()

ODBC Functions, 171
odbc_getdata()
ODBC Functions, 171
odbc_init()
ODBC Functions, 167
odbc_rollback()
ODBC Functions, 174
odbc_tables()
ODBC Functions, 172
Online Applications
Building RingQt Applications for WebAssembly, 593
Open and Close Connection
ODBC Functions, 170
openssl_version()
Security and Internet Functions, 221
openssl_versiontext()
Security and Internet Functions, 220
openWindowAndLink() Function
Objects Library for RingQt Application, 567, 573
openWindowInPackages() Function
Objects Library for RingQt Application, 568
Operator Overloading
Object Oriented Programming, 230
Operators
Arithmetic Operators, 59
Assignment Operators, 60
Bitwise Operators, 60
Introduction, 58
Logical Operators, 59
Misc Operators, 61
Operators Precedence, 61
Relational Operators, 59
Using the Natural Library, 292
Operators Precedence
Operators, 61
OptionalFunc() function
Reflection and Meta-programming, 268
What is new in Ring 1.19, 1908
Options
Distributing Ring Application using Ring2EXE, 932
OSCopyFile() Function
Stdlib Functions, 363
OSCopyFolder() Function
Stdlib Functions, 362
OSCreateOpenFolder() Function
Stdlib Functions, 362
OSDeleteFile() Function
Stdlib Functions, 363
OSDeleteFolder() Function
Stdlib Functions, 362
OSRenameFile() Function
Stdlib Functions, 363
Other Widgets Events
Desktop, WebAssembly and Mobile Development, 436

P

packagename()
Reflection and Meta-programming, 267
Packages
Object Oriented Programming, 234
packages()
Reflection and Meta-programming, 253
packagesclasses()
Reflection and Meta-programming, 256
Page Class
Web Development (*CGI Library*), 647
Pairs Game
What is new in Ring 1.13?, 1814
Pass Error
The Trace Library and the Interactive Debugger, 335
Passing Lists to Functions
Lists, 102
Passing lists to functions
Using References, 1512
Passing Parameters Arguments Using List Array
Lists, 104
Passing Parameters or Arguments Using List
Lists, 103
Performance
Command Line Options, 926
Performance Tips
Arithmetic Operations, 908
Calling Functions and Methods, 909
Creating Lists, 907
Introduction, 904, 905
Using len() and For Loops, 908
permutation()
Stdlib Functions, 357
Perror()
Files, 143
Philosophy behind data types in Ring
Frequently Asked Questions, 1934
Planetary Orbits application
What is new in Ring 1.19, 1904
Play Sound
RingLibSDL, 1097
Playing Sound
Desktop, WebAssembly and Mobile Development, 450
Game Engine for 2D Games, 681
Graphics and Game Programming, 665

Poetry Analysis
What is new in Ring 1.17, 1870
pointer2object()
Low Level Functions, 1519
pointer2string()
Low Level Functions, 1523
Pong Game
What is new in Ring 1.10?, 1774
Post Request
RingLibCurl, 999
PostgreSQL
Examples, 189
Introduction, 188
Loading the library, 189
RingPostgreSQL Constants, 192
RingPostgreSQL Functions, 193
PostgreSQL Class
Stdlib Classes, 390
Practical
Applications developed in a few hours, 23
Prefix for Functions Names
Code Generator, 1590
PrevFileName() Function
System Functions, 156
Print files in Zip file
RingZip, 1063
Print List of ODBC Data Sources
ODBC Functions, 169
Print List of ODBC Drivers
ODBC Functions, 168
Print Query Result
MySQL Functions, 180
print()
Getting Started - Third Style, 55
Print2Str() Function
Getting Started - Third Style, 55
Printing Constants
Using HTTPLib, 980
Printing Final Intermediate Code
Command Line Options, 920
Printing Instruction Operation Code
Command Line Options, 926
Printing Intermediate Code
Command Line Options, 918
Printing List of Installed Packages
Ring Package Manager (*RingPM*), 943
Printing Objects
Object Oriented Programming, 235
Printing Rules
Command Line Options, 913
Printing Tokens
Command Line Options, 911
Printing using QPrinter
Desktop, WebAssembly and Mobile Development, 445
Private Attributes and Methods
Object Oriented Programming, 229
prodlist()
Stdlib Functions, 354
Program Structure
Introduction, 92
Load Again, 94
Load Package, 94
Source Code File Sections, 93
Using Many Source Code Files, 93
Project Folder
Building Games For Android, 727
Project Layers
Game Engine for 2D Games, 672
Protecting the Source Code
Distributing Ring Application, 928
ptrcmp()
Low Level Functions, 1521
Pure Functions
Functional Programming, 243
puts()
Stdlib Functions, 55

Q

QAbstractAspect Class
RingQt Classes Reference, 1207
QAbstractButton Class
RingQt Classes Reference, 1208
QAbstractCameraController Class
RingQt Classes Reference, 1209
QAbstractItemView Class
RingQt Classes Reference, 1209
QAbstractPrintDialog Class
RingQt Classes Reference, 1211
QAbstractScrollArea Class
RingQt Classes Reference, 1211
QAbstractSeries Class
RingQt Classes Reference, 1212
QAbstractSlider Class
RingQt Classes Reference, 1213
QAbstractSocket Class
RingQt Classes Reference, 1213
QAbstractSpinBox Class
RingQt Classes Reference, 1215
QAction Class
RingQt Classes Reference, 1216
QAllEvents Class
RingQt Classes Reference, 1217
QApp Class
RingQt Classes Reference, 1222
QAreaLegendMarker Class
RingQt Classes Reference, 1222

QAreaSeries Class
RingQt Classes Reference, 1222
QAspectEngine Class
RingQt Classes Reference, 1224
QAxBase Class
RingQt Classes Reference, 1224
QAxObject Class
RingQt Classes Reference, 1225
QAxWidget Class
RingQt Classes Reference, 1226
QAxWidget2 Class
RingQt Classes Reference, 1226
QBarCategoryAxis Class
RingQt Classes Reference, 1226
QBarLegendMarker Class
RingQt Classes Reference, 1227
QBarSeries Class
RingQt Classes Reference, 1227
QBarSet Class
RingQt Classes Reference, 1227
QBitmap Class
RingQt Classes Reference, 1229
QBluetoothAddress Class
RingQt Classes Reference, 1229
QBluetoothDeviceDiscoveryAgent Class
RingQt Classes Reference, 1229
QBluetoothDeviceInfo Class
RingQt Classes Reference, 1230
QBluetoothHostInfo Class
RingQt Classes Reference, 1230
QBluetoothLocalDevice Class
RingQt Classes Reference, 1231
QBluetoothServer Class
RingQt Classes Reference, 1232
QBluetoothServiceDiscoveryAgent Class
RingQt Classes Reference, 1232
QBluetoothServiceInfo Class
RingQt Classes Reference, 1233
QBluetoothSocket Class
RingQt Classes Reference, 1234
QBluetoothTransferManager Class
RingQt Classes Reference, 1235
QBluetoothTransferReply Class
RingQt Classes Reference, 1235
QBluetoothTransferRequest Class
RingQt Classes Reference, 1235
QBluetoothUuid Class
RingQt Classes Reference, 1236
QBoxLayout Class
RingQt Classes Reference, 1236
QBoxPlotLegendMarker Class
RingQt Classes Reference, 1237
QBoxPlotSeries Class
RingQt Classes Reference, 1237
QBoxSet Class
RingQt Classes Reference, 1238
QBrush Class
RingQt Classes Reference, 1239
QBuffer Class
RingQt Classes Reference, 1240
QButtonGroup Class
RingQt Classes Reference, 1240
QByteArray Class
RingQt Classes Reference, 1241
QCalendarWidget Class
RingQt Classes Reference, 1243
QCamera Class
RingQt Classes Reference, 1244
QCameraImageCapture Class
RingQt Classes Reference, 1245
QCameraLens Class
RingQt Classes Reference, 1246
QCameraSelector Class
RingQt Classes Reference, 1246
QCameraViewfinder Class
RingQt Classes Reference, 1247
QCandlestickLegendMarker Class
RingQt Classes Reference, 1247
QCandlestickModelMapper Class
RingQt Classes Reference, 1247
QCandlestickSeries Class
RingQt Classes Reference, 1247
QCandlestickSet Class
RingQt Classes Reference, 1249
QCategoryAxis Class
RingQt Classes Reference, 1251
QChar Class
RingQt Classes Reference, 1251
QChart Class
RingQt Classes Reference, 1253
QChartView Class
RingQt Classes Reference, 1255
QCheckBox Class
RingQt Classes Reference, 1256
QChildEvent Class
RingQt Classes Reference, 1256
QClipboard Class
RingQt Classes Reference, 1256
QColor Class
RingQt Classes Reference, 1257
QColorDialog Class
RingQt Classes Reference, 1259
QComboBox Class
RingQt Classes Reference, 1260
QCompleter Class
RingQt Classes Reference, 1262
QCompleter2 Class
RingQt Classes Reference, 1263

QCompleter3 Class
RingQt Classes Reference, 1263
QCompleter4 Class
RingQt Classes Reference, 1263
QConeGeometry Class
RingQt Classes Reference, 1263
QConeMesh Class
RingQt Classes Reference, 1264
QCoreApplication Class
RingQt Classes Reference, 1265
QCuboidMesh Class
RingQt Classes Reference, 1266
QCullFace Class
RingQt Classes Reference, 1266
QCursor Class
RingQt Classes Reference, 1267
QCylinderMesh Class
RingQt Classes Reference, 1267
QDate Class
RingQt Classes Reference, 1267
QDateEdit Class
RingQt Classes Reference, 1268
QDateTime Class
RingQt Classes Reference, 1268
QDateTimeAxis Class
RingQt Classes Reference, 1269
QDateTimeEdit Class
RingQt Classes Reference, 1270
QDepthTest Class
RingQt Classes Reference, 1272
QDesktopServices Class
RingQt Classes Reference, 1272
QDesktopWidget Class
RingQt Classes Reference, 1272
QDial Class
RingQt Classes Reference, 1272
QDialog Class
RingQt Classes Reference, 1273
QDiffuseSpecularMaterial Class
RingQt Classes Reference, 1273
QDir Class
RingQt Classes Reference, 1274
QDockWidget Class
RingQt Classes Reference, 1276
QDrag Class
RingQt Classes Reference, 1277
QDragEnterEvent Class
RingQt Classes Reference, 1277
QDragLeaveEvent Class
RingQt Classes Reference, 1278
QDragMoveEvent Class
RingQt Classes Reference, 1278
QDropEvent Class
RingQt Classes Reference, 1278
QEffect Class
RingQt Classes Reference, 1279
QEntity Class
RingQt Classes Reference, 1279
QEvent Class
RingQt Classes Reference, 1279
QExtrudedTextMesh Class
RingQt Classes Reference, 1279
QFile Class
RingQt Classes Reference, 1280
QFile2 Class
RingQt Classes Reference, 1281
QFileDevice Class
RingQt Classes Reference, 1281
QFileDialog Class
RingQt Classes Reference, 1281
QFileInfo Class
RingQt Classes Reference, 1283
QFileSystemModel Class
RingQt Classes Reference, 1285
QFirstPersonCameraController Class
RingQt Classes Reference, 1286
QFont Class
RingQt Classes Reference, 1286
QFontDialog Class
RingQt Classes Reference, 1288
QFontMetrics Class
RingQt Classes Reference, 1289
QForwardRenderer Class
RingQt Classes Reference, 1290
QFrame Class
RingQt Classes Reference, 1290
QFrame2 Class
RingQt Classes Reference, 1291
QFrame3 Class
RingQt Classes Reference, 1291
QFrameAction Class
RingQt Classes Reference, 1291
QGeoAddress Class
RingQt Classes Reference, 1291
QGeoAreaMonitorInfo Class
RingQt Classes Reference, 1292
QGeoAreaMonitorSource Class
RingQt Classes Reference, 1292
QGeoCircle Class
RingQt Classes Reference, 1293
QGeoCoordinate Class
RingQt Classes Reference, 1293
QGeoPositionInfo Class
RingQt Classes Reference, 1293
QGeoPositionInfoSource Class
RingQt Classes Reference, 1294
QGeoRectangle Class
RingQt Classes Reference, 1294

QGeoSatelliteInfo Class
RingQt Classes Reference, 1295
QGeoSatelliteInfoSource Class
RingQt Classes Reference, 1296
QGeoShape Class
RingQt Classes Reference, 1296
QGoochMaterial Class
RingQt Classes Reference, 1296
QGradient Class
RingQt Classes Reference, 1296
QGraphicsScene Class
RingQt Classes Reference, 1297
QGraphicsVideoItem Class
RingQt Classes Reference, 1299
QGraphicsView Class
RingQt Classes Reference, 1300
QGridLayout Class
RingQt Classes Reference, 1302
QGuiApplication Class
RingQt Classes Reference, 1303
QHBarModelMapper Class
RingQt Classes Reference, 1305
QHBoxLayout Class
RingQt Classes Reference, 1306
QHBoxPlotModelMapper Class
RingQt Classes Reference, 1306
QHCandlestickModelMapper Class
RingQt Classes Reference, 1307
QHeaderView Class
RingQt Classes Reference, 1309
QHorizontalBarSeries Class
RingQt Classes Reference, 1312
QHorizontalPercentBarSeries Class
RingQt Classes Reference, 1312
QHorizontalStackedBarSeries Class
RingQt Classes Reference, 1312
QHostAddress Class
RingQt Classes Reference, 1312
QHostInfo Class
RingQt Classes Reference, 1313
QHPieModelMapper Class
RingQt Classes Reference, 1308
QHXYModelMapper Class
RingQt Classes Reference, 1309
QIcon Class
RingQt Classes Reference, 1314
QImage Class
RingQt Classes Reference, 1315
QInputAspect Class
RingQt Classes Reference, 1316
QInputDialog Class
RingQt Classes Reference, 1316
QIODevice Class
RingQt Classes Reference, 1313
QJsonArray Class
RingQt Classes Reference, 1318
QJsonDocument Class
RingQt Classes Reference, 1319
QJsonObject Class
RingQt Classes Reference, 1319
QJsonParseError Class
RingQt Classes Reference, 1320
QJsonValue Class
RingQt Classes Reference, 1320
QKeySequence Class
RingQt Classes Reference, 1320
QLabel Class
RingQt Classes Reference, 1321
QLayout Class
RingQt Classes Reference, 1322
QLCDNumber Class
RingQt Classes Reference, 1320
QLegend Class
RingQt Classes Reference, 1323
QLegendMarker Class
RingQt Classes Reference, 1324
QLinearGradient Class
RingQt Classes Reference, 1327
QLineEdit Class
RingQt Classes Reference, 1325
QLineEdit Events and QMessageBox
Desktop, WebAssembly and Mobile
Development, 434
QLineSeries Class
RingQt Classes Reference, 1327
QListView Class
RingQt Classes Reference, 1327
QListWidget Class
RingQt Classes Reference, 1328
QListWidgetItem Class
RingQt Classes Reference, 1330
QLocale Class
RingQt Classes Reference, 1331
QLogicAspect Class
RingQt Classes Reference, 1334
QLogValueAxis Class
RingQt Classes Reference, 1333
QMainWindow Class
RingQt Classes Reference, 1334
QMaterial Class
RingQt Classes Reference, 1335
QMatrix4x4 Class
RingQt Classes Reference, 1335
QMdiArea Class
RingQt Classes Reference, 1337
QMdiSubWindow Class
RingQt Classes Reference, 1338
QMediaObject Class

RingQt Classes Reference, 1338
QMediaPlayer Class
RingQt Classes Reference, 1339
QMediaPlaylist Class
RingQt Classes Reference, 1339
QMenu Class
RingQt Classes Reference, 1340
QMenuBar Class
RingQt Classes Reference, 1341
QMesh Class
RingQt Classes Reference, 1341
QMessageBox Class
RingQt Classes Reference, 1342
QMessagebox Example
Desktop, WebAssembly and Mobile Development, 453
QMetalRoughMaterial Class
RingQt Classes Reference, 1343
QMimeData Class
RingQt Classes Reference, 1343
QMorphPhongMaterial Class
RingQt Classes Reference, 1344
QMovie Class
RingQt Classes Reference, 1344
QMutex Class
RingQt Classes Reference, 1346
QMutexLocker Class
RingQt Classes Reference, 1346
QNetworkAccessManager Class
RingQt Classes Reference, 1346
QNetworkProxy Class
RingQt Classes Reference, 1347
QNetworkReply Class
RingQt Classes Reference, 1348
QNetworkRequest Class
RingQt Classes Reference, 1348
QNmeaPositionInfoSource Class
RingQt Classes Reference, 1349
QNode Class
RingQt Classes Reference, 1349
QObject Class
RingQt Classes Reference, 1350
QObjectPicker Class
RingQt Classes Reference, 1350
QOpenGLBuffer Class
RingQt Classes Reference, 1351
QOpenGLContext Class
RingQt Classes Reference, 1352
QOpenGLDebugLogger Class
RingQt Classes Reference, 1353
QOpenGLFramebufferObject Class
RingQt Classes Reference, 1354
QOpenGLFunctions Class
RingQt Classes Reference, 1354
QOpenGLFunctions_3_2_Core Class
RingQt Classes Reference, 1358
QOpenGLPaintDevice Class
RingQt Classes Reference, 1366
QOpenGLShader Class
RingQt Classes Reference, 1367
QOpenGLShaderProgram Class
RingQt Classes Reference, 1367
QOpenGLTexture Class
RingQt Classes Reference, 1371
QOpenGLTimerQuery Class
RingQt Classes Reference, 1373
QOpenGLVersionProfile Class
RingQt Classes Reference, 1373
QOpenGLVertexArrayObject Class
RingQt Classes Reference, 1374
QOpenGLWidget Class
RingQt Classes Reference, 1374
QOrbitCameraController Class
RingQt Classes Reference, 1375
QPageSetupDialog Class
RingQt Classes Reference, 1375
QPaintDevice Class
RingQt Classes Reference, 1375
QPainter Class
RingQt Classes Reference, 1376
QPainter2 Class
RingQt Classes Reference, 1378
QPainterPath Class
RingQt Classes Reference, 1378
QPen Class
RingQt Classes Reference, 1380
QPercentBarSeries Class
RingQt Classes Reference, 1381
QPerVertexColorMaterial Class
RingQt Classes Reference, 1380
QPhongMaterial Class
RingQt Classes Reference, 1381
QPicture Class
RingQt Classes Reference, 1381
QPieLegendMarker Class
RingQt Classes Reference, 1382
QPieSeries Class
RingQt Classes Reference, 1382
QPieSlice Class
RingQt Classes Reference, 1383
QPixmap Class
RingQt Classes Reference, 1385
QPixmap2 Class
RingQt Classes Reference, 1386
QPlainTextEdit Class
RingQt Classes Reference, 1387
QPlaneMesh Class
RingQt Classes Reference, 1389

QPoint Class
RingQt Classes Reference, 1389
QPointF Class
RingQt Classes Reference, 1390
QPointLight Class
RingQt Classes Reference, 1390
QPolarChart Class
RingQt Classes Reference, 1391
QPrintDialog Class
RingQt Classes Reference, 1391
QPrinter Class
RingQt Classes Reference, 1392
QPrinterInfo Class
RingQt Classes Reference, 1394
QPrintPreviewDialog Class
RingQt Classes Reference, 1391
QPrintPreviewWidget Class
RingQt Classes Reference, 1391
QProcess Class
RingQt Classes Reference, 1394
QProgressBar Class
RingQt Classes Reference, 1395
QPushButton Class
RingQt Classes Reference, 1396
QQmlEngine Class
RingQt Classes Reference, 1397
QQmlError Class
RingQt Classes Reference, 1398
QQuaternion Class
RingQt Classes Reference, 1398
QQuickView Class
RingQt Classes Reference, 1399
QQuickWidget Class
RingQt Classes Reference, 1399
QRadioButton Class
RingQt Classes Reference, 1400
QRect Class
RingQt Classes Reference, 1401
QRegion Class
RingQt Classes Reference, 1402
QRegularExpression Class
RingQt Classes Reference, 1403
QRegularExpressionMatch Class
RingQt Classes Reference, 1403
QRegularExpressionMatchIterator Class
RingQt Classes Reference, 1404
QRenderAspect Class
RingQt Classes Reference, 1404
QRenderPass Class
RingQt Classes Reference, 1405
QScatterSeries Class
RingQt Classes Reference, 1405
QSceneLoader Class
RingQt Classes Reference, 1406
QScreen Class
RingQt Classes Reference, 1406
QScrollArea Class
RingQt Classes Reference, 1407
QScrollBar Class
RingQt Classes Reference, 1407
QSerialPort Class
RingQt Classes Reference, 1407
QSerialPortInfo Class
RingQt Classes Reference, 1409
QSize Class
RingQt Classes Reference, 1409
QSkyboxEntity Class
RingQt Classes Reference, 1409
QSlider Class
RingQt Classes Reference, 1410
QSphereMesh Class
RingQt Classes Reference, 1410
QSpinBox Class
RingQt Classes Reference, 1411
QSplashScreen Class
RingQt Classes Reference, 1411
QSplineSeries Class
RingQt Classes Reference, 1412
QSplitter Class
RingQt Classes Reference, 1412
QSqlDatabase Class
RingQt Classes Reference, 1413
QSqlDriver Class
RingQt Classes Reference, 1414
QSqlDriverCreatorBase Class
RingQt Classes Reference, 1414
QSqlError Class
RingQt Classes Reference, 1414
QSqlField Class
RingQt Classes Reference, 1415
QSqlIndex Class
RingQt Classes Reference, 1416
QSqlQuery Class
RingQt Classes Reference, 1416
QSqlRecord Class
RingQt Classes Reference, 1417
QStackedBarSeries Class
RingQt Classes Reference, 1418
QStackedWidget Class
RingQt Classes Reference, 1418
QStandardPaths Class
RingQt Classes Reference, 1418
QStatusBar Class
RingQt Classes Reference, 1419
QString2 Class
RingQt Classes Reference, 1419
QStringList Class
RingQt Classes Reference, 1421

QStringRef Class
    RingQt Classes Reference, 1422
QStyle Class
    RingQt Classes Reference, 1423
QSurfaceFormat Class
    RingQt Classes Reference, 1425
QSystemTrayIcon Class
    RingQt Classes Reference, 1426
Qt Class Converter
    What is new in Ring 1.4?, 1664
Qt configuration file
    Code Generator, 1598
Qt3DCamera Class
    RingQt Classes Reference, 1496
Qt3DWindow Class
    RingQt Classes Reference, 1498
QTabBar Class
    RingQt Classes Reference, 1426
QTableView Class
    RingQt Classes Reference, 1429
QTableWidget Class
    RingQt Classes Reference, 1431
QTableWidgetItem Class
    RingQt Classes Reference, 1433
QTabWidget Class
    RingQt Classes Reference, 1428
QTcpServer Class
    RingQt Classes Reference, 1434
QTcpSocket Class
    RingQt Classes Reference, 1435
QTechnique Class
    RingQt Classes Reference, 1435
QTest Class
    RingQt Classes Reference, 1436
QText2DEntity Class
    RingQt Classes Reference, 1436
QTextBlock Class
    RingQt Classes Reference, 1436
QTextBrowser Class
    RingQt Classes Reference, 1437
QTextCharFormat Class
    RingQt Classes Reference, 1438
QTextCodec Class
    RingQt Classes Reference, 1440
QTextCursor Class
    RingQt Classes Reference, 1440
QTextDocument Class
    RingQt Classes Reference, 1442
QTextEdit Class
    RingQt Classes Reference, 1444
QTextOption Class
    RingQt Classes Reference, 1447
QTextStream Class
    RingQt Classes Reference, 1447
QTextStream2 Class
    RingQt Classes Reference, 1448
QTextStream3 Class
    RingQt Classes Reference, 1449
QTextStream4 Class
    RingQt Classes Reference, 1449
QTextStream5 Class
    RingQt Classes Reference, 1449
QTextToSpeech Class
    RingQt Classes Reference, 1449
QTextureLoader Class
    RingQt Classes Reference, 1450
QTextureMaterial Class
    RingQt Classes Reference, 1450
QThread Class
    RingQt Classes Reference, 1451
QThreadPool Class
    RingQt Classes Reference, 1451
QTime Class
    RingQt Classes Reference, 1452
QTimer Class
    RingQt Classes Reference, 1453
QToolBar Class
    RingQt Classes Reference, 1453
QToolButton Class
    RingQt Classes Reference, 1454
QTorusMesh Class
    RingQt Classes Reference, 1454
QTransform Class
    RingQt Classes Reference, 1455
QTreeView Class
    RingQt Classes Reference, 1456
QTreeWidget Class
    RingQt Classes Reference, 1458
QTreeWidgetItem Class
    RingQt Classes Reference, 1460
Queue Class
    Stdlib Classes, 373
Quotes about Ring
    Applications developed in a few hours, 2
QUrl Class
    RingQt Classes Reference, 1462
QUuid Class
    RingQt Classes Reference, 1463
QValueAxis Class
    RingQt Classes Reference, 1467
QVariant Class
    RingQt Classes Reference, 1469
QVariant2 Class
    RingQt Classes Reference, 1470
QVariant3 Class
    RingQt Classes Reference, 1470
QVariant4 Class
    RingQt Classes Reference, 1470

QVariant5 Class
    RingQt Classes Reference, 1470
QVariantDouble Class
    RingQt Classes Reference, 1471
QVariantFloat Class
    RingQt Classes Reference, 1471
QVariantInt Class
    RingQt Classes Reference, 1471
QVariantString Class
    RingQt Classes Reference, 1471
QVBarModelMapper Class
    RingQt Classes Reference, 1463
QVBoxLayout Class
    RingQt Classes Reference, 1464
QVBoxPlotModelMapper Class
    RingQt Classes Reference, 1464
QVCandlestickModelMapper Class
    RingQt Classes Reference, 1465
QVector2D Class
    RingQt Classes Reference, 1471
QVector3D Class
    RingQt Classes Reference, 1472
QVector4D Class
    RingQt Classes Reference, 1473
QVectorQVoice Class
    RingQt Classes Reference, 1473
QVideoWidget and QMediaPlayer
    Desktop, WebAssembly and Mobile
        Development, 428
QVideoWidget Class
    RingQt Classes Reference, 1474
QVideoWidgetControl Class
    RingQt Classes Reference, 1474
QViewport Class
    RingQt Classes Reference, 1475
QVoice Class
    RingQt Classes Reference, 1475
QVPieModelMapper Class
    RingQt Classes Reference, 1465
QVXYModelMapper Class
    RingQt Classes Reference, 1466
QWebEnginePage Class
    RingQt Classes Reference, 1475
QWebEngineView Class
    RingQt Classes Reference, 1478
QWebView Class
    RingQt Classes Reference, 1479, 1480
QWidget Class
    RingQt Classes Reference, 1480
QWindow Class
    RingQt Classes Reference, 1485
QXmlStreamAttribute Class
    RingQt Classes Reference, 1492
QXmlStreamAttributes Class
    RingQt Classes Reference, 1492
QXmlStreamEntityDeclaration Class
    RingQt Classes Reference, 1492
QXmlStreamEntityResolver Class
    RingQt Classes Reference, 1493
QXmlStreamNamespaceDeclaration Class
    RingQt Classes Reference, 1493
QXmlStreamNotationDeclaration Class
    RingQt Classes Reference, 1493
QXmlStreamReader Class
    RingQt Classes Reference, 1493
QXmlStreamWriter Class
    RingQt Classes Reference, 1495
QXYLegendMarker Class
    RingQt Classes Reference, 1489
QXYSeries Class
    RingQt Classes Reference, 1489

R

Raise()
    Eval() and Debugging, 160
RandBytes()
    Security and Internet Functions, 203
Random Image
    Web Development (*CGI Library*), 615
Random User application
    What is new in Ring 1.18, 1886
Random()
    Mathematical Functions, 132
RandomItem()
    Stdlib Functions, 366
RandomList()
    Stdlib Functions, 365
Read File using Read()
    Files, 138
Read Files application
    What is new in Ring 1.18, 1885
readline()
    Stdlib Functions, 358
Real Natural Code
    Natural Language Programming, 283
Rectangle Scaling
    Using RingRayLib, 760
Recursion
    Functions - First Style, 84
    Functions - Second Style, 88
    Functions - Third Style, 92
reduce()
    Stdlib Functions, 348
ref()
    Using References, 1506
Ref() and sub lists
    Using References, 1514
Ref() and temp. lists

Using References, 1513
Ref() in left side
Using References, 1514
refcount()
Using References, 1507
Reference
Threads, 1016
tilengine, 896
Reflection and Meta-programming
addattribute(), 261
addmethod(), 262
attributes(), 258
cfunctions(), 250
classes(), 254
classname(), 257
functions(), 250
getattribute(), 263
globals(), 249
importpackage() Function, 267
Introduction, 247
isattribute(), 259
iscfunction(), 252
isclass(), 255
isfunction(), 252
isglobal(), 251
islocal(), 251
ismethod(), 260
isobject(), 258
ispackage(), 254
ispackagesclass(), 256
isprivateattribute(), 259
isprivatemethod(), 261
locals(), 249
mergemethods(), 266
methods(), 258
Nothing() function, 268
objectid(), 257
OptionalFunc() function, 268
packagename(), 267
packages(), 253
packagesclasses(), 256
setattribute(), 265
Register New Functions
Code Generator, 1591
Regular Expressions
Desktop, WebAssembly and Mobile Development, 467
Using HTTPLib, 993
Relational Operators
Operators, 59
Remove Package
Ring Package Manager (*RingPM*), 944
Remove()
Files, 139
Rename()
Files, 139
Replicate List Items
Tutorial: Ring Extensions in C/C++, 1553
Resources
Introduction, 1993
Ring Group, 1994
Ring Language Website, 1994
Ring Team, 1994
Source Code, 1994
Restore Image From The Database
MySQL Functions, 183
Return as List or Hash Table
Lists, 104
Return object by reference
Declarative Programming, 271
Return Self by Reference
What is new in Ring 1.3?, 1651
Return Value
Extension, 1574
Functions - First Style, 83
Functions - Second Style, 87
Functions - Third Style, 91
Reverse()
Lists, 100
Rewind()
Files, 142
Right()
Strings, 109
Ring and other languages
Introduction, 27
RING API - list Functions
Extension, 1576
RING API - String Functions
Extension, 1579
Ring CGI Hello World Program
Web Development (*CGI Library*), 596
Ring Extension for Visual Studio Code
What is new in Ring 1.10?, 1777
Ring for MS-DOS
What is new in Ring 1.18, 1892
Ring for WebAssembly
What is new in Ring 1.13?, 1809
Ring for Windows 64bit
What is new in Ring 1.19, 1906
Ring Group
Resources, 1994
Ring Language Website
Resources, 1994
Ring mode for Emacs Editor
What is new in Ring 1.3?, 1649
Ring Notepad
Introduction, 32

Ring Notepad - Creating and running your first console application, 34
Ring Notepad - Creating and running your first game, 39
Ring Notepad - Creating and running your first GUI/WebAssembly/Mobile application, 36
Ring Notepad - Creating and running your first Web application, 38
Ring Notepad - Main Window, 33
The Browser Menu, 44
The Distribute Menu, 45
The Edit Menu, 42
The File Menu, 41
The Help Menu, 46
The Main File in the Project, 40
The Program Menu, 44
The Tools Menu, 45
The View Menu, 42
Ring Notepad - Creating and running your first console application
Ring Notepad, 34
Ring Notepad - Creating and running your first game
Ring Notepad, 39
Ring Notepad - Creating and running your first GUI/WebAssembly/Mobile application
Ring Notepad, 36
Ring Notepad - Creating and running your first Web application
Ring Notepad, 38
Ring Notepad - Main Window
Ring Notepad, 33
Ring Package Manager (*RingPM*)
Create Package in the Current Folder, 947
Creating New Package, 944
Deleting All Packages, 944
Discovering Packages, 941
Features, 941
Installing Packages, 942
Introduction, 940
Printing List of Installed Packages, 943
Remove Package, 944
Run Package, 943
The Package Description File, 946
The RingPM Registry File, 950
Update Package, 944
Updating the RingPM Registry, 942
Ring source code files and permissions
Deploying Web Applications in the Cloud, 654
Ring State
Embedding Ring Language in C/C++ Programs, 1585
Ring State Functions
Embedding Ring Language in C/C++ Programs, 1585
Ring State Variables
Embedding Ring Language in C/C++ Programs, 1586
Ring support in SpaceVim
What is new in Ring 1.13?, 1819
Ring Team
Resources, 1994
Ring VM - Tracing Functions
What is new in Ring 1.5?, 1697
RING_API_STATE
Extension, 1578
ring_ext.c
Extension, 1570
ring_ext.h
Extension, 1570
ring_state_filetokens() function
Embedding Ring in Ring, 343
What is new in Ring 1.12?, 1803
ring_state_new() and ring_state_mainfile()
Embedding Ring in Ring, 341
ring_state_new() and ring_state_mainfile() functions
What is new in Ring 1.7?, 1731
ring_state_setvar()
Embedding Ring in Ring, 340
RingAllegro Functions Reference
Introduction, 1070
RingCodeHighlighter Class
RingQt Classes Reference, 1498
RingConsoleColors Extension
What is new in Ring 1.6?, 1721
RingConsoleColors Functions Reference
Introduction, 1068
RingFreeGLUT Extension
What is new in Ring 1.5?, 1682
RingFreeGLUT Functions Reference
Introduction, 1140
RingLibCurl
Facebook Login, 999
Get Request, 999
Get Stock Data From Yahoo, 1001
Introduction, 998
Post Request, 999
Save output to string, 1000
What is new in Ring 1.2?, 1640
RingLibCurl Functions Reference
Introduction, 1002
RingLibSDL
Close Window Event, 1095

Create Window, 1091
Display Image, 1091
Display PNG Images, 1093
Display Transparent Images, 1094
Draw Rectangle, 1092
Introduction, 1090
Mouse Events, 1095
Play Sound, 1097
Switch Between Two Images, 1092
Use TTF Fonts, 1093
What is new in Ring 1.1?, 1631
RingLibSDL Functions Reference
Introduction, 1097
RingLibui Extension
What is new in Ring 1.14?, 1837
RingLibuv
Client Example, 1126
Client Example - Using Classes, 1128
First Application using RingLibuv, 1123
Introduction, 1122
Server Example, 1124
Server Example using Classes, 1127
The Events Loop, 1124
Threads Example, 1130
Threads Example - Using Classes, 1130
RingLibuv Extension
What is new in Ring 1.7?, 1734
RingLibuv Functions Reference
Introduction, 1131
RingLibZip Functions Reference
Introduction, 1065
RingMurmurHash Extension
What is new in Ring 1.6?, 1721
RingMurmurHash Functions Reference
Example, 1067
Introduction, 1066
MurmurHash1 functions, 1067
MurmurHash2 functions, 1067
MurmurHash3 functions, 1067
RingOpenGL (OpenGL 3.2) Functions Reference
Introduction, 1155
RingOpenGL Extension
What is new in Ring 1.5?, 1692
RingPostgreSQL Constants
PostgreSQL, 192
RingPostgreSQL Extension
What is new in Ring 1.9?, 1760
RingPostgreSQL Functions
PostgreSQL, 193
RingQt Classes and their Qt Documentation
Desktop, WebAssembly and Mobile Development, 504
RingQt Classes Reference
AbstractAxis Class, 1203
AbstractBarSeries Class, 1206
CodeEditor Class, 1207
Introduction, 1202
QAbstractAspect Class, 1207
QAbstractButton Class, 1208
QAbstractCameraController Class, 1209
QAbstractItemView Class, 1209
QAbstractPrintDialog Class, 1211
QAbstractScrollArea Class, 1211
QAbstractSeries Class, 1212
QAbstractSlider Class, 1213
QAbstractSocket Class, 1213
QAbstractSpinBox Class, 1215
QAction Class, 1216
QAllEvents Class, 1217
QApp Class, 1222
QAreaLegendMarker Class, 1222
QAreaSeries Class, 1222
QAspectEngine Class, 1224
QAxBase Class, 1224
QAxObject Class, 1225
QAxWidget Class, 1226
QAxWidget2 Class, 1226
QBarCategoryAxis Class, 1226
QBarLegendMarker Class, 1227
QBarSeries Class, 1227
QBarSet Class, 1227
QBitmap Class, 1229
QBluetoothAddress Class, 1229
QBluetoothDeviceDiscoveryAgent Class, 1229
QBluetoothDeviceInfo Class, 1230
QBluetoothHostInfo Class, 1230
QBluetoothLocalDevice Class, 1231
QBluetoothServer Class, 1232
QBluetoothServiceDiscoveryAgent Class, 1232
QBluetoothServiceInfo Class, 1233
QBluetoothSocket Class, 1234
QBluetoothTransferManager Class, 1235
QBluetoothTransferReply Class, 1235
QBluetoothTransferRequest Class, 1235
QBluetoothUuid Class, 1236
QBoxLayout Class, 1236
QBoxPlotLegendMarker Class, 1237
QBoxPlotSeries Class, 1237
QBoxSet Class, 1238
QBrush Class, 1239
QBuffer Class, 1240
QButtonGroup Class, 1240
QByteArray Class, 1241
QCalendarWidget Class, 1243
QCamera Class, 1244
QCameraImageCapture Class, 1245

QCameraLens Class, 1246
QCameraSelector Class, 1246
QCameraViewfinder Class, 1247
QCandlestickLegendMarker Class, 1247
QCandlestickModelMapper Class, 1247
QCandlestickSeries Class, 1247
QCandlestickSet Class, 1249
QCategoryAxis Class, 1251
QChar Class, 1251
QChart Class, 1253
QChartView Class, 1255
QCheckBox Class, 1256
QChildEvent Class, 1256
QClipboard Class, 1256
QColor Class, 1257
QColorDialog Class, 1259
QComboBox Class, 1260
QCompleter Class, 1262
QCompleter2 Class, 1263
QCompleter3 Class, 1263
QCompleter4 Class, 1263
QConeGeometry Class, 1263
QConeMesh Class, 1264
QCoreApplication Class, 1265
QCuboidMesh Class, 1266
QCullFace Class, 1266
QCursor Class, 1267
QCylinderMesh Class, 1267
QDate Class, 1267
QDateEdit Class, 1268
QDateTime Class, 1268
QDateTimeAxis Class, 1269
QDateTimeEdit Class, 1270
QDepthTest Class, 1272
QDesktopServices Class, 1272
QDesktopWidget Class, 1272
QDial Class, 1272
QDialog Class, 1273
QDiffuseSpecularMaterial Class, 1273
QDir Class, 1274
QDockWidget Class, 1276
QDrag Class, 1277
QDragEnterEvent Class, 1277
QDragLeaveEvent Class, 1278
QDragMoveEvent Class, 1278
QDropEvent Class, 1278
QEffect Class, 1279
QEntity Class, 1279
QEvent Class, 1279
QExtrudedTextMesh Class, 1279
QFile Class, 1280
QFile2 Class, 1281
QFileDevice Class, 1281
QFileDialog Class, 1281
QFileInfo Class, 1283
QFileSystemModel Class, 1285
QFirstPersonCameraController Class, 1286
QFont Class, 1286
QFontDialog Class, 1288
QFontMetrics Class, 1289
QForwardRenderer Class, 1290
QFrame Class, 1290
QFrame2 Class, 1291
QFrame3 Class, 1291
QFrameAction Class, 1291
QGeoAddress Class, 1291
QGeoAreaMonitorInfo Class, 1292
QGeoAreaMonitorSource Class, 1292
QGeoCircle Class, 1293
QGeoCoordinate Class, 1293
QGeoPositionInfo Class, 1293
QGeoPositionInfoSource Class, 1294
QGeoRectangle Class, 1294
QGeoSatelliteInfo Class, 1295
QGeoSatelliteInfoSource Class, 1296
QGeoShape Class, 1296
QGoochMaterial Class, 1296
QGradient Class, 1296
QGraphicsScene Class, 1297
QGraphicsVideoItem Class, 1299
QGraphicsView Class, 1300
QGridLayout Class, 1302
QGuiApplication Class, 1303
QHBarModelMapper Class, 1305
QHBoxLayout Class, 1306
QHBoxPlotModelMapper Class, 1306
QHCandlestickModelMapper Class, 1307
QHeaderView Class, 1309
QHorizontalBarSeries Class, 1312
QHorizontalPercentBarSeries Class, 1312
QHorizontalStackedBarSeries Class, 1312
QHostAddress Class, 1312
QHostInfo Class, 1313
QHPieModelMapper Class, 1308
QHXYModelMapper Class, 1309
QIcon Class, 1314
QImage Class, 1315
QInputAspect Class, 1316
QInputDialog Class, 1316
QIODevice Class, 1313
QJsonArray Class, 1318
QJsonDocument Class, 1319
QJsonObject Class, 1319
QJsonParseError Class, 1320
QJsonValue Class, 1320
QKeySequence Class, 1320
QLabel Class, 1321
QLayout Class, 1322

QLCDNumber Class, 1320
QLegend Class, 1323
QLegendMarker Class, 1324
QLinearGradient Class, 1327
QLineEdit Class, 1325
QLineSeries Class, 1327
QListView Class, 1327
QListWidget Class, 1328
QListWidgetItem Class, 1330
QLocale Class, 1331
QLogicAspect Class, 1334
QLogValueAxis Class, 1333
QMainWindow Class, 1334
QMaterial Class, 1335
QMatrix4x4 Class, 1335
QMdiArea Class, 1337
QMdiSubWindow Class, 1338
QMediaObject Class, 1338
QMediaPlayer Class, 1339
QMediaPlaylist Class, 1339
QMenu Class, 1340
QMenuBar Class, 1341
QMesh Class, 1341
QMessageBox Class, 1342
QMetalRoughMaterial Class, 1343
QMimeData Class, 1343
QMorphPhongMaterial Class, 1344
QMovie Class, 1344
QMutex Class, 1346
QMutexLocker Class, 1346
QNetworkAccessManager Class, 1346
QNetworkProxy Class, 1347
QNetworkReply Class, 1348
QNetworkRequest Class, 1348
QNmeaPositionInfoSource Class, 1349
QNode Class, 1349
QObject Class, 1350
QObjectPicker Class, 1350
QOpenGLBuffer Class, 1351
QOpenGLContext Class, 1352
QOpenGLDebugLogger Class, 1353
QOpenGLFramebufferObject Class, 1354
QOpenGLFunctions Class, 1354
QOpenGLFunctions_3_2_Core Class, 1358
QOpenGLPaintDevice Class, 1366
QOpenGLShader Class, 1367
QOpenGLShaderProgram Class, 1367
QOpenGLTexture Class, 1371
QOpenGLTimerQuery Class, 1373
QOpenGLVersionProfile Class, 1373
QOpenGLVertexArrayObject Class, 1374
QOpenGLWidget Class, 1374
QOrbitCameraController Class, 1375
QPageSetupDialog Class, 1375
QPaintDevice Class, 1375
QPainter Class, 1376
QPainter2 Class, 1378
QPainterPath Class, 1378
QPen Class, 1380
QPercentBarSeries Class, 1381
QPerVertexColorMaterial Class, 1380
QPhongMaterial Class, 1381
QPicture Class, 1381
QPieLegendMarker Class, 1382
QPieSeries Class, 1382
QPieSlice Class, 1383
QPixmap Class, 1385
QPixmap2 Class, 1386
QPlainTextEdit Class, 1387
QPlaneMesh Class, 1389
QPoint Class, 1389
QPointF Class, 1390
QPointLight Class, 1390
QPolarChart Class, 1391
QPrintDialog Class, 1391
QPrinter Class, 1392
QPrinterInfo Class, 1394
QPrintPreviewDialog Class, 1391
QPrintPreviewWidget Class, 1391
QProcess Class, 1394
QProgressBar Class, 1395
QPushButton Class, 1396
QQmlEngine Class, 1397
QQmlError Class, 1398
QQuaternion Class, 1398
QQuickView Class, 1399
QQuickWidget Class, 1399
QRadioButton Class, 1400
QRect Class, 1401
QRegion Class, 1402
QRegularExpression Class, 1403
QRegularExpressionMatch Class, 1403
QRegularExpressionMatchIterator Class, 1404
QRenderAspect Class, 1404
QRenderPass Class, 1405
QScatterSeries Class, 1405
QSceneLoader Class, 1406
QScreen Class, 1406
QScrollArea Class, 1407
QScrollBar Class, 1407
QSerialPort Class, 1407
QSerialPortInfo Class, 1409
QSize Class, 1409
QSkyboxEntity Class, 1409
QSlider Class, 1410
QSphereMesh Class, 1410
QSpinBox Class, 1411

QSplashScreen Class, 1411
QSplineSeries Class, 1412
QSplitter Class, 1412
QSqlDatabase Class, 1413
QSqlDriver Class, 1414
QSqlDriverCreatorBase Class, 1414
QSqlError Class, 1414
QSqlField Class, 1415
QSqlIndex Class, 1416
QSqlQuery Class, 1416
QSqlRecord Class, 1417
QStackedBarSeries Class, 1418
QStackedWidget Class, 1418
QStandardPaths Class, 1418
QStatusBar Class, 1419
QString2 Class, 1419
QStringList Class, 1421
QStringRef Class, 1422
QStyle Class, 1423
QSurfaceFormat Class, 1425
QSystemTrayIcon Class, 1426
Qt3DCamera Class, 1496
Qt3DWindow Class, 1498
QTabBar Class, 1426
QTableView Class, 1429
QTableWidget Class, 1431
QTableWidgetItem Class, 1433
QTabWidget Class, 1428
QTcpServer Class, 1434
QTcpSocket Class, 1435
QTechnique Class, 1435
QTest Class, 1436
QText2DEntity Class, 1436
QTextBlock Class, 1436
QTextBrowser Class, 1437
QTextCharFormat Class, 1438
QTextCodec Class, 1440
QTextCursor Class, 1440
QTextDocument Class, 1442
QTextEdit Class, 1444
QTextOption Class, 1447
QTextStream Class, 1447
QTextStream2 Class, 1448
QTextStream3 Class, 1449
QTextStream4 Class, 1449
QTextStream5 Class, 1449
QTextToSpeech Class, 1449
QTextureLoader Class, 1450
QTextureMaterial Class, 1450
QThread Class, 1451
QThreadPool Class, 1451
QTime Class, 1452
QTimer Class, 1453
QToolBar Class, 1453
QToolButton Class, 1454
QTorusMesh Class, 1454
QTransform Class, 1455
QTreeView Class, 1456
QTreeWidget Class, 1458
QTreeWidgetItem Class, 1460
QUrl Class, 1462
QUuid Class, 1463
QValueAxis Class, 1467
QVariant Class, 1469
QVariant2 Class, 1470
QVariant3 Class, 1470
QVariant4 Class, 1470
QVariant5 Class, 1470
QVariantDouble Class, 1471
QVariantFloat Class, 1471
QVariantInt Class, 1471
QVariantString Class, 1471
QVBarModelMapper Class, 1463
QVBoxLayout Class, 1464
QVBoxPlotModelMapper Class, 1464
QVCandlestickModelMapper Class, 1465
QVector2D Class, 1471
QVector3D Class, 1472
QVector4D Class, 1473
QVectorQVoice Class, 1473
QVideoWidget Class, 1474
QVideoWidgetControl Class, 1474
QViewport Class, 1475
QVoice Class, 1475
QVPieModelMapper Class, 1465
QVXYModelMapper Class, 1466
QWebEnginePage Class, 1475
QWebEngineView Class, 1478
QWebView Class, 1479, 1480
QWidget Class, 1480
QWindow Class, 1485
QXmlStreamAttribute Class, 1492
QXmlStreamAttributes Class, 1492
QXmlStreamEntityDeclaration Class, 1492
QXmlStreamEntityResolver Class, 1493
QXmlStreamNamespaceDeclaration Class, 1493
QXmlStreamNotationDeclaration Class, 1493
QXmlStreamReader Class, 1493
QXmlStreamWriter Class, 1495
QXYLegendMarker Class, 1489
QXYSeries Class, 1489
RingCodeHighlighter Class, 1498
RingRayLib Extension
What is new in Ring 1.11?, 1788
RingREPL
What is new in Ring 1.4?, 1660
RingSockets Extension

What is new in Ring 1.14?, 1841
RingSQLite
What is new in Ring 1.1?, 1633
RingStbImage Extension
What is new in Ring 1.13?, 1823
RingStbImage Functions Reference
Constants, 1153
Functions, 1154
Introduction, 1151, 1152
RingThreads Extension
What is new in Ring 1.14?, 1842
RingTilengine Extension
What is new in Ring 1.14?, 1835
RingTokens Class
Using TokensLib, 995
ringvm_callfunc()
Low Level Functions, 1533
RingVM_CallList()
Low Level Functions, 1529
RingVM_CFunctionsList()
Low Level Functions, 1524
RingVM_ClassesList()
Low Level Functions, 1525
RingVM_CodeList()
Low Level Functions, 1542
ringvm_evalinscope()
Low Level Functions, 1532
RingVM_FilesList()
Low Level Functions, 1530
RingVM_FunctionsList()
Low Level Functions, 1524
ringvm_give() function
Low Level Functions, 1542
ringvm_hideerrorMsg()
Low Level Functions, 1532
ringvm_info()
Low Level Functions, 1542
ringvm_ismempool()
Low Level Functions, 1542
RingVM_MemoryList()
Low Level Functions, 1527
RingVM_PackagesList()
Low Level Functions, 1526
ringvm_passerror()
Low Level Functions, 1532
ringvm_runcode()
Low Level Functions, 1542
ringvm_scopescount()
Low Level Functions, 1532
ringvm_see() and ringvm_give() functions
What is new in Ring 1.7?, 1730
ringvm_see() function
Low Level Functions, 1541
ringvm_settrace()
Low Level Functions, 1531
ringvm_tracedata()
Low Level Functions, 1531
ringvm_traceevent()
Low Level Functions, 1531
ringvm_tracefunc()
Low Level Functions, 1532
RingZip
Create Zip File, 1063
Extract Zip File, 1063
Introduction, 1062
Print files in Zip file, 1063
Using RingZip Classes, 1063
Zip Class Reference, 1065
ZipEntry Class Reference, 1065
RingZip Library
What is new in Ring 1.3?, 1653
Rotate Text
Desktop, WebAssembly and Mobile Development, 464
rsa_decrypt_oaep()
Security and Internet Functions, 209
rsa_decrypt_pkcs()
Security and Internet Functions, 207
rsa_decrypt_raw()
Security and Internet Functions, 212
rsa_encrypt_oaep()
Security and Internet Functions, 208
rsa_encrypt_pkcs()
Security and Internet Functions, 207
rsa_encrypt_raw()
Security and Internet Functions, 211
rsa_export_params()
Security and Internet Functions, 203
rsa_export_pem()
Security and Internet Functions, 205
rsa_generate()
Security and Internet Functions, 203
rsa_import_params()
Security and Internet Functions, 204
rsa_import_pem()
Security and Internet Functions, 206
rsa_is_privatekey()
Security and Internet Functions, 206
rsa_sign_pkcs()
Security and Internet Functions, 213
rsa_sign_pss()
Security and Internet Functions, 217
rsa_signhash_pkcs()
Security and Internet Functions, 214
rsa_signhash_pss()
Security and Internet Functions, 218
rsa_verify_pkcs()
Security and Internet Functions, 215

rsa_verify_pss()
    Security and Internet Functions, 218
rsa_verifyhash_pkcs()
    Security and Internet Functions, 216
rsa_verifyhash_pss()
    Security and Internet Functions, 219
Rules
    The Type Hints Library, 332
Run Package
    Ring Package Manager (*RingPM*), 943
Run the program
    Getting Started - First Style, 47
    Getting Started - Second Style, 50
    Getting Started - Third Style, 53
Running Forms
    Form Designer, 512
Runtime Errors
    Language Specification, 1985
Runtime Errors when Embedding Ring in Ring
    Embedding Ring in Ring, 342

S

Samples
    How to contribute?, 1975
    Using HTTPLib, 980
Samples in this book
    Applications developed in a few hours, 15
Samples Source (*Authors*)
    Using RingOpenGL and RingFreeGLUT for 3D Graphics, 787
Save and Restore Images
    ODBC Functions, 175
Save Image Inside the Database
    MySQL Functions, 183
Save output to string
    RingLibCurl, 1000
Say Hello
    libui, 1020
    Tutorial: Ring Extensions in C/C++, 1547
Say Something
    libui, 1028
Scaling and Rotating Images
    Graphics and Game Programming, 666
Scene
    Using Qt3D, 559
Scope Rules
    Accessing the class attributes from braces inside class methods, 302
    Conflict between Class Attributes and Local Variables, 298
    Conflict between Global Variables and Class Attributes, 297
    Conflict between self inside braces and self in the class region, 305
    Creating a Class for each Window in GUI applications, 304
    Defining Variables and Variables Access, 295
    How Ring find the Variable?, 296
    Introduction, 294
    Summary of Scope Rules, 310
    The For Loops uses the local scope, 309
    The Self Object, 296
    Three Scopes, 295
    Using Braces to access objects inside Class Methods, 300
    Using braces to escape from the current object scope, 308
    Using Object.Attribute, 296
Scope Rules for Functions and Methods
    Calling a function sharing the name with a method in the current class, 314
    Example about Sharing Names between Functions and Methods, 313
    How Ring find a functions and methods?, 312
    Introduction, 311
Screen Shots
    The Gold Magic 800 Game, 866
ScriptFunctions Class
    Web Development (*CGI Library*), 650
Search
    Lists, 98
Search of global names while defining the class attributes
    Frequently Asked Questions, 1941
Security and Internet Functions
    Decrypt(), 201
    Download(), 222
    Encrypt(), 201
    Example, 201
    File Hash, 202
    Introduction, 196
    Large Files Hash, 221
    MD5(), 198
    openssl_version(), 221
    openssl_versiontext(), 220
    RandBytes(), 203
    rsa_decrypt_oaep(), 209
    rsa_decrypt_pkcs(), 207
    rsa_decrypt_raw(), 212
    rsa_encrypt_oaep(), 208
    rsa_encrypt_pkcs(), 207
    rsa_encrypt_raw(), 211
    rsa_export_params(), 203
    rsa_export_pem(), 205
    rsa_generate(), 203

rsa_import_params(), 204
rsa_import_pem(), 206
rsa_is_privatekey(), 206
rsa_sign_pkcs(), 213
rsa_sign_pss(), 217
rsa_signhash_pkcs(), 214
rsa_signhash_pss(), 218
rsa_verify_pkcs(), 215
rsa_verify_pss(), 218
rsa_verifyhash_pkcs(), 216
rsa_verifyhash_pss(), 219
SendEmail(), 222
SHA1(), 198
SHA224(), 200
SHA256(), 199
SHA384(), 200
SHA512(), 199
SupportedCiphers(), 200
Security Class
Stdlib Classes, 392
Send Parameters
Functions - First Style, 82
Functions - Second Style, 86
Functions - Third Style, 90
SendEmail()
Security and Internet Functions, 222
Serial Execution of Programs
Embedding Ring in Ring, 340
Server Class Methods
Using HTTPLib, 979
Server Example
RingLibuv, 1124
Server Example using Classes
RingLibuv, 1127
Set List Item
Lists, 98
setattribute()
Reflection and Meta-programming, 265
setpointer()
Low Level Functions, 1522
Setter and Getter
Object Oriented Programming, 228
Setting the shebang line
Web Development (*CGI Library*), 595
SHA1()
Security and Internet Functions, 198
SHA224()
Security and Internet Functions, 200
SHA256()
Security and Internet Functions, 199
SHA384()
Security and Internet Functions, 200
SHA512()
Security and Internet Functions, 199
Shared Libraries
Extension, 1582
Short-circuit evaluation
Control Structures - First Style, 68
Shutdown() Function
System Functions, 157
sign()
Stdlib Functions, 351
Simple
Language Design, 1614
Simple Client and Server Example
Desktop, WebAssembly and Mobile Development, 468
Sin() Implementation
Extension, 1575
Single: Desktop, 393
Single: Using CSVLib
Introduction, 973
Single: Using HTTPLib
Introduction, 978
Single: Using JSONLib
Introduction, 976
Single: Using Qt3D
Using Qt3D, 520
Single: Using RingRayLib
Introduction, 729
Single: Using TokensLib
Introduction, 994
Single: Using ZeroLib
Introduction, 950
sleep()
Stdlib Functions, 359
Smart Garbage Collector
Language Design, 1623
Snake Game
What is new in Ring 1.11?, 1785
Snakes And Ladder Game
What is new in Ring 1.10?, 1775
socket
Constants, 1013
sockets
Functions, 1012
Introduction, 1011
TCP Client, 1012
TCP Server, 1012
Sokoban Game
What is new in Ring 1.11?, 1783
Sort()
Lists, 99
Sort() and List of Objects
Object Oriented Programming, 236
Sound Class
Game Engine for 2D Games, 677
Sound Loading

Using RingRayLib, 764
Source Code
Resources, 1994
The Gold Magic 800 Game, 868
Using ZeroLib, 953
Source Code File Sections
Program Structure, 93
space()
Low Level Functions, 1518
SpaceVim
Using Other Code Editors, 1924
Special thanks to contributors
How to contribute?, 1974
split()
Stdlib Functions, 348
splitmany()
Stdlib Functions, 349
Sprite Automatic Movement
Game Engine for 2D Games, 685
Sprite Class
Game Engine for 2D Games, 675
Sprite Keypress Event
Game Engine for 2D Games, 686
Sprite Mouse Event
Game Engine for 2D Games, 687
Sprite State Event
Game Engine for 2D Games, 688
SQLite
Introduction, 185
sqlite_close(), 186
sqlite_execute(), 186
sqlite_init(), 186
sqlite_open(), 186
SQLite Class
Stdlib Classes, 389
sqlite_close()
SQLite, 186
sqlite_execute()
SQLite, 186
sqlite_init()
SQLite, 186
sqlite_open()
SQLite, 186
Squares Puzzle Game
Applications developed in a few hours, 14
SRandom()
Mathematical Functions, 133
Stack Class
Stdlib Classes, 372
Stars Fighter Game
Game Engine for 2D Games, 698
startswith()
Stdlib Functions, 352
Static Methods
Code Generator, 1608
StdBase Class
Stdlib Classes, 367
StdLib - More Functions
What is new in Ring 1.11?, 1792
Stdlib Classes
Conversion Class, 385
DataType Class, 384
DateTime Class, 380
Debug Class, 383
File Class, 381
HashTable Class, 374
Internet Class, 392
Introduction, 366
List Class, 370
Math Class, 376
MySQL Class, 387
ODBC Class, 386
PostgreSQL Class, 390
Queue Class, 373
Security Class, 392
SQLite Class, 389
Stack Class, 372
StdBase Class, 367
String Class, 368
System Class, 382
Tree Class, 375
Stdlib Functions
apparguments(), 345
apppath(), 346
ASCIIList2Str(), 364
binarydigits(), 356
capitalized(), 349
changestring(), 359
CheckEquality(), 366
dayofweek(), 357
endswith(), 353
EpochTime(), 361
evenorodd(), 354
factorial(), 350
factors(), 355
fibonacci(), 351
file2list(), 352
filter(), 348
FSize(), 360
gcd(), 353
Introduction, 344
isappcompiled(), 345
IsBetween(), 364
isleapyear(), 356
IsListContainsItems(), 364
ismainsourcefile(), 359
ispalindrome(), 355
isprime(), 351

isspecial(), 349
isvowel(), 350
JustFileName(), 346
JustFilePath(), 346
lcm(), 353
linecount(), 350
List2Code() Function, 363
list2file(), 352
ListAllFiles() Function, 361
makedir(), 360
map(), 347
matrixmulti(), 356
matrixtrans(), 357
OSCopyFile() Function, 363
OSCopyFolder() Function, 362
OSCreateOpenFolder() Function, 362
OSDeleteFile() Function, 363
OSDeleteFolder() Function, 362
OSRenameFile() Function, 363
permutation(), 357
prodlist(), 354
puts(), 55
RandomItem(), 366
RandomList(), 365
readline(), 358
reduce(), 348
sign(), 351
sleep(), 359
split(), 348
splitmany(), 349
startswith(), 352
Str2ASCIIList(), 363
substring(), 358
sumlist(), 354
SystemCmd() Function, 361
SystemSilent() Function, 362
TimeInfo(), 365
times(), 347
TrimAll(), 360
TrimLeft(), 360
TrimRight(), 361
value(), 347
StdLib functions and classes written in Ring
What is new in Ring 1.1?, 1629
Step Option
Control Structures - First Style, 66
Steps to create the extension
Create your first extension, 1612
Stop the Server
Using HTTPLib, 993
StopWatch Application
What is new in Ring 1.8?, 1740
Str2ASCIIList()
Stdlib Functions, 363
Str2Hex()
Data Type, 129
str2list() and list2str()
Strings, 112
strcmp()
Strings, 112
String Class
Stdlib Classes, 368
String Literals
Strings, 108
String()
Data Type, 127
String2Constant Application
What is new in Ring 1.8?, 1739
Strings
Access String Letters, 108
Convert Letters Case, 108
Copy(), 110
Find SubString, 111
Get Number of Characters from position, 111
Get String Length, 108
Get Substring from position to end, 111
Introduction, 107
Left(), 109
Lines(), 110
Merge binary characters, 113
Right(), 109
str2list() and list2str(), 112
strcmp(), 112
String Literals, 108
Substr(), 111
Transform Substring To Another Substring, 112
Trim(), 110
StyleFunctions Class
Web Development (*CGI Library*), 650
Substr()
Strings, 111
substring()
Stdlib Functions, 358
Sudoku Game
What is new in Ring 1.11?, 1786
Sum List of Numbers
Tutorial: Ring Extensions in C/C++, 1548
Sum Two Numbers
Tutorial: Ring Extensions in C/C++, 1546
sumlist()
Stdlib Functions, 354
Summary of Scope Rules
Scope Rules, 310
Super Man 2016 Game
Game Engine for 2D Games, 714

SupportedCiphers()
    Security and Internet Functions, 200
Swap Items
    Lists, 107
Switch Between Two Images
    RingLibSDL, 1092
Switch Statement
    Control Structures - First Style, 63
    Control Structures - Second Style, 70
    Control Structures - Third Style, 74
Syntax Files
    What is new in Ring 1.18, 1890
Syntax Flexibility
    Automatic loading for syntax files, 327
    Change Language Keywords, 316
    Change Language Operators, 317
    Ignore new lines after keywords, 327
    Introduction, 315
    Load Syntax Files, 318
    Using "()" around the function parameters, 318
    Using $ and @ in the start of the variable name, 320
    Using 'break'/'continue' keywords, 325
    Using 'case' as 'on' in switch statements, 324
    Using 'def' as 'func' in functions/methods definition, 324
    Using 'end' keyword after Packages/Classes/Functions, 326
    Using 'endif'/'endfor'/'endwhile'/'endswitch'/'endtry' keywords, 326
    Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions, 326
    Using 'function'/'endfunction' keywords, 326
    Using 'put' and 'get' as 'see' and 'give', 324
    Using braces { } in Packages/Classes/Functions, 325
    Using braces to start and end different control structures, 323
    Using Semi-colon after and between statements, 320
    Using the 'else' keyword as 'other' in switch statement, 321
    Using the 'elseif' keyword as 'but' in if statement, 321
    Using the 'end' keyword in different control structures, 322
    What is new in Ring 1.1?, 1626
SysGet() Function
    System Functions, 151
SysSet() Function
    System Functions, 151
System Class
    Stdlib Classes, 382
System Functions
    ChDir() Function, 157
    CurrentDir() Function, 156
    Example, 153
    ExeFileName() Function, 156
    ExeFolder() Function, 157
    Get Active Source File Name, 155
    Get Command Line Arguments, 154
    GetArch() Function, 153
    Introduction, 149
    IsAndroid() Function, 153
    IsFreeBSD() Function, 152
    IsLinux() Function, 152
    IsMacOSX() Function, 152
    IsMSDOS() Function, 151
    IsUnix() Function, 152
    IsWindows() Function, 151
    IsWindows64() Function, 152
    NofProcessors() Function, 157
    PrevFileName() Function, 156
    Shutdown() Function, 157
    SysGet() Function, 151
    SysSet() Function, 151
    System() Function, 150
    SysUnset() Function, 151
    Version() Function, 157
    Windowsvl() Function, 154
System() Function
    System Functions, 150
SystemCmd() Function
    Stdlib Functions, 361
SystemSilent() Function
    Stdlib Functions, 362
SysUnset() Function
    System Functions, 151

T

TableWidget - AddList() Method
    Desktop, WebAssembly and Mobile Development, 505
TCP Client
    sockets, 1012
TCP Server
    sockets, 1012
Tempfile()
    Files, 141
Templates
    Web Development (*CGI Library*), 610
Tempname()
    Files, 141

Testing
How to contribute?, 1974
Testing the extension
Create your first extension, 1613
Tetris Game
What is new in Ring 1.13?, 1816
Text 2D
Using Qt3D, 543
Text Class
Game Engine for 2D Games, 676
Text Drawing
libui, 1042
Text To Speech Application
What is new in Ring 1.11?, 1788
Texture
Using Qt3D, 533
Texture Source
Using RingRayLib, 769
The Browser Menu
Ring Notepad, 44
The Camera
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 798
The Cards Game
Demo Programs, 493
Desktop, WebAssembly and Mobile Development, 493
The Designer Windows
Form Designer, 511
The Difference between Qt and RingQt
Desktop, WebAssembly and Mobile Development, 503
The Distribute Menu
Ring Notepad, 45
The documentation says functional programming is supported, but then this happens?
Frequently Asked Questions, 1933
The Edit Menu
Ring Notepad, 42
The Events Loop
Desktop, WebAssembly and Mobile Development, 396
RingLibuv, 1124
The File Menu
Ring Notepad, 41
The files ring.ring and ring.ringo
Distributing Ring Application, 929
The First GUI Application
Desktop, WebAssembly and Mobile Development, 394
The First Triangle
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 791
The First Window using RingFreeGLUT
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 788
The For Loops uses the local scope
Scope Rules, 309
What is new in Ring 1.8?, 1745
The Game Story
The Gold Magic 800 Game, 866
The Gold Magic 800 Game
How to play?, 866
Introduction, 865
Screen Shots, 866
Source Code, 868
The Game Story, 866
What will you learn?, 866
The Help Menu
Ring Notepad, 46
The Main File in the Project
Ring Notepad, 40
The Natural Library
What is new in Ring 1.4?, 1657
The Package Description File
Ring Package Manager (*RingPM*), 946
The Power Operator
What is new in Ring 1.18, 1891
The Program Menu
Ring Notepad, 44
The Properties
Form Designer, 511
The Qt project for your Ring application
Building RingQt Applications for Mobile, 579
Building RingQt Applications for WebAssembly, 588
The Ring Package Manager (*RingPM*)
What is new in Ring 1.10?, 1778
The RingPM Registry File
Ring Package Manager (*RingPM*), 950
The Self Object
Scope Rules, 296
The Tools Menu
Ring Notepad, 45
The Trace Library and the Interactive Debugger
BreakPoint, 336
Disable BreakPoints, 336
Execute Program Line by Line, 336
Interactive Debugger, 335
Introduction, 333
Loading the Trace library, 334
Pass Error, 335
Trace All Events, 334
Trace control flow between functions, 334
Using the Interactive Debugger, 337

The Tree Class
    Using References, 1509
The Type Hints Library
    Example, 330
    Introduction to the Type Hints Library, 329
    Rules, 332
    User Types, 330
    Using Override, 331
    Using Types inside Code, 331
    Why Type Hints?, 330
The View Menu
    Ring Notepad, 42
Threads
    Creating Threads, 1014
    Graphics and Game Programming, 668
    Introduction, 1013
    More Examples, 1014
    Reference, 1016
Threads Example
    RingLibuv, 1130
Threads Example - Using Classes
    RingLibuv, 1130
Threads Support
    Code Generator, 1595
Three Scopes
    Scope Rules, 295
TicTacToe 3D Game
    Using RingOpenGL and RingAllegro for 3D Graphics, 852
tilengine
    Getting Started, 870
    Introduction, 869
    More Samples, 871
    Reference, 896
    Using LibSDL and Tilengine, 892
Time()
    Date and Time, 116
TimeInfo()
    Stdlib Functions, 365
TimeList()
    Date and Time, 117
times()
    Stdlib Functions, 347
TokensLib Library
    What is new in Ring 1.17, 1872
Trace All Events
    The Trace Library and the Interactive Debugger, 334
Trace control flow between functions
    The Trace Library and the Interactive Debugger, 334
Trace Library and Interactive Debugger
    What is new in Ring 1.5?, 1702
Transaction Example
    MySQL Functions, 184
Transactions and Using Commit and Rollback
    ODBC Functions, 174
Transform Substring To Another Substring
    Strings, 112
Transparent Image
    Graphics and Game Programming, 667
Transparent Implementation
    Language Design, 1621
Tree Class
    Stdlib Classes, 375
Triangle Rotation
    Using RingOpenGL and RingFreeGLUT for 3D Graphics, 793
Trim()
    Strings, 110
TrimAll()
    Stdlib Functions, 360
TrimLeft()
    Stdlib Functions, 360
TrimRight()
    Stdlib Functions, 361
TrueType Fonts
    Graphics and Game Programming, 664
Try/Carch/Done
    Eval() and Debugging, 159
Trying to be natural
    Language Design, 1615
Tutorial: Number to Words
    What is new in Ring 1.12?, 1801
Tutorial: Ring Extensions in C/C++
    Build the extension on different platforms, 1545
    Create Table, 1565
    Display List, 1558
    Filter List Items, 1551
    Generate List, 1556
    Hello World, 1544
    Increment List Items, 1550
    Replicate List Items, 1553
    Say Hello, 1547
    Sum List of Numbers, 1548
    Sum Two Numbers, 1546
    Tutorial: Ring Extensions in C/C++, 1543
    Update Table, 1561
Two Cubes
    Using RingRayLib, 745
Type Hints Library
    What is new in Ring 1.5?, 1705
Type()
    Data Type, 122

U

Ungetc()
    Files, 144
Unsigned()
    Mathematical Functions, 133
Update Package
    Ring Package Manager (*RingPM*), 944
Update Table
    Tutorial: Ring Extensions in C/C++, 1561
UpdateColumn()
    Using FastPro, 1502
UpdateList()
    Using FastPro, 1499
Updating the RingPM Registry
    Ring Package Manager (*RingPM*), 942
Upload Files
    Web Development (*CGI Library*), 604
Uploading Files
    Using HTTPLib, 990
URL Encode
    Web Development (*CGI Library*), 609
Usage
    Deploying Web Applications in the Cloud, 653
Use TTF Fonts
    RingLibSDL, 1093
User Types
    The Type Hints Library, 330
Users registration and Login
    Web Development (*CGI Library*), 633
Using "()" around the function parameters
    Syntax Flexibility, 318
Using ? to print expression then new line
    Getting Started - First Style, 48
    What is new in Ring 1.6?, 1728
Using $ and @ in the start of the variable name
    Syntax Flexibility, 320
Using && and || operators
    What is new in Ring 1.6?, 1727
Using _ in numbers
    Mathematical Functions, 135
Using ':' then Numbers
    What is new in Ring 1.15?, 1858
Using '<' and ':' operators as 'from' keyword
    What is new in Ring 1.3?, 1652
Using 'break'/'continue' keywords
    Syntax Flexibility, 325
Using 'case' as 'on' in switch statements
    Syntax Flexibility, 324
Using 'def' as 'func' in functions/methods definition
    Syntax Flexibility, 324
Using 'end' keyword after Packages/Classes/Functions
    Syntax Flexibility, 326
Using 'endif'/'endfor'/'endwhile'/'endswitch'/'endtry' keywords
    Syntax Flexibility, 326
Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions
    Syntax Flexibility, 326
Using 'function'/'endfunction' keywords
    Syntax Flexibility, 326
Using 'put' and 'get' as 'see' and 'give'
    Syntax Flexibility, 324
Using Atom
    Using Other Code Editors, 1920
Using Bootstrap Library using Functions
    Web Development (*CGI Library*), 626
Using Bootstrap Library using Objects
    Web Development (*CGI Library*), 628
Using braces { } in Packages/Classes/Functions
    Syntax Flexibility, 325
Using Braces to access objects inside Class Methods
    Scope Rules, 300
Using braces to escape from the current object scope
    Scope Rules, 308
Using braces to start and end different control structures
    Syntax Flexibility, 323
Using C/C++ Compiler and Linker
    Distributing Ring Application, 929
Using CLOC (*Count Lines of Code*)
    What is new in Ring 1.14?, 1847
Using configuration file that wrap C++ library
    Code Generator, 1597
Using Cookies
    Using HTTPLib, 989
Using CR as Carriage return
    What is new in Ring 1.6?, 1726
Using CSVLib
    Examples, 974
    Functions, 974
    Introduction, 974
Using Emacs Editor
    Using Other Code Editors, 1922
Using Eval() with our Natural Code
    Natural Language Programming, 280
Using f after numbers
    Mathematical Functions, 135
Using FastPro

Bytes2List() function, 1499
Introduction, 1498
List2Bytes(), 1499
UpdateColumn(), 1502
UpdateList(), 1499
Using Fonts
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 818
Using Geany
Using Other Code Editors, 1919
Using HTTP Get
Using HTTPLib, 980
Using HTTP Post
Using HTTPLib, 983
Using HTTPLib
Example, 979
Introduction, 979
More Samples, 986
Printing Constants, 980
Regular Expressions, 993
Samples, 980
Server Class Methods, 979
Stop the Server, 993
Uploading Files, 990
Using Cookies, 989
Using HTTP Get, 980
Using HTTP Post, 983
Using Templates, 992
Using WebLib, 982
Using JSONLib
Examples, 977
Functions, 977
Introduction, 977
Using Layout
Desktop, WebAssembly and Mobile Development, 396
Using Layouts
Form Designer, 517
Using len() and For Loops
Performance Tips, 908
Using LibSDL and Tilengine
tilengine, 892
Using Lists during definition
Lists, 102
Using Many Source Code Files
Program Structure, 93
Using Notepad++
Using Other Code Editors, 1918
Using NULL instead of NULLPointer()
What is new in Ring 1.2?, 1640
Using Object.Attribute
Scope Rules, 296
Using ObjectID() and GetWindowByID()
Objects Library for RingQt Application, 568
Using ObjectID() and OpenWindowAndLink()
Objects Library for RingQt Application, 571
Using Other Code Editors
Introduction, 1917
SpaceVim, 1924
Using Atom, 1920
Using Emacs Editor, 1922
Using Geany, 1919
Using Notepad++, 1918
Using Sublime Text 2, 1921
Using Visual Studio IDE, 1922
Visual Studio Code, 1923
Using Override
The Type Hints Library, 331
Using QCheckBox
Desktop, WebAssembly and Mobile Development, 424
Using QComboBox Class
Desktop, WebAssembly and Mobile Development, 405
Using QDateEdit
Desktop, WebAssembly and Mobile Development, 419
Using QDesktopWidget Class
Desktop, WebAssembly and Mobile Development, 463
Using QDial
Desktop, WebAssembly and Mobile Development, 420
Using QDockWidget
Desktop, WebAssembly and Mobile Development, 412
Using QFrame
Desktop, WebAssembly and Mobile Development, 430
Using QInputDialog Class
Desktop, WebAssembly and Mobile Development, 455
Using qLCDNumber
Desktop, WebAssembly and Mobile Development, 451
Using QPrintPreviewDialog
Desktop, WebAssembly and Mobile Development, 446
Using QProgressBar
Desktop, WebAssembly and Mobile Development, 416
Using QProgressBar and Timer
Desktop, WebAssembly and Mobile Development, 440
Using QRadioButton and QButtonGroup

Desktop, WebAssembly and Mobile Development, 425
Using QSlider
Desktop, WebAssembly and Mobile Development, 417
Using QSpinBox
Desktop, WebAssembly and Mobile Development, 417
Using Qt3D
Camera, 556
Draw Office, 551
Drawing Cone, 529
Drawing Cube, 521
Drawing Cylinder, 527
Drawing Plane, 531
Drawing Sphere, 525
Drawing Torus, 523
Extruded Text, 545
Frame Action, 540
Key Press, 535
Many Objects, 553
Model, 547
Model Texture, 549
Object Picker, 537
Scene, 559
Text 2D, 543
Texture, 533
Using QTableWidget
Desktop, WebAssembly and Mobile Development, 415
Using QTabWidget
Desktop, WebAssembly and Mobile Development, 413
Using QTreeView and QFileSystemModel
Desktop, WebAssembly and Mobile Development, 402
Using QTreeWidget and QTreeWidgetItem
Desktop, WebAssembly and Mobile Development, 403
Using QWebView
Desktop, WebAssembly and Mobile Development, 423
Using References
Circular References, 1508
Dependency injection, 1511
Introduction, 1505, 1506
Linked list, 1509
Nested Ref(), 1513
Passing lists to functions, 1512
ref(), 1506
Ref() and sub lists, 1514
Ref() and temp. lists, 1513
Ref() in left side, 1514
refcount(), 1507
The Tree Class, 1509
What is new in Ring 1.18, 1891
Using Ring2EXE
Building RingQt Applications for Mobile, 579
Building RingQt Applications for WebAssembly, 587
Distributing Ring Application using Ring2EXE, 931
Using RING_API_RETMANAGEDCPOINTER() Extension, 1583
Using RingOpenGL and RingAllegro for 3D Graphics
3D Cube and Texture, 841
Introduction, 840
Many Cubes, 846
More 3D Samples, 863
TicTacToe 3D Game, 852
Using RingOpenGL and RingFreeGLUT for 3D Graphics
Drawing using RingOpenGL, 789
Frames Per Second, 826
Introduction, 786
Keyboard Events and Colors, 795
Make a Cube using RingOpenGL and RingFreeGLUT, 836
Menu Events, 810
Mouse Events, 805
Samples Source (*Authors*), 787
The Camera, 798
The First Triangle, 791
The First Window using RingFreeGLUT, 788
Triangle Rotation, 793
Using Fonts, 818
What is RingFreeGLUT?, 788
What is RingOpenGL?, 787
Window Resize Event, 792
Using RingRayLib
3D Camera, 733
3D Camera Free, 735
3D Picking, 742
Basic Shapes, 748
Basic Window, 730
Bezier Lines, 752
Camera First Person, 740
Collision Area, 753
Colors Palette, 758
Cubic Map, 773
Draw Ring, 749
Following Eyes, 756
Full Screen, 744
Functions, 775, 785
Geometric Shapes, 771
Image Drawing, 765

Image Generation, 767
Input Keys, 731
Input Mouse, 732
Input Multi-touch, 738
Introduction, 730
Mouse Wheel, 737
Music Playing, 762
Rectangle Scaling, 760
Sound Loading, 764
Texture Source, 769
Two Cubes, 745
Using RingZip Classes
RingZip, 1063
Using Self.Attribute
Object Oriented Programming, 238
Using Self.Attribute in the Class Region to define new attributes
What is new in Ring 1.1?, 1634
Using Semi-colon after and between statements
Syntax Flexibility, 320
Using String2Constant
Multi-language applications, 574
Using Sublime Text 2
Using Other Code Editors, 1921
Using Tab instead of char(9)
What is new in Ring 1.6?, 1726
Using Templates
Using HTTPLib, 992
Using the ! operator as not
What is new in Ring 1.6?, 1726
Using the 'else' keyword as 'other' in switch statement
Syntax Flexibility, 321
Using the 'elseif' keyword as 'but' in if statement
Syntax Flexibility, 321
Using the 'end' keyword in different control structures
Syntax Flexibility, 322
Using the Interactive Debugger
The Trace Library and the Interactive Debugger, 337
Using the Menubar
libui, 1030
Using the Natural Library
Defining Commands, 289
Defining commands using classes, 293
Introduction, 286
Natural Library - Demo Program, 287
Operators, 292
Using the QColorDialog Class
Desktop, WebAssembly and Mobile Development, 450
Using the QFileDialog Class
Desktop, WebAssembly and Mobile Development, 442
Using the QListWidget Class
Desktop, WebAssembly and Mobile Development, 399
Using the QTextEdit Class
Desktop, WebAssembly and Mobile Development, 398
Using the QTimer Class
Desktop, WebAssembly and Mobile Development, 439
Using the tool
Code Generator, 1588
Using This in the class region as Self
Object Oriented Programming, 239
What is new in Ring 1.8?, 1743
Using This.Attribute and This.Method()
Object Oriented Programming, 239
Using This.Attribute in nested Braces inside the Class Methods
What is new in Ring 1.1?, 1634
Using TokensLib
Constants, 996
Example, 995
Introduction, 995
RingTokens Class, 995
Using Types inside Code
The Type Hints Library, 331
Using Visual Studio IDE
Using Other Code Editors, 1922
Using WebLib
Using HTTPLib, 982
Using XAMPP Apache web server
Configure the Apache web server, 594
Using ZeroLib
Introduction, 951
Source Code, 953
Z() function, 951
ZeroBasedList Class, 951
ZeroBasedString Class, 952
UTF-8 File Names in Microsoft Windows
What is new in Ring 1.16?, 1863

V

value()
Stdlib Functions, 347
Variables
Deep Copy, 57
Dynamic Typing, 57
Implicit Conversion, 58
Introduction, 56
Variables Scope
Functions - First Style, 83

Functions - Second Style, 87
Functions - Third Style, 91
varptr()
Low Level Functions, 1517
Version() Function
System Functions, 157
Video-Music-Player Application
Applications developed in a few hours, 11
What is new in Ring 1.5?, 1669
Virtual Machine Instructions
Language Specification, 1988
Visual Implementation
Language Design, 1622
Visual Studio Code
Using Other Code Editors, 1923

W

Web Development (*CGI Library*)
Application Class, 647
Configure the Apache web server, 594
Cookies, 607
CRUD Example using MVC, 631
Database, ModelBase & ControllerBase classes, 640
Generating Pages using Objects, 620
Gradient, 619
Grant Ring files execution permission, 595
Hash Functions, 613
Hello World Program using the Web Library, 596
HTML Lists, 616
HTML Special Characters, 612
HTML Tables, 618
HtmlPage Class, 625, 652
HTTP Get Example, 597
HTTP POST Example, 602
Introduction, 593
Page Class, 647
Random Image, 615
Ring CGI Hello World Program, 596
ScriptFunctions Class, 650
Setting the shebang line, 595
StyleFunctions Class, 650
Templates, 610
Upload Files, 604
URL Encode, 609
Users registration and Login, 633
Using Bootstrap Library using Functions, 626
Using Bootstrap Library using Objects, 628
Web Library Features, 597
WebLib API, 646
WebPage Class, 651
Web Development Samples
What is new in Ring 1.14?, 1830
Web Library Features
Web Development (*CGI Library*), 597
WebAssembly and Mobile Development
Introduction, 393
WebLib API
Web Development (*CGI Library*), 646
WebPage Class
Web Development (*CGI Library*), 651
Weight History Application
Desktop, WebAssembly and Mobile Development, 472
What about Data Representation in Ring?
Frequently Asked Questions, 1928
What about Memory Management in Ring?
Frequently Asked Questions, 1927
What about predefined parameters or optional parameters in functions?
Frequently Asked Questions, 1944
What about the Boolean values in Ring?
Frequently Asked Questions, 1935
What are the advantages of using Ring over native C or C++?
Frequently Asked Questions, 1931
What are the advantages to using Ring over C# or Java?
Frequently Asked Questions, 1933
What are the advantages to using Ring over Lisp or Smalltalk?
Frequently Asked Questions, 1930
What are the advantages to using Ring over Python and Ruby?
Frequently Asked Questions, 1932
What are the advantages to using Ring over Tcl and Lua?
Frequently Asked Questions, 1932
What happens when we create a new object?
Frequently Asked Questions, 1940
What is new in Ring 1.17
Better GUILib, 1877
Better I/O Functions, 1880
Better Ring API, 1881
Better RingOpenSSL, 1879
Better RingWinAPI, 1879
Better Syntax Flexibility, 1879
Citations Prediction, 1871
Code Generator for Extensions, 1882
CSVLib Library, 1873
HTTPLib Library, 1876
JSONLib Library, 1875
Merge Two Files, 1869
More Improvements, 1882

Poetry Analysis, 1870
TokensLib Library, 1872
What is new in Ring 1.18
Better CitPre application, 1887
Better Documentation, 1899
Better Ring API, 1898
Better RingRayLib, 1894
Better RingSockets, 1895
Better Tools, 1888
Better WebLib, 1897
Faster (For-In) Loop, 1892
Faster BraceError(), 1893
Heroku (*Better support*), 1896
ImportPackage() Function, 1896
MatrixLib Library, 1899
More Improvements, 1900
More Low Level Functions, 1897
More Samples, 1889
New From, 1896
Random User application, 1886
Read Files application, 1885
Ring for MS-DOS, 1892
Syntax Files, 1890
The Power Operator, 1891
Using References, 1891
What is new in Ring 1.19
Better batch files for building Ring, 1906
Better Documentation, 1914
Better Extensions Generator, 1914
Better RingRayLib, 1911
Better RingStbImage, 1912
Better TypeHints, 1910
FastPro Extension, 1909
For Loop - Better Performance, 1907
Hours Counter application, 1904
Image Pixel application, 1903
More Improvements, 1915
More Samples, 1905
Nothing() function, 1908
OptionalFunc() function, 1908
Planetary Orbits application, 1904
Ring for Windows 64bit, 1906
What is new in Ring 1.10?
Better Tests, 1778
Chess Game, 1770
Game of Life Game, 1773
Introduction, 1769
Knight Tour Game, 1772
List of changes and new features, 1770
Minesweeper Game, 1771
More Games, 1776
More Improvements, 1779
Pong Game, 1774
Ring Extension for Visual Studio Code, 1777
Snakes And Ladder Game, 1775
The Ring Package Manager (*RingPM*), 1778
What is new in Ring 1.11?
Better Documentation, 1795
Better Performance, 1795
Better RingQt, 1793
Checkers Game, 1782
Desktop Screen Shot Application, 1787
Introduction, 1780
List of changes and new features, 1781
Maze Game, 1784
More 3D Samples, 1781
More Improvements, 1795
RingRayLib Extension, 1788
Snake Game, 1785
Sokoban Game, 1783
StdLib - More Functions, 1792
Sudoku Game, 1786
Text To Speech Application, 1788
ZeroLib Library, 1791
What is new in Ring 1.12?
ASCII Table application, 1798
Better RingRayLib, 1805
BMI Calculator application, 1798
Calendar application, 1799
Generate Embedded Object File, 1804
Go Game, 1797
Introduction, 1796
Julian Day Calendar application, 1800
List of changes and new features, 1797
Load Again Command, 1802
More Improvements, 1806
ring_state_filetokens() function, 1803
Tutorial: Number to Words, 1801
What is new in Ring 1.13?
Better Organization, 1824
Better RingQt, 1819
Better RingRayLib, 1822
Better Threads Support, 1810
Escape Game, 1817
Hassouna Course Samples, 1818
Introduction, 1807
Laser Game, 1810
List of changes and new features, 1808
Magic Balls Game, 1811
Matching Game, 1813
MoneyBoxes Game, 1812
More Improvements, 1824
More Low Level Functions, 1823
New Book by Apress, 1808
Pairs Game, 1814
Ring for WebAssembly, 1809

Ring support in SpaceVim, 1819
RingStbImage Extension, 1823
Tetris Game, 1816
Word Game, 1815
What is new in Ring 1.14?
Better Functions, 1845
Better Handling For Numbers, 1846
Better Performance For Strings, 1845
Better RingOpenSSL, 1843
Customers Form, 1834
Erlang B Formula, 1833
Introduction, 1827
List of changes and new features, 1828
More Functions, 1843
More Improvements, 1848
More Samples, 1831
MyCoffee (*Web Application*), 1828
RingLibui Extension, 1837
RingSockets Extension, 1841
RingThreads Extension, 1842
RingTilengine Extension, 1835
Using CLOC (*Count Lines of Code*), 1847
Web Development Samples, 1830
What is new in Ring 1.15?
Better Ring Notepad, 1854
Better RingLibCurl, 1858
Better RingQt, 1857
Better StdLib, 1856
Chess 3D (*Qt3D Sample*), 1853
Chess Endgame Application, 1851
Heroku (*Better support*), 1858
Introduction, 1849
List of changes and new features, 1850
More Improvements, 1860
More Samples, 1859
New Arabic Book, 1850
Using ':' then Numbers, 1858
What is new in Ring 1.16?
Better Documentation, 1867
Better Installation Scripts, 1867
Code Runner Extension support Ring, 1864
Introduction, 1861
Light GUILib, 1862
List of changes and new features, 1862
MDI Windows Sample, 1867
More Improvements, 1867
Nested Methods Call in Separate Lines, 1863
UTF-8 File Names in Microsoft Windows, 1863
Zero and Strings, 1866
What is new in Ring 1.17
Introduction, 1868
What is new in Ring 1.17?
List of changes and new features, 1869
What is new in Ring 1.18
Introduction, 1884
What is new in Ring 1.18?
List of changes and new features, 1885
What is new in Ring 1.19
Introduction, 1902
What is new in Ring 1.19?
List of changes and new features, 1903
What is new in Ring 1.1?
Better Code Generator for Extensions, 1634
Better Documentation, 1634
Better Natural Language Programming Support, 1625
Game Engine for 2D Games, 1632
Generate/Execute Ring Object Files (*.*ringo*), 1626
Introduction, 1624
List of changes and new features, 1625
New Functions and Changes, 1628
RingLibSDL, 1631
RingSQLite, 1633
StdLib functions and classes written in Ring, 1629
Syntax Flexibility, 1626
Using Self.Attribute in the Class Region to define new attributes, 1634
Using This.Attribute in nested Braces inside the Class Methods, 1634
What is new in Ring 1.2?
Better Call Command, 1640
Better Functions, 1637
Better Quality, 1641
Better Ring Notepad, 1637
Better RingQt, 1638
Display Warnings Option, 1640
Introduction, 1635
List of changes and new features, 1636
New Functions, 1636
Objects Library for RingQt, 1638
RingLibCurl, 1640
Using NULL instead of NULLPointer(), 1640
What is new in Ring 1.3?
Better Loop|Exit Command, 1650
Better Ring Notepad, 1646
Better RingQt, 1642
Better StdLib, 1650
Embedding Ring in Ring without sharing the State, 1652
Form Designer, 1654
Introduction, 1641
List of changes and new features, 1642

New Functions, 1651
Return Self by Reference, 1651
Ring mode for Emacs Editor, 1649
RingZip Library, 1653
Using '<' and ':' operators as 'from' keyword, 1652
What is new in Ring 1.4.1?
What is new in Ring 1.4?, 1665
What is new in Ring 1.4?
Better RingQt, 1664
Better StdLib, 1662
Better WebLib, 1662
Change: Basic Extensions are separated from RingVM, 1656
Convert between Numbers and Bytes, 1661
Introduction, 1655
List of changes and new features, 1656
New Style to Ring Notepad, 1659
Qt Class Converter, 1664
RingREPL, 1660
The Natural Library, 1657
What is new in Ring 1.4.1?, 1665
What is new in Ring 1.5.1?
What is new in Ring 1.5?, 1707
What is new in Ring 1.5.2?
What is new in Ring 1.5?, 1711
What is new in Ring 1.5.3?
What is new in Ring 1.5?, 1712
What is new in Ring 1.5.4?
What is new in Ring 1.5?, 1716
What is new in Ring 1.5?
Better Code Generator for Extensions, 1696
Better Documentation Generator for Extensions, 1697
Better Objects Library, 1681
Better Quality, 1706
Better Ring Notepad, 1672
Better RingQt, 1679
Better StdLib, 1675
Better WebLib, 1676
Calculator Application, 1671
Introduction, 1668
List of changes and new features, 1669
More Syntax Flexibility, 1704
Ring VM - Tracing Functions, 1697
RingFreeGLUT Extension, 1682
RingOpenGL Extension, 1692
Trace Library and Interactive Debugger, 1702
Type Hints Library, 1705
Video-Music-Player Application, 1669
What is new in Ring 1.5.1?, 1707
What is new in Ring 1.5.2?, 1711
What is new in Ring 1.5.3?, 1712
What is new in Ring 1.5.4?, 1716
Windows StartUp Manager Application, 1670
What is new in Ring 1.6?
Better Ring For Android, 1719
Better Ring Notepad, 1722
Better RingQt, 1724
Better RingREPL, 1725
Better RingVM, 1725
Better Scripts for building Ring, 1721
Better StdLib, 1725
Employee Application, 1718
Introduction, 1717
List of changes and new features, 1718
New Tool: Folder2qrc, 1720
New Tool: Ring2EXE, 1719
RingConsoleColors Extension, 1721
RingMurmurHash Extension, 1721
Using ? to print expression then new line, 1728
Using && and || operators, 1727
Using CR as Carriage return, 1726
Using Tab instead of char(9), 1726
Using the ! operator as not, 1726
What is new in Ring 1.7?
Better Documentation, 1734
Better Ring Notepad, 1732
Better Ring VM, 1734
Better Ring2EXE, 1733
Better RingQt, 1732
Better RingZip, 1733
Better Trace Library, 1731
Introduction, 1728
List of changes and new features, 1729
New Command: Load Package, 1729
ring_state_new() and ring_state_mainfile() functions, 1731
RingLibuv Extension, 1734
ringvm_see() and ringvm_give() functions, 1730
What is new in Ring 1.8?
Better Cards Game, 1748
Better Code Generator For Extensions, 1749
Better Form Designer, 1748
Better Performance, 1738
Better Ring Compiler and VM, 1750
Better RingQt, 1749
Compiling on Manjaro Linux, 1743
Default value for object attributes is NULL, 1745
Find in files Application, 1739
FoxRing Library, 1747

Introduction, 1737
List of changes and new features, 1738
Merge binary characters, 1746
More 3D Samples, 1741
Notes to extensions creators, 1751
StopWatch Application, 1740
String2Constant Application, 1739
The For Loops uses the local scope, 1745
Using This in the class region as Self, 1743
What is new in Ring 1.9?
Better Code Generator for Extensions, 1767
Better Memory Management, 1766
Better Ring Notepad, 1757
Better RingQt, 1763
Better StdLib, 1758
BigNumber Library, 1759
Deploying Web applications in the Cloud, 1763
Introduction, 1751
List of changes and new features, 1752
More Games, 1754
More Improvements, 1768
New Game : Gold Magic 800, 1752
RingPostgreSQL Extension, 1760
What is RingFreeGLUT?
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 788
What is RingOpenGL?
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 787
What is the difference between Ring and Python? And is Ring Open Source?
Frequently Asked Questions, 1931
What is the Ring Architecture?
Frequently Asked Questions, 1926
What will you learn?
The Gold Magic 800 Game, 866
Where can I write a program and execute it?
Frequently Asked Questions, 1943
Which of 3 coding styles are commonly used or recommended by the community?
Frequently Asked Questions, 1926
While Loop
Control Structures - First Style, 64
Control Structures - Third Style, 75
Why do we need Yet Another Programming Language (YAPL)?
Frequently Asked Questions, 1925
Why I get a strange result when printing nl with lists?
Frequently Asked Questions, 1945
Why I get Calling Function without definition Error?
Frequently Asked Questions, 1951
Why is Ring largely focused on UI creation?
Frequently Asked Questions, 1930
Why is Ring weakly typed?
Frequently Asked Questions, 1930
Why Ring is not case-sensitive
Frequently Asked Questions, 1938
Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?
Frequently Asked Questions, 1934
Why Ring?
Language Design, 1614
Why setClickEvent() doesn't see the object methods directly?
Frequently Asked Questions, 1956
Why the ability to define your own languages Instead of parsing?
Frequently Asked Questions, 1934
Why the Assignment operator uses Deep copy?
Frequently Asked Questions, 1938
Why the window title bar is going outside the screen?
Frequently Asked Questions, 1952
Why this example use the GetChar() twice?
Frequently Asked Questions, 1947
Why Type Hints?
The Type Hints Library, 330
Why we don't use () after the QApp class name?
Frequently Asked Questions, 1952
Why you can specify the number of loops you want to break out of?
Frequently Asked Questions, 1934
Window Flags
Form Designer, 516
Window Resize Event
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 792
Windows StartUp Manager Application
What is new in Ring 1.5?, 1670
Windowsnl() Function
System Functions, 154
Word Game
What is new in Ring 1.13?, 1815
Wrap structures
Code Generator, 1590
Wrapping C++ Classes
Code Generator, 1597
Write file using Write()
Files, 138
Writing Comments
Getting Started - First Style, 49

Getting Started - Second Style, 51
Getting Started - Third Style, 54

Z

Z() function
Using ZeroLib, 951
Zero and Strings
What is new in Ring 1.16?, 1866
ZeroBasedList Class
Using ZeroLib, 951
ZeroBasedString Class
Using ZeroLib, 952
ZeroLib Library
What is new in Ring 1.11?, 1791
Zip Class Reference
RingZip, 1065
ZipEntry Class Reference
RingZip, 1065