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 1.10

Ring Team

January 25, 2019

CONTENTS

1 Applications developed in little hours **1**
1.1 Quotes about Ring . . . 1
1.2 FetchStockData Application . . . 5
1.3 Fifteen Puzzle Game 2 . . . 6
1.4 Google API Shortener Application . . . 8
1.5 Analog Clock . . . 9
1.6 TicTacToe Game . . . 9
1.7 Squares Puzzle Game . . . 10
1.8 Video-Music-Player Application . . . 11
1.9 Calculator Application . . . 12
1.10 Windows StartUp Manager Application . . . 13
1.11 Werdy Application . . . 14
1.12 Samples in this book . . . 15
1.13 Innovative . . . 22
1.14 Practical . . . 22

2 Introduction **24**
2.1 Motivation . . . 24
2.2 Ring and other languages . . . 25
2.3 Special Thanks to Matz . . . 25
2.4 History . . . 26
2.5 Features . . . 27
2.6 License . . . 30

3 Language Design **31**
3.1 Why Ring? . . . 31
3.2 Designed for a Clear Goal . . . 31
3.3 Simple . . . 31
3.4 Trying to be natural . . . 32
3.5 Encourage Organization . . . 33
3.6 Compact Syntax . . . 33
3.7 Define Natural Statements . . . 34
3.8 Define Declarative Languages . . . 36
3.9 Syntax Flexibility . . . 37
3.10 Transparent Implementation . . . 37
3.11 Visual Implementation . . . 38
3.12 Smart Garbage Collector . . . 39
3.13 No Global Interpreter (VM) Lock - No GIL . . . 40
3.14 Fast Enough For Many Applications . . . 40

4 What is new in Ring 1.10? **41**

4.1 List of changes and new features . . . 41
4.2 Chess Game . . . 41
4.3 Minesweeper Game . . . 42
4.4 Knight Tour Game . . . 43
4.5 Game of Life Game . . . 44
4.6 Pong Game . . . 45
4.7 Snakes And Ladder Game . . . 46
4.8 More Games . . . 47
4.9 Ring Extension for Visual Studio Code . . . 48
4.10 The Ring Package Manager (RingPM) . . . 49
4.11 Better Tests . . . 49
4.12 More Improvements . . . 50

5 What is new in Ring 1.9? 51
5.1 List of changes and new features . . . 51
5.2 New Game : Gold Magic 800 . . . 51
5.3 More Games . . . 53
5.4 Better Ring Notepad . . . 56
5.5 Better StdLib . . . 57
5.6 BigNumber Library . . . 58
5.7 RingPostgreSQL Extension . . . 59
5.8 Deploying Web applications in the Cloud . . . 61
5.9 Better RingQt . . . 62
5.10 Better Memory Management . . . 65
5.11 Better Code Generator for Extensions . . . 66
5.12 More Improvements . . . 66

6 What is new in Ring 1.8? 68
6.1 List of changes and new features . . . 68
6.2 Better Performance . . . 68
6.3 Find in files Application . . . 69
6.4 String2Constant Application . . . 69
6.5 StopWatch Application . . . 70
6.6 More 3D Samples . . . 71
6.7 Compiling on Manjaro Linux . . . 73
6.8 Using This in the class region as Self . . . 74
6.9 Default value for object attributes is NULL . . . 75
6.10 The For Loops uses the local scope . . . 75
6.11 Merge binary characters . . . 76
6.12 FoxRing Library . . . 77
6.13 Better Form Designer . . . 77
6.14 Better Cards Game . . . 78
6.15 Better RingQt . . . 78
6.16 Better Code Generator For Extensions . . . 79
6.17 Better Ring Compiler and VM . . . 79
6.18 Notes to extensions creators . . . 80

7 What is new in Ring 1.7? 81
7.1 List of changes and new features . . . 81
7.2 New Command: Load Package . . . 81
7.3 ringvm_see() and ringvm_give() functions . . . 82
7.4 ring_state_new() and ring_state_mainfile() functions . . . 83
7.5 Better Trace Library . . . 83
7.6 Better Ring Notepad . . . 84

7.7 Better RingQt . . . 84
7.8 Better Ring2EXE . . . 85
7.9 Better RingZip . . . 85
7.10 Better Documentation . . . 85
7.11 Better Ring VM . . . 85
7.12 RingLibuv Extension . . . 86

8 What is new in Ring 1.6? 89
8.1 List of changes and new features . . . 89
8.2 Employee Application . . . 89
8.3 New Tool: Ring2EXE . . . 90
8.4 Better Ring For Android . . . 90
8.5 New Tool: Folder2qrc . . . 92
8.6 Better Scripts for building Ring . . . 92
8.7 RingConsoleColors Extension . . . 92
8.8 RingMurmurHash Extension . . . 93
8.9 Better Ring Notepad . . . 93
8.10 Better RingQt . . . 95
8.11 Better StdLib . . . 96
8.12 Better RingVM . . . 96
8.13 Better RingREPL . . . 96
8.14 Using Tab instead of char(9) . . . 97
8.15 Using CR as Carriage return . . . 97
8.16 Using the ! operator as not . . . 97
8.17 Using && and || operators . . . 97
8.18 Using ? to print expression then new line . . . 98

9 What is new in Ring 1.5? 100
9.1 List of changes and new features . . . 100
9.2 Video-Music-Player Application . . . 100
9.3 Windows StartUp Manager Application . . . 101
9.4 Calculator Application . . . 102
9.5 Better Ring Notepad . . . 102
9.6 Better StdLib . . . 105
9.7 Better WebLib . . . 106
9.8 Better RingQt . . . 109
9.9 Better Objects Library . . . 111
9.10 RingFreeGLUT Extension . . . 111
9.11 RingOpenGL Extension . . . 121
9.12 Better Code Generator for Extensions . . . 124
9.13 Better Documentation Generator for Extensions . . . 125
9.14 Ring VM - Tracing Functions . . . 125
9.15 Trace Library and Interactive Debugger . . . 129
9.16 More Syntax Flexibility . . . 131
9.17 Type Hints Library . . . 132
9.18 Better Quality . . . 133
9.19 What is new in Ring 1.5.1? . . . 133
9.20 What is new in Ring 1.5.2? . . . 138
9.21 What is new in Ring 1.5.3? . . . 139
9.22 What is new in Ring 1.5.4? . . . 142

10 What is new in Ring 1.4? 144
10.1 List of changes and new features . . . 144
10.2 Change: Basic Extensions are separated from RingVM . . . 144

10.3 The Natural Library . . . 145
10.4 New Style is added to Ring Notepad . . . 147
10.5 RingREPL . . . 148
10.6 Convert between Numbers and Bytes . . . 148
10.7 Better StdLib . . . 149
10.8 Better WebLib . . . 149
10.9 Better RingQt . . . 151
10.10 Qt Class Convertor . . . 151
10.11 What is new in Ring 1.4.1? . . . 152

11 What is new in Ring 1.3? 156
11.1 List of changes and new features . . . 156
11.2 Better RingQt . . . 156
11.3 Better Ring Notepad . . . 160
11.4 Ring mode for Emacs Editor . . . 163
11.5 Better StdLib . . . 164
11.6 Better Loop|Exit Command . . . 164
11.7 New Functions . . . 165
11.8 Return Self by Reference . . . 165
11.9 Using '<' and ':' operators as 'from' keyword . . . 166
11.10 Embedding Ring in Ring without sharing the State . . . 166
11.11 RingZip Library . . . 167
11.12 Form Designer . . . 167

12 What is new in Ring 1.2? 169
12.1 List of changes and new features . . . 169
12.2 New Functions . . . 169
12.3 Better Functions . . . 170
12.4 Better Ring Notepad . . . 170
12.5 Better RingQt . . . 171
12.6 Objects Library for RingQt . . . 171
12.7 RingLibCurl . . . 172
12.8 Better Call Command . . . 173
12.9 Using NULL instead of NULLPointer() . . . 173
12.10 Display Warnings Option . . . 173
12.11 Better Quality . . . 174

13 What is new in Ring 1.1? 175
13.1 List of changes and new features . . . 175
13.2 Better Natural Language Programming Support . . . 175
13.3 Generate/Execute Ring Object Files (*.ringo) . . . 176
13.4 Syntax Flexibility and different styles for I/O and Control Structures . . . 176
13.5 New Functions and Changes . . . 178
13.6 StdLib functions and classes written in Ring . . . 179
13.7 RingLibSDL . . . 181
13.8 Demo Project - Game Engine for 2D Games . . . 181
13.9 RingSQLite . . . 182
13.10 Better Code Generator for Extensions . . . 183
13.11 Using Self.Attribute in the Class Region to define new attributes . . . 183
13.12 Using This.Attribute in nested Braces inside the Class Methods . . . 183
13.13 Better Documentation . . . 184

14 Building From Source Code 185
14.1 Building using Microsoft Windows . . . 185
14.2 Building using Ubuntu Linux . . . 187

14.3 Building using Fedora Linux . . . 189
14.4 Building using MacOS X . . . 192
14.5 Building using CMake . . . 194

15 How to contribute? 195
15.1 Special thanks to contributors . . . 195
15.2 Documentation . . . 195
15.3 Testing . . . 195
15.4 Samples . . . 196
15.5 Applications . . . 196
15.6 Editors Support . . . 196
15.7 Libraries in Ring . . . 196
15.8 Extensions in C/C++ . . . 196
15.9 Compiler and Virtual Machine (VM) . . . 196
15.10 Ideas and suggestions . . . 196

16 Getting Started - First Style 197
16.1 Hello World . . . 197
16.2 Run the program . . . 197
16.3 Create Executable File . . . 197
16.4 Not Case-Sensitive . . . 197
16.5 Multi-Line literals . . . 197
16.6 Getting Input . . . 198
16.7 No Explicit End For Statements . . . 198
16.8 Using ? to print expression then new line . . . 198
16.9 Writing Comments . . . 199

17 Getting Started - Second Style 200
17.1 Hello World . . . 200
17.2 Run the program . . . 200
17.3 Create Executable File . . . 200
17.4 Not Case-Sensitive . . . 200
17.5 Multi-Line literals . . . 200
17.6 Getting Input . . . 201
17.7 No Explicit End For Statements . . . 201
17.8 Writing Comments . . . 201

18 Getting Started - Third Style 202
18.1 Hello World . . . 202
18.2 Run the program . . . 202
18.3 Create Executable File . . . 202
18.4 Not Case-Sensitive . . . 203
18.5 Multi-Line literals . . . 203
18.6 Getting Input . . . 203
18.7 No Explicit End For Statements . . . 203
18.8 Writing Comments . . . 203

19 Using Ring Notepad 205
19.1 Ring Notepad - Main Window . . . 205
19.2 Creating and running your first Console Application . . . 206
19.3 Creating and running your first GUI/Mobile Application . . . 208
19.4 Creating and running your first Web Application . . . 209
19.5 Creating and running your first Desktop/Mobile Game . . . 210
19.6 The Main File in the Project . . . 211
19.7 The File Menu . . . 212

19.8 The Edit Menu . . . 213
19.9 The View Menu . . . 213
19.10 The Program Menu . . . 215
19.11 The Browser Menu . . . 215
19.12 The Tools Menu . . . 216
19.13 The Distribute Menu . . . 216
19.14 The Help Menu . . . 217

20 Using Other Code Editors **218**
20.1 Using Notepad++ . . . 218
20.2 Using Geany . . . 219
20.3 Using Atom . . . 220
20.4 Using Sublime Text 2 . . . 221
20.5 Using Visual Studio IDE . . . 222
20.6 Using Emacs Editor . . . 223
20.7 Visual Studio Code . . . 223

21 Variables **225**
21.1 Dynamic Typing . . . 225
21.2 Deep Copy . . . 225
21.3 Weakly Typed . . . 226

22 Operators **227**
22.1 Arithmetic Operators . . . 227
22.2 Relational Operators . . . 227
22.3 Logical Operators . . . 227
22.4 Bitwise Operators . . . 228
22.5 Assignment Operators . . . 228
22.6 Misc Operators . . . 228
22.7 Operators Precedence . . . 229

23 Control Structures - First Style **230**
23.1 Branching . . . 230
23.2 Looping . . . 231
23.3 Using The Step option with For in . . . 232
23.4 Using For in to modify lists . . . 233
23.5 Do Again Loop . . . 233
23.6 Exit Command . . . 233
23.7 Exit from two loops . . . 234
23.8 Loop Command . . . 234
23.9 Exit/Loop inside sub functions . . . 234
23.10 Short-circuit evaluation . . . 235
23.11 Comments about evaluation . . . 235

24 Control Structures - Second Style **237**
24.1 Branching . . . 237
24.2 Looping . . . 238
24.3 Exceptions . . . 239

25 Control Structures - Third Style **240**
25.1 Branching . . . 240
25.2 Looping . . . 241
25.3 Exceptions . . . 243

26 Getting Input **245**

26.1 Give Command . . . 245
26.2 GetChar() Function . . . 245
26.3 Input() Function . . . 246

27 Functions - First Style **247**
27.1 Define Functions . . . 247
27.2 Call Functions . . . 247
27.3 Declare parameters . . . 248
27.4 Send Parameters . . . 248
27.5 Main Function . . . 248
27.6 Variables Scope . . . 249
27.7 Return Value . . . 249
27.8 Recursion . . . 250

28 Functions - Second Style **251**
28.1 Define Functions . . . 251
28.2 Call Functions . . . 251
28.3 Declare parameters . . . 252
28.4 Send Parameters . . . 252
28.5 Main Function . . . 252
28.6 Variables Scope . . . 253
28.7 Return Value . . . 253
28.8 Recursion . . . 254

29 Functions - Third Style **255**
29.1 Define Functions . . . 255
29.2 Call Functions . . . 255
29.3 Declare parameters . . . 256
29.4 Send Parameters . . . 256
29.5 Main Function . . . 257
29.6 Variables Scope . . . 257
29.7 Return Value . . . 257
29.8 Recursion . . . 258

30 Program Structure **259**
30.1 Source Code File Sections . . . 259
30.2 Using Many Source Code Files . . . 259
30.3 Load Package . . . 260

31 Lists **261**
31.1 Create Lists . . . 261
31.2 Add Items . . . 261
31.3 Get List Size . . . 262
31.4 Delete Item From List . . . 262
31.5 Get List Item . . . 262
31.6 Set List Item . . . 263
31.7 Search . . . 263
31.8 Sort . . . 264
31.9 Reverse . . . 264
31.10 Insert Items . . . 265
31.11 Nested Lists . . . 265
31.12 Copy Lists . . . 265
31.13 First-class lists . . . 266
31.14 Using Lists during definition . . . 266
31.15 Passing Lists to Functions . . . 266

31.16 Access List Items by String Index . . . 267
31.17 Passing Parameters or Argumnents Using List . . . 267
31.18 Passing Parameters or Argumnents Using List Array . . . 268
31.19 Creating a Multi-Dimensional Array using List . . . 268
31.20 Swap Items . . . 269

32 Strings **271**
32.1 String Literals . . . 271
32.2 Get String Length . . . 271
32.3 Convert Letters Case . . . 271
32.4 Access String Letters . . . 272
32.5 Left() Function . . . 272
32.6 Right() Function . . . 272
32.7 Trim() Function . . . 273
32.8 Copy() Function . . . 273
32.9 Lines() Function . . . 273
32.10 Substr() Function . . . 274
32.11 Find substring . . . 274
32.12 Get substring from position to end . . . 274
32.13 Get Number of Characters From Position . . . 274
32.14 Transform Substring To Another Substring . . . 274
32.15 strcmp() Function . . . 275
32.16 str2list() and list2str() Functions . . . 275
32.17 Merge binary characters . . . 276

33 Date and Time **278**
33.1 Clock() Function . . . 278
33.2 ClocksPerSecond() Function . . . 278
33.3 Time() Function . . . 278
33.4 Date() Function . . . 279
33.5 TimeList() Function . . . 279
33.6 AddDays() Function . . . 280
33.7 DiffDays() Function . . . 280
33.8 EpochTime() Function . . . 281

34 Check Data Type and Conversion **282**
34.1 Check Data Type . . . 282
34.2 IsString() Function . . . 282
34.3 IsNumber() Function . . . 282
34.4 IsList() Function . . . 283
34.5 Type() Function . . . 283
34.6 IsNULL() Function . . . 283
34.7 Check Character . . . 284
34.8 IsAlNum() Function . . . 284
34.9 IsAlpha() Function . . . 284
34.10 IsCntrl() Function . . . 285
34.11 IsDigit() Function . . . 285
34.12 IsGraph() Function . . . 285
34.13 IsLower() Function . . . 285
34.14 IsPrint() Function . . . 286
34.15 IsPunct() Function . . . 286
34.16 IsSpace() Function . . . 286
34.17 IsUpper() Function . . . 286
34.18 IsXdigit() Function . . . 287

34.19 Conversion . . . 287
34.20 Number() Function . . . 287
34.21 String() Function . . . 287
34.22 Ascii() Function . . . 288
34.23 Char() Function . . . 288
34.24 Hex() Function . . . 288
34.25 Dec() Function . . . 288
34.26 Str2hex() Function . . . 289
34.27 Hex2str() Function . . . 289

35 Mathematical Functions **290**
35.1 List of functions . . . 290
35.2 Example . . . 290
35.3 Random() Function . . . 292
35.4 SRandom() Function . . . 293
35.5 Unsigned() Function . . . 293
35.6 Decimals() Functions . . . 294
35.7 Using _ in numbers . . . 294
35.8 Using f after numbers . . . 295

36 Files **296**
36.1 Read() Function . . . 297
36.2 Write() Function . . . 297
36.3 Dir() Function . . . 297
36.4 Rename() Function . . . 298
36.5 Remove() Function . . . 298
36.6 Fopen() Function . . . 298
36.7 Fclose() Function . . . 299
36.8 Fflush() Function . . . 299
36.9 Freopen() Function . . . 299
36.10 Tempfile() Function . . . 300
36.11 Tempname() Function . . . 300
36.12 Fseek() Function . . . 300
36.13 Ftell() Function . . . 300
36.14 Rewind() Function . . . 301
36.15 Fgetpos() Function . . . 301
36.16 Fsetpos() Function . . . 301
36.17 Clearerr() Function . . . 301
36.18 Feof() Function . . . 301
36.19 Ferror() Function . . . 301
36.20 Perror() Function . . . 302
36.21 Fgetc() Function . . . 302
36.22 Fgets() Function . . . 302
36.23 Fputc() Function . . . 302
36.24 Fputs() Function . . . 302
36.25 Ungetc() Function . . . 302
36.26 Fread() Function . . . 303
36.27 Fwrite() Function . . . 303
36.28 Fexists() Function . . . 303
36.29 Example . . . 303
36.30 Numbers and Bytes . . . 305

37 System Functions **306**
37.1 System() Function . . . 306

37.2 SysGet() Function . . . 307
37.3 IsMSDOS() Function . . . 307
37.4 IsWindows() Function . . . 307
37.5 IsWindows64() Function . . . 307
37.6 IsUnix() Function . . . 307
37.7 IsMacOSX() Function . . . 308
37.8 IsLinux() Function . . . 308
37.9 IsFreeBSD() Function . . . 308
37.10 IsAndroid() Function . . . 308
37.11 Example . . . 308
37.12 Windowsnl() Function . . . 309
37.13 Get Command Line Arguments . . . 309
37.14 Get Active Source File Name . . . 310
37.15 PrevFileName() Function . . . 310
37.16 CurrentDir() Function . . . 311
37.17 ExeFileName() Function . . . 311
37.18 ChDir() Function . . . 311
37.19 ExeFolder() Function . . . 311
37.20 Version() Function . . . 311
37.21 Shutdown() Function . . . 312

38 Eval() and Debugging **313**
38.1 Try/Catch/Done . . . 313
38.2 Eval() Function . . . 313
38.3 Raise() Function . . . 314
38.4 Assert() Function . . . 315

39 Demo Programs **316**
39.1 Language Shell . . . 316
39.2 Main Menu . . . 317

40 ODBC Functions **320**
40.1 odbc_init() Function . . . 320
40.2 odbc_drivers() Function . . . 321
40.3 odbc_datasources() Function . . . 321
40.4 odbc_close() Function . . . 321
40.5 Print List of ODBC Drivers . . . 321
40.6 Print List of ODBC Data Sources . . . 322
40.7 odbc_connect() Function . . . 323
40.8 odbc_disconnect() Function . . . 323
40.9 Open and Close Connection . . . 323
40.10 odbc_execute() Function . . . 323
40.11 odbc_colcount() Function . . . 323
40.12 odbc_fetch() Function . . . 324
40.13 odbc_getdata() Function . . . 324
40.14 Execute Query and Print Result . . . 324
40.15 odbc_tables() Function . . . 324
40.16 odbc_columns() Function . . . 325
40.17 odbc_autocommit() Function . . . 326
40.18 odbc_commit() Function . . . 326
40.19 odbc_rollback() Function . . . 326
40.20 Transactions and Using Commit and Rollback . . . 326
40.21 Save and Restore images . . . 327

41 MySQL Functions **329**

41.1 MySQL_Info() Function . . . 329
41.2 MySQL_Init() Function . . . 330
41.3 MySQL_Error() Function . . . 330
41.4 MySQL_Connect() Function . . . 330
41.5 MySQL_Close() Function . . . 330
41.6 MySQL_Query() Function . . . 330
41.7 Create Database . . . 331
41.8 Create Table and Insert Data . . . 331
41.9 MySQL_Insert_ID() Function . . . 332
41.10 MySQL_Result() Function . . . 333
41.11 MySQL_Next_Result() Function . . . 333
41.12 Print Query Result . . . 333
41.13 MySQL_Columns() Function . . . 333
41.14 MySQL_Result2() Function . . . 334
41.15 MySQL_Escape_String() Function . . . 335
41.16 Save Image inside the database . . . 335
41.17 Restore Image From The Database . . . 336
41.18 MySQL_AutoCommit() Function . . . 336
41.19 MySQL_Commit() Function . . . 336
41.20 MySQL_Rollback() Function . . . 336
41.21 Transaction Example . . . 336

42 SQLite Functions **338**
42.1 sqlite_init() function . . . 338
42.2 sqlite_open() function . . . 338
42.3 sqlite_execute() function . . . 338
42.4 sqlite_close() function . . . 338
42.5 Example . . . 339

43 PostgreSQL Functions **341**
43.1 Loading the library . . . 341
43.2 Examples . . . 341
43.3 RingPostgreSQL Constants . . . 344
43.4 RingPostgreSQL Functions . . . 345

44 Security and Internet Functions **348**
44.1 MD5() Function . . . 348
44.2 SHA1() Function . . . 349
44.3 SHA256() Function . . . 349
44.4 SHA512() Function . . . 349
44.5 SHA384() Function . . . 350
44.6 SHA224() Function . . . 350
44.7 Encrypt() Function . . . 350
44.8 Decrypt() Function . . . 351
44.9 Encryption and Decryption Example . . . 351
44.10 File Hash . . . 352
44.11 Randbytes() Function . . . 352
44.12 Download() Function . . . 352
44.13 SendEmail() Function . . . 353

45 Object Oriented Programming (OOP) **354**
45.1 Classes and Objects . . . 354
45.2 Access Objects Using Braces . . . 356
45.3 Composition . . . 357
45.4 Setter and Getter . . . 358

45.5 Private Attributes and Methods . . . 358
45.6 Operator Overloading . . . 359
45.7 Inheritance . . . 362
45.8 Dynamic Attributes . . . 363
45.9 Packages . . . 363
45.10 Printing Objects . . . 364
45.11 Find() and List of Objects . . . 364
45.12 Sort() and List of Objects . . . 365
45.13 Using Self.Attribute and Self.Method() . . . 367
45.14 Using This.Attribute and This.Method() . . . 367
45.15 Using This in the class region as Self . . . 368
45.16 Default value for object attributes . . . 369

46 Functional Programming **370**
46.1 Pure Functions . . . 370
46.2 First-class Functions . . . 371
46.3 Higher-order Functions . . . 371
46.4 Anonymous and Nested Functions . . . 372
46.5 Equality of functions . . . 373

47 Reflection and Meta-programming **375**
47.1 locals() Function . . . 376
47.2 globals() Function . . . 376
47.3 functions() Function . . . 377
47.4 cfunctions() Function . . . 377
47.5 islocal() Function . . . 378
47.6 isglobal() Function . . . 378
47.7 isfunction() Function . . . 379
47.8 iscfunction() Function . . . 379
47.9 packages() Function . . . 379
47.10 ispackage() Function . . . 380
47.11 classes() Function . . . 381
47.12 isclass() Function . . . 381
47.13 packageclasses() Function . . . 382
47.14 ispackageclass() Function . . . 382
47.15 classname() Function . . . 383
47.16 objectid() Function . . . 383
47.17 isobject() Function . . . 384
47.18 attributes() Function . . . 384
47.19 methods() Function . . . 384
47.20 isattribute() Function . . . 385
47.21 isprivateattribute() Function . . . 385
47.22 ismethod() Function . . . 386
47.23 isprivatemethod() Function . . . 386
47.24 addattribute() Function . . . 387
47.25 addmethod() Function . . . 387
47.26 getattribute() function . . . 389
47.27 setattribute() function . . . 390
47.28 mergemethods() Function . . . 391
47.29 packagename() Function . . . 392

48 Stdlib Functions **393**
48.1 Puts() function . . . 393
48.2 Print() function . . . 393

48.3 Print2Str() Function . . . 394
48.4 GetString() function . . . 394
48.5 GetNumber() function . . . 394
48.6 AppPath() function . . . 394
48.7 JustFilePath() function . . . 394
48.8 JustFileName() function . . . 395
48.9 Value() function . . . 395
48.10 Times() function . . . 395
48.11 Map() function . . . 396
48.12 Filter() function . . . 396
48.13 Split() function . . . 396
48.14 SplitMany() function . . . 397
48.15 NewList() function . . . 397
48.16 Capitalized() function . . . 397
48.17 IsSpecial() function . . . 398
48.18 IsVowel() function . . . 398
48.19 LineCount() function . . . 398
48.20 Factorial() function . . . 398
48.21 Fibonacci() function . . . 399
48.22 IsPrime() function . . . 399
48.23 Sign() function . . . 399
48.24 List2File() function . . . 400
48.25 File2List() function . . . 400
48.26 StartsWith() function . . . 400
48.27 EndsWith() function . . . 400
48.28 GCD() function . . . 401
48.29 LCM() function . . . 401
48.30 SumList() function . . . 401
48.31 ProdList() function . . . 402
48.32 EvenOrOdd() function . . . 402
48.33 Factors() function . . . 402
48.34 IsPalindrome() function . . . 403
48.35 IsLeapYear() function . . . 403
48.36 BinaryDigits() function . . . 403
48.37 MatrixMulti() function . . . 404
48.38 MatrixTrans() function . . . 404
48.39 DayOfWeek() function . . . 404
48.40 Permutation() function . . . 405
48.41 ReadLine() function . . . 405
48.42 SubString() function . . . 405
48.43 ChangeString() function . . . 406
48.44 Sleep() function . . . 406
48.45 IsMainSourceFile() function . . . 406
48.46 DirExists() function . . . 407
48.47 MakeDir() function . . . 407
48.48 Fsize() function . . . 407
48.49 TrimAll() function . . . 407
48.50 TrimLeft() function . . . 408
48.51 TrimRight() function . . . 408
48.52 EpochTime() function . . . 408
48.53 SystemCmd() Function . . . 408
48.54 ListAllFiles() Function . . . 408
48.55 SystemSilent() Function . . . 409
48.56 OSCreateOpenFolder() Function . . . 409

48.57 OSCopyFolder() Function . . . 409
48.58 OSDeleteFolder() Function . . . 409
48.59 OSCopyFile() Function . . . 410
48.60 OSDeleteFile() Function . . . 410
48.61 OSRenameFile() Function . . . 410
48.62 List2Code() Function . . . 410
48.63 Str2ASCIIList() . . . 410
48.64 ASCIIList2Str() . . . 411

49 Stdlib Classes **412**
49.1 StdBase Class . . . 412
49.2 String Class . . . 413
49.3 List Class . . . 415
49.4 Stack Class . . . 417
49.5 Queue Class . . . 417
49.6 HashTable Class . . . 418
49.7 Tree Class . . . 419
49.8 Math Class . . . 420
49.9 DateTime Class . . . 424
49.10 File Class . . . 425
49.11 System Class . . . 426
49.12 Debug Class . . . 427
49.13 DataType Class . . . 428
49.14 Conversion Class . . . 429
49.15 ODBC Class . . . 430
49.16 MySQL Class . . . 431
49.17 SQLite Class . . . 432
49.18 PostgreSQL Class . . . 433
49.19 Security Class . . . 435
49.20 Internet Class . . . 435

50 Declarative Programming using Nested Structures **436**
50.1 Creating Objects inside Lists . . . 436
50.2 Composition and Returning Objects and Lists by Reference . . . 437
50.3 Executing code after the end of object access . . . 439
50.4 Declarative Programming on the top of Object-Oriented . . . 439
50.5 More beautiful Code . . . 440

51 Natural Language Programming **442**
51.1 History . . . 442
51.2 Example . . . 442
51.3 Change the Ring Keyword ‘And’ . . . 443
51.4 Change the Ring Operator ‘+’ . . . 444
51.5 Change the ‘=’ operator to ‘is’ . . . 445
51.6 Using Eval() with our Natural Code . . . 446
51.7 BraceStart and BraceEnd Methods . . . 447
51.8 BraceExprEval Method . . . 448
51.9 Real Natural Code . . . 448
51.10 BraceError() Method . . . 449
51.11 Clean Natural Code . . . 450

52 Using the Natural Library **452**
52.1 Natural Library - Demo Program . . . 452
52.2 Defining Commands . . . 454
52.3 Natural Library - Operators . . . 457

52.4 Defining commands using classes . . . 457

53 Web Development (CGI Library) 459
53.1 Configure the Apache web server . . . 459
53.2 Ring CGI Hello World Program . . . 460
53.3 Hello World Program using the Web Library . . . 460
53.4 Web Library Features . . . 461
53.5 HTTP Get Example . . . 461
53.6 HTTP POST Example . . . 466
53.7 Upload Files . . . 468
53.8 Cookies . . . 471
53.9 URL Encode . . . 473
53.10 Templates . . . 474
53.11 HTML Special Characters . . . 476
53.12 Hash Functions . . . 477
53.13 Random Image . . . 479
53.14 HTML Lists . . . 480
53.15 HTML Tables . . . 482
53.16 Gradient . . . 483
53.17 Generating Pages using Objects . . . 484
53.18 HtmlPage Class . . . 488
53.19 Using Bootstrap Library using Functions . . . 489
53.20 Using Bootstrap Library using Objects . . . 490
53.21 CRUD Example using MVC . . . 493
53.22 Users registration and Login . . . 495
53.23 Database, ModelBase & ControllerBase classes . . . 501
53.24 WebLib API . . . 506
53.25 Application Class . . . 508
53.26 Page Class . . . 508
53.27 ScriptFunctions Class . . . 510
53.28 StyleFunctions Class . . . 511
53.29 WebPage Class . . . 511
53.30 HtmlPage Class . . . 512

54 Deploying Web Applications in the Cloud 513
54.1 Introduction . . . 513
54.2 Usage . . . 513
54.3 Ring source code files and permissions . . . 514
54.4 Hello World program . . . 515
54.5 Application Database . . . 515
54.6 Deploying after updates . . . 515
54.7 Local Tests . . . 516

55 Using RingLibCurl 517
55.1 Get Request . . . 517
55.2 Post Request . . . 517
55.3 Facebook Login . . . 517
55.4 Save Output to String . . . 518
55.5 Get Stock Data From Yahoo . . . 519

56 Using RingZip 521
56.1 Create Zip File . . . 521
56.2 Extract Zip File . . . 521
56.3 Print Files in Zip file . . . 521
56.4 Using RingZip Classes . . . 521

56.5 Zip Class Reference . . . 523
56.6 ZipEntry Class Reference . . . 523

57 Graphics and 2D Games programming using RingAllegro 524
57.1 Drawing, Animation and Input . . . 524
57.2 Using TrueType Fonts . . . 531
57.3 Playing Sound Files . . . 532
57.4 Scaling and Rotating Images . . . 533
57.5 Display Transparent Image . . . 534
57.6 Using Threads . . . 535

58 Using RingLibSDL 538
58.1 Create Window . . . 538
58.2 Display Image . . . 538
58.3 Switch between two images . . . 539
58.4 Draw Rectangle . . . 539
58.5 Display PNG Images . . . 540
58.6 Use TTF Fonts . . . 540
58.7 Display Transparent Images . . . 540
58.8 Close Window Event . . . 541
58.9 Mouse Events . . . 542
58.10 Play Sound . . . 543

59 Using RingLibuv 545
59.1 First Application using RingLibuv . . . 545
59.2 The Events Loop . . . 546
59.3 Server Example . . . 546
59.4 Client Example . . . 548
59.5 Server Example Using Classes . . . 549
59.6 Client Example Using Classes . . . 550
59.7 Threads Example . . . 551
59.8 Threads Example - Using Classes . . . 552

60 Demo Project - Game Engine for 2D Games 553
60.1 Project Layers . . . 553
60.2 Graphics Library bindings . . . 553
60.3 Interface to graphics library . . . 554
60.4 Game Engine Classes . . . 554
60.5 Games Layer . . . 554
60.6 Game Class . . . 555
60.7 GameObject Class . . . 555
60.8 Sprite Class . . . 556
60.9 Text Class . . . 556
60.10 Animate Class . . . 557
60.11 Sound Class . . . 557
60.12 Map Class . . . 557
60.13 Using the Game Engine - Creating the Game Window . . . 558
60.14 Using the Game Engine - Drawing Text . . . 558
60.15 Using the Game Engine - Moving Text . . . 559
60.16 Using the Game Engine - Playing Sound . . . 561
60.17 Using the Game Engine - Animation . . . 562
60.18 Using the Game Engine - Animation and Functions . . . 563
60.19 Using the Game Engine - Sprite - Automatic Movement using Keyboard . . . 565
60.20 Using the Game Engine - Sprite - Keypress event . . . 566
60.21 Using the Game Engine - Sprite - Mouse event . . . 566

60.22 Using the Game Engine - Sprite - State event . . . 567
60.23 Using the Game Engine - Animate - Events . . . 568
60.24 Using the Game Engine - Map . . . 570
60.25 Using the Game Engine - Map Events . . . 572
60.26 Using the Game Engine - Object and Drawing . . . 573
60.27 Stars Fighter Game . . . 576
60.28 Flappy Bird 3000 Game . . . 584
60.29 Super Man 2016 Game . . . 591

61 Building Games For Android 603
61.1 Download Requirements and Update the Android SDK . . . 603
61.2 Project Folder . . . 603
61.3 Building the project . . . 604

62 Using RingOpenGL and RingFreeGLUT for 3D Graphics 606
62.1 Samples Source (Authors) . . . 606
62.2 What is RingOpenGL? . . . 606
62.3 What is RingFreeGLUT? . . . 607
62.4 The First Window using RingFreeGLUT . . . 607
62.5 Drawing using RingOpenGL . . . 608
62.6 The First Triangle . . . 610
62.7 Window Resize Event . . . 611
62.8 Triangle Rotation . . . 612
62.9 Keyboard Events and Colors . . . 613
62.10 The Camera . . . 616
62.11 Mouse Events . . . 623
62.12 Menu Events . . . 626
62.13 Using Fonts . . . 633
62.14 Frames Per Second . . . 641
62.15 Make a Cube using RingOpenGL and RingFreeGLUT . . . 650

63 Using RingOpenGL and RingAllegro for 3D Graphics 654
63.1 3D Cube and Texture . . . 654
63.2 Many Cubes . . . 658
63.3 TicTacToe 3D Game . . . 664
63.4 More 3D Samples . . . 673

64 The Gold Magic 800 Game 676
64.1 The Game Story . . . 676
64.2 How to play? . . . 676
64.3 What will you learn? . . . 676
64.4 Screen Shots . . . 677
64.5 Source Code . . . 678

65 Desktop and Mobile development using RingQt 680
65.1 The First GUI Application . . . 680
65.2 The Events Loop . . . 682
65.3 Using Layout . . . 682
65.4 Using the QTextEdit Class . . . 683
65.5 Using the QListWidget Class . . . 684
65.6 Using QTreeView and QFileSystemModel . . . 687
65.7 Using QTreeWidget and QTreeWidgetItem . . . 688
65.8 Using QComboBox Class . . . 690
65.9 Creating Menubar . . . 691
65.10 Context Menu . . . 693

65.11 Creating Toolbar . . . 694
65.12 Creating StatusBar . . . 695
65.13 Using QDockWidget . . . 696
65.14 Using QTabWidget . . . 697
65.15 Using QTableWidget . . . 699
65.16 Using QProgressBar . . . 700
65.17 Using QSpinBox . . . 701
65.18 Using QSlider . . . 702
65.19 Using QDateEdit . . . 703
65.20 Using QDial . . . 704
65.21 Using QWebView . . . 707
65.22 Using QCheckBox . . . 708
65.23 Using QRadioButton and QButtonGroup . . . 709
65.24 Adding Hyperlink to QLabel . . . 711
65.25 QVideoWidget and QMediaPlayer . . . 712
65.26 Using QFrame . . . 714
65.27 Display Image using QLabel . . . 715
65.28 Menubar and StyleSheet Example . . . 716
65.29 QLineEdit Events and QMessageBox . . . 718
65.30 Other Widgets Events . . . 720
65.31 Using the QTimer Class . . . 722
65.32 Using QProgressBar and Timer . . . 723
65.33 Display Scaled Image using QLabel . . . 724
65.34 Using the QFileDialog Class . . . 725
65.35 Drawing using QPainter . . . 726
65.36 Printing using QPrinter . . . 728
65.37 Using QPrintPreviewDialog . . . 729
65.38 Creating More than one Window . . . 731
65.39 Playing Sound . . . 732
65.40 Using the QColorDialog Class . . . 733
65.41 Using qLCDNumber Class . . . 734
65.42 Movable Label Example . . . 735
65.43 QMessagebox Example . . . 736
65.44 Using QInputDialog Class . . . 737
65.45 Dialog Functions . . . 738
65.46 KeyPress and Mouse Move Events . . . 739
65.47 Moving Objects using the Mouse . . . 740
65.48 Inheritance from GUI Classes . . . 744
65.49 Using QDesktopWidget Class . . . 745
65.50 Rotate Text . . . 746
65.51 Change Focus . . . 748
65.52 Regular Expressions . . . 749
65.53 Simple Client and Server Example . . . 750
65.54 Dynamic Objects . . . 752
65.55 Weight History Application . . . 753
65.56 Notepad Application . . . 757
65.57 The Cards Game . . . 772
65.58 Classes and their Methods to use the default events . . . 778
65.59 Methods to use Events with Events Filter . . . 781
65.60 The Difference between Qt and RingQt . . . 782
65.61 RingQt Classes and their Qt Documentation . . . 783
65.62 New Classes names - Index Start from 1 . . . 783
65.63 Creating Reports using the WebLib and the GUILib . . . 784

66 Building RingQt Applications for Mobile **787**
66.1 Download Requirements . . . 787
66.2 Update the Android SDK . . . 787
66.3 Install Qt for Android . . . 788
66.4 Using Ring2EXE . . . 788
66.5 The Qt project for your Ring application . . . 788
66.6 Comments about developing for Android using RingQt . . . 789

67 Objects Library for RingQt Application **791**
67.1 Library Usage . . . 791
67.2 Example . . . 791
67.3 Open_WindowAndLink() Function . . . 794
67.4 Open_WindowInPackages() Function . . . 795
67.5 Objects Library Source Code . . . 795

68 Using the Form Designer **796**
68.1 The Designer Windows . . . 797
68.2 The Toolbox . . . 797
68.3 The Properties . . . 797
68.4 Running Forms . . . 798
68.5 Events Code . . . 798
68.6 Keyboard Shortcuts . . . 801
68.7 Menubar Designer . . . 801
68.8 Window Flags . . . 802
68.9 Entering Items . . . 803
68.10 Using Layouts . . . 803
68.11 More Samples and Tests . . . 803

69 Multi-language Applications **804**
69.1 Using String2Constant . . . 804
69.2 Form Designer Translation . . . 805
69.3 Forms Translation . . . 806

70 Scope Rules for Variables and Attributes **808**
70.1 Three Scopes . . . 808
70.2 Defining Variables and Variables Access . . . 808
70.3 How Ring find the variable? . . . 809
70.4 Using Object.Attribute . . . 809
70.5 The Self Object . . . 809
70.6 How Ring Define Variables and Attributes . . . 810
70.7 Conflict between Global Variables and Class Attributes . . . 810
70.8 Conflict between Class Attributes and Local Variables . . . 811
70.9 Using Braces to access objects inside Class Methods . . . 812
70.10 Accessing the class attributes from braces inside class methods . . . 815
70.11 Creating a Class for each Window in GUI applications . . . 816
70.12 Conflict between self inside braces and self in the class region . . . 817
70.13 Using braces to escape from the current object scope . . . 820
70.14 The For Loops uses the local scope . . . 821
70.15 Summary of Scope Rules . . . 821

71 Scope Rules for Functions and Methods **824**
71.1 How Ring find the Functions and Methods? . . . 824
71.2 Example about Sharing Names between Functions and Methods . . . 825
71.3 Calling a function sharing the name with a method in the current class . . . 826

72 Syntax Flexibility **828**
72.1 Change Language Keywords . . . 828
72.2 Change Language Operators . . . 829
72.3 Load Syntax Files . . . 830
72.4 Using "()" around the function parameters . . . 831
72.5 Using Semi-colon after and between statements . . . 831
72.6 Using $ and @ in the start of the variable name . . . 832
72.7 Using the 'elseif' keyword as 'but' in if statement . . . 832
72.8 Using the 'else' keyword as 'other' in switch statement . . . 833
72.9 Using the 'end' keyword in different control structures . . . 833
72.10 Using braces to start and end different control structures . . . 834
72.11 Using 'put' and 'get' as 'see' and 'give' . . . 835
72.12 Using 'case' as 'on' in switch statements . . . 835
72.13 Using 'def' as 'func' in functions/methods definition . . . 836
72.14 Using braces { } in Packages/Classes/Functions . . . 836
72.15 Using 'end' keyword after Packages/Classes/Functions . . . 836
72.16 Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions . . . 837
72.17 Ignore new lines after keywords . . . 837

73 Introduction to the Type Hints Library **838**
73.1 Why Type Hints? . . . 838
73.2 Example . . . 838
73.3 User Types . . . 838
73.4 Using Types inside Code . . . 839
73.5 Rules . . . 839

74 Command Line Options **841**
74.1 Printing Tokens . . . 841
74.2 Printing Rules . . . 843
74.3 Printing Intermediate Code . . . 847
74.4 Printing Final Intermediate Code . . . 850
74.5 CGI Support . . . 854
74.6 No Run . . . 854
74.7 Printing Instruction Operation Code . . . 854
74.8 Performance . . . 854
74.9 Generate Object File . . . 855

75 Performance Tips **856**
75.1 Introduction . . . 856
75.2 Creating Lists . . . 858
75.3 Arithmetic Operations . . . 859
75.4 Using len() and For Loops . . . 859
75.5 Calling Functions and Methods . . . 860

76 Distributing Ring Applications **862**
76.1 Distributing Applications for Microsoft Windows . . . 862
76.2 Protecting the Source Code . . . 862
76.3 The files ring.ring and ring.ringo . . . 863
76.4 Creating Windows Installer . . . 863
76.5 Using C/C++ Compiler and Linker . . . 863
76.6 Distributing Applications and Games for Mobile . . . 863

77 Distributing Ring Applications using Ring2EXE **864**
77.1 Using Ring2EXE . . . 864
77.2 How Ring2EXE works? . . . 864

77.3 Example . . . 865
77.4 Options . . . 865
77.5 Building standalone console application . . . 866
77.6 Distributing RingAllegro Applications . . . 866
77.7 Distributing RingQt Applications . . . 867
77.8 Distributing Applications for Mobile using RingQt . . . 867
77.9 Building the Cards Game for Mobile using RingQt . . . 868
77.10 Building the Weight History Application for Mobile using RingQt . . . 868
77.11 Building the Form Designer for Mobile using RingQt . . . 869
77.12 Creating the Qt resource file using Folder2qrc . . . 871
77.13 Important Information about Ring2EXE . . . 871

78 The Ring Package Manager (RingPM) 873
78.1 Features . . . 873
78.2 Discovering Packages . . . 873
78.3 Updating the RingPM Registry . . . 874
78.4 Installing Packages . . . 874
78.5 Printing List of Installed Packages . . . 875
78.6 Run Package . . . 875
78.7 Update Package . . . 876
78.8 Remove Package . . . 876
78.9 Deleting All Packages . . . 876
78.10 Creating New Package . . . 876
78.11 The Package Description File . . . 878
78.12 Create Package in the Current Folder . . . 879
78.13 The RingPM Registry File . . . 881

79 Low Level Functions 883
79.1 callgc() function . . . 884
79.2 varptr() function . . . 884
79.3 space() function . . . 885
79.4 nullpointer() function . . . 885
79.5 object2pointer() function . . . 885
79.6 pointer2object() function . . . 886
79.7 ispointer() function . . . 886
79.8 ptrcmp() function . . . 887
79.9 ringvm_cfunctionslist() function . . . 887
79.10 ringvm_functionslist() function . . . 887
79.11 ringvm_classeslist() function . . . 888
79.12 ringvm_packageslist() function . . . 889
79.13 ringvm_memorylist() function . . . 890
79.14 ringvm_calllist() function . . . 892
79.15 ringvm_fileslist() function . . . 892
79.16 ringvm_settrace() . . . 893
79.17 ringvm_tracedata() . . . 893
79.18 ringvm_traceevent() . . . 894
79.19 ringvm_tracefunc() . . . 894
79.20 ringvm_scopescount() . . . 894
79.21 ringvm_evalinscope() . . . 895
79.22 ringvm_passerror() . . . 895
79.23 ringvm_hideerrormsg() . . . 895
79.24 ringvm_callfunc() . . . 895
79.25 Example - Using the Trace Functions . . . 895
79.26 Example - The Trace Library . . . 899

79.27 ringvm_see() function . . . 903
79.28 ringvm_give() function . . . 903
79.29 ringvm_info() function . . . 903

80 The Trace Library and the Interactive Debugger 905
80.1 Loading the Trace library . . . 905
80.2 Trace All Events . . . 905
80.3 Trace control flow between functions . . . 905
80.4 Pass Error . . . 906
80.5 Interactive Debugger . . . 906
80.6 Execute Program Line by Line . . . 906
80.7 BreakPoint . . . 907
80.8 Disable BreakPoints . . . 907
80.9 Using the Interactive Debugger . . . 907

81 Embedding Ring in Ring 910
81.1 Embedding Ring in Ring without sharing the State . . . 910
81.2 Serial Execution of Programs . . . 911
81.3 ring_state_setvar() . . . 911
81.4 ring_state_new() and ring_state_mainfile() . . . 912
81.5 Runtime Errors when Embedding Ring in Ring . . . 913

82 Extension using the C/C++ languages 914
82.1 ring_ext.h . . . 914
82.2 ring_ext.c . . . 914
82.3 Module Organization . . . 915
82.4 Function Structure . . . 916
82.5 Check Parameters Count . . . 916
82.6 Display Error Message . . . 917
82.7 Check Parameters Type . . . 917
82.8 Get Parameters Values . . . 917
82.9 Return Value . . . 918
82.10 Function Prototype . . . 918
82.11 Sin() Function Implementation . . . 918
82.12 Fopen() and Fclose() Functions Implementation . . . 918
82.13 Ring API - List Functions . . . 919
82.14 Ring API - String Functions . . . 921
82.15 MySQL_Columns() Function Implementation . . . 921
82.16 Dynamic/Shared Libraries (DLL/So/Dylib) and LoadLib() function . . . 922
82.17 Using RING_API_RETMANAGEDCPOINTER() . . . 923

83 Embedding Ring Language in C/C++ Programs 925
83.1 Ring State . . . 925
83.2 Ring State Functions . . . 925
83.3 Ring State Variables . . . 926

84 Code Generator for wrapping C/C++ Libraries 928
84.1 Using the tool . . . 928
84.2 Configuration file . . . 928
84.3 Using the function prototype . . . 928
84.4 Adding code to the generated code . . . 930
84.5 Prefix for Functions Names . . . 930
84.6 Generate function to wrap structures . . . 930
84.7 Determine Structure Members Types . . . 931
84.8 Defining Constants . . . 931

84.9 Register New Functions . . . 931
84.10 Writing comments in the configuration file . . . 932
84.11 Executing code during code generation . . . 932
84.12 Enum and Numbers . . . 932
84.13 Filtering using Expressions . . . 933
84.14 Constants Type . . . 933
84.15 Configuration file for the Allegro Library . . . 933
84.16 Threads Support . . . 935
84.17 Code Generator Rules for Wrapping C++ Classes . . . 936
84.18 Using configuration file that wrap C++ Library . . . 937
84.19 Configuration file for the Qt Framework . . . 937
84.20 Static Methods . . . 947
84.21 Loading Files . . . 947
84.22 Managed Classes . . . 947
84.23 Configuration Files Examples . . . 948

85 Create your first extension **949**
85.1 Location . . . 949
85.2 Steps to create the extension . . . 949
85.3 Testing the extension . . . 950

86 Frequently Asked Questions (FAQ) **951**
86.1 Why do we need Yet Another Programming Language (YAPL)? . . . 951
86.2 Why is Ring weakly typed? . . . 952
86.3 What are the advantages to using Ring over Lisp or Smalltalk? . . . 952
86.4 Why is Ring largely focussed on UI creation? . . . 953
86.5 Is Ring some sort of an improvement of PHP? . . . 953
86.6 What are the advantages of using Ring over native C or C++? . . . 953
86.7 What is the difference between Ring and Python? And is Ring Open Source? . . . 953
86.8 What are the advantages to using Ring over Perl, PHP, Python or Ruby? . . . 954
86.9 What are the advantages to using Ring over Tcl or Lua? . . . 955
86.10 What are the advantages to using Ring over C# or Java? . . . 955
86.11 The documentation says functional programming is supported, but then this happens? . . . 955
86.12 Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like? . . . 956
86.13 Why you can specify the number of loops you want to break out of? . . . 956
86.14 Why Ring uses ‘See’, ‘Give’, ‘But’ and ‘Ok’ Keywords? . . . 956
86.15 What is the philosophy behind data types in Ring? . . . 957
86.16 What about the Boolean values in Ring? . . . 957
86.17 What is the goal of including the “Main” function in Ring? . . . 958
86.18 Why the list index start from 1 in Ring? . . . 959
86.19 Why Ring is not case-sensitive? . . . 959
86.20 Why the Assignment operator uses Deep Copy? . . . 960
86.21 Is there constructor methods in Ring? . . . 961
86.22 What happens when we create a new object? . . . 961
86.23 Can we use the attributes by accessing the Getter and Setter methods? . . . 962
86.24 Why should a search of global names be made while defining the class attributes? . . . 963
86.25 Why Ring doesn’t avoid the conflict between Global Variables and Class Attributes Names? . . . 964
86.26 Where can I write a program and execute it? . . . 965
86.27 How to get the file size using ftell() and fseek() functions? . . . 965
86.28 How to get the current source file path? . . . 965
86.29 What about predefined parameters or optional parameters in functions? . . . 965
86.30 How to print keys or values only in List/Dictionary? . . . 966
86.31 Why I get a strange result when printing nl with lists? . . . 966

86.32 Could you explain the output of the StrCmp() function? . . . 967
86.33 How to use many source code files in the project? . . . 967
86.34 Why this example use the GetChar() twice? . . . 968
86.35 How to use NULL and ISNULL() function? . . . 969
86.36 How to print lists that contains objects? . . . 970
86.37 How to insert an item to the first position in the list? . . . 971
86.38 How to print new lines and other characters? . . . 971
86.39 Why we don't use () after the qApp class name? . . . 971
86.40 Why the window title bar is going outside the screen? . . . 972
86.41 How to create an array of buttons in GUI applications? . . . 972
86.42 How to Close a window then displaying another one? . . . 973
86.43 How to create a Modal Window? . . . 973
86.44 How can I disable maximize button and resize window? . . . 974
86.45 How to use SQLite using ODBC? . . . 975
86.46 Can I connect to dbase/harbour database? . . . 976
86.47 Why setClickEvent() doesn't see the object methods directly? . . . 977
86.48 Why I get Calling Function without definition Error? . . . 977
86.49 Can Ring work on Windows XP? . . . 978
86.50 How to extend RingQt and add more classes? . . . 978
86.51 How to add Combobox and other elements to the cells of a QTableWidget? . . . 981
86.52 How to perform some manipulations on selected cells in QTableWidget? . . . 981
86.53 Which of 3 coding styles are commonly used or recommended by the community? . . . 982

87 General Information **983**
87.1 Ring Architecture . . . 983
87.2 Memory Management . . . 983
87.3 Data Representation . . . 984

88 Language Reference **987**
88.1 Language Keywords . . . 987
88.2 Language Functions . . . 989
88.3 Compiler Errors . . . 990
88.4 Runtime Errors . . . 990
88.5 Environment Errors . . . 992
88.6 Language Grammar . . . 992
88.7 Virtual Machine (VM) Instructions . . . 993

89 RingQt Classes Reference **1000**
89.1 CodeEditor Class . . . 1000
89.2 QAbstractButton Class . . . 1000
89.3 QAbstractItemView Class . . . 1001
89.4 QAbstractPrintDialog Class . . . 1003
89.5 QAbstractScrollArea Class . . . 1003
89.6 QAbstractSlider Class . . . 1004
89.7 QAbstractSocket Class . . . 1005
89.8 QAbstractSpinBox Class . . . 1006
89.9 QAction Class . . . 1007
89.10 QAllEvents Class . . . 1009
89.11 QApp Class . . . 1013
89.12 QAxBase Class . . . 1013
89.13 QAxObject Class . . . 1015
89.14 QAxWidget Class . . . 1015
89.15 QBitmap Class . . . 1015
89.16 QBluetoothAddress Class . . . 1015

89.17 QBluetoothDeviceDiscoveryAgent Class . . . 1016
89.18 QBluetoothDeviceInfo Class . . . 1016
89.19 QBluetoothHostInfo Class . . . 1017
89.20 QBluetoothLocalDevice Class . . . 1017
89.21 QBluetoothServer Class . . . 1018
89.22 QBluetoothServiceDiscoveryAgent Class . . . 1018
89.23 QBluetoothServiceInfo Class . . . 1019
89.24 QBluetoothSocket Class . . . 1020
89.25 QBluetoothTransferManager Class . . . 1021
89.26 QBluetoothTransferReply Class . . . 1021
89.27 QBluetoothTransferRequest Class . . . 1021
89.28 QBluetoothUuid Class . . . 1022
89.29 QBoxLayout Class . . . 1022
89.30 QBrush Class . . . 1023
89.31 QBuffer Class . . . 1023
89.32 QButtonGroup Class . . . 1023
89.33 QByteArray Class . . . 1024
89.34 QCalendarWidget Class . . . 1026
89.35 QCamera Class . . . 1027
89.36 QCameraImageCapture Class . . . 1028
89.37 QCameraViewfinder Class . . . 1029
89.38 QChar Class . . . 1029
89.39 QCheckBox Class . . . 1031
89.40 QChildEvent Class . . . 1032
89.41 QClipboard Class . . . 1032
89.42 QColor Class . . . 1032
89.43 QColorDialog Class . . . 1035
89.44 QComboBox Class . . . 1035
89.45 QCompleter Class . . . 1037
89.46 QCompleter2 Class . . . 1038
89.47 QCompleter3 Class . . . 1038
89.48 QCoreApplication Class . . . 1038
89.49 QCursor Class . . . 1040
89.50 QDate Class . . . 1040
89.51 QDateEdit Class . . . 1041
89.52 QDateTime Class . . . 1041
89.53 QDateTimeEdit Class . . . 1042
89.54 QDesktopServices Class . . . 1043
89.55 QDesktopWidget Class . . . 1044
89.56 QDial Class . . . 1044
89.57 QDialog Class . . . 1045
89.58 QDir Class . . . 1045
89.59 QDirModel Class . . . 1047
89.60 QDockWidget Class . . . 1048
89.61 QDrag Class . . . 1049
89.62 QDragEnterEvent Class . . . 1049
89.63 QDragLeaveEvent Class . . . 1049
89.64 QDragMoveEvent Class . . . 1050
89.65 QDropEvent Class . . . 1050
89.66 QEvent Class . . . 1050
89.67 QFile Class . . . 1051
89.68 QFile2 Class . . . 1051
89.69 QFileDevice Class . . . 1052
89.70 QFileDialog Class . . . 1052

89.71 QFileInfo Class . . . 1053
89.72 QFileSystemModel Class . . . 1055
89.73 QFont Class . . . 1056
89.74 QFontDialog Class . . . 1058
89.75 QFontMetrics Class . . . 1058
89.76 QFrame Class . . . 1059
89.77 QFrame2 Class . . . 1060
89.78 QFrame3 Class . . . 1060
89.79 QGeoAddress Class . . . 1060
89.80 QGeoAreaMonitorInfo Class . . . 1061
89.81 QGeoAreaMonitorSource Class . . . 1061
89.82 QGeoCircle Class . . . 1061
89.83 QGeoCoordinate Class . . . 1062
89.84 QGeoPositionInfo Class . . . 1062
89.85 QGeoPositionInfoSource Class . . . 1063
89.86 QGeoRectangle Class . . . 1063
89.87 QGeoSatelliteInfo Class . . . 1064
89.88 QGeoSatelliteInfoSource Class . . . 1064
89.89 QGeoShape Class . . . 1064
89.90 QGradient Class . . . 1065
89.91 QGraphicsVideoItem Class . . . 1065
89.92 QGridLayout Class . . . 1065
89.93 QGuiApplication Class . . . 1066
89.94 QHBoxLayout Class . . . 1068
89.95 QHeaderView Class . . . 1068
89.96 QHostAddress Class . . . 1071
89.97 QHostInfo Class . . . 1071
89.98 QIODevice Class . . . 1072
89.99 QIcon Class . . . 1073
89.100 QImage Class . . . 1073
89.101 QInputDialog Class . . . 1074
89.102 QJsonArray Class . . . 1076
89.103 QJsonDocument Class . . . 1076
89.104 QJsonObject Class . . . 1077
89.105 QJsonParseError Class . . . 1077
89.106 QJsonValue Class . . . 1078
89.107 QKeySequence Class . . . 1078
89.108 QLCDNumber Class . . . 1078
89.109 QLabel Class . . . 1079
89.110 QLayout Class . . . 1080
89.111 QLineEdit Class . . . 1081
89.112 QLinearGradient Class . . . 1083
89.113 QListView Class . . . 1083
89.114 QListWidget Class . . . 1084
89.115 QListWidgetItem Class . . . 1085
89.116 QMainWindow Class . . . 1086
89.117 QMatrix4x4 Class . . . 1088
89.118 QMdiArea Class . . . 1089
89.119 QMdiSubWindow Class . . . 1090
89.120 QMediaObject Class . . . 1091
89.121 QMediaPlayer Class . . . 1091
89.122 QMediaPlaylist Class . . . 1092
89.123 QMenu Class . . . 1092
89.124 QMenuBar Class . . . 1093

89.125QMessageBox Class . . . 1094
89.126QMimeData Class . . . 1095
89.127QMutex Class . . . 1096
89.128QMutexLocker Class . . . 1096
89.129QNetworkAccessManager Class . . . 1096
89.130QNetworkProxy Class . . . 1097
89.131QNetworkReply Class . . . 1098
89.132QNetworkRequest Class . . . 1098
89.133QNmeaPositionInfoSource Class . . . 1099
89.134QObject Class . . . 1099
89.135QOpenGLBuffer Class . . . 1100
89.136QOpenGLContext Class . . . 1100
89.137QOpenGLDebugLogger Class . . . 1101
89.138QOpenGLFramebufferObject Class . . . 1102
89.139QOpenGLFunctions Class . . . 1102
89.140QOpenGLFunctions_3_2_Core Class . . . 1107
89.141QOpenGLPaintDevice Class . . . 1114
89.142QOpenGLShader Class . . . 1115
89.143QOpenGLShaderProgram Class . . . 1115
89.144QOpenGLTexture Class . . . 1119
89.145QOpenGLTimerQuery Class . . . 1121
89.146QOpenGLVersionProfile Class . . . 1121
89.147QOpenGLVertexArrayObject Class . . . 1122
89.148QOpenGLWidget Class . . . 1122
89.149QPageSetupDialog Class . . . 1123
89.150QPaintDevice Class . . . 1123
89.151QPainter Class . . . 1123
89.152QPainter2 Class . . . 1126
89.153QPainterPath Class . . . 1126
89.154QPen Class . . . 1127
89.155QPicture Class . . . 1128
89.156QPixmap Class . . . 1128
89.157QPixmap2 Class . . . 1130
89.158QPlainTextEdit Class . . . 1130
89.159QPoint Class . . . 1132
89.160QPointF Class . . . 1133
89.161QPrintPreviewDialog Class . . . 1133
89.162QPrintPreviewWidget Class . . . 1133
89.163QPrinter Class . . . 1134
89.164QPrinterInfo Class . . . 1136
89.165QProcess Class . . . 1136
89.166QProgressBar Class . . . 1137
89.167QPushButton Class . . . 1138
89.168QQmlError Class . . . 1138
89.169QQuaternion Class . . . 1139
89.170QQuickWidget Class . . . 1139
89.171QRadioButton Class . . . 1140
89.172QRect Class . . . 1140
89.173QRegion Class . . . 1142
89.174QRegularExpression Class . . . 1143
89.175QRegularExpressionMatch Class . . . 1143
89.176QRegularExpressionMatchIterator Class . . . 1144
89.177QScreen Class . . . 1144
89.178QScrollArea Class . . . 1145

89.179QScrollBar Class . . . 1146
89.180QSerialPort Class . . . 1146
89.181QSerialPortInfo Class . . . 1147
89.182QSize Class . . . 1148
89.183QSlider Class . . . 1148
89.184QSpinBox Class . . . 1148
89.185QSplashScreen Class . . . 1149
89.186QSplitter Class . . . 1149
89.187QSqlDatabase Class . . . 1150
89.188QSqlDriver Class . . . 1151
89.189QSqlDriverCreatorBase Class . . . 1152
89.190QSqlError Class . . . 1152
89.191QSqlField Class . . . 1152
89.192QSqlIndex Class . . . 1153
89.193QSqlQuery Class . . . 1153
89.194QSqlRecord Class . . . 1154
89.195QStackedWidget Class . . . 1155
89.196QStandardPaths Class . . . 1156
89.197QStatusBar Class . . . 1156
89.198QString2 Class . . . 1156
89.199QStringList Class . . . 1157
89.200QStringRef Class . . . 1158
89.201QSurfaceFormat Class . . . 1160
89.202QSystemTrayIcon Class . . . 1161
89.203QTabBar Class . . . 1161
89.204QTabWidget Class . . . 1163
89.205QTableView Class . . . 1164
89.206QTableWidget Class . . . 1166
89.207QTableWidgetItem Class . . . 1168
89.208QTcpServer Class . . . 1169
89.209QTcpSocket Class . . . 1170
89.210QTest Class . . . 1170
89.211QTextBlock Class . . . 1170
89.212QTextBrowser Class . . . 1171
89.213QTextCharFormat Class . . . 1172
89.214QTextCodec Class . . . 1174
89.215QTextCursor Class . . . 1174
89.216QTextDocument Class . . . 1176
89.217QTextEdit Class . . . 1178
89.218QTextStream Class . . . 1180
89.219QTextStream2 Class . . . 1182
89.220QTextStream3 Class . . . 1182
89.221QTextStream4 Class . . . 1182
89.222QTextStream5 Class . . . 1182
89.223QThread Class . . . 1182
89.224QThreadPool Class . . . 1183
89.225QTime Class . . . 1184
89.226QTimer Class . . . 1184
89.227QToolBar Class . . . 1185
89.228QToolButton Class . . . 1186
89.229QTreeView Class . . . 1186
89.230QTreeWidget Class . . . 1188
89.231QTreeWidgetItem Class . . . 1191
89.232QUrl Class . . . 1192

89.233 QUuid Class . . . 1193
89.234 QVBoxLayout Class . . . 1193
89.235 QVariant Class . . . 1194
89.236 QVariant2 Class . . . 1195
89.237 QVariant3 Class . . . 1195
89.238 QVariant4 Class . . . 1195
89.239 QVariant5 Class . . . 1195
89.240 QVector2D Class . . . 1196
89.241 QVector3D Class . . . 1196
89.242 QVector4D Class . . . 1197
89.243 QVideoWidget Class . . . 1198
89.244 QVideoWidgetControl Class . . . 1198
89.245 QWebEngineView Class . . . 1198
89.246 QWebView Class . . . 1200
89.247 QWebView Class . . . 1200
89.248 QWidget Class . . . 1201
89.249 QWindow Class . . . 1206
89.250 QXmlStreamAttribute Class . . . 1210
89.251 QXmlStreamAttributes Class . . . 1210
89.252 QXmlStreamEntityDeclaration Class . . . 1211
89.253 QXmlStreamEntityResolver Class . . . 1211
89.254 QXmlStreamNamespaceDeclaration Class . . . 1211
89.255 QXmlStreamNotationDeclaration Class . . . 1211
89.256 QXmlStreamReader Class . . . 1212
89.257 QXmlStreamWriter Class . . . 1213
89.258 RingCodeHighlighter Class . . . 1214

90 RingMurmurHash Functions Reference 1215
90.1 MurmurHash1 functions . . . 1215
90.2 MurmurHash2 functions . . . 1215
90.3 MurmurHash3 functions . . . 1215
90.4 Example . . . 1216

91 FoxRing Functions Reference 1217
91.1 FoxRing functions . . . 1217
91.2 frAbs() function . . . 1218
91.3 frAsc() function . . . 1218
91.4 frAddBs() function . . . 1219
91.5 frAt() function . . . 1219
91.6 frAtC() function . . . 1220
91.7 frChr() function . . . 1220
91.8 frEmpty() function . . . 1220
91.9 frFile() function . . . 1221
91.10 frFileToStr() function . . . 1221
91.11 frStr() function . . . 1221
91.12 frSetIfEmpty() function . . . 1222
91.13 frSpace() function . . . 1223
91.14 frInList() function . . . 1223
91.15 frForcePath() function . . . 1223
91.16 frAllTrim() function . . . 1223
91.17 frLTrim() function . . . 1224
91.18 frJustDrive() function . . . 1224
91.19 frJustExt() function . . . 1224
91.20 frJustStem() function . . . 1224

91.21 frRTrim() function . . . 1224
91.22 frJustPath() function . . . 1225
91.23 frForceExt() function . . . 1225
91.24 frALines() function . . . 1225
91.25 frJustFName() function . . . 1225
91.26 frPadL() function . . . 1225
91.27 frPadR() function . . . 1225
91.28 frProper() function . . . 1225
91.29 frReplicate() function . . . 1225
91.30 frLen() function . . . 1226
91.31 frStuff() function . . . 1226
91.32 frSubStr() function . . . 1226
91.33 frStrTran() function . . . 1226
91.34 frListToString() function . . . 1226
91.35 frInt() function . . . 1227
91.36 frStringToList() function . . . 1227
91.37 frIIf() function . . . 1227
91.38 frVal() function . . . 1227
91.39 frBetween() function . . . 1227
91.40 frSetSeparatorTo() function . . . 1228
91.41 frTransform() function . . . 1228
91.42 frVarType() function . . . 1228
91.43 Example . . . 1229

92 BigNumber Library 1231
92.1 Loading the library . . . 1231
92.2 Examples . . . 1231
92.3 BigNumber Functions . . . 1232
92.4 BigNumber Class . . . 1233
92.5 Library Source Code . . . 1233

93 RingLibCurl Functions Reference 1234

94 RingLibZip Functions Reference 1243

95 RingConsoleColors Functions Reference 1244

96 RingAllegro Functions Reference 1246

97 RingLibSDL Functions Reference 1265

98 RingLibuv Functions Reference 1290

99 RingFreeGLUT Functions Reference 1299

100 RingOpenGL (OpenGL 1.1) Functions Reference 1308

101 RingOpenGL (OpenGL 1.2) Functions Reference 1333

102 RingOpenGL (OpenGL 1.3) Functions Reference 1359

103 RingOpenGL (OpenGL 1.4) Functions Reference 1388

104 RingOpenGL (OpenGL 1.5) Functions Reference 1418

105 RingOpenGL (OpenGL 2.0) Functions Reference 1449

106RingOpenGL (OpenGL 2.1) Functions Reference **1483**

107RingOpenGL (OpenGL 3.0) Functions Reference **1525**

108RingOpenGL (OpenGL 3.1) Functions Reference **1570**

109RingOpenGL (OpenGL 3.2) Functions Reference **1616**

110RingOpenGL (OpenGL 3.3) Functions Reference **1663**

111RingOpenGL (OpenGL 4.0) Functions Reference **1709**

112RingOpenGL (OpenGL 4.1) Functions Reference **1756**

113RingOpenGL (OpenGL 4.2) Functions Reference **1803**

114RingOpenGL (OpenGL 4.3) Functions Reference **1850**

115RingOpenGL (OpenGL 4.4) Functions Reference **1897**

116RingOpenGL (OpenGL 4.5) Functions Reference **1944**

117RingOpenGL (OpenGL 4.6) Functions Reference **1991**

118Resources **2039**

118.1 Ring Language Website 2039
118.2 Ring Group 2039
118.3 Source Code 2039
118.4 Arabic Resources 2039
118.5 Contact the Authors 2040

Index **2041**

CHAPTER

ONE

APPLICATIONS DEVELOPED IN LITTLE 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. It's just released in 2016.01.25! In little days 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 little 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. Why bother with C/C++ when the syntax can be cryptic and they are largely antiquated - why should we still be dealing with header files, etc? Moreover, your comments on Java and C# being too verbose and forcing OOP onto the user is spot on as well - yet these languages are still primarily used by millions of programmers for the most mundane tasks. At this point, I'm led to believe that it's simply a matter of legacy applications that still haven't made the transition, as well as the ever-popular "Appeal to Popularity" fallacy shared by many in academia and enterprise. Then we come to languages, that are easier and much more modern, and yet even now, are being to show their age. Languages like Python which stresses whitespace and indentation, as well as (just like the formerly mentioned languages) irrelevant tokens for the conditional statements. Granted, it's not as bad as tracking down various curly braces or semi-colons, but one misstep in indentation might as well be as frustrating to track down. The same could be true of Ruby, except with Ruby, we have even odder conventions embedded into the design of the language, which I don't feel the need to address and moreover, we're left with something like Lua which is missing a great many features that the previously mentioned languages include. There are many more languages and faults with them, we could claim that PHP is nothing more than a glue for web pages and backends but with so many frameworks available now, it's falling by the wayside fast. Javascript is constantly being cloned. So, each of these general purpose languages, which we all know are mostly domain-specific and in some cases, also oriented at certain OS(C# was before 2016 and Obj-C/Swift will likely always be primarily for MacOS apps). So, then we're left with several additional options, involving everything from hybrids to other imperative-based languages like D, GoLang, Scala, Julia, etc. But finally, there's a new language called Ring - and yes, it may just end up ruling them all. It's the unicorn we've finally been waiting to arrive, that can handle multiple domains, tasks, and paradigms. I honestly can't wait to jump in and when I do I probably won't come back up anytime soon. My only apprehension is that the language may not include a library or primitive functions for math as Python and Julia do. Aside from that, if so, hopefully, I won't be having to use linear expressions as arrays (considering how intellectually lazy it is to do) and that there will be more support/tools as the community continues to grow. 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 i , Wim ten Brink (Quora - Top Writer (2018))

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

"Ring is a serious thing in the programming language landscape. One shoud 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 distruption, 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 todays 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 intutively. 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 assesing 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

1.2 FetchStockData Application

URL : https://groups.google.com/forum/#!topic/ring-lang/-fa1U_SXSjo

Author : Bert Mariani

This App is written in Ring.

It will fetch stock data from Yahoo and draw various types of charts. Any valid stock ticker can be entered, or selected from the drop down list - Select Symbol.

It will not guarantee that you make money in the stock market. But it will visualize the history of the stock.

1.3 Fifteen Puzzle Game 2

URL : https://github.com/ring-lang/ring/blob/master/applications/fifteenpuzzle/CalmoSoftFifteenPuzzleGame.ring

Author : Gal Zsolt (CalmoSoft)

CalmoSoft Fifteen Puzzle Game
4x4
5x5
6x6
7x7
34 15 6 33 23 5
12 26 4 8 31
11 1 3 22 17 18
30 2 21 10 19 6
25 20 35 14 24 16
28 7 32 29 13 27
16
<-
Here
->
Scramble
Reset
Save Game
Resume Game
Sleep Time: 1 s
<-
->
In the Right Place : 1
Elapsed Time : 14.28 s

1.4 Google API Shortener Application

Author : John Storm (SienSystem)

Hi everyone,

Wanted to (finally) share a first release version of a small but very nice tool to shorten urls using the Google shortener API. The code is fully documented, maybe too documented, but, provides a good reference as to what is going on.

You can obtain your own API key, or you can use my key to test and use the tool.

Hope you enjoy the application. Please feel free to test at your convenience.

Ring is FUN!!

1.5 Analog Clock

URL : https://github.com/ring-lang/ring/blob/master/applications/analogclock/AnalogClock-Image.ring

Author : Bert Mariani

1.6 TicTacToe Game

URL : https://github.com/AbdelrahmanGIT/RingSamples/blob/master/src/TecTacToe.ring

Author : Abdelrahman Mohammed

1.7 Squares Puzzle Game

URL : https://github.com/MajdiSobain/RingAllegro_SquaresPuzzle

Author : Majdi Sobain

This project is about (Squares Puzzle) popular game that I have programmed using ring language with its RingAllegro Library. The principle of this game is very known to all of us, which is moving squares to get the real full shape of the original picture.

This game could be played using mouse and keyboard as well, showing a message of congratulations at the successful solving.

1.8 Video-Music-Player Application

Author : Bert Mariani

Screen Shot:

1.9 Calculator Application

Author : Magdy Ragab

Updated Version by Gal Zsolt and Bert Mariani

Screen Shot:

1.10 Windows StartUp Manager Application

URL : https://github.com/ring-lang/WinStartupManager

Author : Majdi Sobain

Windows Startup Manager is an application that has the ability to let the user control what programs should start at Windows booting time. It gives you the ability to show, edit, delete, or even add new programs entries to be lunched at Windows booting time.

It is specialized at managing Windows startup entries that are stored in Windows Registry only, but provides a quick option to edit programs shortcuts type entries.

Screen Shot:

1.11 Werdy Application

URL : https://github.com/ring-lang/werdy

Author : Magdy Ragab

Quran application includes reading suras, searching and bookmarking.

The applcation is provided for Windows, Linux and Android.

1.12 Samples in this book

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

The next screen shot for he 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

1.13 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

1.14 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/applications/formdesigner

We can run the Form Designer as Android application!

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 (Windows, Linux, macOS, Android, etc.) and can be used to create Console, GUI, Web, Games and Mobile applications. The language is designed to be Simple, Small, Flexible and Fast. Its Dynamic Language (Dynamic Typing and Weakly Typed) 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

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. Most of the PWCT source code was written in VFP and the software comes with a simple scripting language for creating the components called (RPWI). The software contains components that support code generation in programming languages like Harbour, C, Supernova & Python.

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 & can be used for developing different kinds of applications.

Instead of using a mix of programming languages, I decided to use one programming language for creating the development environment, for components scripting & for creating the applications.

I looked at many programming languages like C, C++, Java, C#, Lua, PHP, Python & Ruby. I avoided using C or C++ directly because I want high-level of productivity more than the level provided by these languages, also a language behind visual programming environment for novice programmers or professionals must be easy to use & productive.

Java & C# are avoided for some reason too! I wanted to use a dynamic programming language and these languages are static typing, Java is multi-platform, also C# through Mono, but the use of huge number of classes and forcing the use of Object-Orientation, using a verbose language is not right for me. I need a small language, but fast and productive, also I need better control on the Garbage Collector (GC), I need a better one that is designed for fast applications.

Lua is small and fast, but it's avoided because I need more powerful language for large applications.

PHP is a Web programming language and it's syntax is very similar to C, this leads to a language not general as I want and not simple as I need to have.

Python & Ruby are more like what I need, but I need something more simple, smaller, faster & productive.

Python and Ruby are Case-Sensitive, the list index start counting from 0, you have to define the function before calling it, Ruby usage of Object-Orientation and message passing is more than what I need and decrease performance, Python syntax (indentation, using self, :, pass & _) is not good for my goals.

All of these languages are successful languages, and very good for their domains, but what I need is a different language that comes with new ideas and intelligent implementation (Innovative, Ready, Simple, Small, Flexible and Fast).

2.2 Ring and other languages

Ring is an innovative 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

2.3 Special Thanks to Matz

Special Thanks to Matz (Yukihiro Matsumoto - The Creator of Ruby Programming Language) for spreading the word.

2.4 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

2.5 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)
- Writen in ANSI C (The code is generated + Looks identical to Handwritten Code)
- **A small language**
  - The compiler + The Virtual Machine (20,000 lines of C code)
  - The other 500,000 lines of code are related to libraries!
- Portable (Windows, Linux, macOS, Android, 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%) using Escape Analysis we don't 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)
- Use exit to go outside from loop in the Caller function (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
- 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
- Create GUI Applications for Desktop 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
- No Global Interpreter (VM) lock (No GIL) - Better for threads and concurrency (Faster)
- Comes with RingPM (Package Manager)
- Many Samples and Applications
- Complete Documentation.

2.6 License

The Ring Programming Language

http://ring-lang.net/

Version 1.10

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

LANGUAGE DESIGN

In this chapter we will learn about the basic concepts behind the language design.

3.1 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.

3.2 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.

3.3 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
```

3.4 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"
                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
```

3.5 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!"
```

3.6 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
```

3.7 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 beatiful programming world!
}
# Our Class
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

        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

3.8 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
          {
                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)
```

3.9 Syntax Flexibility

Ring comes with many styles for writing your source code!

Also you can change the language keywords and operators and create your custom style!

3.10 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

=================================================================

=================================================================
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!
```

3.11 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 "visualsrc" - *.ssf files and the generated source code (In the C Language) in the src folder and the 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)

3.12 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.

3.13 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!)

3.14 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.

Ring can do each of the next tasks in around 1 second using normal computers in the market during the last 5 years

1. Compiling 100,000 lines of code
2. Executing empty loop that count from 1 to 10,000,000
3. Executing 1000 search operation using linear search in a list contains 100,000 items, trying to find the last item (The worst case)
4. Creating list contains 1,000,000 items then summing all of the list items
5. Adding 20,000 items to the ListWidget in GUI applications
6. Adding 5,000 nodes to the TreeWidget in GUI applications
7. Printing 10,000 messages to the terminal in Console applications

Also when we need more speed we can use C/C++ extensions!

CHAPTER

FOUR

WHAT IS NEW IN RING 1.10?

In this chapter we will learn about the changes and new features in Ring 1.10 release.

4.1 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

4.2 Chess Game

Chess Game (2 Players)

4.3 Minesweeper Game

The objective of the game is to clear a rectangular board containing hidden "mines" or bombs without detonating any of them.

4.4 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?

4.5 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.

4.6 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.

4.7 Snakes And Ladder Game

The Snakes and Ladder Game using Ring Game Engine for 2D Games

4.8 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

4.9 Ring Extension for Visual Studio Code

This extension support the Ring programming language in Visual Studio Code

4.10 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
==========================================================================
```

4.11 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
```

4.12 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

FIVE

WHAT IS NEW IN RING 1.9?

In this chapter we will learn about the changes and new features in Ring 1.9 release.

5.1 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

5.2 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

5.3 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

5.4 Better Ring Notepad

1. New Style
2. New Mode
3. Tabs for opened files
4. Support executing batch files

5.5 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"

aList = Str2ASCIILIST(cStr)
? aList
```

```
cStr2 = ASCIIList2Str(aList)
? cStr2
? len(cStr2)
```

Output:

```
77
109
77
109

MmMm
4
```

5.6 BigNumber Library

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

? "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)
3716981956283439342389882561522895 08 (Output continue in next line)
4939646110432289716923898608970 69
```

For more information check the BigNumber Library Chapter in the documentation

5.7 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)

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
4 Ibrahim  24 Egypt   40
```

For more information check the PostgreSQL Chapter in the documentation

5.8 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

Demo : http://testring.herokuapp.com/

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.

5.9 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) {
                        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:

5.10 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

5.11 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 configuraition 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 Garabage Collector will manage these C pointers.

Example

```
<class>
name: QFont
para: QString, int, int, bool
managed
</class>
```

5.12 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 Aliginment for Toolbox Icons.
29. Form Desginer - 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

SIX

WHAT IS NEW IN RING 1.8?

In this chapter we will learn about the changes and new features in Ring 1.8 release.

6.1 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

6.2 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.

6.3 Find in files Application

Ring 1.8 comes with Find in files application

6.4 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.

6.5 StopWatch Application

Ring 1.8 comes with StopWatch application

6.6 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

6.7 Compiling on Manjaro Linux

Ring 1.8 is tested on Manjaro Linux too

Tests by : Iip Rifai

6.8 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.

6.9 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.

6.10 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
```

6.11 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
? "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
```

6.12 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)
? mf.frAbs(30-10)

? mf.frTransform("    Ring is a good language    ",
                 "@! !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!")
? mf.frAllTrim("    Ring is a good language    ", Null)
```

Output:

```
45
20
20
    RING IS A GOOD LANGUAGE
Ring is a good language
```

6.13 Better Form Designer

1. Layout Control - Display the control name when loading forms.
2. Button Control - Display the button images written using relative path.
3. Table Control - Display the control name when loading forms.
4. Better behavior in “Bring to front” and “Send to back” operations.
5. New buttons are added to the toolbar (Duplicate, Bring to front, Send to back, Delete).
6. Using layouts in (Menubar designer, Window Flags window, Selecting Objects window).

7. Better behavior for displaying the properties window when changing the selected objects.
8. New buttons are added to move and resize multiple selection of objects.
9. Window Properties - Add button to select the layout.
10. Opening forms and switching between files is faster.
11. Objects Order window.
12. Select Objects window.
13. When we change the control name, the name will be updated in layout objects.

6.14 Better Cards Game

The Cards game is updated and we can play with the Computer

6.15 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.

6.16 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))
```

6.17 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.

6.18 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

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.

7.1 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

7.2 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
```

7.3 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
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
```

7.4 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.

7.5 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()

func test1
        ? "Hello from test1!"
        test2()

func test2
        ? "Hello from test2!"
        test3()

func test3
        ? "Hello from test3!"
        breakpoint()
```

7.6 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?

7.7 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

7.8 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.

7.9 Better RingZip

The library is updated to support extracting files contains sub folders!

7.10 Better Documentation

1. RingQt Classes Chapter - The classes are sorted.

7.11 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

7.12 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());
        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
                        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()
                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 1.6?

In this chapter we will learn about the changes and new features in Ring 1.6 release.

8.1 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

8.2 Employee Application

The Employee application is added to ring/applications

Developer: Ahmed Hassouna

8.3 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!

8.4 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
```

8.5 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!

8.6 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.

8.7 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.

8.8 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.

8.9 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.

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

8.10 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

8.11 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

8.12 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

8.13 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.

8.14 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
```

8.15 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
```

8.16 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!
```

8.17 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

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
```

8.18 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

NINE

WHAT IS NEW IN RING 1.5?

In this chapter we will learn about the changes and new features in Ring 1.5 release.

9.1 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

9.2 Video-Music-Player Application

The Video-Music-Player application is added to the Applications folder.

Screen Shot:

9.3 Windows StartUp Manager Application

The Windows StartUp Manager

URL : https://github.com/ring-lang/WinStartupManager

Screen Shot:

9.4 Calculator Application

The Calculator application is added to the Applications folder.

Screen Shot:

9.5 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
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
ring

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:

9.6 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
```

9.7 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) }
                          }
                }

                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
                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

                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:

9.8 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()
                           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
#
###----------------------------------------
```

9.9 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"
])
```

9.10 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

// 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()

        // 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)

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


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()

        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
        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
                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
                        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)


// -----------------------------------
//             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:

9.11 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)

  // 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 )
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()

  //  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:

9.12 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>
```

9.13 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.

9.14 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()
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
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
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 ======
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
=========================================
```

9.15 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

9.16 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

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
```

9.17 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.

9.18 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.

9.19 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

                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)
                        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()
        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
                                                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:

9.20 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:

9.21 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"

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:

9.22 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

TEN

WHAT IS NEW IN RING 1.4?

In this chapter we will learn about the changes and new features in Ring 1.4 release.

10.1 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 Convertor

10.2 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"
```

10.3 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)
        }
])
```

10.4 New Style is added to Ring Notepad

In Ring Notepad - From View - Styles - Select the (Modern) Style

Screen Shot:

10.5 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:

10.6 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
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
```

10.7 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                           ;
```

10.8 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 paramter.

```
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 {
                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%"
                                                style="background-color:white;">
                                                </iframe>`)
                                                }
                                        }
                                }
```

```
                        }
                }

        }
        html(template("footer.rhtml",NULL))
}
```

10.9 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

10.10 Qt Class Convertor

Ring 1.4 comes with a simple tool that help in porting Qt classes to RingQt.

You will find it in ring/samples/tools/QtClassConverter

Online : https://github.com/ring-lang/ring/tree/master/samples/tools/QtClassConverter

Screen Shot:

10.11 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

ELEVEN

WHAT IS NEW IN RING 1.3?

In this chapter we will learn about the changes and new features in Ring 1.3 release.

11.1 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

11.2 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)
                {
                        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
        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/applications/formdesigner/tests/indexstart/indexstartView.ring

2. https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/indexstart/indexstartController.ring

11.3 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

11.4 Ring mode for Emacs Editor

Ring 1.3 comes with Ring mode for Emacs Editor

Screen Shot:

11.5 Better StdLib

The StdLib is updated to include the next functions

- SplitMany()
- JustFilePath()
- JustFileName()

11.6 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
```

11.7 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
```

11.8 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

class mytest
        x = 15
        func init
                return self      # Return by reference
```

Output

```
x: 15.000000
x: 20.000000
x: 20.000000
```

11.9 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
```

11.10 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")
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
```

11.11 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()
}
```

11.12 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.

CHAPTER

TWELVE

WHAT IS NEW IN RING 1.2?

In this chapter we will learn about the changes and new features in Ring 1.2 release.

12.1 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

12.2 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
```

12.3 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).

12.4 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.

12.5 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)
```

12.6 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 ) )
                }
                resize(400,400)
        }
```

12.7 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)
```

12.8 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)
```

12.9 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)
```

12.10 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
```

12.11 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

THIRTEEN

WHAT IS NEW IN RING 1.1?

In this chapter we will learn about the changes and new features in Ring 1.1 release.

13.1 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

13.2 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
}

# 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.

13.3 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.

13.4 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.

13.5 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

13.6 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")
        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

13.7 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.

13.8 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
                        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.

13.9 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
next
see copy("*",50)  + nl
for x in aResult
        see x["name"] + nl
next
sqlite_close(oSQLite)
```

13.10 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.

13.11 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
```

13.12 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.

13.13 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

FOURTEEN

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.

14.1 Building using Microsoft Windows

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Build Ring (Compiler/VM)

```
cd ring/src
buildvc.bat
buildvcw.bat
```

Build Ring2EXE

```
cd ../ring2exe
buildring2exe.bat
```

Build RingODBC

```
cd ../extensions/ringodbc
buildvc.bat
```

Build RingMySQL

```
cd ../extensions/ringmysql
buildvc.bat
```

Build RingSQLite

```
cd ../extensions/ringsqlite
buildvc.bat
```

Build RingPostgreSQL

```
cd ../extensions/ringpostgresql
gencode.bat
buildvc.bat
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl
buildvc.bat
```

Build RingInternet

```
cd ../extensions/ringinternet
buildvc.bat
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
buildvc.bat
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
gencode.bat
buildvc.bat
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
gencode.bat
buildvc.bat
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
gencode.bat
buildvc.bat
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
gencode.bat
buildvc.bat
```

Generate RingLibuv Source Code and Build

```
cd ../extensions/ringlibuv
gencode.bat
buildvc.bat
```

Generate RingFreeGLUT Source Code and Build

```
cd ../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 ../extensions/ringopengl/opengl21
gencode.bat
buildvc.bat
```

Generate RingPostgreSQL Source Code and Build

```
cd ../extensions/ringpostgresql
gencode.bat
buildvc.bat
```

Install Qt 5.5 : https://download.qt.io/archive/qt/5.5/5.5.1/

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
gencode.bat
buildmingw32.bat
```

To be able to call ring from any folder

```
cd ../../bin
install.bat
```

Add Ring/bin to System path

```
Hit "windows key".
Type "Edit the System environment variables"
Select "Advanced" tab.
Click on "Enviroment Variables..."
Double click on "Path"
Add at the end the new path separated by semicolon.
    ;C:\Ring\Bin
```

Run Ring Notepad

```
cd applications/rnote
ring rnote.ring
```

14.2 Building using Ubuntu Linux

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Install Libraries

```
cd ring/src
./installdep.sh
```

Build Ring (Compiler/VM)

```
sudo ./buildgcc.sh
```

Build Ring2EXE

```
cd ../ring2exe
sudo ./buildring2exe.sh
```

Build RingODBC

```
cd ../extensions/ringodbc
./buildgcc.sh
```

Build RingMySQL

```
cd ../extensions/ringmysql
./buildgcc.sh
```

Build RingSQLite

```
cd ../extensions/ringsqlite
./buildgcc.sh
```

Build RingPostgreSQL

```
cd ../extensions/ringpostgresql
gencode.sh
buildgcc.sh
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl
./buildgcc.sh
```

Build RingInternet

```
cd ../extensions/ringinternet
./buildgcc.sh
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
./buildgcc.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
./gencode.sh
./buildgcc.sh
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
./gencode.sh
./buildgcc.sh
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
./gencode.sh
./buildgcc.sh
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
./gencode.sh
./buildgcc.sh
```

Generate RingLibuv Source Code and Build

We will build Libuv first

```
cd ../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 ..
./gencode.sh
./buildgcc.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ../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 ../extensions/ringopengl/opengl21
gencode.sh
buildgcc.sh
```

Generate RingPostgreSQL Source Code and Build

```
cd ../extensions/ringpostgresql
./gencode.sh
./buildgcc.sh
```

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
./gencode.sh
./buildgcc.sh
```

To be able to call ring from any folder

```
cd ../../bin
sudo ./install.sh
```

Run Ring Notepad

```
cd applications/rnote
ring rnote.ring
```

14.3 Building using Fedora Linux

Get the source code

```
git clone http://github.com/ring-lang/ring.git
```

Install Libraries

```
cd ring/src
./installdepfedora.sh
```

Build Ring (Compiler/VM)

```
sudo ./buildgcc.sh
```

Build Ring2EXE

```
cd ../ring2exe
sudo ./buildring2exe.sh
```

Build RingODBC

```
cd ../extensions/ringodbc
./buildgcc.sh
```

Build RingMySQL

```
cd ../extensions/ringmysql
./buildgccfedora.sh
```

Build RingSQLite

```
cd ../extensions/ringsqlite
./buildgcc.sh
```

Build RingPostgreSQL

```
cd ../extensions/ringpostgresql
gencode.sh
buildgcc.sh
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl
./buildgcc.sh
```

Build RingInternet

```
cd ../extensions/ringinternet
./buildgcc.sh
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
./buildgcc.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
./gencode.sh
./buildgcc.sh
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
./gencode.sh
./buildgcc.sh
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
./gencode.sh
./buildgcc.sh
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
./gencode.sh
./buildgcc.sh
```

Generate RingLibuv Source Code and Build

We will build Libuv first

```
cd ../extensions/ringlibuv/libuv
sudo dnf install libtool m4 autoconf automake
sh autogen.sh
./configure
make
make check
sudo make install
```

Then we will build RingLibuv

```
cd ..
./gencode.sh
./buildgcc.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ../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 ../extensions/ringopengl/opengl21
gencode.sh
buildgcc.sh
```

Generate RingPostgreSQL Source Code and Build

```
cd ../extensions/ringpostgresql
./gencode.sh
./buildgcc.sh
```

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
./gencode.sh
./buildgccfedora.sh
```

To be able to call ring from any folder

```
cd ../../bin
sudo ./install.sh
```

Run Ring Notepad

```
cd applications/rnote
ring rnote.ring
```

14.4 Building using MacOS X

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/src
./installdepmac.sh
```

Build Ring (Compiler/VM)

```
./buildclang.sh
```

Build Ring2EXE

```
cd ../ring2exe
sudo ./buildring2exe.sh
```

Build RingODBC

```
cd ../extensions/ringodbc
./buildclang.sh
```

Build RingMySQL

```
cd ../extensions/ringmysql
./buildclang.sh
```

Build RingSQLite

```
cd ../extensions/ringsqlite
./buildclang.sh
```

Build RingPostgreSQL

```
cd ../extensions/ringpostgresql
gencode.sh
buildclang.sh
```

Build RingOpenSSL

```
cd ../extensions/ringopenssl
./buildclang.sh
```

Build RingInternet

```
cd ../extensions/ringinternet
./buildclang.sh
```

Build RingMurmurHash

```
cd ../extensions/ringmurmurhash
./buildclang.sh
```

Generate RingConsoleColors Source Code and Build

```
cd ../extensions/ringconsolecolors
./gencode.sh
./buildclang.sh
```

Generate RingAllegro Source Code and Build

```
cd ../extensions/ringallegro
./gencode.sh
./buildclang.sh
```

Generate RingLibCurl Source Code and Build

```
cd ../extensions/ringcurl
./gencode.sh
./buildclang.sh
```

Generate RingZip Source Code and Build

```
cd ../extensions/ringzip
./gencode.sh
./buildclang.sh
```

Generate RingLibuv Source Code and Build

```
cd ../extensions/ringlibuv
./gencode.sh
./buildclang.sh
```

Generate RingFreeGLUT Source Code and Build

```
cd ../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 ../extensions/ringopengl/opengl21
./gencode.sh
./buildclang.sh
```

Generate RingPostgreSQL Source Code and Build

```
cd ../extensions/ringpostgresql
./gencode.sh
./buildclang.sh
```

Generate RingQt Source Code and Build

```
cd ../extensions/ringqt
./gencodemac.sh
./buildclang.sh
```

To be able to call ring from any folder

```
cd ../../bin
sudo ./install.sh
```

Run Ring Notepad

```
cd applications/rnote
sudo ring rnote.ring
```

14.5 Building using CMake

Install libraries (MySQL Client, OpenSSL, LibCurl, Allegro 5 and Qt 5.5)

```
cmake .
make
```

CHAPTER

FIFTEEN

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

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

15.1 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: http://ring-lang.sf.net/team.html

15.2 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/docs/source

The documentation is created using Sphinx : http://www.sphinx-doc.org/en/stable/

15.3 Testing

You can write new tests in this folder

https://github.com/ring-lang/ring/tree/master/tests/scripts

15.4 Samples

You can add new samples to this folder

https://github.com/ring-lang/ring/tree/master/samples/other

15.5 Applications

You can add new applications to this folder

https://github.com/ring-lang/ring/tree/master/applications

15.6 Editors Support

You can help in supporting Ring in different code editors

Check the next folder

https://github.com/ring-lang/ring/tree/master/editor

15.7 Libraries in Ring

You can update and add libraries to this folder

https://github.com/ring-lang/ring/tree/master/ringlibs

15.8 Extensions in C/C++

You can add and update extensions in this folder

https://github.com/ring-lang/ring/tree/master/extensions

15.9 Compiler and Virtual Machine (VM)

- Source Code (C Language) : https://github.com/ring-lang/ring/tree/master/src
- Visual Source (PWCT) : https://github.com/ring-lang/ring/tree/master/visualsrc

15.10 Ideas and suggestions

You can share your ideas, suggestions and questions in this group

https://groups.google.com/forum/#!forum/ring-lang

CHAPTER

SIXTEEN

GETTING STARTED - FIRST STYLE

16.1 Hello World

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

```
see "Hello World"
```

16.2 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
```

16.3 Create Executable File

Using Ring2EXE we can create executable file for our application

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

16.4 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"
```

16.5 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?"
```

16.6 Getting Input

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

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

16.7 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
```

16.8 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
```

16.9 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

SEVENTEEN

GETTING STARTED - SECOND STYLE

17.1 Hello World

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

```
put "Hello World"
```

17.2 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
```

17.3 Create Executable File

Using Ring2EXE we can create executable file for our application

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

17.4 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"
```

17.5 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?"
```

17.6 Getting Input

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

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

17.7 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
```

17.8 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
*/

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

EIGHTEEN

GETTING STARTED - THIRD STYLE

18.1 Hello World

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

```
load "stdlib.ring"

print("Hello World")
```

18.2 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
```

18.3 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

But since the stdlib.ring load libraries like (LibCurl, OpenSSL, MySQL, etc)

You will need these libraries!

To avoid the need to these libraries (If you don't need stdlib classes)

Use stdlibcore.ring instead of stdlib.ring as in the next example

```
load "stdlibcore.ring"

print("Hello World")
```

Using stdlibcore.ring You can access the stdlib functions but not the stdlib classes.

if you want to use stdlib.ring and distribute your application

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

18.4 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

```
LOAD "stdlib.ring"
PRINT("Hello World")
```

```
Load "stdlib.ring"
Print("Hello World")
```

18.5 Multi-Line literals

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

```
Load "stdlib.ring"
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.

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

18.6 Getting Input

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

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

18.7 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.

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

18.8 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
*/

Load "stdlib.ring"

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.

CHAPTER

NINETEEN

USING RING NOTEPAD

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

Ring Notepad is just a simple application developed using the Ring language.

19.1 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

19.2 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.

19.3 Creating and running your first GUI/Mobile Application

To learn how to create GUI applications using Ring check the "Desktop 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

19.4 Creating and running your first Web Application

To learn how support Ring in your web server and how to create web applications using Ring 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.

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.

For Windows users, Ring comes with Apache Web server!

We can run any web application from any folder directly without doing any configuration.

19.5 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
                        Scaled=true
                }
        }
```

We can run the application as any GUI application.

19.6 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:

19.7 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"

19.8 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.

19.9 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

19.10 The Program Menu

From this menu we can run the programs

Also we can set the Main file in the project

19.11 The Browser Menu

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

19.12 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

19.13 The Distribute Menu

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

Also we can prepare the application for distribution

19.14 The Help Menu

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

CHAPTER

TWENTY

USING OTHER CODE EDITORS

In the Ring/Editor folder you will find extensions for the next editors

- Notepad++
- Geany
- Atom
- Sublime Text 2
- Visual Studio IDE
- Emacs
- Visual Studio Code (VSCode)

20.1 Using Notepad++

Folder : ring/editor/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++

20.2 Using Geany

Folder : ring/editor/geany

- Run Geany editor
- Click on “Tools -> configuration files -> filetypes_extensions.conf” menu
- Add this line “Ring=*.ring;” without quotes after [Extensions]
- In unbuntu copy file “filetypes.Ring.conf” to folder “/home/USERNAME/filetypes.Ring.conf”
- You can run your files by pressing F5 button

20.3 Using Atom

Folder : ring/editor/atom

Just Copy the folder atom-language-ring to the next path

```
"C:\Users\{UserName}\.atom\Packages"
```

20.4 Using Sublime Text 2

Folder : ring/editor/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

20.5 Using Visual Studio IDE

Folder : ring/editor/visualstudio

Check the ReadMe file for installation instructions.

20.6 Using Emacs Editor

Folder : ring/editor/emacs

Check the ReadMe file for installation instructions.

Screen Shot:

20.7 Visual Studio Code

Folder : ring/vscode

Check the ReadMe file for installation instructions.

Screen Shot:

File Edit Selection View Go Debug Terminal Help
findinfilesController.ring - ring - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
cards.ring applications\c...
findinfilesController.ring ...
RING
Image
target
Emp.ring
README.md
fifteenpuzzle
fifteenpuzzle3d
findinfiles
images
test
findinfiles.rform
findinfilescontrolle... U
findinfilesController.ring
findinfilesView.ring
README.md
flappybird3000
formdesigner
game2048
getquoteshistory
goldmagic800
knighttour
memorygame
minesweeper
OUTLINE
cards.ring
findinfilesController.ring
Application : Find in files
Author : Mahmoud Fayed <msfclipper@yahoo.com>
Date : 2018.03.06
load "findinfilesView.ring"
import System.GUI
if isMainSourceFile() {
new App
{
StyleFusion()
open_window(:findinfilesController)
exec()
}
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: cmd
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.
B:\ring>cd "b:\ring\applications\findinfiles"
b:\ring\applications\findinfiles>
master*
Ln 4, Col 1 Tab Size: 4 UTF-8 CRLF ring

CHAPTER

TWENTYONE

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).

21.1 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
```

21.2 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
```

21.3 Weakly Typed

Ring is a weakly typed language, this means that the language can automatically convert between data types (like string & numbers) when that conversion make sense.

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

TWENTYTWO

OPERATORS

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

22.1 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

22.2 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

22.3 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

22.4 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

22.5 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

22.6 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

22.7 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

TWENTYTHREE

CONTROL STRUCTURES - FIRST STYLE

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

23.1 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
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
```

23.2 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
                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
```

23.3 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
```

23.4 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
```

23.5 Do Again Loop

Syntax:

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

Example:

```
x = 1
do
        see x + nl
        x++
again x <= 10
```

23.6 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
```

23.7 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
```

23.8 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
        ok
        see x + nl
next
```

23.9 Exit/Loop inside sub functions

While we are inside a loop, we can call a function then use the exit and/or loop command inside that function and the command will work on the outer loop.

Example:

```
# print numbers from 1 to 10 except number 5.

for x = 1 to 10
        ignore(x,5)
        see x + nl
next

func ignore x,y
        if x = y
                loop
        ok
```

23.10 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())
        see "great" + nl
ok

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

23.11 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).

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

TWENTYFOUR

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.

24.1 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
        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
```

24.2 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
        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
```

24.3 Exceptions

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

CHAPTER

TWENTYFIVE

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.

25.1 Branching

- If Statement

Syntax:

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

Example:

```
Load "stdlib.ring"

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:

```
Load "stdlib.ring"

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")
}
```

25.2 Looping

- While Loop

Syntax:

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

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")
        }

}
```

- For Loop

Syntax:

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

Example:

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

Example:

```
load "stdlib.ring"

# 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:

```
load "stdlib.ring"

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

Example:

```
load "stdlib.ring"

# 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:

```
load "stdlib.ring"

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:

```
load "stdlib.ring"

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:

```
load "stdlib.ring"

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
```

25.3 Exceptions

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

CHAPTER

TWENTYSIX

GETTING INPUT

We can get input from the keyboard using

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

26.1 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
```

26.2 GetChar() Function

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

Syntax:

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

Example:

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

26.3 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

TWENTYSEVEN

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

27.1 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
```

27.2 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
```

27.3 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
```

27.4 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
```

27.5 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
```

27.6 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.

27.7 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
```

27.8 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

TWENTYEIGHT

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

28.1 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
```

28.2 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
```

28.3 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
```

28.4 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)

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

28.5 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
```

28.6 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.

28.7 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
```

28.8 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

TWENTYNINE

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

29.1 Define Functions

To define new function

Syntax:

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

Example:

```
load "stdlib.ring"
func hello {
        print("Hello from function \n")
}
```

29.2 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:

```
load "stdlib.ring"

hello()

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

Example:

```
load "stdlib.ring"

first()  second()

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

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

29.3 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:

```
load "stdlib.ring"

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

29.4 Send Parameters

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

Syntax:

```
funcname(parameters)
```

Example:

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

load "stdlib.ring"

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

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

29.5 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

load "stdlib.ring"

print("Hello, World! \n")

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

29.6 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

load "stdlib.ring"

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.

29.7 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:

```
load "stdlib.ring"

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

func novalue { }
```

29.8 Recursion

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

Example:

```
load "stdlib.ring"

print( fact(5) )         # output = 120

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

CHAPTER

THIRTY

PROGRAM STRUCTURE

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

30.1 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.

30.2 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
```

30.3 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
```

CHAPTER

THIRTYONE

LISTS

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

31.1 Create Lists

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

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)
```

Example

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

Note: the list index start from 1

31.2 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
```

31.3 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
```

31.4 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
```

31.5 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
```

31.6 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
```

31.7 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
```

31.8 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 ] ]

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

Output:

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

31.9 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
```

31.10 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
```

31.11 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
```

31.12 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
```

31.13 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

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]
```

31.14 Using Lists during definition

We can use 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
```

31.15 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]
```

31.16 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
```

31.17 Passing Parameters or Argumnents 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]
```

31.18 Passing Parameters or Argumnents Using List Array

Passing Arguments or Parmameters 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
```

31.19 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 dimesions: 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)

     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
```

31.20 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

THIRTYTWO

STRINGS

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

32.1 String Literals

Syntax:

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

32.2 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
```

32.3 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)
```

32.4 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
```

32.5 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
```

32.6 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!
```

32.7 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
```

32.8 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***
```

32.9 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
```

32.10 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

32.11 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
```

32.12 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
```

32.13 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
```

32.14 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
```

32.15 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
```

32.16 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
```

32.17 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
All String
12
1
1
```

CHAPTER

THIRTYTHREE

DATE AND TIME

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

33.1 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
```

33.2 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
```

33.3 Time() Function

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

Example:

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

33.4 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
```

33.5 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
```

33.6 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
```

33.7 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
```

33.8 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
#-------------------------------------------------------------

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

THIRTYFOUR

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

34.1 Check Data Type

The next functions can be used to check the data type

- isstring()
- isnumber()
- islist()
- type()
- isnull()

34.2 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
```

34.3 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
```

34.4 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
```

34.5 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
```

34.6 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
```

34.7 Check Character

The next functions can be used to check character

- isalnum()
- isalpha()
- iscntrl()
- isdigit()
- isgraph()
- islower()
- isprint()
- ispunct()
- isspace()
- isupper()
- isxdigit()

34.8 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
```

34.9 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
```

34.10 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
```

34.11 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
```

34.12 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
```

34.13 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
```

34.14 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
```

34.15 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
```

34.16 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
```

34.17 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
```

34.18 IsXdigit() Function

We can test a character or a string using the IsXdigit() Function

Syntax:

```
IsXdigit(value) ---> 1 if the value is a hexdecimal digit character or 0 if not
```

Example:

```
see isxdigit("0123456789abcdef") + nl +  # print 1
    isxdigit("123z")                     # print 0
```

34.19 Conversion

The next functions can be used for conversion

- number()
- string()
- ascii()
- char()
- hex()
- dec()
- str2hex()
- hex2str()

34.20 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
```

34.21 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
```

34.22 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
```

34.23 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
```

34.24 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
```

34.25 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
```

34.26 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
```

34.27 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

THIRTYFIVE

MATHEMATICAL FUNCTIONS

In this chapter we are going to learn about the mathematical functions

35.1 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
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

35.2 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
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
```

35.3 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
```

35.4 SRandom() Function

The SRandom() function initialize random number generator.

Syntax:

```
SRandom(x)
```

35.5 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
```

35.6 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
1.12346
1.123457
1.1234568
1.12345679
1.123456789
1.1234567890
1.12345678901
1.123456789012
1.1234567890123
1.12345678901230
```

35.7 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
```

35.8 Using f after numbers

We can use the 'f' letter after numbers.

Example:

```
x = 19.99f
see type(x) + nl
```

Output:

```
NUMBER
```

CHAPTER

THIRTYSIX

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()
- Numbers and Bytes

36.1 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")
```

36.2 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)
```

36.3 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)
```

36.4 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")
```

36.5 Remove() Function

We can delete a file using the Remove() function

Syntax:

```
Remove(cFileName)
```

Example:

```
remove("test.txt")
```

36.6 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

36.7 Fclose() Function

When we open a file using fopen() function, we can close it using the Fclose() function

Syntax:

```
Fclose(file handle)
```

36.8 Fflush() Function

We can flushes the output buffer of a stream using the Fflush() function

Syntax:

```
Fflush(file handle)
```

36.9 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

/*
** 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
```

36.10 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
```

36.11 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
```

36.12 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

36.13 Ftell() Function

We can know the current file position of a stream using the Ftell() function

Syntax:

```
Ftell(file handle) ---> file position as number
```

36.14 Rewind() Function

We can set the file position to the beginning of the file using the Rewind() function

Syntax:

```
Rewind(file handle)
```

36.15 Fgetpos() Function

We can get handle to the current file position using the Fgetpos() function

Syntax:

```
Fgetpos(file handle) ---> position handle
```

36.16 Fsetpos() Function

We can set the current file position using the Fgetpos() function

Syntax:

```
Fsetpos(file handle,position handle)
```

36.17 Clearerr() Function

We can clear the EOF error and the error indicators of a stream using the clearerr() function

Syntax:

```
Clearerr(file handle)
```

36.18 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
```

36.19 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
```

36.20 Perror() Function

We can print error message to the stderr using the Perror() function

Syntax:

```
Perror(cErrorMessage)
```

36.21 Fgetc() Function

We can get the next character from the stream using the Fgetc() function

Syntax:

```
Fgetc(file handle) ---> returns character or EOF
```

36.22 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.

36.23 Fputc() Function

We can write a character to the stream using the Fputc() function

Syntax:

```
Fputc(file handle,cChar)
```

36.24 Fputs() Function

We can write a string to the stream using the Fputs() function

Syntax:

```
Fputs(file handle,cString)
```

36.25 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)
```

36.26 Fread() Function

We can read data from a stream using the Fread() function

Syntax:

```
Fread(file handle,nSize)
```

36.27 Fwrite() Function

We can write data to a stream using the Fwrite() function

Syntax:

```
Fwrite(file handle,cString)
```

36.28 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
```

36.29 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")
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)
```

36.30 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

THIRTYSEVEN

SYSTEM FUNCTIONS

In this chapter we are going to learn about the system functions

- System()
- SysGet()
- IsMSDOS()
- IsWindows()
- IsWindows64()
- IsUnix()
- IsMacOSX()
- IsLinux()
- IsFreeBSD()
- IsAndroid()
- Windowsnl()
- Get Command Line Arguments
- Get Active Source File Name
- CurrentDir()
- ExeFileName()
- ChDir()
- ExeFolder()
- Version()
- Shutdown()

37.1 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
```

37.2 SysGet() Function

We can get environment variables using the Get() function

Syntax:

```
SysGet(cVariable)
```

Example:

```
see sysget("path")               # print system path information
```

37.3 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
```

37.4 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
```

37.5 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
```

37.6 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
```

37.7 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
```

37.8 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
```

37.9 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
```

37.10 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
```

37.11 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
```

37.12 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("ouput.txt",cStr)
```

37.13 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
```

37.14 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
```

37.15 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
```

37.16 CurrentDir() Function

Return the path of the current directory

Syntax:

```
CurrenDir() ---> String contains the path of the currect directory
```

37.17 ExeFileName() Function

Return the Ring executable file name

Syntax:

```
exefilename() ---> String contains the Ring executable file name
```

37.18 ChDir() Function

Change the current directory

Syntax:

```
ChDir(cNewPath)
```

37.19 ExeFolder() Function

Return the Ring executable file path

Syntax:

```
exefolder() ---> String contains the Ring executable path
```

37.20 Version() Function

Return the Ring version

Syntax:

```
version() ---> String contains the Ring version
```

Output:

```
1.10
```

37.21 Shutdown() Function

Close the application

Syntax:

```
shutdown(nStatus) ---> Close the application
```

CHAPTER

THIRTYEIGHT

EVAL() AND DEBUGGING

In this chapter we are going to learn about

- Error Handling using Try/Catch/Done
- Eval() function
- Raise() function
- Assert() function

38.1 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) : Cann't divide by zero !
```

38.2 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
```

38.3 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
```

38.4 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

THIRTYNINE

DEMO PROGRAMS

In this chapter we will see simple demo programs

- Language Shell
- Main Menu

39.1 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

code:> bye
```

39.2 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
        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

        *
        **
        ***
        ****
        *****
        ******
        *******
        ********
        *********
        **********


        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

FORTY

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
```

40.1 odbc_init() Function

We can create ODBC Handle using the odbc_init() function

Syntax:

```
odbc_init() ---> ODBC Handle
```

40.2 odbc_drivers() Function

We can get a list of ODBC drivers using the odbc_drivers() function

Syntax:

```
odbc_drivers(ODBC Handle) ---> List of Drivers
```

40.3 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
```

40.4 odbc_close() Function

After the end of using ODBC functions we can free resources using ODBC_Close() function

Syntax:

```
odbc_close(ODBC Handle)
```

40.5 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
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
```

40.6 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)
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
```

40.7 odbc_connect() Function

We can connect to the database using the odbc_connect() function.

Syntax:

```
odbc_connect(ODBC Handle, cConnectionString)
```

40.8 odbc_disconnect() Function

We can close the connection to the database using the odbc_disconnect() function.

Syntax:

```
odbc_disconnect(ODBC Handle)
```

40.9 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...
```

40.10 odbc_execute() Function

We can execute SQL Statements on the database using the odbc_execute() function.

Syntax:

```
odbc_execute(ODBC Handle, cSQLStatement)
```

40.11 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
```

40.12 odbc_fetch() Function

We can fetch a row from the query result using the odbc_fetch() function.

Syntax:

```
odbc_fetch(ODBC Handle)
```

40.13 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
```

40.14 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
                see odbc_getdata(pODBC,x) + " - "
        next
end
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

40.15 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...
```

40.16 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...
```

40.17 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
```

40.18 odbc_commit() Function

We can commit updates to the database using the odbc_commit() function.

Syntax:

```
odbc_commit(ODBC Handle)
```

40.19 odbc_rollback() Function

We can rollback updates to the database using the odbc_rollback() function.

Syntax:

```
odbc_rollback(ODBC Handle)
```

40.20 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
1
insert data...
Close database...
```

40.21 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

FORTYONE

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
```

41.1 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
```

41.2 MySQL_Init() Function

We can start using MySQL Client through the MySQL_Init() function.

Syntax:

```
MySQL_Init() ---> MySQL Handle
```

41.3 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
```

41.4 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
```

41.5 MySQL_Close() Function

We can close the connection to the MySQL database using the MySQL_Close() function

Syntax:

```
MySQL_Close(MySQL Handle)
```

41.6 MySQL_Query() Function

We can execute SQL queries using the MySQL_Query() function

Syntax:

```
MySQL_Query(MySQL Handle, cSQLQuery)
```

41.7 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 "Cann'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
```

41.8 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

        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
```

41.9 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')")
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
```

41.10 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
```

41.11 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)
```

41.12 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
```

41.13 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
11
3
32768
Name
65535
252
16
Salary
11
3
32768
Close database
```

41.14 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
17000
Close database
```

41.15 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
```

41.16 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...
```

41.17 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)
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...
```

41.18 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
```

41.19 MySQL_Commit() Function

We can commit updates to the database using the MySQL_Commit() function.

Syntax:

```
MySQL_Commit(MySQL Handle)
```

41.20 MySQL_Rollback() Function

We can rollback updates to the database using the MySQL_Rollback() function.

Syntax:

```
MySQL_Rollback(MySQL Handle)
```

41.21 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
        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

FORTYTWO

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
```

42.1 sqlite_init() function

Syntax:

```
sqlite_init() ---> SQLite Object
```

42.2 sqlite_open() function

Syntax:

```
sqlite_open(SQLite Object,cFileName)
```

42.3 sqlite_execute() function

Syntax:

```
sqlite_execute(SQLite Object,cSQLStatement)
```

42.4 sqlite_close() function

Syntax:

```
sqlite_close(SQLite Object)
```

42.5 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
15000.0
3
Mohammed
31
Egypt
```

```
20000.0
4
Ibrahim
24
Egypt
65000.0
************************************************
Mahmoud
Ahmed
Mohammed
Ibrahim
```

CHAPTER

FORTYTHREE

POSTGRESQL FUNCTIONS

In this chapter we will learn about using the PostgreSQL database in the Ring programming language.

43.1 Loading the library

Before using the next function load the postgresqllib.ring library

```
load "postgresqllib.ring"
# Use PostgreSQL functions
```

43.2 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)
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 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)

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
4 Ibrahim  24 Egypt   40
```

43.3 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
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
```

43.4 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)
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)
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)
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

FORTYFOUR

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 function load the openssllib.ring library

```
load "openssllib.ring"
# Use OpenSSL functions
```

- MD5()
- SHA1()
- SHA256()
- SHA512()
- SHA384()
- SHA224()
- Encrypt()
- Decrypt()
- Randbytes()

Before using the next function load the internetlib.ring library

```
load "internetlib.ring"
# Use the Internet functions
```

- Download()
- SendEmail()

44.1 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
```

44.2 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
```

44.3 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
```

44.4 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
```

44.5 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
```

44.6 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
```

44.7 Encrypt() Function

We can use the Encrypt() function to encrypts the data using the Blowfish algorithm.

Syntax:

```
Encrypt(cString, cKey, cIV) ---> Encrypted string
```

44.8 Decrypt() Function

We can use the Decrypt() function to decrypt the data encrypted using the Encrypt() function.

Syntax:

```
Decrypt(cCipher, cKey, cIV) ---> Decrypted string
```

44.9 Encryption and Decryption Example

The next example demonstrates how to use the Encrypt() and Decrypt() functions.

These functions use the Blowfish algorithm.

```
See "Enter a string : " give cStr
list = 0:15  cKey=""    for x in list cKey += char(x) next
list = 1:8   cIV = ""   for x in list cIV += char(x) next
cStr = Encrypt(cStr,cKey,cIV)
See "Cipher Text    : " + cStr + nl +
    "Plain Text     : " + Decrypt(cStr,cKey,cIV) + 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)
next

cIV = ""
for x = 1 to 8
        cIV += char(x)
next


cStr = Encrypt(cStr,cKey,cIV)
See "Cipher Text    : " + cStr + nl +
    "Plain Text     : " + Decrypt(cStr,cKey,cIV) + nl
```

Also we can write the password and the IV directly using strings

```
See "Enter a string : " give cStr

# Note: Don't use simple password in real applications!
cKey = "1234567890@#$%^&"
cIV  = "87654321"

cStr = Encrypt(cStr,cKey,cIV)
See "Cipher Text    : " + cStr + nl +
    "Plain Text     : " + Decrypt(cStr,cKey,cIV) + nl
```

44.10 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
```

44.11 Randbytes() Function

We can generate a string of 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
```

44.12 Download() Function

Syntax:

```
Download(cURL) ---> String contains the server response
```

Example:

```
cStr= download("http://doublesvsoop.sourceforge.net/")
see cStr
write("download.txt",cStr)
```

44.13 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
         How are you?
         Are you fine?
         Thank you!
         Greetings,
         Mahmoud")
see "Done.." + nl
```

CHAPTER

FORTYFIVE

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()

45.1 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.

45.2 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
        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
```

45.3 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
                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)
```

45.4 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 }

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
```

45.5 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
                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
```

45.6 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 paramters, 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
                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
        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"
            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
```

45.7 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
```

```
                super.print()
                see "Job   : " + job + nl + "Salary : " + salary + nl
```

Output:

```
Name : test
Age  : 20
Job  : programmer
Salary : 20000000
```

45.8 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

45.9 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
```

45.10 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
```

45.11 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
```

45.12 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"},
                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
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
```

45.13 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

Whay 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

45.14 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
```

45.15 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.

45.16 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
```

CHAPTER

FORTYSIX

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

46.1 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
3
4
5
```

```
aList2
1
4
9
16
25
```

46.2 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!
```

46.3 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!
```

46.4 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
                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
```

46.5 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

FORTYSEVEN

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()

47.1 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
```

47.2 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
```

47.3 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
```

47.4 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 : 208
len()
add()
del()
get()
```

```
clock()
...
```

Note: The complete list is removed from the previous output.

47.5 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
```

47.6 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
```

47.7 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
```

47.8 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
```

47.9 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
```

47.10 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
```

47.11 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
```

47.12 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
```

47.13 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
```

47.14 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
```

47.15 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
```

47.16 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
```

47.17 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
```

47.18 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
```

47.19 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
```

47.20 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
```

47.21 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
```

47.22 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
        func print
                see x + nl + y + nl + z + nl
```

47.23 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
```

47.24 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")
addattribute(o1,"z")
see o1 {x=10 y=20 z=30}
class point
```

Output:

```
x: 10.000000
y: 20.000000
z: 30.000000
```

47.25 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()

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
```

47.26 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
```

```
        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
```

47.27 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
        nSalary
        aColors
```

Output:

```
cname: Mahmoud
nsalary: 1000000.000000
```

```
acolors: List...
white
blue
yellow
```

47.28 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
        Func test
                three()
                two()
                one()
                printline()
```

Output:

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

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

47.29 packagename() Function

We can know the package name of the latest sucessful import command using the packagename() function

Syntax:

```
packagename() --> Returns the package name of the latest sucessful import
```

Example:

```
load "weblib.ring"
import System.web
see packagename()        # system.web
```

CHAPTER

FORTYEIGHT

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.

48.1 Puts() function

print the value then print new line (nl)

Syntax:

```
puts(expr)
```

Example:

```
Load "stdlib.ring"

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

48.2 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")
```

48.3 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
```

48.4 GetString() function

Get input from the keyboard - return value as string

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

48.5 GetNumber() function

Get input from the keyboard - return value as number

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

48.6 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
```

48.7 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\
```

48.8 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
```

48.9 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
```

48.10 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 } )
```

48.11 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 } )
```

48.12 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 } )
```

48.13 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"," ")
```

48.14 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"," ,")
```

48.15 NewList() function

Create a two dimensional list

Syntax:

```
NewList(nRows,nColumns) ---> new list
```

Example:

```
Load "stdlib.ring"

Puts("Test Newlist()")
a1 = 3
a2 = 5
chrArray = newlist(a1,a2)
numArray = newlist(a1,a2)
chrArray[1][1] = "Hello"
numArray[1][1]  = 987.2
See chrArray[1][1] + nl
See numArray[1][1] + nl
```

48.16 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")
```

48.17 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
```

48.18 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
```

48.19 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")
```

48.20 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)
```

48.21 Fibonacci() function

Return the fibonacci number

Syntax:

```
Fibonacci(number) ---> number
```

Example:

```
Load "stdlib.ring"

Puts("Test Fibonacci()")
see "6 Fibonacci is : " + Fibonacci(6)
```

48.22 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
```

48.23 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
```

48.24 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")
```

48.25 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"))
```

48.26 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
```

48.27 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
```

48.28 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
```

48.29 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
```

48.30 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
```

48.31 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
```

48.32 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
```

48.33 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
```

48.34 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()")
cString = "radar"
see IsPalindrome(cString)
```

48.35 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
```

48.36 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)
```

48.37 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)
```

48.38 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)
```

48.39 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
```

48.40 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
```

48.41 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)
```

48.42 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)
```

48.43 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
```

48.44 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
```

48.45 IsMainSourceFile() function

Check if the current file is the main source file

Syntax:

```
IsMainSourceFile() ---> True/False
```

Example:

```
Load "stdlib.ring"

if ismainsourcefile()
        # code
ok
```

48.46 DirExists() function

Check if directory exists

Syntax:

```
DirExists(String) ---> True/False
```

Example:

```
Load "stdlib.ring"

see "Check dir : b:\ring "
puts( DirExists("b:\ring") )
see "Check dir : C:\ring "
Puts( DirExists("C:\ring") )
```

48.47 MakeDir() function

Make Directory

Syntax:

```
MakeDir(String)
```

Example:

```
Load "stdlib.ring"

# Create Directory
puts("create Directory : myfolder")
makedir("myfolder")
```

48.48 Fsize() function

The function return the file size in bytes.

Syntax:

```
FSize(File Handle) ---> Number (File Size in Bytes)
```

48.49 TrimAll() function

Remove all spaces and tabs characters from a string

Syntax:

```
TrimAll(cString) ---> cString # Without Spaces and Tabs
```

48.50 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
```

48.51 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
```

48.52 EpochTime() function

Return the Epoch Time

Syntax:

```
EpochTime(cDate,cTime) ---> nEpochTime
```

Example:

```
see EpochTime( Date(), Time() )
```

48.53 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
```

48.54 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/ringlibs/weblib","") # All Files
```

48.55 SystemSilent() Function

We can execute system commands using the SystemSilent() function to avoid displaying the output!

Syntax:

```
SystemSilent(cCommand)
```

48.56 OSCreateOpenFolder() Function

Create folder then change the current folder to this new folder

Syntax:

```
OSCreateOpenFolder(cCommand)
```

48.57 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")
```

48.58 OSDeleteFolder() Function

Delete Folder in the current Directory

Syntax:

```
OSDeleteFolder(cFolderName)
```

48.59 OSCopyFile() Function

Copy File to the current directory

Syntax:

```
OSCopyFile(cFileName)
```

48.60 OSDeleteFile() Function

Delete File

Syntax:

```
OSDeleteFile(cFileName)
```

48.61 OSRenameFile() Function

Rename File

Syntax:

```
OSRenameFile(cOldFileName,cNewFileName)
```

48.62 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
]
```

48.63 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
```

48.64 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
```

CHAPTER

FORTYNINE

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

49.1 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)

49.2 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()
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!
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?
```

49.3 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
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
```

49.4 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)
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
```

49.5 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
3
4
one
two
three
```

49.6 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
Cairo
KSA
Riyadh
************************************************************
Egypt
Cairo
```

49.7 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")
                }
        }
        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
```

49.8 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

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
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
```

49.9 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

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
```

49.10 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()
```

49.11 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
```

49.12 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!")
```

49.13 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 hexdecimal 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
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
```

49.14 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
```

49.15 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()
}
```

49.16 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
3
Fayed
17000
Close database
```

49.17 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
        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
```

49.18 PostgreSQL Class

Methods:

Method	Description/Output
init(cConString)	Open Database.
close()	Close Database.
execute(cSQL)	Execute Query.

example:

```
load "stdlib.ring"

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
Mohammed
31
Egypt
20000
4
Ibrahim
24
Egypt
65000

**************************************************
```

49.19 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.
decrypt(cString,cKey,cIV)	Decrypt the data encrypted using the Encrypt() method.
randbytes(nSize)	Generate a string of 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
}
```

49.20 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

FIFTY

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

50.1 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
```

50.2 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
                }
                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
```

50.3 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!
```

50.4 Declarative Programming on the top of Object-Oriented

The next features enable us to build and use declartive 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:

```
# Declartive 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
                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!
```

50.5 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 direcly 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

FIFTYONE

NATURAL LANGUAGE PROGRAMMING

Using the Ring programming language, we can create Natural programming languages based on classes and objects.

51.1 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.

51.2 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

        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
```

51.3 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
Instruction : Window Title = hello world
```

51.4 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
```

51.5 Change the '=' operator to 'is'

Example:

```
ChangeRingKeyword       and  _and
ChangeRingOperator      =    is

New App
{
                I want window and the window title is "hello world"
}

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
```

51.6 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 operatos).

  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
              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
```

51.7 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
```

51.8 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
```

51.9 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
        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
```

51.10 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...
```

51.11 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
                # 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

FIFTYTWO

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)

52.1 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
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)
        }
])
```

52.2 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
```

52.3 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
                that looks like Lisp Syntax
                (count  (count 1 5)  (count 20 15))
                Just for fun!
        ")
}
```

52.4 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()
                                }
                        }
                }

        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

FIFTYTHREE

WEB DEVELOPMENT (CGI LIBRARY)

In this chapter we will learn about developing Web applications using a CGI Library written in the Ring language.

53.1 Configure the Apache web server

We can use Ring with any web server that support CGI. In this section we will learn about using Ring with the Apache HTTP Server.

You can download Apache from : http://httpd.apache.org/

Or you can get it included with other projects like

XAMPP : https://www.apachefriends.org/download.html

Install then open the file:

```
xampp\apache\conf\httpd.conf
```

search for

```
<Directory />
```

Then after it add

```
Options FollowSymLinks +ExecCGI
```

So we have

```
<Directory />
Options FollowSymLinks +ExecCGI
```

Search for the next line and be 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
```

Run/Start the server

Create your web applications in a directory supported by the web server.

Example:

```
Apache2.2\htdocs\mywebapplicationfolder
```

Example:

```
xampp\htdocs\mywebapplicationfolder
```

Inside the source code file (*.ring), Add this line

```
#!ring -cgi
```

Note: Change the previous line based on the path to ring.exe in your machine

53.2 Ring CGI Hello World Program

The next program is the Hello World program

```
#!ring -cgi

See "content-type: text/html" +nl+nl+
    "Hello World!" + nl
```

53.3 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
{
        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.

53.4 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

53.5 HTTP Get Example

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web
New Page
{
  Title = "Test HTTP Get"
  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: " )
              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()
  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:

53.6 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:

53.7 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
        # 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:

53.8 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()
        divend()
}
```

Screen Shot:

53.9 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:

53.10 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:

53.11 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:

53.12 Hash Functions

The Page User Interface

```
#!ring -cgi
Load "weblib.ring"
Import System.Web

New Page
{
        boxstart()
                text( "Hash Test")
                newline()
        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"]) )
                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:

53.13 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  ] )
                else
                        text("No images!") newline()
                ok
        divend()
}
```

Screen Shot:

53.14 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
        {
                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:

53.15 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

53.16 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:

53.17 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
                {
                  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"}
                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

53.18 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) }
                          }
                }

                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())
```

53.19 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:

53.20 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
                {
                      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-sucess btn-sm"  text("sucess") }
                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-sucess btn-block"  text("sucess") }
                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") }
                  }
                }
                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:

53.21 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

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:

53.22 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%") +
                                 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()
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"  ])
                  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))
                        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()
```

53.23 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)

  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

        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]
        # 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

  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()
      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)

  func DeleteRecord

        oModel.Delete( aPageVars[cRecID] )
        oView.DeleteView()

  func braceend

        oModel.Disconnect()
```

53.24 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.
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.

53.25 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

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.

53.26 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>

Continued on next page

Table 53.2 – continued from previous page

Method	Parameters	Description
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>
RowStart	aPara	HTML <tr>
RowEnd	None	HTML </tr>
CellStart	aPara	HTML <td>
CellEnd	None	HTML </td>
HeaderStart	aPara	HTML <th>
HeaderEnd	None	HTML </th>

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
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
```

53.27 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

53.28 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;"

53.29 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.
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.

53.30 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

FIFTYFOUR

DEPLOYING WEB APPLICATIONS IN THE CLOUD

In this chapter we will learn about deploying Ring Web Applications in the Cloud using Heroku

54.1 Introduction

We created a new project and tutorial to explain how to deploy Ring web applications in the Cloud using Heroku

Demo : http://testring.herokuapp.com/

Project : https://github.com/ring-lang/RingWebAppOnHeroku

Heroku Website : https://www.heroku.com/

54.2 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 projet 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
```

54.3 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!

54.4 Hello World program

file : ringapp/helloworld.ring

To run it : http://testring.herokuapp.com/ringapp/helloworld.ring

```
#!/app/runring.sh -cgi

see "content-type: text/html" +nl+nl
see "Hello, World!" + nl
```

file : ringapp/helloworld2.ring

To run it : http://testring.herokuapp.com/ringapp/helloworld2.ring

```
#!/app/runring.sh -cgi
load "weblib.ring"
import System.Web
new page {
        text("Hello, World!")
}
```

54.5 Application Database

When you depoly 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

54.6 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
```

54.7 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

FIFTYFIVE

USING RINGLIBCURL

In this chapter we will learn about using RingLibCurl

55.1 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)
```

55.2 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)
```

55.3 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
```

55.4 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)
```

55.5 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)

        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

FIFTYSIX

USING RINGZIP

In this chapter we will learn about using RingZip

56.1 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)
```

56.2 Extract Zip File

Example : Extract myfile.zip to myfolder folder.

```
load "ziplib.ring"
zip_extract_allfiles("myfile.zip","myfolder")
```

56.3 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)
```

56.4 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()
```

```
}
```

56.5 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

56.6 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

FIFTYSEVEN

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.

57.1 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)
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]

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
                            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 Keyborad

57.2 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:

57.3 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()
al_rest(10)

al_destroy_allegro_sample_id(sampleid)
```

```
al_destroy_sample(sample)
al_destroy_display(display)

al_exit()
```

57.4 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:

57.5 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)
al_destroy_display(display)
```

Screen Shot:

57.6 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
        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
!!!!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
!!!!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

FIFTYEIGHT

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.

58.1 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()
```

58.2 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)
SDL_Delay(2000)
SDL_DestroyTexture(tex)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

58.3 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)
```

58.4 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)
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()
```

58.5 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()
```

58.6 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)
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, nullpointer(), surface, nullpointer())
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()
```

58.7 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,nullpointer(),nullpointer())
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,nullpointer(),rect)

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,nullpointer(),rect)

SDL_DestroyTexture(tex)
SDL_Destroy_SDL_Rect(rect)

SDL_RenderPresent(ren)
SDL_Delay(2000)
SDL_DestroyRenderer(ren)
SDL_DestroyWindow(win)
SDL_Quit()
```

58.8 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()
```

58.9 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,nullpointer(),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
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, nullpointer(), surface, nullpointer())
        SDL_UpdateWindowSurface(win)
        SDL_FreeSurface(text)
```

58.10 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

FIFTYNINE

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

59.1 First Application using RingLibuv

Example:

```
load "libuv.ring"

func main

        myloop = new_uv_loop_t()
        uv_loop_init(myloop)
        ? "Now quitting"
        uv_run(myloop, UV_RUN_DEFAULT)
```

```
        uv_loop_close(myloop)
        destroy_uv_loop_t(myloop)
```

Output:

```
Now quitting
```

59.2 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...
```

59.3 Server Example

Example:

```
load "libuv.ring"

? "Testing RingLibuv - Server Side"

DEFAULT_PORT    = 13370
DEFAULT_BACKLOG = 1024

addr    = new_sockaddr_in()
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
```

59.4 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()
        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
```

59.5 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)
                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
```

59.6 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))

        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
```

59.7 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!
```

59.8 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

SIXTY

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.

60.1 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)

60.2 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

60.3 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/ringlibs/gameengine/gl_allegro.ring
- gl_libsdl.ring source code : https://github.com/ring-lang/ring/blob/master/ringlibs/gameengine/gl_libsdl.ring

60.4 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/ringlibs/gameengine/gameengine.ring

60.5 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

60.6 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.
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.

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 ot the game objects.

60.7 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 index of the object in objects list.
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.

60.8 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

60.9 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

60.10 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

60.11 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

60.12 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

60.13 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:

60.14 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:

60.15 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:

60.16 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 ======================================
                        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
```

60.17 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
```

60.18 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 {
          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
              }
        }
       }
}
```

60.19 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
                }
        }                                       # Start the Events Loop
```

60.20 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
                        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
```

60.21 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
                                }
                        }
                        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
```

60.22 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]
                                        }
                                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
```

60.23 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
                                                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:

60.24 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],
                [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:

60.25 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
                  }
                ok
          }

        }
  }           # Start the Events Loop
```

Screen Shot:

60.26 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
        {
                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)
                                        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:

60.27 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

          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
                  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
                        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
                        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
                      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
          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
                }
          }
          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)
                        ok
                  ok
                }
          }
        }
  }


class gamestate
  score = 0
  level = 1
  enemies = 1
  value = 100
  playerindex = 2
  gameresult = false
  startplay=false
```

Screen Shot:

60.28 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"
              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
                                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],
                [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"
                                          }
                                        }
                                  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 {
                        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
  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:

60.29 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 {
          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
        }

        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()
          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,
4,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,0,0,0,1,0,0,0,1,0,3,3,3,5,3,3,3,3,0],
                  [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,
4,4,0,0,4,0,0,0,0,0,0,4,2,0,0,4,1,4,1,4,2,4,1,0,2,0,1,0,4,4,4,4,4,4,4,4,0],
                  [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],
                  [0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,0,1,0,0,0,2,0,0,0,0,0,
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],
                  [0,0,2,0,0,2,0,0,2,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,
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],
                  [0,0,1,0,0,1,0,0,1,3,0,0,0,1,0,0,0,3,0,0,0,1,0,0,0,0,0,
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0]
                ]
          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

          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
                      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
                    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)

          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
  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
                        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 {
                        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 {
                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
  DoorKey = false
  playerindex = 4
  value = 1000
  moveplayer = false
```

Screen Shot:

Super Man 2016
ENERGY : 867
SCORE : 2200

CHAPTER

SIXTYONE

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.

61.1 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
```

61.2 Project Folder

Open the project folder : ring/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"
```

61.3 Building the project

Move to the ring/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
```

CHAPTER

SIXTYTWO

USING RINGOPENGL AND RINGFREEGLUT FOR 3D GRAPHICS

In this chapter we will learn about using RingOpenGL

62.1 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

62.2 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"
```

62.3 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"
```

62.4 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

62.5 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)
        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

62.6 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)

        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

62.7 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)

        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)
```

62.8 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()
```

```
                // 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

62.9 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)
        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
                                blue = 0.0
                on GLUT_KEY_F3
                                red = 0.0
                                green = 0.0
                                blue = 1.0
        off
```

Screen Shot

62.10 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)

// 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)


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

        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)

        // 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)
        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
                        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()
```

62.11 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)

        // 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)
                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
                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)

        // OpenGL init
        glEnable(GL_DEPTH_TEST)

        // enter GLUT event processing cycle
        glutMainLoop()
```

62.12 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

// 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)
        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)
                        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


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


// ----------------------------------
//             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

        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)


// -----------------------------------
//              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

62.13 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
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.
        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

        // 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)
                        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

        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


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  )
        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
        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

62.14 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
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
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()

// 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)



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))
                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

        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
// -----------------------------------

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

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)
        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

62.15 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 )
  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
                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:

CHAPTER

SIXTYTHREE

USING RINGOPENGL AND RINGALLEGRO FOR 3D GRAPHICS

In this chapter we will learn about using RingOpenGL and RingAllegro

63.1 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)

        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)
                // 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()
                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)
                                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:

63.2 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

        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)
            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
```

```
                        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)
                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
                                            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:

63.3 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()
        }

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)
                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

    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
                        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
                //horzintal
                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

        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)
                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)
                    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
                        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()
                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
                                                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:

63.4 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

SIXTYFOUR

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

Steam Page (44 Levels) : https://store.steampowered.com/app/939200/Gold_Magic_800/

Ring comes with a Demo Version (21 Levels Only)

You will find the game in ring/applications/goldmagic800 folder

64.1 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.

64.2 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.

64.3 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!

64.4 Screen Shots

We can select the level

The next screen shot for level (1)

The Gold Magic 800 Level Editor

64.5 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

SIXTYFIVE

DESKTOP 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 and Mobile Applications.

65.1 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()")
                }

                btn1 = new qPushButton(win1) {
                        setGeometry(150,200,100,30)
                        settext("Close")
                        setclickevent("pClose()")
                }

                lineedit1 = new qLineEdit(win1) {
                        setGeometry(10,100,350,30)
                }

                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

65.2 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 stoping 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
```

65.3 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!

65.4 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

65.5 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)
                        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()")
                }

                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 )
        }
```

65.6 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)

                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

65.7 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

65.8 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()
}
```

The application during the runtime

65.9 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 {
                                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")
                                        }
                                        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

65.10 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())
        }
```

65.11 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

65.12 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

65.13 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)
                }

                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

65.14 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

65.15 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()

        }
```

```
        exec()
}
```

The application during the runtime

65.16 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

65.17 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

65.18 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

65.19 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

65.20 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()")
                        }
                        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())
```

65.21 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

65.22 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
```

65.23 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

65.24 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

65.25 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")
                }

                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

65.26 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

65.27 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)
                        setGeometry(0,0,image.width(),image.height())
                }
```

```
                showMaximized()
        }
        exec()
}
```

The application during the runtime

65.28 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,
                stop: 0 #eef, stop: 1 #ccf);")
                showmaximized()
        }
        exec()
}
```

The application during the runtime

65.29 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()")
                }

                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

65.30 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

65.31 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

65.32 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
        if BarMan.value() >= 100        ### ProgressBar start over.
            BarMan.setvalue(0)
        ok
        BarMan{ setvalue(value() + Increment) }
```

```
Func theTime
        return "Time : " + Time()
```

65.33 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()

                        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()
}


###-------------------------------------------------------
### Fuction 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)
```

65.34 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

65.35 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

65.36 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()")
                }
                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)
                myweb.show()
        }

        system ("test.pdf")
        system ("test2.pdf")
```

65.37 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()
                }
        }
        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:

65.38 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

65.39 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()
}
```

65.40 Using the QColorDialog Class

Example:

```
Load "guilib.ring"

oApp = new myapp { start() }

Class MyApp

        oColor  win1

        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

65.41 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

65.42 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

65.43 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

65.44 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)
                setlabeltext("User Name")
                settextvalue("Mahmoud")
                lcheck = exec()
```

```
                if lCheck win1.setwindowtitle(oInput.textvalue()) ok
        }
```

The application during the runtime

65.45 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
}
```

65.46 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()
        }

func pMove
        win1.setwindowtitle("Mouse Move , X : " + myfilter.getx() +
                            " Y : " + myfilter.gety() )
```

The application during the runtime

65.47 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()
        {

                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
                }

        ok
```

The application during the runtime

Move this label!
Welcome

Move this label!
Welcome

65.48 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

65.49 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()
}

Func pCenter
        oDesktop  = new qDesktopWidget()
```

```
        oRect = oDesktop.screenGeometry( oDesktop.primaryScreen()  )
        win1.move((oRect.width()-win1.width()) /2 , (oRect.Height()-win1.Height())/2 )
        win1.show()
```

The application during the runtime

65.50 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) {
                        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

65.51 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)

                }
                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
```

65.52 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)
                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)
```

65.53 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()

                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)
                }

                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

65.54 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)
                        show()
                }


Class MyButton from qPushButton
        nIndex = 0
```

65.55 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()
      {
        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()
        }
      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()
                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

65.56 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"
```

```
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")
                                        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)
                        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)")
                                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) {
                                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()")
                                }
                                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)
        }

        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)
                }

                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)
                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()")
                }
                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()
                }
                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)
                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
                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] + ")")

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]+
                        ","+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.

65.57 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
                  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")
                        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)

        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
         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()
                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

65.58 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()
	SetCursorPositionChangedEvent()
	SetRedoAvailableEvent()

Continued on next page

Table 65.1 – continued from previous page

Class Name	Methods to use the default Events
	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()
	SetCellDoubleClickedEvent()

Continued on next page

Table 65.1 – continued from previous page

Class Name	Methods to use the default Events
	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()

Continued on next page

Table 65.1 – 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()

65.59 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)	

65.60 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

65.61 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.

65.62 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/applications/formdesigner/tests/indexstart/indexstartView.ring
2. https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/indexstart/indexstartController.ring

65.63 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

new qApp {
        open_window(:CustomersReportController)
        exec()
}

class CustomersReportController

        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())

        func PrintEvent
                printer1 = new qPrinter(0) {
                        setoutputformat(1)
                        setoutputfilename("report.pdf")
                }
                oView {
                        web.print(printer1)
                        web.show()
                }
                system ("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

SIXTYSIX

BUILDING RINGQT APPLICATIONS FOR MOBILE

In this chapter we will learn about Building RingQt Applications for Mobile.

66.1 Download Requirements

Check the next link : http://doc.qt.io/qt-5/androidgs.html

Download

- 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

66.2 Update the Android SDK

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
```

66.3 Install Qt for Android

- You can install Qt for Android from the next link

  https://download.qt.io/archive/qt/5.5/5.5.1/

- **Run Qt Creator, Select Tools > Options > Android to add the** Android NDK and SDK paths.

  http://doc.qt.io/qtcreator/creator-developing-android.html

66.4 Using Ring2EXE

Starting from Ring 1.6 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

66.5 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

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
```

66.6 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

SIXTYSEVEN

OBJECTS LIBRARY FOR RINGQT APPLICATION

In this chapter we will learn about the objects library for RingQt applications.

Ring 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

67.1 Library Usage

- Use the Open_Window(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 Last_Window() 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 Open_Window() It will be called.
- In the view class, Define the GUI window object as an attribute called win.
- You can use Open_WindowNoShow() to avoid displaying the window.
- You can use Open_WindowAndLink() to quickly get methods to access the windows.

67.2 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 {
        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 ) )
                }
                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.

67.3 Open_WindowAndLink() Function

We can use the Open_WindowAndLink() 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/applications/formdesigner/tests/twowindowspart5/firstwindowView.ring
2. https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/twowindowspart5/firstwindowController.ring

Second Window

1. https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/twowindowspart5/secondwindowView.ring
2. https://github.com/ring-lang/ring/blob/master/applications/formdesigner/tests/twowindowspart5/secondwindowController.ring

In the next code for example (from FirstWindowController.ring)

The Open_WindowAndLink() will create an object from the SecondWindowController Class

Then will add the Method : SecondWindow(), IsSecondWindow() Methods to the FirstWindowController Class

Also will add the Method : FirstWindow(), 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 Last_Window(), Parent() and SetParentObject() methods.

```
class firstwindowController from windowsControllerParent

    oView = new firstwindowView

    func OpenSecondWindow
        Open_WindowAndLink(:SecondWindowController,self)

    func SendMessage
        if IsSecondWindow()
            SecondWindow().setMessage("Message from the first window")
        ok

    func setMessage cMessage
        oView.Label1.setText(cMessage)
```

67.4 Open_WindowInPackages() Function

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"
])
```

67.5 Objects Library Source Code

The library source code is very simple, You can check the source code files

- https://github.com/ring-lang/ring/blob/master/extensions/ringqt/objectslib/objects.ring
- https://github.com/ring-lang/ring/blob/master/extensions/ringqt/objectslib/subwindows.ring

CHAPTER

SIXTYEIGHT

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.

68.1 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.

68.2 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.

68.3 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

68.4 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.

68.5 Events Code

1. Just type the method name in the event property.

(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() )
        }
```

68.6 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.

68.7 Menubar Designer

From the Window properties we can open the Menubar Designer

68.8 Window Flags

From the Window properties we can open the Window Flags window.

68.9 Entering Items

For some controls like the List Widget we can enter items separated by comma ‘,’

68.10 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.

68.11 More Samples and Tests

Check the folder : ring/applications/formdesigner/tests

Online : https://github.com/ring-lang/ring/tree/master/applications/formdesigner/tests

CHAPTER

SIXTYNINE

MULTI-LANGUAGE APPLICATIONS

There are many ways to create multi-language Ring application!

In this chapter we will learn about using the String2Constant tool

69.1 Using String2Constant

Starting from Ring 1.8 we have the String2Constant application

You will find this tool in the ring/samples/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.

69.2 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/formdesinger/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.

69.3 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

SEVENTY

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

70.1 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

70.2 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 attibutes 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.

70.3 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).

70.4 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.

70.5 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.

70.6 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.

70.7 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
        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()

70.8 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
        name
        address
        phone
```

70.9 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


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.

70.10 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 insead braces.

```
Class MyApp

        oCon    # Attribute

        # some code here

        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()
                        }
                }
                # some code here
```

Note: We used this.attribute to access the class attribute (oCon) while we are inside Braces.

70.11 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 ) )
                }


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 ) )
                }
```

70.12 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 —> Gloabl 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"
```

```
        }
        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 attribue 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
                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
```

70.13 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
                }
                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
```

70.14 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

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
```

70.15 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

SEVENTYONE

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

71.1 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.

71.2 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
                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
========================================
```

71.3 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

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

SEVENTYTWO

SYNTAX FLEXIBILITY

In this chapter we will learn about some options that are provided automatically by the Ring compiler for syntax flexibility.

72.1 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
                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
```

72.2 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
```

72.3 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

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.

72.4 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)

myfunc2 = func (x,y) { see x+y+nl }

call myfunc(3,4)
```

Output:

```
7
7
```

72.5 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
```

72.6 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

        @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
```

72.7 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
```

72.8 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
```

72.9 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...
one
try catch...
catching error
```

72.10 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
for loop..
12345678910
switch...
one
try catch...
catching error
```

72.11 Using 'put' and 'get' as 'see' and 'give'

We can replace the 'see' keyword with the 'put' keyword.

Also we can replacew the 'give' keyword with the 'get' keyword.

Example:

```
put "Hello World" + nl
put "Enter Your Name ? " Get Name
Put "Hello " + Name
```

72.12 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 "thre" + 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 "thre" + nl
        else put "else" + nl
        }
}
```

72.13 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
```

72.14 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")
                }
        }
}
```

72.15 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
```

72.16 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
```

72.17 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
```

CHAPTER

SEVENTYTHREE

INTRODUCTION TO THE TYPE HINTS LIBRARY

In this chapter we will learn about the Type Hints Library

73.1 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.

73.2 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 ;
}
```

73.3 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
                }
        }
}
```

73.4 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 ;
}
```

73.5 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
long
float
```

```
double
void

# High Level Types
string
list
number
object

# Other
public
static
abstract
protected
override
```

CHAPTER

SEVENTYFOUR

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
-w	Display Warnings

74.1 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 : =
    Number : 30
  Operator : }
   EndLine
   Keyword : SEE
Identifier : o1
   EndLine
   Keyword : CLASS
Identifier : point
Identifier : x
Identifier : y
Identifier : z
```

```
   EndLine

=====================================================
```

74.2 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

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
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
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
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
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

====================================================
```

74.3 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
          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
 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

==================================================
```

74.4 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
         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
 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
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
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

====================================================
```

74.5 CGI Support

Command:

```
ring test.ring -cgi
```

74.6 No Run

Command:

```
ring test.ring -norun
```

74.7 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!

74.8 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
7
8
9
10
welcome
x: 10.000000
y: 20.000000
z: 30.000000

=====================================================
Date  : 2015/09/15 Time : 15:56:17
Clock : 0
=====================================================
```

74.9 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
```

CHAPTER

SEVENTYFIVE

PERFORMANCE TIPS

In this chapter we will learn more about the Ring performance.

75.1 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 next tasks in around 1 second using normal computers in the market during the last 5 years

1. Compiling 100,000 lines of code
2. Executing empty loop that count from 1 to 10,000,000
3. Executing 1000 search operation using linear search in a list contains 100,000 items, trying to find the last item (The worst case)
4. Creating list contains 1,000,000 items then summing all of the list items
5. Adding 20,000 items to the ListWidget in GUI applications
6. Adding 5,000 nodes to the TreeWidget in GUI applications
7. Printing 10,000 messages to the terminal in Console applications

Also when we need more speed we can use C/C++ extensions!

Example:

```
? "Create list contains 100,000 items"
aList = 1:100000

? "Do 1000 search operation - Find the last item (Worst Case!)"
c = clock()

for t = 1 to 1000
        find(alist,100000)
next

? "Time: " + ( clock() - c ) / clockspersecond() + " seconds"
```

Output:

```
Create list contains 100,000 items
Do 1000 search operation - Find the last item (Worst Case!)
Time: 0.87 seconds
```

Example:

```
load "guilib.ring"

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 5000
                                oItem = new qTreeWidgetItem()
                                oItem.settext(0,"Item " + t)
                                root.addchild(oItem)
                        next
                        cTime = (clock()-t1)/clockspersecond()
                        setHeaderLabel("Creating 5000 nodes in " + cTime + " seconds.")
                        addTopLevelItem(root)
                        expanditem(root)
                        blocksignals(False) setUpdatesEnabled(True)
                }
                oLayout = new qVBoxLayout() {
                        addWidget(tree)
                }
                setLayout(oLayout)
                show()
        }
        exec()
}
```

Output:

75.2 Creating Lists

Example:

```
? "Create the list using the Range operator"
t1 = clock()
aList = 1:1000000
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "Create the list using the For loop"
t1 = clock()
aList = []
for x = 1 to 1000000
        aList + x
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "Create the list using the list() function and the For loop"
t1 = clock()
aList = list(1000000)
for x = 1 to 1000000
        aList[x] = x
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"
```

Output:

```
Create the list using the Range operator
Time : 0.48 seconds
Create the list using the For loop
Time : 0.79 seconds
Create the list using the list() function and the For loop
Time : 1.56 seconds
```

Note: Creating lists using the Range operator is faster than using the For loop or the list() function

75.3 Arithmetic Operations

Example:

```
? "Using * operator"
t1 = clock()
for x = 1 to 1000000
        out = x * 2
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"

? "Using *= operator"
t1 = clock()
for x = 1 to 1000000
        out = x
        out *= 2
next
? "Time : " + ((clock()-t1)/clockspersecond()) + " seconds"
```

Output:

```
Using * operator
Time : 1.34 seconds
Using *= operator
Time : 0.47 seconds
```

Note: Using the *= operator is faster that using the * operator

75.4 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 : 5.50 seconds
Using len() before the For loop
Time : 0.24 seconds
```

Note: Using len() before the For loop is faster than using len() in the For loop.

75.5 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 "
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 : 0.28 seconds
calling 100000 methods using the dot operator
Time : 0.36 seconds
calling 100000 methods using braces
Time : 1.19 seconds
```

Note: Calling functions is faster than calling methods but the difference is very small.

Note: Using the dot operator to call methods is faster than using braces.

CHAPTER

SEVENTYSIX

DISTRIBUTING RING APPLICATIONS

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!

76.1 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.

76.2 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.

76.3 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)

76.4 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

76.5 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.

76.6 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

SEVENTYSEVEN

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 and Mobile devices

Note: We can use the Distribute Menu in the Ring Notepad application (More Easy)

77.1 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 ge generated by the Visual C/C++ Linker)
filename                 (Executable File - On Linux & MacOS X platforms)
```

77.2 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.

77.3 Example

We have test.ring contains the next code

```
see "Hello, World!" + nl
```

To build th executable file for Windows, Linux or macOS

```
ring2exe test.ring
```

To run the program (Windows)

```
test
```

To run the program (Linux and macOS)

```
./test
```

77.4 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
-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
-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
```

77.5 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
```

77.6 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
```

77.7 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
```

77.8 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

77.9 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>
        </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!

77.10 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.

77.11 Building the Form Designer for Mobile using RingQt

To build the Form Designer application (ring/applications/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>
                <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/applications/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) {
                                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.

77.12 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!

77.13 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/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/ring2exe/libs

CHAPTER

SEVENTYEIGHT

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.

78.1 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
```

78.2 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 registred 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
```

78.3 Updating the RingPM Registry

The RingPM Registry is a local copy of all registred 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)
```

78.4 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
```

78.5 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
```

78.6 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
```

78.7 Update Package

We can update a package using the Update command

```
ringpm update <packagename>
```

Example:

```
ringpm update ringnotepad
```

78.8 Remove Package

We can remove a package using the Remove command

```
ringpm remove <packagename>
```

Example:

```
ringpm remove ringnotepad
```

78.9 Deleting All Packages

We can delete all packages using the Format command

Example:

```
ringpm format
```

78.10 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 =         [

        ],
        :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 = ""
]
```

78.11 The Package Description File

The package description file contains the package information defined in the list aPackageInfo

Attribute	Description
Name	Package Name
Description	Package Desciption
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

Continued on next page

Table 78.1 – continued from previous page

Attribute	Description
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

78.12 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",
                        :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",
                "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"
]
```

78.13 The RingPM Registry File

The RingPM Registry is a local copy of all registred 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/ringpm/registry/registry.ring

CHAPTER

SEVENTYNINE

LOW LEVEL FUNCTIONS

In this chapter we will learn about the low level functions provided by Ring

- callgc()
- varptr()
- space()
- nullpointer()
- object2pointer()
- pointer2object()
- ispointer()
- ptrcmp()
- 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()

79.1 callgc() function

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.

79.2 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)

example:

```
r = 10
z = 20
see r + nl
see varptr("r","int")
see varptr("z","int")
```

Output:

```
10
00E3C740
int
2
00E3BEC0
int
2
```

Note: the low level object is a list contains three items (The Pointer, The Type, The Status)

79.3 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.

79.4 nullpointer() function

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 NULLPointer()

79.5 object2pointer() function

Use this function to get a C pointer for Ring lists and objects

Syntax:

```
object2pointer(List|Object) --> Low Level Object ( C Pointer )
```

79.6 pointer2object() function

Use this function to get the Ring list and/or object from the low level object (C Pointer)

Syntax:

```
pointer2object(Low Level Object) ---> List|Object
```

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"

# print the list items
y = pointer2object(x)
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 by reference Just use the object2pointer() and pointer2object() functions.

Tip: The object2pointer() and pointer2object() are used in the stdlib - Tree Class implementation to create a reference for the parent node (object) in the child node (another object).

79.7 ispointer() function

Check if the parameter is a pointer (C Object) or not.

Syntax:

```
IsPointer(vPara) ---> True|False
```

Example :

```
fp = fopen(filename(),"r")

? type(fp)

? ispointer(fp)
```

Output :

```
file
1
```

79.8 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
```

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
```

79.9 ringvm_cfunctionslist() function

The Function return a list of functions written in C.

Syntax:

```
RingVM_CFunctionsList() ---> List
```

Example:

```
See RingVM_CFunctionsList()
```

79.10 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
```

79.11 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
```

79.12 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

package package3
        class class1
```

Output:

```
package1
class1
11

0
00FEF838
package2
class1
17

0
00FEF978
```

```
package3
class1
23

0
00FEFF68
```

79.13 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
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
```

79.14 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)
- Temp. Memory List
- Method or Function Flag
- Caller PC
- FuncExec Flag
- ListStart Flag
- Nested Lists Pointer
- State List

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
```

79.15 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
stdsecurity.ring
stdinternet.ring
stdhashtable.ring
stdtree.ring
```

79.16 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)
```

79.17 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
```

79.18 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
```

79.19 ringvm_tracefunc()

The function return the name of the function that we are using for tracing events.

Syntax:

```
RingVM_TraceEvent() ---> cCode
```

79.20 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
```

79.21 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 scope in a specific scope.

Syntax:

```
RingVM_EvalInScope(nScope,cCode)
```

79.22 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()
```

79.23 ringvm_hideerrormsg()

We can disable/enable displaying the runtime error messages using the RingVM_HideErrorMsg() function.

Syntax:

```
RingVM_HideErrorMsg(lStatus)
```

79.24 ringvm_callfunc()

We can call a function from a string without using eval() using the ringvm_callfunc()

Syntax:

```
RingVM_CallFunc(cFuncName)
```

79.25 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 +
                "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 :
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
==========================================
====== 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
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
=========================================
```

79.26 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

# 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"
                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
        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()')
        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
```

79.27 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
y: 20.000000
z: 30.000000
```

79.28 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
```

79.29 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
```

CHAPTER

EIGHTY

THE TRACE LIBRARY AND THE INTERACTIVE DEBUGGER

In this chapter we will learn about the Trace Library and the Interactive Debugger

80.1 Loading the Trace library

To start using the Trace library, We must load it first!

```
load "tracelib.ring"
```

80.2 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
```

80.3 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"
```

80.4 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!"
```

80.5 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
        see "t = " see t see nl
        see "x = " see x see nl
```

80.6 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
```

80.7 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
```

80.8 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
        BreakPoint()
        see "After breakpoint!" +nl
        see "t = " + t + nl
        see "End of program!" + nl
```

80.9 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

We can change the variables values then continue execution

We can run the Interactive Debugger in the Output Window

CHAPTER

EIGHTYONE

EMBEDDING RING IN RING

In this chapter we will learn about embedding Ring in Ring programs and applications.

81.1 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
20
10
20
```

81.2 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')
```

81.3 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()")
                        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 enviroment
                ring_state_runcode(pState,"
                        new qWidget {
                                pObject = x
                                setwindowtitle('Message from the Sub Ring Environment')
                        }
                ")
        ring_state_delete(pState)
```

81.4 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!

81.5 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
in file Ring_EmbeddedCode
End of test!
```

CHAPTER

EIGHTYTWO

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) 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

82.1 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
```

82.2 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 */
        #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
}
```

82.3 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_vm_funcregister() function.

The ring_vm_funcregister() 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_vm_funcregister("sin",ring_vm_math_sin);
        ring_vm_funcregister("cos",ring_vm_math_cos);
```

```
        ring_vm_funcregister("tan",ring_vm_math_tan);
        ring_vm_funcregister("asin",ring_vm_math_asin);
        ring_vm_funcregister("acos",ring_vm_math_acos);
        ring_vm_funcregister("atan",ring_vm_math_atan);
        ring_vm_funcregister("atan2",ring_vm_math_atan2);
        ring_vm_funcregister("sinh",ring_vm_math_sinh);
        ring_vm_funcregister("cosh",ring_vm_math_cosh);
        ring_vm_funcregister("tanh",ring_vm_math_tanh);
        ring_vm_funcregister("exp",ring_vm_math_exp);
        ring_vm_funcregister("log",ring_vm_math_log);
        ring_vm_funcregister("log10",ring_vm_math_log10);
        ring_vm_funcregister("ceil",ring_vm_math_ceil);
        ring_vm_funcregister("floor",ring_vm_math_floor);
        ring_vm_funcregister("fabs",ring_vm_math_fabs);
        ring_vm_funcregister("pow",ring_vm_math_pow);
        ring_vm_funcregister("sqrt",ring_vm_math_sqrt);
        ring_vm_funcregister("unsigned",ring_vm_math_unsigned);
        ring_vm_funcregister("decimals",ring_vm_math_decimals);
        ring_vm_funcregister("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).

82.4 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.

82.5 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 */
}
```

82.6 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!"
```

82.7 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_ISPOINTER(int nParameterNumber);
```

The output of these functions will be 1 (True) or 0 (False).

82.8 Get Parameters Values

We can get paramters 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);
```

82.9 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 *))
```

82.10 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);
```

82.11 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);
        }
}
```

82.12 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));
                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_ISPOINTER(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_ISPOINTER() 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

82.13 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.

```
List * ring_list_new ( int nSize ) ;
void ring_list_newitem ( List *pList ) ;
Item * ring_list_getitem ( List *pList,int index ) ;
List * ring_list_delete ( List *pList ) ;
void ring_list_deleteitem ( List *pList,int index ) ;
void ring_list_print ( List *pList ) ;
int ring_list_gettype ( List *pList, int index ) ;
void ring_list_setint ( List *pList, int index ,int number ) ;
```

```
void ring_list_addint ( List *pList,int x ) ;
void ring_list_setpointer ( List *pList, int index ,void *pValue ) ;
void ring_list_addpointer ( List *pList,void *pValue ) ;
void ring_list_setfuncpointer ( List *pList, int index ,void (*pFunc)(void *) ) ;
void ring_list_addfuncpointer ( List *pList,void (*pFunc)(void *) ) ;
int ring_list_isfuncpointer ( List *pList, int index ) ;
void ring_list_setdouble ( List *pList, int index ,double number ) ;
void ring_list_adddouble ( List *pList,double x ) ;
void ring_list_setstring ( List *pList, int index ,const char *str ) ;
void ring_list_setstring2 ( List *pList, int index ,const char *str,int nStrSize ) ;
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 ) ;
List * ring_list_getlist ( List *pList, int index ) ;
void ring_list_setlist ( List *pList, int index ) ;
void ring_list_setactiveitem ( List *pList, Items *pItems, int index ) ;
void ring_list_copy ( List *pNewList, List *pList ) ;
int ring_list_isnumber ( List *pList, int index ) ;
int ring_list_isstring ( List *pList, int index ) ;
int ring_list_islist ( List *pList, int index ) ;
int ring_list_ispointer ( List *pList, int index ) ;
void ring_list_deleteallitems ( List *pList ) ;
void ring_list_insertitem ( List *pList,int x ) ;
void ring_list_insertint ( List *pList,int nPos,int x ) ;
void ring_list_insertdouble ( List *pList,int nPos,double x ) ;
void ring_list_insertpointer ( List *pList,int nPos,void *pValue ) ;
void ring_list_insertstring ( List *pList,int nPos,const char *str ) ;
void ring_list_insertstring2 ( List *pList,int nPos,const char *str,int nStrSize ) ;
void ring_list_insertfuncpointer ( List *pList,int nPos,void (*pFunc)(void *) ) ;
List * ring_list_insertlist ( List *pList,int nPos ) ;
int ring_list_isiteminsidelist ( List *pList,Item *pItem ) ;
int ring_list_findstring ( List *pList,const char *str,int nColumn ) ;
int ring_list_finddouble ( List *pList,double nNum1,int nColumn ) ;
void ring_list_sortnum ( List *pList,int left,int right,int nColumn ) ;
void ring_list_sortstr ( List *pList,int left,int right,int nColumn ) ;
int ring_list_binarysearchnum ( List *pList,double nNum1,int nColumn ) ;
int ring_list_binarysearchstr ( List *pList,const char *cFind,int nColumn ) ;
void ring_list_swap ( List *pList,int x,int y ) ;
double ring_list_getdoublecolumn ( List *pList,int nIndex,int nColumn ) ;
char * ring_list_getstringcolumn ( List *pList,int nIndex,int nColumn ) ;
void ring_list_genarray ( List *pList ) ;
void ring_list_deletearray ( List *pList ) ;
void ring_list_genhashtable ( List *pList ) ;
void ring_list_genhashtable2 ( List *pList ) ;
void ring_list_refcopy ( List *pNewList, List *pList ) ;
void ring_list_clear ( List *pList ) ;
/* Macro */
ring_list_isdouble(pList,index)
ring_list_isint(pList,index)
ring_list_deletelastitem(x)
ring_list_gethashtable(x)
ring_list_getint(pList,index)
ring_list_getpointer(pList,index)
ring_list_getfuncpointer(pList,index)
ring_list_callfuncpointer(pList,index,x)
ring_list_getdouble(pList,index)
ring_list_getstring(pList,index)
ring_list_getstringobject(pList,index)
```

```
ring_list_getstringsize(pList,index)
ring_list_getsize(x) (x->nSize)
```

82.14 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.

```
String * ring_string_new ( const char *str ) ;
String * ring_string_new2 ( const char *str,int nStrSize ) ;
String * ring_string_delete ( String *pString ) ;
int ring_string_size ( String *pString ) ;
void ring_string_set ( String *pString,const char *str ) ;
void ring_string_set2 ( String *pString,const char *str,int nStrSize ) ;
void ring_string_add ( String *pString,const char *str ) ;
void ring_string_add2 ( String *pString,const char *str,int nStrSize ) ;
void ring_string_print ( String *pString ) ;
void ring_string_setfromint ( String *pString,int x ) ;
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 ) ;
char * ring_string_find ( char *cStr1,char *cStr2 ) ;
char * ring_string_find2 ( char *cStr1,int nStrSize1,char *cStr2,int nStrSize2 ) ;
/* Macro */
ring_string_tolower(x)
ring_string_toupper(x)
ring_string_get(x)
```

82.15 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_ISPOINTER(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(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 previoud 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() 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.

82.16 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_DLL void ringlib_init(RingState *pRingState)
{
        ring_vm_funcregister("dlfunc",ring_ringlib_dlfunc);
}
```

the idea is to create the ringlib_init() function, this function will be called by the RingVM when we use the generated DLL file though the LoadLib() function.

Inside the ringlib_init() 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
```

82.17 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_ISPOINTER(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 ;
                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 ;
}
```

CHAPTER

EIGHTYTHREE

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);
```

83.1 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 execut 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
```

83.2 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 ) ;
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[] ) ;
```

```
void ring_state_runfile ( RingState *pRingState,const char *cFileName ) ;
void ring_state_runobjectfile ( RingState *pRingState,const char *cFileName ) ;
```

83.3 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");
  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

EIGHTYFOUR

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.

84.1 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
```

84.2 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

84.3 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 ;
        }
        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.

84.4 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.

84.5 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>
```

84.6 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).

84.7 Determine Structure Members Types

You can determine the pointer name before the strucuture member name.

Example:

```
SDL_Surface {flags,SDL_PixelFormat *format,w,h,pitch,void *pixels}
```

84.8 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
```

84.9 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.

84.10 Writing comments in the configuration file

- To type comments just type it between <comment> and </comment>

Example:

```
<comment>
configuration files
</comment>
```

84.11 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 previoud 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.

84.12 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>
```

84.13 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>
```

84.14 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
        GLUT_BITMAP_HELVETICA_18
</constant>
```

84.15 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 ;
        }
        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.

84.16 Threads Support

Next, another part of the configutaiton 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;
}

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)
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)
```

84.17 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++)

84.18 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
```

84.19 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"
#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>
#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) ;
        }

}
</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>

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)
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)
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)
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)
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)
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)
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.

84.20 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))
```

84.21 Loading Files

Starting from Ring 1.9 the code generator support the <loadfile> command.

```
<loadfile> filename.cf
```

This is useful to separate the extension configuraition 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
```

84.22 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 Garabage Collector will manage these C pointers.

Example

```
<class>
name: QFont
para: QString, int, int, bool
managed
</class>
```

84.23 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 modifing 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

EIGHTYFIVE

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

85.1 Location

You will find this extension in the ring/extensions/ringbeep folder

85.2 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
```

85.3 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

EIGHTYSIX

FREQUENTLY ASKED QUESTIONS (FAQ)

86.1 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 maight 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.

86.2 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.

86.3 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.

86.4 Why is Ring largely focussed 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.

86.5 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.

86.6 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.

86.7 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.

86.8 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.

86.9 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)

86.10 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

86.11 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.

86.12 Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like?

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.

86.13 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).

86.14 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' insetad 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.

86.15 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

86.16 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

86.17 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
       See x + nl
   next
```

Output

```
1
2
3
4
5
X value = 10
```

86.18 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 !

86.19 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
```

86.20 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 references and avoid creating a manager,** You can use Object2Pointer() and Pointer2Object() functions But It's not the Ring way "Spirit" to do things.

86.21 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
```

86.22 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
```

86.23 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
```

86.24 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.

86.25 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 that are more important than this problem.

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

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.

86.26 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

86.27 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"))
```

86.28 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
```

86.29 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
```

86.30 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
```

86.31 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.

86.32 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

86.33 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
```

86.34 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)

86.35 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
```

86.36 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
```

86.37 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
```

86.38 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
```

86.39 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.

86.40 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()
        {
              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()
}
```

86.41 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")
                }

                button = [btn1, btn2]
                show()
        }

        exec()

}

func click

        button[1] { settext ("Button3") }
        button[2] { settext ("Button4") }
```

86.42 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()

}
```

86.43 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

86.44 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()
                }
```

86.45 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".

86.46 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
See "Close database..." + nl
odbc_disconnect(pODBC)
odbc_close(pODBC)
```

86.47 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).

86.48 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
```

86.49 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/src/buildvccomplete.bat

86.50 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 http://ring-lang.sourceforge.net/doc/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 you 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/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.

86.51 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

                        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()
}
```

86.52 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()) )
                                ok
                        }
                next
        next
```

86.53 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/applications/formdesigner

CHAPTER

EIGHTYSEVEN

GENERAL INFORMATION

This chapter contains general information about the Ring programming language

1. Ring Architecture
2. Memory Management
3. Data Representation

87.1 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/ringlibs
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 (ringlibdepwin) ====> C libraries used for building Ring Extensions (written in C) on Windows platform.

Folder (ringringlibs) ====> Ring libraries written in Ring itself (StdLib, WebLib, GameEngine, etc)

Folder (ringrnoteexe) ====> Just C source file to create executable file that will run rnote.ring

Folder (ringvisualsrc) ====> 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++

87.2 Memory Management

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).

87.3 Data Representation

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 Stings —-> 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]
}

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
```

CHAPTER

EIGHTYEIGHT

LANGUAGE REFERENCE

In this chapter we will learn about

- Language keywords
- Language Functions
- Compiler Errors
- Runtime Errors
- Environment Errors
- Language Grammar
- Virtual Machine (VM) Instructions

88.1 Language Keywords

Keywords Count : 49

- 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

88.2 Language Functions

Functions Count : 208

acos()	add()	addattribute()	adddays()
addmethod()	ascii()	asin()	assert()
atan()	atan2()	attributes()	binarysearch()
bytes2double()	bytes2float()	bytes2int()	callgc()
ceil()	cfunctions()	char()	chdir()
classes()	classname()	clearerr()	clock()
clockspersecond()	closelib()	copy()	cos()
cosh()	currentdir()	date()	dec()
decimals()	del()	diffdays()	dir()
double2bytes()	eval()	exefilename()	exefolder()
exp()	fabs()	fclose()	feof()
ferror()	fexists()	fflush()	fgetc()
fgetpos()	fgets()	filename()	find()
float2bytes()	floor()	fopen()	fputc()
fputs()	fread()	freopen()	fseek()
fsetpos()	ftell()	functions()	fwrite()
getattribute()	getchar()	globals()	hex()
hex2str()	input()	insert()	int2bytes()
intvalue()	isalnum()	isalpha()	isandroid()
isattribute()	iscfunction()	isclass()	iscntrl()
isdigit()	isfreebsd()	isfunction()	isglobal()
isgraph()	islinux()	islist()	islocal()
islower()	ismacosx()	ismethod()	ismsdos()
isnull()	isnumber()	isobject()	ispackage()
ispackageclass()	ispointer()	isprint()	isprivateattribut
isprivatemethod()	ispunct()	isspace()	isstring()
isunix()	isupper()	iswindows()	iswindows64()
isxdigit()	left()	len()	lines()
list()	list2str()	loadlib()	locals()
log()	log10()	lower()	max()
mergemethods()	methods()	min()	murmur3hash()
nullpointer()	number()	object2pointer()	objectid()
packageclasses()	packagename()	packages()	perror()
pointer2object()	pow()	prevfilename()	ptrcmp()
raise()	random()	read()	remove()
rename()	reverse()	rewind()	right()
ring_give()	ring_see()	ring_state_delete()	ring_state_findva
ring_state_init()	ring_state_main()	ring_state_mainfile()	ring_state_new()
ring_state_newvar()	ring_state_runcode()	ring_state_runfile()	ring_state_runob
ring_state_setvar()	ringvm_callfunc()	ringvm_calllist()	ringvm_cfunctions
ringvm_classeslist()	ringvm_evalinscope()	ringvm_fileslist()	ringvm_functionsl
ringvm_give()	ringvm_hideerrormsg()	ringvm_info()	ringvm_memorylist
ringvm_packageslist()	ringvm_passerror()	ringvm_scopescount()	ringvm_see()
ringvm_settrace()	ringvm_tracedata()	ringvm_traceevent()	ringvm_tracefunc
setattribute()	shutdown()	sin()	sinh()
sort()	space()	sqrt()	srandom()
str2hex()	str2hexcstyle()	str2list()	strcmp()
string()	substr()	swap()	sysget()
system()	tan()	tanh()	tempfile()
tempname()	time()	timelist()	trim()
type()	ungetc()	unsigned()	upper()
varptr()	version()	windowsnl()	write()

88.3 Compiler Errors

- Error (C1) : Error in parameters list, expected identifier
- Error (C2) : Error in class name
- Error (C3) : Unclosed control strucutre, 'ok' is missing
- Error (C4) : Unclosed control strucutre, 'end' is missing
- Error (C5) : Unclosed control strucutre, 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 strucutre, '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!

88.4 Runtime Errors

- Error (R1) : Cann'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 destory 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 <=14) !
- 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

88.5 Environment Errors

- Error (E1) : Caught SegFault
- Error (E2) : Out of Memory
- Error (E3) : Deleting scope while no scope!
- Error (E4) : Long VM Instruction!

88.6 Language Grammar

Program —> {statement}

Statement —> 'package' <Identifier> { '.' <Identifier> } ['{' {statement} '}'] ['end'|'endpackage']

Statement —> 'class' <Identifier> ['from'|':'|'<' <Identifier>] ['{' {statement} '}'][''end'|'endclass']

Statement —> 'func'|'def' <Identifier> [ParaList] ['{' {statement} '}'][''end'|'endfunc']

Statement —> 'import' <Identifier> { '.' <Identifier> }

Statement —> 'private'

Statement —> 'load' ['package'] <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'|'}'

Statement —> 'Switch' <Expr> ['{'] { 'on'|'case' <Expr> {statement} } ['other' {Statement}] 'off'|'end'|'}'

Statement —> 'for' <Identifier> '=' <Expr> 'to' <Expr> ['step' <Expr>] ['{'] {Statement} 'next'|'end'|'}'

Statement —> 'for' <Identifier> 'in' <Expr> ['step' <Expr>] ['{'] {statement} 'next'|'end'|'}'

Statement —> 'while' <Expr> ['{'] {statement} 'end'|'}'

Statement —> 'do' {statement} 'again' <Expr>

Statement —> 'try' {statement} ['{'] 'catch' {statement} 'done'|'end'|'}'

Statement —> 'return' <Expr>

Statement —> 'bye'

Statement —> 'exit'

Statement —> 'loop'

Statement —> <Expr>

Statement —> epslion

ParaList —> epslion

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' <Identifier>

Factor —> <AnonymousFunction>

Factor —> 'call' <identifier> { '.' <Identifier> } '(' <Parameters> ')'

List —> '[' [<Expr> { ',' <Expr> }] ']'

Mixer —> { '.' <Identifier> }

Mixer —> '[' <Expr> ']'

Mixer —> '(' [<Expr> [{ ',' <Expr> }]] ')'

Mixer —> '{' {Statement} '}'

AnonymousFunction —> 'func'|'def'| [<ParaList>] '{' {Statement} '}'

88.7 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_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_GREATEREQUAL	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_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

(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_JUMPVARLENUM	Jump if variable value is <= numeric value
• ICO_JUMPVARPLENUM	Jump if variable value (using pointer) <= numeric value
• ICO_LOADFUNCP	Push function pointer
• ICO_PUSHPLOCAL	Push pointer to local variable
• ICO_INCLPJUMP	Increment value using pointer to local variable then jump
• ICO_JUMPVARLPLENUM	Jump if the variable value (using pointer) <= numeric value
• ICO_INCPJUMPSTEP1	Increment value using variable pointer then jump (for loop step = 1)
• ICO_JUMPVARPLENUMSTEP1	Increment value using variable pointer then jump (for loop step = 1)

(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 call 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

EIGHTYNINE

RINGQT CLASSES REFERENCE

89.1 CodeEditor Class

C++ Reference : http://doc.qt.io/qt-5/CodeEditor.html

Parameters : QWidget *

Parent Class : QPlainTextEdit

- void setCompleter(QCompleter *c)
- QCompleter *completer(void)
- void setLineNumbersAreaColor(QColor oColor)
- void setLineNumbersAreaBackColor(QColor oColor)

89.2 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)

89.3 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)

89.4 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)

89.5 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)

89.6 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)

89.7 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)

89.8 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)

89.9 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)

89.10 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)

89.11 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)

89.12 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,QString)
- QVariant dynamicCall_3(const char *function, QList<QVariant>)
- QVariant dynamicCall_4(char *function,int)
- QString generateDocumentation(void)
- bool isNull(void)
- QAxObject *querySubObject(const char *name)
- QAxObject *querySubObject_2(const char * name, QList<QVariant>)

- QAxObject *querySubObject_3(const char * name,const char *)
- QAxObject *querySubObject_4(const char * name,int)
- QAxObject *querySubObject_5(const char * name,const char *,int)
- QAxObject *querySubObject_6(const char * name,const char *,const char *)
- QAxObject *querySubObject_7(const char * name,const char *,const char *,int)
- QAxObject *querySubObject_8(const char * name,const char *,int,int)
- QAxObject *querySubObject_9(const char * name,float)
- QAxObject *querySubObject_10(const char * name,const char *,float)
- QAxObject *querySubObject_11(const char * name,const char *,const char *,float)
- QAxObject *querySubObject_12(const char * name,const char *,float,float)
- QAxObject *querySubObject_13(const char * name,double)
- QAxObject *querySubObject_14(const char * name,const char *,double)
- QAxObject *querySubObject_15(const char * name,const char *,const char *,double)
- QAxObject *querySubObject_16(const char * name,const char *,double,double)
- QAxObject *querySubObject_17(const char * name,int,const char *)
- QAxObject *querySubObject_18(const char * name,int,const char *,const char *)
- QAxObject *querySubObject_19(const char * name,int,const char *,const char *,const char *)
- QAxObject *querySubObject_20(const char * name,int,const char *,const char *,int)
- QAxObject *querySubObject_21(const char * name,int,const char *,int,const char *)
- QAxObject *querySubObject_22(const char * name,float,const char *)
- QAxObject *querySubObject_23(const char * name,float,const char *,const char *)
- QAxObject *querySubObject_24(const char * name,float,const char *,const char *,const char *)
- QAxObject *querySubObject_25(const char * name,float,const char *,const char *,float)
- QAxObject *querySubObject_26(const char * name,float,const char *,float,const char *)
- QAxObject *querySubObject_27(const char * name,double,const char *)
- QAxObject *querySubObject_28(const char * name,double,const char *,const char *)
- QAxObject *querySubObject_29(const char * name,double,const char *,const char *,const char *)
- QAxObject *querySubObject_30(const char * name,double,const char *,const char *,double)
- QAxObject *querySubObject_31(const char * name,double,const char *,double,const char *)
- QAxObject *querySubObject_32(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_33(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_34(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_35(const char *name,QVariant,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_36(const char *name,QVariant,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_37(const char *name,QVariant,QVariant,QVariant)
- QAxObject *querySubObject_38(const char *name,QVariant,QVariant)

- QAxObject *querySubObject_39(const char *name,QVariant)
- bool setControl(QString)
- QStringList verbs(void)

89.13 QAxObject Class

C++ Reference : http://doc.qt.io/qt-5/QAxObject.html

Parameters : QString

Parent Class : QAxBase

- bool doVerb(QString)

89.14 QAxWidget Class

C++ Reference : http://doc.qt.io/qt-5/QAxWidget.html

Parameters : QWidget *parent, Qt::WindowFlags f

Parent Class : QAxBase

- bool doVerb(QString)

89.15 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)

89.16 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)

89.17 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)

89.18 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)

89.19 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)

89.20 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)

89.21 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)

89.22 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)

89.23 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)

89.24 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)

89.25 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)

89.26 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)

89.27 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)

89.28 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)

89.29 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)

89.30 QBrush Class

C++ Reference : http://doc.qt.io/qt-5/QBrush.html

Parameters : void

- QColor color(void)
- QGradient *gradient(void)
- bool isOpaque(void)
- QMatrix matrix(void)
- void setColor(QColor)
- void setMatrix(QMatrix)
- 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)

89.31 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)

89.32 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)

89.33 QByteArray Class

C++ Reference : http://doc.qt.io/qt-5/QByteArray.html

Parameters : void

- QByteArray append(const char *str)
- 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)
- 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)

89.34 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)

89.35 QCamera Class

C++ Reference : http://doc.qt.io/qt-5/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)

89.36 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)

89.37 QCameraViewfinder Class

C++ Reference : http://doc.qt.io/qt-5/QCameraViewfinder.html

Parameters : QWidget *

Parent Class : QVideoWidget

89.38 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)

89.39 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)

89.40 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)

89.41 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)

89.42 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)

89.43 QColorDialog Class

C++ Reference : http://doc.qt.io/qt-5/QColorDialog.html

Parameters : void

Parent Class : QDialog

- QColor currentColor(void)
- void open(void)
- int options(void)
- QColor selectedColor(void)
- void setCurrentColor(QColor)
- void setOption(QColorDialog::ColorDialogOption option, bool on)
- void setOptions(QColorDialog::ColorDialogOption options)
- bool testOption(QColorDialog::ColorDialogOption option)
- QColor customColor(int index)
- int customCount(void)
- void setCustomColor(int index, int color)
- void setStandardColor(int index, int color)
- int getcolor(void)

89.44 QComboBox Class

C++ Reference : http://doc.qt.io/qt-5/QComboBox.html

Parameters : QWidget *

Parent Class : QWidget

- void addItem(QString,int)
- 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 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)

89.45 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)

89.46 QCompleter2 Class

C++ Reference : http://doc.qt.io/qt-5/QCompleter2.html

Parameters : QAbstractItemModel *model, QObject *parent

Parent Class : QCompleter

89.47 QCompleter3 Class

C++ Reference : http://doc.qt.io/qt-5/QCompleter3.html

Parameters : QStringList list, QObject *parent

Parent Class : QCompleter

89.48 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)

89.49 QCursor Class

C++ Reference : http://doc.qt.io/qt-5/QCursor.html

Parameters : void

- QBitmap *bitmap(void)
- QPoint hotSpot(void)
- QBitmap *mask(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)

89.50 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)
- QString longDayName(int weekday)
- QString longMonthName(int month)
- QString shortDayName(int weekday)
- QString shortMonthName(int month)

89.51 QDateEdit Class

C++ Reference : http://doc.qt.io/qt-5/QDateEdit.html

Parameters : QWidget *parent

Parent Class : QDateTimeEdit

89.52 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)
- void setTime_t(uint seconds)

- 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)
- uint toTime_t(void)
- 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)
- QDateTime fromTime_t(uint seconds)

89.53 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)

89.54 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)

89.55 QDesktopWidget Class

C++ Reference : http://doc.qt.io/qt-5/QDesktopWidget.html

Parameters : void

Parent Class : QWidget

- QRect availableGeometry(int screen)
- bool isVirtualDesktop(void)
- int primaryScreen(void)
- QWidget *screen(int screen)
- int screenCount(void)
- QRect screenGeometry(int screen)
- int screenNumber(QWidget *widget)

89.56 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)

89.57 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)

89.58 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)

89.59 QDirModel Class

C++ Reference : http://doc.qt.io/qt-5/QDirModel.html

Parameters : void

- QIcon fileIcon(QModelIndex)
- QFileInfo fileInfo(QModelIndex)
- QString fileName(QModelIndex)
- QString filePath(QModelIndex)
- int filter(void)
- QFileIconProvider *iconProvider(void)
- QModelIndex index(QString path, int column)
- bool isDir(QModelIndex)
- bool isReadOnly(void)
- bool lazyChildCount(void)
- QModelIndex mkdir(QModelIndex parent, QString name)
- QStringList nameFilters(void)
- bool remove(QModelIndex index)
- bool resolveSymlinks(void)
- bool rmdir(QModelIndex index)
- void setFilter(QDir::Filter filters)
- void setIconProvider(QFileIconProvider *provider)
- void setLazyChildCount(bool enable)
- void setNameFilters(QStringList filters)
- void setReadOnly(bool enable)
- void setResolveSymlinks(bool enable)

- void setSorting(QDir::SortFlag sort)
- int sorting(void)
- void refresh(QModelIndex parent)

89.60 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)

89.61 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)

89.62 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

89.63 QDragLeaveEvent Class

C++ Reference : http://doc.qt.io/qt-5/QDragLeaveEvent.html

Parameters : void

Parent Class : QEvent

89.64 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)

89.65 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)

89.66 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)

89.67 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)

89.68 QFile2 Class

C++ Reference : http://doc.qt.io/qt-5/QFile2.html

Parameters : QString

Parent Class : QFile

89.69 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)

89.70 QFileDialog Class

C++ Reference : http://doc.qt.io/qt-5/QFileDialog.html

Parameters : QWidget *parent

Parent Class : QWidget

- int acceptMode(void)
- QString defaultSuffix(void)
- QDir directory(void)
- QUrl directoryUrl(void)
- int fileMode(void)
- int 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, const char *member)
- int options(void)
- QAbstractProxyModel *proxyModel(void)
- bool restoreState(QByteArray)
- QByteArray saveState(void)
- void selectFile(QString)

- void selectMimeTypeFilter(QString)
- void selectNameFilter(QString)
- void selectUrl(QUrl)
- QStringList selectedFiles(void)
- QString selectedNameFilter(void)
- void setDefaultSuffix(QString)
- void setDirectory(QString)
- void setDirectoryUrl(QUrl)
- void setFileMode(QFileDialog::FileMode mode)
- void setFilter(QDir::Filter filters)
- void setHistory(QStringList)
- void setIconProvider(QFileIconProvider *provider)
- void setItemDelegate(QAbstractItemDelegate *delegate)
- void setLabelText(QFileDialog::DialogLabel label, QString)
- void setMimeTypeFilters(QStringList)
- void setNameFilter(QString)
- void setNameFilters(QStringList)
- void setOption(QFileDialog::Option option, bool)
- void setOptions(QFileDialog::Option options)
- void setProxyModel(QAbstractProxyModel *proxyModel)
- int viewMode(void)
- QString getExistingDirectory(QWidget *,QString,QString, QFileDialog::Option)
- QUrl getExistingDirectoryUrl(QWidget *,QString,QUrl, QFileDialog::Option,QStringList)
- QString getOpenFileName(QWidget *,QString,QString,QString)
- QUrl getSaveFileUrl(QWidget *,QString, QUrl,QString, QString *, QFileDialog::Option options,QStringList)

89.71 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)
- QDateTime created(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)

89.72 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)

89.73 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)
- QString lastResortFamily(void)
- QString lastResortFont(void)
- double letterSpacing(void)
- int letterSpacingType(void)

- bool overline(void)
- int pixelSize(void)
- int pointSize(void)
- double pointSizeF(void)
- bool rawMode(void)
- QString rawName(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 setRawMode(bool enable)
- void setRawName(QString)
- 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)

89.74 QFontDialog Class

C++ Reference : http://doc.qt.io/qt-5/QFontDialog.html

Parameters : void

Parent Class : QDialog

- QFont currentFont(void)
- void open(QObject *receiver, const char *member)
- int options(void)
- QFont selectedFont(void)
- void setCurrentFont(QFont)
- void setOption(QFontDialog::FontDialogOption option, bool on)
- void setOptions(QFontDialog::FontDialogOption options)
- bool testOption(QFontDialog::FontDialogOption option)
- int getfont(void)

89.75 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 width(QString text, int len)
- int width_2(QChar ch)
- int xHeight(void)

89.76 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)

89.77 QFrame2 Class

C++ Reference : http://doc.qt.io/qt-5/QFrame2.html

Parameters : void

Parent Class : QFrame

89.78 QFrame3 Class

C++ Reference : http://doc.qt.io/qt-5/QFrame3.html

Parameters : QWidget *parent

Parent Class : QFrame

89.79 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)

89.80 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)

89.81 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)

89.82 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)

89.83 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)

89.84 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)

89.85 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)

89.86 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)

89.87 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)

89.88 QGeoSatelliteInfoSource Class

C++ Reference : http://doc.qt.io/qt-5/QGeoSatelliteInfoSource.html

- QString sourceName(void)
- int updateInterval(void)

89.89 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)

89.90 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)

89.91 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)

89.92 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)

89.93 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)

89.94 QHBoxLayout Class

C++ Reference : http://doc.qt.io/qt-5/QHBoxLayout.html

Parameters : void

Parent Class : QBoxLayout

- void addWidget(QWidget *)
- void addLayout(QLayout *)

89.95 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)

89.96 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)

89.97 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)

89.98 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)

89.99 QIcon Class

C++ Reference : http://doc.qt.io/qt-5/QIcon.html

Parameters : QPixmap

89.100 QImage Class

C++ Reference : http://doc.qt.io/qt-5/QImage.html

Parameters : void

- bool allGray(void)
- int bitPlaneCount(void)
- uchar *bits(void)
- int byteCount(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)
- QImage transformed(QMatrix, Qt::TransformationMode mode)
- bool valid(int x, int y)
- int width(void)

89.101 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)

89.102 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)

89.103 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)
- const char * rawData(int * size)
- void setArray(QJsonArray array)
- void setObject(QJsonObject object)
- QByteArray toBinaryData(void)
- QByteArray toJson(QJsonDocument::JsonFormat format)
- QVariant toVariant(void)
- QJsonDocument fromBinaryData(QByteArray data, QJsonDocument::DataValidation validation)
- QJsonDocument fromJson(QByteArray json, QJsonParseError * error)
- QJsonDocument fromRawData(char * data, int size, QJsonDocument::DataValidation validation)
- QJsonDocument fromVariant(QVariant variant)

89.104 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)

89.105 QJsonParseError Class

C++ Reference : http://doc.qt.io/qt-5/QJsonParseError.html

Parameters : void

- QString errorString(void)

89.106 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)

89.107 QKeySequence Class

C++ Reference : http://doc.qt.io/qt-5/QKeySequence.html

Parameters : QString

89.108 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)

89.109 QLabel Class

C++ Reference : http://doc.qt.io/qt-5/QLabel.html

Parameters : QWidget *

Parent Class : QWidget

- 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(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)

89.110 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)

89.111 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)
- 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)
- 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)

89.112 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)

89.113 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)

89.114 QListWidget Class

C++ Reference : http://doc.qt.io/qt-5/QListWidget.html

Parameters : QWidget *

Parent Class : QListView

- void addItem(QString)
- int count(void)
- 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)

89.115 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)

89.116 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)

89.117 QMatrix4x4 Class

C++ Reference : http://doc.qt.io/qt-5/QMatrix4x4.html

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)
- QMatrix toAffine(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)

89.118 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)

89.119 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)

89.120 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)

89.121 QMediaPlayer Class

C++ Reference : http://doc.qt.io/qt-5/QMediaPlayer.html

Parameters : void

- int bufferStatus(void)
- QMediaContent currentMedia(void)
- QNetworkConfiguration currentNetworkConfiguration(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 volume(void)
- void pause(void)
- void play(void)
- void setMuted(bool muted)
- void setPlaylist(QMediaPlaylist *playlist)

- void setPosition(int position)
- void setVolume(int volume)
- void stop(void)

89.122 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)

89.123 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)

89.124 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)

89.125 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 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)

89.126 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)

89.127 QMutex Class

C++ Reference : http://doc.qt.io/qt-5/QMutex.html

Parameters : QMutex::RecursionMode

- bool isRecursive(void)
- void lock(void)
- void unlock(void)

89.128 QMutexLocker Class

C++ Reference : http://doc.qt.io/qt-5/QMutexLocker.html

Parameters : QMutex *

- QMutex * mutex(void)
- void relock(void)
- void unlock(void)

89.129 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)

89.130 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)

89.131 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)

89.132 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)

89.133 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)

89.134 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 signalsBlocked(void)
- int startTimer(int interval)
- QThread *thread(void)
- void deleteLater(void)

89.135 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)

89.136 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)

89.137 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)

89.138 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)

89.139 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)

89.140 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)

89.141 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)

89.142 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)

89.143 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)

89.144 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)

89.145 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)

89.146 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)

89.147 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)

89.148 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)

89.149 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)

89.150 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)

89.151 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)
- void initFrom(QWidget *widget)

- 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)

89.152 QPainter2 Class

C++ Reference : http://doc.qt.io/qt-5/QPainter2.html

Parameters : QPaintDevice *

Parent Class : QPainter

89.153 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)

89.154 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)

89.155 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)

89.156 QPixmap Class

C++ Reference : http://doc.qt.io/qt-5/QPixmap.html

Parameters : const char *

- 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(QTransform transform, Qt::TransformationMode mode)
- QPixmap transformed_2(QMatrix 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)
- QMatrix trueMatrix_2(QMatrix m, int w, int h)

89.157 QPixmap2 Class

C++ Reference : http://doc.qt.io/qt-5/QPixmap2.html

Parameters : int width, int height

Parent Class : QPixmap

89.158 QPlainTextEdit Class

C++ Reference : http://doc.qt.io/qt-5/QPlainTextEdit.html

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- 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 setTabStopWidth(int width)
- void setTextCursor(QTextCursor cursor)
- void setTextInteractionFlags(Qt::TextInteractionFlags flags)
- void setUndoRedoEnabled(bool enable)
- void setWordWrapMode(QTextOption::WrapMode policy)
- bool tabChangesFocus(void)
- int tabStopWidth(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)

89.159 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)

89.160 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)

89.161 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)

89.162 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)

89.163 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)

89.164 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

89.165 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_2(QString command, 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)

89.166 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)

89.167 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)

89.168 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)

89.169 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)

89.170 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)

89.171 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)

89.172 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)

89.173 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)
- QVector<QRect> rects(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)

89.174 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)

89.175 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)

89.176 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)

89.177 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)

89.178 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)

89.179 QScrollBar Class

C++ Reference : http://doc.qt.io/qt-5/QScrollBar.html

Parameters : QWidget *parent

Parent Class : QAbstractSlider

89.180 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)

89.181 QSerialPortInfo Class

C++ Reference : http://doc.qt.io/qt-5/QSerialPortInfo.html

Parameters : void

- QString description(void)
- bool hasProductIdentifier(void)
- bool hasVendorIdentifier(void)
- bool isBusy(void)
- bool isNull(void)
- QString manufacturer(void)
- QString portName(void)
- quint16 productIdentifier(void)
- void swap(QSerialPortInfo other)
- QString systemLocation(void)
- quint16 vendorIdentifier(void)

89.182 QSize Class

C++ Reference : http://doc.qt.io/qt-5/QSize.html

Parameters : int width, int height

89.183 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)

89.184 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)

89.185 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)

89.186 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)

89.187 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)

89.188 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)

89.189 QSqlDriverCreatorBase Class

C++ Reference : http://doc.qt.io/qt-5/QSqlDriverCreatorBase.html

Parameters : void

89.190 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)
- int number(void)
- void setDatabaseText(QString)
- void setDriverText(QString)
- void setNumber(int number)
- void setType(QSqlError::ErrorType type)
- QString text(void)
- QSqlError::ErrorType type(void)

89.191 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)

89.192 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)

89.193 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)

89.194 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)

89.195 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)

89.196 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)

89.197 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)

89.198 QString2 Class

C++ Reference : http://doc.qt.io/qt-5/QString2.html

Parameters : void

- QString append(QString str)
- QByteArray toUtf8(void)
- QByteArray toLatin1(void)

- QByteArray toLocal8Bit(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)
- 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)

89.199 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)
- void swap(int,int)
- QString takeAt(int)
- QString takeFirst(void)
- QString takeLast(void)
- QString value(int)

89.200 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)

89.201 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)

89.202 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)

89.203 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)

89.204 QTabWidget Class

C++ Reference : http://doc.qt.io/qt-5/QTabWidget.html

Parameters : QWidget *parent

Parent Class : QWidget

- 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)

89.205 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)

89.206 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)

89.207 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)

89.208 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)

89.209 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)

89.210 QTest Class

C++ Reference : http://doc.qt.io/qt-5/QTest.html

- void qsleep(int)

89.211 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)

89.212 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)

89.213 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)

89.214 QTextCodec Class

C++ Reference : http://doc.qt.io/qt-5/QTextCodec.html

- QTextCodec *codecForName(const char *name)
- void setCodecForLocale(QTextCodec *c)

89.215 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)

89.216 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)

89.217 QTextEdit Class

C++ Reference : http://doc.qt.io/qt-5/QTextEdit.html

Parameters : QWidget *

Parent Class : QAbstractScrollArea

- 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 setTabStopWidth(int width)
- void setTextCursor(QTextCursor)
- void setTextInteractionFlags(Qt::TextInteractionFlag flags)
- void setUndoRedoEnabled(bool enable)
- void setWordWrapMode(QTextOption::WrapMode policy)
- bool tabChangesFocus(void)
- int tabStopWidth(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)

89.218 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)

89.219 QTextStream2 Class

C++ Reference : http://doc.qt.io/qt-5/QTextStream2.html

Parameters : QIODevice * device

Parent Class : QTextStream

89.220 QTextStream3 Class

C++ Reference : http://doc.qt.io/qt-5/QTextStream3.html

Parameters : FILE * fileHandle, QIODevice::OpenMode

Parent Class : QTextStream

89.221 QTextStream4 Class

C++ Reference : http://doc.qt.io/qt-5/QTextStream4.html

Parameters : QString *, QIODevice::OpenMode

Parent Class : QTextStream

89.222 QTextStream5 Class

C++ Reference : http://doc.qt.io/qt-5/QTextStream5.html

Parameters : QByteArray *, QIODevice::OpenMode

Parent Class : QTextStream

89.223 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)

89.224 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)

89.225 QTime Class

C++ Reference : http://doc.qt.io/qt-5/QTime.html

Parameters : void

- QTime addMSecs(int ms)
- QTime addSecs(int s)
- int elapsed(void)
- int hour(void)
- bool isNull(void)
- bool isValid(void)
- int minute(void)
- int msec(void)
- int msecsSinceStartOfDay(void)
- int msecsTo(QTime)
- int restart(void)
- int second(void)
- int secsTo(QTime)
- bool setHMS(int h, int m, int s, int ms)
- void start(void)
- QString toString(QString)
- QTime currentTime(void)
- QTime fromMSecsSinceStartOfDay(int msecs)
- QTime fromString(QString,QString)

89.226 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)

89.227 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)

89.228 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)

89.229 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)

89.230 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)
- bool isFirstItemColumnSpanned(QTreeWidgetItem *item)
- 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 setFirstItemColumnSpanned(QTreeWidgetItem *item, bool span)
- 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)

89.231 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)

89.232 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 topLevelDomain(QUrl::ComponentFormattingOption options)
- QString userInfo(QUrl::ComponentFormattingOption options)
- QString userName(QUrl::ComponentFormattingOption options)
- QUrl fromLocalFile(QString)

89.233 QUuid Class

C++ Reference : http://doc.qt.io/qt-5/QUuid.html

Parameters : void

- QString toString(void)

89.234 QVBoxLayout Class

C++ Reference : http://doc.qt.io/qt-5/QVBoxLayout.html

Parameters : void

Parent Class : QBoxLayout

- void addWidget(QWidget *)
- void addLayout(QLayout *)

89.235 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)

89.236 QVariant2 Class

C++ Reference : http://doc.qt.io/qt-5/QVariant2.html

Parent Class : QVariant

Parameters : int

89.237 QVariant3 Class

C++ Reference : http://doc.qt.io/qt-5/QVariant3.html

Parent Class : QVariant

Parameters : float

89.238 QVariant4 Class

C++ Reference : http://doc.qt.io/qt-5/QVariant4.html

Parent Class : QVariant

Parameters : double

89.239 QVariant5 Class

C++ Reference : http://doc.qt.io/qt-5/QVariant5.html

Parent Class : QVariant

Parameters : QString

89.240 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)

89.241 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)
- void setY(float y)
- void setZ(float z)
- QPoint toPoint(void)

- QPointF toPointF(void)
- QVector2D toVector2D(void)
- QVector4D toVector4D(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)

89.242 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)

89.243 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)

89.244 QVideoWidgetControl Class

C++ Reference : http://doc.qt.io/qt-5/QVideoWidgetControl.html

Parent Class : QMediaControl

89.245 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)

89.246 QWebView Class

C++ Reference : http://doc.qt.io/archives/qt-5.5/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)

89.247 QWebView Class

C++ Reference : http://doc.qt.io/archives/qt-5.5/qwebview.html

Parameters : QWidget *parent

Parent Class : QWebEngineView

89.248 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)
- void getContentsMargins(int *left, int *top, int *right, int *bottom)
- 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)

89.249 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)

89.250 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)

89.251 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)

89.252 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)

89.253 QXmlStreamEntityResolver Class

C++ Reference : http://doc.qt.io/qt-5/QXmlStreamEntityResolver.html

Parameters : void

89.254 QXmlStreamNamespaceDeclaration Class

C++ Reference : http://doc.qt.io/qt-5/QXmlStreamNamespaceDeclaration.html

Parameters : void

- QStringRef namespaceUri(void)
- QStringRef prefix(void)

89.255 QXmlStreamNotationDeclaration Class

C++ Reference : http://doc.qt.io/qt-5/QXmlStreamNotationDeclaration.html

Parameters : void

- QStringRef name(void)
- QStringRef publicId(void)
- QStringRef systemId(void)

89.256 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)

89.257 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)

89.258 RingCodeHighlighter Class

Parameters : QTextDocument *parent

- void setColors(QColor c1,QColor c2,QColor c3,QColor c4,QColor c5)
- void setKeywordsBold(int nStatus)

CHAPTER

NINETY

RINGMURMURHASH FUNCTIONS REFERENCE

" **MurmurHash is a non-cryptographic hash function suitable for general hash-based lookup.** It was created by Austin Appleby in 2008 and is currently hosted on Github along with its test suite named 'SMHasher'. It also exists in a number of variants,[5] all of which have been released into the public domain. The name comes from two basic operations, multiply (MU) and rotate (R), used in its inner loop. "

Murmurhash extension is an extension written to implement a full implementation for the MurmurHash library.

Developer: Hassan Ahmed

90.1 MurmurHash1 functions

uint32_t murmurhash1(string key, int seed, [bool return_type]);

uint32_t murmurhash1_aligned(string key, int seed, [bool return_type]);

90.2 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]);

90.3 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.

90.4 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

NINETYONE

FOXRING FUNCTIONS REFERENCE

As a FoxPro developer, I started writing a class with some of the functions I used in FoxPro.

The goal is to create as many functions as possible with the behavior similar to FoxPro functions.

Developer: Jose Rosado

91.1 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 ocurrence 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.

Continued on next page

Table 91.1 – continued from previous page

Function Name	Description
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
	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
	characteres 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.

91.2 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.
```

91.3 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()
*               : 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.
```

91.4 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.
```

91.5 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.
```

91.6 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.
```

91.7 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
```

91.8 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
```

91.9 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.
```

91.10 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.
*               :
```

91.11 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.
```

91.12 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.
```

91.13 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
```

91.14 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.
```

91.15 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.
```

91.16 frAllTrim() function

```
Syntax  : lcReturnValue = frAllTrim(tcExpression, tcCharacter)
```

91.17 frLTrim() function

```
Syntax  : lcRet = frLTrim(tcExpression, tcCharacter)
```

91.18 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.
```

91.19 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>
*              : Returns the drive characters of a file extension
*              : from a complete path.
```

91.20 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.
```

91.21 frRTrim() function

```
Syntax : lcRet = frRTrim(tcExpression, tcCharacter)
```

91.22 frJustPath() function

```
Syntax  : tcReturnValue = frJustPath(tcExpression)
```

91.23 frForceExt() function

```
Syntax  : tcReturnValue = frForceExt(tcFileName, tcNewExtension)
```

91.24 frALines() function

```
Syntax  : tnReturnValue = frALines(taList, tcExpression, tcSeparator)
```

91.25 frJustFName() function

```
Syntax  : tcReturnValue = frJustFName(tcExpression)
```

91.26 frPadL() function

```
Syntax  : tcReturnValue = frPadL(tcString, tnLen, tcChar)
```

91.27 frPadR() function

```
Syntax  : tcReturnValue = frPadR(tcString, tnLen, tcChar)
```

91.28 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>
```

91.29 frReplicate() function

```
Syntax  : tcReturnValue = frReplicate(tcString, tnTimes)
```

91.30 frLen() function

```
Syntax  : tnReturnValue = frLen(tcString)
```

91.31 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.
*              :
*              : <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
```

91.32 frSubStr() function

```
Syntax  : tcReturnValue = frSubStr(tcString, tnInitialPosition, tnNumberBytes)
```

91.33 frStrTran() function

```
Syntax  : tcReturnValue = frStrTran(tcString, tcOldString, tcNewString)
```

91.34 frListToString() function

```
* Syntax       : lcRet = frListToString(taList)
* Remarks      : This function doesn't exist in VFP.
```

91.35 frInt() function

```
Syntax          : lnInt = frInt(tnExpression)
```

91.36 frStringToList() function

```
* Syntax       : laList = frStringToList(tcExpression)
* Remarks      : This function doesn't exist in VFP.
```

91.37 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>
```

91.38 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.
```

91.39 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 oder null value.
```

91.40 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 de default, for example space " " or a comma ",".
*               : Issue frSetSeparatorTo(Null) to reset the value to its default.
*               :
* Returns       : None
```

91.41 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.
---------------------------------------------------------------------------
```

91.42 frVarType() function

```
* Syntax       : lcRet = frVarType(tuExpression)
* Description  : Returns the data type of an expression.
*              :
* Arguments    : <tuExpression>
*              : Expecifies 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 characteres that frVarType()
*              : returns for each data type.
*              :
*              : -------------------   -------------------------------------
*              : Return Value          Data Type
*              : -------------------   -------------------------------------
*              : C                     Character
*              : N                     Numeric
*              : A                     List
*              : O                     Object
*              : U                     Undefined type
*              : -------------------   -------------------------------------
*              :
* Returns      : Character
```

91.43 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
?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
?mf.frTransform("123", "@L 999999")
?mf.frTransform(123, "@L 999999")
```

CHAPTER

NINETYTWO

BIGNUMBER LIBRARY

In this chapter we will learn about using the Big Number library in the Ring programming language.

Developers : Bert Mariani, Gal Zsolt (~ CalmoSoft ~)

92.1 Loading the library

Before using the next function load the bignumber.ring library

```
load "bignumber.ring"
# Use Big Number library functions
```

92.2 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

? "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)
3716981956283439342389882561522895 08 (Output continue in next line)
4939646110432289716923898608970 69
```

92.3 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)
```

92.4 BigNumber Class

The library contains the next class

```
class BigNumber
        func init aPara
        func operator cOperator, Para
        func print
        func value
```

92.5 Library Source Code

You can see the library source code in : ring/ringlibs/bignumber folder

Source Code : https://github.com/ring-lang/ring/blob/master/ringlibs/bignumber/bignumber.ring

CHAPTER

NINETYTHREE

RINGLIBCURL FUNCTIONS REFERENCE

- 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
- 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

NINETYFOUR

RINGLIBZIP FUNCTIONS 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_file(const char *cZIPFile,const char *cFile)
- 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)

CHAPTER

NINETYFIVE

RINGCONSOLECOLORS FUNCTIONS 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

NINETYSIX

RINGALLEGRO FUNCTIONS 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)

CHAPTER

NINETYSEVEN

RINGLIBSDL FUNCTIONS 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

NINETYEIGHT

RINGLIBUV FUNCTIONS 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

NINETYNINE

RINGFREEGLUT FUNCTIONS 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)
- GLfloat glutStrokeWidthf(void *font, int character)
- int glutBitmapLength(void *font, char * string)

- int glutStrokeLength(void *font, char * string)
- GLfloat glutStrokeLengthf(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)

CHAPTER

RINGOPENGL (OPENGL 1.1) FUNCTIONS 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
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *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 glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)

- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *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 void *pointer)
- 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 glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)

- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- 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 glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)

- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(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 glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- 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, void *pixels)
- 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 glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)

- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, 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 glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- 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 glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)

- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- 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 glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)

- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)

- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- 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 glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)

- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)
- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvi(GLenum target, GLenum pname, GLint param)
- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)

- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)
- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

CHAPTER

ONE

RINGOPENGL (OPENGL 1.2) FUNCTIONS 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
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *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 glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)

- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *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 void *pointer)
- 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 glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)

- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- 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 glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)

- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(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 glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- 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, void *pixels)
- 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 glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)

- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, 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 glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- 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 glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- 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 glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)

- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)

- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- 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 glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)

- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)
- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvi(GLenum target, GLenum pname, GLint param)
- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)

- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)
- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

CHAPTER

TWO

RINGOPENGL (OPENGL 1.3) FUNCTIONS 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
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)

- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *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 glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)

- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *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 void *pointer)
- 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 glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)

- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- 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 glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(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 glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- 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, void *pixels)
- 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 glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)

- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, 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 glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- 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 glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)

- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- 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 glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)

- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- 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 glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)

- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvi(GLenum target, GLenum pname, GLint param)
- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)

- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)
- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

CHAPTER

THREE

RINGOPENGL (OPENGL 1.4) FUNCTIONS 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
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)

- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *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 glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)

- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *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 void *pointer)
- 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 glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)

- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- 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 glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(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 glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- 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, void *pixels)
- 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 glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)

- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, 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 glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- 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 glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)

- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- 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 glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)

- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)

- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- 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 glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)

- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)
- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvi(GLenum target, GLenum pname, GLint param)
- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)

- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)
- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

CHAPTER

FOUR

RINGOPENGL (OPENGL 1.5) FUNCTIONS 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
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *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 glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)

- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)
- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *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 void *pointer)
- 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 glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)
- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- 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 glFogfv(GLenum pname, const GLfloat *params)

- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(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 glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)
- void glGetPolygonStipple(GLubyte *mask)
- 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, void *pixels)
- 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 glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)
- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, 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 glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- 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 glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)
- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- 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 glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)
- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)

- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- 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 glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)
- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)

- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)
- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvi(GLenum target, GLenum pname, GLint param)
- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)

- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)
- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

CHAPTER

FIVE

RINGOPENGL (OPENGL 2.0) FUNCTIONS 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
- void glAccum(GLenum op, GLfloat value)
- void glAlphaFunc(GLenum func, GLclampf ref)
- GLboolean glAreTexturesResident(GLsizei n, const GLuint *textures, GLboolean *residences)
- void glArrayElement(GLint i)
- void glBegin(GLenum mode)
- void glBindTexture(GLenum target, GLuint texture)
- void glBitmap(GLsizei width, GLsizei height, GLfloat xorig, GLfloat yorig, GLfloat xmove, GLfloat ymove, const GLubyte *bitmap)
- void glBlendFunc(GLenum sfactor, GLenum dfactor)
- void glCallList(GLuint list)
- void glCallLists(GLsizei n, GLenum type, const void *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 glClipPlane(GLenum plane, const GLdouble *equation)
- void glColor3b(GLbyte red, GLbyte green, GLbyte blue)
- void glColor3bv(const GLbyte *v)
- void glColor3d(GLdouble red, GLdouble green, GLdouble blue)
- void glColor3dv(const GLdouble *v)
- void glColor3f(GLfloat red, GLfloat green, GLfloat blue)
- void glColor3fv(const GLfloat *v)
- void glColor3i(GLint red, GLint green, GLint blue)
- void glColor3iv(const GLint *v)
- void glColor3s(GLshort red, GLshort green, GLshort blue)
- void glColor3sv(const GLshort *v)
- void glColor3ub(GLubyte red, GLubyte green, GLubyte blue)
- void glColor3ubv(const GLubyte *v)
- void glColor3ui(GLuint red, GLuint green, GLuint blue)
- void glColor3uiv(const GLuint *v)
- void glColor3us(GLushort red, GLushort green, GLushort blue)
- void glColor3usv(const GLushort *v)
- void glColor4b(GLbyte red, GLbyte green, GLbyte blue, GLbyte alpha)
- void glColor4bv(const GLbyte *v)
- void glColor4d(GLdouble red, GLdouble green, GLdouble blue, GLdouble alpha)
- void glColor4dv(const GLdouble *v)
- void glColor4f(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha)
- void glColor4fv(const GLfloat *v)
- void glColor4i(GLint red, GLint green, GLint blue, GLint alpha)
- void glColor4iv(const GLint *v)
- void glColor4s(GLshort red, GLshort green, GLshort blue, GLshort alpha)
- void glColor4sv(const GLshort *v)
- void glColor4ub(GLubyte red, GLubyte green, GLubyte blue, GLubyte alpha)
- void glColor4ubv(const GLubyte *v)
- void glColor4ui(GLuint red, GLuint green, GLuint blue, GLuint alpha)
- void glColor4uiv(const GLuint *v)

- void glColor4us(GLushort red, GLushort green, GLushort blue, GLushort alpha)
- void glColor4usv(const GLushort *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 void *pointer)
- 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 glCullFace(GLenum mode)
- void glDeleteLists(GLuint list, GLsizei range)
- void glDeleteTextures(GLsizei n, const GLuint *textures)
- void glDepthFunc(GLenum func)
- void glDepthMask(GLboolean flag)
- void glDepthRange(GLclampd zNear, GLclampd zFar)
- void glDisable(GLenum cap)
- void glDisableClientState(GLenum array)
- void glDrawArrays(GLenum mode, GLint first, GLsizei count)
- void glDrawBuffer(GLenum mode)
- void glDrawElements(GLenum mode, GLsizei count, GLenum type, const void *indices)
- void glDrawPixels(GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glEdgeFlag(GLboolean flag)
- void glEdgeFlagPointer(GLsizei stride, const void *pointer)
- void glEdgeFlagv(const GLboolean *flag)
- void glEnable(GLenum cap)
- void glEnableClientState(GLenum array)
- void glEnd(void)
- void glEndList(void)
- void glEvalCoord1d(GLdouble u)
- void glEvalCoord1dv(const GLdouble *u)
- void glEvalCoord1f(GLfloat u)
- void glEvalCoord1fv(const GLfloat *u)
- void glEvalCoord2d(GLdouble u, GLdouble v)

- void glEvalCoord2dv(const GLdouble *u)
- void glEvalCoord2f(GLfloat u, GLfloat v)
- void glEvalCoord2fv(const GLfloat *u)
- void glEvalMesh1(GLenum mode, GLint i1, GLint i2)
- void glEvalMesh2(GLenum mode, GLint i1, GLint i2, GLint j1, GLint j2)
- 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 glFogfv(GLenum pname, const GLfloat *params)
- void glFogi(GLenum pname, GLint param)
- void glFogiv(GLenum pname, const GLint *params)
- void glFrontFace(GLenum mode)
- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- GLuint glGenLists(GLsizei range)
- void glGenTextures(GLsizei n, GLuint *textures)
- void glGetBooleanv(GLenum pname, GLboolean *params)
- void glGetClipPlane(GLenum plane, GLdouble *equation)
- void glGetDoublev(GLenum pname, GLdouble *params)
- GLenum glGetError(void)
- void glGetFloatv(GLenum pname, GLfloat *params)
- void glGetIntegerv(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 glGetPixelMapfv(GLenum map, GLfloat *values)
- void glGetPixelMapuiv(GLenum map, GLuint *values)
- void glGetPixelMapusv(GLenum map, GLushort *values)
- void glGetPointerv(GLenum pname, void* *params)

- void glGetPolygonStipple(GLubyte *mask)
- 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, void *pixels)
- 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 glHint(GLenum target, GLenum mode)
- void glIndexMask(GLuint mask)
- void glIndexPointer(GLenum type, GLsizei stride, const void *pointer)
- void glIndexd(GLdouble c)
- void glIndexdv(const GLdouble *c)
- void glIndexf(GLfloat c)
- void glIndexfv(const GLfloat *c)
- void glIndexi(GLint c)
- void glIndexiv(const GLint *c)
- void glIndexs(GLshort c)
- void glIndexsv(const GLshort *c)
- void glIndexub(GLubyte c)
- void glIndexubv(const GLubyte *c)
- void glInitNames(void)
- void glInterleavedArrays(GLenum format, GLsizei stride, const void *pointer)
- GLboolean glIsEnabled(GLenum cap)
- GLboolean glIsList(GLuint list)
- GLboolean glIsTexture(GLuint texture)
- void glLightModelf(GLenum pname, GLfloat param)
- void glLightModelfv(GLenum pname, const GLfloat *params)
- void glLightModeli(GLenum pname, GLint param)
- void glLightModeliv(GLenum pname, const GLint *params)
- void glLightf(GLenum light, GLenum pname, GLfloat param)
- void glLightfv(GLenum light, GLenum pname, const GLfloat *params)

- void glLighti(GLenum light, GLenum pname, GLint param)
- void glLightiv(GLenum light, GLenum pname, const GLint *params)
- void glLineStipple(GLint factor, GLushort pattern)
- void glLineWidth(GLfloat width)
- void glListBase(GLuint base)
- void glLoadIdentity(void)
- void glLoadMatrixd(const GLdouble *m)
- void glLoadMatrixf(const GLfloat *m)
- void glLoadName(GLuint name)
- void glLogicOp(GLenum opcode)
- void glMap1d(GLenum target, GLdouble u1, GLdouble u2, GLint stride, GLint order, const GLdouble *points)
- void glMap1f(GLenum target, GLfloat u1, GLfloat u2, GLint stride, GLint order, 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 glMap2f(GLenum target, GLfloat u1, GLfloat u2, GLint ustride, GLint uorder, GLfloat v1, GLfloat v2, GLint vstride, GLint vorder, const GLfloat *points)
- 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 glMaterialfv(GLenum face, GLenum pname, const GLfloat *params)
- void glMateriali(GLenum face, GLenum pname, GLint param)
- void glMaterialiv(GLenum face, GLenum pname, const GLint *params)
- void glMatrixMode(GLenum mode)
- void glMultMatrixd(const GLdouble *m)
- void glMultMatrixf(const GLfloat *m)
- void glNewList(GLuint list, GLenum mode)
- void glNormal3b(GLbyte nx, GLbyte ny, GLbyte nz)
- void glNormal3bv(const GLbyte *v)
- void glNormal3d(GLdouble nx, GLdouble ny, GLdouble nz)
- void glNormal3dv(const GLdouble *v)
- void glNormal3f(GLfloat nx, GLfloat ny, GLfloat nz)
- void glNormal3fv(const GLfloat *v)
- void glNormal3i(GLint nx, GLint ny, GLint nz)
- void glNormal3iv(const GLint *v)
- void glNormal3s(GLshort nx, GLshort ny, GLshort nz)

- void glNormal3sv(const GLshort *v)
- void glNormalPointer(GLenum type, GLsizei stride, const void *pointer)
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble zNear, GLdouble zFar)
- 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 glPointSize(GLfloat size)
- void glPolygonMode(GLenum face, GLenum mode)
- void glPolygonOffset(GLfloat factor, GLfloat units)
- void glPolygonStipple(const GLubyte *mask)
- void glPopAttrib(void)
- void glPopClientAttrib(void)
- void glPopMatrix(void)
- void glPopName(void)
- void glPrioritizeTextures(GLsizei n, const GLuint *textures, const GLclampf *priorities)
- void glPushAttrib(GLbitfield mask)
- void glPushClientAttrib(GLbitfield mask)
- void glPushMatrix(void)
- void glPushName(GLuint name)
- void glRasterPos2d(GLdouble x, GLdouble y)
- void glRasterPos2dv(const GLdouble *v)
- void glRasterPos2f(GLfloat x, GLfloat y)
- void glRasterPos2fv(const GLfloat *v)
- void glRasterPos2i(GLint x, GLint y)
- void glRasterPos2iv(const GLint *v)
- void glRasterPos2s(GLshort x, GLshort y)
- void glRasterPos2sv(const GLshort *v)
- void glRasterPos3d(GLdouble x, GLdouble y, GLdouble z)
- void glRasterPos3dv(const GLdouble *v)

- void glRasterPos3f(GLfloat x, GLfloat y, GLfloat z)
- void glRasterPos3fv(const GLfloat *v)
- void glRasterPos3i(GLint x, GLint y, GLint z)
- void glRasterPos3iv(const GLint *v)
- void glRasterPos3s(GLshort x, GLshort y, GLshort z)
- void glRasterPos3sv(const GLshort *v)
- void glRasterPos4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glRasterPos4dv(const GLdouble *v)
- void glRasterPos4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glRasterPos4fv(const GLfloat *v)
- void glRasterPos4i(GLint x, GLint y, GLint z, GLint w)
- void glRasterPos4iv(const GLint *v)
- void glRasterPos4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glRasterPos4sv(const GLshort *v)
- void glReadBuffer(GLenum mode)
- void glReadPixels(GLint x, GLint y, GLsizei width, GLsizei height, GLenum format, GLenum type, void *pixels)
- void glRectd(GLdouble x1, GLdouble y1, GLdouble x2, GLdouble y2)
- void glRectdv(const GLdouble *v1, const GLdouble *v2)
- void glRectf(GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2)
- void glRectfv(const GLfloat *v1, const GLfloat *v2)
- void glRecti(GLint x1, GLint y1, GLint x2, GLint y2)
- void glRectiv(const GLint *v1, const GLint *v2)
- void glRects(GLshort x1, GLshort y1, GLshort x2, GLshort y2)
- void glRectsv(const GLshort *v1, const GLshort *v2)
- GLint glRenderMode(GLenum mode)
- void glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z)
- void glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z)
- 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 glSelectBuffer(GLsizei size, GLuint *buffer)
- void glShadeModel(GLenum mode)
- void glStencilFunc(GLenum func, GLint ref, GLuint mask)
- void glStencilMask(GLuint mask)
- void glStencilOp(GLenum fail, GLenum zfail, GLenum zpass)

- void glTexCoord1d(GLdouble s)
- void glTexCoord1dv(const GLdouble *v)
- void glTexCoord1f(GLfloat s)
- void glTexCoord1fv(const GLfloat *v)
- void glTexCoord1i(GLint s)
- void glTexCoord1iv(const GLint *v)
- void glTexCoord1s(GLshort s)
- void glTexCoord1sv(const GLshort *v)
- void glTexCoord2d(GLdouble s, GLdouble t)
- void glTexCoord2dv(const GLdouble *v)
- void glTexCoord2f(GLfloat s, GLfloat t)
- void glTexCoord2fv(const GLfloat *v)
- void glTexCoord2i(GLint s, GLint t)
- void glTexCoord2iv(const GLint *v)
- void glTexCoord2s(GLshort s, GLshort t)
- void glTexCoord2sv(const GLshort *v)
- void glTexCoord3d(GLdouble s, GLdouble t, GLdouble r)
- void glTexCoord3dv(const GLdouble *v)
- void glTexCoord3f(GLfloat s, GLfloat t, GLfloat r)
- void glTexCoord3fv(const GLfloat *v)
- void glTexCoord3i(GLint s, GLint t, GLint r)
- void glTexCoord3iv(const GLint *v)
- void glTexCoord3s(GLshort s, GLshort t, GLshort r)
- void glTexCoord3sv(const GLshort *v)
- void glTexCoord4d(GLdouble s, GLdouble t, GLdouble r, GLdouble q)
- void glTexCoord4dv(const GLdouble *v)
- void glTexCoord4f(GLfloat s, GLfloat t, GLfloat r, GLfloat q)
- void glTexCoord4fv(const GLfloat *v)
- void glTexCoord4i(GLint s, GLint t, GLint r, GLint q)
- void glTexCoord4iv(const GLint *v)
- void glTexCoord4s(GLshort s, GLshort t, GLshort r, GLshort q)
- void glTexCoord4sv(const GLshort *v)
- void glTexCoordPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glTexEnvf(GLenum target, GLenum pname, GLfloat param)
- void glTexEnvfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexEnvi(GLenum target, GLenum pname, GLint param)

- void glTexEnviv(GLenum target, GLenum pname, const GLint *params)
- void glTexGend(GLenum coord, GLenum pname, GLdouble param)
- void glTexGendv(GLenum coord, GLenum pname, const GLdouble *params)
- void glTexGenf(GLenum coord, GLenum pname, GLfloat param)
- void glTexGenfv(GLenum coord, GLenum pname, const GLfloat *params)
- void glTexGeni(GLenum coord, GLenum pname, GLint param)
- void glTexGeniv(GLenum coord, GLenum pname, const GLint *params)
- void glTexImage1D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexImage2D(GLenum target, GLint level, GLint internalformat, GLsizei width, GLsizei height, GLint border, GLenum format, GLenum type, const void *pixels)
- void glTexParameterf(GLenum target, GLenum pname, GLfloat param)
- void glTexParameterfv(GLenum target, GLenum pname, const GLfloat *params)
- void glTexParameteri(GLenum target, GLenum pname, GLint param)
- void glTexParameteriv(GLenum target, GLenum pname, const GLint *params)
- void glTexSubImage1D(GLenum target, GLint level, GLint xoffset, GLsizei width, GLenum format, GLenum type, const void *pixels)
- void glTexSubImage2D(GLenum target, GLint level, GLint xoffset, GLint yoffset, GLsizei width, GLsizei height, GLenum format, GLenum type, const void *pixels)
- void glTranslated(GLdouble x, GLdouble y, GLdouble z)
- void glTranslatef(GLfloat x, GLfloat y, GLfloat z)
- void glVertex2d(GLdouble x, GLdouble y)
- void glVertex2dv(const GLdouble *v)
- void glVertex2f(GLfloat x, GLfloat y)
- void glVertex2fv(const GLfloat *v)
- void glVertex2i(GLint x, GLint y)
- void glVertex2iv(const GLint *v)
- void glVertex2s(GLshort x, GLshort y)
- void glVertex2sv(const GLshort *v)
- void glVertex3d(GLdouble x, GLdouble y, GLdouble z)
- void glVertex3dv(const GLdouble *v)
- void glVertex3f(GLfloat x, GLfloat y, GLfloat z)
- void glVertex3fv(const GLfloat *v)
- void glVertex3i(GLint x, GLint y, GLint z)
- void glVertex3iv(const GLint *v)
- void glVertex3s(GLshort x, GLshort y, GLshort z)
- void glVertex3sv(const GLshort *v)

- void glVertex4d(GLdouble x, GLdouble y, GLdouble z, GLdouble w)
- void glVertex4dv(const GLdouble *v)
- void glVertex4f(GLfloat x, GLfloat y, GLfloat z, GLfloat w)
- void glVertex4fv(const GLfloat *v)
- void glVertex4i(GLint x, GLint y, GLint z, GLint w)
- void glVertex4iv(const GLint *v)
- void glVertex4s(GLshort x, GLshort y, GLshort z, GLshort w)
- void glVertex4sv(const GLshort *v)
- void glVertexPointer(GLint size, GLenum type, GLsizei stride, const void *pointer)
- void glViewport(GLint x, GLint y, GLsizei width, GLsizei height)

CHAPTER

SIX

RINGOPENGL (OPENGL 2.1) FUNCTIONS 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
- 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)

CHAPTER

SEVEN

RINGOPENGL (OPENGL 3.0) FUNCTIONS 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
- 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)

CHAPTER

EIGHT

RINGOPENGL (OPENGL 3.1) FUNCTIONS 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
- 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)

CHAPTER

NINE

RINGOPENGL (OPENGL 3.2) FUNCTIONS 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)

CHAPTER

RINGOPENGL (OPENGL 3.3) FUNCTIONS 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
- GL_RGB10_A2UI
- 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)

CHAPTER

ONE

RINGOPENGL (OPENGL 4.0) FUNCTIONS 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
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- 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)

CHAPTER

TWO

RINGOPENGL (OPENGL 4.1) FUNCTIONS 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
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- 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)

CHAPTER

THREE

RINGOPENGL (OPENGL 4.2) FUNCTIONS 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
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- 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)

CHAPTER

FOUR

RINGOPENGL (OPENGL 4.3) FUNCTIONS 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
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- 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)

CHAPTER

FIVE

RINGOPENGL (OPENGL 4.4) FUNCTIONS 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
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- GL_PRIMITIVE_RESTART_FOR_PATCHES_SUPPORTED
- GL_MAX_VERTEX_ATTRIB_STRIDE
- GL_TEXTURE_BUFFER_BINDING
- 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)

CHAPTER

SIX

RINGOPENGL (OPENGL 4.5) FUNCTIONS 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
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- GL_PRIMITIVE_RESTART_FOR_PATCHES_SUPPORTED
- GL_MAX_VERTEX_ATTRIB_STRIDE
- GL_TEXTURE_BUFFER_BINDING
- GL_CONTEXT_FLAG_ROBUST_ACCESS_BIT
- 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)

CHAPTER

SEVEN

RINGOPENGL (OPENGL 4.6) FUNCTIONS 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
- GL_VERTEX_ATTRIB_ARRAY_DIVISOR
- GL_RGB10_A2UI
- GL_SAMPLE_SHADING
- GL_MIN_SAMPLE_SHADING_VALUE
- GL_MIN_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_OFFSET
- GL_MAX_PROGRAM_TEXTURE_GATHER_COMPONENTS
- GL_TEXTURE_CUBE_MAP_ARRAY

- GL_TEXTURE_BINDING_CUBE_MAP_ARRAY
- GL_PROXY_TEXTURE_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY
- GL_SAMPLER_CUBE_MAP_ARRAY_SHADOW
- GL_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_UNSIGNED_INT_SAMPLER_CUBE_MAP_ARRAY
- GL_TRANSFORM_FEEDBACK_PAUSED
- GL_TRANSFORM_FEEDBACK_ACTIVE
- GL_COMPRESSED_RGBA_BPTC_UNORM
- GL_COMPRESSED_SRGB_ALPHA_BPTC_UNORM
- GL_COMPRESSED_RGB_BPTC_SIGNED_FLOAT
- GL_COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT
- GL_COPY_READ_BUFFER_BINDING
- GL_COPY_WRITE_BUFFER_BINDING
- GL_NUM_SHADING_LANGUAGE_VERSIONS
- GL_VERTEX_ATTRIB_ARRAY_LONG
- GL_PRIMITIVE_RESTART_FOR_PATCHES_SUPPORTED
- GL_MAX_VERTEX_ATTRIB_STRIDE
- GL_TEXTURE_BUFFER_BINDING
- GL_CONTEXT_FLAG_ROBUST_ACCESS_BIT
- GL_PARAMETER_BUFFER
- GL_PARAMETER_BUFFER_BINDING
- GL_TRANSFORM_FEEDBACK_OVERFLOW
- GL_TRANSFORM_FEEDBACK_STREAM_OVERFLOW
- GL_VERTICES_SUBMITTED
- GL_PRIMITIVES_SUBMITTED
- GL_VERTEX_SHADER_INVOCATIONS
- GL_TESS_CONTROL_SHADER_PATCHES
- GL_TESS_EVALUATION_SHADER_INVOCATIONS
- GL_GEOMETRY_SHADER_PRIMITIVES_EMITTED
- GL_FRAGMENT_SHADER_INVOCATIONS
- GL_COMPUTE_SHADER_INVOCATIONS
- GL_CLIPPING_INPUT_PRIMITIVES
- GL_CLIPPING_OUTPUT_PRIMITIVES
- GL_TEXTURE_MAX_ANISOTROPY
- GL_MAX_TEXTURE_MAX_ANISOTROPY

- GL_POLYGON_OFFSET_CLAMP
- GL_SHADER_BINARY_FORMAT_SPIR_V
- GL_SPIR_V_BINARY
- GL_SPIR_V_EXTENSIONS
- GL_NUM_SPIR_V_EXTENSIONS
- 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)

CHAPTER

EIGHT

RESOURCES

In this section you will find resources about the language

118.1 Ring Language Website

For news about the language check the website

http://ring-lang.net

http://ring-lang.sf.net

118.2 Ring Group

For questions use the Ring Group (English)

https://groups.google.com/forum/#!forum/ring-lang

118.3 Source Code

Ring is Free-Open Source (MIT License)

Source Code : https://github.com/ring-lang/ring

118.4 Arabic Resources

- Arabic Tutorials by Sara Hamdy

URL : https://ringprogramming4arab.wordpress.com/%D8%A7%D9%84%D9%85%D8%AD%D8%AA%D9%88%D9%8A-2/

- Arabic Tutorials by Ahmed Hassouna (YouTube)

URL : https://www.youtube.com/watch?v=6VIHMyrEilw&list=PLHIfW1KZRIfl6KzfLziFl650MmThnQ0jT

- Arabic Tutorials by Alaa Mohamed (YouTube)

URL : https://www.youtube.com/watch?v=uSv5rJp6snw&list=PLgc0fyJgivssosIJu4kJGk5avIjaMzYHK

- Arabic Tutorials by Mahmoud Fayed (Facebook (Live) + YouTube)

URL : https://github.com/ring-lang/ring/blob/master/developers/Arabic/README.md

118.5 Contact the Authors

Ring Team : http://ring-lang.sourceforge.net/team.html

3D Cube and Texture
Using RingOpenGL and RingAllegro for 3D Graphics, 654

Access List Items by String Index
Lists, 267
Access Objects Using Braces
Object Oriented Programming, 356
Access String Letters
Strings, 271
Accessing the class attributes from braces inside class methods
Scope Rules, 814
Add Items
Lists, 261
addattribute()
Reflection and Meta-programming, 387
AddDays()
Date and Time, 280
Adding code to the generated code
Code Generator, 930
Adding Hyperlink to QLabel
Desktop and Mobile Development, 711
addmethod()
Reflection and Meta-programming, 387
Analog Clock
Applications developed in little hours, 8
Animate Class
Game Engine for 2D Games, 557
Animate Events
Game Engine for 2D Games, 568
Animation
Game Engine for 2D Games, 562
Animation and Functions
Game Engine for 2D Games, 563
Anonymous and Nested Functions
Functional Programming, 372
Application Class
Web Development (CGI Library), 507
Application Database
Deploying Web Applications in the Cloud, 515
Applications
How to contribute?, 196
Applications developed in little hours
Analog Clock, 8
Calculator Application, 12
FetchStockData Application, 5
Fifteen Puzzle Game 2, 6
Google API Shortener Application, 7
Innovative, 21
Introduction, 1
Practical, 22
Quotes about Ring, 1
Samples in this book, 15
Squares Puzzle Game, 10
TicTacToe Game, 9
Video-Music-Player Application, 11
Werdy Application, 14
Windows StartUp Manager Application, 13
apppath()
Stdlib Functions, 394
Arabic Resources
Resources, 2039
Arithmetic Operations
Performance Tips, 859
Arithmetic Operators
Operators, 227
Ascii()
Data Type, 288
ASCIIList2Str()
Stdlib Functions, 410
Assert()
Eval() and Debugging, 315
Assignment Operators
Operators, 228
attributes()
Reflection and Meta-programming, 384

Better Call Command
What is new in Ring 1.2?, 173
Better Cards Game
What is new in Ring 1.8?, 78
Better Code Generator For Extensions
What is new in Ring 1.8?, 79
Better Code Generator for Extensions
What is new in Ring 1.1?, 183

What is new in Ring 1.5?, 124
What is new in Ring 1.9?, 66
Better Documentation
What is new in Ring 1.1?, 183
What is new in Ring 1.7?, 85
Better Documentation Generator for Extensions
What is new in Ring 1.5?, 125
Better Form Designer
What is new in Ring 1.8?, 77
Better Functions
What is new in Ring 1.2?, 170
Better Loop|Exit Command
What is new in Ring 1.3?, 164
Better Memory Management
What is new in Ring 1.9?, 65
Better Natural Language Programming Support
What is new in Ring 1.1?, 175
Better Objects Library
What is new in Ring 1.5?, 111
Better Performance
What is new in Ring 1.8?, 68
Better Quality
What is new in Ring 1.2?, 173
What is new in Ring 1.5?, 133
Better Ring Compiler and VM
What is new in Ring 1.8?, 79
Better Ring For Android
What is new in Ring 1.6?, 90
Better Ring Notepad
What is new in Ring 1.2?, 170
What is new in Ring 1.3?, 160
What is new in Ring 1.5?, 102
What is new in Ring 1.6?, 93
What is new in Ring 1.7?, 84
What is new in Ring 1.9?, 56
Better Ring VM
What is new in Ring 1.7?, 85
Better Ring2EXE
What is new in Ring 1.7?, 85
Better RingQt
What is new in Ring 1.2?, 170
What is new in Ring 1.3?, 156
What is new in Ring 1.4?, 151
What is new in Ring 1.5?, 109
What is new in Ring 1.6?, 95
What is new in Ring 1.7?, 84
What is new in Ring 1.8?, 78
What is new in Ring 1.9?, 62
Better RingREPL
What is new in Ring 1.6?, 96
Better RingVM
What is new in Ring 1.6?, 96
Better RingZip
What is new in Ring 1.7?, 85
Better Scripts for building Ring
What is new in Ring 1.6?, 92
Better StdLib
What is new in Ring 1.3?, 164
What is new in Ring 1.4?, 149
What is new in Ring 1.5?, 105
What is new in Ring 1.6?, 96
What is new in Ring 1.9?, 57
Better Tests
What is new in Ring 1.10?, 49
Better Trace Library
What is new in Ring 1.7?, 83
Better WebLib
What is new in Ring 1.4?, 149
What is new in Ring 1.5?, 106
bignumber
BigNumber Class, 1233
BigNumber Functions, 1232
Examples, 1231
Introduction, 1230
Library Source Code, 1233
Loading the library, 1231
BigNumber Class
bignumber, 1233
BigNumber Functions
bignumber, 1232
BigNumber Library
What is new in Ring 1.9?, 58
binarydigits()
Stdlib Functions, 403
Bitwise Operators
Operators, 228
BraceError() Method
Natural Language Programming, 449
BraceExprEval Method
Natural Language Programming, 448
BraceStart and BraceEnd Methods
Natural Language Programming, 447
Branching
Control Structures, 230
Control Structures - Second Style, 237
Control Structures - Third Style, 240
BreakPoint
The Trace Library and the Interactive Debugger, 907
Building From Source Code
Building using CMake, 193
Building using Fedora Linux, 189
Building using MacOS X, 191
Building using Microsoft Windows, 185
Building using Ubuntu Linux, 187
Introduction, 184
Building Games For Android
Building the project, 604

Download Requirements and Update the Android SDK, 603
Introduction, 602
Project Folder, 603
Building RingQt Applications for Mobile
Comments about developing for Android using RingQt, 788
Download Requirements, 787
Install Qt for Android, 787
Introduction, 786
The Qt project for your Ring application, 788
Update the Android SDK, 787
Using Ring2EXE, 788
Building standalone console application
Distributing Ring Application using Ring2EXE, 865
Building the Cards Game for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 867
Building the Form Designer for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 869
Building the project
Building Games For Android, 604
Building the Weight History Application for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 868
Building using CMake
Building From Source Code, 193
Building using Fedora Linux
Building From Source Code, 189
Building using MacOS X
Building From Source Code, 191
Building using Microsoft Windows
Building From Source Code, 185
Building using Ubuntu Linux
Building From Source Code, 187

Calculator Application
Applications developed in little hours, 12
What is new in Ring 1.5?, 101
Call Functions
Functions - First Style, 247
Functions - Second Style, 251
Functions - Third Style, 255
callgc()
Low Level Functions, 884
Calling a function sharing the name with a method in the current class
Scope Rules for Functions and Methods, 826
Calling Functions and Methods
Performance Tips, 860
Can I connect to dbase/harbour database?
Frequently Asked Questions, 976
Can Ring work on Windows XP?
Frequently Asked Questions, 978
capitalized()
Stdlib Functions, 397
cfunctions()
Reflection and Meta-programming, 377
CGI Support
Command Line Options, 854
Change Focus
Desktop and Mobile Development, 748
Change Language Keywords
Syntax Flexibility, 828
Change Language Operators
Syntax Flexibility, 829
Change the '=' operator to 'is'
Natural Language Programming, 445
Change the Ring Keyword 'And'
Natural Language Programming, 443
Change the Ring Operator '+'
Natural Language Programming, 444
Change: Basic Extensions are separated from RingVM
What is new in Ring 1.4?, 144
changestring()
Stdlib Functions, 406
Char()
Data Type, 288
ChDir() Function
System Functions, 311
Check Character
Data Type, 283
Check Data Type
Data Type, 282
Check Parameters Count
Extension, 916
Check Parameters Type
Extension, 917
Chess Game
What is new in Ring 1.10?, 41
Classes and Objects
Object Oriented Programming, 354
Classes and their Methods to use the default events
Desktop and Mobile Development, 778
classes()
Reflection and Meta-programming, 381
classname()
Reflection and Meta-programming, 383
Clean Natural Code
Natural Language Programming, 450
Clearerr()
Files, 301
Client Example
RingLibuv, 547
Client Example - Using Classes
RingLibuv, 550
clock()
Date and Time, 278
Close Window Event

RingLibSDL, 541
ClosPerSecond()
Date and Time, 278
Code Generator
Adding code to the generated code, 930
Comments in configuration file, 932
Configuration file, 928
Configuration file for the Allegro library, 933
Configuration Files Examples, 948
Constants Type, 933
Defining Constants, 931
Determine Structure Members Types, 931
Enum and Numbers, 932
Executing code during code generation, 932
Filtering using Expressions, 932
function prototype, 928
Introduction, 927
Loading Files, 947
Managed Classes, 947
Prefix for Functions Names, 930
Qt configuration file, 937
Register New Functions, 931
Static Methods, 946
Threads Support, 935
Using configuration file that wrap C++ library, 937
Using the tool, 928
Wrap structures, 930
Wrapping C++ Classes, 936
CodeEditor Class
RingQt Classes Reference, 1000
Command Line Options
CGI Support, 854
Generate Object File, 855
Introduction, 840
No Run, 854
Performance, 854
Printing Final Intermediate Code, 849
Printing Instruction Operation Code, 854
Printing Intermediate Code, 847
Printing Rules, 843
Printing Tokens, 841
Comments about developing for Android using RingQt
Building RingQt Applications for Mobile, 788
Comments about evaluation
Control Structures - First Style, 235
Comments in configuration file
Code Generator, 932
Compact Syntax
Language Design, 33
Compiler and Virtual Machine (VM)
How to contribute?, 196
Compiler Errors
Reference, 989
Compiling on Manjaro Linux
What is new in Ring 1.8?, 73
Composition
Object Oriented Programming, 357
Configuration file
Code Generator, 928
Configuration file for the Allegro library
Code Generator, 933
Configuration Files Examples
Code Generator, 948
Configure the Apache web server
Web Development (CGI Library), 459
Conflict between Class Attributes and Local Variables
Scope Rules, 811
Conflict between Global Variables and Class Attributes
Frequently Asked Questions, 964
Scope Rules, 810
Conflict between self inside braces and self in the class region
Scope Rules, 817
Constants Type
Code Generator, 933
Constructor methods in Ring
Frequently Asked Questions, 960
Contact the Authors
Resources, 2039
Context Menu
Desktop and Mobile Development, 693
Control Structures
Branching, 230
Looping, 241
Control Structures - First Style
Comments about evaluation, 235
Do Again Loop, 233
Exit, 233
Exit from two loops, 233
Exit/Loop inside sub functions, 234
For In Loop, 232
for in to modify lists, 232
For Loop, 231
Introduction, 229
Loop Command, 234
Looping, 231
Short-circuit evaluation, 234
Step Option, 232
Switch Statement, 230
While Loop, 231
Control Structures - Second Style
Branching, 237
Exceptions, 239
For In Loop, 239
For Loop, 238
If Statement, 237
Introduction, 236
Looping, 238

Switch Statement, 237
Control Structures - Third Style
Branching, 240
Exceptions, 243
For In Loop, 243
For Loop, 242
If Statement, 240
Introduction, 239
Switch Statement, 240
While Loop, 241
Conversion
Data Type, 287
Conversion Class
Stdlib Classes, 429
Convert between Numbers and Bytes
What is new in Ring 1.4?, 148
Convert Letters Case
Strings, 271
Cookies
Web Development (CGI Library), 471
Copy Lists
Lists, 265
Copy()
Strings, 273
Could you explain the output of the StrCmp() function?
Frequently Asked Questions, 967
Create Database
MySQL Functions, 330
Create Executable File
Getting Started - First Style, 197
Getting Started - Second Style, 200
Getting Started - Third Style, 202
Create Lists
Lists, 261
Create Multi Dimension Array Using List and Recursion
Lists, 268
Create Package in the Current Folder
Ring Package Manager (RingPM), 879
Create Table and Insert Data
MySQL Functions, 331
Create Window
RingLibSDL, 538
Create your first extension
Introduction, 948
Location, 949
Steps to create the extension, 949
Testing the extension, 950
Create Zip File
RingZip, 521
Creating a Class for each Window in GUI applications
Scope Rules, 816
Creating Lists
Performance Tips, 858
Creating Menubar
Desktop and Mobile Development, 690
Creating more than one window
Desktop and Mobile Development, 731
Creating New Package
Ring Package Manager (RingPM), 876
Creating Reports using the WebLib and the GUILib
Desktop and Mobile Development, 783
Creating StatusBar
Desktop and Mobile Development, 695
Creating the Game Window
Game Engine for 2D Games, 557
Creating the Qt resource file using Folder2qrc
Distributing Ring Application using Ring2EXE, 871
Creating Toolbar
Desktop and Mobile Development, 694
Creating Windows Installer
Distributing Ring Application, 863
CRUD Example using MVC
Web Development (CGI Library), 493
CurrentDir() Function
System Functions, 311

Data Representation
General Information, 984
Data Type
Ascii(), 288
Char(), 288
Check Character, 283
Check Data Type, 282
Conversion, 287
Dec(), 288
Hex(), 288
Hex2str(), 289
Introduction, 281
IsAlNum(), 284
IsAlpha(), 284
IsCntrl(), 284
IsDigit(), 285
IsGraph(), 285
IsList(), 283
IsLower(), 285
IsNULL(), 283
IsNumber(), 282
IsPrint(), 285
IsPunct(), 286
IsSpace(), 286
IsString(), 282
IsUpper(), 286
IsXdigit(), 286
Number(), 287
Str2Hex(), 289
String(), 287
Type(), 283
Database, ModelBase & ControllerBase classes

Web Development (CGI Library), 501
DataType Class
Stdlib Classes, 428
Date and Time
AddDays(), 280
clock(), 278
ClosPerSecond(), 278
Date(), 278
DiffDays(), 280
EpochTime(), 280
Introduction, 277
Time(), 278
TimeList(), 279
Date()
Date and Time, 278
DateTime Class
Stdlib Classes, 423
dayofweek()
Stdlib Functions, 404
Debug Class
Stdlib Classes, 427
Dec()
Data Type, 288
Decimals()
Mathematical Functions, 294
Declarative Programming
Declarative programming on the top of Object-Oriented, 439
executing code after the end of object access, 439
Introduction, 435
More Beatiful Code, 440
Objects inside lists, 436
Return object by reference, 437
Declarative programming on the top of Object-Oriented
Declarative Programming, 439
Declare parameters
Functions - First Style, 248
Functions - Second Style, 252
Functions - Third Style, 256
Decrypt()
Security and Internet Functions, 351
Deep Copy
Variables, 225
Default value for object attributes
Object Oriented Programming, 369
Default value for object attributes is NULL
What is new in Ring 1.8?, 75
Define Declarative Languages
Language Design, 36
Define Functions
Functions - First Style, 247
Functions - Second Style, 251
Functions - Third Style, 255
Define Natural Statements
Language Design, 34
Defining Commands
Using the Natural Library, 454
Defining commands using classes
Using the Natural Library, 457
Defining Constants
Code Generator, 931
Defining Variables and Variables Access
Scope Rules, 808
Delete Item From List
Lists, 262
Deleting All Packages
Ring Package Manager (RingPM), 876
Demo Programs
Introduction, 315
Language Shell, 316
Main Menu, 316
The Cards Game, 772
Deploying after updates
Deploying Web Applications in the Cloud, 515
Deploying Web Applications in the Cloud
Application Database, 515
Deploying after updates, 515
Hello World program, 514
Introduction, 512, 513
Local Tests, 515
Ring source code files and permissions, 514
Usage, 513
Deploying Web applications in the Cloud
What is new in Ring 1.9?, 61
Designed for a clear goal
Language Design, 31
Desktop and Mobile Development
Adding Hyperlink to QLabel, 711
Change Focus, 748
Classes and their Methods to use the default events, 778
Context Menu, 693
Creating Menubar, 690
Creating more than one window, 731
Creating Reports using the WebLib and the GUILib, 783
Creating StatusBar, 695
Creating Toolbar, 694
Dialog Functions, 738
Display Image using QLabel, 715
Display Scaled Image using QLabel, 724
Drawing using QPainter, 726
Dynamic Objects, 752
Inheritance from GUI Classes, 744
KeyPress and Mouse Move Events, 739
Menubar and StyleSheet Example, 716
Methods to use Events with Events Filter, 781
Movable Label Example, 735

Moving Objects using the Mouse, 740
New Classes names - Index Start from 1, 783
Notepad Application, 757
Other Widgets Events, 720
Playing Sound, 732
Printing using QPrinter, 728
QLineEdit Events and QMessageBox, 718
QMessagebox Example, 736
QVideoWidget and QMediaPlayer, 712
Regular Expressions, 749
RingQt Classes and their Qt Documentation, 783
Rotate Text, 746
Simple Client and Server Example, 750
The Cards Game, 772
The Difference between Qt and RingQt, 782
The Events Loop, 681
The First GUI Application, 680
Using Layout, 682
Using QCheckBox, 708
Using QComboBox Class, 689
Using QDateEdit, 703
Using QDesktopWidget Class, 745
Using QDial, 704
Using QDockWidget, 696
Using QFrame, 713
Using QInputDialog Class, 737
Using qLCDNumber, 734
Using QPrintPreviewDialog, 729
Using QProgressBar, 700
Using QProgressBar and Timer, 723
Using QRadioButton and QButtonGroup, 709
Using QSlider, 701
Using QSpinBox, 701
Using QTableWidget, 699
Using QTabWidget, 697
Using QTreeView and QFileSystemModel, 687
Using QTreeWidget and QTreeWidgetItem, 688
Using QWebView, 707
Using the QColorDialog Class, 733
Using the QFileDialog Class, 725
Using the QListWidget Class, 684
Using the QTextEdit Class, 683
Using the QTimer Class, 722
Weight History Application, 753

Determine Structure Members Types
Code Generator, 931

Dialog Functions
Desktop and Mobile Development, 738

DiffDays()
Date and Time, 280

Dir()
Files, 297

direxists()
Stdlib Functions, 406

Disable BreakPoints
The Trace Library and the Interactive Debugger, 907

Discovering Packages
Ring Package Manager (RingPM), 873

Display Error Message
Extension, 917

Display Image
RingLibSDL, 538

Display Image using QLabel
Desktop and Mobile Development, 715

Display PNG Images
RingLibSDL, 539

Display Scaled Image using QLabel
Desktop and Mobile Development, 724

Display Transparent Images
RingLibSDL, 540

Display Warnings Option
What is new in Ring 1.2?, 173

Distributing Applications and Games for Mobile
Distributing Ring Application, 863

Distributing Applications for Microsoft Windows
Distributing Ring Application, 862

Distributing Applications for Mobile using RingQt
Distributing Ring Application using Ring2EXE, 867

Distributing Ring Application
Creating Windows Installer, 863
Distributing Applications and Games for Mobile, 863
Distributing Applications for Microsoft Windows, 862
Introduction, 861
Protecting the Source Code, 862
The files ring.ring and ring.ringo, 863
Using C/C++ Compiler and Linker, 863

Distributing Ring Application using Ring2EXE
Building standalone console application, 865
Building the Cards Game for Mobile using RingQt, 867
Building the Form Designer for Mobile using RingQt, 869
Building the Weight History Application for Mobile using RingQt, 868
Creating the Qt resource file using Folder2qrc, 871
Distributing Applications for Mobile using RingQt, 867
Distributing RingAllegro Applications, 866
Example, 864
How Ring2EXE works?, 864
Important Information about Ring2EXE, 871
Introduction, 863
Options, 865
Using Ring2EXE, 864

Distributing RingAllegro Applications
Distributing Ring Application using Ring2EXE, 866

Do Again Loop
Control Structures - First Style, 233
Documentation
How to contribute?, 195
Download Requirements
Building RingQt Applications for Mobile, 787
Download Requirements and Update the Android SDK
Building Games For Android, 603
Download()
Security and Internet Functions, 352
Draw Rectangle
RingLibSDL, 539
Drawing Text
Game Engine for 2D Games, 558
Drawing using QPainter
Desktop and Mobile Development, 726
Drawing using RingOpenGL
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 608
Drawing, Animation and Input
Graphics and Game Programming, 524
Dynamic Attributes
Object Oriented Programming, 363
Dynamic Objects
Desktop and Mobile Development, 752
Dynamic Typing
Variables, 225

Editors Support
How to contribute?, 196
Embedding Ring in Ring
Embedding Ring in Ring, 909
Embedding Ring in Ring without sharing the State, 910
ring_state_new() and ring_state_mainfile(), 912
ring_state_setvar(), 911
Runtime Errors when Embedding Ring in Ring, 913
Serial Execution of Programs, 910
Embedding Ring in Ring without sharing the State
Embedding Ring in Ring, 910
What is new in Ring 1.3?, 166
Embedding Ring Language in C/C++ Programs
Introduction, 924
Ring State, 925
Ring State Functions, 925
Ring State Variables, 926
Employee Application
What is new in Ring 1.6?, 89
Encourage Organization
Language Design, 33
Encrypt()
Security and Internet Functions, 350
endswith()
Stdlib Functions, 400
Entering Items
Form Designer, 802
Enum and Numbers
Code Generator, 932
Environment Errors
Reference, 991
EpochTime()
Date and Time, 280
Stdlib Functions, 408
Equality of functions
Functional Programming, 373
Eval()
Eval() and Debugging, 313
Eval() and Debugging
Assert(), 315
Eval(), 313
Introduction, 312
Raise(), 314
Try/Carch/Done, 313
evenorodd()
Stdlib Functions, 402
Events Code
Form Designer, 798
Example
Distributing Ring Application using Ring2EXE, 864
Files, 303
FoxRing Functions Reference, 1229
Mathematical Functions, 290
Natural Language Programming, 442
Objects Library for RingQt Application, 791
RingMurmurHash Functions Reference, 1215
Security and Internet Functions, 351
System Functions, 308
The Type Hints Library, 838
Example - The Trace Library
Low Level Functions, 899
Example - Using the Trace Functions
Low Level Functions, 895
Example about Sharing Names between Functions and Methods
Scope Rules for Functions and Methods, 824
Examples
bignumber, 1231
PostgreSQL, 341
Exceptions
Control Structures - Second Style, 239
Control Structures - Third Style, 243
Execute Program Line by Line
The Trace Library and the Interactive Debugger, 906
Execute Query and Print Result
ODBC Functions, 324
executing code after the end of object access
Declarative Programming, 439
Executing code during code generation

Code Generator, 932
ExeFileName() Function
System Functions, 311
ExeFolder() Function
System Functions, 311
Exit
Control Structures - First Style, 233
Exit from two loops
Control Structures - First Style, 233
Exit/Loop inside sub functions
Control Structures - First Style, 234
Extension
Check Parameters Count, 916
Check Parameters Type, 917
Display Error Message, 917
Fopen() and Fclose() Implementation, 918
Function Prototype, 918
Function Structure, 916
Get Parameters Values, 917
Introduction, 913
Module Organization, 915
MySQL_Columns() Implementation, 921
Return Value, 917
RING API - list Functions, 919
RING API - String Functions, 921
ring_ext.c, 914
ring_ext.h, 914
Shared Libraries, 922
Sin() Implementation, 918
Using RING_API_RETMANAGEDCPOINTER(), 923
Extensions in C/C++
How to contribute?, 196
Extract Zip File
RingZip, 521

Facebook Login
RingLibCurl, 517
factorial()
Stdlib Functions, 398
factors()
Stdlib Functions, 402
Fast Enough For Many Applications
Language Design, 40
Fclose()
Files, 298
Features
Introduction, 26
Ring Package Manager (RingPM), 873
Feof()
Files, 301
Ferror()
Files, 301
FetchStockData Application
Applications developed in little hours, 5
Fexists()
Files, 303
Fflush()
Files, 299
Fgetc()
Files, 302
Fgetpos()
Files, 301
Fgets()
Files, 302
fibonacci()
Stdlib Functions, 399
Fifteen Puzzle Game 2
Applications developed in little hours, 6
File Class
Stdlib Classes, 425
File Hash
Security and Internet Functions, 351
file2list()
Stdlib Functions, 400
Files
Clearerr(), 301
Dir(), 297
Example, 303
Fclose(), 298
Feof(), 301
Ferror(), 301
Fexists(), 303
Fflush(), 299
Fgetc(), 302
Fgetpos(), 301
Fgets(), 302
Fopen(), 298
Fputc(), 302
Fputs(), 302
Fread(), 303
Freopen(), 299
Fseek(), 300
Fsetpos(), 301
Ftell(), 300
Fwrite(), 303
Introduction, 295
Numbers and Bytes, 305
Perror(), 301
Read File using Read(), 297
Remove(), 298
Rename(), 298
Rewind(), 300
Tempfile(), 300
Tempname(), 300
Ungetc(), 302
Write file using Write(), 297
filter()

Stdlib Functions, 396
Filtering using Expressions
Code Generator, 932
Find in files Application
What is new in Ring 1.8?, 69
Find SubString
Strings, 274
Find() and List of Objects
Object Oriented Programming, 364
First Application using RingLibuv
RingLibuv, 545
First-Class Functions
Functional Programming, 371
First-Class Lists
Lists, 266
Flappy Bird 3000 Game
Game Engine for 2D Games, 584
Fopen()
Files, 298
Fopen() and Fclose() Implementation
Extension, 918
For In Loop
Control Structures - First Style, 232
Control Structures - Second Style, 239
Control Structures - Third Style, 243
for in to modify lists
Control Structures - First Style, 232
For Loop
Control Structures - First Style, 231
Control Structures - Second Style, 238
Control Structures - Third Style, 242
Form Designer
Entering Items, 802
Events Code, 798
Introduction, 795
Keyboard Shortcuts, 801
Menubar Designer, 801
More Samples and Tests, 803
Running Forms, 798
The Designer Windows, 797
The Properties, 797
Using Layouts, 803
What is new in Ring 1.3?, 167
Window Flags, 802
Form Designer Translation
Multi-language applications, 805
Forms Translation
Multi-language applications, 806
FoxRing functions
FoxRing Functions Reference, 1217
FoxRing Functions Reference
Example, 1229
FoxRing functions, 1217
frAbs() function, 1218
frAddBs() function, 1219
frALines() function, 1225
frAllTrim() function, 1223
frAsc() function, 1218
frAt() function, 1219
frAtC() function, 1219
frBetween() function, 1227
frChr() function, 1220
frEmpty() function, 1220
frFile() function, 1221
frFileToStr() function, 1221
frForceExt() function, 1225
frForcePath() function, 1223
frIIf() function, 1227
frInList() function, 1223
frInt() function, 1226
frJustDrive() function, 1224
frJustExt() function, 1224
frJustFName() function, 1225
frJustPath() function, 1224
frJustStem() function, 1224
frLen() function, 1225
frListToString() function, 1226
frLTrim() function, 1223
frPadL() function, 1225
frPadR() function, 1225
frProper() function, 1225
frReplicate() function, 1225
frRTrim() function, 1224
frSetIfEmpty() function, 1222
frSetSeparatorTo() function, 1228
frSpace() function, 1222
frStr() function, 1221
frStringToList() function, 1227
frStrTran() function, 1226
frStuff() function, 1226
frSubStr() function, 1226
frTransform() function, 1228
frVal() function, 1227
frVarType() function, 1228
Introduction, 1216
FoxRing Library
What is new in Ring 1.8?, 77
Fputc()
Files, 302
Fputs()
Files, 302
frAbs() function
FoxRing Functions Reference, 1218
frAddBs() function
FoxRing Functions Reference, 1219
frALines() function
FoxRing Functions Reference, 1225
frAllTrim() function

FoxRing Functions Reference, 1223
Frames Per Second
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 641
frAsc() function
FoxRing Functions Reference, 1218
frAt() function
FoxRing Functions Reference, 1219
frAtC() function
FoxRing Functions Reference, 1219
frBetween() function
FoxRing Functions Reference, 1227
frChr() function
FoxRing Functions Reference, 1220
Fread()
Files, 303
frEmpty() function
FoxRing Functions Reference, 1220
Freopen()
Files, 299
Frequently Asked Questions
Can I connect to dbase/harbour database?, 976
Can Ring work on Windows XP?, 978
Conflict between Global Variables and Class Attributes, 964
Constructor methods in Ring, 960
Could you explain the output of the StrCmp() function?, 967
Getter and Setter Methods, 962
Goal of including the Main function in Ring, 958
How can I disable maximize button and resize window?, 974
How to add Combobox and other elements to the cells of a QTableWidget?, 981
How to Close a window then displaying another one?, 973
How to create a Modal Window?, 973
How to create an array of buttons in GUI applications?, 972
How to extend RingQt and add more classes?, 978
How to get the current source file path?, 965
How to get the file size using ftell() and fseek() functions?, 965
How to insert an item to the first position in the list?, 970
How to perform some manipulations on selected cells in QTableWidget?, 981
How to print keys or values only in List/Dictionary?, 966
How to print lists that contains objects?, 970
How to print new lines and other characters?, 971
How to use many source code files in the project?, 967
How to use NULL and ISNULL() function?, 969
How to use SQLite using ODBC?, 975
Introduction, 950
Is Ring some sort of an improvement of PHP?, 953
List index start from 1, 959
Philosophy behind data types in Ring, 956
Search of global names while defining the class attributes, 963
The documentation says functional programming is supported, but then this happens?, 955
What about predefined parameters or optional parameters in functions?, 965
What about the Boolean values in Ring?, 957
What are the advantages of using Ring over native C or C++?, 953
What are the advantages to using Ring over C# or Java?, 955
What are the advantages to using Ring over Lisp or Smalltalk?, 952
What are the advantages to using Ring over Python and Ruby?, 954
What are the advantages to using Ring over Tcl and Lua?, 955
What happens when we create a new object?, 961
What is the difference between Ring and Python? And is Ring Open Source?, 953
Where can I write a program and execute it?, 964
Which of 3 coding styles are commonly used or recommended by the community?, 982
Why do we need Yet Another Programming Language (YAPL)?, 951
Why I get a strange result when printing nl with lists?, 966
Why I get Calling Function without definition Error?, 977
Why is Ring largely focussed on UI creation?, 953
Why is Ring weakly typed?, 952
Why Ring is not case-sensitive, 959
Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?, 956
Why setClickEvent() doesn't see the object methods directly?, 977
Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like?, 956
Why the Assignment operator uses Deep copy?, 960
Why the window title bar is going outside the screen?, 971
Why this example use the GetChar() twice?, 968
Why we don't use () after the qApp class name?, 971
Why you can specify the number of loops you want to break out of?, 956
frFile() function
FoxRing Functions Reference, 1221
frFileToStr() function

FoxRing Functions Reference, 1221
frForceExt() function
FoxRing Functions Reference, 1225
frForcePath() function
FoxRing Functions Reference, 1223
frIIf() function
FoxRing Functions Reference, 1227
frInList() function
FoxRing Functions Reference, 1223
frInt() function
FoxRing Functions Reference, 1226
frJustDrive() function
FoxRing Functions Reference, 1224
frJustExt() function
FoxRing Functions Reference, 1224
frJustFName() function
FoxRing Functions Reference, 1225
frJustPath() function
FoxRing Functions Reference, 1224
frJustStem() function
FoxRing Functions Reference, 1224
frLen() function
FoxRing Functions Reference, 1225
frListToString() function
FoxRing Functions Reference, 1226
frLTrim() function
FoxRing Functions Reference, 1223
frPadL() function
FoxRing Functions Reference, 1225
frPadR() function
FoxRing Functions Reference, 1225
frProper() function
FoxRing Functions Reference, 1225
frReplicate() function
FoxRing Functions Reference, 1225
frRTrim() function
FoxRing Functions Reference, 1224
frSetIfEmpty() function
FoxRing Functions Reference, 1222
frSetSeparatorTo() function
FoxRing Functions Reference, 1228
frSpace() function
FoxRing Functions Reference, 1222
frStr() function
FoxRing Functions Reference, 1221
frStringToList() function
FoxRing Functions Reference, 1227
frStrTran() function
FoxRing Functions Reference, 1226
frStuff() function
FoxRing Functions Reference, 1226
frSubStr() function
FoxRing Functions Reference, 1226
frTransform() function
FoxRing Functions Reference, 1228
frVal() function
FoxRing Functions Reference, 1227
frVarType() function
FoxRing Functions Reference, 1228
Fseek()
Files, 300
Fsetpos()
Files, 301
FSize()
Stdlib Functions, 407
Ftell()
Files, 300
Function Prototype
Extension, 918
function prototype
Code Generator, 928
Function Structure
Extension, 916
Functional Programming
Anonymous and Nested Functions, 372
Equality of functions, 373
First-Class Functions, 371
Higher-order Functions, 371
Introduction, 369
Pure Functions, 370
Functions - First Style
Call Functions, 247
Declare parameters, 248
Define Functions, 247
Introduction, 246
Main Function, 248
Recursion, 250
Return Value, 249
Send Parameters, 248
Variables Scope, 249
Functions - Second Style
Call Functions, 251
Declare parameters, 252
Define Functions, 251
Introduction, 250
Main Function, 252
Recursion, 254
Return Value, 253
Send Parameters, 252
Variables Scope, 253
Functions - Third Style
Call Functions, 255
Declare parameters, 256
Define Functions, 255
Introduction, 254
Main Function, 256
Recursion, 258
Return Value, 257

Send Parameters, 256
Variables Scope, 257
functions()
Reflection and Meta-programming, 377
Fwrite()
Files, 303

Game Class
Game Engine for 2D Games, 555
Game Engine Classes
Game Engine for 2D Games, 554
Game Engine for 2D Games
Animate Class, 557
Animate Events, 568
Animation, 562
Animation and Functions, 563
Creating the Game Window, 557
Drawing Text, 558
Flappy Bird 3000 Game, 584
Game Class, 555
Game Engine Classes, 554
GameObject Class, 555
Games Layer, 554
Graphics Library Bindings, 553
Interface to graphics library, 553
Introduction, 552
Map, 570
Map Class, 557
Map Events, 571
Moving Text, 559
Object and Drawing, 573
Playing Sound, 561
Project Layers, 553
Sound Class, 557
Sprite Automatic Movement, 564
Sprite Class, 556
Sprite Keypress Event, 565
Sprite Mouse Event, 566
Sprite State Event, 567
Stars Fighter Game, 576
Super Man 2016 Game, 591
Text Class, 556
What is new in Ring 1.1?, 181
Game of Life Game
What is new in Ring 1.10?, 44
GameObject Class
Game Engine for 2D Games, 555
Games Layer
Game Engine for 2D Games, 554
gcd()
Stdlib Functions, 401
General Information
Data Representation, 984
Introduction, 982
Memory Management, 983
Ring Architecture, 983
Generate Object File
Command Line Options, 855
Generate/Execute Ring Object Files (*.ringo)
What is new in Ring 1.1?, 176
Generating Pages using Objects
Web Development (CGI Library), 483
Get Active Source File Name
System Functions, 310
Get Command Line Arguments
System Functions, 309
Get List Item
Lists, 262
Get List Size
Lists, 262
Get Number of Characters from position
Strings, 274
Get Parameters Values
Extension, 917
Get Request
RingLibCurl, 517
Get Stock Data From Yahoo
RingLibCurl, 519
Get String Length
Strings, 271
Get Substring from position to end
Strings, 274
getattribute()
Reflection and Meta-programming, 389
GetChar()
Getting Input, 245
getnumber()
Stdlib Functions, 394
getstring()
Stdlib Functions, 394
Getter and Setter Methods
Frequently Asked Questions, 962
Getting Input
GetChar(), 245
Getting Started - First Style, 198
Getting Started - Second Style, 201
Getting Started - Third Style, 203
Give Command, 245
Input(), 246
Introduction, 244
Getting Started - First Style
Create Executable File, 197
Getting Input, 198
Hello World, 197
Introduction, 196
Multi-Line literals, 197
No Explicit End For Statements, 198
Not Case-Sensitive, 197

Run the program, 197
Using ? to print expression then new line, 198
Writing Comments, 199
Getting Started - Second Style
Create Executable File, 200
Getting Input, 201
Hello World, 200
Introduction, 199
Multi-Line literals, 200
No Explicit End For Statements, 201
Not Case-Sensitive, 200
Run the program, 200
Writing Comments, 201
Getting Started - Third Style
Create Executable File, 202
Getting Input, 203
Hello World, 202
Introduction, 201
Multi-Line literals, 203
No Explicit End For Statements, 203
Not Case-Sensitive, 202
Run the program, 202
Writing Comments, 203
Give Command
Getting Input, 245
globals()
Reflection and Meta-programming, 376
Goal of including the Main function in Ring
Frequently Asked Questions, 958
Google API Shortener Application
Applications developed in little hours, 7
Gradient
Web Development (CGI Library), 482
Graphics and Game Programming
Drawing, Animation and Input, 524
Introduction, 523
Playing Sound, 532
Scaling and Rotating Images, 533
Threads, 535
Transparent Image, 534
TrueType Fonts, 531
Graphics Library Bindings
Game Engine for 2D Games, 553

Hash Functions
Web Development (CGI Library), 477
HashTable Class
Stdlib Classes, 418
Hello World
Getting Started - First Style, 197
Getting Started - Second Style, 200
Getting Started - Third Style, 202
Hello World program
Deploying Web Applications in the Cloud, 514
Hello World Program using the Web Library
Web Development (CGI Library), 460
Hex()
Data Type, 288
Hex2str()
Data Type, 289
Higher-order Functions
Functional Programming, 371
History
Introduction, 26
Natural Language Programming, 442
How can I disable maximize button and resize window?
Frequently Asked Questions, 974
How Ring find a functions and methods?
Scope Rules for Functions and Methods, 824
How Ring find the Variable?
Scope Rules, 809
How Ring2EXE works?
Distributing Ring Application using Ring2EXE, 864
How to add Combobox and other elements to the cells of a QTableWidget?
Frequently Asked Questions, 981
How to Close a window then displaying another one?
Frequently Asked Questions, 973
How to contribute?
Applications, 196
Compiler and Virtual Machine (VM), 196
Documentation, 195
Editors Support, 196
Extensions in C/C++, 196
Ideas and suggestions, 196
Introduction, 194
Libraries in Ring, 196
Samples, 195
Special thanks to contributors, 195
Testing, 195
How to create a Modal Window?
Frequently Asked Questions, 973
How to create an array of buttons in GUI applications?
Frequently Asked Questions, 972
How to extend RingQt and add more classes?
Frequently Asked Questions, 978
How to get the current source file path?
Frequently Asked Questions, 965
How to get the file size using ftell() and fseek() functions?
Frequently Asked Questions, 965
How to insert an item to the first position in the list?
Frequently Asked Questions, 970
How to perform some manipulations on selected cells in QTableWidget?
Frequently Asked Questions, 981
How to play?
The Gold Magic 800 Game, 676
How to print keys or values only in List/Dictionary?

Frequently Asked Questions, 966
How to print lists that contains objects?
Frequently Asked Questions, 970
How to print new lines and other characters?
Frequently Asked Questions, 971
How to use many source code files in the project?
Frequently Asked Questions, 967
How to use NULL and ISNULL() function?
Frequently Asked Questions, 969
How to use SQLite using ODBC?
Frequently Asked Questions, 975
HTML Lists
Web Development (CGI Library), 480
HTML Special Characters
Web Development (CGI Library), 476
HTML Tables
Web Development (CGI Library), 481
HtmlPage Class
Web Development (CGI Library), 488, 512
HTTP Get Example
Web Development (CGI Library), 461
HTTP POST Example
Web Development (CGI Library), 466

Ideas and suggestions
How to contribute?, 196
If Statement
Control Structures - Second Style, 237
Control Structures - Third Style, 240
Ignore new lines after keywords
Syntax Flexibility, 837
Important Information about Ring2EXE
Distributing Ring Application using Ring2EXE, 871
Inheritance
Object Oriented Programming, 362
Inheritance from GUI Classes
Desktop and Mobile Development, 744
Innovative
Applications developed in little hours, 21
Input()
Getting Input, 246
Insert()
Lists, 265
Install Qt for Android
Building RingQt Applications for Mobile, 787
Installing Packages
Ring Package Manager (RingPM), 874
Interactive Debugger
The Trace Library and the Interactive Debugger, 906
Interface to graphics library
Game Engine for 2D Games, 553
Internet Class
Stdlib Classes, 435
Introduction
Deploying Web Applications in the Cloud, 513
Features, 26
History, 26
Introduction, 23
Motivation, 24
Performance Tips, 856
Ring and other languages, 25
Special Thanks, 25
Is Ring some sort of an improvement of PHP?
Frequently Asked Questions, 953
IsAlNum()
Data Type, 284
IsAlpha()
Data Type, 284
IsAndroid() Function
System Functions, 308
isattribute()
Reflection and Meta-programming, 385
iscfunction()
Reflection and Meta-programming, 379
isclass()
Reflection and Meta-programming, 381
IsCntrl()
Data Type, 284
IsDigit()
Data Type, 285
IsFreeBSD() Function
System Functions, 308
isfunction()
Reflection and Meta-programming, 378
isglobal()
Reflection and Meta-programming, 378
IsGraph()
Data Type, 285
isleapyear()
Stdlib Functions, 403
IsLinux() Function
System Functions, 308
IsList()
Data Type, 283
islocal()
Reflection and Meta-programming, 378
IsLower()
Data Type, 285
IsMacOSX() Function
System Functions, 307
ismainsourcefile()
Stdlib Functions, 406
ismethod()
Reflection and Meta-programming, 386
IsMSDOS() Function
System Functions, 307
IsNULL()
Data Type, 283

IsNumber()
Data Type, 282
isobject()
Reflection and Meta-programming, 384
ispackage()
Reflection and Meta-programming, 380
ispackagesclass()
Reflection and Meta-programming, 382
ispalindrome()
Stdlib Functions, 403
ispointer()
Low Level Functions, 886
isprime()
Stdlib Functions, 399
IsPrint()
Data Type, 285
isprivateattribute()
Reflection and Meta-programming, 385
isprivatemethod()
Reflection and Meta-programming, 386
IsPunct()
Data Type, 286
IsSpace()
Data Type, 286
isspecial()
Stdlib Functions, 397
IsString()
Data Type, 282
IsUnix() Function
System Functions, 307
IsUpper()
Data Type, 286
isvowel()
Stdlib Functions, 398
IsWindows() Function
System Functions, 307
IsWindows64() Function
System Functions, 307
IsXdigit()
Data Type, 286

JustFileName()
Stdlib Functions, 395
JustFilePath()
Stdlib Functions, 394

Keyboard Events and Colors
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 613
Keyboard Shortcuts
Form Designer, 801
KeyPress and Mouse Move Events
Desktop and Mobile Development, 739
Knight Tour Game
What is new in Ring 1.10?, 43

Language Design
Compact Syntax, 33
Define Declarative Languages, 36
Define Natural Statements, 34
Designed for a clear goal, 31
Encourage Organization, 33
Fast Enough For Many Applications, 40
Introduction, 30
No Global Interpreter (VM) Lock - No GIL, 39
Simple, 31
Smart Garbage Collector, 39
Syntax Flexibility, 37
Transparent Implementation, 37
Trying to be natural, 32
Visual Implementation, 38
Why Ring?, 31
Language Functions
Reference, 988
Language Grammar
Reference, 992
Language Keywords
Reference, 987
Language Shell
Demo Programs, 316
lcm()
Stdlib Functions, 401
Left()
Strings, 272
Libraries in Ring
How to contribute?, 196
Library Source Code
bignumber, 1233
Library Usage
Objects Library for RingQt Application, 791
linecount()
Stdlib Functions, 398
Lines()
Strings, 273
List Class
Stdlib Classes, 415
List index start from 1
Frequently Asked Questions, 959
List of changes and new features
What is new in Ring 1.10?, 41
What is new in Ring 1.1?, 175
What is new in Ring 1.2?, 169
What is new in Ring 1.3?, 156
What is new in Ring 1.4?, 144
What is new in Ring 1.5?, 100
What is new in Ring 1.6?, 89
What is new in Ring 1.7?, 81
What is new in Ring 1.8?, 68

What is new in Ring 1.9?, 51
list of functions
Mathematical Functions, 290
List2Code() Function
Stdlib Functions, 410
list2file()
Stdlib Functions, 399
ListAllFiles() Function
Stdlib Functions, 408
Lists
Access List Items by String Index, 267
Add Items, 261
Copy Lists, 265
Create Lists, 261
Create Multi Dimension Array Using List and Recursion, 268
Delete Item From List, 262
First-Class Lists, 266
Get List Item, 262
Get List Size, 262
Insert(), 265
Introduction, 260
Nested Lists, 265
Passing Lists to Functions, 266
Passing Parameters Arguments Using List Array, 268
Passing Parameters or Arguments Using List, 267
Reverse(), 264
Search, 263
Set List Item, 262
Sort(), 263
Swap Items, 269
Using Lists during definition, 266
Load Package
Program Structure, 259
Load Syntax Files
Syntax Flexibility, 829
Loading Files
Code Generator, 947
Loading the library
bignumber, 1231
PostgreSQL, 341
Loading the Trace library
The Trace Library and the Interactive Debugger, 905
Local Tests
Deploying Web Applications in the Cloud, 515
locals()
Reflection and Meta-programming, 376
Location
Create your first extension, 949
Logical Operators
Operators, 227
Loop Command
Control Structures - First Style, 234
Looping
Control Structures, 241
Control Structures - First Style, 231
Control Structures - Second Style, 238
Low Level Functions
callgc(), 884
Example - The Trace Library, 899
Example - Using the Trace Functions, 895
Introduction, 882
ispointer(), 886
nullpointer(), 885
object2pointer(), 885
pointer2object(), 885
ptrcmp(), 887
ringvm_callfunc(), 895
RingVM_CallList(), 892
RingVM_CFunctionsList(), 887
RingVM_ClassesList(), 888
ringvm_evalinscope(), 894
RingVM_FilesList(), 892
RingVM_FunctionsList(), 887
ringvm_give() function, 903
ringvm_hideerrorMsg(), 895
ringvm_info(), 903
RingVM_MemoryList(), 890
RingVM_PackagesList(), 889
ringvm_passerror(), 895
ringvm_scopescount(), 894
ringvm_see() function, 903
ringvm_settrace(), 893
ringvm_tracedata(), 893
ringvm_traceevent(), 894
ringvm_tracefunc(), 894
space(), 884
varptr(), 884

Main Function
Functions - First Style, 248
Functions - Second Style, 252
Functions - Third Style, 256
Main Menu
Demo Programs, 316
Make a Cube using RingOpenGL and RingFreeGLUT
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 650
makedir()
Stdlib Functions, 407
Managed Classes
Code Generator, 947
Many Cubes
Using RingOpenGL and RingAllegro for 3D Graphics, 658
Map
Game Engine for 2D Games, 570

Map Class
Game Engine for 2D Games, 557
Map Events
Game Engine for 2D Games, 571
map()
Stdlib Functions, 396
Math Class
Stdlib Classes, 420
Mathematical Functions
Decimals(), 294
Example, 290
Introduction, 289
list of functions, 290
Random(), 292
SRandom(), 293
Unsigned(), 293
Using _ in numbers, 294
Using f after numbers, 294
matrixmulti()
Stdlib Functions, 403
matrixtrans()
Stdlib Functions, 404
MD5()
Security and Internet Functions, 348
Memory Management
General Information, 983
Menu Events
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 626
Menubar and StyleSheet Example
Desktop and Mobile Development, 716
Menubar Designer
Form Designer, 801
Merge binary characters
Strings, 276
What is new in Ring 1.8?, 76
mergemethods()
Reflection and Meta-programming, 391
Methods to use Events with Events Filter
Desktop and Mobile Development, 781
methods()
Reflection and Meta-programming, 384
Minesweeper Game
What is new in Ring 1.10?, 42
Misc Operators
Operators, 228
Module Organization
Extension, 915
More 3D Samples
Using RingOpenGL and RingAllegro for 3D Graphics, 673
What is new in Ring 1.8?, 71
More Beatiful Code
Declarative Programming, 440
More Games
What is new in Ring 1.10?, 47
What is new in Ring 1.9?, 53
More Improvements
What is new in Ring 1.10?, 50
What is new in Ring 1.9?, 66
More Samples and Tests
Form Designer, 803
More Syntax Flexibility
What is new in Ring 1.5?, 131
Motivation
Introduction, 24
Mouse Events
RingLibSDL, 542
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 622
Movable Label Example
Desktop and Mobile Development, 735
Moving Objects using the Mouse
Desktop and Mobile Development, 740
Moving Text
Game Engine for 2D Games, 559
Multi-language applications
Form Designer Translation, 805
Forms Translation, 806
Introduction, 803
Using String2Constant, 804
Multi-Line literals
Getting Started - First Style, 197
Getting Started - Second Style, 200
Getting Started - Third Style, 203
MurmurHash1 functions
RingMurmurHash Functions Reference, 1215
MurmurHash2 functions
RingMurmurHash Functions Reference, 1215
MurmurHash3 functions
RingMurmurHash Functions Reference, 1215
MySQL Class
Stdlib Classes, 431
MySQL Functions
Create Database, 330
Create Table and Insert Data, 331
Introduction, 328
MySQL_AutoCommit(), 336
MySQL_Close(), 330
MySQL_Columns(), 333
MySQL_Commit(), 336
MySQL_Connect(), 330
MySQL_Error(), 330
MySQL_Escape_String(), 335
MySQL_Info(), 329
MySQL_Init(), 330
MySQL_Inser_ID(), 332
MySQL_Next_Result(), 333

MySQL_Query(), 330
MySQL_Result(), 333
MySQL_Result2(), 334
MySQL_Rollback(), 336
Print Query Result, 333
Restore Image From The Database, 335
Save Image Inside the Database, 335
Transaction Example, 336
MySQL_AutoCommit()
MySQL Functions, 336
MySQL_Close()
MySQL Functions, 330
MySQL_Columns()
MySQL Functions, 333
MySQL_Columns() Implementation
Extension, 921
MySQL_Commit()
MySQL Functions, 336
MySQL_Connect()
MySQL Functions, 330
MySQL_Error()
MySQL Functions, 330
MySQL_Escape_String()
MySQL Functions, 335
MySQL_Info()
MySQL Functions, 329
MySQL_Init()
MySQL Functions, 330
MySQL_Inser_ID()
MySQL Functions, 332
MySQL_Next_Result()
MySQL Functions, 333
MySQL_Query()
MySQL Functions, 330
MySQL_Result()
MySQL Functions, 333
MySQL_Result2()
MySQL Functions, 334
MySQL_Rollback()
MySQL Functions, 336

Natural Language Programming
BraceError() Method, 449
BraceExprEval Method, 448
BraceStart and BraceEnd Methods, 447
Change the '=' operator to 'is', 445
Change the Ring Keyword 'And', 443
Change the Ring Operator '+', 444
Clean Natural Code, 450
Example, 442
History, 442
Introduction, 441
Real Natual Code, 448
Using Eval() with our Natural Code, 446
Natural Library - Demo Program
Using the Natural Library, 452
Nested Lists
Lists, 265
New Classes names - Index Start from 1
Desktop and Mobile Development, 783
New Command: Load Package
What is new in Ring 1.7?, 81
New Functions
What is new in Ring 1.2?, 169
What is new in Ring 1.3?, 165
New Functions and Changes
What is new in Ring 1.1?, 178
New Game : Gold Magic 800
What is new in Ring 1.9?, 51
New Style to Ring Notepad
What is new in Ring 1.4?, 147
New Tool: Folder2qrc
What is new in Ring 1.6?, 91
New Tool: Ring2EXE
What is new in Ring 1.6?, 90
newlist()
Stdlib Functions, 397
No Explicit End For Statements
Getting Started - First Style, 198
Getting Started - Second Style, 201
Getting Started - Third Style, 203
No Global Interpreter (VM) Lock - No GIL
Language Design, 39
No Run
Command Line Options, 854
Not Case-Sensitive
Getting Started - First Style, 197
Getting Started - Second Style, 200
Getting Started - Third Style, 202
Notepad Application
Desktop and Mobile Development, 757
Notes to extensions creators
What is new in Ring 1.8?, 80
nullpointer()
Low Level Functions, 885
Number()
Data Type, 287
Numbers and Bytes
Files, 305

Object and Drawing
Game Engine for 2D Games, 573
Object Library Source Code
Objects Library for RingQt Application, 795
Object Oriented Programming
Access Objects Using Braces, 356
Classes and Objects, 354
Composition, 357

Default value for object attributes, 369
Dynamic Attributes, 363
Find() and List of Objects, 364
Inheritance, 362
Introduction, 353
Operator Overloading, 359
Packages, 363
Printing Objects, 364
Private Attributes and Methods, 358
Setter and Getter, 358
Sort() and List of Objects, 365
Using Self.Attribute, 367
Using This in the class region as Self, 368
Using This.Attribute and This.Method(), 367
object2pointer()
Low Level Functions, 885
objectid()
Reflection and Meta-programming, 383
Objects inside lists
Declarative Programming, 436
Objects Library for RingQt
What is new in Ring 1.2?, 171
Objects Library for RingQt Application
Example, 791
Introduction, 790
Library Usage, 791
Object Library Source Code, 795
Open_WindowAndLink() Function, 794, 795
Open_WindowInPackages() Function, 795
ODBC Class
Stdlib Classes, 430
ODBC Functions
Execute Query and Print Result, 324
Introduction, 319
odbc_autocommit(), 326
odbc_close(), 321
odbc_colcount(), 323
odbc_columns(), 325
odbc_commit(), 326
odbc_connect(), 322
odbc_datasources(), 321
odbc_disconnect(), 323
odbc_drivers(), 320
odbc_execute(), 323
odbc_fetch(), 324
odbc_getdata(), 324
odbc_init(), 320
odbc_rollback(), 326
odbc_tables(), 324
Open and Close Connection, 323
Print List of ODBC Data Sources, 322
Print List of ODBC Drivers, 321
Save and Restore Images, 327
Transactions and Using Commit and Rollback, 326
odbc_autocommit()
ODBC Functions, 326
odbc_close()
ODBC Functions, 321
odbc_colcount()
ODBC Functions, 323
odbc_columns()
ODBC Functions, 325
odbc_commit()
ODBC Functions, 326
odbc_connect()
ODBC Functions, 322
odbc_datasources()
ODBC Functions, 321
odbc_disconnect()
ODBC Functions, 323
odbc_drivers()
ODBC Functions, 320
odbc_execute()
ODBC Functions, 323
odbc_fetch()
ODBC Functions, 324
odbc_getdata()
ODBC Functions, 324
odbc_init()
ODBC Functions, 320
odbc_rollback()
ODBC Functions, 326
odbc_tables()
ODBC Functions, 324
Open and Close Connection
ODBC Functions, 323
Open_WindowAndLink() Function
Objects Library for RingQt Application, 794, 795
Open_WindowInPackages() Function
Objects Library for RingQt Application, 795
Operator Overloading
Object Oriented Programming, 359
Operators
Arithmetic Operators, 227
Assignment Operators, 228
Bitwise Operators, 228
Introduction, 226
Logical Operators, 227
Misc Operators, 228
Operators Precedence, 228
Relational Operators, 227
Using the Natural Library, 456
Operators Precedence
Operators, 228
Options
Distributing Ring Application using Ring2EXE, 865
OSCopyFile() Function
Stdlib Functions, 409

OSCopyFolder() Function
Stdlib Functions, 409
OSCreateOpenFolder() Function
Stdlib Functions, 409
OSDeleteFile() Function
Stdlib Functions, 410
OSDeleteFolder() Function
Stdlib Functions, 409
OSRenameFile() Function
Stdlib Functions, 410
Other Widgets Events
Desktop and Mobile Development, 720

packagename()
Reflection and Meta-programming, 392
Packages
Object Oriented Programming, 363
packages()
Reflection and Meta-programming, 379
packagesclasses()
Reflection and Meta-programming, 382
Page Class
Web Development (CGI Library), 508
Pass Error
The Trace Library and the Interactive Debugger, 906
Passing Lists to Functions
Lists, 266
Passing Parameters Arguments Using List Array
Lists, 268
Passing Parameters or Arguments Using List
Lists, 267
Performance
Command Line Options, 854
Performance Tips
Arithmetic Operations, 859
Calling Functions and Methods, 860
Creating Lists, 858
Introduction, 855, 856
Using len() and For Loops, 859
permutation()
Stdlib Functions, 404
Perror()
Files, 301
Philosophy behind data types in Ring
Frequently Asked Questions, 956
Play Sound
RingLibSDL, 543
Playing Sound
Desktop and Mobile Development, 732
Game Engine for 2D Games, 561
Graphics and Game Programming, 532
pointer2object()
Low Level Functions, 885
Pong Game
What is new in Ring 1.10?, 45
Post Request
RingLibCurl, 517
PostgreSQL
Examples, 341
Introduction, 340
Loading the library, 341
RingPostgreSQL Constants, 344
RingPostgreSQL Functions, 345
PostgreSQL Class
Stdlib Classes, 433
Practical
Applications developed in little hours, 22
Prefix for Functions Names
Code Generator, 930
PrevFileName() Function
System Functions, 310
Print files in Zip file
RingZip, 521
Print List of ODBC Data Sources
ODBC Functions, 322
Print List of ODBC Drivers
ODBC Functions, 321
Print Query Result
MySQL Functions, 333
print()
Stdlib Functions, 393
Print2Str() Function
Stdlib Functions, 393
Printing Final Intermediate Code
Command Line Options, 849
Printing Instruction Operation Code
Command Line Options, 854
Printing Intermediate Code
Command Line Options, 847
Printing List of Installed Packages
Ring Package Manager (RingPM), 875
Printing Objects
Object Oriented Programming, 364
Printing Rules
Command Line Options, 843
Printing Tokens
Command Line Options, 841
Printing using QPrinter
Desktop and Mobile Development, 728
Private Attributes and Methods
Object Oriented Programming, 358
prodlist()
Stdlib Functions, 402
Program Structure
Introduction, 258
Load Package, 259
Source Code File Sections, 259
Using Many Source Code Files, 259

Project Folder
Building Games For Android, 603
Project Layers
Game Engine for 2D Games, 553
Protecting the Source Code
Distributing Ring Application, 862
ptrcmp()
Low Level Functions, 887
Pure Functions
Functional Programming, 370
puts()
Stdlib Functions, 393

QAbstractButton Class
RingQt Classes Reference, 1000
QAbstractItemView Class
RingQt Classes Reference, 1001
QAbstractPrintDialog Class
RingQt Classes Reference, 1003
QAbstractScrollArea Class
RingQt Classes Reference, 1003
QAbstractSlider Class
RingQt Classes Reference, 1004
QAbstractSocket Class
RingQt Classes Reference, 1005
QAbstractSpinBox Class
RingQt Classes Reference, 1006
QAction Class
RingQt Classes Reference, 1007
QAllEvents Class
RingQt Classes Reference, 1008
QApp Class
RingQt Classes Reference, 1013
QAxBase Class
RingQt Classes Reference, 1013
QAxObject Class
RingQt Classes Reference, 1015
QAxWidget Class
RingQt Classes Reference, 1015
QBitmap Class
RingQt Classes Reference, 1015
QBluetoothAddress Class
RingQt Classes Reference, 1015
QBluetoothDeviceDiscoveryAgent Class
RingQt Classes Reference, 1015
QBluetoothDeviceInfo Class
RingQt Classes Reference, 1016
QBluetoothHostInfo Class
RingQt Classes Reference, 1016
QBluetoothLocalDevice Class
RingQt Classes Reference, 1017
QBluetoothServer Class
RingQt Classes Reference, 1018
QBluetoothServiceDiscoveryAgent Class
RingQt Classes Reference, 1018
QBluetoothServiceInfo Class
RingQt Classes Reference, 1019
QBluetoothSocket Class
RingQt Classes Reference, 1020
QBluetoothTransferManager Class
RingQt Classes Reference, 1021
QBluetoothTransferReply Class
RingQt Classes Reference, 1021
QBluetoothTransferRequest Class
RingQt Classes Reference, 1021
QBluetoothUuid Class
RingQt Classes Reference, 1021
QBoxLayout Class
RingQt Classes Reference, 1022
QBrush Class
RingQt Classes Reference, 1022
QBuffer Class
RingQt Classes Reference, 1023
QButtonGroup Class
RingQt Classes Reference, 1023
QByteArray Class
RingQt Classes Reference, 1024
QCalendarWidget Class
RingQt Classes Reference, 1026
QCamera Class
RingQt Classes Reference, 1027
QCameraImageCapture Class
RingQt Classes Reference, 1028
QCameraViewfinder Class
RingQt Classes Reference, 1029
QChar Class
RingQt Classes Reference, 1029
QCheckBox Class
RingQt Classes Reference, 1031
QChildEvent Class
RingQt Classes Reference, 1032
QClipboard Class
RingQt Classes Reference, 1032
QColor Class
RingQt Classes Reference, 1032
QColorDialog Class
RingQt Classes Reference, 1035
QComboBox Class
RingQt Classes Reference, 1035
QCompleter Class
RingQt Classes Reference, 1037
QCompleter2 Class
RingQt Classes Reference, 1038
QCompleter3 Class
RingQt Classes Reference, 1038
QCoreApplication Class
RingQt Classes Reference, 1038
QCursor Class

RingQt Classes Reference, 1039
QDate Class
RingQt Classes Reference, 1040
QDateEdit Class
RingQt Classes Reference, 1041
QDateTime Class
RingQt Classes Reference, 1041
QDateTimeEdit Class
RingQt Classes Reference, 1042
QDesktopServices Class
RingQt Classes Reference, 1043
QDesktopWidget Class
RingQt Classes Reference, 1044
QDial Class
RingQt Classes Reference, 1044
QDialog Class
RingQt Classes Reference, 1045
QDir Class
RingQt Classes Reference, 1045
QDirModel Class
RingQt Classes Reference, 1047
QDockWidget Class
RingQt Classes Reference, 1048
QDrag Class
RingQt Classes Reference, 1048
QDragEnterEvent Class
RingQt Classes Reference, 1049
QDragLeaveEvent Class
RingQt Classes Reference, 1049
QDragMoveEvent Class
RingQt Classes Reference, 1049
QDropEvent Class
RingQt Classes Reference, 1050
QEvent Class
RingQt Classes Reference, 1050
QFile Class
RingQt Classes Reference, 1051
QFile2 Class
RingQt Classes Reference, 1051
QFileDevice Class
RingQt Classes Reference, 1051
QFileDialog Class
RingQt Classes Reference, 1052
QFileInfo Class
RingQt Classes Reference, 1053
QFileSystemModel Class
RingQt Classes Reference, 1055
QFont Class
RingQt Classes Reference, 1056
QFontDialog Class
RingQt Classes Reference, 1058
QFontMetrics Class
RingQt Classes Reference, 1058
QFrame Class
RingQt Classes Reference, 1059
QFrame2 Class
RingQt Classes Reference, 1060
QFrame3 Class
RingQt Classes Reference, 1060
QGeoAddress Class
RingQt Classes Reference, 1060
QGeoAreaMonitorInfo Class
RingQt Classes Reference, 1061
QGeoAreaMonitorSource Class
RingQt Classes Reference, 1061
QGeoCircle Class
RingQt Classes Reference, 1061
QGeoCoordinate Class
RingQt Classes Reference, 1062
QGeoPositionInfo Class
RingQt Classes Reference, 1062
QGeoPositionInfoSource Class
RingQt Classes Reference, 1062
QGeoRectangle Class
RingQt Classes Reference, 1063
QGeoSatelliteInfo Class
RingQt Classes Reference, 1064
QGeoSatelliteInfoSource Class
RingQt Classes Reference, 1064
QGeoShape Class
RingQt Classes Reference, 1064
QGradient Class
RingQt Classes Reference, 1064
QGraphicsVideoItem Class
RingQt Classes Reference, 1065
QGridLayout Class
RingQt Classes Reference, 1065
QGuiApplication Class
RingQt Classes Reference, 1066
QHBoxLayout Class
RingQt Classes Reference, 1068
QHeaderView Class
RingQt Classes Reference, 1068
QHostAddress Class
RingQt Classes Reference, 1071
QHostInfo Class
RingQt Classes Reference, 1071
QIcon Class
RingQt Classes Reference, 1072
QImage Class
RingQt Classes Reference, 1073
QInputDialog Class
RingQt Classes Reference, 1074
QIODevice Class
RingQt Classes Reference, 1072
QJsonArray Class
RingQt Classes Reference, 1075
QJsonDocument Class

RingQt Classes Reference, 1076
QJsonObject Class
RingQt Classes Reference, 1077
QJsonParseError Class
RingQt Classes Reference, 1077
QJsonValue Class
RingQt Classes Reference, 1077
QKeySequence Class
RingQt Classes Reference, 1078
QLabel Class
RingQt Classes Reference, 1079
QLayout Class
RingQt Classes Reference, 1080
QLCDNumber Class
RingQt Classes Reference, 1078
QLinearGradient Class
RingQt Classes Reference, 1083
QLineEdit Class
RingQt Classes Reference, 1081
QLineEdit Events and QMessageBox
Desktop and Mobile Development, 718
QListView Class
RingQt Classes Reference, 1083
QListWidget Class
RingQt Classes Reference, 1084
QListWidgetItem Class
RingQt Classes Reference, 1085
QMainWindow Class
RingQt Classes Reference, 1086
QMatrix4x4 Class
RingQt Classes Reference, 1088
QMdiArea Class
RingQt Classes Reference, 1089
QMdiSubWindow Class
RingQt Classes Reference, 1090
QMediaObject Class
RingQt Classes Reference, 1090
QMediaPlayer Class
RingQt Classes Reference, 1091
QMediaPlaylist Class
RingQt Classes Reference, 1092
QMenu Class
RingQt Classes Reference, 1092
QMenuBar Class
RingQt Classes Reference, 1093
QMessageBox Class
RingQt Classes Reference, 1094
QMessagebox Example
Desktop and Mobile Development, 736
QMimeData Class
RingQt Classes Reference, 1095
QMutex Class
RingQt Classes Reference, 1095
QMutexLocker Class
RingQt Classes Reference, 1096
QNetworkAccessManager Class
RingQt Classes Reference, 1096
QNetworkProxy Class
RingQt Classes Reference, 1097
QNetworkReply Class
RingQt Classes Reference, 1097
QNetworkRequest Class
RingQt Classes Reference, 1098
QNmeaPositionInfoSource Class
RingQt Classes Reference, 1098
QObject Class
RingQt Classes Reference, 1099
QOpenGLBuffer Class
RingQt Classes Reference, 1100
QOpenGLContext Class
RingQt Classes Reference, 1100
QOpenGLDebugLogger Class
RingQt Classes Reference, 1101
QOpenGLFramebufferObject Class
RingQt Classes Reference, 1102
QOpenGLFunctions Class
RingQt Classes Reference, 1102
QOpenGLFunctions_3_2_Core Class
RingQt Classes Reference, 1106
QOpenGLPaintDevice Class
RingQt Classes Reference, 1114
QOpenGLShader Class
RingQt Classes Reference, 1115
QOpenGLShaderProgram Class
RingQt Classes Reference, 1115
QOpenGLTexture Class
RingQt Classes Reference, 1119
QOpenGLTimerQuery Class
RingQt Classes Reference, 1121
QOpenGLVersionProfile Class
RingQt Classes Reference, 1121
QOpenGLVertexArrayObject Class
RingQt Classes Reference, 1122
QOpenGLWidget Class
RingQt Classes Reference, 1122
QPageSetupDialog Class
RingQt Classes Reference, 1123
QPaintDevice Class
RingQt Classes Reference, 1123
QPainter Class
RingQt Classes Reference, 1123
QPainter2 Class
RingQt Classes Reference, 1126
QPainterPath Class
RingQt Classes Reference, 1126
QPen Class
RingQt Classes Reference, 1127
QPicture Class

RingQt Classes Reference, 1128
QPixmap Class
RingQt Classes Reference, 1128
QPixmap2 Class
RingQt Classes Reference, 1130
QPlainTextEdit Class
RingQt Classes Reference, 1130
QPoint Class
RingQt Classes Reference, 1132
QPointF Class
RingQt Classes Reference, 1133
QPrinter Class
RingQt Classes Reference, 1134
QPrinterInfo Class
RingQt Classes Reference, 1135
QPrintPreviewDialog Class
RingQt Classes Reference, 1133
QPrintPreviewWidget Class
RingQt Classes Reference, 1133
QProcess Class
RingQt Classes Reference, 1136
QProgressBar Class
RingQt Classes Reference, 1137
QPushButton Class
RingQt Classes Reference, 1138
QQmlError Class
RingQt Classes Reference, 1138
QQuaternion Class
RingQt Classes Reference, 1139
QQuickWidget Class
RingQt Classes Reference, 1139
QRadioButton Class
RingQt Classes Reference, 1140
QRect Class
RingQt Classes Reference, 1140
QRegion Class
RingQt Classes Reference, 1142
QRegularExpression Class
RingQt Classes Reference, 1143
QRegularExpressionMatch Class
RingQt Classes Reference, 1143
QRegularExpressionMatchIterator Class
RingQt Classes Reference, 1144
QScreen Class
RingQt Classes Reference, 1144
QScrollArea Class
RingQt Classes Reference, 1145
QScrollBar Class
RingQt Classes Reference, 1146
QSerialPort Class
RingQt Classes Reference, 1146
QSerialPortInfo Class
RingQt Classes Reference, 1147
QSize Class
RingQt Classes Reference, 1147
QSlider Class
RingQt Classes Reference, 1148
QSpinBox Class
RingQt Classes Reference, 1148
QSplashScreen Class
RingQt Classes Reference, 1149
QSplitter Class
RingQt Classes Reference, 1149
QSqlDatabase Class
RingQt Classes Reference, 1150
QSqlDriver Class
RingQt Classes Reference, 1151
QSqlDriverCreatorBase Class
RingQt Classes Reference, 1151
QSqlError Class
RingQt Classes Reference, 1152
QSqlField Class
RingQt Classes Reference, 1152
QSqlIndex Class
RingQt Classes Reference, 1153
QSqlQuery Class
RingQt Classes Reference, 1153
QSqlRecord Class
RingQt Classes Reference, 1154
QStackedWidget Class
RingQt Classes Reference, 1155
QStandardPaths Class
RingQt Classes Reference, 1155
QStatusBar Class
RingQt Classes Reference, 1156
QString2 Class
RingQt Classes Reference, 1156
QStringList Class
RingQt Classes Reference, 1157
QStringRef Class
RingQt Classes Reference, 1158
QSurfaceFormat Class
RingQt Classes Reference, 1160
QSystemTrayIcon Class
RingQt Classes Reference, 1161
Qt Class Convertor
What is new in Ring 1.4?, 151
Qt configuration file
Code Generator, 937
QTabBar Class
RingQt Classes Reference, 1161
QTableView Class
RingQt Classes Reference, 1164
QTableWidget Class
RingQt Classes Reference, 1165
QTableWidgetItem Class
RingQt Classes Reference, 1168
QTabWidget Class

RingQt Classes Reference, 1163
QTcpServer Class
RingQt Classes Reference, 1169
QTcpSocket Class
RingQt Classes Reference, 1170
QTest Class
RingQt Classes Reference, 1170
QTextBlock Class
RingQt Classes Reference, 1170
QTextBrowser Class
RingQt Classes Reference, 1171
QTextCharFormat Class
RingQt Classes Reference, 1172
QTextCodec Class
RingQt Classes Reference, 1174
QTextCursor Class
RingQt Classes Reference, 1174
QTextDocument Class
RingQt Classes Reference, 1176
QTextEdit Class
RingQt Classes Reference, 1178
QTextStream Class
RingQt Classes Reference, 1180
QTextStream2 Class
RingQt Classes Reference, 1182
QTextStream3 Class
RingQt Classes Reference, 1182
QTextStream4 Class
RingQt Classes Reference, 1182
QTextStream5 Class
RingQt Classes Reference, 1182
QThread Class
RingQt Classes Reference, 1182
QThreadPool Class
RingQt Classes Reference, 1183
QTime Class
RingQt Classes Reference, 1184
QTimer Class
RingQt Classes Reference, 1184
QToolBar Class
RingQt Classes Reference, 1185
QToolButton Class
RingQt Classes Reference, 1185
QTreeView Class
RingQt Classes Reference, 1186
QTreeWidget Class
RingQt Classes Reference, 1188
QTreeWidgetItem Class
RingQt Classes Reference, 1190
Queue Class
Stdlib Classes, 417
Quotes about Ring
Applications developed in little hours, 1
QUrl Class
RingQt Classes Reference, 1192
QUuid Class
RingQt Classes Reference, 1193
QVariant Class
RingQt Classes Reference, 1194
QVariant2 Class
RingQt Classes Reference, 1195
QVariant3 Class
RingQt Classes Reference, 1195
QVariant4 Class
RingQt Classes Reference, 1195
QVariant5 Class
RingQt Classes Reference, 1195
QVBoxLayout Class
RingQt Classes Reference, 1193
QVector2D Class
RingQt Classes Reference, 1195
QVector3D Class
RingQt Classes Reference, 1196
QVector4D Class
RingQt Classes Reference, 1197
QVideoWidget and QMediaPlayer
Desktop and Mobile Development, 712
QVideoWidget Class
RingQt Classes Reference, 1197
QVideoWidgetControl Class
RingQt Classes Reference, 1198
QWebEngineView Class
RingQt Classes Reference, 1198
QWebView Class
RingQt Classes Reference, 1199, 1200
QWidget Class
RingQt Classes Reference, 1200
QWindow Class
RingQt Classes Reference, 1206
QXmlStreamAttribute Class
RingQt Classes Reference, 1210
QXmlStreamAttributes Class
RingQt Classes Reference, 1210
QXmlStreamEntityDeclaration Class
RingQt Classes Reference, 1211
QXmlStreamEntityResolver Class
RingQt Classes Reference, 1211
QXmlStreamNamespaceDeclaration Class
RingQt Classes Reference, 1211
QXmlStreamNotationDeclaration Class
RingQt Classes Reference, 1211
QXmlStreamReader Class
RingQt Classes Reference, 1211
QXmlStreamWriter Class
RingQt Classes Reference, 1213

Raise()
Eval() and Debugging, 314

RandBytes()
Security and Internet Functions, 352
Random Image
Web Development (CGI Library), 479
Random()
Mathematical Functions, 292
Read File using Read()
Files, 297
readline()
Stdlib Functions, 405
Real Natual Code
Natural Language Programming, 448
Recursion
Functions - First Style, 250
Functions - Second Style, 254
Functions - Third Style, 258
Reference
Compiler Errors, 989
Environment Errors, 991
Introduction, 986
Language Functions, 988
Language Grammar, 992
Language Keywords, 987
Runtime Errors, 990
Virtual Machine Instructions, 993
Reflection and Meta-programming
addattribute(), 387
addmethod(), 387
attributes(), 384
cfunctions(), 377
classes(), 381
classname(), 383
functions(), 377
getattribute(), 389
globals(), 376
Introduction, 374
isattribute(), 385
iscfunction(), 379
isclass(), 381
isfunction(), 378
isglobal(), 378
islocal(), 378
ismethod(), 386
isobject(), 384
ispackage(), 380
ispackagesclass(), 382
isprivateattribute(), 385
isprivatemethod(), 386
locals(), 376
mergemethods(), 391
methods(), 384
objectid(), 383
packagename(), 392
packages(), 379
packagesclasses(), 382
setattribute(), 390
Register New Functions
Code Generator, 931
Regular Expressions
Desktop and Mobile Development, 749
Relational Operators
Operators, 227
Remove Package
Ring Package Manager (RingPM), 876
Remove()
Files, 298
Rename()
Files, 298
Resources
Arabic Resources, 2039
Contact the Authors, 2039
Introduction, 2038
Ring Group, 2039
Ring Language Website, 2039
Source Code, 2039
Restore Image From The Database
MySQL Functions, 335
Return object by reference
Declarative Programming, 437
Return Self by Reference
What is new in Ring 1.3?, 165
Return Value
Extension, 917
Functions - First Style, 249
Functions - Second Style, 253
Functions - Third Style, 257
Reverse()
Lists, 264
Rewind()
Files, 300
Right()
Strings, 272
Ring and other languages
Introduction, 25
RING API - list Functions
Extension, 919
RING API - String Functions
Extension, 921
Ring Architecture
General Information, 983
Ring CGI Hello World Program
Web Development (CGI Library), 460
Ring Extension for Visual Studio Code
What is new in Ring 1.10?, 48
Ring Group
Resources, 2039
Ring Language Website
Resources, 2039

Ring mode for Emacs Editor
What is new in Ring 1.3?, 163
Ring Notepad
Introduction, 204
Ring Notepad - Creating and running your first console application, 205
Ring Notepad - Creating and running your first game, 210
Ring Notepad - Creating and running your first GUI application, 208
Ring Notepad - Creating and running your first Web application, 209
Ring Notepad - Main Window, 205
The Browser Menu, 215
The Distribute Menu, 216
The Edit Menu, 213
The File Menu, 212
The Help Menu, 216
The Main File in the Project, 211
The Program Menu, 215
The Tools Menu, 216
The View Menu, 213
Ring Notepad - Creating and running your first console application
Ring Notepad, 205
Ring Notepad - Creating and running your first game
Ring Notepad, 210
Ring Notepad - Creating and running your first GUI application
Ring Notepad, 208
Ring Notepad - Creating and running your first Web application
Ring Notepad, 209
Ring Notepad - Main Window
Ring Notepad, 205
Ring Package Manager (RingPM)
Create Package in the Current Folder, 879
Creating New Package, 876
Deleting All Packages, 876
Discovering Packages, 873
Features, 873
Installing Packages, 874
Introduction, 872
Printing List of Installed Packages, 875
Remove Package, 876
Run Package, 875
The Package Description File, 878
The RingPM Registry File, 881
Update Package, 875
Updating the RingPM Registry, 874
Ring source code files and permissions
Deploying Web Applications in the Cloud, 514
Ring State
Embedding Ring Language in C/C++ Programs, 925
Ring State Functions
Embedding Ring Language in C/C++ Programs, 925
Ring State Variables
Embedding Ring Language in C/C++ Programs, 926
Ring VM - Tracing Functions
What is new in Ring 1.5?, 125
ring_ext.c
Extension, 914
ring_ext.h
Extension, 914
ring_state_new() and ring_state_mainfile()
Embedding Ring in Ring, 912
ring_state_new() and ring_state_mainfile() functions
What is new in Ring 1.7?, 83
ring_state_setvar()
Embedding Ring in Ring, 911
RingAllegro Functions Reference
Introduction, 1245
RingCodeHighlighter Class
RingQt Classes Reference, 1214
RingConsoleColors Extension
What is new in Ring 1.6?, 92
RingConsoleColors Functions Reference
Introduction, 1243
RingFreeGLUT Extension
What is new in Ring 1.5?, 111
RingFreeGLUT Functions Reference
Introduction, 1298
RingLibCurl
Facebook Login, 517
Get Request, 517
Get Stock Data From Yahoo, 519
Introduction, 516
Post Request, 517
Save output to string, 518
What is new in Ring 1.2?, 172
RingLibCurl Functions Reference
Introduction, 1233
RingLibSDL
Close Window Event, 541
Create Window, 538
Display Image, 538
Display PNG Images, 539
Display Transparent Images, 540
Draw Rectangle, 539
Introduction, 537
Mouse Events, 542
Play Sound, 543
Switch Between Two Images, 538
Use TTF Fonts, 540
What is new in Ring 1.1?, 181
RingLibSDL Functions Reference
Introduction, 1264
RingLibuv

Client Example, 547
Client Example - Using Classes, 550
First Application using RingLibuv, 545
Introduction, 544
Server Example, 546
Server Example using Classes, 549
The Events Loop, 546
Threads Example, 551
Threads Example - Using Classes, 552
RingLibuv Extension
What is new in Ring 1.7?, 86
RingLibuv Functions Reference
Introduction, 1289
RingLibZip Functions Reference
Introduction, 1242
RingMurmurHash Extension
What is new in Ring 1.6?, 92
RingMurmurHash Functions Reference
Example, 1215
Introduction, 1214
MurmurHash1 functions, 1215
MurmurHash2 functions, 1215
MurmurHash3 functions, 1215
RingOpenGL (OpenGL 1.1) Functions Reference
Introduction, 1307
RingOpenGL (OpenGL 1.2) Functions Reference
Introduction, 1332
RingOpenGL (OpenGL 1.3) Functions Reference
Introduction, 1358
RingOpenGL (OpenGL 1.4) Functions Reference
Introduction, 1387
RingOpenGL (OpenGL 1.5) Functions Reference
Introduction, 1417
RingOpenGL (OpenGL 2.0) Functions Reference
Introduction, 1448
RingOpenGL (OpenGL 2.1) Functions Reference
Introduction, 1482
RingOpenGL (OpenGL 3.0) Functions Reference
Introduction, 1524
RingOpenGL (OpenGL 3.1) Functions Reference
Introduction, 1569
RingOpenGL (OpenGL 3.2) Functions Reference
Introduction, 1615
RingOpenGL (OpenGL 3.3) Functions Reference
Introduction, 1662
RingOpenGL (OpenGL 4.0) Functions Reference
Introduction, 1708
RingOpenGL (OpenGL 4.1) Functions Reference
Introduction, 1755
RingOpenGL (OpenGL 4.2) Functions Reference
Introduction, 1802
RingOpenGL (OpenGL 4.3) Functions Reference
Introduction, 1849
RingOpenGL (OpenGL 4.4) Functions Reference
Introduction, 1896
RingOpenGL (OpenGL 4.5) Functions Reference
Introduction, 1943
RingOpenGL (OpenGL 4.6) Functions Reference
Introduction, 1990
RingOpenGL Extension
What is new in Ring 1.5?, 120
RingPostgreSQL Constants
PostgreSQL, 344
RingPostgreSQL Extension
What is new in Ring 1.9?, 59
RingPostgreSQL Functions
PostgreSQL, 345
RingQt Classes and their Qt Documentation
Desktop and Mobile Development, 783
RingQt Classes Reference
CodeEditor Class, 1000
Introduction, 999
QAbstractButton Class, 1000
QAbstractItemView Class, 1001
QAbstractPrintDialog Class, 1003
QAbstractScrollArea Class, 1003
QAbstractSlider Class, 1004
QAbstractSocket Class, 1005
QAbstractSpinBox Class, 1006
QAction Class, 1007
QAllEvents Class, 1008
QApp Class, 1013
QAxBase Class, 1013
QAxObject Class, 1015
QAxWidget Class, 1015
QBitmap Class, 1015
QBluetoothAddress Class, 1015
QBluetoothDeviceDiscoveryAgent Class, 1015
QBluetoothDeviceInfo Class, 1016
QBluetoothHostInfo Class, 1016
QBluetoothLocalDevice Class, 1017
QBluetoothServer Class, 1018
QBluetoothServiceDiscoveryAgent Class, 1018
QBluetoothServiceInfo Class, 1019
QBluetoothSocket Class, 1020
QBluetoothTransferManager Class, 1021
QBluetoothTransferReply Class, 1021
QBluetoothTransferRequest Class, 1021
QBluetoothUuid Class, 1021
QBoxLayout Class, 1022
QBrush Class, 1022
QBuffer Class, 1023
QButtonGroup Class, 1023
QByteArray Class, 1024
QCalendarWidget Class, 1026
QCamera Class, 1027
QCameraImageCapture Class, 1028
QCameraViewfinder Class, 1029

QChar Class, 1029
QCheckBox Class, 1031
QChildEvent Class, 1032
QClipboard Class, 1032
QColor Class, 1032
QColorDialog Class, 1035
QComboBox Class, 1035
QCompleter Class, 1037
QCompleter2 Class, 1038
QCompleter3 Class, 1038
QCoreApplication Class, 1038
QCursor Class, 1039
QDate Class, 1040
QDateEdit Class, 1041
QDateTime Class, 1041
QDateTimeEdit Class, 1042
QDesktopServices Class, 1043
QDesktopWidget Class, 1044
QDial Class, 1044
QDialog Class, 1045
QDir Class, 1045
QDirModel Class, 1047
QDockWidget Class, 1048
QDrag Class, 1048
QDragEnterEvent Class, 1049
QDragLeaveEvent Class, 1049
QDragMoveEvent Class, 1049
QDropEvent Class, 1050
QEvent Class, 1050
QFile Class, 1051
QFile2 Class, 1051
QFileDevice Class, 1051
QFileDialog Class, 1052
QFileInfo Class, 1053
QFileSystemModel Class, 1055
QFont Class, 1056
QFontDialog Class, 1058
QFontMetrics Class, 1058
QFrame Class, 1059
QFrame2 Class, 1060
QFrame3 Class, 1060
QGeoAddress Class, 1060
QGeoAreaMonitorInfo Class, 1061
QGeoAreaMonitorSource Class, 1061
QGeoCircle Class, 1061
QGeoCoordinate Class, 1062
QGeoPositionInfo Class, 1062
QGeoPositionInfoSource Class, 1062
QGeoRectangle Class, 1063
QGeoSatelliteInfo Class, 1064
QGeoSatelliteInfoSource Class, 1064
QGeoShape Class, 1064
QGradient Class, 1064
QGraphicsVideoItem Class, 1065
QGridLayout Class, 1065
QGuiApplication Class, 1066
QHBoxLayout Class, 1068
QHeaderView Class, 1068
QHostAddress Class, 1071
QHostInfo Class, 1071
QIcon Class, 1072
QImage Class, 1073
QInputDialog Class, 1074
QIODevice Class, 1072
QJsonArray Class, 1075
QJsonDocument Class, 1076
QJsonObject Class, 1077
QJsonParseError Class, 1077
QJsonValue Class, 1077
QKeySequence Class, 1078
QLabel Class, 1079
QLayout Class, 1080
QLCDNumber Class, 1078
QLinearGradient Class, 1083
QLineEdit Class, 1081
QListView Class, 1083
QListWidget Class, 1084
QListWidgetItem Class, 1085
QMainWindow Class, 1086
QMatrix4x4 Class, 1088
QMdiArea Class, 1089
QMdiSubWindow Class, 1090
QMediaObject Class, 1090
QMediaPlayer Class, 1091
QMediaPlaylist Class, 1092
QMenu Class, 1092
QMenuBar Class, 1093
QMessageBox Class, 1094
QMimeData Class, 1095
QMutex Class, 1095
QMutexLocker Class, 1096
QNetworkAccessManager Class, 1096
QNetworkProxy Class, 1097
QNetworkReply Class, 1097
QNetworkRequest Class, 1098
QNmeaPositionInfoSource Class, 1098
QObject Class, 1099
QOpenGLBuffer Class, 1100
QOpenGLContext Class, 1100
QOpenGLDebugLogger Class, 1101
QOpenGLFramebufferObject Class, 1102
QOpenGLFunctions Class, 1102
QOpenGLFunctions_3_2_Core Class, 1106
QOpenGLPaintDevice Class, 1114
QOpenGLShader Class, 1115
QOpenGLShaderProgram Class, 1115
QOpenGLTexture Class, 1119
QOpenGLTimerQuery Class, 1121

QOpenGLVersionProfile Class, 1121
QOpenGLVertexArrayObject Class, 1122
QOpenGLWidget Class, 1122
QPageSetupDialog Class, 1123
QPaintDevice Class, 1123
QPainter Class, 1123
QPainter2 Class, 1126
QPainterPath Class, 1126
QPen Class, 1127
QPicture Class, 1128
QPixmap Class, 1128
QPixmap2 Class, 1130
QPlainTextEdit Class, 1130
QPoint Class, 1132
QPointF Class, 1133
QPrinter Class, 1134
QPrinterInfo Class, 1135
QPrintPreviewDialog Class, 1133
QPrintPreviewWidget Class, 1133
QProcess Class, 1136
QProgressBar Class, 1137
QPushButton Class, 1138
QQmlError Class, 1138
QQuaternion Class, 1139
QQuickWidget Class, 1139
QRadioButton Class, 1140
QRect Class, 1140
QRegion Class, 1142
QRegularExpression Class, 1143
QRegularExpressionMatch Class, 1143
QRegularExpressionMatchIterator Class, 1144
QScreen Class, 1144
QScrollArea Class, 1145
QScrollBar Class, 1146
QSerialPort Class, 1146
QSerialPortInfo Class, 1147
QSize Class, 1147
QSlider Class, 1148
QSpinBox Class, 1148
QSplashScreen Class, 1149
QSplitter Class, 1149
QSqlDatabase Class, 1150
QSqlDriver Class, 1151
QSqlDriverCreatorBase Class, 1151
QSqlError Class, 1152
QSqlField Class, 1152
QSqlIndex Class, 1153
QSqlQuery Class, 1153
QSqlRecord Class, 1154
QStackedWidget Class, 1155
QStandardPaths Class, 1155
QStatusBar Class, 1156
QString2 Class, 1156
QStringList Class, 1157
QStringRef Class, 1158
QSurfaceFormat Class, 1160
QSystemTrayIcon Class, 1161
QTabBar Class, 1161
QTableView Class, 1164
QTableWidget Class, 1165
QTableWidgetItem Class, 1168
QTabWidget Class, 1163
QTcpServer Class, 1169
QTcpSocket Class, 1170
QTest Class, 1170
QTextBlock Class, 1170
QTextBrowser Class, 1171
QTextCharFormat Class, 1172
QTextCodec Class, 1174
QTextCursor Class, 1174
QTextDocument Class, 1176
QTextEdit Class, 1178
QTextStream Class, 1180
QTextStream2 Class, 1182
QTextStream3 Class, 1182
QTextStream4 Class, 1182
QTextStream5 Class, 1182
QThread Class, 1182
QThreadPool Class, 1183
QTime Class, 1184
QTimer Class, 1184
QToolBar Class, 1185
QToolButton Class, 1185
QTreeView Class, 1186
QTreeWidget Class, 1188
QTreeWidgetItem Class, 1190
QUrl Class, 1192
QUuid Class, 1193
QVariant Class, 1194
QVariant2 Class, 1195
QVariant3 Class, 1195
QVariant4 Class, 1195
QVariant5 Class, 1195
QVBoxLayout Class, 1193
QVector2D Class, 1195
QVector3D Class, 1196
QVector4D Class, 1197
QVideoWidget Class, 1197
QVideoWidgetControl Class, 1198
QWebEngineView Class, 1198
QWebView Class, 1199, 1200
QWidget Class, 1200
QWindow Class, 1206
QXmlStreamAttribute Class, 1210
QXmlStreamAttributes Class, 1210
QXmlStreamEntityDeclaration Class, 1211
QXmlStreamEntityResolver Class, 1211
QXmlStreamNamespaceDeclaration Class, 1211

QXmlStreamNotationDeclaration Class, 1211
QXmlStreamReader Class, 1211
QXmlStreamWriter Class, 1213
RingCodeHighlighter Class, 1214
RingREPL
What is new in Ring 1.4?, 147
RingSQLite
What is new in Ring 1.1?, 182
ringvm_callfunc()
Low Level Functions, 895
RingVM_CallList()
Low Level Functions, 892
RingVM_CFunctionsList()
Low Level Functions, 887
RingVM_ClassesList()
Low Level Functions, 888
ringvm_evalinscope()
Low Level Functions, 894
RingVM_FilesList()
Low Level Functions, 892
RingVM_FunctionsList()
Low Level Functions, 887
ringvm_give() function
Low Level Functions, 903
ringvm_hideerrorMsg()
Low Level Functions, 895
ringvm_info()
Low Level Functions, 903
RingVM_MemoryList()
Low Level Functions, 890
RingVM_PackagesList()
Low Level Functions, 889
ringvm_passerror()
Low Level Functions, 895
ringvm_scopescount()
Low Level Functions, 894
ringvm_see() and ringvm_give() functions
What is new in Ring 1.7?, 82
ringvm_see() function
Low Level Functions, 903
ringvm_settrace()
Low Level Functions, 893
ringvm_tracedata()
Low Level Functions, 893
ringvm_traceevent()
Low Level Functions, 894
ringvm_tracefunc()
Low Level Functions, 894
RingZip
Create Zip File, 521
Extract Zip File, 521
Introduction, 520
Print files in Zip file, 521
Using RingZip Classes, 521
Zip Class Reference, 523
ZipEntry Class Reference, 523
RingZip Library
What is new in Ring 1.3?, 166
Rotate Text
Desktop and Mobile Development, 746
Rules
The Type Hints Library, 839
Run Package
Ring Package Manager (RingPM), 875
Run the program
Getting Started - First Style, 197
Getting Started - Second Style, 200
Getting Started - Third Style, 202
Running Forms
Form Designer, 798
Runtime Errors
Reference, 990
Runtime Errors when Embedding Ring in Ring
Embedding Ring in Ring, 913

Samples
How to contribute?, 195
Samples in this book
Applications developed in little hours, 15
Samples Source (Authors)
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 606
Save and Restore Images
ODBC Functions, 327
Save Image Inside the Database
MySQL Functions, 335
Save output to string
RingLibCurl, 518
Scaling and Rotating Images
Graphics and Game Programming, 533
Scope Rules
Accessing the class attributes from braces inside class methods, 814
Conflict between Class Attributes and Local Variables, 811
Conflict between Global Variables and Class Attributes, 810
Conflict between self inside braces and self in the class region, 817
Creating a Class for each Window in GUI applications, 816
Defining Variables and Variables Access, 808
How Ring find the Variable?, 809
Introduction, 807
Summary of Scope Rules, 821
The For Loops uses the local scope, 821
The Self Object, 809
Three Scopes, 808

Using Braces to access objects inside Class Methods, 812
Using braces to escape from the current object scope, 820
Using Object.Attribute, 809
Scope Rules for Functions and Methods
Calling a function sharing the name with a method in the current class, 826
Example about Sharing Names between Functions and Methods, 824
How Ring find a functions and methods?, 824
Introduction, 823
Screen Shots
The Gold Magic 800 Game, 676
ScriptFunctions Class
Web Development (CGI Library), 510
Search
Lists, 263
Search of global names while defining the class attributes
Frequently Asked Questions, 963
Security and Internet Functions
Decrypt(), 351
Download(), 352
Encrypt(), 350
Example, 351
File Hash, 351
Introduction, 347
MD5(), 348
RandBytes(), 352
SendEmail(), 352
SHA1(), 349
SHA224(), 350
SHA256(), 349
SHA384(), 350
SHA512(), 349
Security Class
Stdlib Classes, 434
Send Parameters
Functions - First Style, 248
Functions - Second Style, 252
Functions - Third Style, 256
SendEmail()
Security and Internet Functions, 352
Serial Execution of Programs
Embedding Ring in Ring, 910
Server Example
RingLibuv, 546
Server Example using Classes
RingLibuv, 549
Set List Item
Lists, 262
setattribute()
Reflection and Meta-programming, 390
Setter and Getter
Object Oriented Programming, 358
SHA1()
Security and Internet Functions, 349
SHA224()
Security and Internet Functions, 350
SHA256()
Security and Internet Functions, 349
SHA384()
Security and Internet Functions, 350
SHA512()
Security and Internet Functions, 349
Shared Libraries
Extension, 922
Short-circuit evaluation
Control Structures - First Style, 234
Shutdown() Function
System Functions, 312
sign()
Stdlib Functions, 399
Simple
Language Design, 31
Simple Client and Server Example
Desktop and Mobile Development, 750
Sin() Implementation
Extension, 918
Single: Desktop and Mobile Development
Introduction, 679
sleep()
Stdlib Functions, 406
Smart Garbage Collector
Language Design, 39
Snakes And Ladder Game
What is new in Ring 1.10?, 46
Sort()
Lists, 263
Sort() and List of Objects
Object Oriented Programming, 365
Sound Class
Game Engine for 2D Games, 557
Source Code
Resources, 2039
The Gold Magic 800 Game, 678
Source Code File Sections
Program Structure, 259
space()
Low Level Functions, 884
Special Thanks
Introduction, 25
Special thanks to contributors
How to contribute?, 195
split()
Stdlib Functions, 396
splitmany()
Stdlib Functions, 396

Sprite Automatic Movement
Game Engine for 2D Games, 564
Sprite Class
Game Engine for 2D Games, 556
Sprite Keypress Event
Game Engine for 2D Games, 565
Sprite Mouse Event
Game Engine for 2D Games, 566
Sprite State Event
Game Engine for 2D Games, 567
SQLite
Introduction, 337
sqlite_close(), 338
sqlite_execute(), 338
sqlite_init(), 338
sqlite_open(), 338
SQLite Class
Stdlib Classes, 432
sqlite_close()
SQLite, 338
sqlite_execute()
SQLite, 338
sqlite_init()
SQLite, 338
sqlite_open()
SQLite, 338
Squares Puzzle Game
Applications developed in little hours, 10
SRandom()
Mathematical Functions, 293
Stack Class
Stdlib Classes, 417
Stars Fighter Game
Game Engine for 2D Games, 576
startswith()
Stdlib Functions, 400
Static Methods
Code Generator, 946
StdBase Class
Stdlib Classes, 412
Stdlib Classes
Conversion Class, 429
DataType Class, 428
DateTime Class, 423
Debug Class, 427
File Class, 425
HashTable Class, 418
Internet Class, 435
Introduction, 411
List Class, 415
Math Class, 420
MySQL Class, 431
ODBC Class, 430
PostgreSQL Class, 433
Queue Class, 417
Security Class, 434
SQLite Class, 432
Stack Class, 417
StdBase Class, 412
String Class, 413
System Class, 426
Tree Class, 419
Stdlib Functions
apppath(), 394
ASCIIList2Str(), 410
binarydigits(), 403
capitalized(), 397
changestring(), 406
dayofweek(), 404
direxists(), 406
endswith(), 400
EpochTime(), 408
evenorodd(), 402
factorial(), 398
factors(), 402
fibonacci(), 399
file2list(), 400
filter(), 396
FSize(), 407
gcd(), 401
getnumber(), 394
getstring(), 394
Introduction, 392
isleapyear(), 403
ismainsourcefile(), 406
ispalindrome(), 403
isprime(), 399
isspecial(), 397
isvowel(), 398
JustFileName(), 395
JustFilePath(), 394
lcm(), 401
linecount(), 398
List2Code() Function, 410
list2file(), 399
ListAllFiles() Function, 408
makedir(), 407
map(), 396
matrixmulti(), 403
matrixtrans(), 404
newlist(), 397
OSCopyFile() Function, 409
OSCopyFolder() Function, 409
OSCreateOpenFolder() Function, 409
OSDeleteFile() Function, 410
OSDeleteFolder() Function, 409
OSRenameFile() Function, 410
permutation(), 404

print(), 393
Print2Str() Function, 393
prodlist(), 402
puts(), 393
readline(), 405
sign(), 399
sleep(), 406
split(), 396
splitmany(), 396
startswith(), 400
Str2ASCIIList(), 410
substring(), 405
sumlist(), 401
SystemCmd() Function, 408
SystemSilent() Function, 409
times(), 395
TrimAll(), 407
TrimLeft(), 407
TrimRight(), 408
value(), 395
StdLib functions and classes written in Ring
What is new in Ring 1.1?, 179
Step Option
Control Structures - First Style, 232
Steps to create the extension
Create your first extension, 949
StopWatch Application
What is new in Ring 1.8?, 70
Str2ASCIIList()
Stdlib Functions, 410
Str2Hex()
Data Type, 289
str2list() and list2str()
Strings, 275
strcmp()
Strings, 275
String Class
Stdlib Classes, 413
String Literals
Strings, 271
String()
Data Type, 287
String2Constant Application
What is new in Ring 1.8?, 69
Strings
Access String Letters, 271
Convert Letters Case, 271
Copy(), 273
Find SubString, 274
Get Number of Characters from position, 274
Get String Length, 271
Get Substring from position to end, 274
Introduction, 270
Left(), 272
Lines(), 273
Merge binary characters, 276
Right(), 272
str2list() and list2str(), 275
strcmp(), 275
String Literals, 271
Substr(), 273
Transform Substring To Another Substring, 274
Trim(), 273
StyleFunctions Class
Web Development (CGI Library), 510
Substr()
Strings, 273
substring()
Stdlib Functions, 405
sumlist()
Stdlib Functions, 401
Summary of Scope Rules
Scope Rules, 821
Super Man 2016 Game
Game Engine for 2D Games, 591
Swap Items
Lists, 269
Switch Between Two Images
RingLibSDL, 538
Switch Statement
Control Structures - First Style, 230
Control Structures - Second Style, 237
Control Structures - Third Style, 240
Syntax Flexibility
Change Language Keywords, 828
Change Language Operators, 829
Ignore new lines after keywords, 837
Introduction, 827
Language Design, 37
Load Syntax Files, 829
Using 'case' as 'on' in switch statements, 835
Using 'def' as 'func' in functions/methods definition, 835
Using 'end' keyword after Packages/Classes/Functions, 836
Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions, 836
Using 'put' and 'get' as 'see' and 'give', 835
Using () around the function parameters, 829
in the start of the variable name, 832
Using braces { } in Packages/Classes/Functions, 836
Using braces to start and end different control structures, 834
Using Semi-colon after and between statements, 831
Using the 'else' keyword as 'other' in switch statement, 832
Using the 'elseif' keyword as 'but' in if statement, 832

Using the 'end' keyword in different control structures, 833
What is new in Ring 1.1?, 176
SysGet() Function
System Functions, 307
System Class
Stdlib Classes, 426
System Functions
ChDir() Function, 311
CurrentDir() Function, 311
Example, 308
ExeFileName() Function, 311
ExeFolder() Function, 311
Get Active Source File Name, 310
Get Command Line Arguments, 309
Introduction, 305
IsAndroid() Function, 308
IsFreeBSD() Function, 308
IsLinux() Function, 308
IsMacOSX() Function, 307
IsMSDOS() Function, 307
IsUnix() Function, 307
IsWindows() Function, 307
IsWindows64() Function, 307
PrevFileName() Function, 310
Shutdown() Function, 312
SysGet() Function, 307
System() Function, 306
Version() Function, 311
Windowsnl() Function, 309
System() Function
System Functions, 306
SystemCmd() Function
Stdlib Functions, 408
SystemSilent() Function
Stdlib Functions, 409

Tempfile()
Files, 300
Templates
Web Development (CGI Library), 474
Tempname()
Files, 300
Testing
How to contribute?, 195
Testing the extension
Create your first extension, 950
Text Class
Game Engine for 2D Games, 556
The Browser Menu
Ring Notepad, 215
The Camera
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 616
The Cards Game
Demo Programs, 772
Desktop and Mobile Development, 772
The Designer Windows
Form Designer, 797
The Difference between Qt and RingQt
Desktop and Mobile Development, 782
The Distribute Menu
Ring Notepad, 216
The documentation says functional programming is supported, but then this happens?
Frequently Asked Questions, 955
The Edit Menu
Ring Notepad, 213
The Events Loop
Desktop and Mobile Development, 681
RingLibuv, 546
The File Menu
Ring Notepad, 212
The files ring.ring and ring.ringo
Distributing Ring Application, 863
The First GUI Application
Desktop and Mobile Development, 680
The First Triangle
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 609
The First Window using RingFreeGLUT
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 607
The For Loops uses the local scope
Scope Rules, 821
What is new in Ring 1.8?, 75
The Game Story
The Gold Magic 800 Game, 676
The Gold Magic 800 Game
How to play?, 676
Introduction, 675
Screen Shots, 676
Source Code, 678
The Game Story, 676
What will you learn?, 676
The Help Menu
Ring Notepad, 216
The Main File in the Project
Ring Notepad, 211
The Natural Library
What is new in Ring 1.4?, 145
The Package Description File
Ring Package Manager (RingPM), 878
The Program Menu
Ring Notepad, 215
The Properties
Form Designer, 797
The Qt project for your Ring application

Building RingQt Applications for Mobile, 788
The Ring Package Manager (RingPM)
What is new in Ring 1.10?, 49
The RingPM Registry File
Ring Package Manager (RingPM), 881
The Self Object
Scope Rules, 809
The Tools Menu
Ring Notepad, 216
The Trace Library and the Interactive Debugger
BreakPoint, 907
Disable BreakPoints, 907
Execute Program Line by Line, 906
Interactive Debugger, 906
Introduction, 904
Loading the Trace library, 905
Pass Error, 906
Trace All Events, 905
Trace control flow between functions, 905
Using the Interactive Debugger, 907
The Type Hints Library
Example, 838
Introduction to the Type Hints Library, 837
Rules, 839
User Types, 838
Using Types inside Code, 839
Why Type Hints?, 838
The View Menu
Ring Notepad, 213
Threads
Graphics and Game Programming, 535
Threads Example
RingLibuv, 551
Threads Example - Using Classes
RingLibuv, 552
Threads Support
Code Generator, 935
Three Scopes
Scope Rules, 808
TicTacToe 3D Game
Using RingOpenGL and RingAllegro for 3D Graphics, 663
TicTacToe Game
Applications developed in little hours, 9
Time()
Date and Time, 278
TimeList()
Date and Time, 279
times()
Stdlib Functions, 395
Trace All Events
The Trace Library and the Interactive Debugger, 905
Trace control flow between functions
The Trace Library and the Interactive Debugger, 905
Trace Library and Interactive Debugger
What is new in Ring 1.5?, 129
Transaction Example
MySQL Functions, 336
Transactions and Using Commit and Rollback
ODBC Functions, 326
Transform Substring To Another Substring
Strings, 274
Transparent Image
Graphics and Game Programming, 534
Transparent Implementation
Language Design, 37
Tree Class
Stdlib Classes, 419
Triangle Rotation
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 612
Trim()
Strings, 273
TrimAll()
Stdlib Functions, 407
TrimLeft()
Stdlib Functions, 407
TrimRight()
Stdlib Functions, 408
TrueType Fonts
Graphics and Game Programming, 531
Try/Carch/Done
Eval() and Debugging, 313
Trying to be natural
Language Design, 32
Type Hints Library
What is new in Ring 1.5?, 132
Type()
Data Type, 283

Ungetc()
Files, 302
Unsigned()
Mathematical Functions, 293
Update Package
Ring Package Manager (RingPM), 875
Update the Android SDK
Building RingQt Applications for Mobile, 787
Updating the RingPM Registry
Ring Package Manager (RingPM), 874
Upload Files
Web Development (CGI Library), 468
URL Encode
Web Development (CGI Library), 473
Usage
Deploying Web Applications in the Cloud, 513
Use TTF Fonts
RingLibSDL, 540

User Types
The Type Hints Library, 838
Users registration and Login
Web Development (CGI Library), 495
Using '<' and ':' operators as 'from' keyword
What is new in Ring 1.3?, 165
Using 'case' as 'on' in switch statements
Syntax Flexibility, 835
Using 'def' as 'func' in functions/methods definition
Syntax Flexibility, 835
Using 'end' keyword after Packages/Classes/Functions
Syntax Flexibility, 836
Using 'endpackage'/'endclass'/'endfunc' keywords after Packages/Classes/Functions
Syntax Flexibility, 836
Using 'put' and 'get' as 'see' and 'give'
Syntax Flexibility, 835
Using () around the function parameters
Syntax Flexibility, 829
Using ? to print expression then new line
Getting Started - First Style, 198
What is new in Ring 1.6?, 98
in the start of the variable name
Syntax Flexibility, 832
Using && and || operators
What is new in Ring 1.6?, 97
Using _ in numbers
Mathematical Functions, 294
Using Atom
Using Other Code Editors, 220
Using Bootstrap Library using Functions
Web Development (CGI Library), 489
Using Bootstrap Library using Objects
Web Development (CGI Library), 490
Using braces { } in Packages/Classes/Functions
Syntax Flexibility, 836
Using Braces to access objects inside Class Methods
Scope Rules, 812
Using braces to escape from the current object scope
Scope Rules, 820
Using braces to start and end different control structures
Syntax Flexibility, 834
Using C/C++ Compiler and Linker
Distributing Ring Application, 863
Using configuration file that wrap C++ library
Code Generator, 937
Using CR as Carriage return
What is new in Ring 1.6?, 97
Using Emacs Editor
Using Other Code Editors, 222
Using Eval() with our Natural Code
Natural Language Programming, 446
Using f after numbers
Mathematical Functions, 294
Using Fonts
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 633
Using Geany
Using Other Code Editors, 219
Using Layout
Desktop and Mobile Development, 682
Using Layouts
Form Designer, 803
Using len() and For Loops
Performance Tips, 859
Using Lists during definition
Lists, 266
Using Many Source Code Files
Program Structure, 259
Using Notepad++
Using Other Code Editors, 218
Using NULL instead of NULLPointer()
What is new in Ring 1.2?, 173
Using Object.Attribute
Scope Rules, 809
Using Other Code Editors
Introduction, 217
Using Atom, 220
Using Emacs Editor, 222
Using Geany, 219
Using Notepad++, 218
Using Sublime Text 2, 221
Using Visual Studio IDE, 222
Visual Studio Code, 223
Using QCheckBox
Desktop and Mobile Development, 708
Using QComboBox Class
Desktop and Mobile Development, 689
Using QDateEdit
Desktop and Mobile Development, 703
Using QDesktopWidget Class
Desktop and Mobile Development, 745
Using QDial
Desktop and Mobile Development, 704
Using QDockWidget
Desktop and Mobile Development, 696
Using QFrame
Desktop and Mobile Development, 713
Using QInputDialog Class
Desktop and Mobile Development, 737
Using qLCDNumber
Desktop and Mobile Development, 734
Using QPrintPreviewDialog
Desktop and Mobile Development, 729
Using QProgressBar
Desktop and Mobile Development, 700
Using QProgressBar and Timer
Desktop and Mobile Development, 723

Using QRadioButton and QButtonGroup
Desktop and Mobile Development, 709
Using QSlider
Desktop and Mobile Development, 701
Using QSpinBox
Desktop and Mobile Development, 701
Using QTableWidget
Desktop and Mobile Development, 699
Using QTabWidget
Desktop and Mobile Development, 697
Using QTreeView and QFileSystemModel
Desktop and Mobile Development, 687
Using QTreeWidget and QTreeWidgetItem
Desktop and Mobile Development, 688
Using QWebView
Desktop and Mobile Development, 707
Using Ring2EXE
Building RingQt Applications for Mobile, 788
Distributing Ring Application using Ring2EXE, 864
Using RING_API_RETMANAGEDCPOINTER()
Extension, 923
Using RingOpenGL and RingAllegro for 3D Graphics
3D Cube and Texture, 654
Introduction, 653
Many Cubes, 658
More 3D Samples, 673
TicTacToe 3D Game, 663
Using RingOpenGL and RingFreeGLUT for 3D Graphics
Drawing using RingOpenGL, 608
Frames Per Second, 641
Introduction, 605
Keyboard Events and Colors, 613
Make a Cube using RingOpenGL and RingFreeGLUT, 650
Menu Events, 626
Mouse Events, 622
Samples Source (Authors), 606
The Camera, 616
The First Triangle, 609
The First Window using RingFreeGLUT, 607
Triangle Rotation, 612
Using Fonts, 633
What is RingFreeGLUT?, 607
What is RingOpenGL?, 606
Window Resize Event, 610
Using RingZip Classes
RingZip, 521
Using Self.Attribute
Object Oriented Programming, 367
Using Self.Attribute in the Class Region to define new attributes
What is new in Ring 1.1?, 183
Using Semi-colon after and between statements
Syntax Flexibility, 831
Using String2Constant
Multi-language applications, 804
Using Sublime Text 2
Using Other Code Editors, 221
Using Tab instead of char(9)
What is new in Ring 1.6?, 96
Using the
operator as not
What is new in Ring 1.6?, 97
Using the 'else' keyword as 'other' in switch statement
Syntax Flexibility, 832
Using the 'elseif' keyword as 'but' in if statement
Syntax Flexibility, 832
Using the 'end' keyword in different control structures
Syntax Flexibility, 833
Using the Interactive Debugger
The Trace Library and the Interactive Debugger, 907
Using the Natural Library
Defining Commands, 454
Defining commands using classes, 457
Introduction, 451
Natural Library - Demo Program, 452
Operators, 456
Using the QColorDialog Class
Desktop and Mobile Development, 733
Using the QFileDialog Class
Desktop and Mobile Development, 725
Using the QListWidget Class
Desktop and Mobile Development, 684
Using the QTextEdit Class
Desktop and Mobile Development, 683
Using the QTimer Class
Desktop and Mobile Development, 722
Using the tool
Code Generator, 928
Using This in the class region as Self
Object Oriented Programming, 368
What is new in Ring 1.8?, 73
Using This.Attribute and This.Method()
Object Oriented Programming, 367
Using This.Attribute in nested Braces inside the Class Methods
What is new in Ring 1.1?, 183
Using Types inside Code
The Type Hints Library, 839
Using Visual Studio IDE
Using Other Code Editors, 222

value()
Stdlib Functions, 395
Variables
Deep Copy, 225
Dynamic Typing, 225
Introduction, 224

Weakly Typed, 226
Variables Scope
Functions - First Style, 249
Functions - Second Style, 253
Functions - Third Style, 257
varptr()
Low Level Functions, 884
Version() Function
System Functions, 311
Video-Music-Player Application
Applications developed in little hours, 11
What is new in Ring 1.5?, 100
Virtual Machine Instructions
Reference, 993
Visual Implementation
Language Design, 38
Visual Studio Code
Using Other Code Editors, 223

Weakly Typed
Variables, 226
Web Development (CGI Library)
Application Class, 507
Configure the Apache web server, 459
Cookies, 471
CRUD Example using MVC, 493
Database, ModelBase & ControllerBase classes, 501
Generating Pages using Objects, 483
Gradient, 482
Hash Functions, 477
Hello World Program using the Web Library, 460
HTML Lists, 480
HTML Special Characters, 476
HTML Tables, 481
HtmlPage Class, 488, 512
HTTP Get Example, 461
HTTP POST Example, 466
Introduction, 458
Page Class, 508
Random Image, 479
Ring CGI Hello World Program, 460
ScriptFunctions Class, 510
StyleFunctions Class, 510
Templates, 474
Upload Files, 468
URL Encode, 473
Users registration and Login, 495
Using Bootstrap Library using Functions, 489
Using Bootstrap Library using Objects, 490
Web Library Features, 461
WebLib API, 506
WebPage Class, 511
Web Library Features
Web Development (CGI Library), 461
WebLib API
Web Development (CGI Library), 506
WebPage Class
Web Development (CGI Library), 511
Weight History Application
Desktop and Mobile Development, 753
Werdy Application
Applications developed in little hours, 14
What about predefined parameters or optional parameters in functions?
Frequently Asked Questions, 965
What about the Boolean values in Ring?
Frequently Asked Questions, 957
What are the advantages of using Ring over native C or C++?
Frequently Asked Questions, 953
What are the advantages to using Ring over C# or Java?
Frequently Asked Questions, 955
What are the advantages to using Ring over Lisp or Smalltalk?
Frequently Asked Questions, 952
What are the advantages to using Ring over Python and Ruby?
Frequently Asked Questions, 954
What are the advantages to using Ring over Tcl and Lua?
Frequently Asked Questions, 955
What happens when we create a new object?
Frequently Asked Questions, 961
What is new in Ring 1.10?
Better Tests, 49
Chess Game, 41
Game of Life Game, 44
Introduction, 40
Knight Tour Game, 43
List of changes and new features, 41
Minesweeper Game, 42
More Games, 47
More Improvements, 50
Pong Game, 45
Ring Extension for Visual Studio Code, 48
Snakes And Ladder Game, 46
The Ring Package Manager (RingPM), 49
What is new in Ring 1.1?
Better Code Generator for Extensions, 183
Better Documentation, 183
Better Natural Language Programming Support, 175
Game Engine for 2D Games, 181
Generate/Execute Ring Object Files (*.ringo), 176
Introduction, 174
List of changes and new features, 175
New Functions and Changes, 178
RingLibSDL, 181
RingSQLite, 182
StdLib functions and classes written in Ring, 179

Syntax Flexibility, 176
Using Self.Attribute in the Class Region to define new attributes, 183
Using This.Attribute in nested Braces inside the Class Methods, 183
What is new in Ring 1.2?
Better Call Command, 173
Better Functions, 170
Better Quality, 173
Better Ring Notepad, 170
Better RingQt, 170
Display Warnings Option, 173
Introduction, 168
List of changes and new features, 169
New Functions, 169
Objects Library for RingQt, 171
RingLibCurl, 172
Using NULL instead of NULLPointer(), 173
What is new in Ring 1.3?
Better Loop|Exit Command, 164
Better Ring Notepad, 160
Better RingQt, 156
Better StdLib, 164
Embedding Ring in Ring without sharing the State, 166
Form Designer, 167
Introduction, 155
List of changes and new features, 156
New Functions, 165
Return Self by Reference, 165
Ring mode for Emacs Editor, 163
RingZip Library, 166
Using '<' and ':' operators as 'from' keyword, 165
What is new in Ring 1.4.1?
What is new in Ring 1.4?, 152
What is new in Ring 1.4?
Better RingQt, 151
Better StdLib, 149
Better WebLib, 149
Change: Basic Extensions are separated from RingVM, 144
Convert between Numbers and Bytes, 148
Introduction, 143
List of changes and new features, 144
New Style to Ring Notepad, 147
Qt Class Convertor, 151
RingREPL, 147
The Natural Library, 145
What is new in Ring 1.4.1?, 152
What is new in Ring 1.5.1?
What is new in Ring 1.5?, 133
What is new in Ring 1.5.2?
What is new in Ring 1.5?, 138
What is new in Ring 1.5.3?
What is new in Ring 1.5?, 139
What is new in Ring 1.5.4?
What is new in Ring 1.5?, 142
What is new in Ring 1.5?
Better Code Generator for Extensions, 124
Better Documentation Generator for Extensions, 125
Better Objects Library, 111
Better Quality, 133
Better Ring Notepad, 102
Better RingQt, 109
Better StdLib, 105
Better WebLib, 106
Calculator Application, 101
Introduction, 99
List of changes and new features, 100
More Syntax Flexibility, 131
Ring VM - Tracing Functions, 125
RingFreeGLUT Extension, 111
RingOpenGL Extension, 120
Trace Library and Interactive Debugger, 129
Type Hints Library, 132
Video-Music-Player Application, 100
What is new in Ring 1.5.1?, 133
What is new in Ring 1.5.2?, 138
What is new in Ring 1.5.3?, 139
What is new in Ring 1.5.4?, 142
Windows StartUp Manager Application, 101
What is new in Ring 1.6?
Better Ring For Android, 90
Better Ring Notepad, 93
Better RingQt, 95
Better RingREPL, 96
Better RingVM, 96
Better Scripts for building Ring, 92
Better StdLib, 96
Employee Application, 89
Introduction, 88
List of changes and new features, 89
New Tool: Folder2qrc, 91
New Tool: Ring2EXE, 90
RingConsoleColors Extension, 92
RingMurmurHash Extension, 92
Using ? to print expression then new line, 98
Using && and || operators, 97
Using CR as Carriage return, 97
Using Tab instead of char(9), 96
Using the operator as not, 97
What is new in Ring 1.7?
Better Documentation, 85
Better Ring Notepad, 84
Better Ring VM, 85
Better Ring2EXE, 85
Better RingQt, 84

Better RingZip, 85
Better Trace Library, 83
Introduction, 80
List of changes and new features, 81
New Command: Load Package, 81
ring_state_new() and ring_state_mainfile() functions, 83
RingLibuv Extension, 86
ringvm_see() and ringvm_give() functions, 82
What is new in Ring 1.8?
Better Cards Game, 78
Better Code Generator For Extensions, 79
Better Form Designer, 77
Better Performance, 68
Better Ring Compiler and VM, 79
Better RingQt, 78
Compiling on Manjaro Linux, 73
Default value for object attributes is NULL, 75
Find in files Application, 69
FoxRing Library, 77
Introduction, 67
List of changes and new features, 68
Merge binary characters, 76
More 3D Samples, 71
Notes to extensions creators, 80
StopWatch Application, 70
String2Constant Application, 69
The For Loops uses the local scope, 75
Using This in the class region as Self, 73
What is new in Ring 1.9?
Better Code Generator for Extensions, 66
Better Memory Management, 65
Better Ring Notepad, 56
Better RingQt, 62
Better StdLib, 57
BigNumber Library, 58
Deploying Web applications in the Cloud, 61
Introduction, 50
List of changes and new features, 51
More Games, 53
More Improvements, 66
New Game : Gold Magic 800, 51
RingPostgreSQL Extension, 59
What is RingFreeGLUT?
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 607
What is RingOpenGL?
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 606
What is the difference between Ring and Python? And is Ring Open Source?
Frequently Asked Questions, 953
What will you learn?
The Gold Magic 800 Game, 676
Where can I write a program and execute it?
Frequently Asked Questions, 964
Which of 3 coding styles are commonly used or recommended by the community?
Frequently Asked Questions, 982
While Loop
Control Structures - First Style, 231
Control Structures - Third Style, 241
Why do we need Yet Another Programming Language (YAPL)?
Frequently Asked Questions, 951
Why I get a strange result when printing nl with lists?
Frequently Asked Questions, 966
Why I get Calling Function without definition Error?
Frequently Asked Questions, 977
Why is Ring largely focussed on UI creation?
Frequently Asked Questions, 953
Why is Ring weakly typed?
Frequently Asked Questions, 952
Why Ring is not case-sensitive
Frequently Asked Questions, 959
Why Ring uses 'See', 'Give', 'But' and 'Ok' Keywords?
Frequently Asked Questions, 956
Why Ring?
Language Design, 31
Why setClickEvent() doesn't see the object methods directly?
Frequently Asked Questions, 977
Why the ability to define your own languages Instead of just handing over the syntax so you can parse it using whatever code you like?
Frequently Asked Questions, 956
Why the Assignment operator uses Deep copy?
Frequently Asked Questions, 960
Why the window title bar is going outside the screen?
Frequently Asked Questions, 971
Why this example use the GetChar() twice?
Frequently Asked Questions, 968
Why Type Hints?
The Type Hints Library, 838
Why we don't use () after the qApp class name?
Frequently Asked Questions, 971
Why you can specify the number of loops you want to break out of?
Frequently Asked Questions, 956
Window Flags
Form Designer, 802
Window Resize Event
Using RingOpenGL and RingFreeGLUT for 3D Graphics, 610
Windows StartUp Manager Application
Applications developed in little hours, 13
What is new in Ring 1.5?, 101
Windowsnl() Function

System Functions, 309
Wrap structures
Code Generator, 930
Wrapping C++ Classes
Code Generator, 936
Write file using Write()
Files, 297
Writing Comments
Getting Started - First Style, 199
Getting Started - Second Style, 201
Getting Started - Third Style, 203

Zip Class Reference
RingZip, 523
ZipEntry Class Reference
RingZip, 523