

Index: Windows Programming in Assembly Language

Symbols

!(boolean_expression) 54
#(...)# in macro parameters 121
#(...)# macro quoting symbols 128
#{ ... }# sequence for manually passing parameters 91
#{...}# parameter quoting mechanism 86
#{...}#" code brackets in boolean expressions 56
#closeread compile-time statement 118
#closewrite compile-time statement 117
#else clause 111
#elseif clause in #if statement 111
#endif clause 111
#endmacro 120
#ERROR directive 117
#for..#endfor Compile-Time Loop 113
#KEYWORD reserved word 122
#macro 120
#openread compile-time statement 118
#openwrite compile-time statement 117
#PRINT directive 117
#system compile-time statement 116
#TERMINATOR keyword 122
#text..#endtext statement 119
#while..#endwhile Compile-Time Loop 115
#write compile-time statement 117
&& operator in boolean expressions 55
@a 54
@abs function 108
@ae 54
@align 79
@Align procedure option 74
@alignstack 79
@alignstack procedure option 74, 75
@arity compile-time function 35
@arity function 107
@b 54
@be 54
@byte compile-time function 106
@byte function 108
@c 54, 199
@cdecl procedure option 74, 75
@ceil function 108
@char compile-time function 106
@cos function 108
@cset compile-time function 106
@date function 108
@defined function 107
@delete function 109
@dim compile-time function 35
@dim function 107
@display 79
@display procedure option 74
@e 54
@elements compile-time function 35
@elements function 107
@elementsize compile-time function 35
@elementsize function 107
@enter 79
@enter procedure option 76
@eval function 128
@exp function 108
@External procedures 78
@extract function 108
@floor function 108
@frame 79
@g 54
@ge 54
@index function 109
@insert function 109
@int64 compile-time function 106
@int8 compile-time function 106
@isalpha function 108
@isalphanum function 108
@isconst function 107
@isdigit function 108
@IsExternal function 107
@isfreg function 107
@islower function 108
@ismem function 107
@isreg function 107
@isreg16 function 107
@isreg32 function 107
@isreg8 function 107
@isspace function 108
@istype function 107
@isupper function 108
@isxdigit function 108
@l 54
@le 54
@leave 79
@leave procedure option 76
@length function 109
@linenumber function 108, 130

- @log function 109
- @log10 function 109
- @lowercase function 110
- @max function 109
- @min function 109
- @na 54
- @nae 54
- @name function 106
- @nb 54
- @nbe 54
- @nc 54, 199
- @ne 54
- @nge 54
- @nl 54
- @nle 54
- @no 54, 200
- @noalignstack 79
- @noalignstack procedure option 74, 75
- @nodisplay 79
- @nodisplay procedure option 74
- @noenter 79
- @noenter procedure option 76
- @noframe 79
- @noframe procedure option 74
- @noleave 79
- @noleave procedure option 76
- @ns 54, 200
- @nz 54, 200
- @o 200
- @o, 54
- @odd function 109
- @offset function 107
- @pascal procedure option 74, 75
- @qword compile-time function 106
- @random function 109
- @randomize function 109
- @read compile-time function 118
- @real32 compile-time function 106
- @Returns procedure option 74
- @returns procedure option 76
- @rindex function 109
- @s 54, 200
- @sin function 108
- @size compile-time function 35
- @size function 107
- @sqrt function 109
- @stdcall procedure option 74, 75
- @string compile-time function 106
- @substr function 110

- @tan function 108
- @text operator 121
- @time function 109
- @tostring operator 127
- @typename function 107
- @uns64 compile-time function 106
- @uns8 compile-time function 106
- @uppercase function 110
- @use procedure option 76
- @use reg32 procedure option 74
- @z 54, 200
- _display_ variable 78
- _parms_ constant 78
- _vars_ constant 78
- || operator in boolean expressions 55
- A
- ABI (Application Binary Interface) 218
- Accessing the fields of a record 38
- Accessing the fields of a union 37
- Align directives in a record 42
- align procedure option 75
- AllocConsole 476, 482
- AND operator in boolean expressions 55
- Anonymous records 45
- Anonymous unions 46, 102
- Anonymous Unions and Records 44
- ANYEXCEPTION clause in the TRY..ENDTRY statement 71
- API Functions 231
- Arithmetic expressions (run-time) 177
- Array Types 32
- Array types 147
- array.index array indexing function 34
- Assignments at compile-time 98
- Associativity 189, 195
- Automatic code generation in procedures 80
- B
- BeginPaint 322, 347
- Boolean expressions 50, 199
- BREAK 64
- Break statement 216
- breakif 64
- C
- C calling convention 220
- C programming language 141
- C scalar data types 142
- C Types 224
- C/C++ Naming conventions 236
- Call instruction 85

- Callback functions 440
- Calling conventions 220
- Calling HLA Procedures 85
- Cascading exceptions 71
- case neutrality 238
- Character data types 144
- Character string types 153
- Class types 161
- Client regions 321
- Command Line Development 246
- Compile-time arithmetic and relational operators 99
- Compile-Time Assignment Statements 98
- Compile-time functions 105
- Compile-Time I/O 116
- Compile-Time Language 96
- Compile-time language performance 130
- Compile-time variable declarations 97
- Composite Data Types 31
- Composite data types 146
- Conditional assembly 110
- Conjunction in boolean expressions 55
- Console API 475
- Console applications 320
- Console I/O 476
- Constant declarations 169
- Constant declarations (compile-time language) 97
- Context-Free macros 123
- Context-free macros 348
- Context-free macros (HLA) 347
- CONTINUE 64
- Continue statement 216
- continueif 64
- Converting C/C++ boolean expressions to HLA boolean expressions 202
- Converting expressions into assembly language 180
- CreateConsoleScreenBuffer 483
- CreateFontIndirect 373
- CreateWindowEx 405
- Creating a New Project in RadASM 267
- Creating a Window 297
- Creating HLA Procedure Prototypes for Win32 API Functions 236
- D
- Data alignment and padding 219
- Data Types 26
- DEFAULT section of a SWITCH statement 210
- Deferred macro parameter expansion 128
- Device capabilities 352
- Device context 322
- Device context attributes 324
- Directly compatible types 28
- Discriminant union type 37
- Disjunction in boolean expressions 55
- Displaying a window 297
- Displaying messages during compilation 117
- Dithering 513
- DO..WHILE Loops 213
- DrawText 325
- DUP operator (array constants) 100
- E
- Eager evaluation of macro parameters 128
- Editing HLA source files within RadASM 272
- ELSE 203
- else 57
- ELSEIF 203
- elseif 57
- ENDFOR 215
- ENDIF 203
- EndPoint 323, 347
- ENDWHILE 212
- EnumFontFamilies 379
- Errors 117
- Escape character sequence 144
- ESP 218
- Event-oriented programming 286
- Exception handling 66
- Exception numbers 68
- EXIT 64
- exitif 65
- Expressions and temporary values 198
- ExtTextOut 336
- F
- Fields of a union 37
- FillConsoleOutputAttribute 483
- FillConsoleOutputCharacter 485
- FindWindow 438
- Floating point (real) data types 145
- FlushConsoleInputBuffer 486
- Fonts 368
- FOR loops 214
- FOR loops (for..endfor, forever..endfor, foreach..endfor) 62
- forever..endfor 216
- frame procedure option 74
- FreeConsole 486
- Function Calls 217
- Function return results 219

G
 GetBkColor 339
 GetBkMode 340
 GetConsoleCursorInfo 486
 GetConsoleScreenBufferInfo 487
 GetConsoleTitle 488
 GetConsoleWindow 489
 GetDC 347
 GetDeviceCaps 352
 GetNumberOfConsoleInputEvents 489
 GetScrollPos 405
 GetScrollRange 405
 GetStdHandle 490
 GetSystemMetrics 410
 GetTextAlign 338
 GetTextColor 339
 GetTextMetrics 377
 GetWindowDC 324, 347
 GOTO statement 216
 H
 Handles 289
 High Level Assembler (HLA) 11
 HLA Naming Conventions 236
 HLA Standard Library 11
 HLA.INI initialization file 251
 Hungarian Notation 241
 Hybrid high level boolean expressions 55
 Hybrid languages 141
 Hybrid parameter passing in HLA 90
 I
 IDE 246
 IF Statement 203
 If statements and time-critical code 58
 IF..ENDIF 56
 Implicit rules in a make file 20
 IN operator 54
 IN reg parameter specification 74
 Inherited fields in records 40
 inherits keyword 40
 Integer data types 142
 Integrated Development Environments 246
 L
 Lazy (pass by lazy evaluation) parameter option 73
 Leading (spacing in a font) 369
 Local symbols in macros 120
 Local symbols in multi-part macros 125
 Logical inch 510
 M
 Macro parameters containing commas or parentheses
 128
 Macros 119
 Make 14
 Make files 281
 Make menu in RadASM 253
 Make program (make.exe) 11
 Memory protection 441
 Message passing 287, 436
 MessageBox 475
 Monospaced fonts 320, 372
 multi-part macros 123
 N
 Name (pass by name) parameter option 73
 Naming Conventions 236, 238
 Nested records and unions 44
 NMake 14
 Non-client regions 321
 Non-proportional fonts 372
 NULL 47
 O
 OllyDbg Debugger 11
 Operator precedence in run-time expressions 189
 OR operator in boolean expressions 55
 P
 PAINTSTRUCT 324
 Parenthesis in macro parameters 121
 Pascal calling convention 220
 Pass by Reference 225
 Pass by Value 225
 Pass by value 86
 Passing parameters in registers 74, 91
 Path specifications in RadASM 251
 PeekConsoleInput 490
 Pointer constants 104
 Pointer types 46, 161
 Points (font sizes) 368
 PostMessage 439
 Procedural programming languages 141
 Procedure declarations 73
 Procedure Invocations 73
 Project organization 247
 Project types in RadASM 252
 Prototypes for Win32 API Functions 236
 R
 RadASM Integrated Development Environment (IDE)
 11
 RadASM project management 259
 RadASM templates 278
 RADASM.INI file 248

RAISE statement 66
 Raster fonts 372
 ReadConsole 493
 ReadConsoleInput 494
 ReadConsoleOutput 494
 ReadConsoleOutputAttribute 495
 ReadConsoleOutputCharacter 496
 Real data types 145
 Record (Structure) Types 38
 Record/structure types 148
 Reference parameters 87
 Regions 321
 Register parameters 91
 Register Preservation and Scratch Registers in Win32 Calls 218
 Registering a window class 290
 RegisterWindowMessage 437
 Relational operators 200
 ReleaseDC 324, 347
 REPEAT..UNTIL 61
 repeat..until 213
 Reraising an exception 71
 Result (pass by result) parameter option 73
 RET with NOFRAME option 82
 RGB value 340
 row-major order 34
 S
 Scratch Registers in Win32 Calls 218
 Scroll bars 404
 ScrollConsoleScreenBuffer 496
 SelectObject 376
 SendMessage 438
 SendMessageCallback 439
 SendMessageTimeout 439, 441
 Sequence points 190
 SetBkColor 339
 SetBkMode 339
 SetConsoleActiveScreenBuffer 498
 SetConsoleCursorInfo 498
 SetConsoleCursorPosition 498
 SetConsoleScreenBufferSize 499
 SetConsoleTextAttribute 499
 SetConsoleTitle 500
 SetConsoleWindowInfo 500
 SetScrollPos 407
 SetScrollRange 406
 SetStdHandle 501
 SetTextAlign 338
 SetTextColor 339
 Setting Up RadASM 248
 Side effects 190
 Signed vs. unsigned comparisons in boolean expressions 52
 Small projects and RadASM 284
 Snippets 283
 Stack Pointer 218
 Stdcall calling convention 220
 stdout.put macro implementation 131
 str.init 325
 str.put 325
 stralloc 325
 Structure Types 38
 Structured constants 99
 SWITCH/CASE statement 209
 T
 Templates in RadASM 278
 Temporary values in an expression 198
 TextOut 325, 329
 Thunk types 47
 Translating C/C++ expressions to assembly language 194
 True Type fonts 372
 TRY..EXCEPTION..ENDTRY statement 66
 tstralloc 326
 Type Coercion 48
 Typefaces 368
 Typefaces and fonts 368
 U
 Unicode 233
 Union data types 152
 Union Types 36
 UNPROTECTED clause in the TRY..ENDTRY statement 69
 Untyped reference parameters 74, 89
 User-defined compilation errors 117
 User-defined exceptions 68
 V
 Val (pass by value) parameter option 73
 Valres (pass by value/result) parameter option 73
 Value parameters 86
 Var (pass by reference) parameter option 73
 VAR (untyped reference parameters) 74
 Var type (untyped reference parameters) 90
 Variable parameter lists in a macro 126
 Variable parameter lists in macros 120
 Variant types 37
 Viewport origin 514

W
 w.AllocConsole 476, 482
 w.BeginPaint 322
 w.CreateConsoleScreenBuffer 483
 w.CreateFontIndirect 373
 w.CreateWindowEx 405
 w.DrawText 325
 w.EndPaint 323
 w.EnumFontFamilies 379
 w.ExtTextOut 336
 w.FillConsoleOutputAttribute 483
 w.FillConsoleOutputCharacter 485
 w.FindWindow 438
 w.FlushConsoleInputBuffer 486
 w.FreeConsole 486
 w.GetBkColor 340
 w.GetConsoleCursorInfo 486
 w.GetConsoleScreenBufferInfo 487
 w.GetConsoleTitle 488
 w.GetConsoleWindow 489
 w.GetDC 324
 w.GetDeviceCaps 352
 w.GetNumberOfConsoleInputEvents 489
 w.GetScrollPos 405
 w.GetScrollRange 405
 w.GetStdHandle 490
 w.GetSystemMetrics 410
 w.GetTextAlign 338
 w.GetTextColor 340
 w.GetTextMetrics 377
 w.GetWindowDC 324
 w.MessageBox 475
 w.PeekConsoleInput 490
 w.PostMessage 439
 w.ReadConsole 493
 w.ReadConsoleInput 494
 w.ReadConsoleOutput 494
 w.ReadConsoleOutputAttribute 495
 w.ReadConsoleOutputCharacter 496
 w.RegisterWindowMessage 437
 w.ReleaseDC 324
 w.ScrollConsoleScreenBuffer 496
 w.SelectObject 376
 w.SendMessage 438
 w.SendMessageCallback 439
 w.SendMessageTimeout 439, 441
 w.SetBkColor 340
 w.SetConsoleActiveScreenBuffer 498
 w.SetConsoleCursorInfo 498
 w.SetConsoleCursorPosition 498
 w.SetConsoleScreenBufferSize 499
 w.SetConsoleTextAttribute 499
 w.SetConsoleTitle 500
 w.SetConsoleWindowInfo 500
 w.SetScrollPos 407
 w.SetScrollRange 406
 w.SetStdHandle 501
 w.SetTextAlign 338
 w.SetTextColor 340
 w.TextOut 325, 329
 w.WriteConsole 501
 w.WriteConsoleAttribute 503
 w.WriteConsoleInput 502
 w.WriteConsoleOutput 502
 w.WriteConsoleOutputCharacter 504
 w else clause in a while statement 60
 WHILE loops 212
 WHILE statement 212
 WHILE..ENDWHILE 59
 Win32 API Functions 231
 Window classes 290, 295
 Window origin 514
 Window procedures 304
 Windows coordinate systems 327
 Windows Structured Exception Handler 68
 WM_COPYDATA 442
 WM_PAINT message 323
 WndProc 438
 WriteConsole 501
 WriteConsoleAttribute 503
 WriteConsoleInput 502
 WriteConsoleOutput 502
 WriteConsoleOutputCharacter 504