INDEX

Special Characters & Numbers

#define statements, 257
#hint text# element, 338
#ifdef block, 258
#ifdef/#else block, 258
$ idc_array, 301, 331
$ prefix, 297
$HOME/.idapro/ida.key, 192
$HOME/.idapro/ida.reg file, 44, 207
* (asterisk key), 144
: (colon) hotkey, 107
; (semicolon) hotkey, 107
{ } (bracing) syntax, 254
>> (right-shift operator), 253, 458
1 byte of storage (db), 97
2 bytes of storage (dw), 97
4 bytes of storage (dd), 97
8-byte doubles, 136
32-bit version, vs. 64-bit version, 38
64-bit version, vs. 32-bit version, 38

A

A hotkey, 122
-A option, 197
-a option, 218
A suffix, ASCII strings, 447
-a switch, 71
Abort command, 205
absolute jumps, 436–437
accept_file function, 359, 362, 365, 367
ACCEPT_FIRST flag, 359
accept_simpleton_file function, 362, 367
accept_simpleton_loader function, 362
access specifiers, IDC, 256
Actions box, Breakpoint Settings dialog, 526
activation records, 65, 83
ActiveSync, 517
Add Breakpoint option, 463, 523
Add standard structure button, Create Structure / Union dialog, 143, 152
Add Watch option, 529–530
add_auto_stkpnt2 function, 392, 394
add_entry function, 364
add_segm functions, 308
add_segm_ex function, 308
add_struc function, 307
add_struc_member function, 307
add_til functions, 367
add_til2 function, 367
AddBpt function, 531, 554
AddBptEx function, 531
AddEntryPoint function, 357
AddHotkey function, 261
Address box, Breakpoint Settings dialog, 524
Address field, Assemble Instruction dialog, 240
AddressOfEntryPoint field, 351
ADDSEG_*XXX* values, 308
advanced mode toolbar, 53, 208
aiSee, GDL viewer, 193
AL register, 458
algorithmic analysis, 416
alias = register syntax, 105
alignment, 352

© 2011 by Chris Eagle

All segments button, Memory snapshot confirmation dialog, 542
allins.hpp file, 235, 303
allmake.mak file, 289
allmake.unx file, 289
Allocate Heap Block option, Functions menu, 471
Allocate Stack Block option, Functions menu, 471
alphabetically sorting, in Functions window, 82
ALT-B hotkey, 99
alternate display format, selecting, 96
ALT-F8 hotkey, 461
ALT-H hotkey, 207
ALT-K hotkey, 118
ALT-L (Anchor) command, 243
ALT-P hotkey, 230, 424
ALT-Q hotkey, 147
altset function, 300
altvals, 297
ALT-*x* method, 191
Amini, Pedram, 204
ana function, 391
ana.cpp file, 385
analysis.idc script, 197
analyzer, for processor modules, 385–390
analyzing
  algorithms, 416
  binary, for different platform, 455
Anchor (ALT-L) command, 243
android_server server component, 570
anterior and posterior lines, 108
anti-debugging
  hiding debugger, 555–560
  technique, 452–454
  and x86emu emulation-oriented de-obfuscation of binaries, 471–472
anti–dynamic analysis techniques, 449–454
  detecting debuggers, 452–453
  detecting instrumentation, 451–452
  detecting virtualization, 449–451
  preventing debugging, 453–454
antipiracy techniques, 32
anti–reverse engineering techniques, 433–434
anti–static analysis techniques, 434–449
  disassembly desynchronization, 434–437
  dynamically computed target addresses, 437–444
  imported function obfuscation, 444–448
  targeted attacks on analysis tools, 448–449
Apache web server, 23
API (Application Programming Interface), 289–314
  header files, 290–294
  iteration techniques using, 310–314
  netnodes, 294–301
    creating, 295–297
    data storage in, 297–301
    deleting, 301
  SDK datatypes, 302–303
  SDK functions, 304–309
App TRK, 517
Appcall feature, for Bochs, 578–580
Appcall variable, 579
Append Function Tail option, 115
Application option, debugger process options dialog, 571
Application Programming Interface. *See* API
Apply new signature option, Signatures window, 75
AR_LONG constant, 260
AR_STR constant, 260
ar2idt.exe parser, 231
architectures
  of processor modules, 409–411
  RISC-style, 387
archive files, 155
area control block, 310
area_t (*area.hpp*), datatypes for SDK, 291, 293, 302
areacb_t class, 310
areacb_t variables, 310
area.hpp, 291, 310
arg_ prefix, 95
argc, 422, 425
argv, 422, 425
arithmetic instructions, simple, 11
ARM code, 410

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

armlinux_server server component, 570
array access operations, 135, 172
array elements
  accessing, 131
  selecting size for, 125
Array option, 125
array tag parameter, 298
Array-creation dialog, 124–125
array-manipulation functions, 259–260
arrays, 130–135
  attributes for, 124–126
  globally allocated arrays, 131–132
  globally allocated structures, 137
  heap-allocated arrays, 134–135
  heap-allocated structures, 138–140
  stack-allocated arrays, 132–134
  stack-allocated structures, 138
  structure member access, 135–137
  of structures, 140–141
arrows window, IDA text view, 65
asc_ prefix, 123
ASCII characters, 121, 447
Ascii column, PDF Objects window, 510
ASCII dump, searching, 99
ASCII printable characters, 27
ASCII String Style option, Option menu, 122
ASCII strings, 447
ash variable, 292, 399
asize_t get_struc_size function, 307
AskFile function, 263, 265
askfile_c function, 305
AskStr function, 263
askstr function, 305
AskUsingForm_c dialogs, 340
AskUsingForm_c function, 305, 337–338, 341
Ask*XXX* functions, 263, 292
ask*XXX* interface functions, 334
AskYN function, 263
askyn_c function, 305
ASM files, generating, 242–243
asm_t struct, 380, 399, 402
asms
  data member, 402
  field, 402
ASPack program, 441
ASProtect program, 441
Assemble dialog, 239–241
Assemble Instruction dialog, 240
Assemble option, Patch Program menu, 239
assembler tool, 4
assembly language call statement, 164
assembly languages, 4
Assume GCC v3.x names checkbox, 163
asterisk key (*), 144
asynchronous communications, 504
asynchronous interaction, 536–537
Asynchronous Sockets techniques, Windows, 504
AT&T assembly syntax, 9
atoll function, 264
Attach option, Debugger menu, 514, 518, 573
Attach to Process option, Debugger menu, 516, 574
attributes
  for arrays, 124–126
  for functions, 115–118
auto comments, 233
Auto comments option, 110
auto keyword, IDC, 252
Autogenerated name option, for named locations, 104
autogenerated names
  in Names window, 104
  prefixes for, 73
auto.hpp, for API, 291

B

B (button) field, 339
-B option, 197
b parameter, 94, 160
Bachaalany, Elias, 574
backdoor-style communications channels, 450
backward navigation button, 83
Bad instruction <BAD> marks option, 110–111
bar function, 106
base address, of array, 131
BaseClass, 158–159
basic blocks, 61–62, 176–177
basic mode toolbar, 53, 208
batch mode, 189, 196–198

- BDS (Binary Diffing Suite), 485
- beginner mode, 206
- big-endian, CUP, 10
- *bin* directory
  - FLAIR tools, 217
  - for SDK, 287
- binaries
  - OS X Mach-O, 24
  - searches, 493
  - statically linked, 178
  - used in first-generation languages, 4
- Binary Diffing Suite (BDS), 485
- binary executable files, 18, 434
- Binary File Descriptor library (libbfd), 24
- Binary File entry, 45–46
- binary file obfuscation, 19
- binary files, 347–375
  - alternative loaders, 372–373
  - analysis of unknown files, 348–349
  - loader for, 47–48
  - loader modules for
    - overview, 358
    - pcap loader, 366–372
    - simpleton loader, 361–366
    - writing using SDK, 358–360
  - manually loading Windows PE file, 349–357
  - scripted loaders, 373–375
- binary form, plug-ins, 500
- Binary Search dialog, 99
- binary searches, of database, 99–100
- `BINARY_ADD` byte code instruction, 379
- `BinDiff`, 485
- `BinNavi`, 280
- binutils tool suite, GNU, 24
- `block` statement, 160
- blocking operation, 286
- blocks, in disassembly window, 64
- Bochs, 574–580
  - Appcall feature for, 578–580
  - disk image mode for, 577
  - IDB mode for, 575–576
  - PE mode for, 576–577
- Bochs configuration dialog, 575
- Bochs control module, 576
- *bochsrc* file, 577
- Borland
  - code, 419
  - tools, 404
- Borland's Turbo Assembler (TASM), 9
- Borland-style make files, 289
- BOUNDS problem, Problems Window, 77
- BP based frame attribute, 117–118, 424
- BP equals to SP attribute, 118
- `bpt_NtContinue` function, 567–568
- bracing ({ }) syntax, 254
- branches, 171
- Break checkbox, Breakpoint Settings dialog, 526
- `break` statement, 279
- Breakpoint List option, Debugger menu, 523
- breakpoint manipulation tools, 519
- Breakpoint Settings dialog, 523–526
- breakpoints, in debugger, 522–526
- bss section, 68, 356
- B-tree–style database, 49
- buffer array, 94
- Bug Reports forum, Hex-Rays bulletin board, 58
- bugs, reporting, 58
- *BugScam* scripts, 481
- build scripts, 461
- *BUILDING.txt* file, 503
- bulletin boards, Hex-Rays, 58
- bundled graphing applications, 176
- Burneye ELF encryption tool, 442, 455–459, 465, 467
- button (B) field, 339
- `bximage` tool, 577
- byte code, 4, 379
- `Byte` function, 262
- `byte_patched` notification message, 322
- *bytes.hpp* file, 291, 399

C

- C enum, Enums window, 70
- C hotkey, 48, 120
- C notation, 130
- `-c` option, 197

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

c parameter, 94
C_HEADER_PATH option, 203
C_PREDEFINED_MACROS option, 203
C++, 156–166
  calling conventions, 88
  inheritance relationships, 164–165
  name mangling in, 162–163
  object life cycle in, 160–161
  reverse engineering references, 165–166
  RTTI, 163–164
  this pointer in, 156–157
  virtual functions and vtables, 157–160
c++filt utility, 25–26
calculator program, Windows, 25
call esi instruction, 492
call flow type, 62, 171
call graphs, 178–180
call instructions, 112, 272, 456, 467, 480
Call Stack dialog, 529
call statement, 437
call_vfunc function, 159
call-by-reference, 255
call-by-value, 255
callflow function, 171
calling conventions
  compiler differences for, 430–432
  for stack frames, 85–89
call-style cross-references, 171
callui function, 305
canned search features, 98
canonical feature (CF), 381
Capture the Flag
  binary, DEFCON, 278
  network, DEFCON, 496
Case-sensitive option, 99–100
case-sensitive searches, 100, 493
catalog, of named constants, 112–113
cdecl calling convention, 85–86, 129
cdecl functions, 116
_cdecl modifier, 85
CF (canonical feature), 381
CF_CALL flag, 381
CF_CHGn flag, 381
CF_STOP flag, 381
CF_USEn flag, 381
cfg directory, 39
Change Byte menu option, Patch Program, 238
Change Color button, color selection dialog, 208
Change exception definition, Exception Handling dialog, 564
Change segment attributes dialog, 543
Change Stack Pointer option, 118
Character terminated strings, 122
Characteristics field, 357
charset function, 300
charval function, 297, 300
charval interface, 300
CheckBptfunction, 531–532
choose function, 305, 334
Choose project to attach to dialog, 515
choose2 dialog, 337
choose2 function, 305, 334, 336, 566
chooser dialog, 335
chunked functions, 114–115, 272
CL register, 431
Clampi trojan, 442
class constructor, 160
.class file, 472
Class Informer plug-in, 420, 506–508
class relationships, deducing between, 165
click-and-drag operation, 119
C-like pseudocode, 500
closing, IDA database files, 51–52
cmd variable, 385, 394
cmd.Operands array, 387, 390
cnbits field, 385
code
  converting to data, 119–120
  display options for, 109–111
code argument, 339
code bytes, distinguishing from data bytes, 48
CODE class, 308
code cross-references, 65, 168–169
Code option, 435
CODE XREF, 169
coding scheme, used in Names window, 73
COFF libraries, 219
collabREate plug-in, 503–506

Collapse Group button, 187
collapsed node demo, 187
collapsed structures, 146, 153–154
collapsing blocks, in disassembly window, 64
Collect garbage option, 52
collisions, 221, 223
colon (:) hotkey, 107
color key, 54
color palettes, 192
color selection dialog, 208
coloring nodes, 186
colorized disassembly listings, 245
colors, customizing, 207–208
Colors command, 54
Columns menu option, Hex window, 67
comma operator, 253
COMMAND function, 536
command history list, 40
command-line
  arguments, 22
  tools
    Borland, 418, 426
    Exports window, 68
    IDA, 251
    Segments window, 74
Comment directive, 232
comment member, for plug-ins, 317
comment.cmt file, 234–235
CommentEx function, 270
commenting
  anterior and posterior lines, 108
  augmenting information for, with loadint, 233–235
  auto, 233
  function comments, 108
  overview, 106
  regular comments, 107
  repeatable comments, 107–108
  virtual repeatable, 108
comments option, 110
common operations bar, console user interface, 190
compact_til function, 369
Compilation successful message, parsing header files, 150
Compiler configuration dialog, Option menu, 151
compiler differences, 415–432
  alternative calling conventions, 430–432
  debug vs. release binaries, 428–430
  jump tables, 416–420
  locating main, 421–428
  RTTI implementations, 420
  switch statements, 416–420
compiler validation, reasons for disassembly, 7
compilers
  GNU, 86, 136
  utilizing stack frames, 83
compiling functions, 89
computer licenses, 33
concrete_class object, 164
Condition field, Breakpoint Settings dialog, 525
conditional branching, 11, 171
conditional breakpoints, 523
conditional jumps, 436–437
configuration dialog, Bochs, 575
configuration files, 39
configuring plug-ins, 330–331
connect function, 69, 127
console mode, 190–196
  common features of, 190–191
  specific features of
    for Linux, 192–194
    for OS X, 194–196
    for Windows, 191
consoles
  I/O library, 190
  limitations of, 190
  mouse server, Linux, 192
  user interface, 190
constant index values, 137
constants, formatting options for, 112
CONTEXT record, 567
CONTEXT structure, 440, 472, 568
context-sensitive menus, 60, 102, 112, 501
Continue button, toolbar buttons, 521
Continue command, 521
Continue with Unpacked Base option, 53
control flow graphs, 169, 178, 185
control module, Bochs, 576
converting data, to code (and vice versa), 119–120

Cooper, Jeremy, 193, 195
copyright notices, 218
CPU flag, 437, 520
CPU instructions
  sets, 286
  undocumented, 110
CPU registers, 440
cpu_data function, 394
crashes, restarting after, 52–53
CRC16 value, 220
crc16.cpp file, 220
Create as array option, 126
Create C File option, File menu, 500
Create EXE File command, 360, 365
Create EXE File option, File menu, 542
create function, 296
Create function tails loader option, Kernel Options, 115
Create name anyway option, for named locations, 105
Create Segment command, 353
Create Structure/Union dialog, 143, 152
Create union checkbox, Create Structure/Union dialog, 143
CREATE_BACKUPS option, 202
create_filename_cmt function, 363
create_func_frame function, 401
CreateArray function, 260, 301
createImportLabel function, 553–554
CreateNetnode function, 332
CreateThread function, 471
CRITICAL_SECTION object, 121
Cross References option, View menu, 477
cross-references (xrefs), 168–176
  code cross-references, 169–171
  data cross-references, 171–173
  display window, 174
  enumerating, using API, 311–314
  for function calls, 175–176
  graphs for, 180–185
  lists of, 173–175
  navigational purposes, 81
  subview, opening, 174
  text, mousing over, 173
Cross-References tab, 172, 187
cryptographic library, OpenSSL, 215–216, 229
C-style strings, 71, 122
C-style union, 143
CTRL-B hotkey, 100
CTRL-ENTER hotkey, 83
CTRL-F1 hotkey, 203
CTRL-F4 hotkey, 191
CTRL-Q hotkey, 204
CTRL-T hotkey, 99
CTRL-X hotkey, 174
current instruction location, 386
current position indicator, 54
custom cross-reference graphs, 183
custom data
  formats, 474
  data types, 474
custom_ana code, 408
custom_emu code, 408
custom_mnem code, 408
custom_out code, 408
custom_outop code, 408
customizing, 201–210
  colors, 207–208
  configuration files, 201–207
  toolbars, 208–210
cya instruction, 408
Cygwin environment, 17

D

D command, 144
D hotkey, 120, 122
d_out function, 394, 398, 401
dashed line break, 171
data
  converting to code, 119–120
  specifying sizes for, 121–122
data bytes, distinguishing from code bytes, 48
data carousel, 121–122
DATA class, 308
data cross-references, 168
data displays, 55
Data Format menu, Hex window, 67
data storage, in netnodes, 297–301
data structures. *See also* datatypes
  arrays, 130–135
  IDA structures, 142–146
  importing new, 149–151
  and TIL files, 154–156

- data structures (*continued*)
  - using standard structures, 151–154
  - using structure templates, 146–149
- `DATA XREF`, 169
- database events, 321
- Database Restore dialog, 52–53
- data-flow analysis, 481
- DataRescue, 32
- datatype setup dialog, 121
- datatypes. *See also* data structures
  - custom, 474
  - Hex-Rays, 501
  - for SDK, 302–303
  - toggling through, 122
- db (1 byte of storage), 97
- `dbg_notification_t` enum, 536
- `dbg_step_until_ret` notification, 537
- `dbg_trace` notification, 537
- *dbg.hpp*, for API, 291
- dd (4 bytes of storage), 97
- dead listings, 79
- debug binaries, vs. release binaries, 428–430
- `DEBUG` flag, 330
- debug registers, 440
- debugger, 513–580
  - automating with plug-ins, 536–538
  - detecting, 452–453
  - displays, reasons for disassembly, 7
  - displays in, 518–521
  - handling exceptions with, 561–568
  - and IDA databases, 541–543
  - IdaStealth plug-in for, 560–561
  - instruction pointer warning, 549
  - launching, 514–518
  - preventing, 453–454
  - process control with, 521–530
    - breakpoints, 522–526
    - stack traces, 528–529
    - tracing, 526–528
    - watches, 529–530
  - process options dialog, 571–572
  - remote debugging with, 569–574
    - attaching to remote process, 573–574
    - exception handling during, 574
    - using Hex-Rays debugging server, 570–573
    - using scripts and plug-ins during, 574
  - selection dialog, 515–516
  - sessions, MyNav, 508
  - setup dialog, 545–546, 562–563, 574
  - scripting for, 530–535
  - using Bochs, 574–580
    - Appcall feature for, 578–580
    - disk image mode for, 577
    - IDB mode for, 575–576
    - PE mode for, 576–577
  - using with obfuscated code, 543–560
    - decryption and decompression loops, 546–550
    - hiding debugger, 555–560
    - import table reconstruction, 550–555
    - launching process, 545–546
    - overview, 540–541
  - warning message, 518
- Debugger menu
  - Attach option, 514, 518, 573
  - Attach to Process option, 516, 574
  - Breakpoint List option, 523
  - Debugger Options command, 562
  - Function Tracing option, 526
  - Instruction Tracing option, 526
  - Pause Process option, 516
  - Process Options command, 571
  - Refresh memory command, 579
  - Run option, 516
  - Run to Cursor option, 516
  - Select debugger option, 548
  - Stack Trace command, 528–529
  - Start Process option, 516, 518
  - Switch Debugger menu, 516
  - Take Memory Snapshot command, 542
  - Terminate Process option, 517
  - Watch List option, 530
- DECISION problem, Problems Window, 77
- declarations, in IDA text view, 65
- decoding function, Burneye, 456–459, 465
- decompiler editing options, Hex-Rays, 501
- decompilers, 5
- dedicated frame pointer, 91
- *.def* files, 403

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

- Default checkbox, Save Disassembly Desktop dialog, 209
- Default offset column, 136
- DEFAULT_FILE_FILTER option, 206
- DEFCON 18, Capture the Flag network, 496
- Deflate (Pack database) option, 52
- DelBpt function, 531
- deltas, 392
- Demangled Names, Options menu, 162
- Denial of Service attack, 168
- de-obfuscation of binaries, static, 454–472
  - script-oriented, 455–460
  - x86emu emulation-oriented, 460–472
    - and anti-debugging, 471–472
    - de-obfuscation using, 465–470
    - features of, 470–471
    - initialization of, 462
    - operation of, 463–465
- de-obfuscation stub, 441–442, 446
- destination buffer (dest), 273
- destructor table, 492
- destructors, 160–161
- detecting
  - debuggers, 452–453
  - instrumentation, 451–452
  - virtualization, 449–451
- DIF files, generating, 244
- directory layout
  - overview, 38
  - for SDK, 287–289
  - *sig* directory, 39
  - *til* directory, 40
- Directory option, debugger process options dialog, 572
- disassembler analysis tool, 454
- disassembly, 3–14
  - anti–static analysis techniques, 434–437
  - first-generation languages, 4
  - fourth-generation languages, 4
  - how performed, 7–14
    - basic algorithm for, 8–9
    - linear sweep disassembly, 9–10
    - recursive descent disassembly, 11–14
  - overview, 5
  - reasons for, 6–7
  - second-generation languages, 4
  - theory of, 4
  - third-generation languages, 4
- disassembly line display options, 109
- disassembly line parts, 109
- disassembly location, jumping to, 82
- Disassembly tab
  - color selection dialog, 208
  - Options menu, 109
- disassembly view
  - IDA desktop, 55
  - synchronizing with hex view, 67
- disassembly window, 60–65
  - IDA graph view in, 61–64
  - IDA text view in, 64–65
- disassembly window scroll bar, 82
- disclosure, of vulnerability, 483
- disclosure event, 483
- discovery event, 483
- disk image mode, for Bochs, 577
- *diskio.hpp* file, 359, 362
- dispatcher functions, 305
- Display at startup checkbox, 44
- Display Disassembly Line Parts section, Disassembly tab, 110
- display format, selecting, 96
- Display Graphs option, 487
- Display indexes option, 126
- Display only defined strings option, Strings window, 71
- display options, disassembly line, 109
- DISPLAY_COMMAND_LINE option, 204, 206, 251
- DISPLAY_PATCH_SUBMENU option, 204, 206
- displays, 59–77
  - context-sensitive menu actions in, 60
  - in debugger, 518–521
  - principal, 60–66
    - disassembly window, 60–65
    - Functions window, 66
    - Output window, 66
  - secondary, 66–70
    - Enums window, 70
    - Exports window, 68
    - Hex View window, 67–68
    - Imports window, 68–69
    - Structures window, 69

displays (*continued*)
- tertiary, 70–77
  - Function Calls window, 76
  - Names window, 72–74
  - Problems window, 76–77
  - Segments window, 74
  - Signatures window, 74–75
  - Strings window, 70–72
  - Type Libraries window, 75
- and undo, 59

DisplayWelcome value, 44
diStorm utility, 28
divide-by-zero error, 440
DLL (Dynamic Link Library), 462
dll2idt.exe parser, 231
dnbits field, 385
Dfirst function, 268
DfirstB function, 268
Dnext function, 268
DnextB function, 268
Do not display this dialog box again option, 207
Does not return attribute, 117
Don’t pack database option, 51
DON’T SAVE database option, 52
dos.ldw (MS-DOS EXE loader), 45
doStruct function, 369
DOT language, 176
dotty tool, 176
double word. *See* 4 bytes of storage (dd)
double-click navigation, 80–81, 185
double-clicking
- cross-reference address, 173
- in Function Calls Window, 76
- function chunks, 115
- hexadecimal values, 81
- listed scripts, 250
- in Names window, 72
- in Output window, 81
- in Segments window, 74
- strings, in Stings windows, 70
- structure names, 146
- symbol names, 175

download page, Hex-Rays, 499
DR0–3 registers, 523
DR0–7 registers, 523
dt_*xxx* values, 388
dummy names, 102, 104, 128, 214
Dump Embedded PE option, File menu, 471
Dump option, File menu, 470
Dump Typeinfo to IDC File command, 155
dumpbin utility, 25
dup2 function, 498
dw (2 bytes of storage), 97
Dword function, 262, 269, 456
dynamic analysis, of malware, 6
Dynamic Link Library (DLL), 462
dynamic linking, 22
dynamic memory allocation function, 134
dynamic_cast operator, 163
dynamically computed target addresses, anti–static analysis techniques, 437–444

E

-e command-line argument, 28
e_lfanew field, 350
EAX register, 89, 94, 436, 439, 559–560
EBP (extended base pointer) register, 91, 439, 451
ebc.py, 411
ebx register, 552
Edit Breakpoint option, 523
Edit Exceptions button, Debugger Setup dialog, 562
Edit menu, Plugins menu, 485, 508
editing imported functions, 230
EIP instruction pointer, 462
Element width attribute, 126
ELF binaries, 17
ELF encryption tool, Burneye, 455–458
ELF libraries, 219
ELF-specific parsing, 24
empty structure definition, 143
emu function, 390–391
emu.cpp file, 391
Emulate menu, Switch Thread option, 471
emulation, advantage of over debugging, 461
emulators, 390–394. *See also* x86emu emulator, de-obfuscation of binaries using
Enabled checkbox, Breakpoint Settings dialog, 524

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

EnableTracing function, 533
enabling line prefixes, 63
End address attribute, 116
ENTRY symbol, 575
entry.hpp, for API, 291
Enumerate Heap option, View menu, 471
enumerated datatype, C enum, 70
enumerating
  cross-references, 311–314
  functions, 310
  single stack frame, 490
  structure members, 311
Enums window, 70
envp array, 422
epilogue, of functions, 85
Erdelyi, Gergely, 250, 280
error handling, in IDC language, 258–259
error messages, 258
error strings, 218
ESC key, 60, 82
ESI register, 457
ESP-based stack frame, 90–92
etc directory, for SDK, 288
event notification, for plug-ins, 321–322
exact matches binary searches, 493
exception confirmation dialog, 564–565
Exception Editing dialog, 563
exception handlers, 438–440, 472, 565
Exception Handling dialog, 564
exceptions
  handling during remote debugging, 574
  handling with debugger, 561–568
Exceptions dialog, 562–563
exceptions.cfg file, 563, 574
EXE files, generating, 243–244
exec_request_t function, 286
execstack command, 38
executable files
  Exports window, 68
  using strings on, 28
executable statements, grouping, 83
execute_sync function, 286
execution
  of plug-ins, 322–324
  of scripts, 250–251
execution control commands, 522
execution traces, 526
exe.sig file, 421
_exit function, 422
Expand Struct Type option, Edit menu, 145
expanding collapsed structures, 153
exploit-development, 6, 488–495
  finding useful virtual addresses, 494–495
  locating instruction sequences, 492–494
  stack frame breakdown, 488–492
export entry, 231
export ordinal number, 68
Exports window, 68, 545
expressions, in IDC language, 253
expr.hpp file, 292, 331
extended base pointer (EBP) register, 91, 439, 451
extending IDC, with plug-ins, 331–333
extern keyword, 252
extern section, 477–478
external (global) symbols, 20
external mode graphs, 177
external-style graph, 176
extract_address function, 401

F

f argument, 129
f_LOADER type, 410
F2 hotkey, 523
F12 hotkey, 177
fake interrupt descriptor table, 462
Falliere, Nicolas, 453, 555–558
far addresses, 169
Far function attribute, 117
Fast Library Acquisition for Identification and Recognition (FLAIR), 216–217, 583
Fast Library Identification and Recognition Technology (FLIRT) signatures. *See* FLIRT signatures
fastcall calling convention, 157
fastcall convention for x86, 87–88
fastcall modifier, 88
fclose function, 265

feature field, 381
FF_XXX constants, 307
fgetc function, 265
field names, 135
file classification, 16–20
  file, 16–18
  PE Tools, 18–19
  PEiD, 19–20
File column, FLIRT signature selection, 214
file command, 16
File dialog, 45
file extensions, 16
file loading, 45–48
File menu
  Create C File option, 500
  Create EXE File option, 542
  Dump Embedded PE option, 471
  Dump option, 470
  Script File option, 554
File offset value, 239
File Open dialog, 44
File Save dialog, 365
FILE stream pointer, 365
FILE type, 359
file utility, 16–18, 218–219
FILE_EXTENSIONS option, 205–206
file2base function, 364–365
FileAlignment field, 352
FileAlignment value, 352
fileformatname parameter, 359, 362, 365
filelength function, 265
file-loading dialog, 358
FilemonClass class, 452
filename pattern, 205
FILEREG_PATCHABLE, 364
Find all occurrences checkbox, Text Search dialog, 99
FindBinary function, 269, 493
FindCode function, 269, 272
FindData function, 269
FindText function, 269
FindWindow function, 452
FindXXX functions, 269
first_from function, 313
first_to function, 313
first-generation languages, 4
fix_proc utility, 404
fixed-length instructions, 9
fl_CF-type cross-references, 273
flag field, 385
flags field, for plug-ins, 317
flags field for loaders, 359
FLAIR (Fast Library Acquisition for Identification and Recognition), 216–217, 583
flair directory, 216
flair57.zip version, 216
Flake, Halvar, 481
flat memory model, 117
FLIRT (Fast Library Identification and Recognition Technology) signatures, 211–225
  applying, 212–216
  creating signature files, 216–225
    creating pattern files, 219–221
    static libraries for, 217–219
  overview, 212
  startup signatures, 224–225
flowcharts, 177–178
flowchart-style graph, graph view, 55
flows, 62
Follow system keyboard layout option, Preferences dialog, 196
Follow TCP Stream command, 496
-fomit-frame-pointer compiler option, 91
Font command, 57
Font menu, 519
foo function, 12
foobar subroutine, 82
footer function, 401
fopen function, 265
for loops, 254
forking existing projects, CollabREate, 505
form argument, 338
form parameter, 305
form function, 264
formal parameter names, 228
format strings, 305, 492
formatting
  constants, options for, 112
  global variables, as structures, 149
  instruction operands, 112–113
  stack-allocated structures, 148
formcb_t function, 339

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

forward navigation button, 83
fourth-generation languages, 4
fprintf function, 265, 490–492
fpro.h, for API, 292
fputc function, 265
frame pointer, 84, 118
Frame pointer delta attribute, 117
frame.hpp, 292, 306
free_til function, 369
FreeBSD application, 213, 224, 422, 498
freeware versions, of IDA, 33
from address, in cross-references, 168
frregs field, 311
frsize field, 311
FS register, 439
full-line comments, 108
func_t (*funcs.hpp*), datatypes for SDK, 302, 308, 310
FuncItems generator, 283
funcs control block, 310
funcs.hpp, 292, 310
function call
  graphs, 76, 169, 178
  instructions, 12
  tree, 76
Function Calls window, 76
function comments, 108
Function editing dialog, 116
function parameters, 83, 85
function tails, 115
Function Tracing option, Debugger menu, 526
functions, 113–119
  attributes for, 115–118
  augmenting information for, 228–233
  calling, 84–85
  chunks of, 114–115
  compiling, 89
  creating new, 114
  deleting, 114
  emulated by x86emu, 467
  enumerating, using API, 310
  in IDC language
    code cross-reference, 267
    data cross-reference, 268
    database manipulation, 268–269
    database search, 269–270
    dealing with functions, 266–267
    disassembly line components, 270
    file input/output, 264–265
    manipulating database names, 266
    reading and modifying data, 262–263
    string-manipulation, 264
    for user interaction, 263–264
  oriented control flow graph, 185
  overloading, C++, 162
  overview of, 83
  for SDK, 304–309
  signatures for, 229
  stack pointer adjustments, 118–119
  tracing, 526
  types, setting, 129
  undefine, 119
Functions data display, 55
Functions list generator, 282
Functions menu, 471
Functions window, 56, 60, 66, 82, 175, 443
fuzzing technique, 6

G

G hotkey, 82, 207
g++ compiler, GNU, 86, 156
g++ versions, 163
Gaobot worm, 19
Gas (GNU Assembler), 9
gcc compiler, GNU, 86
GCC tags, 219
gdb(GNU Debugger), 454, 517
GDB Configuration dialog, 572–573
gdb sessions, 569
gdb_server, 569, 572–573
gdbserver component, GNU Debugger, 517
GDL (Graph Description Language), 176, 193
gdl.hpp, for API, 292
General dialog, 60, 123
General Registers view, 519–520
General Registers window, 520, 525
general-purpose searches, 98
Generate serial names option, 124

generating signatures, 39
Get prefix, 262
get_byte function, 304
get_first_cref_from function, 309
get_first_cref_to function, 309
get_first_dref_from function, 309
get_first_dref_to function, 309
get_frame function, 306
get_frame_retsize function, 401
get_func function, 306
get_func_name function, 306
get_func_qty function, 306
get_long function, 304
get_next_func function, 306
get_many_bytes function, 304
get_member function, 307
get_member_by_name function, 307
get_name function, 306
get_name_ea function, 306
get_next_area function, 310
get_next_cref_from function, 309
get_next_cref_to function, 309
get_next_dref_from function, 309
get_next_dref_to function, 309
get_original_byte function, 304
get_original_long function, 304
get_original_word function, 304
get_original_*XXX* functions, 304
get_reg_val function, 538
get_screen_ea function, 305
get_segm_by_name function, 307
get_segm_name function, 308
get_struc function, 307
get_struc_id function, 307
get_true_segm_name function, 308
get_word function, 304
GetArrayElement function, 301
GetBptAttr function, 531
GetBptEA function, 531
GetBptQty function, 531
GetCommandLine function, 426–427
GetCommandLineA function, 552
GetDebuggerEvent function, 532–533, 538, 556
GetDisasm function, 270
GetEntryPointQty function, 275
GetEnvironmentStrings function, 427
GetEvent*XXX* function, 533–535
GetFrameLvarSize function, 490
GetFrameRegsSize function, 490
getFuncAddr function, 479
GetFunctionAttr function, 266, 272
GetFunctionFlags function, 277
GetFunctionName function, 266
GetInputFile function, 275
getline function, 334
getmainargs library function, 425
GetMemberName function, 482
GetMemberOffset function, 271
GetMemberSize function, 482–483
GetMnem function, 270
GetModulehandleA function, 444
GetOpnd function, 270
getn_area function, 310
getn_func function, 306
getnseg function, 308
getopcode.c program, 493
GetOperandValue function, 270
GetOpType function, 270
GetProcAddress function, 445–446, 448, 468–469, 550, 552, 554
GetRegValue function, 525, 530
getseg function, 307, 363
GetStrucSize, 271
Gigapede, 541
gl_comm variable, 397
global (external) symbols, 20
global array, 534
global offset table (GOT), 274, 492, 494–495
global persistent arrays, 259
global variables, formatting as structures, 149
globally allocated arrays, 131–132
globally allocated structures, 137
gnome-terminal, Gnome, 193
GNU Assembler (Gas), 9
GNU binutils tool suite, 24
GNU compilers, 86, 136, 156
GNU Debugger (gdb), 454, 517
GNU linker, 404
Go button, 45
GOT (global offset table), 274, 492, 494–495
got section, 477–478, 495
goto statements, 502
graph components, 168
Graph Description Language (GDL), 176, 193
graph mode, 185

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

graph node, 178
Graph Overview data display, 55, 62, 185
Graph tab, 60
graph view, 55, 185–186
graph viewer, qwingraph, 176
graph view–style display, 55
GRAPH_FORMAT variable, 176
GRAPH_VISUALIZER option, 176, 193,194, 202
graph-based display mode, IDA freeware 5.0, 583
graphing, 176–187
  integrated graph view, 185–188
  third-party graphing, 176–185
    call graphs, 178–180
    cross-reference graphs, 180–182
    custom cross-reference graphs, 182–185
    flowcharts, 177–178
graphing applications, 176
graphs
  grouping nodes in, 187
  used in Function Calls Window, 76
graphviz project, 176
grep-style search, 290
Group Nodes option, 64, 187
grouping
  blocks, in disassembly window, 64
  executable statements, 83
  nodes, within graphs, 187
GUI configuration file, 39
GUI versions, of IDA, 197
gunzip archive, 37

H

.h suffix, 290
Hall of Shame, Hex-Rays website, 32
handling exceptions
  with debugger, 561–568
  during remote debugging, 574
Hardware Breakpoint checkbox, Breakpoint Settings dialog, 524
Hardware breakpoint mode radio buttons, Breakpoint Settings dialog, 524
hardware breakpoints, 523, 524, 544, 546
HAS_CALL flags, 389
HAS_JABS flag, 389
HAS_JREL flag, 389
hash function, 447
hashset function, 300
hashstr function, 300
hashval function, 300
hashval_long function, 300
hashvals, 297, 300
hash*XXX* functions, 300
head command, 212
header fields, PE Tools, 19
header files, for API, 290–294
  area.hpp, 291
  auto.hpp, 291
  bytes.hpp, 291
  dbg.hpp, 291
  entry.hpp, 291
  expr.hpp, 292
  fpro.h, 292
  frame.hpp, 292
  funcs.hpp, 292
  gdl.hpp, 292
  ida.hpp, 292
  idp.hpp, 292
  kernwin.hpp, 292
  lines.hpp, 292
  loader.hpp, 292
  name.hpp, 293
  netnode.hpp, 293
  pro.h, 293
  search.hpp, 293
  segment.hpp, 293
  struct.hpp, 293
  typeinf.hpp, 293
  ua.hpp, 293
  xref.hpp, 293–294
header function, 401
header structure, MS-DOS, 152
.headers program segment, 462
.headers section, 354
.heap database segment, 462
heap program, 134
heap_array variable, 135
HeapAlloc function, 468
heap-allocated arrays, 134–135
heap-allocated structures, 138–140
help files, 204
help member, for plug-ins, 318

Help menu, IDA, 34
HELPFILE option, 203
hex dumps, 99, 191
hex editor, 67
hex searches, conducting, 100
hex values, two-digit, 99
Hex View window, 40, 67–68, 99, 519
hexadecimal constant, 112
hexadecimal values, 81
Hex-Rays
  blog, 579
  bulletin boards, posting on, 58
  debugging server, remote debugging using, 570–573
  download page for, 499
  plug-in, 500–502
  stance on piracy, 32
  support page and forums, 35
hidden messages, restoring, 44
Hide Casts option, 501
Hide Group option, 187
hide_wait_box function, 323
HideDebugger.idc script, 560–561
hiding debugger, 555–560
History subkey, IDA Windows registry key, 45
HKEY_CURRENT_USER\ Software\Hex-Rays\IDA registry key, 44, 207
HKEY_CURRENT_USER\Software\Hex-Rays\IDA\Hidden Messages registry key, 207
hook_to_notification_point function, 321, 399, 536
Hostname option, debugger process options dialog, 572
hotkey field, 331
hotkey reassignment, in *idagui.cfg*, 204
hotkeys, 40, 261
.hpp suffix, 290
HT_DBG function, 537
HT_DBG notification type, 536
HTI_PAK1 constant, 368
HTI_*XXX* values, 368
HTML document, 16
HTML files, 204, 245
HTTP response packet, 371
hyperlinks, vs. names, 80
HyperUnpackMe2, 472–473

I

icebp instruction, 564
id field for processors, 385
.id0 file, 49
.id1 file, 49
IDA command line, 251
IDA comments, using semicolon prefix in, 107–108
IDA
  configuration file, 37
  crashes, restarting after, 52–53
  cross-references, 76
  database, as virtual memory, 460–461
  database files
    closing, 51–52
    creation of, 50–51
    and debugger, 541–543
    overview, 48–50
    reopening, 52–53
    searching, 98–100
  desktop
    behavior of during analysis, 56–57
    overview, 53–56
    tips and tricks for, 57
  directory, 36
  download page, 190
  executables, 36
  extensions, loaders directory, 39
  freeware 5.0, 581–583
  graph view, in disassembly window
    creating additional disassembly windows, 64
    grouping and collapsing blocks in, 64
    overview, 61
    panning in, 62–63
    rearranging blocks in, 64
  IDA Palace, 36
  IDA Sync, Windows Asynchronous Sockets techniques used by, 504
  installer, 34
  loader, 50
  modules, plug-ins directory, 39
  notifications, CollabREate, 504
  parser, 150
  scripting, 256, 455

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

  as software
    licenses, 33
    purchasing, 34
    upgrading, 34
    user interface of, 40
    versions, 33
  stack-pointer analysis, 230
  Strings options, 123
  structures, 142–146
    creating new, 142–143
    editing structure members, 144–146
    stack frames as, 146
  text view, in disassembly window, 64–65
  View-EIP disassembly window, 519–520
  View-ESP disassembly window, 520
  Windows registry key, 45
  workspace, 44
ida_export function, 294
IDA_SDK_VERSION macro, 293
idaapi module, 281, 579
idaapi.processor_t class, 411
ida.cfg file, 39, 111, 176, 193, 202–203
<*IDADIR*> install location, 36
idag64.exe, 38
IDA-generated variable names, mapping, 96–97
idag.exe, 36
idagui.cfg configuration file, 39, 203–206, 238, 251
IDA.HLP file, 338
ida.hpp file, 290, 292
ida.idc file, 261
ida.int file, 233
idaidp.hpp 380
idainfo (*ida.hpp*), datatypes for SDK, 303
idainfo structure, 292
ida.int comment file, 234
ida.key file, 32
idamake.pl, 324
IdaPdf plug-in, 509–510
IDAPython plug-in, 37, 503
  examples, 281–284
    enumerating cross-references, 283
    enumerating exported functions, 283–284
    enumerating functions, 282
    enumerating instructions, 282–283
idaq64.exe, 38
idaq.exe, 36
idasdk61.zip file, 286
IDAStealth configuration dialog, 561
IdaStealth plug-in, for debugger, 560–561
.idata section, 241
idatui.cfg file, 39, 206–207
idauser.cfg file, 203
idauserg.cfg file, 206
idausert.cfg file, 206
idautils module, 281–282
IDA-View window, 55, 60
idaw.exe, 36
ida-x86emu plug-in, 342, 461–462, 506
.idb extension, 51
.idb files, 229, 504
IDB mode, for Bochs, 575–576
IDB_2_PAT utility, 221
idb_event::byte_patched, 321
idb_event::cmt_changed, 321
IDC command dialog, 255
idc directory, 39
IDC functions, SDK implementation, 586–608
IDC language
  error handling in, 258–259
  examples, 270–280
    emulating assembly language behavior, 278–280
    enumerating cross-references, 272–274
    enumerating exported functions, 275
    enumerating functions, 270–271
    enumerating instructions, 271–272
    finding and labeling function arguments, 275–277
  expressions, 253
  functions
    code cross-reference, 267
    data cross-reference, 268
    database manipulation, 268–269
    database search, 269–270
    dealing with functions, 266–267
    disassembly line components, 270

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

IDC language (*continued*)
  functions (*continued*)
    file input/output, 264–265
    manipulating database names, 266
    reading and modifying data, 262–263
    string-manipulation, 264
    for user interaction, 263–264
  objects, 256–257
  persistent data storage in, 259–260
  programs, 257–258
  SDK cross-reference for, 585–608
  statements, 254
  variables, 252–253
idc module, 281
IDC script, 455
IDC slices, 253
IDC statements, 553
idc_create_netnode function, 332
idc_func_t datatype, 331
idc_value_t (*expr.hpp*), datatypes for SDK, 302, 332
IDC-based loader, 373
idc.idc file, 257
Identical Functions, PatchDiff2, 486
Identifier search, 99
IDP_INTERFACE_VERSION constant, 316
idp.def file, 404
idp.hpp file, 292, 400
ids directory, 39
IDS files
  augmenting information for functions, 230–233
  IDA parlance, 39
ids hierarchy, 231
IDS utilities, 228-229
idsnames file, 233
idsutils, 229
.*idt* file, 275
.*idt* generator script, 283
Ignore instructions/data definitions option, Strings window, 71–72
IMAGE_DOS_HEADER structure, 152–154, 350
IMAGE_NT_HEADERS structure, 152–154, 350, 352
IMAGE_SECTION_HEADER structure, 352
IMAGE_SECTION_HEADER template, 352
ImageBase field, 351
Import REConstruction (ImpREC) utility, 541
import table, 68
import_node netnode, 294
import_type function, 369
imported functions
  editing, 230
  obfuscation, anti–static analysis techniques, 444–448
Imports window, 68–69, 443–444
ImpREC (Import REConstruction) utility, 541
in instruction, 451
include (INC) files, generating, 243
include directive, 151, 261
include directory, 288
include files, 151
Include in names list option, for named locations, 104
include statement, 261
indent parameter, 397
INDENTATION option, 202
index function, 222
Index of IDC functions, 252, 261
Indexes radio buttons, 126
inf.mf flag, 400
inheritance hierarchy, determining, 164
inheritance relationships, in C++, 164–165
init member, for plug-ins, 317
init method, 536
init_loader_options function, 360, 363
initialization, of plug-ins, 320–321
inline constructors, 164
inline functions, 164
Input file option, debugger process options dialog, 572
Input File options, 47
ins.cpp file, 381
INSERT key, 143, 152, 155
Insert option, 149
insn_t (*ua.hpp*), datatypes for SDK, 293, 303, 385
install_make.txt file, 289
install_visual.txt file, 326
install_xxx.txt files, 288, 324
installation of 32-bit Python, IDAPython, 503

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

installing, 36–40
  32-bit vs. 64-bit, 38
  directory layout, 38–40
  on Linux, 37–38
  on OS X, 37–38
  plug-ins, 329–330
  and SELinux, 38
  on Windows, 36–37
instruction emulator, 380, 460–461
instruction operands, formatting, 112–113
Instruction Pointer (IP), 527
instruction sets, CPU, 286
Instruction Tracing option, Debugger menu, 526
Instructions constant, 383
instrumentation, detecting, 451–452
int 3 instruction, 439, 523, 564
int get_segm_qty function, 308
int set_segm_name function, 308
int type, 128
integer index, 230
Intel syntax, 9
intel_data function, 398
internal heap implementation, 468
interpreter, for Python byte code, 379
invoke_callbacks function, 400
I/O functions, 292
IP (Instruction Pointer), 527
iret instruction, 435
is_far_jump function, 401
is_sp_based function, 401
is_switch function, 401
iscode member, 313
IsDebugged field, PEB, 556
IsDebuggerPresent function, 452, 468, 556
isLoaded function, 262, 263, 304
Items on line attribute, 125
iteration techniques, using API, 310–314
  enumerating cross-references, 311–314
  enumerating functions, 310
  enumerating structure members, 311
iTERM, 194
itype field, 386

J

j suffix, 171
ja instruction, 418
Java byte code, 379
Java Database Connectivity (JDBC), 505
Java loader, 372
Java virtual machine, 472
JDBC (Java Database Connectivity), 505
jmp esi instruction, 492
jmp esp instruction, 492–493
jmp statement, 10
Jump command, 477
jump flow type, 62, 171
Jump function, 263, 264, 428
jump tables, compiler differences for, 416–420
Jump to Address command, 477
Jump to Address dialog, 82
Jump To Cursor button, x86emu Emulator dialog, 464
Jump to Next Position option, 83
Jump to Previous Position operation, 82
Jump to Problem command, 204
JumpQ option, 204
jump-style cross-references, 171
jumpto function, 305
junk strings, 71
jz instruction, 436

K

Kernel Options, 46, 115
kernel32_GetCommandLineA, 552
kernel32_VirtualAlloc function, 578–579
kernel32.dll, 446, 448, 520, 546, 552, 559
kernwin.hpp, for API, 292
key file, safeguarding, 34
keyboards
  different layouts, 194
  zoom control, 62
Kiel OMF 166 object files, 219
konsole, KDE's, 193
Koret, Joxean, 508

L

-L option, 23
label component, 338
launching, 44–48
  debugger, 514–518
  Go button, 45
  New button, 44
  Previous button, 45
  process, 517
  Windows installer, 36
ldd (list dynamic dependencies) utility, 22–23
ldr directory, for SDK, 288
LDRF_RELOAD flag, 359
LDSC (loader description) object, 359
leave instruction, 93, 408
legacy mode graphs, 193
len function, 283
letter codes, 21
Levine, John R., 22
lib directory, for SDK, 288
libbfd (Binary File Descriptor library), 24
libc_FreeBSD80.exc file, 222
libc_FreeBSD80.pat file, 220
libc_start_main function, 423–424, 427
libc.a version, 213
Library func attribute, 117
library handle, 468
Library name column, FLIRT signature selection, 214
license agreement dialog, 197
license enforcement, 32
licenses, for IDA, 33
life cycle, of plug-ins, 318–319
limitations
  of consoles, 190
  of IDA freeware 5.0, 582
line prefixes, enabling, 63
Line prefixes option, 110
linear sweep disassembly, 9–10
lines.hpp file, 292, 395
link libraries, 343
linking, 22
linput_t (loader input type), 359
Linux
  based IDA installation, 193
  console mode for, 192–194
  console mouse server for, 192
  installing on, 37–38
  terminal programs on, 192
  text display in, 192
linux_server server component, 570
linux_serverx64 server component, 570
list dynamic dependencies (ldd) utility, 22–23
list_callers function, 313
listing view, 55
listing-style display, 55
Litchfield, David, 493
little-endian, CUP, 10
lnames data member, 402
Load a New File dialog, 46
Load Desktop command, 57
Load desktop option, Windows menu, 209
Load from file radio button, x86emu Set Memory Values dialog, 465
Load type library option, in Type Libraries window, 75
load_file function, 359, 372, 410
load_pcap_file function, 369–370
load_simpleton_file, 363
loader description (LDSC) object, 359
loader input type (linput_t), 359
loader modules, for binary files
  overview, 358
  pcap loader, 366–372
  simpleton loader, 361–366
  writing using SDK, 358–360
Loader segment checkbox, Change segment attributes dialog, 543
Loader segments button, Memory snapshot confirmation dialog, 542
loader warnings, 49
LOADER_EXT variable, 366
loader_failure function, 359
loader_t structure, 292, 358
loader-generated informational messages, 49
loader.hpp file, 292, 316, 358
loaders directory, 39, 45
loadfile function, 265
loading files, 45–47, 155
Loading Offset field, 46
loading process, 358
Loading Segment field, 46
loadint utilities, 233–235

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

loadint57.zip version, 233
LoadLibrary function, 445–446, 550
LoadLibraryA function, 447, 468
Local Bochs debugger, 575
local debugging, 517
Local name option, for named locations, 104
Local Types command, 149
Local Types entry dialog, 150
Local Types window, 149–150
local variables
  layout, in stack frames, 89
  naming, 102–103
Local variables area attribute, 116
locations, renaming, 104
LocByName function, 267, 274
LocByNameEx function, 266
lodsb instruction, 458
Log if same IP option, Tracing Options dialog, 527
logical addresses, 242
loopne instruction, 10
lowercase letter codes, 21
LPH struct, 380, 385
lpSubKey parameter, 229
lread4bytes function, 362
LST files, generating, 243
ltoa function, 264

M

Mac keyboard, vs. PC keyboard, 194
mac_server server component, 570
mac_serverx64 server component, 570
Machine field, 351
machine languages, 4, 111
Mach-O loader, 410
MackT, 541
MACRO keyword, 207
macros, 206–207, 249
magic files, 16
magic numbers, 16
main method, compiler differences for, 421–428
Main toolbar, turning off, 208
make files, plug-ins, 500
Make imports section option, 244
make_data notification, 401
MakeByte function, 269
MakeCode function, 268
makecode parameter, 364
MakeComm function, 269
MakeFunction function, 269
MakeLine function, 395, 397
MakeNameEx function, 266
MakeStr function, 269
MakeUnkn function, 268
malicious PDF files, 509
malloc function, 66, 134, 477
malware analysis, reasons for disassembly, 6
mangled names, 163
manipulating disassembly, 101–126
  arrays, attributes for, 124–126
  code display options, 109–111
  commenting, 106–108
  converting data to code (and vice versa), 119–120
  data transformations, 121–124
  formatting instruction operands, 112–113
  functions, 113–119
  naming, 102–105
Manual load option, for file headers, 152
manually overriding purged bytes, 230
MAP files, generating, 242
mapping, IDA-generated variable names, 96–97
Mark as autogenerated option, 124
Mark consecutive traced events with same IP option, Tracing Options dialog, 527–528
MASM (Microsoft Assembler), 9
master list of structures, 152
Matched Functions, PatchDiff2, 486–487
MAX_NAMES_LENGTH option, 202
Maximum possible size attribute, 125
MAXSTR constant, 586
MD5 value, CollabREate, 505
mem2base function, 370
member_t (*struct.hpp*), datatypes for SDK, 303, 307
members array, 311
.memcpy, 274
Memory Organization dialog, 48
Memory snapshot confirmation dialog, 542
memory usage parameters, 202

memqty field, 311
memset operation, 430, 495
menu bar, console user interface, 190
Message function, 254, 263, 579
MessageBoxA function, 444
messages
  loader-generated informational, 49
  restoring hidden, 44
Metasploit project, 493, 496
Micallef, Steve, 35, 289
Microsoft, Patch Tuesday cycle, 476
Microsoft Assembler (MASM), 9
Microsoft Developer Network (MSDN), 25
Microsoft linker, 404
Microsoft Visual C++ compiler, 114
Microsoft Visual Studio suite, 25
Minimum offset column, 136
MIPS binary, 278
MIPS processor module, 240
Misc tab, color selection dialog, 208
mitigation, of vulnerability, 483
mitigation event, 483
mkidp syntax, 404
mkidp.exe utility, 404
mnemonics, 4
modal dialogs, 174, 337
modeless dialogs, 174, 337
Modify menu item, 520
module directory, for SDK, 288
Modules view, 519
Modules window, 520–521
mouse support, 190
mov instructions, 12, 92, 234
mov statements, 275–276
Move Current Segment command, 354
move_segm function, 360
movsb instruction, 527
MSDN (Microsoft Developer Network), 25
MS-DOS 8.3 name-length convention, 221
MS-DOS EXE loader (*dos.ldw*), 45
MS-DOS executable file, 18
MS-DOS header structure, 152
MS-DOS stub, 403–405
msfpescan tool, Metasploit project, 493
msg function, 305
mutual ptrace technique, 453
my_func function, 255
MyNav plug-in, 508–509
mynav.py script, 508
MZ magic number, 16, 152

N

N hotkey, 102–103, 105
-n option (sigmake), 224
-n option (loadint), 234
nalt.hpp file, 294
.nam file, 49
Name conflict dialog, 105
name decoration, 162
Name directive, 231
Name generation area, 123
name mangling, 26, 162–163
Name of function attribute, 115
Name function, 266
name parameter, 308, 364
name-change dialog, 102
NameChars set, 202
NameEx function, 266
named constants, catalog of, 112–113
named licenses, 33
named locations, 103–105
  Autogenerated name option, 104
  Create name anyway option, 105
  Include in names list option, 104
  Local name option, 104
  Public name option, 104
  Weak name option, 105
name-demangling options, 162
name.hpp, 293
name-length convention, MS-DOS 8.3, 221
names, vs. hyperlinks, 80
Names window, 72–74, 102
naming, 102–105
  conventions, Hex-Rays, 501
  import table entries, 552
  parameters and local variables, 102–103
  register names, 105
NASM (Netwide Assembler), 9, 28
navigating disassembly
  double-click navigation, 80–81
  jump to address, 82
  navigation history, 82–83
  searching database, 98–100

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

navigation band, 54, 443–444
navigation history list, 185
ncol parameter, 336
ndisasm utility, 28
NEF_*XXX* flags, 359
neflags parameter, 359
negative deltas, 392
netnode class, 259, 293–295, 301
netnode index value, 331
netnode.hpp file, 293–295
netnodenumber member, 294, 296
netnodes, 294–301, 585
  creating, 295–297
  data storage in, 297–301
  deleting, 301
Netwide Assembler (NASM), 9, 28
network attack sessions, 496
network connection, X.25-style, 113
New button, 44
new operator, 159–160
New Project dialog, Visual Studio, 326–327
new vertices, introducing, 64
Newger, Jan, 560
Next Sequence of Bytes option, Search menu, 100
NextFunction function, 266
nm utility, 20–21
No edge arrow, 62
no operation (NOP) instructions, 240, 494
NO_OBSOLETE_FUNCS macro, 316
nodeidx_t operator, 296
nodes, 168, 187
noGPM option, TVision, 192
nonmodal dialog, 337
nonstandard structures, 142
NOP (no operation) instructions, 240, 494
NOP slides, 494, 496–497
Normal edge arrow, 62
normal flow, 62
notification event, 483
notification of vulnerability, 483
notify field, 399
notify function, 400
NOVICE option, 206
NtContinue function, 567
ntdll_NtQueryInformationProcess function, 557
ntdll.dll, 546, 557–558, 567
NtGlobalFlags field, PEB, 556–557
NtQueryInformationProcess function, 557–558
NtSetInformationThread function, 558
NTSTATUS code, 558
NULL pointer, 299
Number of elements attribute, 125
Number of opcode bytes option, 111
NumberOfSections field, 352

O

o_displ type, 392
o_imm type, 392
o_mem type, 392
o_near type, 392
obfuscated code analysis, 433–474
  anti–dynamic analysis techniques, 449–454
    detecting debuggers, 452–453
    detecting instrumentation, 451–452
    detecting virtualization, 449–451
    preventing debugging, 453–454
  anti–static analysis techniques, 434–449
    disassembly desynchronization, 434–437
    dynamically computed target addresses, 437–444
    imported function obfuscation, 444–448
    targeted attacks on analysis tools, 448–449
  static de-obfuscation of binaries, 454–472
    script-oriented, 455–460
    x86emu emulation-oriented, 460–472
  using debugger with, 543–560
    decryption and decompression loops, 546–550
    hiding debugger, 555–560
    import table reconstruction, 550–555
    launching process, 545–546
    overview, 540–541
  virtual machine-based, 472–474
obfuscation process, 19, 541

obfuscators, 540, 548
objdump utility
  debugging information, 24
  disassembly listing, 24
  private headers, 23
  section headers, 23
  symbol information, 24
object class, 256
object life cycle, in C++, 160–161
objects, in IDC language, 256–257
OEP (original entry point) recognition, 540
Offset column, 90
offset cross-reference, 172–173
OllyDbg, 540
OllyDump, 541
OMF libraries, 219
op_t (*ua.hpp*), datatypes for SDK, 293, 303, 387
opcode bytes, 202
opcodes (operation codes), 4
Open command, file loading, 45
Open Register Window menu item, 520
Open Subviews command, 57, 521
Open Subviews menu, 55, 60, 191
OpenRCE, 35, 280, 453, 499
OpenSSL cryptographic library, 215–216, 229
operand values, 303
operation codes (opcodes), 4
optimization, 428
Options checkboxes, 47
options for constants, formatting, 112
Options menu, Font menu, 519
optype_t constants, 388
OR operation, 458
ord function, 264
ord parameter, 364
ordinal number, 230
ordinary flow type, 62, 170
original entry point (OEP) recognition, 540
Original value field, 239
OS X
  console mode for, 194–196
  installing on, 37–38
OS X Mach-O binaries, 24
Other option, IdaPdf, 510
otool utility, 23–24
out function, 395–396
out instruction, 456
out_line function, 396
out_one_operand function, 394, 395, 397
out_register function, 396
out_snprintf function, 395
out_symbol function, 396
out_tagoff function, 396
out_tagon function, 396
out.cpp file, 394
OUTDIR variable, 366
OutLine function, 396
OutMnem function, 395
outop function, 394, 398
output generator, 380
Output window, 56, 60, 66, 469
OutputDebugString function, 546
OutputDebugStringA function, 559–560
outputter, for processor modules, 394–399
OutValue function, 396
overlapping windowing capability, TVision library, 190
overriding purged bytes, manually, 230
Overview Navigator, 54, 215
overview navigator, IDA desktop, 54

P

p suffix, 171
__p__environ library function, 425
-P<*password*> command-line option, 571
-p<*port number*> command-line option, 571
Pack database (Deflate) option, 52
Pack database (Store) option, 52
pack pragma, 136
packed data, restoring from, 53
PaiMei framework, 177
panning, in disassembly window, 62–63
para parameter, 308
parameters
  names, formal, 228
  naming, 102–103
  passing, 255
  recognition, automating, 277
Parameters option, debugger process options dialog, 572

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

- parsing errors, 258
- `Pascal` directive, 231–232
- Pascal-style strings, 71
- Password option, debugger process options dialog, 572
- patch application event, 484
- patch availability event, 484
- Patch Bytes dialog, 238
- Patch Program menu, 238–241
  - changing individual database bytes, 238–239
  - changing word in database, 239
  - using Assemble dialog, 239–241
- Patch Tuesday cycle, Microsoft, 476
- Patch Word dialog, 239
- `patch_byte` funtion, 304
- `patch_long` function, 304
- `patch_many_bytes` function, 304
- `patch_word` function, 304
- `patchable` parameter, 364
- `PatchByte` function, 262, 280, 458, 556
- `PatchDbgByte` function, 556
- *PatchDiff2*, 485–487
  - graphical function comparison, 487
  - Identical Functions, 486
  - Matched Functions, 486–487
  - Set Match dialog, 486
  - Set Match feature, 486
  - Unmatched Functions, 486–487
- `PatchDword` function, 262, 279
- patched files, 484
- patching binaries, 237–245
  - after discovering vulnerability, 484–487
  - Patch Program menu, 238–241
  - Produce File menu, 241–245
    - ASM files, 242–243
    - DIF files, 244
    - EXE files, 243–244
    - HTML files, 245
    - INC (include) files, 243
    - LST files, 243
    - MAP files, 242
- `PatchWord` function, 262
- `Patch`*`XXX`* functions, 262, 465
- pattern files, for FLIRT signature files, 219–221
- pattern-matching, 39, 212
- *pat.txt* file, FLAIR, 220
- Pause button, toolbar buttons, 522
- Pause Process option, Debugger menu, 516
- PC keyboard, vs. Mac keyboard, 194
- *pcap* file format, 366
- *pcap* loader, 366–372
- `pcap_file_header` structure, 366, 369
- `pcap_types` string, 368
- *pc.cmt* file, 234
- PDB (Program Database) file, 49
- PDF files, 509
- PDF Objects window, 510
- PE (Portable Executable) format, 8, 19, 45, 224, 410, 545
  - binaries, Windows, 462
  - files, Windows, 467
  - mode, for Bochs, 576–577
  - signatures, 224
- PE Sniffer utility, 19
- PE Tools, 18–19
- `pe_` prefix, 224
- *pe_*.pat* file, 224
- *pe_gcc.pat* file, 224
- *pe_sections.idc* script, 244
- *pe_vc.pat* file, 224
- PEB (process environment block), 462, 555–557, 576
- PEiD, 19–20
- *pe.ldw* (Windows PE loader), 45
- persistent data storage, in IDC language, 259–260
- persistent named objects, 259
- personal settings directory, 192
- *pe.sig* file, 421
- `pfn` pointer, 392
- `ph` variable, 292
- `phrase` field, 396
- piracy, Hex-Rays stance on, 33
- Pistelli, Daniel, 342
- PlayStation PSX libraries, Sony, 219
- *plb.exe* parser, 220
- *plb.txt* file, 220
- Please confirm dialog, 542
- PLT (procedure linkage table), 274
- `.plt` section, 478
- plug-in configuration values, Visual Studio, 328–329
- `PLUGIN` object, 316
- `PLUGIN_ENTRY` function, 344–345
- `PLUGIN_EXT` variable, 366
- `plugin_file` field, 330

- PLUGIN_FIX bit, 318
- PLUGIN_FIX flag, 319–320, 329
- PLUGIN_KEEP value, 319
- PLUGIN_OK value, 319
- PLUGIN_PROC bit, 319
- PLUGIN_PROC flag, 319–320
- PLUGIN_SKIP value, 319
- plugin_t class, 292, 316, 344, 359
- PLUGIN_UNL flag, 319
- PLUGIN_*XXX* constants, 316
- PLUGIN.flags, 318
- PLUGIN.init function, 319–320
- PLUGIN.run function, 319, 323
- PLUGIN.term function, 319, 322
- PLUGIN.wanted_hotkey, 331
- plug-ins, 315–346, 499–510
  - building, 324–329
  - class informer, 506–508
  - collabREate, 503–506
  - configuring, 330–331
  - for debugger, 536–538
  - directory for, 39, 288
  - event notification for, 321–322
  - execution of, 322–324
  - extending IDC with, 331–333
  - Hex-Rays, 500–502
  - IdaPdf, 509–510
  - IDAPython, 503
  - ida-x86emu, 506
  - initialization of, 320–321
  - installing, 329–330
  - life cycle of, 318–319
  - MyNav, 508–509
  - scripted, 344–346
  - user interface options for, 333–344
    - customized forms with SDK, 337–341
    - with Qt, 342–344
    - using SDK chooser dialogs, 334–337
    - Windows-only, 341–342
  - using during remote debugging, 574
  - writing, 316–324
    - comment member, 317
    - flags field, 317
    - help member, 318
    - init member, 317
    - run member, 317
    - term member, 317
    - version field, 317
    - wanted_hotkey member, 318
    - wanted_name member, 318
- plug-ins configuration file, 201
- Plugins menu, Edit menu, 485, 508
- *plugins.cfg* file, 201, 330
- PointerToRawData field, 353, 355
- polymorphism, 163
- pop instruction, 392, 436
- popa instruction, 459, 547–548
- popf instruction, 459, 564
- pop-up windows, tool tip–style, 129
- Port option, debugger process options dialog, 572
- Portable Executable (PE) format. *See* PE format
- positive deltas, 392
- POSIX wait function, 454
- POSIX-style regular expressions, 99
- PR_*xxx* flags, 385
- pragma pack directive, 150
- predecessor instruction, 177
- Predefined symbols section, 258
- Preferences dialog, X11, 196
- prefixes, for autogenerated names, 73
- Preserve case, 124
- preventing debuggers, 453–454
- PrevFunction function, 267
- Previous button, 45
- print function, 263
- Print recursion dots, 184
- print_type function, 164
- printable characters, ASCII, 27
- printf function, 87
- printf_line function, 397
- printf-style format string, 263
- Problems window, 76–77
- procedure linkage table (PLT), 274
- process control tools, 519
- process control, with debugger, 521–530
  - breakpoints, 522–526
  - stack traces, 528–529
  - tracing, 526–528
  - watches, 529–530
- process environment block (PEB), 462, 555–557, 576
- process image, 541–542

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

Process Monitor, 451
Process Options command, Debugger menu, 571
Process Stalker component, PaiMei framework, 177
process tracing, 454
ProcessDebugPort function, 557–558
processes, attaching debuggers to, 514–515
ProcessInformation parameter, 557
ProcessInformationClass parameter, 557–558
processor modules
- architecture of, 409–411
- building, 403–407
- customizing existing, 407–409
- and Python byte code, 378–379
- scripting, 411–412
- using SDK, 380–403
  - analyzer, 385–390
  - emulator, 390–394
  - initialization of LPH structure, 381–385
  - outputter, 394–399
  - processor notifications, 399–401
  - processor_t members, 401–403
  - processor_t struct, 380–381

processor notifications, 321
Processor Options button, 47
Processor options section, 203
Processor Type drop-down menu, 46
PROCESSOR_ENTRY function, 411
processor_t object, 320
processor_t struct, 380–381
processor_t.newprc notification, 402
procs directory, 39, 46
procs file, 403
proctemplate.py, 411
Produce File menu, 241–245
- ASM files, 242–243
- DIF files, 244
- EXE files, 243–244
- HTML files, 245
- INC (include) files, 243
- LST files, 243
- MAP files, 242

Produce file submenu, File menu, 177
Program Database (PDB) file, 49
program entry point, 8
programs, in IDC language, 257–258
pro.h, for API, 290, 293
project properties dialog, Visual Studio, 328
Project Selection dialog, CollabREate, 505
prologue, of functions, 85
Propagate option, Set Match dialog, 487
Properties dialog, 66
protected binary, Burneye, 467
pseudocode, Hex-Rays, 502
Pseudocode option, View menu, 500
pseudocode window, 501
ptrace API, 454
Public name option, for named locations, 104
purchasing IDA, 34
purecall function, 159
Purged bytes attribute, 116–117
Purged bytes field, 230
Push Data button, Set Memory Values dialog, 465
push instruction, 91, 392
push operations, 86
Push Stack Data dialog, 465
push statements, 275
pusha instruction, 459
pushf instruction, 459, 466
puts function, 181
.pyc files, 378, 393
Python byte code, 378–379
Python function, 481–482
Python interpreter, 378
Python script, 495, 549
Python website, IDAPython, 503
python_data function, 394
PYTHON_LAST constant, 382

Q

QApplication class, 343
qfopen function, 291–292
qfprintf function, 291
qnumber macro, 383
qsnprintf function, 291, 343
qstrlen function, 343
qstrncpy function, 291
qstr*XXXX* function, 291

QT namespace, 342–343
Qt port, 176
Qt socket classes, 504
QuickEdit mode, 191
QuickUnpack, 442
Quit action, 205
qwingraph graph viewer, 176
qword field, 140

R

r value, 98
radio buttons, 339–340
RCE forums, 35, 499
.rdata section, 355, 419
rdtsc instruction, 471–472
read cross-reference, 172
read function, POSIX, 363
readelf utility, 24
readlong function, 265
readshort function, 265
README file, tilib utility, 156
readme.txt file
  FLAIR, 219
  idsutils, 231
  SDK, 287, 380
readstr function, 265
read/write traces, 526
realcvt function, 401
rearranging blocks, in disassembly window, 64
reasons, for disassembly
  compiler validation, 7
  debugging displays, 7
  malware analysis, 6
  software interoperability, 7
  vulnerability analysis, 6–7
Rebase Program menu option, 351
Recent Scripts menu option, 250
Recent Scripts window, 250
recoverying source code, 5
recursive descent algorithm, 13
recursive descent disassembly, 11–14
  conditional branching instructions, 11
  function call instructions, 12
  return instructions, 12–14
  sequential flow instructions, 11
  unconditional branching instructions, 11
Recursive option, 183
recvfrom function, 498
Red Hat distributions, 219
redefine process, 436
referenced variables, stack frame view, 97
references, in C++, 165–166
Refresh memory command, Debugger menu, 579
reg.cpp file, 383
register names, naming, 105
register-renaming dialog, 105
registry key, Windows, 45
RegNames array, 383
RegOpenKey function, 127, 228–229
regular comments, 107
regular expressions, POSIX-style, 99
relationships, deducing between classes, 165
relative virtual address (RVA), 351–352
release binaries, vs. debug binaries, 428–430
Remote debugger configuration dialog, 573–574
remote debugging, 569–574
  attaching to remote process, 573–574
  exception handling during, 574
  using Hex-Rays debugging server, 570–573
  using scripts and plug-ins during, 574
Remove Function Tail option, 115
remove option (qwingraph), 194
Rename and Set Type option, 502
Rename option, context-sensitive menu, 102
renaming
  import table entries, 553
  locations, 104–105
renimp.idc script, 552–554
reopening, IDA database files, 52–53
REP prefix, 527
repair option, Database Repair dialog, 53
repeatable comments, 107–108
reporting bugs, 58
request_COMMAND function, 536

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

res->num field, 332
res->set_string, 333
Research & Resources forum, Hex-Rays, 288
Reset Desktop command, 57
Reset desktop option, Windows menu, 209
restarting IDA, after crashes, 52–53
restoring
  hidden messages, 44
  from packed data, 53
ResumeProcess macro, 533
RET instruction, 87
ret instruction, 91, 129
RET N variant, 117
return instructions, 12–14
return statement, 255–256, 466, 537
reversing engineer programs, 5
Rfirst function, 267
RfirstB function, 267
right-click options
  constants, 112
  data items, 121
  and name changing, 102
  in Segments window, 74
  in Signatures window, 75
  in Type Libraries window, 75
right-shift operator (>>), 253, 458
RISC-style architectures, 387
Rnext function, 267
RnextB function, 267
Roberts, J. C., 221
Rolles, Rolf, 378, 473
ROM images, 29, 348
RTCx, 428
RtlUserThreadStart function, 546
RTTI (Runtime Type Identification) implementations
  in C++, 163–164
  compiler differences for, 420
RTTICompleteObjectLocator structure, 164
rules, for working with malware in debugging environment, 543
Run button
  exception confirmation dialog, 565
  x86emu Emulator dialog, 464
Run command, 521
run function, 333, 536
run member, for plug-ins, 317
Run option, Debugger menu, 516
Run to Cursor button
  toolbar buttons, 522
  x86emu Emulator dialog, 463, 466
Run to Cursor command, in Burneye, 467
Run to Cursor option, Debugger menu, 516
Run Until Return button, toolbar buttons, 522
run_requests function, 536–537
runtime errors, 258
Runtime Type Identification implementations. *See* RTTI implementations
RunTo function, 532
Rutkowska, Joanna, 451
RVA (relative virtual address), 351–352

S

-S option (IDA), 197
Sabanal, Paul Vincent, 165
safeguarding key file, 34
sandbox environments, 443
Save Database dialog, 51
Save Desktop command, 57
Save Desktop option, Windows menu, 519
Save Disassembly Desktop dialog, 209
save_file function, 360, 365
Saved registers attribute, 116
savefile function, 265
ScreenEA function, 263, 272
Script cancellation dialog, 258
script de-obfuscation of binaries, 455–460
script entry dialog, 251
Script File option, File menu, 554
script-based behavior, 576
scripting, 249–284
  associating IDC scripts with hotkeys, 261
  for debugger, 530–535
  execution of, 250–251
  IDAPython, 280–281
  IDAPython examples, 282–284
  IDC examples, 270–280
    emulating assembly language behavior, 278–280

scripting (*continued*)
IDC examples (*continued*)
enumerating cross-references, 272–274
enumerating exported functions, 275
enumerating functions, 270–271
enumerating instructions, 271–272
finding and labeling function arguments, 275–277
IDC functions, 261–270
code cross-reference, 267
data cross-reference, 268
database manipulation, 268–269
database search, 269–270
dealing with functions, 266–267
disassembly line components, 270
file input/output, 264–265
manipulating database names, 266
reading and modifying data, 262–263
string-manipulation, 264
for user interaction, 263–264
IDC language, 251–260
error handling in, 258–259
expressions, 253
functions, 254–256
objects, 256–257
persistent data storage in, 259–260
programs, 257–258
statements, 254
variables, 252–253
loaders, 373–375
plug-ins, 344–346
processor modules, 411–412
using during remote debugging, 574
scripting functions, Hex-Rays, 532
SDK (software development kit), 285–314
API (Application Programming Interface), 289–314
header files, 290–294
iteration techniques using, 310–314
netnodes, 294–301
SDK datatypes, 302–303
SDK functions, 304
configuring build environment, 289
creating loader modules using, 358–360
creating processor modules using, 380–403
analyzer, 385–390
emulator, 390–394
initialization of LPH structure, 381–385
outputter, 394–399
processor notifications, 399–401
processor_t members, 401–403
processor_t struct, 380–381
directory layout
bin directory, 287
etc directory, 288
include directory, 288
ldr directory, 288
lib directory, 288
module directory, 288
plug-ins directory, 288
top-level directory, 288–289
functions, 587
IDC language cross-reference for, 585–608
implementation, IDC functions, 586–608
installing, 287
support, Hex-Rays, 58
sdk directory, 36
sdk_versions.h file, 293
search features, Search menu, 82
SEARCH_DOWN flag, 270
search.hpp, for API, 293
second-generation languages, 4
section:address portion, 110
SectionAlignment field, 352
SectionAlignment value, 352
SecureCRT, 193
segend function, 401
Segment Configuration dialog, 464
segment_t (*segment.hpp*), datatypes for SDK, 293, 302
segment-creation dialog, 353
segmented addresses, 169
segment.hpp file, 293, 307, 353

Segments button, x86emu Emulator dialog, 464
Segments window, 74, 543
segstart function, 401
SEH (structured exception handling) process, 472
  Chain plug-in, 566
  exceptions, Windows, 565
  handlers, 565–566
Select a debugger dialog, 516
Select Command dialog, CollabREate, 505
Select Debugger option, Debugger menu, 515–516, 548
SELinux, 38
semaphore, 438
semicolon (;) hotkey, 107
semicolon prefix, used for IDA comments, 107–108
Sequence of Bytes option, 99, 493
sequential flow instructions, 11
Set Breakpoint option, 463
Set Function Type command, 128, 579
Set Import Address Save Point option, 470
Set Match dialog
  PatchDiff2, 486
  Propagate option, 487
Set Match feature, PatchDiff2, 486
Set Match option, 487
Set Memory button, x86emu Emulator dialog, 464–465
Set Memory Values dialog, 465
Set node color to default option, 186
Set specific options button, 572
Set Video Mode menu option, Window menu, 191
set_idc_func_ex function, 331
set_idp_options function, 401
set_name function, 306
set_processor_type function, 410
set_reg_val function, 538
set_segm_addressing function, 363
SetArrayLong function, 301
SetArrayString function, 301
SetBptAttr function, 531
SetBptCnd function, 531, 554
SetRegValuefunction, 531
setting function type, 129
Setup Data Types dialog, Options menu, 121, 144
Setup long names button, 163
Setup option, Strings window, 458
Setup short names button, 163
Setup Strings window, 70–71
shared library, 516
sharing TIL files, 155–156
shell script (*#!/bin/sh*), 16
shellcode, 29, 495–498
SHIFT-down arrow, 243
SHIFT-up arrow, 243
Shiva ELF obfuscation tool, 453
Shiva process, 454
Shiva program, 434–435, 437, 442
shnames data member, 401
SHOW_SP option, 202
show_wait_box function, 323
SHOW_XREFS option, 202
shr instruction, 458
shrd instruction, 458
Shrink Struct Type option, Edit menu, 145
sidt instruction, 451
Siemens C166 microcontroller application, 349
sig directory, 39
.sig file, 214
sigmake documentation file, 221
sigmake.exe utility, FLAIR, 221
sigmake.txt file, 222
signature selection dialog, 214
signature selection, FLIRT, 214
signatures
  function type, 229
  generating, 39
  Signatures Window, 74–75
Signatures window, 74–75
Signed elements option, 126
signed shifts, 458
simple arithmetic instructions, 11
Simpleton file format, 373
simpleton loader, 361–366
simplex method, 230
Simplified Wrapper Interface Generator (SWIG), IDAPython, 503
Sirmabus, 420, 506
size field, 386
size parameter, 307

SizeOfRawData field, 354
sizer function, 334
sizer parameter, 334
Skip button, x86emu Emulator dialog, 463–464
Skochinsky, Igor, 165, 420, 507
slice operator, 253
sockaddr data structure, 69
socket descriptor, 489
SoftIce, 452
software breakpoints, 453, 523, 544, 546
software development kit. *See* SDK
software interoperability, reasons for disassembly, 7
Solaris 10 x86 system, 219
solid arrows, 65
Sony PlayStation PSX libraries, 219
sorting alphabetically, in Functions window, 82
source code recovery, 5
SPARC code, 410
sparse arrays, 259
splash screen, 44
sprintf function, 264, 273, 477
ssleay32.dll library, 232
SSLEAY32.idt file, 232
stack adjustments, 118
stack cleanup, 228
.stack database segment, 462
stack frames, 83–98
  calling conventions for, 85–89
  examples of, 89–93
  as IDA structures, 146
  local variable layout in, 89
  viewing, 93–98
Stack pointer option, 110
stack pointers, adjustments for, 118–119
Stack Trace command, Debugger menu, 528–529
stack traces, in debugger, 528–529
stack variables, 95, 102
Stack View window, 519
stack-allocated arrays, 132–134
stack-allocated structures, 138, 148
stack-based buffer overflow, 488
stack-manipulation operations, 11
standard calling convention, 87
standard structures, 151–154
standard template library (STL), 486
Start address attribute, 116
start function, 213, 443
Start Process option, Debugger menu, 516, 518
start symbol, 546
STARTITEM directives, 340
startup directory, FLAIR, 217, 224
startup routine, 224
startup signatures, 224–225
startup.bat file, 224
startup.idc, 577
statements, in IDC language, 254
static analysis, of malware, 6
static de-obfuscation of binaries, 454–472
  script-oriented, 455–460
  x86emu emulation-oriented, 460–472
    and anti-debugging, 471–472
    de-obfuscation using, 465–470
    features of, 470–471
    initialization of, 462
    operation of, 463–465
Static func attribute, 117
static keyword, 254–255
static libraries, for FLIRT signatures, 217–219
static linking function, 22
statically linked binaries, 178
stats netnode, 537
stdcall calling convention, 87, 118, 230, 294, 468
stdcall functions, 116, 228, 464, 467, 558–559
_stdcall modifier, 87
Step button, x86emu Emulator dialog, 463
Step command, 521
Step Into button, toolbar buttons, 522
Step Over button, toolbar buttons, 522
StepInto function, 532
StepOver function, 532–533
StepUntilRet function, 532
STL (standard template library), 486

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

Stop on debugging message option, Debugger Setup dialog, 546
Stop on debugging start option, Debugger Setup dialog, 546
Stop on library load/unload option, Debugger Setup dialog, 546
Stop on process entry point option, Debugger Setup dialog, 546
Stop on thread start/exit option, Debugger Setup dialog, 546
`STOP_CODE` constant, 383
storage, of bytes, 97
Store (Pack database) option, 52
`store_til` function, 369
`stosb` instruction, 458
`strcat` function, 253
`strcpy` function, 175, 253, 273, 477–478, 480
`strdup` function, 253
stream argument, 491
stream disassemblers, 28
string data configuration, 72, 123
string scanning, 70
strings
  C-style null-terminated, 122
  displaying in Strings windows, 70
  double-clicking, 70
  options for, 122–124
  Unicode, 99
  using on executable files, 28
  utility, 27–28
`strings` command, 71, 212
`strings` utility, 446
Strings window
  Display only defined strings option, 71
  Ignore instructions/data definitions option, 71–72
  overview, 70
`strip` utility, 18
stripping binary executable files, 18
`strlen` function, 264
`strstr` function, 264
`struc_t` (*struct.hpp*), datatypes for SDK, 293, 303, 306, 308, 311
`Struct Var` option, Edit menu, 147
struct.hpp (`struc_t`), datatypes for SDK, 303
struct.hpp, for API, 293
structure definition
  collapsed, 146
  empty, 143
structure members, enumerating, 311
Structure name field, Create Structure/Union dialog, 143
structure notation, 149
structure offset, applying, 147
structure selection dialog, 147
structure templates, using, 146–149
structured exception handling (SEH) process, 472
structures
  collapsing, 154
  expanding, 153
  fields, changing name of, 144
  formatting global variables as, 149
  master list of, 152
Structures window, 69, 142–143
stubs, 403–405
`substr` function, 264
successor instruction, 177
summary stack view, 97
superclass constructors, 164
support
  Hex-Rays support page and forums, 35
  IDA Palace, 36
  Ilfak's blog, 36
  official help documentation, 35
  OpenRCE.org, 35
  RCE forums, 35
`supset` function, 299
`supstr` function, 299
`supval` function, 299
supvals, 297–298
`swidth` component, 338
SWIG (Simplified Wrapper Interface Generator), IDAPython, 503
Switch Debugger menu, Debugger menu, 516
`switch` statements, compiler differences for, 416–420
Switch Thread option, Emulate menu, 471
switch variable, 417

symbols
appearing in comments, 175
dispalyed on Imports window, 69
global (external), 20
symbol-selection dialog, 113
Synchronize to idb option, 150
synchronizing activities, using CollabREate, 504
synchronous debugger function, 532
synchronous interaction, 536–537
system calls, 89

T

-t command-line argument (strings), 28
tabs, IDA desktop, 55
tags, 297
Take Memory Snapshot command, Debugger menu, 542
.tar file, 36
Target assembler, 243
target assembly language syntax, 243
TASM (Borland's Turbo Assembler), 9
TCP session, 496
TEB (thread environment block), 439, 462, 556, 565, 576
tElock program, 438, 440, 442
Tenable Security, 342
term member, for plug-ins, 317
term method, 536
term_output_buffer function, 395
Terminal application, Mac, 194
Terminal keyboard settings dialog, Mac, 195
terminal programs, Linux, 192
Terminate button, toolbar buttons, 522
Terminate Process option, Debugger menu, 517
text display, Linux, 192
Text option, Hex window, 67
Text Search dialog, 99
text searches, of database, 99
.text section, 241, 353, 355, 423
text view, switching to graph view, 185
text-mode user interface configuration file, 39
The initial autoanalysis has been finished message, 57, 211
third-generation languages, 4
third-party graph viewer, 176
this pointer, in C++, 156–157
This type of output file is not supported message, 243
thiscall calling convention, 88, 156
thread environment block (TEB), 439, 462, 556, 565, 576
thread information block (TIB), 556
Thread Local Storage (TLS) callback functions, 545–546, 556
ThreadInformationClass parameter, 559
Threads view, 519
thunk functions, 428–429
ThunRTMain function, 427
TIB (thread information block), 556
TIB[*NNNNNNNN*] database section, 565
til directory, 40
TIL files, 49
loading new, 155
overview, 154
sharing, 155–156
til2idb function, 367
tilib tool, Hex-Rays, 155
time stamp counter (TSC), 471
timelimit option, 194
tips and tricks, for IDA Desktop, 57
Title case, 124
TLS (Thread Local Storage) callback functions, 545–546, 556
tmainCRTStartup function, 426
to address, in cross-references, 168
toggling values, 520
tool tip–style pop-up window, 129
toolbar
area, IDA desktop, 53
arrangements, 208
buttons, 208, 521–522
configuration menu, 209
customizing, 208–210
Toolbars command, 53
tools
c++filt utility, 25–26
for deep inspection, 27–29
dumpbin utility, 25
for file classification, 16–20
ldd utility, 22–23
nm utility, 20–21
objdump utility, 23–24
otool utility, 24

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

Tools menu, PE Tools, 19
top-level directory, for SDK, 288–289
TouchArg function, 391
Trace buffer size option, Tracing Options dialog, 526
Trace checkbox, Breakpoint Settings dialog, 526
trace option, 526
Trace over debugger segments option, Tracing Options dialog, 528
Trace over library functions option, Tracing Options dialog, 528
trace_level parameter, 533
tracing, in debugger, 526–528
Tracing Options dialog, 526–528
trampoline, 493
translate function, 401
TriMedia libraries, 219
TSC (time stamp counter), 471
TTY console, 197
Turbodiff, 485
turn color off tag, 396
turn color on tag, 396
TVHEADLESS environment variable, 197
TVision library, 190
TVision port, 193
TVOPT settings, 193
tvtuning.txt, 193
two-digit hex values, 99
type component, 338
type field, 303, 338, 388
Type Libraries window, 75
typedef statement, 151
TypeDescriptor structure, 164
typeid operator, 163
typeinf.hpp, 293
typinf.hpp, 367

U

U hotkey, 119, 144
u_ana member, 385
u_emu member, 391
u_out member, 394
u_outspec function, 401
ua_next_*xxx* functions, 386
ua.hpp file, 293, 385
ui_notification_t constants, 305
uname command, 326
uncollapsing nodes, 187
uncompressing UPX binary, using emulator, 467
unconditional branching instructions, 11
Undefine option, 119, 435
undefine process, 436
undefining functions, 119
undetected string data, 72
undo command, absence of, 59
undo feature, 40
undocumented CPU instructions, 110
Ungroup Nodes option, 187
Unicode strings option, 71, 99, 447
universal unpacker, Hex-Rays, 550
Unix-style make files, 289
Unmatched Functions, PatchDiff2, 486–487
unsigned shifts, 458
untar archive, 37
upgrading, 34
uppercase letter codes, 21
UPX
- decompression routine, 547–548
- decompression stub, 442
- packer, 442
- program, 441, 548, 552–553

UPX-packed binaries, 540
Use “dup” construct option, 126
Use graph view by default checkbox, Graph tab, 55
Use option key as meta key checkbox, Terminal application, 194
USE_DANGEROUS_FUNCTIONS macro, 290
USE_STANDARD_FILE_FUNCTIONS macro, 291, 365
User cross-reference graph dialog, 183
user interface
- of IDA Pro, 40
- for plug-ins, 333–344
  - customized forms with SDK, 337–341
  - with Qt, 342–344
  - using SDK chooser dialogs, 334–337
  - Windows-only, 341–342

user interface notifications, 321
User xref charts, 182
User xref graph, 184
User Xrefs Chart option, Graphs menu, 182

__usercall calling convention, 431
user-generated cross-reference graphs, 185
utilities directory, 36

V

-v command-line option (debugging server), 571
va_arg macro, C++, 322
var_ prefix, 95
variables
  in IDC language, 252–253
  index values of, 132
  names, IDA-generated, 96–97
vc32rtf signatures, 75
vcsample file, 289
Veracode, 476
version field, 317, 385
version member, 359
versions, 33
vertices, 64, 168
VGA font, 193
View menu
  Cross References option, 477
  Enumerate Heap option, 471
  Pseudocode option, 500
View window, 530
viewing machine language bytes, 111
virtual addresses, 64
virtual functions, 157–160, 173
virtual machine-based obfuscation, 472–474
virtual repeatable comment, 108
VirtualAddress field, 353
VirtualAlloc function, 468, 477, 576, 578–579
virtualization
  detecting, 449–451
    processor-specific behavioral changes, 451
    specific behaviors, 450–451
    specific hardware, 450
    specific software, 450
  software, 449
virtualizing obfuscator, 442
Visual C++ compiler, Microsoft, 114
Visual Studio suite, Microsoft, 25
Visual Studio Win32 Application Wizard, 327
VMProtect, 442, 472
VMware Tools collection, 450–451
VPAGESIZE option, 202
vtables, in C++, 157–160
vulnerability advisory, 484
vulnerability analysis, 475–498
  analyzing shellcode, 495–498
  discovering vulnerabilities, 476–483
  exploit-development process, 488–495
    finding useful virtual addresses, 494–495
    locating instruction sequences, 492–494
    stack frame breakdown, 488–492
  handling after-fact discoveries, 483–487
  reasons for disassembly, 6–7
vulnerability discovery, 6

W

w suffix, 172, 447
Wait For Next Event (WFNE) flags, 532
wait_for_next_event function, 538
wanted_hotkey data member, 318, 330
wanted_hotkey value, 318
wanted_name data member, 318, 330
Warning function, 263, 272
warning function, 305
warnings, for loaders, 49
wasBreak function, 323
Watch Address dialog, 530
Watch List option, Debugger menu, 530
watch lists, 529
watch points, 529
watches, in debugger, 529–530
Weak name option, for named locations, 105
weak symbol, marking, 105
web server, Apache, 23
Welcome dialog, 44

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle

WFNE (Wait For Next Event)
    flags, 532
WFNE_CONT flag, 533
WFNE_SUSP event type, 533
Whittaker, Andy, 349
width characters, 395
width component, 338
Width field, 124
widths parameter, 336
wildcards, 205
Win32 Application Wizard, Visual
    Studio, 327
Win32 Project template, 327
win32_remote.exe server
    component, 570
win64_remotex64.exe server
    component, 570
wince_remote_arm.dll server
    component, 570
Windows
  console mode for, 191
  installing on, 36–37
  launching installer, 36
"Windows Anti-Debug Reference"
    article, 555–558
Windows Asynchronous Sockets
    techniques, 504
Windows calculator program, 25
Windows CE ARM, 517
Windows library handle, 468
Windows menu, Save Desktop
    option, 519
Windows PE binaries, ida-x86emu
    plug-in, 462
Windows PE file, manually loading,
    349–357
Windows PE loader (*pe.ldw*), 45
Windows registry key, 45
Windows SEH exceptions, 565
Windows SEH handlers, 565
wingraph32 application, 176
WinGraph32 window, 180
WinHelp-style help files, 204
wininet.dll file, 516
WinLicense, 442, 448
WinMain function, 422
WinMain variation, 421
Wireshark, 366, 451, 496
word. *See* 1 byte of storage (db)
Word function, 262
word-patching capability, 239
wrapper code, 180
write cross-references, 172
write traces, 526
write4 capability, 488
writelong function, 265
writeshort function, 265
writestr function, 265
ws2_32 networking library, 553

X

X Windows consoles, 193
X11, installing, 195
X.25-style network connection, 113
x86 code, 410
x86 compiler, 87
x86 hardware-debug registers, 472
x86 instruction, 204
x86 processor module, 47
x86emu breakpoints, 463
x86emu emulator, de-obfuscation of
    binaries using, 460–472
  and anti-debugging, 471–472
  de-obfuscation using, 465–470
  features of, 470–471
  initialization of, 462
  operation of, 463–465
x86emu Emulator dialog
  Jump To Cursor button, 464
  Push Data button, 465
  Run button, 464
  Run To Cursor button, 463, 466
  Segments button, 464
  Set Memory button, 464–465
  Skip button, 463–464
  Step button, 463
x86emu library function dialog, 469
x86emu plug-in, 461
x86emu Set Memory Values
    dialog, 465
.*xinitrc* file, 195
XML templates, 360
xmodmap command, 196
.*Xmodmap* file, 195

xmodmap utility, 195
xor instruction, 436
xrefblk_t structure, 283, 309, 312–313
xref.hpp file, 293–294, 309, 392
xrefs (cross-references). *See* cross-references
Xrefs From graph, 181–182
Xrefs To graph, 180–181
XrefsFrom generator, 283
XrefType function, 267-268, 273, 309
xterm, running, 193
xtol function, 264
*XXX*set function, 298
*XXX*val function, 298

Y

Y hotkey, 128
y variable, 91, 94
Yason, Mark Vincent, 165
Yes edge arrow, 62
You may start to explore the input file right now message, 57

Z

Zbikowski, Mark, 16
zoom control, keyboard, 62
ZwContinue function, 567

The IDA Pro Book, 2nd Edition
© 2011 by Chris Eagle