

Index

Note: Page numbers in italics refer to figures and tables.

- 3D graphics using OpenGL ES 217–18
- animating a scene 231–4
- adapting performance of 234–5
- applying a texture to a 3D object 228–9
- displaying a 3D object 222–6
- full-screen setup 219–22
- OpenGL to OpenGL ES, version mapping 218–19
- part-screen setup 229–31
- resources 235–6
- rotating a 3D object 227
- translating a 3D object 226–7
- active objects 65–6
 - active scheduler 70–2
 - CActive class 67–9
 - common problems with 72–3
 - preemption 66–7
 - TRequestStatus class 70
 - active scheduler 70–2
- AdoptFromXyz() methods 104–6
- advanced application deployment
 - automatic installation 326
 - custom install 326–7
 - embedded binaries 327
- advanced technologies
 - messaging, new protocol for 323–5
 - publish and subscribe API 319–20
 - server process, file sharing 320–1
 - for Symbian Partners 325–6
 - user interface features 321–3
- animation
 - GIF images, loading 209–11
 - of a 3D object 231–5
- application 329
 - distribution 341–9
 - internationalization 338–40
 - optimization 333–4
 - protection 334–6
 - signing digital certificate 336–8
 - testing 331–3
 - UI standard look and feel 329–31
- Application Binary Interface (ABI) 4
- AppTestPro, testing tool 332
- ARM hardware 4
- arrays 54
 - CArrayX 57
 - dynamic 55–6
- RArray classes, cleanup of 56–7
 - sorting and searching 57–8
 - static 54
- audio clips, playing 238–41
- audio operations, performing basic 241–2
- audio recording 253–5
- audio streaming 250–2
- audio tones, playing 242–3
- authorization server, software protection 335
- automatic installation 326
- background color, drawing 198–9
- battery status, retrieving 271–2
- binary data
 - reading from a file 83–5
 - writing to a file 80–3
- Bluetooth
 - advertise services 297–8
 - create a simple OBEX Server 295–7
 - discover devices in range 286–8

- discover services available
 - 288–90
- for OBEX communication
 - 294–5
- transfer data using Sockets API
 - 290–2
- buffer descriptors 46–7
- build configurations 13–14

- C classes 31–2
- CActive class 67–9
- Calendar 106–7
 - add new event 117–20
 - Calendar database 108–9
 - create repeating event 126–8
 - find out if you are available 133–4
 - get attendee list 134–6
 - modify an event 115–17
 - remove an event 120
 - send/receive events between different devices 124–6
- call barring status, retrieving
 - 281–2
- call forwarding status, retrieving
 - 280–1
- callbacks 74–5
- cameras
 - capturing still images from 258–60
 - recording video from 260–3
 - viewfinder 255–6
 - displaying 257–8
 - getting camera information 256–7
 - initializing camera 257
- capabilities 24
 - and data caging 62
 - platform security 60–1
 - for recipes 177
- capability-related tests 337–8
- Carbide.c++ 3
 - advanced topics 21
 - changing certificate/key pair 24–7
 - importing project files 24
 - modifying the project files 22–4
 - downloading the IDE 2
 - running the IDE 9–10
 - CArrayX classes 57
 - certificate/key pair, changing 24–7
 - character sizes 44
 - class conventions 30–3
 - cleanup stack 36–8
 - advanced information 38–40
 - command line tools 6
 - CommsDat API and database 137–8, 140
 - compilers 2
 - connectivity 282–3
 - Bluetooth devices, discovering 286–8
 - Bluetooth services
 - advertising 297–8
 - discovering 288–90
 - create simple OBEX client 294–5
 - over Bluetooth 295–7
 - discover infrared devices 284–6
 - print over IrDA 283–4
 - Sockets API 290–3
 - see also* networking
 - construction, two-phase 40–2
 - contacts 106–7
 - add a new contact 113–14
 - contacts database 107–8
 - find a contact 128–31
 - move contact to another group 131–3
 - read data from a contact 111–13
 - remove a contact 114–15
 - send/receive between different devices 122–4
 - sort contacts 121–2
 - write data to a contact 109–11
 - content aggregators, distribution 344
 - controller plug-ins 326
 - controls, drawing inside another container 206–7
 - custom installations 326–7
- data caging 62
- data types, fundamental 29–30
- deployment 326–7
- descriptor classes 44
 - APIs 50–3
 - base classes 45
 - character size 44
 - choosing descriptor type 52
 - class inheritance 46
 - derived classes 45
 - buffer descriptors 46–7
 - pointer descriptors 45–6
 - dynamic descriptors 47–9
 - function parameters and return types 53–4
 - literal descriptors 49–50
 - summary of 51
- developer certificates 24–7
- development communities 3
- development efforts 4
- digital certificates 336
 - Publisher IDs 336–7
 - Symbian Signed tests 337–8
- direct file stores 99
- direct screen access (DSA) 195, 216
- directory structures 6–7
- distribution 341
 - content aggregators 344
 - independent software channels 342–4
 - network operators 341–2
 - open source 345
 - pre-installed applications 344
- DoCancel() 69
- drawing *see* graphics and drawing
- dual-tone multi-frequency (DTMF) tones 243, 269
- dynamic arrays 55–6
- dynamic descriptors 47–9
- Dynamic Navigation Links (DNL) 322

- email 172–3, 177
 - classes for editing message settings 187
 - location in message store 174
 - retrieving addresses 134–6

- embedded binaries, deployment 327
- emulators 5–6, 7
 - configuring 15–16
 - running applications on 14–15
- error codes 35
- ESOCK (Sockets Server) 136–7, 282–98
- event handling 65–73, 195–6
- exception handling 33–6
- executable files 58–9
 - UIDs 59–60
- ExternalizeL() method 64–5

- file handling 78–9
 - file server session 79–80
 - file store, read from and write to 98–104
 - file stream, read from and write to 90–6
 - file stream, read and write class members 96–8
 - private folder path 86–90
 - read binary data from a file 83–5
 - read text from a file 85–6
 - resources 106
 - share files between processes 104–6
 - server file sharing 320–1
 - write binary data to a file 80–3
- File Server plug-in, deployment 325
- file stores 98–9
 - compacting 103
 - reading from 101
 - into an array 103–4
 - multiple streams 101–3
 - writing to 99–101
- filling shapes 198–9
- flicker-free graphics 208–9
- flight mode status, retrieving 274
- fonts, loading 205–6
- Forum Nokia Launchpad 347–8
- Forum Nokia PRO 348–9
- fundamental data types 29–30
- global positioning system (GPS) 299–317
- graphics and drawing 194–6
 - background color/fill a shape 198–9
 - controls, drawing multiple 206–7
 - direct screen access, drawing with 216
 - double-buffering for flicker-free graphics 208–9
 - drawing outside Symbian application framework 213–15
 - fonts, loading 205–6
 - GIF animation images, loading 209–11
 - image with transparent section, drawing 200–1
 - lines and shapes, drawing 197–8
 - load and draw MBM/MIF images 199–200
 - load JPG or PNG images 201–3
 - skins as backgrounds, drawing 211–13
 - text drawing to the screen 203–5
 - see also* 3D graphics using OpenGL ES
- GUIDs 117–20
- HBufC, dynamic descriptor 47–8, 52–3
- heap size 62–3
- Hello World project
 - building 13–14
 - generating 10–13
 - modifying 18–22
 - running on the device 15–18
 - running on the emulator 14–16
 - template 9
- HTTP framework 141
 - HTTP GET request 150–3
 - HTTP POST request 164–7
- HTTP proxy information, retrieving 169–71
- setting advanced HTTP properties 167–8

- images
 - drawing with a transparent section 200–1
 - loading/drawing MBM or MIF 199–200
 - loading JPG or PNG 201–3
 - see also* graphics and drawing
- importing project files 24
- independent software channels, distribution 342–4
- InetProtUtils 142
 - loading JPG or PNG 201–3
- Infrared Data Association (IrDA)
 - create an OBEX client over 294–5
 - discovering devices 284–6
 - printing a document to an IR printer 283–4
 - transfer data over 291–2
- initialization 40–2
- installation 1–2
 - automatic 326
 - custom 326–7
 - post-installation 6–7
 - SIS file 16–18
- Integrated Development Environment (IDE) 2
 - running 9–10
- Inter Process Communication (IPC) 320–1
- InternalizeL() method 64–5
- International Mobile Equipment Identity (IMEI), retrieving 2721
- internationalization of application 338–40
- Internet Access Point (IAP) 138, 140, 144
 - force a connection to use a specific IAP 146–8
- Internet protocols 136–7
- Internet, selling application over 335

- leaves 33–4
 - causes of 35–6
 - new (ELeave) 36
 - TRAP and TRAPD macros 34–5
 - versus panics 34
- libraries
 - adding to project file 23
 - messaging 176
- lines and shapes, drawing 197–8
- literal descriptors 49–50
- localization 338–40
- location-based services (LBS) 299–302
 - list positioning technology modules 302–4
- location information, requesting 310–15
 - extended information 315–17
 - receive module status change notifications 305–8
 - retrieve current module status information 304–5
 - set module selection criteria 308–10
- M classes 32
- mail store debugging 176
- memory management 40–2, 56–7
- menus
 - adding a new menu item 18–20
 - handling a menu event 20–2
- Message Type Module (MTM)
 - plug-in 323–5
- messaging 172
 - applications, tips for writing 176
 - copy a message 188–9
 - delete a message 191–2
 - edit message 185–6
 - folder creation 180–1
 - handle incoming messages 192–4
 - initialize application to use 177–80
 - message creation 182–3
 - message store
 - structure and main classes 173–4
 - types of entry 174–5
 - Message Type Module (MTM)
 - plug-in 323–5
 - move a message 189–90
 - read message details 183–5
 - resources 194
 - retrieve and edit message settings 187
 - send a message 190–1
 - SendAs server 173
 - services 173
 - supported bears 172–3
- MIDI files, playing 243–4
- MMS (Multimedia Message Service) see messaging
- mobile development concepts 4
- multi-image files (MBMs or MIFs) 199–200
- multimedia 236–8
 - additional format streaming 325–6
 - audio clip, playing 238–41
 - audio recording 253–5
 - audio streaming 250–2
 - audio tone, playing 242–3
 - basic audio operations 241–2
 - camera, capturing still images from 258–60
 - camera viewfinder, displaying 255–8
 - default file storage location 244–7
 - MIDI file, playing 243–4
 - phone calls, recording 255
 - resources 263
 - video clips
 - playing 247–50
 - saving to file 260–3
- multimedia framework (MMF) 236–8
 - controller plug-in 326
- network information 279–80
- network name, retrieving current 273
- network operators, distribution 341–2
- network registration status, retrieving 275
- network signal strength, retrieving 271
- networking 136–7
 - active connection information, retrieving 161–2
 - advanced HTTP properties, setting 167–8
 - APIs needed 137–8
 - connection status, monitoring 159–61
 - domain name, resolving 148–50
 - environment setup 138–9
 - SDK configuration tools 139
 - WinPCap 139–40
 - WinSock and WinTAP 141
 - forcing connection to use specific bearer 145–6
 - forcing connection to use specific IAP 146–8
 - HTTP framework 141
 - HTTP GET request 150–3
 - HTTP POST request 164–7
 - HTTP proxy information, retrieving 169–71
 - InetProtUtils 142
 - listen for incoming connection using TCP 157–9
 - resources 171–2
 - secure socket connection, using 162–4
 - send/receive data using TCP sockets 142–4
 - URI creation 154–7
 - URI filename, extracting 168–9
 - URI parsing 153–4
- new (ELeave) 36
- NewL() and NewLC() 42
- Nokia Software Market 343–4
- Object EXchange (OBEX)
 - create a simple client 294–5
 - create simple Server over Bluetooth 295–7

- off-screen bitmaps 208–9
- open source 4, 338, 345
- OpenGL ES 217–18
 - mapping to OpenGL version 218–19
 - see also 3D graphics using OpenGL ES
- operator name, retrieving current 273–4
- optimization of application 333–4
- OTI (over the Internet) purchasing 335, 342–3

- panics 34, 38–9
- permanent file stores 99–103
- personal information management (PIM) services 106–36
- phone calls
 - handling 268–9
 - recording 255
- phone line state, retrieving 269–70
- phone lock status, retrieving 278–9
- phone numbers
 - identifying contact associated with 276–7
 - retrieving from incoming/outgoing call 276
 - setting restrictions 268
- platform security capabilities 60–1
- pointer descriptors 45–6
- positioning technology see location-based services
- preemption 66–7
- private folder path 86–90
- projects
 - building 13–14
 - creating new 10–13
 - importing project files 24
 - modifying project files 22–4
 - running 14–16
- protection of application 334–6
- publish and subscribe API 319–20

- Publisher ID digital certificates 336–7
- PushL () 38

- R classes 32
- RArray classes, cleanup of 56–7
- RBuf, dynamic descriptor 48–9
- release of application
 - distribution 341–9
 - what to do before 329–40
- resources 345–6
 - 3D graphics 235–6
 - Development Communities 3
 - file handling 106
 - messaging 194
 - multimedia 263
 - networking 171–2
 - Symbian Press books 346–7
- RSS and RLS resource files, internationalization 339–40
- RThread class 73–4
- RunError() 69
- RunL() 67–9, 71, 73–4
- running an application 14–16
- RVCT (Real View Compilation Tools), optimizing code 334

- S60 platform 5
 - advanced UI features 322
 - configuring emulator 15
- SDKs 2
 - configuration tools 139
 - directories structure 6–7
- searching
 - arrays 57–8
 - for Bluetooth services 288–90
 - for devices in range 286–8
 - finding a contact 128–31
 - for infrared devices in range 284–6
- security
 - network 162–4
 - platform 60–1
- selling application over the Internet 335, 342–4
- serialization 96–8
- shapes and lines, drawing 197–8

- sharing files between processes 104–6, 320–1
- signing digital certificate into application 336–8
- SIM card, retrieve IMSI number of 277–8
- SIS installation files 16–18, 326–7
- skins, drawing as backgrounds 211–13
- SMS (Short Message Service)
 - activation of software 334–5
 - see also messaging
- Sockets API, using to transfer data over Bluetooth 290–3
- sorting
 - arrays 57–8
 - contacts in database 121–2
- stack-based buffer descriptors
 - TBufC and TBuf 46–7
- stack size 62–3
- static arrays 54
- static classes 33
- static code analysis 333
- stray signal panics 72
- streams 63–4
 - appending to a file stream 93–4
 - detecting end of file 94–5
 - externalize and internalize 64–5
 - read and write class members 96–8
 - reading from a file stream 92
 - writing to a file stream 90–2
- string classes see descriptor classes
- Symbian C++ recipes 77–8
 - 3D graphics using OpenGL ES 217–36
 - connectivity 282–98
 - contacts and Calendar 106–36
 - file handling 78–106
 - graphics and drawing 194–216
 - location-based services (LBS) 299–317
 - messaging 172–94
 - multimedia 236–63

- Symbian C++ recipes (*continued*)
 - networking 136–72
 - telephony 263–82
- Symbian Developer Network 345–6
- Symbian Ecosystem, partnering
 - within 347–9
- Symbian OS system model 345
- Symbian Partners 347
 - advanced features available to 325–6
- Symbian Press books 346–7
- Symbian Signed Testing 337–8
- SymbianOSUnit, testing tool 332
- system requirements 1–2

- T classes 30–1
- TBufC and TBuf, stack-based buffer descriptors 46–7, 52–3
- TCP sockets
 - listening for incoming connection using 157–9
 - send/receiving data using 142–4
- TDes and TDesC, descriptor classes 45
- telephony
 - accessing services 263–8
 - battery status 271–2
 - call barring status 281–2
 - call forwarding status 280–1
 - current network name 273
 - current operator name 273–4
 - flight mode status 274
 - identifier number (IMEI) of device 272
- IMSI number of SIM Card 277–8
- network cell information 279–80
- network registration status 275
- network signal strength 271
- phone calls
 - handling 268–9
 - recording 255
- phone line
 - monitoring state of 269–70
 - sending DTMF tones 269
- phone lock status 278–9
- phone numbers
 - match a name to 276–7
 - restrictions on showing 268
 - retrieving from incoming/outgoing call 276
- templates
 - for projects 9
 - thin templates 42–4
- testing the application 331–3
- TestQuest Pro, testing tool 332
- text
 - drawing to the screen 203–5
 - reading from a file 85–6
- text messaging *see* messaging
- thin templates 42–4
- threads
 - and active objects 65–73
 - multi-threaded code 73–4
- time zones 109
- timers 74–5
- tones 242–3, 269
- TPtrC and TPtr, pointer descriptors 45–6, 51–2
- TransferToXyz() methods 104–6, 320–1
- TRAP and TRAPD macros 34–5
- TRequestStatus class 70

- UIDs 12–13, 59–60
- UIQ 3 platform 6
 - advanced UI features 322–3
 - configuring emulator 16
 - enabling IP networking 139
- UIQ Alliance 349
- Universal Resource Identifier (URI)
 - creating 154–7
 - extract local filename from 168–9
 - InetProtUtils API 142
 - parsing 153–4
- user interface (UI)
 - advanced platform-specific 321–3
 - standard look and feel 329–31
 - unresponsive 73

- vCal Format, Calendar 124–6
- vCard Format, contacts 122–4
- VFAT (Virtual File Allocation Table) file system 78–9
- video clips
 - playing 247–50
 - recording 260–3

- WinPCap 139–40
- WinSock 141
- WinTAP 141
- workspaces 3, 9–10