
Index

Note: Page numbers ending in “f” refer to figures. Page numbers ending in “t” refer to tables.

A

abstract classes, 38-45
Abstract Syntax Notation.1, 372
Abstract Window Toolkit (AWT), 70
account credentials, 84
actionPerformed() method, 66
Adaptive Server Anywhere. *See* ASA
addCookie() method, 118
addingService() method, 71-73
addMovieRating() method, 93
addPartToScreen() method, 154
addProperty() method, 311
Advanced Encryption Standard. *See* AES
Advanced Graphics, 24, 24t
AES (Advanced Encryption Standard),
371, 384-87
AESDecrypt() method, 389
AESEncrypt() method, 389
AESEngine class, 384
AESFastDecrypt() method, 386
AESFastEncrypt() method, 386
AESFastEngine class, 384, 386
AESLightEngine class, 384
Agea, 12
Amazon literal XML service, 292
Amazon Web Services, 291-94
AmazonLite class, 294, 298
AMS (Application Management Software),
52, 55, 419-20, 429
Anderson, Jim, 378
ant command, 35-36, 167
ant run command, 36, 167
Antepo J2ME Jabber library, 183

AOL (America Online), 164, 179
Apache Axis, 177, 328, 342, 346, 351
Apache Axis facade, 351-53
API (Applications Programming Inter-
face)
compatibility of, 128-29
and GUI, 20
and OSGi, 55
support of, 177, 178
Application Management Software. *See*
AMS
application provisioning, 54
application services, 53
Applications Programming Interface. *See*
API
architecture solution, 18-19
arrayType array, 323
ASA (Adaptive Server Anywhere), 225,
260
ASN.1 (Abstract Syntax Notation.1), 372
asynchronous messaging, 141, 167, 170-
71
AT&T, 12, 16
Authorization header, 119
Authors tag, 295
AvantGo, 5, 6, 12
AWT code, 70
AWT event handler, 65-66
AxisFacade class, 351-53

B

B3 Security, 376-78
B3 Tamper Detection, 376-78
B3E2E (B3 End-to-End), 376, 378
B3Sig (B3 Digital Signature), 376-77
backend applications, 17, 132, 174-79
backend databases, 271-81

- direct database access, 273
- J2ME SDK, 273-74
- JDBC API, 273
- legacy applications, 274-78
- middleware, 273
- remote database access schemes, 273
- remote database connections, 272-73
 - synchronization, 250-53, 259-60
- BackgroundTask class, 42-45
- Ballard, Kenneth, 122
- Base64 content, 307
- BasicAuthHandler class, 120-21
- battery technology, 137
- BC (Bouncy Castle), 371-72, 375, 383-87, 393-95, 399-402, 411-12
- BEA, 334
- beginConversation() method, 183
- binary data exchange, 46-48
- binary RPC over HTTP, 103-6
- Binary Runtime Environment for Wireless (BREW), 25
- BlackBerry, 130, 132, 148
- BlackBerry Enterprise Server, 158-60, 161t
- blockArrived() method, 183
- Bluetooth API, 22t, 257
- BMW, 55
- bodyIn variable, 329
- bodyOut variable, 329
- BookDetails object, 295, 297
- Books object, 297
- Books vector, 295
- Borland, 334
- Bouncy Castle. *See* BC
- BREW (Binary Runtime Environment for Wireless), 25
- browsers, 70, 77
- BufferedReader, 134
- BufferedWriter, 134
- build.xml script, 35
- BundleActivator class, 62, 65, 70
- BundleActivator interface, 70
- business to business (B2B), 5-6
- business to consumer (B2C), 5
- business to employee (B2E), 6-7
- ByteArrayInputStream class, 114
- ByteArrayOutputStream class, 114
- Bytes format, 241

C

- C/C++, 227, 353
- call() method, 311
- call model, 38-42
- CallableStatement interface, 221-23
- Cancel button, 44
- CBS (Cell Broadcast Short Message Service), 164, 165, 173
- CDC (Connected Device Configuration), 21, 23t-24t, 55, 59, 216-37, 434
- CDMA (Code Division Multiple Access), 164
- CDSECT event, 290
- Cell Broadcast Short Message Service. *See* CBS
- cell phone locations, 358
- cell phones, 133
- cellular networks, 357-58
- chained data handlers, 98-102
- Cingular, 165, 334
- ClassMap object, 318, 319, 321, 324
- CLDC (Connected Limited Device Configuration), 20, 21, 129
- clientside threading model, 106-9, 106f
- close() method, 72, 131
- Code Division Multiple Access (CDMA), 164
- code redundancy, 54
- CodeWarrior Wireless Studio, 316
- commandAction() method, 142, 420
- CommandListener class, 142, 420
- COMMENT event, 290
- commercial email, 156-58
- commitMovieRatings() method, 93
- Common Public License (CPL), 149
- ConfirmTicketUI class, 89-92
- Connected Device Configuration. *See* CDC
- Connected Limited Device Configuration. *See* CLDC
- Connection class, 273
- Connection object, 218-19, 240
- connectionTerminated() method, 183
- Connector class, 156
- Connector.open() method, 124, 166, 184, 420
- Contact class, 156
- Contact Manager application, 230-36, 230f, 263-68, 264f
- ContactList class, 156

- container-managed applications, 52-55, 54f, 74-77. *See also* frameworks
 - converged messaging, 179-84
 - CookieHandler class, 118-19, 120
 - cookies, 117-19
 - cookies header, 117
 - corporate portal servers, 158-60
 - crackers, 121, 123
 - CREATE PUBLICATION statement, 261
 - CREATE SYNCHRONIZATION SUBSCRIPTION, 261
 - CREATE SYNCHRONIZATION USER, 261
 - cross platforms, 16, 18-19
 - CryptoEngine class, 382-87, 393, 399, 401, 403, 407
 - cryptography, 368-79. *See also* encryption; security
 - cryptography tasks, 382-418, 382t, 383t
 - custom services, 53
 - CustomConnection class, 112-14
 - CustomConnection decorator, 112-13
 - CustomConnector class, 112-13
 - CustomConnector.open() method, 112
- D**
- Data Encryption Standard. *See* DES
 - data exchange protocol, 46-48
 - Data Representation Simplicity XML, 273
 - databases
 - backend access, 271-81
 - for CDC, 216-37
 - benefits of, 216-17
 - database on the go, 216-17
 - DB2e, 228, 229t
 - example application, 230-36
 - executing SQL statement, 219-21
 - extracting search results, 219
 - JDBC example, 217-18
 - JDBC Optional Package, 223-24
 - obtaining connection, 218-19
 - Oracle9i Lite, 228-29
 - PointBase, 229, 230
 - stored procedures, 221-22
 - UltraLite custom database, 226-27
 - using JDBC, 217-19
 - FastRecordStore, 243-46, 257, 259-60
 - for MIDP, 239-47
 - FastRecordStore, 243-46, 259-60
 - Oracle J2ME SDK, 241-43, 246
 - PointBase Micro Edition, 240-41, 246
 - SODA, 242t, 243-46
 - SQL script, 244
 - synchroization, 249-69
 - architecture of, 250-53, 251f
 - engines, 132, 138, 213-81
 - sources, 137, 253-63
 - DataInputStream class, 46
 - DataOutputStream class, 46
 - Date format, 241
 - Date object, 322
 - DateTime string, 220
 - DB2 databases, 228, 253, 273
 - DB2 Everyplace. *See* DB2e
 - DB2 Universal Database, 257
 - DB2e, 199, 228, 229t, 243-46, 257-60
 - DB2e Sync server, 257-60, 258f
 - DBManager class, 230-36, 265
 - Decimal format, 241
 - decorator classes, 112-14, 117-18
 - decryption, 369, 383, 386-92, 399-409
 - DES (Data Encryption Standard), 375, 384, 395
 - destroy() method, 131
 - Details tag, 294, 295
 - Deutsche Telekom, 55
 - developer opportunities, 19
 - device security. *See* security
 - DeviceTop framework, 70
 - digest. *See* hashes
 - DigestAuthHandler class, 122-23
 - Digital Signal Processor (DSP), 375
 - digital signature, 409-17, 410f
 - Digital Signature Algorithm. *See* DSA
 - disk files, 204
 - Display.getDisplay() method, 420
 - dispose() method, 131
 - DOCDECL event, 290
 - DoCoMo, 10
 - DOM (Document Object Model), 291, 291f
 - Domino, 257
 - DriverManager class, 273
 - driving directions, 348-61, 349t, 354f
 - driving directions application, 353-61

- DSA (Digital Signature Algorithm), 375, 411, 413-15
- DSP (Digital Signal Processor), 375
- dynamic services, 58-59, 78
- E**
- E911, 358
- EAIF (External Application InterFace), 177
- ECC (Elliptic Curve Cryptography), 411
- Echelon, 55
- echo bundles, 59-69, 63f
- echo consumer manifest file, 69
- echo service example, 59-69
- EchoActivator class, 62, 64-65
- EchoFrame class, 65, 67-69
- EchoFrame event handler, 69
- EchoService bundle, 61-65
- EchoService interface, 62-63, 66
- EchoService object, 65
- EchoServiceImpl class, 62-64
- EchoUIConsumer bundle, 65-70
- EchoUIConsumer class, 65-67
- EchoUIConsumerBundle.stop() method, 69
- EchoUIConsumer.start() method, 65
- EchoUIConsumer.stop() method, 66
- Eclipse, 316
- ECs (executive committees), 17, 18t
- EJB (Enterprise JavaBean), 53, 95, 103
- Element array, 329
- Elliptic Curve Cryptography, 411
- email, 147-62
 - APIs, 149
 - attachments, 151, 154
 - basics of, 148-49
 - commercial email, 156-58
 - deleting messages, 153
 - displaying details, 153-55
 - and HTTP proxy, 151-52
 - life-cycle of, 148, 149f
 - and MIDlet Suite, 150-53
 - MIDP specifications, 150
 - outgoing message, 151
 - and PIM suites, 148, 156-58
 - retrieving messages, 148, 149f, 152-53
 - sending and receiving, 148, 149f, 150-53
 - servers, 148-49, 149f
- Embedded Linux, 12, 70, 199
- encryption, 85, 123-24, 135, 137, 205, 367, 369-71, 375, 382-418, 384f. *See also* cryptography; security
- end-to-end best practices, 127-44
 - impatient user, 141-43
 - deployment descriptors, 143
 - rich UI clients, 141
 - screen flow process, 143
 - threads usage, 142
 - user preferences, 143
 - limited device hardware, 131-32
 - backend services, 132
 - bundled services, 132
 - closing connections, 131
 - design patterns, 130-32
 - lightweight libraries, 128-29
 - messaging servers, 132
 - minimizing garbage collector, 130-31
 - mobile portals, 131-32
 - native libraries, 131
 - packaging process, 129
 - partitioning applications, 129
 - reducing footprints, 129-30
 - reusing objects, 130
 - using arrays, 130-31
 - using strings, 130-31
 - pervasive devices, 136-38
 - centralized provisioning, 138, 139f
 - encryption, 137
 - on-device data, 137
 - optimization, 137
 - synchronization, 137
 - slow networks, 132-36
 - buffers, 134
 - encryption, 135
 - mobile portals, 134, 135f
 - offline modes, 133
 - remote facades, 134
 - server status, 136
 - ubiquitous integration, 138-41
 - comparison chart, 139t
 - integration schemes, 140f
 - messaging, 141
 - mobile RPC framework, 140
 - XML Web Services integration, 141
- END_DOCUMENT event, 290

END_TAG event, 290
Enhanced 911 (E911), 358
Enterprise JavaBean. *See* EJB
enterprise messaging, 187-211
 iBus//Mobile JMS, 195-99
 client libraries, 195, 197
 connection initialization, 196, 197
 and email, 198
 non-programmable clients, 198-99
 receiving messages, 198
 sending messages, 197
 and SMS messages, 198
 and WAP, 199
 JMS, 190-95
 combining messaging models, 194
 point-to-point model, 193-94, 193f
 publish-and-subscribe model, 190-93, 190f
 top-level interfaces, 190, 191t
 MOM, 188-89
 quality-of-service, 189
 queues, 193, 200-208
 reliability, 188
 universal integration, 188
 WebSphere MQe, 199-210
 administration queue, 204-8
 communications adapters, 209-10, 209t, 210t
 HelloWorld example, 200-204
 mobile MOM solution, 199
 queue manager, 200-204
 storage adapters, 204, 205t
 enterprise mobility eco-system, 10-11
Enterprise Server, 132
ENTITY_REF event, 290
Ericsson, 334
Espial DeviceTop, 70
Espial Suites, 157
Espresso UI library, 70
Event class, 156
EventDispatcher class, 89, 106
EventDispatcher.run() method, 89
EventList class, 156
Exchange, 257
Executive Committees (ECs), 17, 18t
Export-Package attribute, 56
eXtensible Markup Language. *See* XML
External Application InterFace (EAIF), 177

F

facade patterns, 93-98, 132
FastRecordStore class, 243-46, 257, 259-60
FedEx, 6
feedback, 33-34. *See also* iFeedBack
Field Reader module, 279
Fielding, Roy, 292
FileSystemListener interface, 156
FileSystemRegistry class, 156
Flash, 77
Form module, 278
Foundation Profile. *See* FP
Fowler, Martin, 88
FP (Foundation Profile), 23t, 59, 128-29, 434
frameworks, 53, 54. *See also* container-managed applications
France Telecom, 165
freedom economy, 4, 13
frontend applications, 17
Fujitsu, 16

G

game profile, 23t
gateway servlet, 273, 277-78, 279f
gateways, 77-79, 78f, 341, 343f
GCF (Generic Connection Framework), 156, 175, 184
generateAESKey() method, 384, 387, 388
GenerateAllKeys class, 385, 403, 407
generic binary data, 46-48
Generic Connection Framework. *See* GCF
Geographic Information System (GIS), 346-47
GET operation, 136
getAttributeValue() method, 290
getBody() method, 318
getBookDetailsViaDOM() method, 298-99
getBookDetailsViaPull() method, 294-97
getBooksViaDOM() method, 298-99
getBooksViaPull() method, 294-96
getBytes() method, 241
getCookie() parses, 118
getDirections() method, 353
getDSAalgID() method, 414
getMessageList() method, 152
getMovie() method, 101-2

getPort() method, 124
 getResult() method, 318
 getSecurityInfo() method, 124
 getTheaters() method, 104
 GIS (Geographic Information System), 346-47
 Global Systems for Mobile Communications. *See* GSM
 go() method, 46
 Google search, 326-27
 Google spell check, 311-16, 318, 325-26, 329-30
 GPS device, 357-58
 graphical user interface. *See* GUI
 GSM (Global Systems for Mobile Communications), 164, 176, 379
 GUI (graphical user interface), 20, 70, 77

H

handlers chain, 114
 hashes, 121, 376-78, 409-10, 410f
 Hashing for Message Authentication (HMAC), 376-78
 HostConnection module, 278
 HSQL Database Engine, 224, 225f
 HTML, 70, 133
 HTTP Authentication headers, 366
 HTTP front end, 70-77, 71f
 HTTP headers, 112, 117
 HTTP protocol, 46, 52, 136, 138-39, 257
 HTTP proxy, 150-52
 HTTP services, 70-77, 71f, 77, 150, 292
 HTTP techniques, 111-25

- basic authentication, 119-24
- decorator approach, 112-14
- HTTPS protocol, 121-25, 124f
- importance of, 112
- process-chain approach, 114-17
- secure HTTP, 123-25
- session tracking with cookies, 117-19

 HttpClient class, 114-17, 120, 122
 HttpClient framework, 114, 115t, 118
 HttpClient source code, 114-17
 HTTPCommunicationHandler class, 100-104
 HttpConnection class, 124
 HttpConnection interface, 112
 HttpRequest object, 73
 HttpResponse object, 73
 HTTPS protocol, 123-25, 124f, 135, 366, 367, 378
 HTTPS support, 124-25
 HttpsConnection interface, 124
 HttpsConnection object, 124
 HttpService object, 71-72
 HttpTransport class, 313-16, 318
 HttpTransport object, 311
 HttpTransport.call() method, 312-13, 329
 hubs-and-spokes synchronization, 253-57, 254f
 Hypersonic SQL database, 224

I

I/O methods, 46
 IAIK JCE-ME, 373-75, 374t, 387-88, 396, 402-4, 412
 iAnywhere, 158
 iAnywhere Solutions, 224-26
 iAnywhere Studio, 260-62
 IBM, 6, 25, 55, 129, 132, 134, 158, 179, 188, 195, 228, 243, 252, 257, 316, 334
 iBus//Mobile JMS, 129, 188
 IDEs, 129
 IETF, 148, 151
 iFeedBack, 31-49

- application, 36-38, 37f
- architecture of, 34-35, 34f
- authentication process, 34-35, 367
- components of, 34-36
- data exchange, 46-48
- deployment of, 35-38
- implementation of, 38-48
- introducing, 33-35
- threading model, 42-46, 42f, 43f
- UI call model, 38-42
- usage scenarios, 35-38

 IGNOREABLE_WHITESPACE event, 290
 IM AIPs, 22t
 IM systems, 164, 179-84
 IMAP (Internet Message Access Protocol), 148-49, 149f, 152
 Import-Package attribute, 56
 IN parameters, 221, 222
 InboxClient object, 152
 InboxDemo, 150
 INSERT statement, 244-45, 261

Insignia, 25
instant messaging systems. *See* IM systems
Institute for Applied Information Processing and Communications (IAIK), 373
Intel, 11
Internet email. *See* email
Internet Message Access Protocol. *See* IMAP

J

J2EE (Java 2 Enterprise Edition)
 advantages of, 17
 containers in, 53
 developer opportunities, 19
 Executive Committee, 17, 18t
 provisioning server, 138, 139f
J2ME (Java 2 Micro Edition)
 advantages of, 16-17
 API compatibility, 128-29, 334
 architecture of, 20
 basics of, 419-30
 BREW, 25
 building process, 426-29, 427f
 code example, 421-25
 components of, 20-21, 21f
 crypto recipes, 381-418
 driving directions application, 358-61
 enterprise-readiness, 15-27
 explanation of, 20-24
 Generic Connection Framework, 156
 JAX-RPC API, 336-41, 338f
 JCP, 17-18, 24
 life cycle methods, 53, 419-20, 420t
 location API, 358-61, 359t
 .NET CF, 25
 optional packages, 20-21, 22t
 packaging process, 425-26
 platforms, 18-19
 profiles, 23t-24t
 runtimes, 55, 59
 Services Specification, 287
 Web Services Optional Package, 333-44
 Web Services Specification, 334
 Wireless Toolkit, 173, 174f, 175f
 XML API, 334, 335, 337t
 XML support, 287-88
J2SE (Java 2 Standard Edition)
 API compatibility, 128-29
 architecture of, 18-19
 Generic Connection Framework, 175
 and OSGi, 59
 XML support, 287-88
JAAS (Java Authentication and Authorization Service), 379
Jabber clients, 183-84
Jabber Instant Messaging systems, 164, 179-84
Jabber libraries, 182-83
Jabber protocol, 181-82
Jabber XMPP, 181-82
JabberListener interface, 182
JabberStream class, 182
JAD files, 143, 426
JAIN SIMPLE Instant Messaging, 184
JAIN SIMPLE Presence, 184
JAR files, 36, 55, 65, 143, 383, 426
Java 2 Enterprise Edition. *See* J2EE
Java 2 Micro Edition. *See* J2ME
Java 2 Standard Edition. *See* J2SE
Java Applet, 77
Java Application Descriptor files. *See* JAD files
Java Archive files. *See* JAR files
Java Authentication and Authorization Service. *See* JAAS
Java Blueprints, 82
Java Community Process (JCP), 17-18, 165
Java Cryptography Architecture (JCA), 369
Java Cryptography Extension (JCE), 379
Java Development Environment (JDE), 130, 158-60
Java Messaging Service (JMS), 188
Java Native Interface (JNI), 20
Java Secure Socket Extension (JSSE), 379
Java SMS (jSMS), 175-76
Java Virtual Machine (JVM), 52
JavaBean, 337, 343
JavaPhone API, 155, 156t
JavaTutorial sample, 226f
JAX-RPC API, 336-42
JAXP, 335
JAXP SAX API, 335-36

JBoss, 177
 JCA (Java Cryptography Architecture), 369
 JCE (Java Cryptography Extension), 379
 JCP (Java Community Process), 17-18, 165
 JDBC API, 216-18, 273
 JDBC database, 257
 JDBC extension, 228
 JDBC interfaces, 218t
 JDBC Optional Package, 24t, 216, 223-24
 JDE (Java Development Environment), 130, 158-60
 JDK JCE, 373, 374t
 Jeode PersonalJava VM, 230
 Jini service, 77
 JMS (Java Messaging Service), 188
 JmsLc (JMS lightweight client), 197
 jNeo. *See* NTRU jNeo
 JNI (Java Native Interface), 20
 JP Mobile, 12
 jSMS (Java SMS), 175-76
 JSSE (Java Secure Socket Extension), 379
 JVM (Java Virtual Machine), 52
 JXME network, 184, 288
 JXTA network, 184

K

kCommand toolkit, 140
 kDOM, 329
 kDOM button, 294
 kDOM parsing model, 297-99
 key serialization, 383-84
 keyed hashing, 376-78
 killer mobile applications, 9-11, 13, 164
 kilobyte virtual machines, 20
 Knudsen, Jonathan, 301
 kSOAP
 advanced features, 323-25
 API classes, 323, 324t
 custom mapping, 319-23
 default mapping, 317-19, 317t
 explanation of, 317-23
 Google search, 326-27
 Google spell check, 311-16, 318, 325-26, 329-30
 introducing, 309-16
 library, 48, 342-43

 making RPC calls, 311
 message transport, 313-16
 platforms, 309
 stub generators, 316
 v1.2, 309, 313
 v2.0, 309, 313, 328-32
 KVMJab library, 182, 183
 KVMs (kilobyte virtual machines), 20
 KvmSerializable object, 317, 329
 kXML library, 342-43
 kXML parser, 294
 kXML pull parser, 328

L

LBS (Location-Based Services), 345-62
 legacy applications, 274-78, 278f, 279f
 legacy recorder, 277-79, 280f
 LG Electronics, 16, 334
 Liberty Alliance Project, 367
 life cycle methods, 53, 419-20, 420t
 Linux, 12, 70, 199
 List object, 152
 local data, 420
 LocalModel class, 93
 Location API, 22t
 location API, 358-61
 Location-Based Services (LBS), 345-62
 location determination techniques, 357-58
 logging services, 74-77
 LogService class, 75
 LogService interface, 74-75
 LogService object, 71
 LogTracker class, 75-77
 LogTracker object, 74-75
 LogTracker.log() method, 75
 LogTracker.open() method, 74
 Long, Ju, 328
 Lotus Domino, 158
 Lotus Notes, 252

M

Mail Anywhere Studio, 158
 Mail4ME, 149-55, 155f
 MailSample, 150, 150f
 MailSample.jad, 150
 MapPoint facade, 132, 134
 MapPoint J2ME clients, 353-57
 MapPoint v3.0, 347-48, 349t

- MapPoint Web Services
 - aggregated APIs, 348-50, 350f
 - authentication process, 350
 - driving directions, 348-61, 349t, 354f
 - remote facade, 348, 350, 350f, 351
 - SOAP APIs, 347-48, 349t, 351
- Marshal interface, 320
- Marshal object, 319, 321, 329
- MarshalBase64 class, 323, 331-32
- MarshalDate class, 320-23
- MarshalHashtable class, 323
- Marshal.register() method, 321
- Marshal.writeInstance() method, 321
- MASPs (Mobile Application Service Providers), 12
- maxIteration property, 114
- meConnector class, 182
- Message constructor, 150
- Message object, 150
- MessageConnection class, 166, 168, 169
- MessageConnection.receive() method, 169
- MessageConstants class, 103
- MessageListener class, 192
- messaging-oriented middleware. *See* MOM
- messaging solutions, 141
- Micro Console, 241, 241f
- Microsoft, 128, 132, 164, 179, 346, 347, 350, 353, 367
- Microsoft Exchange, 158, 252
- Microsoft Outlook, 148
- middleware, 273
- MIDlet Suite, 150-53
- MIDlet suites, 241
- MIDlets, 150-53, 171, 276-77, 419-21, 420t, 421f
- MIDP Console, 241f
- MIDP (Mobile Information Device Profile)
 - and AMS, 52, 55, 171
 - applications, 425-29, 433-34, 434f
 - buffered input, 134-35
 - clients, 46, 83, 84-85
 - components, 21
 - deployment of, 429
 - driving direction clients, 356-57, 357f
 - and HTTPS support, 124-25
 - libraries, 129
 - limitations of, 157
 - optional packages, 22t
 - specifications, 334
 - success of, 155
- MIDP PUSH, 171
- Miller, Charles, 136
- Miller, Jeremie, 180
- MIME (Multipurpose Internet Messaging Extensions), 151, 154
- MISPs (Mobile Internet Service Providers), 12
- Mitsubishi, 16
- MMS (Multimedia Messaging Services), 165, 177, 178, 309
- Mobile 3D Graphics API, 22t
- Mobile Application Service Providers (MASPs), 12
- mobile commerce
 - business models, 11-12
 - and freedom economy, 4, 13
 - killer mobile applications, 9-11, 13
 - providers, 11-12
 - revenue models, 11-12
 - technology adoptions, 7-9
 - value propositions, 4-7
- mobile databases, 213-81. *See also* databases
- mobile design patterns, 81-110
- mobile devices
 - with email capabilities, 148
 - manufacturers, 11
 - updating software, 138
 - vendors, 16
- mobile gateways. *See* gateways
- Mobile Information Device Profile. *See* MIDP
- Mobile Internet Service Providers (MISPs), 12
- Mobile Media API, 22t, 23
- mobile messaging applications, 145-211
- mobile MOM solutions, 188, 189, 195, 199
- mobile portals, 134, 135f
- mobile security, 353-418
 - advanced mobile security, 365
 - APIs, 378-79
 - content-based security, 366-67
 - cryptography, 368-79
 - device security, 368, 378
 - distributed access control, 367-68
 - for enterprise, 365-80
 - lightweight API, 371-72

- lightweight cryptography toolkits, 368-70
 - security solutions, 368-79
 - solution providers, 373-75
 - Mobile Server, 262-63
 - mobile software platform providers, 12
 - mobile technology adoption, 7-9, 9f
 - mobile transaction, 366-67, 366f
 - mobile Web Services, 48
 - MobiLink, 260-62
 - Model-View-Controller pattern. *See* MVC pattern
 - ModelFacade class, 93-94
 - MOM (messaging-oriented middleware), 188, 189, 195, 199
 - Moore's law, 4
 - Motorola, 11, 16, 32, 33, 55, 165, 334, 378
 - MPCClient object, 351-53
 - MQeAdapter class, 204, 204t
 - MQeCommunicationsAdapter methods, 209
 - MQeDiskFieldsAdapter class, 204
 - MQeQueueManager object, 201, 202
 - MQeTcpipHistoryAdapter methods, 209
 - MS SQL Server database, 253
 - msgNumbers vector, 152-53
 - MSN, 164
 - Mueller, Thomas, 224
 - Multimedia Messaging Services. *See* MMS
 - multiple cookies, 117
 - multiple devices, 137
 - Multipurpose Internet Messaging Extensions (MIME), 151, 154
 - multitiered application models, 54
 - MVC pattern, 88-92, 132, 141
 - MVCComponent class, 38-42
- N**
- NAME=VALUE text, 117
 - NEC, 16
 - .NET Compact Framework, 25
 - .Net Passport, 367
 - New Economy, 4
 - next() method, 290
 - NexTel, 6, 16, 32, 33
 - nextScreen variable, 45
 - nextToken() method, 290
 - Nokia, 11, 16, 165, 334
 - Nokia Mobile Server Services SDK, 177-78
 - noLogService() method, 75
 - notifyIncomingMessage() method, 171
 - NTRU jNeo, 375-76, 390-92, 406-9, 415-17
 - NTRUEncrypt, 375-76, 383, 406
 - NTRUSign, 375-76, 410f, 415-17
 - NTT, 334
 - Numeric Computer Systems, 6
- O**
- Oak, 16
 - object-oriented language, 17
 - one-way hashes, 121
 - open() method, 71, 74
 - Open Services Gateway initiative. *See* OSGi
 - Open Source, 140, 148, 180, 182, 224, 371
 - openDataInputStream() method, 113
 - openInputStream() method, 113
 - Oracle, 55, 132, 357-58
 - Oracle database, 253, 254
 - Oracle GIS server, 347
 - Oracle J2ME SDK, 241-43, 246, 273-74
 - Oracle9i Application Server, 262
 - Oracle9i Lite, 228-29, 241
 - Oracle9i Mobile Server, 262-63
 - OSGi Alliance, 55
 - OSGi bundles, 55-59, 64f, 138
 - OSGi containers, 55-59, 77-78, 78f
 - OSGi runtime requirements, 59
 - OSGi Service Platform Release 2, 55, 56, 57t, 60
 - OSGi Service Platform Release 3, 55, 56, 58t, 60
 - OSGi services, 52, 56-59, 57t-58t, 78
 - Otis, 7
 - OurPrice tag, 295
 - OUT parameters, 221, 222
 - Over-the-Air server, 429
 - Over-the-Air support, 138
- P**
- P2P mobile messaging, 163-86
 - Palm OS, 12, 158, 199, 228
 - Palm PDAs, 155
 - Palm.net, 12

- password-based encryption schemes. *See*
PBES
- password string, 119, 121
- Patterns of Enterprise Application Architecture*, 88
- PBES (password-based encryption schemes),
375, 392-98, 393f
- PBP (Personal Basis Profile), 23t, 52
- PDA devices
and converged messaging, 179
cost of, 20
and HTML, 133
recharging, 137
runtimes, 60, 431-36
and SMS messages, 164
and synchronization, 252
updating software, 138
and WMQe, 199
- PDA Optional Package (OP), 22t, 155-57
- PDA Profile, 155
- PeekAndPick application, 301, 301f
- peer-to-peer (P2P) mobile messaging, 163-86
- Perl, 353
- Personal Basis Profile. *See* PBP
- Personal Profile. *See* PP
- PersonalJava, 128, 155, 157, 199, 216, 224, 225f, 230
- PersonalJava platform, 55, 59, 70
- Phaos Micro Foundation (MF), 373-75, 388-90, 396-98, 404-6, 413-15
- Phaos Technology, 373
- Phase Forward, 5
- PIM applications, 155-56, 156t, 158, 252
- PIM suites, 148
- Pizza servlet, 73-74
- PizzaBundle class, 72-73
- PKI (Public Key Infrastructure), 375-77, 411
- platform architecture, 18-19, 19f
- PIM suites, 156-58
- PocketPC devices, 60, 61f, 70, 158, 199, 252
- PointBase, 129, 229, 230
- PointBase Contact Manager, 230-36, 230f, 263-68, 264f
- PointBase Embedded, 253
- PointBase Lite API, 240-41
- PointBase Micro Edition, 240-41, 241f, 246, 253-56
- PointBase UniSync, 253-57, 254f, 263
- POP3 (Post Office Protocol), 148-49, 149f, 152-53
- portability, 18-19
- portal servers, 158-60
- Post Office Protocol. *See* POP3
- PP (Personal Profile), 23t, 128-29, 134, 434-35, 435f
- PreparedStatement class, 241
- PreparedStatement interface, 220-21
- prevScreen screen, 44
- private key encryption, 398, 402
- processHeaders() method, 122
- PROCESSING_INSTRUCTION event, 290
- ProductName tag, 295
- Progress Gauge object, 44
- ProgressObserverUI class, 106-9
- PropertyInfo object, 318, 329
- ProSyst, 55
- ProtocolStackRegistry class, 195
- provisioning server, 138, 139f
- Psion, 16
- public key encryption, 375-77, 398-409, 399f
- Public Key Infrastructure. *See* PKI
- public safety, 7
- purchaseTickets() method, 93
- PUSH-based protocols, 136, 171
- ## Q
- QNX, 199
- Qualcomm, 11, 25
- query() method, 114
- Queue Producer, 195
- queues, 193, 200-208
- ## R
- RAD (Rapid Application Development), 276
- RC2, 384
- RC4, 384
- readUTF() method, 46
- Really Simple Syndication (RSS), 300-303
- receive() method, 170
- Record Management System. *See* RMS
record data

Recorder module, 278, 280f
 recorder tool, 274-76
 Reference Implementation. *See* RI
 remote data, 420
 remote facade patterns, 93-98, 134
 remote procedure calls. *See* RPC requests
 RemoteModel interface, 93, 101, 130
 RemoteModelProxy class, 93, 95, 98, 100-101, 103
 RemoteModelRequestHandler class, 98-102
 removedService() method, 72
 RequestHandler interface, 98
 ReqWireless Suite, 157-58
 Research in Motion. *See* RIM
 reserveSeats() method, 93
 ResetServer class, 263
 RESTful Web Service, 292
 ResultSet class, 241, 273
 ResultSet object, 219
 RI (Reference Implementation), 167, 172
 rich browsers, 77
 rich UI clients, 77
 Rijndael, 384
 RIM (Research in Motion), 16, 158, 334
 RMI Optional Package, 24t
 RMS record data, 46, 131
 RMSCacheHandler class, 100-102
 RPC requests, 93, 95, 103-6, 188, 311, 342
 RSA algorithm, 411-12
 RSA key pair, 399-409
 RSAEncrypt() method, 382
 RSAPrivKey class, 402
 RSAPubKey class, 402
 RSS (Really Simple Syndication), 300-303
 runTask() method, 45
 runWithProgress() method, 106, 107

S
 Samsung, 16, 55
 SAP, 6
 Savi Technology, 6
 SAX (Simple API for XML), 288-89, 289f, 335-36
 screen scraping, 274-76, 275f
 SDKs (Software Developer Kits), 12, 132
 SearchResults class, 326

SearchResults object, 326
 seConnector class, 182
 Secure Socket Layer. *See* SSL
 security, 7, 16-17, 85, 137, 205, 365-80, 382-418. *See also* cryptography; encryption
 Security and Trust API, 22t
 SecurityInfo object, 124
 Select command, 219
 Select query, 219
 SELECT statement, 261
 self-contained applications, 53
 SendDemo, 150
 Serializable object, 307
 serverside messaging API, 178-79
 service device, 78
 Service Provider Interfaces. *See* SPIs
 Services Management Framework. *See* SMF
 ServiceTracker object, 65, 66, 71-72, 74
 ServiceTrackerCustomizer object, 66
 Session Initiation Protocol. *See* SIP
 session tracking, 117-19
 SessionConnector class, 117
 set-cookie header, 117
 setBytes() method, 241
 setJabberListener() method, 182
 Sharp, 16, 334
 Short Message Service. *See* SMS
 Siemens, 16, 165, 334
 Simple API for XML. *See* SAX
 Simple Mail Transport Protocol. *See* SMTP
 Simple Object Access Protocol. *See* SOAP
 Simple Object Database Access. *See* SODA
 Simplewire Java SMS SDK, 176
 Simplicity Enterprise IDE, 277-79, 278f, 279f
 Simplicity Enterprise Mobile Server, 132
 Simplicity for Mobile Devices IDE, 276
 Simplicity IDE, 140, 276-79, 277f
 Simplicity Legacy Rejuvenation, 277-79, 279f
 Simplicity RAD, 140
 single architecture solution, 18-19
 single sign-on service, 34-36, 367-68, 368f, 369f
 SIP API, 22t, 136, 184-85
 SIP IM applications, 184
 SIP Lite API, 184

- SIP (Session Initiation Protocol), 22t, 136
- SipClientConnection object, 184
- SipConnectionNotifier object, 184
- SipDialog object, 185
- SipServerConnection object, 184
- smart applications, 252
- Smart Chain, 6
- smart clients
 - authentication process, 34-35
 - benefits of, 32-33
 - and iFeedBack, 31-49
 - implementation walk-through, 38-48
 - managed smart clients, 51-79
 - container-managed applications, 52-55
 - echo service example, 59-69
 - and HTTP front end, 70-77, 71f
 - IBM Services Management Framework, 59-61
 - mobile gateways, 77-79, 78f
 - OSGi bundles, 55-59
 - OSGi containers, 55-56
 - pizza order example, 70-77, 71f
 - migration path, 133
 - threading model, 42-46
 - UI call model, 38-42
- smart phone devices, 16, 20, 83, 155, 252
- Smart Ticket Wireless Blueprint, 81-110
 - in action, 84-88
 - architecture of, 88-92, 92f
 - binary RPC over HTTP, 103-6
 - chained data handlers, 98-102
 - controller classes, 89
 - and cookies, 117-18
 - environment variables, 83
 - features
 - account credentials, 84
 - movie ratings, 85-86, 87f
 - theater schedules, 87-88, 87f
 - ticket purchases, 85, 86f
 - user preferences, 84-85, 84f
 - user profile, 84-85
 - getting started, 82-83
 - history of, 82
 - model classes, 89, 93-94
 - MVC pattern, 88-92
 - patterns, 88-98
 - remote facade patterns, 93-98
 - requirements for, 82-83
 - techniques, 98-109
 - threading model, 106-9, 106f
 - view classes, 89
- SmartPhrases, 328, 328f
- SmartTicketBD class, 95-97
- SmartTicketFacadeBean session, 95, 97-98
- SmartTicketServlet, 95-96, 106
- SMF (Services Management Framework), 52, 55, 59-61, 61f, 62t, 70-77, 71f, 199
- SMS (Short Message Service), 136, 164-79
- SMSC (SMS Center), 175, 176
- SMTP (Simple Mail Transport Protocol), 148-49, 149f, 151
- SmtplibClient object, 151
- SOAP RPC, 292
- SOAP Web Services, 305-32
 - advantages of, 306-7
 - architecture of, 292, 308-9, 310f
 - documents, 323-25
 - explanation of, 306-9
 - headers, 329
 - HelloWorld example, 307-8
 - kSOAP, 309-32
 - mappings, 317-24, 337-38, 339t
 - parsing, 310-11, 318, 319f
 - servers, 329
 - stock trade example, 321-23, 325
 - support for, 306-7
 - templates, 323-25
 - writing messages, 320
 - XML API, 48
- SoapEnvelope class, 329
- SoapEnvelope object, 329
- SoapEnvelope.getBody() method, 318
- SoapEnvelope.getResult() method, 318
- SoapFault class, 313
- SoapObject object, 311, 317-18, 320-21, 323
- SoapObject template, 323-25
- SoapObject.getName () method, 318
- SoapObject.getNamespace() method, 318
- SoapPrimitive object, 317-18, 320-23
- SoapPrimitive.getName() method, 317
- SoapPrimitive.getNamespace() method, 317
- SoapPrimitive.toString() method, 317
- SoapSerializationEnvelope object, 329

- SODA (Simple Object Database Access), 241-43, 242t, 246
- Software Developer Kits (SDKs), 12, 132
- software updates, 138, 139f
- Softwired, 195
- Sony, 16, 334
- SPIs (Service Provider Interfaces), 341, 342t, 343
- SprintPCS, 12, 16, 165
- SQL Anywhere Studio, 226
- SQL databases, 132, 224, 273-74
- sql/init.sql script, 36
- SQL scripts, 244, 261
- SQL string, 219, 220
- SSL (Secure Socket Layer), 123, 366, 376
- standalone GUI, 77
- start() method, 71
- START_TAG event, 290
- state-change APIs, 69
- Statement class, 273
- Statement interface, 220-21
- Statement object, 219-20
- Statement.update() method, 219
- static sharing, 56
- Stop button, 106
- stop() method, 66, 72
- stored procedures, 221-22
- Streampath, 183
- String format, 150, 241
- String object, 131, 307
- StringBuffer object, 131
- Stub class, 339-41
- stub generators, 316, 339-41
- SubmitAllTask class, 46-48
- Sun Microsystems, 32, 33, 48, 55, 82, 109, 117, 167, 334, 367
- SunONE Studio, 316
- Suto, Larry, 378
- Sutton, Al, 182, 183
- Swing API, 24
- switch statements, 89
- Sybase, 158, 224, 225, 253, 260-61
- Symbian OS, 12, 129, 158, 199, 334
- symmetric encryption, 384-92, 384f
- synchronization engines, 132, 138, 213-81
- synchronization solutions, 249-69, 251f
- SynchronizationAgent class, 93
- synchronizeMovieRatings() method, 93
- SynchronizingBean session, 95
- synchronous messaging, 167, 169
- SyncML, 252, 257
- ## T
- TableMetaData class, 244
- tamper detection, 376-78
- TCP/IP sockets, 46, 148, 150, 172
- Telcordia, 55
- TEXT event, 290
- Theater class, 104-5
- thin clients, 32, 133
- threading model, 42-46, 42f, 43f, 106-9, 106f
- threads usage, 142f
- TicketingBean session, 95
- TimeStamp string, 220
- TipicME, 183
- TLS (Transaction Layer Security), 123, 366
- ToDo class, 156
- ToDoList class, 156
- Tomcat, 177, 351
- tracking services, 65-66, 71-72, 74
- tracking sessions, 117
- Transaction Layer Security. *See* TLS
- Transaction module, 278
- ## U
- UDDI registry, 309
- UI lock-up, 142
- UI model, 38-42, 39f, 420
- UI Optional Package, 24, 24t
- UI (user interface), 38-42, 77, 88. *See also* GUI
- UIController class, 89, 106-7
- UltraLite database, 226-27, 260, 262
- UltraLite Deployment Option, 225, 226f
- Unicode Transformation Formats (UTF), 103-4
- UniSync. *See* PointBase UniSync
- University Wireless Developer Contest, 32, 33
- UNIX, 199
- Update command, 219
- update() method, 219
- Update Token object, 40-42
- updateProgress() method, 106-7
- UpdateTokenTask class, 45
- UPnP service, 77

UPS, 6
URL cbs, 167
URL sms, 166-67
UsedPrice tag, 295
user credentials, 119, 121
user interface. *See* UI
user preferences, 84-85, 84f
user profile, 84-85
username string, 119, 121
UTF (Unicode Transformation Formats)
 strings, 103-4

V

value propositions, 4-7
Virtual Private Network (VPN), 367
voice callers, 133
Voice over IP (VoIP), 184
voice portal server, 133
VoiceXML, 133

W

W3C, 148
WAP (Wireless Application Protocol), 32,
 124, 124f, 257
WAP/WML, 25, 32
WAP/XML, 25
WCE (WebSphere Custom Environment
 JVM), 60
Web Service Developer Pack, 48
Web Services API, 22t
Web Services Definition Language. *See*
 WSDL
Web Services Optional Package, 333-44
 APIs, 335-38, 337t, 338, 340t
 client stubs, 341
 features, 336-38, 343
 gateway, 341, 343f
 JAX-RPC API, 336-41, 338f
 JAXP SAX API, 335-36
 and kSOAP, 342-43
 and kXML, 342-43
 Service Provider Interfaces, 341, 342t
 stub generators, 339-41
 user scenario, 339-41
 XML API, 334, 335, 337t
 XML support, 342-43
Web Services Tool Kit for Mobile De-
 vices, 334

WebSphere Custom Environment JVM,
 60
WebSphere Micro Environment JVM. *See*
 WME
WebSphere MQe
 administration queue, 204-8
 communications adapters, 209-10, 209t,
 210t
 HelloWorld example, 200-204
 mobile MOM solution, 199
 mobile portals, 134
 queue manager, 200-204
 storage adapters, 204, 205t
WebSphere Studio Device Developer IDE,
 60, 129, 316, 334, 431-33, 432f
WiFi network, 10, 77, 124, 164
Windows, 199
Windows CE, 12
Wintel PCs, 16
Wireless Application Protocol. *See* WAP
wireless carriers, 309
Wireless Messaging API. *See* WMA
wireless networks, 357-58
Wireless Toolkit. *See* WTK
WMA (Wireless Messaging API), 164-79
 in action, 167, 168f
 advantages of, 164
 architecture of, 22t, 173
 connections, 166-67
 interfaces, 165t, 166f
 PUSH-based protocols, 136
 Reference Implementation, 167, 172
 runtime properties, 172-73
 URLs, 166-67
 Wireless Toolkit, 173, 174f, 175f
WMAsync, 167
WMA Tester, 165, 167, 168f, 172-73
WME (WebSphere Micro Environment
 JVM), 59, 60, 228
WMQe. *See* WebSphere MQe
“Write Once Run Anywhere” (WORA),
 18
writeUTF() method, 46
WSDL (Web Services Definition Language),
 48, 309
WTK (Wireless Toolkit), 173, 174f

X

XcelleNet, 6

Xerox, 7

XML

Amazon Web Services, 291-94, 295f

DOM parsing process, 291, 291f

explanation of, 286-87

kDOM parsing model, 297-99

and mobile web services, 283-362

parsing models, 288-91, 289f, 290f,
291f, 294-97

RESTful Web Service, 292

RSS parsing model, 300-303, 301f

sample document, 286

SAX parsing process, 288-89, 289f

for small devices, 285-304

streams, 181

structures, 292

support for, 287

XML API, 334, 335, 337t

XML Exporter module, 278, 279

XML User Interface. *See* XUL

XML Web Services, 112, 306

XML Web Services interface, 35

XmlPull API, 289-90, 290f, 294-97, 331

XmlPullParser interface, 290

XmlPullParserFactory class, 290

XmlReader class, 292

XUL (XML User Interface), 77, 133

Y

Yahoo!, 164, 179

Yuan, Michael, 328