Procedure/ Function/Trigger	Schema	Package	Listing	Special	Execute By	Called By
p_get_class_conns	appsec	appsec_only_pkg	12-68			setDecryptConns, copyPreviousConns, getCryptConns
p_set_class_conns	appsec	appsec_only_pkg	12-68			setDecryptConns, copyPreviousConns
f_get_crypt_conns	appsec	appsec_only_pkg	ref 10-46	Java Stored Proc		p_get_app_conns
p_check_role_access	appsec		10-6	CURRENT_USER	PUBLIC	getAAConnRole
p_appver_logon	appsec		10-14	CURRENT_USER	PUBLIC	t_screen_appver_access
t_screen_appver_access	appsec		10-13	Trigger		LOGON appver
p_create_template_class	appsec	appsec_admin_pkg	12-45		appver_admin, appver_conns	OJSAdmin
f_set_decrypt_conns	appsec	appsec_admin_pkg	10-25	Java Stored Proc	appver_admin, appver_conns	putAppConnections
p_copy_app_conns	appsec	appsec_admin_pkg	11-27		appver_admin, appver_conns	copyAppConnections
p_get_app_conns	appsec	appsec_public_pkg	10-46		PUBLIC	getAppConnections
f_mask	appsec		11-18ff	Wrapped		setDecryptConns, copyPreviousConns

Procedure/ Function/Trigger	Schema	Package	Listing	Special	Execute By	Called By
f_unmask	appsec		ref 11-18	Wrapped		getCryptConns, copyPreviousConns
p_create_user_once	sys	usr_role_adm_pkg	12-28	Autonomous	ojsaadmor ojs_adm_admin	OJSAdmin app, p_create_apver_user
p_create_user_many	sys	usr_role_adm_pkg	12-28	Autonomous	ojsaadmor ojs_adm_admin	OJSAdmin app, p_create_apver_user
p_drop_user	sys	usr_role_adm_pkg	12-28	Autonomous	ojsaadmor ojs_adm_admin	[OJSAdmin app], p_drop_apver_user
p_set_proxy_through	sys	usr_role_adm_pkg	12-28	Autonomous	ojsaadmor ojs_adm_admin	OJSAdmin app, p_set_apver_proxy_through
p_drop_proxy_through	sys	usr_role_adm_pkg	12-28	Autonomous	ojsaadmor ojs_adm_admin	OJSAdmin app, p_drop_apver_proxy_through
p_grant_appver_conns_role	sys	appver_conns_role_pkg	12-50	Autonomous	ojsaadm, appver_admin	OJSAdmin app, p_grant_apver_appver_conns
p_revoke_appver_conns_role	sys	appver_conns_role_pkg	12-50	Autonomous	ojsaadm, appver_admin	[OJSAdmin app], p_revoke_apver_appver_conns
t_application_registry_iar	appsec		12-56	Trigger		INSERT t_application_registry
apps_for_admin	appsec		12-64	For VPD		apps_for_admin_policy
apps_for_user	appsec		12-66	For VPD		apps_for_user_policy

Procedure/ Function/Trigger	Schema	Package	Listing	Special	Execute By	Called By
p_create_apver_user	ojsaadm	apver_usr_adm_pkg	12-30		ojs_adm_admin	OJSAdmin app
p_drop_apver_user	ojsaadm	apver_usr_adm_pkg	12-30		ojs_adm_admin	[OJSAdmin app]
p_set_apver_proxy_through	ojsaadm	apver_usr_adm_pkg	12-30		ojs_adm_admin	OJSAdmin app
p_drop_apver_proxy_through	ojsaadm	apver_usr_adm_pkg	12-30		ojs_adm_admin	OJSAdmin app
p_grant_apver_appver_conns	ojsaadm	apver_usr_adm_pkg	12-51		ojs_adm_admin	OJSAdmin app
p_revoke_apver_appver_conns	ojsaadm	apver_usr_adm_pkg	12-51		ojs_adm_admin	OJSAdmin app
p_check_secadm_access	sys		2-1	CURRENT_USER		Logon secadm

Index

A

access control list (ACL), 179
access user, to test application, 23
ACL (access control list), 179
acquisition, of security administrator role, 13–14
Add/Modify User functional screen, 341–354
    employees
        creating new, 352
        saving data for, 352–354
        selecting existing, 347–351
    initializing data selection components, 345–347
    instantiating AddUser screen, 342–345
addButton_actionPerformed( ) method, 369
addItem( ) method, 366
add_job_history, 128
addresses
    employee, getting for delivery of two-factor code, 216–217
    setting company-specific, 194–195
AddUser class, 341, 344
AddUser screen, instantiating, 342–345
addUserButton_actionPerformed( ) method, 341
Administration interface, 329
Administration menu, security, 339–341
administrative privileges, updates to, 362–364
administrative roles, for application verification, 292–293
administrative users
    compiling TestOracleJavaSecure application and running as, 326–327
    managing connection strings for applications, 291–292
administrator role, application verification, 374
administrators
    application, tables of, 377–379
    limiting to certain applications, 391–398
        adding dynamic where clause, 396–398
        VPD, 392–395
AdminUsers class, 354
AFTER INSERT, 116
AlgorithmParameterSpec, 88
algorithms
    and code, obfuscating, 272–275
    method used to show, 100
    negotiated
        for coding automatic upgrades, 88
        for password-based encryption, 104–105
ALL_PROCEDURES PUBLIC, 303
ALL_REGISTRY_BANNERS, 312
ALL_SOURCE PUBLIC, 303
ALL_USERS, 15, 59, 225, 303, 310, 316
ALTER command, 14, 17, 20–21, 77, 112, 116, 132, 156, 158
ALTER USER, 225–226, 291, 308, 314–315, 355–356, 358, 375
alter_command, 132
ancillary methods, for array conversion, 99–100
App class, 238
App Class button, 375–377
appaccess, 8
AppAccessSecure, 145
appAuthCipherDES, 234, 245–246, 248, 301
appAuthSessionSecretDESKey, 234, 245, 248–249, 301
appClass, 231, 235, 253, 261, 325, 336, 338, 383, 385, 387
AppException, 43
application administrators, tables of, 377–379
Application Assignment screen, 365–371
    initializing data selection components, 366–368
    proxies
        adding to user's list, 369–370
        removing from user's list, 370
        saving updates to user's, 370–371
        selecting available in table, 368–369
    selecting user from list, 369
application context, setting, 231–232, 253–254, 261

Application ID class
and HashMap list of connections, 239–240
updating two-factor code functions to use, 211
application Oracle connection, 172
Application Registration screen, 372–385
App Class button, 375–377
application verification administrator role, 374
Register Application button, 382–385
security table access analysis, 379–382
tables of application administrators and application to class registry, 377–379
application registry
finding out if entry exists for server application, 243–244
getting entries from, 243
getting list of connection strings for registered application, 244
setting values in, 242–243
application security
package body, 95–97
package specification, 93–95
packaging template to implement data encryption, 144–145
procedures, accessing HR tables from, 190
updating packages, 117–119
Application Security users, 17–18, 112–117
error log management procedure, 115–116
granting more system privileges to, 112
permitting to read web pages, 182
role, 17–18
tables for error logging, 113–117
Application Selection screen, 385–388
initializing list of applications, 386
selecting Manage Selected Application button, 386–388
application users, 18–20
application verification, 255
administrator role, 374
establishing connection for, 249
application_registry, 213, 380, 384, 400
applications, 209–265
authentication
connection and role, 259–260
testing, 251–256, 260–262
authorization
for authenticating on separate instance of Oracle Database, 317–318
overview, 220–221
structures for, 226–227
user for, 221–226
to class registry, and tables of application administrators, 377–379
client call, for copying connection strings from previous version, 294
connection strings
Java client side, 244–249
list of for server side, 250–251
saving, 233–244
set of for applications, 227–233
initializing list of, 386
limiting administrators to certain, 391–398
adding dynamic where clause, 396–398
VPD, 392–395
Login screen, 330–339
application inner class, 331–332
background processing thread, 336–337
center method, 333
closing, 338–339
constructors, 334–335
Continue button, 337–338
wait while processing modal dialog, 335–336
managing connection strings for, 291–299
administrative role for application verification, 292–293
administrative user, 291–292
copying from previous version, 293
deleting, 293
passwords, editing, 321–322
revoking user access to run, 364
rewrite and refactor method to distribute two-factor code, 216–219
changes to, 218–219
getting employee addresses for delivery, 216–217
stored procedure to update two-factor code cache, 217–218
secure application role procedure for multiple, 210–215
moving test for SSO to separate function, 212
rebuilding two-factor cache table, 211
updating two-factor code functions to use application ID, 211
test, user access to, 23
updating security structures, 301–302
updating to two-factor distribution formats, 219

use of SSO, 171–176
code template for developers, 175–176
example application, 172–173
updates to OracleJavaSecure class, 173–175
verification
administrative role for, 292–293
removing from ORCL instance, 319
APPSEC user
CREATE_SESSION_ROLE role and, configuring, 312–313
creating personal database links as, 314–315
granting access to reach out of JVM security sandbox, 315
permitting to create database links, 314
appsec_admin_pkg, 293–294, 299, 325, 327, 375, 377, 380–381
appsec.appsec_admin_pkg, 293, 374–375, 381
appsec.appsec_only_pkg, 212, 239–240, 242, 296–297, 299, 396
appsec.app_sec_pkg, 93, 112, 122–124, 126–127, 144
appsec.app_sec_pkg.f_get_crypt_data, 123–124
appsec.apps_for_admin, 393
appsecConnString, 78, 80, 102–103
appsec_only_pkg, 212–214, 216–217, 223–224, 250, 294
appsec.p_check_hrview_access, 18–21, 134, 141, 156, 161, 214–215
appsec.p_check_role_access, 214–215, 255, 259, 355, 374
app_sec_pkg, 93, 100, 107, 119, 125, 217, 294, 350, 377, 381
APPSEC_PUBLIC_PKG, 318, 379
appsec_role, 9, 17, 53, 58, 93, 112–113, 121, 179, 204
AppSec.sql file, 211
appsec.t_app_class_id, 374, 378, 394
appsec.t_app_conn_registry, 227, 377–378, 392, 394, 397–398
appsec.t_application_admins, 374, 377–378, 391–394
appsec.v_app_class_id, 378–380, 383–384, 386, 396
appsec.v_app_conn_registry, 323, 373–374, 380, 393, 396–398, 400
appsec.v_app_conn_registry_filtered, 397–398
appsec.v_application_admins, 378–380, 383, 392–393, 396–398
appsec.v_application_registry, 210, 325, 344, 366–368, 374–375, 384, 396
appsec.v_two_fact_cd_cache, 191–192, 203, 205, 322, 327
appsHashtable, 367–369
APPUSR, 59, 156, 160–162, 164, 167
appusr user, 172
appusrConnOCIURL, 163
appusrConnPassword, 163, 166, 171
appusrConnURL, 163, 166, 171
appusrConnUser, 163, 166, 171
APPVER connection string
hiding, 267–277
generating encoded APPVER connection string, 276–277
getting from encoded string, 270
getting from encrypted Java class, 269–270
getting from encrypted string, 270
getting from JNI, 269
getting from secondary source, 268
obfuscating algorithm/code, 272–275
using encode method, 270–272
and keys, 248–249
user, password, 320–321
appver_role, 9
APVER instance, encoding APPVER user password for, 320–321
ApVerDBCreate.sql, 305–306, 308
apver_link, 359–361, 382
archives, Java language, 145
args, 33–34, 36, 38, 40, 42, 73, 133
ARRAY class, 28–31, 33–36, 39
array conversion, ancillary methods for, 99–100
arraycopy( ) method, 274
ARRAY.java, 28
arrays, building class from, 237–238
artifacts
building public keys from, 66–67
of password keys, sharing, 86–87
password, package to get, 93
and passwords, generating, 86–87
returning to client, 90–92
RSA public key, 70–71
AS LANGUAGE JAVA, 193, 195
AS SYSDBA privilege, connecting from remote, 312
asktom.oracle.com, 62
ASSIGN_ACL, 179–180
AssignApp class, 366
at rest data, encrypting and decrypting, 287–289
at rest key store, 286–287
AUDIT ANY, 14

audit logs, for sensitive view, 24
audit trail, SYS user, 15
auditing
  activity with client identifier set, 157
  application verification, 226
  changes to procedures, 20
  failed attempts to access HR data, 20
  proxy sessions, 161–162
authenticating
  on separate instance of Oracle Database, 302–319
    changing passwords for SYS and SYSTEM users, 308
    configuring CREATE_SESSION_ROLE role and APPSEC user, 312–313
    connecting from remote AS SYSDBA privilege, 312
    creating new Oracle service, 304–305
    creating structures for, 317–319
    database, 307–311, 313–315
    granting access for APPSEC user to reach out of JVM security sandbox, 315
    increasing quantity of processes, 309–310
    installing DBMS_JAVA package, 311
    removing from ORCL instance, 319
    revoking PUBLIC view grant on sensitive data dictionary views, 316
    viewing installed services, 311–312
    writing create database command, 305–306
authentication
  adding credentials for, 301
  connection and role, 259–260
  testing, 251–256, 260–262
    adding application connection strings, 256
    connections, 254
    getting application connection with role, 261
    Oracle Database, 252–254, 261–262
    for second application, 257–259
    setting application context, 253–254, 261
    steps of, 252–253
    viewing proxy connection, 262
authentication layers, 149–150
AUTHID CURRENT_USER, 13, 19, 212–215, 223, 379, 392
authorization
  application
    for authenticating on separate instance of Oracle Database, 317–318
    overview, 220–221
    user for, 221–226
  structures for, 226–227
auto-commit, disabled in Oracle JVM, 61–62
automatic upgrades, coding, 88
avadminCheckBox, 362–364
availableProxiesTable_mouseClicked( ) method, 368

B

background processing thread, 336–337
BEFORE UPDATE OR INSERT, 129
BEGIN, 13, 19, 96, 115, 118, 122, 127–128, 132, 157
BERLIN_WALL_FELL Date, 69
BETWEEN, 19–20, 156, 161
BigInteger, 66–67, 70–71
binary math, 195
blankAll( ) method, 346–347, 361, 363–364
BLOB data type, 66, 124, 227
blocks
  catch
    giving feedback from, 48
    one per try block approach, 47
  finally
    clean up in, 42–47
    closing local resources in, 49
boolean, 80, 98, 103, 168
bootstrapping, OJSAdmin interface, 399–402
bottomMenuPanel, 340–341
bugs, Java 1.5 password-based encryption, 88
business hours, normal, 20
ButtonGroup, 343, 345
BY ACCESS, 15, 20–21, 23
BY SESSION, 15
byte arrays, building class from, 237–238
byte code compilation, and JVM, 31–33
  finding referenced code and classes, 32
  running compiled code, 32–33
  using Java compiler, 31
ByteArrayInputStream, 238, 240–241, 249, 290, 295, 297–298
ByteArrayOutputStream, 235, 242
byteArrayToCharArray( ) method, 99
BytesClassLoader class, 257

C

cache aging, testing, 191–192

cache tables
two-factor codes, 190–191
two-factor, rebuilding, 211
cached_two_factor_cd, 192–193
cache_ts, 190–192, 199, 201
caching
pass codes, in Oracle Database, 201
statements, 167
callable statements, reusing, 134–135
calls
client, to store list of connection strings in Oracle Database, 234–236
stacked, 79
catch blocks
giving feedback from, 48
one per try block approach, 47
cell phones, two-factor delivery to, 184–185
center( ) method, 333, 336
characters, random, 87
charArrayToByteArray( ) method, 99
CharArrayWriter, 90–91
checkFormat2Factor( ) method, 232–233
Cipher class, 67–70, 73, 84, 88–89, 92, 99
Cipher.DECRYPT_MODE, 68, 73, 79, 98
Cipher.doFinal( ) method, 84
Cipher.ENCRYPT_MODE, 68, 73, 78–79, 90, 92
Cipher.getInstance( ) method, 67
cipherRSA.doFinal( ) method, 74, 79–80
class equality, testing for, 240
class loaders, URLClassLoader, 268–269
class members, static, 87–88
Class object, 238
class registry, applications to, 377–379
ClassCastException, 246
classes
building from byte arrays, 237–238
finding out whether registered, 239
inner
application, 331–332
implementing in OracleJavaSecure class, 229–230
to represent application, 228
and null, 36–37
referenced, and referenced code, 32
Class.forName( ) method, 55, 152
Class.getMethod( ) method, 152
classInstance, 237–238, 295, 297
ClassLoader, 257, 269–270
classMethod, 152–153, 238, 296
className, 238–240, 242, 290, 295–299
Class.newInstance( ) method, 152
classObject, 238, 240–241, 249, 295–297
CLASSPATH parameter, 33, 245, 257, 280–281, 283–284, 383, 387
classTextField, 375, 383
classVersion, 238–240, 242, 290, 295–299
clean up
in finally block, 42–47
of Java language stored procedures, 60
as part of exception handling, 41–42
clear( ) method, 367–368
clearBytes, 68, 73–74, 89–90
clearConnsHash, 241–242, 251
clear_raw, 287–289
clearText String, 79
client application
getting application list of Java client side connection strings, 246
saving connection strings for, 233–236
client call to store list of in Oracle Database, 234–236
method to put in list for application, 233–234
client code, tracing, 282–283
client identifiers, 155–157
client keys, testing encryption failure with, 140–141
client/server data transmission, 107–110
client/server password encryption, coding to test, 102–107
negotiated algorithm for password-based encryption, 104–105
Oracle Database, 103–107
secure client/server data transmission, 107–110
server and client, 102–103
server portion of main( ) method, 103
client wallets, 277–282
administering security of, 281–282
from Java language, 280–281
from SQL*Plus interface, 280
CLIENT_IDENTIFIER, 223, 262, 383, 385, 393–398
clientPubExponent, 73–74, 81, 101, 104–105
clientPubModulus, 73–74, 81, 101, 104–105
clients
decrypting at, 82
generating keys on, 63–64
returning artifacts to, 90–92
and servers
Java code for both, 72–73
testing PKE on, 80–84

clients (*continued*)
servers and, coding to test password encryption, 102–103
testing PKE on, 73–75
running code, 74–75
writing main( ) method, 73–74
clone( ) method, 69
Cloneable interface, 69
close( ) method, 40, 42, 44–45, 52, 104, 106–107, 136, 138
closeConnection( ) method, 256, 343–344
code
byte, compiling, 31–33
compiled, running, 32–33
distributing
to e-mail, 203
to pager URL, 202–203
to SMS, 201–202
Java language, for password encryption, 86–93
referenced, and classes, 32
and syntax, Java language, 33–39
code compilation, scripts execution and, 398–399
code functions, two-factor, 211
code templates, for developers, 175–176
coding cross-platform-specific, with reflection, 151–152
Collection class, 227
columns
in encrypted string, selecting all, 137–138
sensitive data, in EMPLOYEES table, 122–124
command prompt, development at, 28–33
beginning Java syntax, 30–31
byte code compilation and JVM, 31–33
environment, 28–29
COMMISSION_PCT, 21–23, 123, 127, 137, 143, 146
commissionPctTextField, 349, 353
COMMIT, 61–62, 115, 117–118, 133, 146, 179, 181, 189
company-specific addresses, setting, 194–195
compilation, of code, 398–399
compile/run command, 169–170
compiled code, running, 32–33
compilers, Java language, 31
compiling byte code, and JVM, 31–33
conn.close( ) method, 141, 160
CONNECT command, 8–10, 15, 21, 23, 75, 158, 176, 307
Connection class, 55–56, 155, 173
Connection members, instantiating from static initializer, 71–72
Connection parameter, 173
connection pools, 162–171
proxy connections, from OCI connection pool, 162–166
proxy sessions, from thin client connection pool, 166–169
UCP, 169–171
compile/run command, 169–170
connection, 171
connection pool factory, 170
summary, 171
thin client implementation, 170–171
Connection String Copy screen, 391
Connection String Editor, 388–390
initializing list of connection strings, 389
saving list of connection strings to database, 390
selecting existing connection string, 390
updating connection string in list, 390
connection strings
application, adding, 256
APPVER, hiding, 267–277
copying from previous version
application client call for, 294
Java language, 295–299
overview, 293
deleting, 293
initializing list of, 389
Java client side, 244–249
establishing connection for application verification processes, 249
getting application list of connection strings, 245–249
using static class members to retain APPVER connection and keys, 248–249
list of for server side, 250–251
managing for applications, 299
administrative role for application verification, 292–293
administrative user, 291–292
connection strings, 293
running main to copy to new version, 323–324
saving
for client application, 233–236
list of to database, 390
for server application, 236–244
selecting existing, 390

sending list of to Oracle Database for storage, 254
set of for applications, 227–233
deserialization and serialVersion UID member variable, 230
formatting user-input two-factor code, 232–233
inner class, 228–230
setting application context, 231–232
storing, in Oracle Database, 261
updating in list, 390
connections
application
authentication, 259–260
call to get, 254
getting encrypted data with, 256
getting with role, 261
Oracle, 172
verification, 255
establishing for application verification processes, 249
getting for application and associated secure application roles, 255–256
getting unique for use in application, 254–255
HashMap list of, Application ID class and, 239–240
local, 16
opening and closing, 141
proxy, viewing, 262
UCP, 171
connsHash object, storing for server application, 242
consHash.size( ) method, 338
constraints, integrity, 128–129
constructors, 71, 334–335
contains( ) method, 371
Continue button, 337–338
continueButton_actionPerformed( ) method, 337–338
converting arrays, ancillary methods for, 99–100
copyAppConnections( ) method, 294, 323–325
copyPreviousConns( ) method, 295
count(*) method, 129, 134
cpool class, 163–168, 170–171
CREATE ANY PROCEDURE, 14, 19
CREATE command, 376
create database command, writing, 305–306
CREATE OR REPLACE function, 13, 19, 57, 75, 78, 80, 114, 116, 121, 128
CREATE PROCEDURE, 17, 53, 121–122, 144
CREATE SESSION, 10–11
create-session role, 10
CREATE statement, 53, 131, 178
CREATE TABLE, 17, 113–114, 121
CREATE TRIGGER, 112, 121
CREATE VIEW, 17
CREATE_ACL, 179–180
CREATE_SESSION, 10
create_session_role, 9–10, 16, 18, 222, 291, 355, 357, 375, 379
CREATE_SESSION_ROLE role, and APPSEC user, 312–313
createTemplateButton_actionPerformed( ) method, 375
cross-platform-specific coding, with reflection, 151–152
cryptBytes, 73–74, 99
cryptConnsHash, 241–242, 251
cryptData, 79–80, 82, 97–98, 101, 106–107, 120
CURRENT_TIMESTAMP, 76–77, 83
CURRENT_USER, 13, 19, 160, 164, 213
CURSOR TYPE, 146

D

Data Dictionary, 15
data encryption, 111–147
application security user, 112–117
error log management procedure, 115–116
tables, 113–115
trigger to maintain error log table, 116
demonstrations and tests of, 142–144
demonstrating scenarios, 143–144
observing results, 142–143
querying employees to see updates, 144
encrypted data exchange, 133–142
opening and closing connection, 141
preliminary steps, 133–135
running basic key exchange without data encryption, 142
selecting all columns in encrypted string, 137–138
selecting EMPLOYEES data, 139–140
selecting from EMPLOYEES table, 135–139
sending to Oracle database for Insert/Update methods, 138–139
testing encryption failure, 141
inserting record into EMPLOYEES table, 131–133

data encryption (*continued*)
loading updated OracleJavaSecure class into Oracle Database, 121
methods for using and testing in transit, 119–121
building password key, 119–120
temporary method to reset all keys, 120–121
packaging template to implement, 144–145
Java language archive for use by applications, 145
for Java language calls to application security, 145
for Oracle Database application security structures, 144
security administrator, 111–112
security structures for HR user, 121–131
executing HR package specification and body, 131
privileges that enable HR tasks, 121–122
revising procedure to get shared passphrase, 125
security package, 122
selecting all data as single sensitive string, 124
selecting sensitive data for Employee ID, 125
sensitive data columns in EMPLOYEES table, 122–129
SQL injection, 129–131
updating application security package, 117–119
Data Encryption Standard. *See* DES
data selection components, initializing, 345–347, 366–368
data transmission, secure client/server, 107–110
database administrator (DBA) role, 9
DATABASE command, 306
database links
creating personal as APPSEC user, 314–315
enabling OJSAAdm user across, 359–361
to ORCL instance, 313
permitting appsec user to create, 314
updating TNSNAMES.ora file for, 313
databases
configuring to use UTL_MAIL package, 310–311
creating and configuring, 307–308
finding test, 5
function to find user, 224–225
Java language, method to copy connection strings, 295–299
Oracle, structures of, 3
running Oracle scripts to add basic capabilities, 310
saving list of connection strings to, 390
storing database parameters in, 308–309
VPD, 392–395
working from existing, 6
dataInit( ) method, 343, 345–347, 354, 366–367, 386, 389–390
Date class, 35, 74
Date( ) method, 73, 139
Date object, 198
Date type, 56
DATE type, 60
dates, Oracle Database and Java language, 197–198
DBA (database administrator) role, 9
DBA_USERS, 15
DBMS_CRYPTO package, 285
DBMS_CRYPTO.DECRYPT, 289
DBMS_CRYPTO.ENCRYPT, 288–289
DBMS_JAVA package, 181–182, 311
DBMS_JAVA.GRANT_POLICY_PERMISSION command, 181
DBMS_JOB PL/SQL, 116
DBMS_NETWORK_ACL_ADMIN, 179–180
DBMS_SCHEDULER PL/SQL, 116
DDL command, 376
debugOut.println( ) method, 45–46
decode( ) method, 275
decrypt data, local, 106–107
decrypting
at client, 82
connection strings, for storage and reuse, 241–242
data, procedure variables and, 127
encrypting and, data at rest, 287–289
list of connection strings, function to call Java language for, 236–237
of passwords, Java language methods for, 97–100
with private key, 79–80
DECRYPT_MODE, 74
decryptSessionSecretDESPassPhrase( ) method, 98–100
Default procedure, 146
DefaultTableModel, 366–367
defineClass( ) method, 270

DELETE, 15, 22–23, 62, 115, 133
delivery route codes, two-factor, 195
demonstrations
  of data encryption, 142–144
    demonstrating scenarios, 143–144
    observing results, 142–143
    querying employees to see updates, 144
  structures, for testing PKE, 83–84
DEPARTMENT_ID, 128
DES (Data Encryption Standard)
  getting password from Oracle Database, 103–104
  passphrases, decrypting using RSA private key, 98–99
  testing on client, 100–102
deserialization, and serialVersion UID member variable, 230
developers, code template for, 175–176
dictionaries, sensitive data, 316
directory separators, JAR files, 28
distribCode, 196, 200–203
distribToEMail( ) method, 219
distribToPagerURL( ) method, 203, 219
distribToSMS( ) method, 200, 202–203, 219
distribute2Factor( ) method, 193–195, 198–199, 201, 211, 217–219
distributing
  code
    to e-mail, 203
    to pager URL, 202–203
    to SMS, 201–202
  pass codes, to routes, 200–201
  two-factor codes, method to distribute, 195–196
distribution formats, two-factor, 219
DLL (dynamic load library), 150
doFinal( ) method, 68, 74, 79
doTest2( ) method, 159–160, 171
doTest4( ) method, 169
doWork( ) method, 40–43
dynamic load library (DLL), 150
dynamic where clause
  adding to procedures, 396–397
  adding to view, 397–398

E

e-mail
  distributing code to, 203
  sending from Oracle Database, 178–181
    testing, 180–181
    UTL_MAIL package, 178–180
  two-factor delivery to, 183–184
EditAppConns.dataInit( ) method, 389
EJBs (Enterprise Java Beans), 162
employee addresses, getting for delivery of two-factor code, 216–217
Employee IDs, selecting sensitive data for, 125
employee_id, 187–192, 199, 201
employees
  mobile numbers, tables of, 187–189
  querying to see updates, 144
EMPLOYEES data
  selecting
    by last name, 139–140
    by RAW, 140
EMPLOYEES table
  inserting record into, updating sequence, 131–133
  selecting encrypted data from, 135–137
    selecting single row, 139
  sensitive data columns in
    selecting, 122–124
    updating, 125–129
  views of
    public, 22
    sensitive, 22–23
EMPLOYEES_SEQ.NEXTVAL, 132
emp_mobile_nos, 187–188, 190, 192, 313, 318, 349–350, 354, 377
emp_name_ix, 114
encode( ) method, 270–272, 275
encoded APPVER connection string, 276–277
encoded strings, getting APPVER connection string from, 270
encoding APPVER, user password for APVER instance, 320–321
encrypted data, 133–142
  calling Oracle Database to get, 105–106
  getting from Oracle Database, 262
  getting with application, 256
  opening and closing connection, 141
  Oracle Database, testing, 106–107
  password artifacts and, package to get, 93
  preliminary steps, 133–135
    encryption, 134
    main( ) method and method members, 133–134
    reusing callable statements, 134–135
    setting non-default role, 134

encrypted data (*continued*)
running basic key exchange without data encryption, 142
selecting all columns in encrypted string, 137–138
selecting EMPLOYEES data
by last name, 139–140
by RAW, 140
selecting from EMPLOYEES table, 135–139
sending to Oracle database, 107, 138–139
testing encryption failure
with new client keys, 140–141
with new Oracle Database connection, 141
encrypted Java classes, getting APPVER connection string from, 269–270
encrypted strings
getting APPVER connection string from, 270
selecting all columns in, 137–138
EncryptedClassLoader, 269–270
encrypting
data stored on Oracle Database, 285–290
changes to setDecryptConns( ) and getCryptConns( ) methods, 290
DBMS_CRYPTO package, 285
functions to encrypt and decrypt data at rest, 287–289
passwords and keys, 285–286
at rest key store, 286–287
wrap utility, 289–290
and decrypting, data at rest, 287–289
with password, 92–93
with public key, 75–79
with public RSA key, 89–90
SYSDATE function, procedure for, 76–78
encryption, 134
Java 1.5 password-based, evaluating, 88
password-based, negotiated algorithms for, 104–105
testing, 140–141
encryption, password. *See* password encryption
ENCRYPT_MODE, 68, 74
END_TO_END_CLIENTID_INDEX, 155, 159, 174
END_TO_END_STATE_INDEX_MAX, 155, 159, 174
Enterprise Java Beans (EJBs), 162
environments, for development at command prompt, 28–29
equality, class, 240
equals( ) method, 154, 220, 240
equalsIgnoreCase( ) method, 154
error logging
management procedure for, 115–116
tables for
overview, 113–114
table for managing, 114–115
trigger to maintain, 116
error messages, logging, 124
errors, handling, 81–82
errorText.toString( ).getBytes( ), 90–91
etAppAuthConn( ) method, 245
etAppConnections( ) method, 255
etAppContext( ) method, 260
Exception class, 41
exception handling, 40–49
approaches to, 47–49
catch blocks, 47–48
catching and handling exception where it occurs, 47–48
closing local resources in finally block, 49
generic exceptions, 49
governing amount of exception reporting, 48
syntax, 40–47
clean up, 41–47
printing exception message to system output stream, 41
exception messages, printing to system output stream, 41
ExceptionDemoA.java, 40–41
ExceptionDemoB.java, 41–42
Exception.printStackTrace( ) method, 74
exceptions
generic, 49
handling, 53–55
Exception.toString( ) method, 54
exec( ) method, 387
EXECUTE IMMEDIATE, 13, 19, 132, 156–157, 161
EXECUTE object, 144
executeUpdate( ) method, 81, 104–107, 134–135, 138–139, 141
existingEmpComboBox_actionPerformed( ) method, 346–348, 351
existingEmpRadioButton, 343, 345
expectedDomain, 152–154
extRSAPubKey, 67–68, 73–74, 78–79, 90, 121

F

factories, connection pool, 170
feedback, giving from catch blocks, 48

FGA (Fine-Grained Access), 155
f_get_app_role, 213–215
f_get_crypt_conns, 250–251
f_get_crypt_data, 94, 96–97, 106, 108, 124
f_get_crypt_secret_ algorithm, 94
f_get_crypt_secret_ count, 94
f_get_crypt_secret_pass, 94–96, 123
f_get_crypt_secret_salt, 94–96, 122
f_get_decrypt_data, 94, 107–108, 110, 126–127
f_get_oracle_time, 52, 57–58, 60
f_get_rsa_crypt, 75–76, 78, 83–84
FiIeInputStream class, 40, 42–43
finally blocks
  clean up in, 42–47
  closing local resources in, 49
Fine-Grained Access (FGA), 155
first_name, 22, 114, 123–124, 126, 128, 131–132
f_is_cur_cached_cd, 192, 194, 211–212, 214, 219, 250, 294, 318
flush( ) method, 44–46
Frame class, 334
FROM DUAL, 16, 58, 60, 62, 77, 132, 160
f_send_2_factor, 193–195, 211–212, 214, 250, 294
f_set_decrypt_conns, 235–237, 293, 325, 327, 374
f_show_algorithm, 94–95, 100, 104–105, 108, 117
functions, of package bodies, 95
f_unmask, 289–291, 296, 298, 317

G

garbage collection, 37
GenAppException, 49
generic exceptions, 49
GET method, 182, 202–203
get off function, 223–224
getAAConnRole( ) method, 259, 261, 263, 265, 326, 343–344, 380, 399
getAAConString( ) method, 332
getAppAuthConn( ) method, 245–246, 252, 254–255, 259, 399
getAppConnections( ) method, 235, 245, 248, 252, 255, 263, 326, 336, 338, 385
getArray( ) method, 33–36, 39
getBERLIN_WALL_FELL( ) method, 69
getBytes( ) method, 73, 78–79, 89–92, 98–99, 140
getConnection( ) method, 167, 281
getContent( ) method, 203
getCryptConns( ) method, 251, 290
getCryptData( ) method, 92, 138, 147, 249
getCryptSessionSecretDESAlgorithm ( ) method, 91
getCryptSessionSecretDESIterationCount ( ) method, 92
getCryptSessionSecretDESPassPhrase( ) method, 90, 95, 99, 101, 249
getCryptSessionSecretDESSalt ( ) method, 91
getDecryptData( ) method, 97–98, 101, 106, 109, 120, 137, 249
getLocRSAPubExp( ) method, 70, 73–74, 101, 134, 140
getLocRSAPubMod( ) method, 70–71, 73–74, 101, 134, 140–141
getMethod( ) method, 238
getModulus( ) method, 66
getName( ) method, 151–152, 238, 296
getOracleTime( ) method, 52–53, 55, 57, 59–60
getOSUserID( ) method, 152, 155, 159, 174, 176
getPassword( ) method, 390
getProxyConnection( ) method, 164
getPublicExponent( ) method, 66
getRevLvl( ) method, 228–229, 237–238, 258, 295, 323, 331–332, 376
getRSACryptData( ) method, 78–80, 83, 89–91
getRSADecryptData( ) method, 79, 82
getTime( ) method, 138–139
GOTO statement, 273
GRANT appver_admin, 292, 301
GRANT EXECUTE, 215, 223–224, 285, 293, 314, 358, 361, 381–382
GRANT SELECT, 316, 347, 355–356, 360, 368, 377, 380, 393, 398
GRANT statement, 112
GRANT SYSDBA, 312
GRANT_PERMISSION, 182
GUI interface, 74, 333, 345
GUI object, 333, 366

H

hard coding, encoded APPVER connection string, 277
HashMap list, of connections, 239–240
HashMap object, 236, 296
HashMaps, 228
Hashtable, 367–369, 389
hDateFormat.parse( ) method, 353

HR (Human Resources), 18–23
auditing failed attempts to access, 20
permitting application user to acquire, 20
preparing to access, 155–156
security structures for, 121–131
executing HR package specification and body, 131
privileges that enable HR tasks, 121–122
revising procedure to get shared passphrase, 125
security package, 122
selecting all data as single sensitive string, 124
selecting sensitive data for Employee ID, 125
sensitive data columns in EMPLOYEES table, 122–129
SQL injection, 129–131
sensitive data in schema, 21
testing
for normal business hours, 20
for subnet, 19
verifying by procedure, 19
views of EMPLOYEES table
public, 22
sensitive, 22–23
HR schema, 156
HR tables, accessing from application security procedures, 190
HR.EMPLOYEES table, 24, 59, 128, 183, 187, 292, 313, 348, 354, 392
HR.emp_mobile_nos table, 207, 216, 226
hr.hr_sec_pkg, 112, 122, 131, 135, 137–140, 142
hr_public_pkg, 353
hr_pub_pkg.p_select_emp_mobile_nos_by_id, 349, 351
hr_sec_pkg, 112, 125, 127, 130–131, 133, 142, 146
HR.update_job_history, 128
hr.v_amp_mobile_nos, 353
hr.v_employees_public, 22–24, 135, 319, 345–347, 354–355
hr.v_emp_mobile_nos, 216–217, 292, 315, 319, 349, 353–355, 360
hrview_role, 9, 20, 23, 112, 131, 155, 161, 176, 205, 208
HRVIEW_ROLE procedure, 193–194
HR.v_sms_carrier_host, 347
HS_ADMIN_SELECT_ROLE, 17
Human Resources. *See* HR

ID class, 242, 244–246
ID object, 247, 251
IDENTIFIED EXTERNALLY modifier, 159, 225, 291, 356–357, 359
IDENTIFIED GLOBALLY, 159
IDENTIFIED USING sys.p_check_secadm_access, 12
identifiers, client, 155–157
if rs.next( ) method, 52, 56–57, 136–137, 139
IF statement, 19
IMMEDIATE command, 224, 360
IMMEDIATE statement, 360
import oracle.sql.ARRAY, 30, 33
IN parameter, and OUT parameter, 80–81
INCREMENT BY, 132–133
INDEX_SS, 114
initializers, static, 71–72
inner classes
application, 331–332
implementing in OracleJavaSecure class, 229–230
to represent application, 228
innerClassRevLvl, 228–230, 239, 323–324, 331–332, 376, 391
InnerRevLvlClass, 229–230, 253, 323–324, 331–332, 336, 376, 401
INOUT parameter, 81
INSERT command, 22–23, 115, 118, 126, 128, 132, 147, 161, 287
Insert/Update methods, sending encrypted data to Oracle Database for, 138–139
installed services, viewing, 311–312
instance_proxy_users, 361
Integer.parseInt( ) method, 37, 272
integers, 66, 69
Integer.toHexString( ) method, 272
integrity constraints, on EMPLOYEES table, 128–129
InvalidClassException, 220, 240, 297
invoke( ) method, 238
IOException, 41, 43, 45
IP_ADDRESS, 13, 16, 19, 156, 160–161
ip_address, 190–192, 199, 201
isAppverOK( ) method, 336–337
isProxySession( ) method, 164, 174
isTesting boolean, 205
itemVector, 367–372

iterationCount, 86–87, 89, 92, 99
IterationCount, 249

J

J2EE (Java Enterprise Edition), 162
JAAS (Java authentication and authorization system) package, 150
Janitor procedure, 117
JAR (Java Archive) files, 28–29
Java 1.5, password-based encryption bug, 88
Java Archive (JAR) files, 28–29
Java authentication and authorization system (JAAS) package, 150
Java class, 31–32, 56, 64, 121, 178, 269–270, 374, 377
Java client side connection strings, 244–249
  establishing connection for application verification processes, 249
  getting application list of, 249
    calling stored procedure for, 246–247
    from Oracle Database, 245–246
  using static class members to retain APPVER connection and keys, 248–249
Java code, for RSA public key cryptography
  for both client and server, 72–73
  Cipher class, 67–70
  instantiating Connection member from static initializer, 71–72
  RSA public key artifacts, 70–71
  static methods and private constructor, 71
Java command, 29, 283, 382
Java database, 55
Java Database Connectivity (JDBC), 27, 29
Java Development Kit (JDK), 27
Java Enterprise Edition (J2EE), 162
Java file, 28, 30–31, 40, 53, 145
Java language
  archive for use by applications, 145
  for authenticating on separate instance of Oracle Database, 319
  client wallets from, 280–281
  code for password encryption, 86–93
    coding automatic upgrade, 88
    encrypting, 89–93
    evaluating Java 1.5 password-based encryption bug, 88
    initializing static class members, 87–88
    password key, 86–92
  code for RSA public key cryptography, 64–73
    public key, 65–67
    serializing objects, 66
    set of PKE keys, 64–65
  database, method to copy connection strings, 295–299
  function to call to decrypt list of connection strings, 236–237
  methods for password decryption, 97–100
    ancillary methods for array conversion, 99–100
    DES passphrase, 98–99
    method used to show algorithm, 100
    password key, 97–98
  Oracle Database and, dates, 197–198
  packaging template to implement data encryption, 145
  reflection, using to call method, 238
  secure development concepts, 27–50
    development at command prompt, 28–33
    exception handling, 40–49
    JAR file directory separator, 28
    Java code and syntax, 33–39
    Java packages, 28
    JDK, 27
    JVM sandbox, 49–50
    Oracle JDBC, 27
  stored procedures, 51–62
    cleaning up, 60
    for copying connection strings from previous version, 295
    example of, 51–57
    method syntax in, 57–60
    Oracle JVM, 60–62
Java method, calling Oracle Database from, 55–57
  Connection and Statement classes, 55–56
  OracleDriver class, 55
  ResultSet class, 56–57
Java Native Interface (JNI), 150, 269
Java objects, 3, 35, 37, 54, 61, 66, 227, 235
Java policy, delegating to security administrator, 181–182
Java SE 1.5, Oracle JVM based on, 60–61
Java-stored procedure, 178
java TestOracleJavaSecure, 205, 207
Java Virtual Machine. *See* JVM
javac.exe, 29, 31–32, 74–75, 169
javadev/mypkg directory, 30–31
java.exe, 29, 31–33, 38, 75, 169

java.lang.String, 52, 57, 75–76, 95
java_pool_size, 311
java.sql.Date, 139, 146, 197–198, 349, 353
java.sun.com, 27
java.util.Date( ) method, 138–139, 146, 197–198, 349, 353
jbInit( ) method, 333–334, 337, 339–340, 343, 400
JCE class, 65
JDBC (Java Database Connectivity), 27, 29
JDeveloper tool, 7
JDialog class, 331–332, 334–335
JDK (Java Development Kit), 27
JNI (Java Native Interface), 150, 269
JOB_HISTORY table, 128
JOB_ID, 128
jobIDComboBox, 349, 352–353
JPanel, 335, 340–341
JPasswordField type, 390
JTable, 366
just_os_user, 156–157, 161
JVM (Java Virtual Machine)
  and byte code compilation, 31–33
    finding referenced code and classes, 32
    running compiled code, 32–33
    using Java compiler, 31
  sandbox, 49–50, 315

K

key exchange, 75–80
  decrypting data with private key, 79–80
  encrypting data with public key, 75–79
  loading OracleJavaSecure.java file into Oracle Database, 78
  procedure to get SYSDATE function in encrypted form, 76–78
  running basic without data encryption, 142
  stacked calls, 79
key stores, at rest, 286–287
KeyPairGenerator, 64–65
keys
  APPVER connection and, using static class members to retain, 248–249
  client, 63–64, 140–141
  passwords and, 285–286
  PKE, set of, 64–65
  private, decrypting with, 79–80
  public
    building from artifacts, 66–67
    encrypting with, 75–79
    handing across network, 65–66
  temporary method to reset all, 120–121
keySpec class, 67, 89
kFactory, 67

L

last names, selecting EMPLOYEES data by, 139–140
LAST_NAME, 59, 129–131, 139–140
LAST_NUMBER, 131–133
LIKE, 19, 77, 156, 161
links, database
  creating personal as APPSEC user, 314–315
  enabling OJSAAdm user across, 359–361
  to ORCL instance, 313
  permitting appsec user to create, 314
  updating TNSNAMES.ora file for, 313
list( ) method, 153
listConnNames( ) method, 389
lists
  application, of Java client side connection strings, 245–249
  of applications, initializing, 386
  of connection strings
    function to call Java language for decrypting, 236–237
    initializing, 389
    method to store for application, 237
    saving to database, 390
    sending to Oracle Database for storage, 254
    for server side, 250–251
  HashMap, of connections, 239–240
  method to put connection strings in for application, 233–234
  selecting users from, 369
  updating connection string in, 390
  user's
    adding proxy to, 369–370
    removing proxy from, 370
loadjava, 53, 78
local connections, acquiring secadm_role role from SQL*Plus tool, 16
local decrypt data, testing, 106–107
local resources, closing in finally block, 49
locRSAPrivKey, 64–65
locRSAPrivkey, 73
locRSAPrivKey, 73–74, 79, 98
locRSAPubExp, 65–66, 70–71

locRSAPubKey, 64–66
locRSAPubMod, 65–66, 70–71, 121
logging
  errors
    management procedure for, 115–116
    messages, 124
    tables for, 113–114
  thin client trace data, 283–285
logic, obfuscating, 273–274
Login class, 331, 335, 337–338, 340, 345, 374–375, 401
Login screen, application, 330–339
  background processing thread, 336–337
  center method, 333
  closing, 338–339
  constructors, 334–335
  Continue button, 337–338
  inner class, 331–332
  wait while processing modal dialog, 335–336
Login$InnerRevLvlClass, 332, 375, 401
Login.sayWaitDialog, 336, 347–348, 352
logon
  procedure, 223
  trigger, 222

M

mail.jar file, 178
main( ) method
  and method members, 133–134
  planning to pass two-factor codes as argument to, 205
  running, 322–324
  server portion of, coding to test password encryption, 103
  static modifier and, 38–39
  writing, 73–74
makeCryptUtilities( ) method, 67, 70, 87
makeDESKey( ) method, 119–120, 141–143, 247
makeExtRSAPubKey( ) method, 67, 70, 74, 79
makeLocRSAKeys( ) method, 64, 67, 70
makeSessionSecretDESKey( ) method, 88–92, 97–98, 106, 120
makeSessionSecretDESPassPhrase( ) method, 86, 89
Manage Selected Application button, 386–388
manageAppButton_actionPerformed( ) method, 386, 399
MANAGER_ID column, 59
managerIDComboBox, 349, 352
math, binary, 195
MAX(update_ts), 116
maxPassPhraseBytes, 86–87
m_connections, 240, 243–244, 396–397
mCryptSessionSecretDESSalt, 101, 104, 106–107
m.doWork( ) method, 40–42
members, 35–36
MEMORY_TARGET, 311
m_employee_id, 125–128
m_err_txt, 76, 94, 96, 115, 118, 122–123, 127, 193–194
Method class, 152–153
method members, main( ) method and, 133–134
Method object, 238
method syntax, in Java language stored procedures, 57–60
  calling Java language from Oracle Database, 57–58
  installing and testing example code, 58–59
methodName( ) method, 44–47
methods
  overview, 34
  static, and private constructor, 71
  using Java language reflection to call, 238
m.getArray( ) method, 33–34, 36, 38–39
Microsoft Windows, and UNIX/Linux, 1–2
m_log_maint_dt, 116
mobile numbers, employees, tables of, 187–189
modal dialogs, wait while processing, 335–336
modifiers
  IDENTIFIED EXTERNALLY, 159
  public and private, 39
  static, and main( ) method, 38–39
m.setRef( ) method, 38
msg_txt, 180
m_stmt, 360–361, 382
MyApp class, 38
MyApp2.java, 33–34
MyApp4, 52–53, 57–59
MyApp.java, 30–32, 38
mynewpkg.Login$InnerRevLvlClass, 331
mypkg/MyApp.class, 32
MyRef class, 36
MyRef.class, 31–32
MyRef.java file, 30–32
myString, 37, 54

N

NAME, 24, 52, 57, 75–76, 130, 139, 162–163, 166

naming, obfuscating, 275
negotiated algorithms
for coding automatic upgrades, 88
for password-based encryption, 104–105
networks, handing public keys across, 65–66
new AddUser( ) method, 341
new Cipher( ) method, 67
new Date( ) method, 198, 200
new MyApp2, 33, 36, 38–39
new OracleDriver( ) method, 52, 55, 72, 173
new Runnable( ) method, 336–337, 347, 352
new String( ) method, 99
newEmpRadioButton, 343, 345
newSampleData, 73–74, 82
next( ) method, 56, 135
NEXTVAL, 126, 131–132
NO_DATA_FOUND, 194
non-default roles, setting, 134
non-proxy sessions, 156–157
normal business hours, testing for, 20
NOT IDENTIFIED, 10, 12, 17, 112
NTSystem class, 150–155, 159, 161, 163, 173, 175
NTSystem.getDomain( ) method, 153
null, classes and, 36–37
NullPointerException, 41, 54, 326
NUMBER, 76, 94, 96, 115, 118, 126–127, 131–132, 157
NUMBER(6), 211, 216, 350

O

Object object, 41
Object type, 152
ObjectInputStream, 237–238, 240–241, 249, 290, 295, 297–298
ObjectOutputStream, 235, 242
objects, 36, 257
ObjectStream, 66
ObjectWriter class, 66
OCI (Oracle call interface), proxy connections from, 162–166
closing proxy session, 165
configuring, 163
reviewing pool properties, 165
summary, 165–166
viewing proxy connection, 164–165
viewing proxy session, 164
OFF command, 311
OID (Oracle Internet directory), 159
ojdbc6.jar file, 27–28, 31, 33, 55, 75, 81, 107, 169, 171
OJSAAdm user, 355–361
ojsaadm.apver_usr_adm_pkg.p_create_apver_user, 362, 371
ojsaadm.apver_usr_adm_pkg.p_set_apver_proxy_through, 363–364
ojsaadm.instance_proxy_users, 359–360, 362, 364, 368–369, 372, 394, 396
ojs_adm_admin, 344, 355, 358–361, 368, 380, 382
OJSAdmin class, 339, 400
OJSAdmin interface, bootstrapping, 399–402
ojsadmin.Login$InnerRevLvlClass, 345
OJSAdmin.ojsInit( ) method, 340–341, 387
OJSCode class, 270, 272, 302
OJSCode.encode( ) method, 270–271
OJSC.y( ) method, 277
ojsInit( ) method, 334, 336–337, 339–340, 400
open( ) method, 40, 42–43, 49
openProxySession( ) method, 159, 167
operating system. *See* OS
OPS$OSUSER, 159
Oracle call interface. *See* OCI
Oracle command, 8, 56, 59–60
Oracle Database
accessing as identified user, 154–157
assuring client identifier and OS_USER connection trait, 157
auditing activity with client identifier set, 157
client identifier, 155
preparing to access HR data, 155–156
SSO options for programmers, 154–155
updating p_check_hrview_access procedure, 156–157
application Oracle connection, 172
audit trail logs for sensitive view, 24
authenticating on separate instance of, 302–319
changing passwords for SYS and SYSTEM users, 308
configuring CREATE_SESSION_ROLE role and APPSEC user, 312–313
connecting from remote AS SYSDBA privilege, 312
creating new Oracle service, 304–305
creating structures for, 317–319
database, 311
database link, 313–315
granting access for APPSEC user to reach out of JVM security sandbox, 315

increasing quantity of processes, 309–310
installing DBMS_JAVA package, 311
removing from ORCL instance, 319
revoking PUBLIC view grant on sensitive data dictionary views, 316
viewing installed services, 311–312
writing create database command, 305–306
browsing web pages with, 181–182
calling from Java method, 55–57
Connection and Statement classes, 55–56
OracleDriver class, 55
ResultSet class, 56–57
calling to get encrypted data, 105–106
client call to store list of connection strings in, 234–236
client wallet, 277–282
administering security of, 281–282
from Java language, 280–281
from SQL*Plus interface, 280
connection, testing encryption failure with new, 141
encrypting data stored on, 285–290
changes to setDecryptConns( ) and getCryptConns( ) methods, 290
DBMS_CRYPTO package, 285
functions to encrypt and decrypt data at rest, 287–289
passwords and keys, 285–286
at rest key store, 286–287
wrap utility, 289–290
getting application list of Java client side connection strings from, 245–246
getting DES password from, 103–104
getting encrypted data from, 262
getting new structures into, 252
handling errors reported by, 81–82
HR schema user, 21–23
sensitive data in schema, 21
views of EMPLOYEES table, 22–23
individual person users in, 158
Java language
calling from Oracle Database, 57–58
dates, 197–198
loading in Oracle Database, 53
loading OracleJavaSecure.java file into, 78
loading updated OracleJavaSecure class into, 121
logging thin client trace data, 283–285
organization of next few sections, 7
packaging template to implement data encryption for application security structures, 144
placing stub class on, 257–259
procedures, application registry, 242–243
security administrator, 15–20
acquiring secadm_role role from SQL*Plus local connection, 16
Application Security user, 17–18
application user, 18
auditing, 20
HR view role, 18–20
toggling between roles, 16–17
security of
finding test database, 5
working from existing database, 6
sending e-mail from, 178–181
testing, 180–181
UTL_MAIL package, 178–180
sending encrypted data to, 107, 138–139
sending list of connection strings to for storage, 254
SQL*Plus, 7
storing connection strings in, 261
structures for password encryption, 93–97
application security, 93–97
package to get password artifacts and encrypted data, 93
structures, Java objects and, 3
structures supporting two-factor authentication, 185–194
tables, 185–191
testing cache aging, 191–192
two-factor pass codes, 192–193
updating secure application role, 193–194
synonyms, 24–25
SYS user, 8–15
audit trail, 15
Data Dictionary, 15
roles, 8–11
security administrator, 12–14
system privileges, 8
test application user access, 23
testing encrypt and local decrypt data, 106–107
tracing client code, 282–283
updating OracleJavaSecure.java file in, 204
users and schemas, 6–7
Oracle instances, code for, 399
Oracle Internet directory (OID), 159

Oracle JDBC, 27
Oracle JVM (Java Virtual Machine), 60–62
  auto-commit disabled in, 61–62
  based on Java SE 1.5, 60–61
  sandbox, 61
  separate JVM for each Oracle session, 61
Oracle procedure, 205, 209, 239, 289, 294, 303, 362
Oracle scripts, running to add basic database capabilities, 310
Oracle services, creating new, 304–305
Oracle-specific driver, 56, 103
Oracle statement, 53, 57, 62, 119, 217–218
Oracle table, 58, 209, 228, 237, 240, 267, 286
OracleCallableStatement class, 80–81
OracleConnection class, 155, 159–161, 164, 167, 172–175
OracleConnection.isProxySession( ) method, 164
OracleDataSource class, 167, 281, 284–285
OracleDriver class, 52, 55–56, 80, 103, 134, 152, 172, 174
ORACLE_HOME, 278–280, 283, 289, 304, 307, 309, 312, 314
OracleJavaSecure class
  final updates to, 399
  implementing innner classes in, 229–230
  updated, loading into Oracle Database, 121
  updates to, 173–175
    closing proxy connection, 175
    setConnection( ) method, 173–174
OracleJavaSecure.checkFormat2Factor( ) method, 205
OracleJavaSecure.copyAppConnections( ) method, 391
OracleJavaSecure.distribute2Factor( ) method, 193, 218
OracleJavaSecure.getAAConnRole( ) method, 263, 265
OracleJavaSecure.getDecryptData( ) method, 348
OracleJavaSecure.getRSACryptData( ) method, 75
OracleJavaSecure.InnerRevLvlClass, 229, 257
OracleJavaSecure.java file
  loading into Oracle Database, 78
  updating in Oracle Database, 204
OracleJavaSecure.listConnNames( ) method, 389–390
OracleJavaSecure.main( ) method, 73
OracleJavaSecure.makeDESKey( ) method, 125, 141–142
OracleJavaSecure.putAppConnections( ) method, 261, 325, 385, 390, 400
OracleJavaSecure.putAppConnString( ) method, 390
OracleJavaSecure.setAppContext( ) method, 245
OracleJavaSecure.setConnection( ) method, 176
OracleJavaSecure.test2Factor( ) method, 338
OracleJavaSecurity.java file, updating for two-factor authentication, 194–203
  compiling two-factor delivery route codes, 195
  distributing code, 201–203
  Oracle Database and Java language dates, 197–198
  pass codes, 200–201
  routes available to user, 199
  setting company-specific addresses, 194–195
  two-factor codes, 195–197
oracle.jdbc.pool.OracleDataSource, 166, 168, 170
OracleLog.properties file, 284
OracleOCIConnectionPool class, 163–164
OracleOraDb11g_homeTNSListener, 305
OracleServiceapver, 307
OracleServiceORCL, 305
ORACLE_SID, 307–309
oracle.sql package, 31, 57
oracle.sql.ARRAY, 29, 31–32
OracleTypes class, 58, 81, 103–107, 135, 138
OracleTypes.CURSOR, 135
OracleTypes.NUMBER, 81, 104–105, 135, 138
OracleTypes.RAW, 81, 103–106, 135
OracleTypes.VARCHAR, 58, 81, 104–105, 107, 135, 138
orajavsec.Login$InnerRevLvlClass, 331, 378–379, 401
OraSSOTests class, 159–160, 163–164, 168, 171, 174
ORCL instance
  database links to, 313
  removing authentication on separate instance of Oracle database from, 319
OS (operating system), identifying users, 150–154
  accessing Oracle Database, 154–157
  better source of identity, 150
  cross-platform-specific coding with reflection, 151–152
  more stringent identity, 152–154
  with NTSystem and UnixSystem classes, 150–151
OSADMIN role, 326–327

OS_AUTHENT_PREFIX, 159
OS_USER parameter, 157, 160–162, 165
OSUSER role, 327
osUserIDTextField, 347, 351–354
OUT parameter, 58, 80–81, 95, 109, 134, 137, 247, 250, 296
output streams, system, 41
overloaded setConnection( ) method, 174

P

package bodies
  executing, 118–119
  functions, 95
  HR, executing, 131
  procedures, 95–97
package specifications, 93–95
  executing, 118–119
  HR, executing, 131
packages
  application security, updating, 117–119
  to get password artifacts and encrypted data, 93
  Java language, 28
  permitting users to execute in other schemas, 112
  security, 122
  template to implement data encryption, 144–145
pager URLs, distributing code to, 202–203
PAGERID, 195, 202–204
pagers, two-factor delivery to, 184
p_appsec_errors_janitor, 115–116
p_appver_logon, 222–224, 302
paramSpec, 88–89, 92, 98
parent.setVisible( ) method, 342
pass codes
  caching in Oracle Database, 201
  checking validity of current, 200
  distributing to routes, 200–201
passphrases, shared, 125
password encryption
  approach to, 85–86
  bugs, Java 1.5, 88
  client/server, coding to test, 102–107
  Java code for, 86–93
    coding automatic upgrade, 88
    encrypting, 89–93
    evaluating Java 1.5 password-based encryption bug, 88
    initializing static class members, 87–88
    password key, 86–92
  Java language methods for decryption, 97–100
    ancillary methods for array conversion, 99–100
    DES passphrase, 98–99
    method used to show algorithm, 100
    password key, 97–98
  negotiated algorithms for, 104–105
  Oracle Database structures for, 93–97
    application security, 93–97
    package to get password artifacts and encrypted data, 93
  testing DES encryption on client, 100–102
password keys, 97–98
  building, 119–120
  generating, 88–89
  returning artifacts to client, 90–92
  sharing artifacts of, 86–87
password-protected roles, 11
passwords
  application, editing, 321–322
  APPVER user, encoding for APVER instance, 320–321
  calculating size of, 87
  changing for SYS and SYSTEM users, 308
  generating artifacts and, 86–87
  and keys, 285–286
PATH, 28–29, 74, 145
PBEKeySpec, 89, 100
PBEWithMD5AndDES, 88, 102, 104, 107–108
p_check_hrview_access procedure, updating
  non-proxy sessions, 156–157
  proxy sessions, 161
p_check_role_access, 213–215, 255–256, 260, 303, 344, 380
p_check_secadm_access, 12–14
p_copy_app_conns, 294–295, 299, 374
p_count_class_conns, 239, 243–244, 296–297
p_create_apver_user, 360–361
p_create_template_class, 374–377
p_create_user_many, 356–357, 360–362, 371
p_create_user_once, 356–357, 360–362, 371
p_drop_apver_user, 360–361, 364
p_drop_proxy_through, 358, 361, 363–364, 371–372
p_drop_user, 357–358, 361, 364
permissions, granting access to UTL_MAIL package, 179–180

p_error_log, 118
PFILE file, 309
P_GET_APP_CONNS, 215
p_get_app_conns, 246–247, 250–251, 256, 379
p_get_class_conns, 239–240, 243–244, 296–297, 396, 398
p_get_des_crypt_test_data, 94, 96–97, 105–106, 108–109, 117
p_get_emp_2fact_nos, 216–218
p_get_emp_2fact_nos., 318
p_get_emp_2fact_nos, 318
p_get_rsa_crypt_sysdate, 76, 81–84
p_get_shared_pass_phrase, 108
p_get_shared_passphrase, 344, 354
p_grant_appver_conns_role, 381–382, 384–385
p_grant_apver_appver_conns, 363–364, 382
PickAppManage class, 344, 386, 399
PKE (public key encryption), 63–84
  clients, generating keys on, 63–64
  key exchange, 75–80
    decrypting data with private key, 79–80
    encrypting data with public key, 75–79
    loading OracleJavaSecure.java file into Oracle Database, 78
    procedure to get SYSDATE function in encrypted form, 76–78
    stacked calls, 79
  RSA public key cryptography, 64–73
  testing
    on client, 73–75
    on client and server, 80–84
pkg4, 52–53, 57, 59
PL/SQL file, 178
plain old telephone systems (POTS), 185
p_log_error, 118, 123–124, 127, 180, 196
policies, Java, 181–182
pool/cache, configuring, 166–167
PoolDataSource class, 170
PoolDataSourceFactory class, 170
port servers, getting APPVER connection string from, 268
POTS (plain old telephone systems), 185
PRAGMA AUTONOMOUS_TRANSACTION, 115, 224, 357–358, 381
preferred two-factor delivery, 185
prepareCall( ) method, 105
PreparedStatement, 130
p_revoke_appver_conns_role, 381–382
primitives, 37
printing exception message, to system output stream, 41
PrintStream, 44–46
PrintWriter, 90–91
private constructors, static methods and, 71
Private database, 359
private keys, decrypting with, 79–80
private modifiers, public modifiers and, 39
private RSA keys, decrypting DES passphrases using, 98–99
private static BigInteger locRSAPubMod, 65, 70
privileges
  enabling HR tasks, 121–122
  to load Java stored procedures, acquiring, 53
  system
    granted to security administrator users, 14
    granting more to application security user, 112
    SYS users, 8
procedure variables, and data decryption, 127
procedures
  adding dynamic where clause to, 396–397
  of package bodies, 95–97
ProcessBuilder class, 387
profiles, with limits and unlimited, 221–222
programmers, SSO options for, 154–155
Properties class, 153, 159, 163–164, 281
Property class, 167
providedClass, 238, 241, 295–297
proxies
  adding to user's list, 369–370
  removing from user's list, 370
  selecting available in table, 368–369
  user's, saving updates to, 370–371
proxy connections
  closing, 175
  for example application, 172–173
  from OCI connection pool, 162–166
    closing proxy session, 165
    configuring, 163
    reviewing pool properties, 165
    summary, 165–166
    viewing proxy connection, 164–165
    viewing proxy session, 164
  viewing, 262
proxy sessions, 158–162
  auditing, 161–162
  closing, 165
  establishing, 159–160
  IDENTIFIED EXTERNALLY modifier, 159
  individual person users in Oracle Database, 158

from thin client connection pool, 166–169
configuring pool/cache, 166–167
reviewing pool properties, 168–169
statement caching, 167
summary, 169
viewing proxy session, 167–168
updating p_check_hrview_access procedure, 161
viewing, 164
proxy, through application verification, 225–226
PROXY_SESSION, 160, 162, 165, 167, 175
PROXYTYPE_USER_NAME, 159, 164, 167, 174
PROXY_USER_NAME, 159–160, 163–164, 167, 174
prvtmail.plb file, 178
p_select_employee_by_id_sens, 125, 139, 348
p_select_employee_by_ln_sens, 130, 139–140
p_select_employee_by_raw_sens, 130, 140
p_select_employees_secret, 124, 137
p_select_employees_sensitive, 122–123, 135, 146, 206, 256, 262, 327
p_set_apver_proxy_through, 361, 364
p_set_class_conns, 242–243, 299, 396–398
p_set_proxy_through, 358, 361, 363, 371–372
PUBLIC, 9, 15, 20, 22, 59, 135, 142, 156, 161, 193
public class MyApp2, 33–34
public key encryption. *See* PKE
public keys
artifacts
building public keys from, 66–67
RSA, 70–71
encrypting with, function for, 75–76
handing across network, 65–66
public modifiers, and private modifiers, 39
public MyApp2( ) method, 34
public RSA keys, encrypting with, 89–90
PUBLIC view, revoking grant on sensitive data dictionary views, 316
public views, of EMPLOYEES table, 22
PublicKey class, 65–66
pullIDFromParens( ) method, 346, 349, 361
p_update_2fact_cache, 217–218
p_update_employees_sensitive, 353
p_update_emp_mobile_nos, 349–351, 353–354
putAppConnections( ) method, 234–235, 252, 265, 294, 324, 327, 385, 400–401
putAppConnString( ) method, 249, 252, 390, 401
putAppConString( ) method, 399
putConnections( ) method, 400

Q

querying employees, 144

R

random characters, 87
RAW parameter, 80, 82, 138, 295
RAW type, 76, 90–91, 123, 125, 140, 227, 237, 240
readObject( ) method, 238, 240–241, 291, 295, 297–298
records, inserting into EMPLOYEES table, 131–133
reEnterLabel, 336, 338
referenced classes, and referenced code, 32
referenced code, and classes, 32
reflection
cross-platform-specific coding with, 151–152
Java language, using to call method, 238
Register Application button, 382–385
registerButton_actionPerformed( ) method, 382, 385
registerOutParameter( ) method, 104
registration, finding out whether class has been registered, 239
registries, class, 377–379
registry tables, application connection, 227
RegNewApp, 344, 399–401
Remote Method Invocation (RMI), 268
remove( ) method, 370
removeAppConnString( ) method, 293, 322, 324, 399
REPLACE command, 376
REPLACE statement, 290
reporting exceptions, governing amount of, 48
requirements, for Microsoft Windows and UNIX/Linux users, 1–2
resetKeys( ) method, 119–121, 140, 248, 263, 343–344
resetting all keys, temporary method for, 120–121
RESOURCE, 121
resources, local, 49
results
of data encryption test, 142–143
of testing DES on client, 102
of testing PKE, observing, 82–83
of testing two-factor authentication, 206–207

ResultSet class, 56–57, 234, 253, 261, 346, 361, 364, 367, 369, 371
ResultSet.getDate( ) method, 198
ResultSet.getString( ) method, 56
ResultSet.next( ) method, 56, 137
RESULTSET_OUT, 124
RESULTSET_TYPE, 122–123
RETURNCODE, 24
RevLvlClassIntfc interface, 228, 231
REVOKE CONNECT THROUGH, 357–358
revokeUserButton_actionPerformed( ) method, 364
rewrite and refactor method, to distribute two-factor code, 216–219
  changes to, 218–219
  getting employee addresses for delivery, 216–217
  stored procedure to update two-factor code cache, 217–218
Rivest, Shamir, and Adleman. *See* RSA
RMI (Remote Method Invocation), 268
RMIClassLoader, 257, 270
ROLE command, 208
roles
  administrative, for application verification, 292–293
  administrator, application verification, 374
  application authentication, 259–260
  Application Security users, 17–18
  getting application connection with, 261
  non-default, setting, 134
  secure application
    getting connections for application and associated, 255–256
    role procedure, for multiple applications, 210–215
    updating, 193–194
  security administrator, 12–14
    acquisition of, 13–14
    verification of, 12–13
  SYS users, 8–11
    create-session role, 10
    DBA role, 9
    password-protected roles, 11
  toggling between, 16–17
routes
  available to user, 199
  codes, two-factor delivery, compiling, 195
  distributing pass codes to, 200–201
rows, selecting single in EMPLOYEES table, 139
RSA (Rivest, Shamir, and Adleman)
  Java code for, 64–73
    for both client and server, 72–73
    Cipher class, 67–70
    instantiating Connection member from static initializer, 71–72
    public key, 65–67
    RSA public key artifacts, 70–71
    serializing objects, 66
    set of PKE keys, 64–65
    static methods and private constructor, 71
  private keys, decrypting DES passphrases using, 98–99
  public keys, encrypting with, 89–90
RSACipher class, 86
RSAPublicKey class, 64–67
RSAPublicKey type, 65
RSAPublicKeySpec, 67
rs.getString( ) method, 56
rtrnRAW, 90–92
rtrnString, 70–71, 79–80, 97–98, 100, 120, 153
Run command, 311
run( ) method, 337, 347–348, 351–352, 361–362, 369–370
Runtime.exec( ) command, 387, 399

S

SALARY, 21–23, 123, 127, 137, 143, 146
salaryTextField, 349, 353
SALT_LENGTH, 86–87
sampleData, 73–74
sandboxes, JVM, 49–50, 61, 315
saveButton_actionPerformed( ) method, 352, 362, 370, 390
sayWaitDialog, 335–337, 347–348, 352, 369, 382
schemas, users and, 6–7, 112
scripts, execution of, 398–399
secadm_role, 9, 12, 15–16, 18–19, 111, 179, 181–182
SecretKeyFactory, 89
secure application roles
  getting connections for application and associated, 255–256
  planning to acquire, 205–206
  procedure for multiple applications, 210–215
    moving test for SSO to separate function, 212
    rebuilding two-factor cache table, 211

updating two-factor code functions to use application ID, 211
updating, HRVIEW_ROLE procedure, 193–194
secure client/server data transmission, testing, 107–110
SECURE_DML, 133
SecureRandom class, 64–65, 86–88, 108
security
Add/Modify User functional screen, 341–354
employees, 347–354
initializing data selection components, 345–347
instantiating AddUser screen, 342–345
adding authentication credentials, 301
administration
interface, 329
menu, 339–341
application, adding more space for, 226
Application Assignment screen, 365–371
initializing data selection components, 366–368
proxies, 368–371
selecting user from list, 369
application Login screen, 330–339
application inner class, 331–332
background processing thread, 336–337
center method, 333
closing, 338–339
constructors, 334–335
Continue button, 337–338
wait while processing modal dialog, 335–336
Application Registration screen, 372–385
App Class button, 375–377
application verification administrator role, 374
Register Application button, 382–385
security table access analysis, 379–382
tables of application administrators and application to class registry, 377–379
Application Selection screen, 385–388
initializing list of applications, 386
selecting Manage Selected Application button, 386–388
bootstrapping OJSAdmin interface, 399–402
of client wallets, administering, 281–282
Connection String Copy screen, 391
Connection String Editor, 388–390
initializing list of connection strings, 389
saving list of connection strings to database, 390
selecting existing connection string, 390
updating connection string in list, 390
connection strings, managing for applications, 291–299
final updates to OracleJavaSecure class, 399
hiding APPVER connection string, 267–277
generating encoded, 276–277
getting from encoded string, 270
getting from encrypted Java class, 269–270
getting from encrypted string, 270
getting from JNI, 269
getting from secondary source, 268
obfuscating algorithm/code, 272–275
using encode method, 270–272
limiting administrators to certain applications, 391–398
adding dynamic where clause, 396–398
VPD, 392–395
Oracle Database, 5–25
authenticating on separate instance of, 302–319
client wallet, 277–282
encrypting data stored on, 285–290
finding test database, 5
logging thin client trace data, 283–285
tracing client code, 282–283
working from existing database, 6
scripts execution and code compilation, 398–399
single Oracle instance code, 399
testing, 320–324
editing application passwords, 321–322
encoding APPVER user password for APVER instance, 320–321
running main, 322
from TestOracleJavaSecure application, 324–327
updating application security structures, 301–302
User Administration screen, 354–364
OJSAAdm user, 355–359
revoking user access to run applications, 364
saving updates to administrative privileges, 362–364
selecting existing user, 361–362

security administrator users
role, 12–14
acquisition of, 13–14
verification of, 12–13
system privileges granted to, 14
security administrators, 15–20, 111–112
acquiring secadm_role role from SQL*Plus local connection, 16
Application Security user, 17–18
application user, 18
auditing, 20
delegating Java policy to, 181–182
granting more system privileges to application security user, 112
HR view role, 18–20
permitting application user to acquire, 20
testing, 19–20
verifying by procedure, 19
permitting users to execute packages in other schemas, 112
toggling between roles, 16–17
security packages, 122
security table access, analysis of, 379–382
SEGMENT SPACE MANAGEMENT AUTO, 310
SELECT _CATALOG_ROLE, 17
SELECT command, 312
SELECT INTO, 116, 127
SELECT SEQUENCE.NEXTVAL, 133
SELECT statement, 24, 59–60, 318, 394
SELECT_CATALOG_ROLE, 15–16
sensitive data
columns in EMPLOYEES table
selecting, 122–124
updating, 125–129
dictionaries, revoking PUBLIC view grant on views of, 316
in HR schema, 21
selecting for Employee ID, 125
sensitive strings, selecting all data as single, 124
sensitive views
audit logs for, 24
of EMPLOYEES table, 22–23
SEQUENCE.NEXTVAL, 131
sequences, updating, 131–133
Serializable interface, 66, 84, 229, 231
serializing objects, 66
serialVersionUID, 228–230, 239, 258, 323, 331–332, 376
server application, saving connection strings for, 236–244
building class from byte array, 237–238
decrypting for storage and reuse, 241–242
finding out if registry entry exists for server application, 243–244
finding out whether class has been registered, 239
function to call Java language to decrypt list of, 236–237
getting application ID class and HashMap list of connections, 239–240
getting list of for registered application, 244
method to store list of for application, 237
Oracle Database procedures, 242–243
storing connsHash object for server application, 242
testing for class equality, 240
using Java language reflection to call method, 238
server side connections, lists of, 250–251
servers
clients and
coding to test password encryption, 102–103
Java code for both, 72–73
testing PKE on, 80–84
port, getting APPVER connection string from, 268
server portion of main( ) method, coding to test password encryption, 103
SERVICE_NAME, 163, 166
SESSION_ROLES, 16
sessionSecretDESAlgorithm, 88–89, 91, 99–100, 121, 136–137, 142
sessionSecretDESKey, 88–90, 92, 98, 120–121, 241, 249, 251
sessionSecretDESPassPhrase, 343–344, 349, 354
sessionSecretDESPassPhraseChars, 86, 89–90, 99, 121
SESSIONS_PER_USER, 221, 309
SESSION_USER, 156, 160–161, 164
Set class, 219, 231
SET ROLE, 10–11, 16–17, 19, 52, 75, 93, 113, 161
SET ROLE ALL, 11, 18, 53
SET ROLE appsec_role, 18, 52–53, 75, 93, 113, 190
setAppContext( ) method, 231–232, 253, 261, 263, 337, 385
setAppVerConnection( ) method, 246, 249, 252, 277, 320–321, 398
setAutoCommit( ) method, 62
setConnection( ) method, 72, 172–174, 176, 246, 277, 281, 399
setConnectionCacheProperties( ) method, 167

setDataVector( ) method, 367, 370
setDecryptConns( ) method, 236–238, 241, 290
setEnabled( ) method, 341
setImplicitCachingEnabled(true), 167
setNull, 81, 103–106, 135, 138
setPoolConfig( ) method, 163
setRef( ) method, 34–35, 39
shared passphrases, revising procedure to get, 125
shared secret password encryption. *See* password encryption
Short Message Service. *See* SMS
showAlgorithm( ) method, 100–101
SIM (subscriber identity module) card, 184
simple mail transport protocol (SMTP) commands, 178
SimpleDateFormat class, 198
SimpleDateFormat.format( ) method, 198
SimpleDateFormat.parse( ) method, 198
single sign-on. *See* SSO
SMS (Short Message Service), 178
  carrier host tables, 185–186
  distributing code to, 201–202
sms_carrier_cd, 185–189, 199
sms_carrier_host, 185–188, 199
sms_carrier_url, 186–187, 199
SMTP (simple mail transport protocol) commands, 178
SocketPermission, 181–182
SOURCE statement, 319
SQL command, 7, 129, 182, 203, 285, 308, 320, 358
SQL database, 56, 61
SQL (Structured Query Language)
  Developer tool, 7
  injection, 139–140
    avoiding, 129–130
    demonstrating failure of in stored procedure, 130–131
  Plus tool, 7, 16
SQLCODE, 76, 96–97, 115, 123, 127
SQLERRM, 76, 96–97, 115, 123, 127
SQLException, 41, 53–54
SQL*Plus interface, client wallets from, 280
SSO (single sign-on), 149–176
  application use of, 171–176
    code template for developers, 175–176
    example SSO application, 172–173
    updates to OracleJavaSecure class, 173–175
  authentication layers, 149–150
  connection pools, 162–171
    proxy connections, 162–166
    proxy sessions, 166–169
    UCP, 169–171
  identifying OS user, 150–154
    accessing Oracle Database, 154–157
    better source of identity, 150
    cross-platform-specific coding with reflection, 151–152
    more stringent identity, 152–154
    with NTSystem and UnixSystem classes, 150–151
  proxy sessions, 158–162
    auditing, 161–162
    establishing, 159–160
    IDENTIFIED EXTERNALLY modifier, 159
    individual person users in Oracle Database, 158
    updating p_check_hrview_access procedure, 161
  testing for, moving to separate function, 212
stacked calls, 79
STARTUP command, 305
Statement class, 55–56, 62, 80, 364
Statement object, 54
Statement.getBytes( ) method, 242
statements, callable, 134–135
static class members
  initializing, 87–88
  using to retain APPVER connection and keys, 248–249
static initializers, instantiating Connection members from, 71–72
static methods, and private constructor, 71
static modifier, and main( ) method, 38–39
stmt.executeQuery( ) method, 56
stmt.setNull( ) method, 81
storage
  and reuse, decrypting connection strings for, 241–242
  sending list of connection strings to Oracle Database for storage, 254
stored procedures
  calling for getting application list of Java client side connection strings, 246–247
  demonstrating failure of SQL injection in, 130–131
  Java language, for copying connection strings from previous version, 295
  to update two-factor code cache, 217–218