

storing  
     connection strings, in Oracle Database, 261  
     connsHash object for server application, 242  
     database parameters in database, 308–309

String class, 37–38, 57, 91, 140

String type, 41, 56–57, 60

StringBuffer, 232, 272–273

String.equals() method, 38, 200

String.getBytes() method, 74, 274

strings  
     connection  
         application, 256  
         APPVER, 267–277  
         copying from previous version, 293  
         deleting, 293  
         initializing list of, 389  
         Java client side, 244–249  
         lists of for server side, 250–251  
         managing for applications, 291–299  
         running main to copy to new version, 323–324  
         saving, 233–244, 390  
         selecting existing, 390  
         sending list of to Oracle Database for storage, 254  
         set of for applications, 227–233  
         storing, 261  
         updating in list, 390  
     encoded, getting APPVER connection string from, 270  
     encrypted  
         getting APPVER connection string from, 270  
         selecting all columns in, 137–138  
         sensitive, selecting all data as single, 124  
     String.toUpperCase() method, 37  
     String.trim() method, 233  
     stringValue() method, 136–137, 143  
     Structured Query Language. *See* SQL  
     stub class, placing on Oracle Database, 257–259  
     subnets, testing for, 19  
     subscriber identity module (SIM) card, 184  
     Successful parameter, 283  
     super() method, 139, 335  
     SuppressWarnings() method, 246  
     SwingUtilities class, 337  
     SwingUtilities.invokeLater() method, 337, 348, 402  
     synchOnThis, 46–47  
     synonyms, 24–25  
     syntax, code and, Java language, 33–39

SYS users, 8–15  
     audit trail, 15  
     changing passwords for, 308  
     Data Dictionary, 15  
     roles, 8–11  
         create-session, 10  
         DBA, 9  
         password-protected, 11  
     security administrator  
         role, 12–14  
         system privileges granted to, 14  
         user, 12  
         system privileges, 8  
     sys.all\_java\_classes, 376–377  
     SYS.ALL\_SOURCE, 225  
     SYS.ALL\_USERS, 225  
     sys.appver\_conns\_role\_pkg, 381–382, 384–385  
     SYS\_CONTEXT, 13, 19, 156, 211, 217, 262, 383, 394, 396, 398  
     SYSDATE function, procedure to get in encrypted form, 76–78  
     sys.p\_check\_secadm\_access, 12–16, 24, 111  
     sys.proxy\_users, 356–360, 381  
     System database, 306, 309  
     system privileges  
         granted to security administrator users, 14  
         granting more to application security user, 112  
         SYS users, 8  
     System property, 281  
     SYSTEM users, changing passwords for, 308  
     System.exit() method, 40, 42, 74, 338  
     System.out, 40–41, 99, 104, 106, 135, 137, 152–153, 165  
     sys.usr\_role\_adm\_pkg, 357–358, 360–364, 371–372

■ T

tables  
     of application administrators, and application to class registry, 377–379  
     of employee mobile numbers, 187–189  
     for error logging, 113–115  
     HR, accessing from application security procedures, 190  
     security, analysis of access to, 379–382  
     selecting available proxy in, 368–369  
     SMS carrier host, 185–186  
     two-factor codes cache, 190–191

- t\_app\_class\_id, 378
- t\_app\_class\_registry, 393
- t\_app\_conn\_registry, 239, 267, 394–396
- t\_application\_admins, 377
- t\_application\_key, 286–287
- t\_application\_registry, 210, 212–213, 384, 394
- t\_appsec\_errors\_iar, 116
- t\_appsec\_errors\_maint, 114–117
- templates
  - code, for developers, 175–176
  - packaging, to implement data encryption, 144–145
- t\_emp\_mobile\_nos, 199, 341
- test code, editing, 204
- test2Factor() method, 338
- testAsClientAndServer, 97–98, 101, 120
- test\_data, 94–97
- testing
  - authentication, 251–256, 260–262
    - adding application connection strings, 256
    - connections, 254–256
    - getting application connection with role, 261
    - Oracle Database, 252–262
    - for second application, 257–259
    - setting application context, 253–254, 261
    - steps of, 252–253
    - viewing proxy connection, 262
  - cache aging, 191–192
  - for class equality, 240
  - client/server password encryption, coding to test, 102–107
  - data encryption, 142–144
    - demonstrating scenarios, 143–144
    - observing results, 142–143
    - querying employees to see updates, 144
    - in transit, methods for, 119–121
  - DES, testing on client, 100–102
  - encryption failure, 140–141
  - installing example code and, 58–60
  - for normal business hours, 20
  - Oracle Database encrypt and local decrypt data, 106–107
  - PKE
    - on client, 73–75
    - on client and server, 80–84
    - issues with completing, 82
  - secure client/server data transmission, 107–110
  - security, 320–324
    - editing application passwords, 321–322
    - encoding APPVER user password for APVER instance, 320–321
    - running main, 322
    - from TestOracleJavaSecure application, 324–327
  - sending e-mail from Oracle Database, 180–181
  - for SSO, moving to separate function, 212
  - for subnet, 19
  - trigger to maintain error log table, 117
  - two-factor authentication, 203–207
    - editing test code, 204
    - and observing results, 206–207
    - planning, 205–206
    - updating OracleJavaSecure.java file in Oracle Database, 204
- testingOnServer, 80, 103, 108
- testojs.TestOracleJavaSecure, 320, 326–328, 378
- TestOracleJavaSecure application, testing
  - security from, 324–327
    - compiling and running as administrative user, 326–327
    - running as non-administrative user, 327
- TestOracleJavaSecure class, 133–134, 140, 142, 145, 205, 207, 260–261, 263
- TestOracleJavaSecure.java file, 133
- TestOracleJavaSecure.main() method, 140
- TestWallet class, 283
- TestWallet.java file, 281
- thin client connection pool, proxy sessions from, 166–169
  - configuring pool/cache, 166–167
  - reviewing pool properties, 168–169
  - statement caching, 167
  - summary, 169
  - viewing proxy session, 167–168
- thin client implementations, UCP as, 170–171
- thin client trace data, logging, 283–285
- this() method, 342–343
- this\_employee\_id, 392
- this\_WindowClosing() method, 338, 344
- threads, background processing, 336–337
- throws Exception, 40–42, 44, 46, 64, 67, 89–90, 98
- timeString = rs.getString(1), 52, 54, 56
- tmaDate, 200
- TNSlistener, 309
- TNSListener, 311
- TNSlistener, 311
- TNSListener, 312

TNSNAMES.ora file, 313  
 TOAD (Tool for Oracle Application Developers), 7  
 toByteArray() method, 235, 242  
 TO\_CHAR method, 19–20, 76, 123–124, 132, 156, 161, 198–199  
 TO\_DATE, 198  
 Tool for Oracle Application Developers (TOAD), 7  
 topMenuPanel, 340–341  
 toString() method, 41, 54, 70, 73, 82, 104, 106  
 toUpperCase() method, 37, 153  
 trace data, thin client, 283–285  
 Trace file, 283  
 tracing client code, 282–283  
 transit, methods for using and testing data  
   encryption in, 119–121  
   building password key, 119–120  
   temporary method to reset all keys, 120–121  
 traverse() method, 274–275  
 triggers  
   logon, 222  
   to maintain error log table, 116–117  
 t\_screen\_appver\_access, 222, 302, 318  
 t\_two\_fact\_cd\_cache, 190, 192–193, 195, 205, 211  
 t\_two\_factor\_cd\_cache, 191  
 two-factor authentication, 177–208  
   Oracle Database, 185–194  
     browsing web pages with, 181–182  
     sending e-mail from, 178–181  
     tables, 185–191  
     testing cache aging, 191–192  
     two-factor pass codes, 192–193  
     updating secure application role, 193–194  
   process of, 183  
   security considerations with, 183–185  
   testing, 203–207  
     editing test code, 204  
     and observing results, 206–207  
     planning, 205–206  
     updating OracleJavaSecure.java file in Oracle Database, 204  
   updating OracleJavaSecurity.java file for, 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  
 two-factor cache tables, rebuilding, 211  
 two-factor codes, 196–197  
   cache tables, 190–191  
   functions, updating to use application ID, 211  
   method to distribute, 195–196  
   planning to pass as argument to main() method, 205  
   rewrite and refactor method to distribute, 216–219  
     changes to, 218–219  
     getting employee addresses for delivery, 216–217  
     stored procedure to update two-factor code cache, 217–218  
     user-input, formatting, 232–233  
 two-factor delivery  
   to cell phone, 184–185  
   compiling route codes, binary math, 195  
   to e-mail, 183–184  
   to pagers, 184  
   preferred, 185  
 two-factor distribution formats, update to, 219  
 two-factor pass codes  
   sending, 193  
   verifying current cached, 192–193  
 twoFactorCodeTextField, 336–338  
 twoFactorAuth, 219, 231, 245, 247, 253–255, 259, 261, 325  
 twoFactorAuthChars, 196–197  
 two\_factor\_cd, 190–194, 201  
 twoFactorLength, 232–233

## ■ U

UCP (universal connection pool), 169–171  
   compile/run command, 169–170  
   connection, 171  
   connection pool factory, 170  
   summary, 171  
   thin client implementation, 170–171  
 UID (user ID) member variables, serialVersion, 230  
 UNIQUE, 114–115, 117, 128, 139  
 universal connection pool. *See* UCP  
 UNIX/Linux, Microsoft Windows and, 1–2

UnixSystem class, 150–151, 161, 163, 167, 173, 175–176  
 UPDATE command, 147  
 Update statement, 58  
 updateButton\_actionPerformed() method, 390  
 updates  
     to administrative privileges, saving, 362–364  
     final, to OracleJavaSecure class, 399  
     to user's proxies, saving, 370–371  
 update\_ts, 180  
 UPDATE\_TS column, 113, 115  
 URL class, 202  
 URLClassLoader class, 257, 268–270  
 USE\_EMAIL, 195, 203  
 USE\_PAGER, 195, 203  
 USER#, 181–182  
 User Administration screen, 354–364  
     OJSAAdm user, 355–361  
     revoking user access to run applications, 364  
     saving updates to administrative privileges, 362–364  
     selecting existing user, 361–362  
 USER command, 14–15, 21, 24, 112, 158, 161, 163, 226, 305  
 user ID (UID) member variables, 230  
 user-input two-factor code, formatting, 232–233  
 USER\$ table, 181  
 userComboBox\_actionPerformed() method, 361, 369  
 useRef, 33, 35–36, 39  
 user\_id, 187–190, 192, 199  
 userInputEmpID, 130  
 USERNAME, 59, 150, 162  
 userProxiesTM.setDataVector() method, 370  
 users  
     access to test application, 23  
     administrative  
         compiling TestOracleJavaSecure application and running as, 326–327  
         managing connection strings for applications, 291–292  
     application, 18–20  
     for application authorization, 221–226  
         application verification, 222–226  
         function to find database user, 224–225  
         get off function, 223–224  
         profile with limits and unlimited, 221–222  
     individual person in Oracle Database, 158  
     lists  
         adding proxy to, 369–370  
         removing proxy from, 370

        non-administrative, running  
             TestOracleJavaSecure application as, 327  
     OJSAAdm, 355–361  
     permitting to execute packages in other schemas, 112  
     proxies, saving updates to, 370–371  
     revoking access to run applications, 364  
     routes available to, 199  
         and schemas, 6–7  
         selecting existing, 361–362  
         selecting from list, 369  
 user\_sequences, 131–132  
 USER\_USERS, 15  
 USE\_SMS, 195, 202  
 usr\_role\_adm\_pkg, 356, 358–359, 361  
 utAppConnections() method, 254  
 Utility.pullIdFromParens() method, 348  
 UTL\_MAIL package  
     configuring database to use, 310–311  
     granting access to, 179–180  
     installing, 178  
 utlmail.sql file, 178

## ■ V

values, 34–35, 242–243  
 v\_app\_class\_id, 379, 400  
 v\_app\_conn\_registry, 239, 242, 251, 294, 373, 380, 393, 396, 398, 401  
 v\_application\_admin, 380, 400–401  
 v\_application\_key, 286–288  
 v\_application\_registry, 213–215, 345, 374, 379–380, 384, 400–401  
 v\_appsec\_errors, 215  
 VARCHAR2, 52, 58–59, 75, 96, 118, 127, 190, 192, 194  
 variables, procedure, 127  
 v\_employees\_public, 22, 24, 156, 318  
 verification  
     application, establishing connection for, 249  
     of applications, 255  
         administrative role for, 292–293  
         administrator role, 374  
         auditing, 226  
         logon procedure, 223  
         logon trigger, 222  
         proxy through, 225–226  
         user for, 222

verifying  
 HR view role by procedure, 19  
 security administrator role, 12–13

views  
 adding dynamic where clause to, 397–398  
 audit logs for sensitive, 24  
 of EMPLOYEES table, 22–23

Virtual Private Database (VPD), 392–395

virtual private network (VPN), 149

VPD (Virtual Private Database), 392–395

VPN (virtual private network), 149

V\$PWFILERS, 8

v\_salary, 126–128

v\_sms\_carrier\_host, 185–186, 190, 199

v\_sms\_carrier\_host@orcl\_link, 318–319

v\_two\_fact\_cd\_cache, 191–192, 199, 201, 212, 214, 216, 218, 319, 327

## ■ W

wait while processing modal dialog, 335–336

WALLET\_LOCATION, 279

WALLET\_OVERRIDE, 279, 283

wallets, client, 277–282  
 administering security of, 281–282

from Java language, 280–281

from SQL\*Plus interface, 280

web pages, browsing with Oracle Database, 181–182  
 delegating Java policy to security administrator, 181–182  
 permitting application security user to read, 182

where clause, dynamic  
 adding to procedures, 396–397  
 adding to view, 397–398

while rs.next() method, 57, 136, 139–140, 346

Windows command, 305

Windows, Microsoft. *See* Microsoft Windows, and UNIX/Linux

WITH GRANT OPTION, 393, 397

wrap utility, 289–290

## ■ X

x() method, 275

x.toString() method, 40–43, 52, 54, 98, 100

## ■ Y, Z

y() method, 275