

# Index

## A

**ACCDB file format, 2, 29**

**ACCDB files, 253, 278, 281**

**ACCDE file format, 2**

**ACCDE files, 331**

**Access 2007**

- changes from Access 2003, 2
- as a front end to other databases, 132
- three-tiered applications and, 15

**Access 2007 VBA**

- changes from Access 2003, 2–3
- programming concepts, 3–25
- uses of, 1

**Access database files. See ACCDB files**

**Access databases**

- attaching SharePoint lists to, 231
- changes to existing prior to migration, 277
- connecting to, 132
- importing a SQL Server database, 203–204
- importing data from, 201
- separating into two database files, 319–322
- upsizing, 277–284

**Access designer, 271**

**Access field data types. See field data types**

**Access forms, generating prototype screens, 336–340**

**Access macros, VBA instead of or with, 1**

**Access Options, Advanced, 316–317**

**Access Project file, 278, 281**

**Access Projects**

- based on SQL Server databases, 260, 262–263
- creating, 259–263
- described, 253
- displaying new, 278, 284
- as a front end to SQL Server, 253–254
- migrating existing Access databases to, 277–284
- reusing code across multiple, 308
- storage locations for objects seen from, 254
- working with SQL Server objects, 264–277

**Access runtime, 330**

**Access tables**

- checking dates in, 277
- exporting contents to a spreadsheet, 205, 206

**ActionSuccess event, 116, 120–121**

**Activate event, 247**

**active plans, retrieving, 472**

**ActiveConnection property, 143, 147, 151, 155**

**ActiveX controls, 305–308**

**ActiveX Data Objects. See ADO**

**Activity Diagrams, 7–8, 9, 10**

**actor in a Use Case Diagram, 5**

**Add Comment button, 337**

**Add Current Contact To Project button, 339, 340**

**Add method, 112, 113**

**add mode, 442–443, 449–450**

**Add mode flag, 169, 176**

**Add New navigation button, 137**

**Add Table dialog box, 275**

**Add Watch dialog box, 69, 71**

# AddEmptyProjectRecord procedure

---

AddEmptyProjectRecord **procedure**, 419, 433

**Add-in Manager**, 313

**add-ins**, 312–313

**Add-Ins tab**, 289

AddItem **method**, 189, 190–191, 192–193

AdditionalData **parameter**, 209

**Add/Manage Contacts button**, 453, 455

AddNew **method**, 141–142

AddParameters **procedure**, 497–499, 517

AddRecord **procedure**

calling from OnClick, 144, 152

indicating Add mode, 166, 172

adLockBatchOptimistic **value for the LockType property**, 134, 317

adLockOptimistic **value for the LockType property**, 134, 317

adLockPessimistic **value for the LockType property**, 134, 317

adLockReadOnly **value for the LockType property**, 134, 317

**ADO (ActiveX Data Objects)**

adding a reference to, 358

connecting to the ProjectTrackerDb, 344

described, 129–130

storing database errors, 75

treating data sources as virtual tables, 163

versions of, 130–131

**ADO Command object**, 269

**ADO Connection object**, 313

**ADO object library**, 130

**ADO object model**, 130–131

**ADO Parameter object**, 269

**ADO recordsets**

based on SQL statements, 272

binding a form to, 138

specifying as a report's data source, 236–237

**Adobe Acrobat**, launching, 457

**ADODB library**. See **ADO object library**

**ADO.NET**, 130

adOpenDynamic **value for the CursorType property**, 133

adOpenForwardOnly **value for the CursorType property**, 133

adOpenKeyset **value for the CursorType property**, 133

adOpenStatic **value for the CursorType property**, 133

**ADOX library**, 177–178

**ADP file**, opening, 278

**ADPs**. See **Access Projects**

**All Object Types tab**, 326

**All Records option**, 317

AllForms **collection**, 97

AllPrinters **collection**, 247

AllReports **collection**, 238

**Analyze Performance**, 326

**And statements in a Where clause**, 161

**announcements in SharePoint**, 221, 229–232

**ANSI SQL**, 159

**application(s)**

determining the purpose of, 3

distributing, 329–331

elements of, 11–12

exposing as a web service, 214

finishing, 315–332

identifying the objectives of, 3

maintaining, 331

modeling the design of, 4–11

optimizing, 322–327

securing, 327–329

writing to handle multiple users, 315–322

**application architecture**, 11–18

**application design**, 4–11

**application logic**, 104

**Application object**, 94

**architecture for applications**, 11–18

**arrays**, 51–53

**arrows in an Activity Diagram**, 8, 9, 10

**AS predicate**, 160

**ascending order**, 162

**Attachment data type**, 50, 222

**attachments**

database objects as, 210–211

procedure for previewing, 385, 401

procedures for saving, 439

**Auto Order option**, 198

**AutoExec macro**

creating, 46–47, 185

opening a switchboard form, 187

specifying form to open, 182

**automation**, 294–305

**automation client**, 294

**automation objects**, 294–295

**automation server**, 294

**AutoNumber (Long Integer) field data type**, 51

**AutoStart parameter**, 249

**B**

**backup copies of databases, 331**  
**batch optimistic recordset, 151**  
BeforeUpdate **event, 196**  
BeforeUpdate **procedure, 40, 41**  
BeginTrans **method, 313, 314**  
**Blank Database section, 259**  
bln (**Boolean**) **prefix, 55**  
BOF **property of a recordset, 141**  
Bookmark **property, 159**  
**bookmarks, 159**  
**Boolean data type, 50**  
**bound forms, 139, 316–317**  
**break mode, 70, 71**  
**breakpoints**  
    adding to code, 24  
    adding to frmSearch, 512, 513  
    stepping through code, 66–68  
**builder add-ins, 313**  
BuildSQLDeleteProjectsComments **function, 397–398, 409**  
BuildSQLInsertContacts **procedure, 391–392, 406**  
BuildSQLInsertProjectComments **function, 398–399, 409**  
BuildSQLSelectFrom **function, 494**  
BuildSQLSelectFrom **procedure, 513**  
BuildSQLWhere **function, 494–495, 514**  
BuildSQLWhere **procedure, 513–514**  
**built-in constants, 54**  
**built-in functions, 45–46**  
**business logic, 14**  
**business logic layer, 17, 104**  
**business rules, 12**  
**buttons, adding to a command bar, 290**  
byt (**Byte**) **prefix, 55**  
**Byte data type, 50**

**C**

**Calculated data type, 222**  
CalculateSalesTax **function, 43–45**  
**Calendar control, 306, 307**  
**call stack, 70**  
Cancel **parameter, 90**  
Caption **properties for command buttons, 115, 118**  
Caption **property**  
    changing at runtime, 81  
    changing for a label, 115, 118

    changing for command buttons, 142  
    of a command button, 86  
    of a command button control, 95  
    of a list box, 95  
    setting for a form, 80  
Catalog **object, 177**  
**CE devices, managing a SQL Server database, 255**  
**cells, assigning values to, 298**  
Cells **property, 298**  
**child class, 122**  
**Choice data types, 222**  
**Choose Builder screen, 21, 22, 38**  
**class, 115, 121–122**  
**class diagrams, 102–105**  
**class hierarchies, 122**  
**class modules**  
    advanced techniques, 121–123  
    building for objects in Project Tracker, 357  
    creating custom objects, 101  
    creating objects, 101–123  
    for Customer Service application, 481–488  
    defined, 101  
    described, 29  
    inserting new, 102  
    public variable declarations, 105  
    writing code in, 37–42  
**Classes list, limiting, 91, 92**  
ClearContactsControls **procedure, 447**  
ClearControlsOnForm **procedure, 146–147, 154**  
ClearObject **procedure**  
    adding to clsCustomer class module, 487–488  
    adding to clsProjects class module, 365, 370, 377–378, 382–383  
ClearProjectControls **procedure, 420–421, 435**  
Click **event**  
    adding code for, 120  
    adding to a class, 116  
    for a command button, 90  
Click **event procedures**  
    adding to command buttons, 413–414, 429–431  
    adding to frmProjects, 411–413, 428  
    extracting data, 91  
    managing comments, 416–417, 431  
    managing contacts, 415–416, 430–431  
    running a search, 501–503  
**client cursors, 134**  
**client machine, 13**  
**client-server applications, 13–14**

## **client-server architecture, example of, 253, 254**

CloseDbConnection **procedure, 390, 405, 493**

cls **prefix, 56**

clsCar **class**

methods for, 112–113

properties for, 107–111

clsContacts **class module, 372–383**

clsCustomer **class module, 482–488**

clsProjects **class module, 358–372**

cmdAddAttachment\_Click **event, 417–418, 431–432**

cmdAddNew\_Click **event, 442–443, 449–450**

cmdAddNew\_Click **event procedure, 411–412, 428**

cmdAddToProject\_Click **event, 442, 449**

cmdDeleteContacts\_Click **procedure, 415, 430**

cmdEmailContact\_Click **procedure, 415–416, 431**

cmdFileBrowse\_Click **event, 417–418, 432**

cmdManageContacts\_Click **procedure, 415, 430**

cmdMoveFirst\_Click **event, 444–445, 450**

cmdMoveFirst\_Click **procedure, 413, 429–430**

cmdOpenFile\_Click **event, 418, 432–433**

cmdRun\_Click **event procedure, 40**

cmdRun\_Click **procedure, 37, 39**

cmdSave\_Click **event procedure, 412, 428**

cmdViewContact\_Click **procedure, 416, 431**

## **code**

debugging, 66–71

documenting, 63–64

encapsulating into objects, 101

running and debugging, 24–25

separating into logical tiers, 18, 103

stepping through, 66–68, 512

## **Code Builder**

choosing, 84

creating a procedure for an object, 37–40

selecting, 95

writing code for an event, 86–87

## **Code Builder option, 21, 22**

## **code execution, 69**

## **code library, 308–312**

## **code modules. See modules**

## **Code window**

adding code to, 21, 23

creating additional procedures, 40–42

in Visual Basic Editor, 19, 20

CodePage **parameter, 206**

**collections, 80, 94–100, 130**

## **column-level security, 271**

**columns, selecting in a table, 160**

**combo boxes, 188–193**

**comma-delimited records, 207–208**

## **command(s)**

displaying options for, 21, 23

typing in the Code window, 21, 23

writing within procedures, 56

**command bar, creating a custom, 288–291**

**command button control, 94, 95**

**Command button, properties of, 86**

## **command buttons**

adding to a form, 142, 150

Click event for, 90

creating a switchboard form, 186

drawing on a form, 29, 30

**command completion feature, 112**

## **Command object**

assigning to an open connection, 132

declaring and instantiating, 132

declaring a new, 167, 173

setting to the existing connection, 167, 173

CommandBar **object, 100**

CommandBars **collection, 288**

CommandBars **property, 288**

CommandText **property**

assigning to a SQL statement, 167, 173

assigning to the SQL statement, 169, 175

of the Command object, 518

**comments, saving to the database, 439**

**Comments/Tasks tab, 336, 337**

CommitTrans **method, 314**

**compact Access databases, 331**

**comparison operators in a Where clause, 161**

**compile errors, 64, 65**

**compiling applications, 330**

**conditional If statements, 59**

**connected recordsets, 132**

## **Connection object**

adding code for, 150

in the ADO object model, 131

declaring, 131, 138

instantiating, 131

opening, 131

**connection string, 132, 138**

**connection string variable, 150**

**connections, opening, 138, 167, 173. See also database connections**

**Connectors in an Activity Diagram, 7**

Const **statement, 53**

**constants**

built-in, 54

creating in VBA, 53–54

declaring and using, 53

defined, 53

limiting a property value to, 122

scope and lifetime of, 54–55

**consuming program for web services, 214**

Contact **object, class for, 357**

**contacts**

adding to project records, 442, 449

displaying, 452–453

procedures for saving to the database, 439

sending a letter to, 298–302

as a SharePoint list, 221

Contacts **class, 372–383**

**Contacts form**

building for Project Tracker, 440–451

to connect and display data from a recordset, 134–140

opening, 453, 455

**Contacts list, 227–229**

**Contacts prototype form, 341**

**Contacts screen**

designing, 335–336

with one filtered record, 453, 455

prototype for, 339–340

**Contacts tab, 338**

ControlExcel **procedure, 296, 297**

ControlOutlook **procedure, 303–304**

**controls**

adding to forms, 344–357

adding to reports, 238–241

aligning on forms, 197

creating for reports, 245

populating with the current record, 167, 174

working with, 187–195

**Controls ribbon, 306**

ControlSource **property, 139**

ControlWord **procedure, 299–300**

**copies, number for printing, 247**

**Create Back-End Database dialog box, 320, 321**

**Create New Database option, 277, 279**

CREATE PROCEDURE **statement, 470**

**Create ribbon**

Create Module option, 30, 32

selecting Macro, 46, 47

selecting Module, 19, 20

CreateCommandBar **function, 289**

CreateCommandBar **procedure, 290**

CreateComplaintsReport **procedure, 242–244**

CreateObject **method, 295**

CreateParameter **method, 517**

CreateReport **method, 238, 245**

CreateReportControl **method, 238–241, 245**

**cross-reference tables**

deleting records from, 397–398, 409

inserting records into, 398–399, 409

cur **(Currency) prefix, 55**

**Currency data type, 43, 50, 221**

CurrentProject **object, 97**

**cursor, type of, 325**

CursorLocation **property, 133–134**

CursorType **property, 133, 143, 151**

**custom button, adding to a command bar, 291**

**custom class modules, 357, 482**

**custom event, 120–121**

**custom groupings, 2**

**custom lists in SharePoint, 221**

**custom methods, 125–126**

**custom objects, 102**

**custom properties, 124–125**

**custom ribbons, 2**

**custom toolbar, 247**

Customer **class, 482**

**customer history record, 471–472**

**customer records**

code for adding, 506–507

code for navigating, 508–509

code for populating, 509–511

updating an existing, 473–474

**Customer Search form, 500**

**Customer Search screen, prototype for, 464**

**Customer Service application**

building the class module for objects, 481–488

building the database and database objects, 465–475

connecting the user interface to the code, 499–512

design specifications, 461–465

standard modules, 488–499

touring the finished, 512–518

user interface, 475–480

**customers, retrieving information on, 473**

## D

### DAO (Data Access Objects), 129

#### data

- deleting using SQL statements, 165–166
- exporting from Access to other formats, 248
- exporting to a spreadsheet, 205, 206
- exporting using `OutputTo`, 211
- importing from a text file, 206–208
- importing from the Northwind database, 202–203
- retrieving using SQL `Select` statements, 160–164
- updating using SQL statements, 165
- using from SharePoint lists, 222–232
- using from web services, 214–220

#### data access

- improving, 324–325
- introduction to, 129–130

#### data access layer, 17, 104, 105

#### Data Access Objects (DAO), 129

#### data elements, determining, 11

#### data entry controls, emptying, 146–147, 154

#### data error, correcting up front, 196

#### data objects, manipulating, 177–178

#### data sources

- linking to and importing from, 200
- retrieving a set of records from, 132
- specifying for a connection, 132

#### data tables, 318

#### data types

- Access compared to SharePoint, 221
- available in VBA, 49–50
- changing from Access to equivalent SQL Server, 278, 282
- user-defined, 52–53

#### data validation, adding, 196–197

#### database(s)

- building for the Customer Service application, 465–475
- building for the Project Tracker application, 340–343
- compacting and repairing, 331
- connecting to, 132
- creating, 29–31
- default format for, 29
- encrypting, 329
- making backup copies of, 331
- updating from a disconnected recordset, 147, 155

#### database access code, isolating, 18

#### database calls, layer for, 16–17

#### database connections. *See also* connections, opening

- closing and releasing from memory, 143, 151
- maintaining open, 139–140
- minimizing open, 324

#### database element, 12

#### database layer

- in n-tier architecture, 17
- in three-tier architecture, 15, 16

#### database location, making configurable, 340

#### database objects

- attaching to an e-mail, 210
- for the Customer Service application, 465–475

#### Database parameter for the `TransferSQLDatabase` method, 204

#### database password, 327–328

#### database server, 14

#### Database Splitter, 319

#### Database Splitter Wizard, 319–322

#### database tables

- retrieving data from, 163
- separating from the user interface, 319–322

#### Database Tools ribbon, 18, 19

#### database transfers, 201

#### database updates, 517

#### database window, replaced by the navigation pane, 2

#### DatabaseName parameter, 201

#### DatabaseType parameter, 201

#### DataSource parameter, 209

#### DataSource property, 236

#### DataTarget parameter, 209

#### Date data type, 50

#### Date of Birth text box, 87, 89

#### Date or Time data type, 221

#### dates, checking, 277

#### dbl (Double) prefix, 55

#### Deactivate event, 247

#### Debug object, 82

#### debugging

- code, 66–71
- errors, 64–71

#### Debug.Print method, 82

#### Debug.Print statement, 68

#### decision making in code, 56–61

#### Decision Points in Activity Diagrams, 8

#### Declarations/Procedure navigation box, 27, 28

#### default error messages, 72

#### Default File Format option, 29

- default record locking, 317**
  - Delete **command, 167, 173**
  - Delete Comment button, 337**
  - Delete **method**
    - adding to the clsCar class, 112–113
    - of the Recordset object, 142
  - delete operation, prompting the user to proceed, 388, 403**
  - Delete **procedure, 365–366, 370–371, 378, 382–383**
  - Delete Selected Contact button, 338**
  - Delete **statements, 165–166**
  - DeleteRecord **procedure, 144, 152, 166–167, 172**
  - delimited files, 208**
  - demo CDs, creating, 256**
  - descending order, 162**
  - Description **property of the Err object, 74**
  - design guidelines, 322–323**
  - design phase of SDLC, 4–18**
  - Design ribbon**
    - adding controls to a form, 187, 188
    - selecting and drawing controls, 115, 117
    - SQL View, 468, 469
    - ToolBox displayed on, 86
  - design specifications**
    - for the Customer Service application, 461–465
    - for the Project Tracker application, 334–340
  - Design tab, enabling icons, 272**
  - design time settings, 81**
  - Design View**
    - accessing in hidden mode, 236
    - retrieving data from multiple tables, 271, 272
    - switching to from Layout View, 29
  - designer window for a stored procedure, 267, 268**
  - Desktop Edition. See SQL Server 2005 Express Edition**
  - Destination **parameter, 201**
  - detail section of a report, 245**
  - developer extensions, 330**
  - development methodology, 3**
  - development phase of SDLC, 18–23**
  - diagramming techniques, 5**
  - Dim **statement**
    - declaring a class, 115
    - declaring variables, 51
  - disconnected recordsets**
    - building unbound forms, 142–156
    - described, 132
    - navigation procedures for, 145, 153
    - as recommended, 156
    - saving control values in, 147, 154–155
    - updating databases from, 147, 155
    - working with, 142
  - DisplayMessage **procedure, 291**
  - DisplaySystemInfo **procedure, 292, 293, 294**
  - DISTINCT **predicate, 160**
  - distribution options for applications, 329**
  - divide by 0 error, 74**
  - Dividing Lines **property, 142, 149**
  - DLLs (Dynamic Link Libraries)**
    - referencing external, 291–294
    - standard, 287–291
    - using, 287–294
  - Do...Loop **statement, 62–63**
  - Do...Until **loop, 62, 63**
  - Do...While **loop, 62**
  - Do...While **statement, 63**
  - doccmd **command, 21, 23**
  - DoCmd **object**
    - performing macro commands or menu options, 97–98
    - programmatically linking a list, 227–229
    - TransferDatabase method, 200–203
  - documenting code, 63–64**
  - Double data type, 50**
  - DSN, 230**
  - dt (**Date**) **prefix, 55**
  - dynamic arrays, 51, 52**
  - Dynamic Link Libraries. See DLLs**
- ## E
- early binding, 295, 324**
  - Edited Record option, 317**
  - elements**
    - of applications, 11–12
    - running on different computers, 15
  - e-mails**
    - generating empty, 415–416, 431
    - integrating into applications, 453, 454
    - sending from VBA code, 210–211
    - sending through Outlook from Access, 302–305
  - embedded macros, 2**
  - empty forms for a user interface, 344, 475–481**
  - empty record, adding to a recordset, 141–142**
  - Enable this Content option, 34**
  - Encode Database, 329**
  - Encoding **parameter**
    - for the ExportXML method, 209
    - for the OutputTo method, 249
  - Encrypt with Password option, 327**

**encrypting a database, 329**  
**end users, interviewing, 3–4**  
**Ending Points in Activity Diagrams, 8**  
**Enhanced Rich Text data type, 222**  
**enumerated types, 122–123**  
**EOF property of a recordset, 141**  
**error handler, 73–74**  
**error messages, 72, 77–78**  
**ErrorHandlerLibrary database, 308, 311**  
**error-handling code, writing, 71–78**  
**error-handling procedure, 308**  
**errors**  
    debugging, 64–71  
    raising in code, 75  
    types of, 64–66  
**Errors collection, 75**  
**event(s)**  
    calling functions from, 45  
    calling sub procedures from, 36  
    common to objects, 84  
    creating, 113–115  
    declaring and raising, 114  
    defined, 79  
    described, 84–86  
    example using, 86–91  
    initializing and terminating, 115  
    as a SharePoint list, 221  
**event handlers. See event procedures**  
**Event Procedure option, 85**  
**event procedures**  
    adding to a form, 143, 151  
    adding to handle recordset navigation, 144, 152  
    creating, 90  
    navigating through records, 444–445, 450  
**event sub procedures, 114–115**  
**Event tab**  
    from the cmdRun Properties window, 37  
    of the Properties dialog box, 84, 95  
**Excel**  
    controlling from Access, 295–298  
    creating a new instance of, 295  
**Execute method, 132, 269, 517**  
**execute-only applications, 331**  
**ExecuteSPRetrieveRS function, 495–496**  
**ExecuteSQLCommand procedure, 389–390, 404–405**  
**ExecuteStoredProcedure procedure, 497**  
**ExecuteStoreProcedure procedure, 517–518**

**Exit event, 86–87**  
**Exit event procedure, 90**  
**exporting**  
    data using OutputTo, 211  
    defined, 200  
**ExportXML method, 208–209**  
**Express Edition. See SQL Server 2005 Express Edition**  
**Expression Builder, 84**  
**eXtensible Markup Language. See XML**  
**external applications, automating, 294–295**  
**external data sources, 199–211**  
**External Data tab, 200, 222, 223**  
**external DLLs, 291–294**  
**external function calls, 456–458**  
**external functions**  
    calling, 293  
    declarations to, 400  
**external library, 293**  
**Extreme Programming, 3**

## F

**field data types, 50–51**  
**fields**  
    specifying for sorting, 157  
    viewing for a table, 134  
**file attachments, 456. See also attachments**  
**File Attachments tab, 339**  
**File Open button, 339**  
**file-level security, 327**  
**FileName parameter**  
    for the TransferSpreadsheet method, 205  
    for the TransferText method, 206  
**Filter property, 158–159**  
**FinalizeOrder procedure, 35, 36**  
**Find method, 157–158**  
**firing events. See raising events**  
**fixed-size array, 52**  
**FixNull function, 489**  
**For Each...Next loops, 61**  
**For Each loops, 99–100**  
**For...Next loops, 61–62**  
**For...Next statement, 62**  
**form(s)**  
    adding to a database, 29  
    binding to ADO recordsets, 138  
    creating the user interface, 18

getting to the Visual Basic Editor from, 21  
 hiding to be reopened later, 325  
 loading, 47, 48  
 navigation and flow of, 181–187  
 obtaining a list of all open, 94–96  
 obtaining currently active, 98  
 opening from a switchboard form, 187  
 polishing, 197–198  
 properties of, 80  
 saving, 30, 87

**form modules, 29**

Form **object, 94**

Form\_frmText **object, 40, 41**

Form\_Load **event**  
 adding code to, 116, 119  
 adding to a form, 143, 150  
 code in, 138  
 ensuring changes are saved, 148, 156  
 for frmCustomers form, 505–506  
 for the frmProjects form, 410, 427  
 initializing frmContacts form, 441, 448–449

Form\_Load **event procedure, 511–512**

Form\_Load **procedure, 135–136**

Form\_Unload **event, 448, 451**

Form\_Unload **procedure, 422–423, 437**

Forms **collection, 94**

**forward-only cursor, 133**

frm **(Form) prefix, 56**

frmContacts **form**  
 building for Project Tracker, 440–451  
 creating, 351, 354–356, 357  
 writing code for, 441–451

frmCustomers **form, 477–480, 504–512**

frmProjects **form**  
 creating, 344–353, 357  
 example of, 410, 411  
 writing code for, 410–440

frmSearch **form**  
 adding a breakpoint to, 512, 513  
 creating, 475–477  
 writing code for, 500–504

frmTest **form**  
 Code Builder from, 40  
 creating a new procedure for, 37–40  
 displaying objects associated with, 40, 41  
 opening in View mode, 38  
 selecting from the Form Name field, 47, 48

FROM **clause, 160–161**

fsub **(SubForm) prefix, 56**

FullName **property, 97**

Function **keyword, 42**

**function procedures, 42–44**

**functions**  
 behind methods, 82–83  
 built-in, 45–46  
 built-in SQL Server, 275–277  
 calling from events, 45  
 calling from other procedures, 44–45  
 creating, 43–44, 114–115  
 defined, 33  
 scope and lifetime of, 45

## G

**general comment section, 63–64**

**General Declarations section, 54**

GeneralErrorHandler **procedure**  
 adding, 75–76  
 adding to modBusinessLogic, 492  
 adding to the modBusinessLogic module, 389, 404  
 calling, 77  
 placing into a new database, 311

**generic error handler, 75–78**

**Get External Data - SharePoint Site Wizard, 222–226**

GetFileNameBrowse **function, 385–386, 401–402**

GetLookupDisplayValues **parameter, 226**

GetOpenFileName **external function, 384, 400**

GetPlaceFacts **method, 218–220**

GetSQL **function, 512**

GetSystemInfo **external library, 293**

**global scope of the Err object, 74**

## H

HandleError **section, 74, 77**

HasFieldNames **parameter**  
 for TransferSpreadsheet method, 205  
 for TransferText method, 206

**hidden mode, 236**

**hourglass, 325**

**HTML pages, creating static, 249–251**

HTMLTableName **parameter, 206**

**Hyperlink data type, 50, 222**

## I

### icon

- for events, 92, 93
- for a method, 83
- for a property, 81, 82

### ID data type, 221

### ID of a SharePoint list, 227

### If statement, 34

#### If...Then statements, 56–59

#### If...Then...Else statement

- as faster than IIF statements, 324
- syntax for, 57

#### If...Then...ElseIf statement, 57

### IIf function, 59

### ImageTarget parameter, 209

### Immediate window

- calling TestError2, 76
- displaying the currently active form, 98
- running a new function in, 43
- running a procedure from, 34
- running the new TestIfStatement procedure, 58
- running the TestCaseStatement procedure, 60
- running the TestError procedure, 73
- running the TestLoop procedure, 62, 67, 68, 69
- seeing report names in, 97
- using, 68
- in the Visual Basic Editor, 24, 25

### implementation phase of SDLC, 25

### implements keyword, 123

### importing

- defined, 200
- SharePoint lists, 222–226
- situations for, 200

### ImportOptions parameter, 208

### ImportXML method, 208

### indexes, 325

### infinite loop, 63

### Initialize event, 115

### in-line comments, 63, 64

### inner join, 161

### Insert ActiveX Control window, 306

### Insert statement, 164–165

### int (Integer) prefix, 55

### intCounter variable, 67

### Integer data type, 50

### IntelliSense, 112

### interactive forms, 181–198

### interface inheritance, 123

### interviewing end users, 3–4

### intrinsic constants. See built-in constants

### issues in SharePoint, 221

### iterative process, testing as, 24

## J

### Jet SQL, 159

### join syntax, 161

## K

### KeyPress event, 85–86

## L

### label

- changing properties of, 86
- selecting for a list box, 95

### late binding, 295, 324

### left join, 161

### letters, creating Word from Access, 298–302

### libraries

- available for use in Access applications, 100
- based on specific Access databases, 308–312

### Libraries list, 93

### life cycle

- of a procedure, 45
- of variables, 54

### LIKE keyword, 515

### LimitToList property, 193

### LinkChildFields property, 194

### linking

- to SharePoint lists within Access, 225–232
- situations for, 200

### LinkMasterFields property, 194

### links in SharePoint, 221

### list box control

- dragging onto a new form, 94, 95
- populating, 422, 436–437

### list boxes, versus combo boxes, 188–193

### ListID parameter, 226

### lists, choices in, 188

### lng (Long) prefix, 55

### LoadRecords procedure, 423, 437–438

### local string variable, declaring, 219

**local variables**

- adding for `clsProjects`, 372–373, 379
- adding for Property Get and Property Set, 111
- declaring and assigning values to, 40
- declaring as Report objects, 97
- declaring for `clsProjects`, 359, 366–367
- in a procedure, 36

**Local window, 24, 25****Locals Window, 68–69****Lock Project For Viewing option, 328****LockType property**

- of the `Recordset` object, 134, 317
- setting, 143, 151

**logic errors, 66****logical tiers, 18****Login parameter, 204****Long data type, 50****loops, declaring, 61–63****M****Macro Builder, 84****macros**

- embedded in Access 2007, 2
- examples of, 46–48, 49
- instead of procedures, 46
- naming, 47, 49
- versus procedures, 46–48, 49
- responding to events, 84
- saving, 47

**“the Macros in this project are disabled” error message, 34****mcr (Macro) prefix, 56****MDB file format, 2****Me keyword, 94, 125, 126****Memo field data type, 51****memo fields, 2****menu add-ins, 312****menus, importing or linking data, 200****methods**

- adding to a class module, 111
- adding to a form, 126
- calling from within `Property Let`, 106
- in a Class Diagram, 103
- creating, 111–113
- defined, 79
- described, 82–83

example using, 86–91

implementing with interface inheritance, 123

**Microsoft Access. See Access 2007****Microsoft Content Management Server 2002, 221****Microsoft Excel. See Excel****Microsoft IntelliSense, 112****Microsoft Office 12.0 Object Library, 288–291****Microsoft Office 2003 Web Services Toolkit, 214****Microsoft Outlook. See Outlook****Microsoft SharePoint Portal Server 2003, 221****Microsoft SQL Server. See SQL Server****Microsoft SQL Server Database Wizard, 262–263, 466****Microsoft Visio. See Visio****Microsoft Visual Basic, 1****Microsoft Word. See Word****middle (business) layer in three-tier architecture, 15, 16****mod (Module) prefix, 56**

`modBusinessLogic` module, 384–389, 400–404, 489–499

`modDatabaseLogic` module, 384, 389–399, 400, 404–408

**modeling, 4**

`modError` module, 308, 311

**Modify Columns and Settings link, 227****modules**

- authorizing to run, 34
- creating, 27–28, 29
- defined, 27
- inserting into databases, 242
- types of, 29, 101

**Modules tab, 29****MouseDown event, 92, 93****MouseWheel event, 92, 93****Move Data ribbon, 319**

`MoveToFirstRecord` procedure, adding to `modBusinessLogic`, 490

`MoveToLastRecord` procedure, adding to `modBusinessLogic`, 490–491

`MoveToNextRecord` procedure, adding to `modBusinessLogic`, 491–492

`MoveToPreviousRecord` procedure, adding to `modBusinessLogic`, 491

**msgbox command, 21, 23****msgbox function, 195–196, 293****Multiple lines of text data type, 221****multiuser architecture considerations, 318–322****multiuser considerations for applications, 315–322**

## N

### Name **properties**

- changing for command buttons, 115, 118
- changing for controls, 30, 31
- setting for command buttons, 142

### Name **property**

- changing for a label, 115, 118
- changing for a text box, 86, 115, 118
- of a command button, 86, 95
- of a label, 86
- of a list box, 95
- modifying for text boxes, 135

### **naming conventions**

- for objects, 55–56
- prefixes used as, 55–56

### **navigating through a recordset, 141**

Navigation Buttons **property, 142, 149**

### **navigation pane, 2**

**navigation procedures, 145, 153**

### **No Locks option, 317**

### **nonvisual objects, 82**

**Northwind database, 202–203, 235**

**n-tier architecture. See three-tiered architecture**

**Number (Byte) field data type, 50**

**Number data type, 221**

**Number (Double) field data type, 50**

**Number (Integer) field data type, 51**

**Number (Long Integer) field data type, 51**

**Number (Single) field data type, 51**

## O

obj **(Object) prefix, 55**

### **object(s)**

- building class modules for Project Tracker, 357
- characteristics of, 80
- class module for in Customer Service, 481–488
- creating custom methods for existing, 125–126
- creating custom properties for existing, 124–125
- creating your own, 101–127
- defined, 79
- elements of, 79–91
- instantiating at runtime, 295
- referring to, 93–94
- seeing methods available for, 83
- standardized way of naming, 55

- using, 94–100
- using class modules to create, 101–123
- viewing using the Object Browser, 91–93

### **Object Browser**

- opening, 91, 92
- viewing objects using, 91–93

**Object data type, 50**

**Object navigation box, 27, 28**

### **object variables**

- declaring and using, 53
- declaring for Excel, 294
- defined, 53

ObjectFormat **parameter, 210**

ObjectName **parameter**

- for OutputTo, 249
- for SendObject, 210

**object-oriented programming. See OOP**

**object-oriented programming environment, 79–100**

ObjectType **parameter**

- for the ExportXML method, 209
- for the OutputTo method, 249
- for SendObject, 210
- for the TransferDatabase method, 201

objProjects **object, populating, 412, 428–429**

**ODBC database, importing data from, 201**

**ODBC data source name, 201**

**Office and Home Application Download Category, 214**

**Office Button, 79**

**Office SharePoint Server 2007, 221**

On Click **event, 37, 38, 87**

On Click **event procedure, 95**

On Error **statement, 72, 77**

On Exit **event, 86–87**

On Load **event, 116**

**1&1 Internet, 222**

**OOP (object-oriented programming), 79**

**Open Databases Using Record-Level Locking option, 317**

**Open Exclusive setting, 323**

Open **method**

- of the Connection object, 131
- populating a recordset, 133

OpenDbConnection **procedure**

- adding to modDatabaseLogic module, 492–493
- adding to open database connections, 390, 405

OpenFileAttachment **procedure, 385, 401**

**OpenForm action, 47, 48**

**optimistic locking, 317, 318**

Option Explicit **statement, 31, 32**

Options **parameter of the Recordset object, 134**

**OR, in an Activity Diagram, 8, 9, 10**

Order By **clause, 161–162**

OtherFlags **parameter, 209**

**Outlook, 302–305**

**Outlook Calendar, 163–164**

OutputFile **parameter for the OutputTo method, 249**

OutputFormat **parameter for the OutputTo method, 249**

OutputQuality **parameter for the OutputTo method, 249**

OutputTo **method, 211, 248–251**

OutputToHTML **procedure, 250**

## P

**Packaging Wizard, 330**

PadQuotes **function, 495**

**page break, inserting into a Word document, 302**

**Parallel Activities in Activity Diagrams, 8**

**parameters, 33**

**parent-child class relationships, 122**

**password, 328–329**

Password **parameter, 204**

**Password Required dialog, 328**

**perceived performance, 325–326**

**percent meter, 325–326**

**performance**

improving actual for applications, 322–325

improving perceived, 325–326

slowed by encryption, 329

**Performance Analyzer, 326–327**

**period (.), referring to objects, 94**

**pessimistic locking, 317**

**Phone field, 514–515**

**physical tier, 18**

Picture **property, 142**

PopulateComboBoxes **procedure, 421, 435–436**

PopulateContactsControls **procedure, 446–447, 450–451**

PopulateControlsOnForm **procedure, 146, 153–154**

PopulateListFromRecordset **procedure, 422, 436–437, 489–490**

PopulateProjectsControls **procedure, 419–420, 433–434**

PopulatePropertiesFromForm **procedure**

adding to clsCustomer class module, 487

adding to clsProjects class module, 364–365, 369–370, 377, 381

PopulatePropertiesFromRecordset **procedure, 363–364, 368–369, 376, 380–381, 486**

**populating combo boxes and list boxes, 190–193**

**predicates, restricting records returned, 160**

**prefixes, as naming conventions, 55–56**

**presentation (user interface) layer**

creating diagrams for, 104, 105

in n-tier architecture, 17

in three-tier architecture, 15, 16

PresentationTarget **parameter, 209**

Preserve **statement, 52**

**Preview button, 339**

**preview mode, 245, 246**

**primary key for a SQL Server table, 265**

**printer devices, displaying in the Debug window, 98–100**

**printer margins, setting, 247**

Printer **object, 98, 246–247**

Printers **collection, 98–100, 246–247**

**private declarations of procedures, 45**

**private variable, 54**

**procedures**

calling sub procedures from, 35–36

creating and calling, 33–46

declaring as private, 45

defined, 33

versus macros, 46–48, 49

for making updates to data, 507–508

making use of If . . . Then statements, 57–58

retrieving values for, 36

variables and constants in, 49

ProceedWithDelete **function, 388, 403**

**process Flow Diagrams, 7**

ProcessRecordset **function, 493–494**

ProcessRecordset **procedure, 396, 407–408**

ProcessUpdate **procedure, 396–397, 408, 496**

**program flow, controlling, 56–63**

**progress bar, 325–326**

Project **class, 358–372**

**Project Explorer, 357**

Project **object, class for, 357**

**project records, creating procedures for, 393–395, 406–407**

## **Project Tracker application**

- building class modules for objects, 357–383
- building the database, 340–343
- building the user interface, 344–357
- connecting the user interface to the code, 410–451
- Contacts class, 372–383
- design specifications, 334–340
- opening, 335
- Project class, 358–372
- standard modules, 383–409
- touring the finished, 452–459

## **Project Tracker screen**

- dividing into tabs, 336–339
- prototype, 342
- sample project record on, 452

## **Project window, 19, 20**

ProjectId field, **410, 427**

## **Projects form, 410–440**

ProjectTrackerDb database, **340–343**

ProjectTrackerUI database, **344**

ProjectType property, **97**

## **properties**

- adding to forms, 124–125
- in a Class Diagram, 103
- creating, 105–111
- creating custom, 124–125
- creating for a class, 105
- creating for the Car class, 107–111
- described, 80–82
- of the Err object, 74
- example using, 86–91
- with interface inheritance, 123
- listing for an object, 81
- of objects, 79
- shared by objects, 81

## **Properties dialog box**

- displaying for a form, 30
- Event tab, 84, 86, 116
- for a form, 80
- for a text box control, 86

## **Properties module, 19, 20**

## **Properties window, 30, 32**

Property Get procedure

- adding, 111
- creating properties for existing objects, 124
- syntax for, 106–107

- using, 106
- using SpecialFeaturesList enumeration, 122–123

Property Let procedure

- adding, 111
- creating properties for existing objects, 124
- general syntax for, 106
- using, 106
- using SpecialFeaturesList enumeration, 122–123

## **property procedures**

- adding to clsCustomer class module, 483–485
- adding to clsProjects class module, 359–361, 367, 373–375, 380

Property Set procedure, **106, 107, 124**

property value, limiting to specified constants, **122**

Protection tab, selecting, **328, 329**

## **prototype screens**

- for the Customer Service application, 464–465
- generating with Access forms, 336–340
- generating with Visio, 336, 337
- for managing cars, 102, 103

## **public functions, 111**

Public keyword, **45, 54**

## **public sub procedures**

- adding to a class, 111
- creating custom methods, 125–126

## **public variables**

- creating properties for existing objects, 124
- declaring, 54–55
- implementing properties for custom classes, 105
- using, 105

PublicEvent statement, **114**

purpose of an application, **3**

# Q

qry (Query) prefix, **56**

## **queries**

- errors in upsizing, 278, 283
- returning rows, 325

Queries category in the Navigation Pane, **267**

Queries objects, **278, 284**

Query Designer, **160**

query processor for SQL Server, **259**

**R****Raise method of the Err object, 74–75****RaiseEvent statement, 114****raising events, 113****Range parameter, 205****Rapid Application Design (RAD), 3****read-only locks, 317****read-only properties, 107****record count, 434****record locking**

on bound forms, 316–317

on unbound forms, 317–318

**record navigation methods, 141****Record Selectors property, 142, 149****Recordcount property, 140–141****records**

adding to a table, 242

counting in a recordset, 140–141

deleting current from the recordset, 142

deleting from cross-reference tables, 397–398, 409

event procedures for navigating, 444–445, 450

inserting into cross-reference tables, 398–399, 409

navigating to particular, 157–158

procedures for saving displayed, 425–426, 439

sorting for the unbound contacts form, 156–157

updating the current in a recordset, 142

**recordset navigation procedures, 386–388, 402****Recordset object**

adding code for, 142, 150

creating, 133–140

declaring a new, 138

instantiating a new, 138

navigation methods, 141

properties and methods of, 83, 156–159

retrieving a set of records from a data source, 132

setting properties of, 143, 151

**Recordset property, 138–139****recordsets**

adding, editing, and deleting records in, 141–156

counting records in, 140–141

declaring in frmProjects, 428

filtering to smaller subsets, 158–159

navigating through, 141

populating controls on a form, 143, 151

procedures for navigating through, 144, 152

reconnecting to underlying data sources, 147, 155

repopulating values in, 405

repopulating with the most current values, 169, 175

sorting, finding, and filtering records in, 156–159

storing in local memory, 143, 151

using SQL and ADO to populate, 162–164

**RecordSource property**

of CreateReport, 245

of a report, 236

specifying a SQL statement as, 160

**ReDim statement, 52****references**

adding to the ADOX library, 177

adding to web services, 215–218

**References dialog box, 288****RefreshContacts procedure, 426–427, 439–440****RefreshHistory procedure, 511****registering add-ins, 313****repairing Access databases, 331****replication, removed for ACCDB files, 2****report events, 247–248****Report object, 96–97****report properties, changing, 236****reports**

adding controls to, 238–241

creating programmatically, 237–248

designating printing options for, 246–247

manipulating from VBA code, 235–248

opening in preview mode, 245, 246

setting the data source for, 236–237

**Reports collection, 96–97****Requery method, 167, 173****RequeryRecordset procedure, 391, 405–406****Require Variable Declaration option, 31****Requirement Number for a Use Case, 5****Resume Next statement, 72****Resume statement, 72****Retrieve method, 112, 113****RetrieveAttachments function, 363, 368****RetrieveComments function, 362, 367–368****RetrieveContacts function, 362–363, 368, 375–376, 380****RetrieveCustomers function, 485–486****RetrieveProjects function, 361–367****reusable modules, writing, 322****ribbons, 2****rich client, 16****Rich Text data type, 222**

**rich text in memo fields, 2**

**right join, 161**

**roadmap for building a system, 4–5**

**RollbackTrans method, 313, 314**

**row-level security, 271**

**RowSourceType property, 95**

**rpt (Report) prefix, 56**

**rsub (Subreport) prefix, 56**

**RunSearch procedure, 512**

**runtime, 330**

**runtime errors, 64, 66, 74**

**/runtime switch, 330**

## S

**Save Changes button, 337**

**Save event, 116, 120**

**Save procedure**

adding to `clsCustomer` class module, 488

adding to `clsProjects` class module, 366, 371, 378–379, 383

**SaveAllRecords procedure, 144, 147, 152, 155**

**SaveCurrentRecord procedure, 147, 154–155, 168, 174**

**scaling up a database server, 14**

**SchemaTarget parameter, 209**

**scope**

of an Activity Diagram, 7

defined, 45

global, 74

of a procedure, 45

of a variable, 54

**Screen object, 98**

**screen prototypes, 8–11**

**SDLC (Systems Development Life Cycle), 3**

**Search Inventory Screen**

Activity Diagram for, 8, 9

designing for Wrox Auto Sales, 5

mapping Use Cases to, 10–11

prototype for, 11

separate actions on, 5, 6

Use Case Diagram requirements, 8

**search results, clearing, 503–504**

**Search Results screen, 515, 516**

**securing applications, 327–329**

**security information, specifying, 278, 279**

**security model, 3**

**Select clause, 160**

**Select statements, 160–164**

**Select...Case statements, 59–61, 60–61**

**Send method, 305**

**SendObject method, 210–211**

**separation of tiers concept, 103–104**

**server cursors, 134**

**server machine, 13**

**Server parameter, 204**

**services, installed with SQL Server 2005 Express Edition, 257–259**

**Set statement, 115**

**SharePoint data types, 221**

**SharePoint DSN, 229, 230–231**

**SharePoint hosting, 222**

**SharePoint lists**

importing, 222–226

introduction to, 221–222

linking to, 225–232

types of, 221

using data from, 222–232

**SHELLEXECUTE, external DLL function call to, 456, 457**

**ShellExecute external function, 384, 400**

**Shift key, selecting multiple controls to align, 197**

**Single data type, 50**

**Single line of text data type, 221**

**single-tier applications, 12–13**

**SiteAddress parameter, 226**

**sng (Single) prefix, 55**

**sort order, 161–162**

**Sort property, 156–157**

**source argument, 134**

**Source parameter, 201**

**SourceObject property of a subform, 194**

**SourceTableName property, 229, 231**

**SpecialFeaturesList enumerated type, 122, 123**

**SpecificationName parameter, 206**

**spInsertCustomer stored procedure, 469–470**

**splash screen, 325**

**Split Database button, 320**

**spreadsheets, importing, linking, and exporting, 204–205**

**SpreadsheetType parameter, 205**

**spRetrievalAllCustomers stored procedure, 470, 471**

**spRetrieveCustomerHistory stored procedure, 471–472**

- spRetrievePlans **stored procedure, 472**
- spRetrieveSelectedCustomer **stored procedure, 473**
- spUpdateCustomer **stored procedure, 473–474, 517**
- SQL (Structured Query Language)**
  - introduction to, 159–164
  - syntax of, 159–160
- SQL code**
  - adding, 470
  - for a stored procedure, 268
- SQL Insert statements, 164–165**
- SQL queries, 324**
- SQL Server**
  - Access as a front end to, 132
  - Access Projects as a front end to, 253–254
  - built-in functions, 275–277
  - setting up, 254–259
  - stored procedures, 267–271
  - versions of, 255
  - views, 271–274
- SQL Server Browser Service, 259**
- SQL Server Configuration Manager, 257–258**
- SQL Server database**
  - linking to, 253
  - modifying strConnection string to point to, 492–493
  - specifying, 466
  - specifying details necessary to connect to, 260, 261
  - transferring a complete, 203–204
  - using a new, 262–263
  - using an existing, 260–262
- SQL Server Mobile Edition, 255**
- SQL Server objects, 264–277**
- SQL Server Service, 259**
- SQL Server storage engine, 259**
- SQL Server tables, 264–267**
- SQL Server 2005 Developer Edition, 255**
- SQL Server 2005 Enterprise Edition, 255**
- SQL Server 2005 Evaluation Edition, 255**
- SQL Server 2005 Express Edition, 255–259**
- SQL Server 2005 Standard Edition, 255**
- SQL Server 2005 Workgroup Edition, 255**
- SQL statements**
  - adding procedures for creating, 391–393, 406–408
  - with and without ADO, 160
  - building dynamic, 512
  - deleting data using, 165–166
  - dialects of, 159
  - inserting current values on a form, 168, 174–175
  - modifying the unbound contacts form, 166–176
  - as a report's RecordSource property, 241
  - running against a database, 167, 173
  - setting changes and removing passwords, 327–328
  - as the source for the Open method, 162
  - specifying the DataSource of a report, 236
  - for stored procedures, 269
  - updating an existing record, 168–169, 175
  - updating data using, 165
  - updating tables, 393–395, 406–407
  - viewing in Query Designer, 160
- SQL View on the Design ribbon, 468, 469**
- standalone application, 13**
- standard DLLs, 287–291**
- standard modules**
  - building for Project Tracker, 383–409
  - for the Customer Service application, 488–499
  - defined, 101
  - described, 29
- Starting Point for an Activity Diagram, 7**
- static arrays, 51**
- static HTML pages, 249–251**
- Static keyword, 45, 54**
- Step Into option, 66**
- Step Over option, 66**
- storage engine for SQL Server, 259**
- stored procedures**
  - adding to the Customer Service application, 467
  - calling, 269
  - creating for SQL Server, 269–271
  - described for SQL Server, 267
  - running, 269, 271
- StoreLogin parameter, 201**
- str (String) prefix, 55**
- String data type, 60**
- string variable, 60**
- structure variable, declaring, 219**
- Structured Query Language. See SQL**
- StructureOnly parameter, 201**
- sub procedures**
  - behind methods, 82–83
  - calling from events, 36
  - calling from other procedures, 35–36
  - creating and calling new, 33–35
  - defined, 33
  - scope and lifetime of, 45

# Subform control

---

Subform control, 194

## subforms

- described, 194
- example of, 194, 195
- minimizing the use of, 322
- tying to a parent form, 194

## subqueries, 162

## switchboard forms, 182–187

## syntax errors, 64, 65

## system information, from an external DLL, 292–294

SYSTEM\_INFO type declaration, 293

Systems Development Life Cycle. See SDLC

# T

## tab controls, 193–195, 434

## Tab Order screen, 197–198

## tab stops, order of, 197

## table(s)

- adding to the Customer Service application, 466–467
- adding to databases, 241–242
- combining data from two, 162
- creating in a database, 134
- creating with ADOX library, 177–178
- exporting to XML, 209
- inserting results from a Select statement, 165
- for Project Tracker, 341–343
- retrieving data from more than one, 161
- specifying to upsize, 278, 280

## table alias, 160

## table attributes, specifying to upsize, 278, 280

## table names, removing spaces in, 277

## table scans, reducing, 325

## table view, 140

## TableDef object

- creating, 229, 230
- programmatically linking a list, 229–232
- with a SharePoint list as its source, 229

## TableName parameter

- for the TransferSharePointList method, 226
- for the TransferSpreadsheet method, 205
- for the TransferText method, 206

## Tables category in Access Projects, 264

## tabs

- controlling the visibility of, 194
- for the Project Tracker screen, 336–339

Tag property, 124

tags in an XML file, 209

tasks in SharePoint, 221

tbl (Table) prefix, 56

tblComplaints table, exporting to HTML, 249–251

tblContacts table, mapping to fields on, 341, 343

tblCustomers table, adding to the Customer Service application, 466

tblCustomersPlanHistory table, adding to the Customer Service application, 467

tblPlans table, adding to the Customer Service application, 467

tblPlansStatus table, adding to the Customer Service application, 467

tblProjects table, adding, 341, 343

tblProjectsComments table, adding to Project Tracker, 342, 343

tblProjectsContacts table, adding to Project Tracker, 342, 343

tblProjectsFileAttachments table, 342, 343

TemplateFile parameter for the OutputTo method, 249

## templates, prebuilt, 2

Terminate event for a class, 115

TerraService web services, 215–216, 218–220

Test Connection button, 261

TestCaseStatement procedure, 60–61

TestError procedure, 73–74

TestError2 procedure, 76

TestIfStatement procedure, 57–58

testing phase of SDLC, 24–25

TestLibrary procedure, 311–312

TestLoop procedure, 61

TestMethod method, 111

TestPrinter procedure, 99

TestPrinter sub procedure, 99–100

TestTransferDatabase procedure, 202–203

TestWebService procedure, 218, 219

## text box controls

- displaying a selected date, 307
- setting to current values, 146, 153–154

## text boxes

- adding to a form, 135
- code displaying values in, 188
- drawing on a form, 29, 30

text box object variable, 53

Text field data type, 51

**text files, 205–208**  
**thin client, 16**  
**three-tiered architecture, 15–17**  
**tiers of an application architecture, 12**  
**toggle button, 438–439**  
 togShowAll\_Click **event, 424, 438–439**  
**ToolBox, 86, 88**  
**tooltip, 220**  
 TOP **predicate, 160**  
**transactions, 313–314**  
**transactions form, displaying, 194, 195**  
 TransferCopyData **parameter, 204**  
 TransferDatabase **method, 200–203**  
 TransferSharePointList **method, 225–229**  
 TransferSpreadsheet **method, 204–205, 206**  
 TransferSQLDatabase **method, 203–204**  
 TransferText **method, 205–208**  
 TransferType **parameter**  
     for the TransferDatabase method, 201  
     for the TransferSharePointList method, 226  
     for the TransferSpreadsheet method, 205  
     for the TransferText method, 206  
**Trust Center, accessing, 47**  
**T-SQL, 159**  
**two-tier applications. See client-server applications**  
 typ (User-Defined Type) **prefix, 55**  
**Type Mismatch error, 64, 66**  
 typTripInfo **user-defined type, 52–53**

## U

**unbound contacts form**  
     building using a disconnected recordset, 142–156  
     modifying to use SQL, 166–176  
     sorting records for, 156–157  
**unbound forms**  
     better than bound, 324  
     record locking on, 317–318  
**unhiding the switchboard form, 187**  
 Union **statement, 162**  
**unique index number of each array element, 51**  
**updatable, recordsets as, 134**  
**update conflicts, 318**  
**Update method**  
     adding the clsCar class, 112  
     calling, 120  
     committing a new empty record to the local  
         disconnected recordset, 144, 152

    modifying in clsCar, 114  
     of the Recordset object, 142  
**Update statements, 165**  
**Upsizing Wizard, 277–284**  
**URL, decoding, 227**  
**Use Case Diagrams**  
     for the Customer Service application, 462–463  
     for the Project Tracker application, 334, 335–336  
     using, 5–7  
**Use Case Numbers, 5**  
**Use Cases, 5, 8–11**  
 UseOA **parameter, 205**  
**user ID, 194**  
**user input, validating, 196–197**  
**user interaction, 195–197**  
**user interface**  
     of an application, 12  
     building for Project Tracker, 344–357  
     building for the Customer Service application, 475–480  
     connecting to code, 410–451, 499–512  
     determining properties and methods of an object,  
         102, 103  
     implementing a basic, 139  
     separating from the data tables, 318  
**user interface database, opening exclusively, 323**  
**user interface elements, mapping Use Cases to, 10–11**  
**user interface objects, 344**  
**user-defined data types, 50, 52–53**  
**user-defined functions**  
     described, 275  
     embedding within a basic SQL statement, 275  
**user-interface database, 322**  
**user-interface objects, 319**  
**user-level security, 2**  
**users, interacting with controls, 195**  
 UseTrustedConnection **parameter, 204**

## V

**validation checks, 106**  
 ValidationRule **property, 196**  
 ValidationText **property, 196**  
**values**  
     calling a procedure with hard-coded, 58  
     preserving after a procedure ends, 45  
     returning from function procedures, 44  
     using variables and constants to store, 49–56  
**var (Variant) prefix, 55**

# variables

---

## variables

- declaring and using, 51–53
- declaring in functions, 44
- defined, 33
- evaluating multiple times, 59
- making visible in any module, 54–55
- naming conventions for, 55–56
- scope and lifetime of, 54–55
- standardized way of naming, 55
- storing the number of loop iterations, 62
- types of, 49–51

## Variant data type, 50

## variants, minimizing the use of, 324

## VB (Visual Basic), 1

## VBA (Visual Basic for Applications), 1

## VBA code

- adding a password for, 328–329
- basics of writing and testing, 27–78
- handling data validation from, 196
- linking to another database, 319
- optimizing to improve performance, 323–324
- responding to events, 84
- sending an e-mail from, 210–211
- using to link, import, and export, 199–211
- working with controls, 187–195
- working with reports from, 235–248
- writing behind report events, 247
- writing for the business logic and database access logic, 18

## VBA data types, correspondence to field data types, 50–51

## VBA Help documentation, obtaining a list of available functions, 45

## VBA source code, securing separately, 328

## VBA version 6.0, 1

## VB.NET, 1

## VerifyCreditAvail procedure

- calling from another procedure, 35–36
- creating, 33–35

## View button, 273

## View Details for Selected Contact button, 453

## View mode, 139

## ViewID parameter for the TransferShare-PointList method, 226

## View/Manage Car Details Screen

- Activity Diagram for, 8, 10
- prototype for, 11, 12
- Use Case Diagram for, 5, 7

## View/Manage CustomerAccounts form, 504–505

## View/Manage Customer Accounts screen

- displaying a customer record on, 515, 516
- opening independently of the Customer Search screen, 517
- prototype for, 464–465

## View/Manage Customer Details screen, 462–463

## views

- advantages of, 271
- creating, 273–274

## Views tab of the Add Table dialog box, 275

## virtual tables, 163

## Visible property

- hiding and redisplaying forms, 325
- of the objExcel object, 295
- setting to False, 186
- setting to True, 291

## Visio

- creating Activity Diagrams, 8
- generating prototype screens, 336

## Visual Basic, 1

## Visual Basic Editor (VBE)

- for Access 2007, 2
- display of, 19, 20
- opening, 37, 38
- opening with a new empty procedure, 85
- running and debugging code in, 24–25
- sample module displayed, 27–28
- writing VBA code in, 18–23

## Visual Basic for Applications, 1

## Visual Basic .NET, 1

## visual objects, 82

## Visual Studio Tools for Microsoft Office, 2

## Visual Studio Tools for Office (VSTO), using ADO.NET, 130

# W

## Watch window, 69–70, 71

## Watches window, 24, 25

## Web browser interface. See thin client

## Web formats, exporting Access data to, 248–251

## web services

- adding references to, 215–218
- consuming, 214
- others available, 218
- using data from, 214–220

## Web Services References option, 215

**Web Services Toolkit, 214**

Where **clause**, 161

Where **keyword**, 514

WhereCondition **parameter**, 209

While...Wend **statement**, 63

**Windows integrated security, 466**

With...EndWith, 323–324

WithEvents **keyword**

    declaring an object variable, 114

    as part of a declaration, 119

**wizard add-ins, 312****Word**

    controlling from Access, 298–302

    displaying in Print Preview mode, 302

Word Application **object**, 301

workstations in client-server architecture, 14

Wrox Auto Sales Application, 4

**X**

XML (eXtensible Markup Language), 208

XML files, importing and exporting, 208–210

**Y**

Yes/No data type in SharePoint, 222

Yes/No field data type, 50

