Index

Note that you can download the PDF file for this book from **www.hentzenwerke.com** (see the section “How to download files” at the beginning of this book). The PDF is completely searchable and will provide additional keyword lookup capabilities not practical in an index.

A

aApplication, 49–50, 91
abstract classes, 33
abstract factory, 37
aBuilderForm, 265–267
ACLASS( ), 145–146
Activate event, 173
ACTIVEFORM command, 66–67
ActiveX control, 253
ADD operation, 96
AddError( ), 146–147
AddMethods( ), 261
AddNewChild( ), 191
AddNewParent( ), 191
AddProperty( ), 261
AddProps( ), 261
AddToFormsCollection( ), 181–182
AddObject( ), 255
aForm, 172–176
AfterLabelEdit event, 71
aLanguages( ), 254
Album Maintenance Screen, 190
Albums table, 190
Alias( ), 117
alias, of cursor, 118
AllowNav( ), 201–202
aMembers( ), 236
APP files, 288
App.Data.dbc, 46, 156–158
Append( ), 100
APPEND BLANK command, 119, 120
APPEND command, 120
application database, creating, 274
application development time, 12
application directory, 267
application environment, 47, 89
application files, 44–46
application folders, configuring, 43
application-specific procedures, 45
application startup, loading toolbars, 214–215, 217–218
Application Variable Maintenance form, 158
application variable server, 28
Apply( ), 185
App.Proc.prg, 45
AppVars module, 294
AppVarsID, 156
ASelObj( ), 259
aSession, 93
Assembled In App approach, 77
ASSERTS, 10
auto-incrementing field, 119
AutoCenter property, 171
AutoSize property, 62
AutoStyle class, 37
AVCXClasses( ), 275

B

BackColor property, 260
back-end database, 149
BackStyle property, 62, 68
base classes, 57–58, 253–256, 258, 259
base form, 172–176
BeforeOpenTables( ), 142, 143, 148–150
BINDEVENT command, 36
bldDefault, 265
BlindEvent( ), 205
BorderWidth property, 68
Box control, 203–205
bubbling, 238–239
buffering, 111–112, 124–125
BuildDE( ), 205–206
builders
    advantages, 10, 253
    builder forms, 267–268
    creating, 262–268
Builders.dbf, 262
business logic, 141, 169
business objects. *See also* data entry forms; MyDataEnvironment
    building on the fly, 205–206
    definition, 137
    external representation, 138–140
    internal representation, 137–138
    remote data, 148–150
    retrieving information from SysCodes table, 159–160
    sample application, 296–297

- SysCodes, 160
- three-tier architecture, 138
- whole view, 139–140

C

call stack, capturing, 237
Calvin, 44
CamelCase, 21–22
Cancel( ), 185
CanHandleError( ), 240
capturing
- call stack, 237
- display status, 238
- lines of code, 235–236
- object properties, 236–237

CASE statement, 51, 240, 248
centralized error handling, 227
class-based approach, 84–86
class-based forms, 181
class definitions, 22
class libraries
- base classes, 57–58
- combo boxes, 69–70
- command buttons, 63–67
- compared to frameworks, 4
- containers, 68–69
- controls, 72–73
- definition, 57
- features, 59–62
- forms, 75
- grids, 73–74
- loading into memory, 277–278
- page frame, 73
- running applications and, 288
- shapes, 71–72
- text boxes, 67–68
- treeview control, 70–71

Class Designer, 74, 257
classes
- abstract, 33
- collection, 99–108
- composite, 32
- creating, 274–276
- custom, 39
- flexibility, 31
- as form properties, 22
- prefixes, 20
- setting, 277–278
- on tap, 279–280
- values for, 294

ClassList array, 259
ClearErrorState( ), 260
Click( ), 228
Close( ), 175, 219
CloseAllForms( ), 52–53, 199
CodeBook Style approach, 76–77
collections
- classes, 99–108
- elements, 95–96
- in MyFrame, 98–108
- types of, 96–97
- using, 97–98

combo boxes, 69–70
command buttons, 63–67
CommandButton class, 74
CommandGroup class, 72–73
comments, coding, 17–18
common environment classes, 88
composite classes, 32
computer network capacities, 129
conditional formatting, 260–261
Config.fpw, 46
configurable frameworks, 11–12
ConfigureDisplay( ), 64
confirmation messages, 193
constants, predefined, 32, 46
containers, transparent, 68–69
containership, 228–229
Contains( ), 101–102, 105
ControlSource property, 69
conventions, framework development, 16–22
COPY FILE command, 273
Create Application Database checkbox, 274
CREATE CLASS command, 253, 274–275
CREATE PROJECT command, 278
CreateObject( ), 173, 181, 212, 254–255
CREATEOBJECT command, 55
CreateProject( ), 277
CreateToolbars( ), 218
cursor
- alias, 118
- finding, 148

Cursor class, 111
CursorAdapter
- advantages, 111
- capabilities, 132–133
- naming, 315
- retrieving data, 149

CursorFill( ), 132, 133–134
CursorGetProp( ), 124–125
CursorSetProp( ), 112, 125, 131, 132
custom classes, 39

D

data access page, 316
data binding, 71
data-driven (dynamic) solutions, 32
data entry forms. *See also* business objects; forms
  capabilities, 187
  controls, 192, 194
  data entry methods, 195–200
  data entry status, 191
  data entry styles, 190–191
  form methods, 193–194
  navigating, 200–205
  prompting users, 193
  styles of data entry, 187–191
  validation errors, 207–208
data environment class, 87–88
data files, 273
data objects, 127–129. *See also* records; tables
data sessions, 81–82. *See also* environments
data source, identifying, 314
Data Source Name (DSN), 130
data structure, sample application, 287
database design, 111
DataEnvironment Builder, 132, 149
DataSession property, 82, 143–144
DataSource property, 132, 317
DEClass property, 296
DEClassLibrary property, 296
defaults
  builder class, 265
  error handler, 224, 225
  error messages, 226
  labels, 294–295
  overwriting, 56
  security code values, 245
delegation (bridge), 37–38
Delete( )
  business objects, 140
  data entry forms, 195–196
  records, 121
  toolbars, 219
DELETE command, 121
DELETED( ), 121
design environment, running applications, 288
design requirements, 14
design review, 303–304, 311–312
design-time flexibility, 32
Destroy( ), 86, 119, 221
development environment, 89–90
DevProc.prg, 45
dialog boxes, 184–185
Direct approach, 75
Direct Subclass approach, 76
display status, capturing, 238
DISPLAY STATUS command, 238
DO FORM command, 173
DO WHILE loop, 96, 124–125, 251
docking status, toolbars, 163
DoClose( ), 174–176, 184, 198
DODEFAULT( ), 240
DoNew( ), 147–148
DoSave( ), 144–145
Dr. Dave, 44
DSN (Data Source Name), 130

E

Edit Always style, 187–188, 201–202
Edit First style, 187–188, 201–202
Edit Mode property, 220
editing
  forms, 197
  records, 187–190
Enabled property, 248
entity maintenance, 189–190
environment classes, 86–87, 91–93
environments, 81, 82–83
Error( ), 224
error handling
  capabilities, 223
  centralized, 227
  close to their source, 227
  containership, 228–229
  default error messages, 226
  developing an error handler, 229–230
  during development, 229
  error sequence, 225
  framework data model, 29
  methods, 230–233
  ObjectError( ), 238–240
  for objects, 224
  procedural errors, 224, 229
  recording error information, 233–235
  strategies, 226–227
  trapping errors, 227, 240
  TRY/CATCH and THROW, 225
  viewing error log, 240–241
error messages, in MyFrame, 141
ErrorLog.dbf, 234
ErrorNo property, 225
event binding, 36
EXCEPTION class, 254
exception object, 225

EXE files, 288
ExecScript( ), 260

F

factory method, 36–37
False value
  from Init( ), 52
  reversing, 51
Field Caption property, 111
field mapping, 294–295
fields
  prefixes, 21
  updating, 317
FileToStr( ), 277
FILTER TO command, 127
FINALLY block, 225
FitToScreen( ), 178
flexibility
  abstract factory, 37
  avoiding hard coding, 31–32
  changing object composition, 32–33
  classes, 31
  considerations, 11–12
  delegation (bridge), 37–38
  factory method, 36–37
  mediator and event binding, 35–36
  patterns, 33, 34
  proxy, 38–39
  styles, 34–35
  wrappers, 39
folder structure
  arranging project folders, 42
  configuring application folders, 43
  requirements, 41
FontBold property, 62
FOR EACH loop, 96
ForceDataFlash( ), 219
Form Controls toolbar, 211
Form Designer. *See also* business objects
  controls, 111
  SysCodes, 160
forms. *See also* data entry forms
  advantages, 169
  base form, 172–176
  builder forms, 267–268
  closing, 199–200
  controls, 245, 301
  dialog boxes, 184–185
  editing, 197
  hierarchy, 75
  location, 177–178
  locking, 193–194
  methods, 193–194
  modal forms, 182–185
  navigating, 200–205
  preferences, 179–181
  private data session with, 81–82
  resizing, 176, 178–179
  sample application, 294–296, 298
  setting security, 247–248
  splash screens, 169–172
  switching between, 211
  synchronizing, 218–220
  types of, 292
  VCX form support, 181
forms collection, 28
FOR...NEXT loop, 96
FoxPro designers, 13
FoxPro error dialog, 226
FramePems table, 259
FramePems.dbf, 256–257
FrameProc.prg, 45
framework architecture
  data model, 29
  functional view, 27–29
  strategies, 23
  tiers, 23–27
framework classes
  Assembled In App, 77
  CodeBook Style, 76–77
  Direct, 75
  Direct Subclass, 76
  Meta classes, 79–80
  Rebuild, 78
framework development. *See also* classes; methods; properties; Visual FoxPro
  application database, 274
  application files, 44–46
  definition, 6
  folder structure
    application folders, 42
    framework folders, 43–44
    new projects, 271–273
    project folders, 42
    requirements, 41
  main program, 276–278
  project manager, 44
  project planning
    conventions, 16–22
    implementation plan, 9, 13–16
    project file, 278
    steering document, 9–13
  sample application
    compiling and running, 303

creating, 289–292
customer business object, 296–297
customer form, 294–296, 298
customer table, 292–294
design review, 303–304
one-to-many form, 305–312
process, 287
remote data, 313–319
SampleApp, 288–289
security, 298–303
source code considerations, 288
validation rules, 297–298
Task Pane, 280–284
tools
adding and removing properties, 261
adding common behavior, 255–256
base classes, 253–255, 259
changing parent, 268–269
conditional formatting, 260–261
creating builders, 262–268
method code, 261–262
Timer class, 259–260
zVisualBaseClassConfigurator.prg, 256–259
framework procedures, 45
framework services
AppVars.dbf, 156–158
availability, 165–167
characteristics, 153
controls, 176–178
instantiating, 165–167
message services, 153–156
object locations, 162–164
system codes, 158–162
frameworks
advantages, 3–4
compared to class libraries, 4
development considerations, 5–8
elements, 4–5
Fulton, David, 44
functions
generic, 44–45
prefixes, 21

G

GenProc.prg, 45, 107, 118
GetAppVars( ), 157
GetCurrentItem( ), 102
GetCurrentRecord( ), 127–129
GetCursor( ), 148
GetErrorMessage( ), 147
GetFile( ), 275
GetFldState( ), 123–124
GetForm( ), 264, 265
GetInstanceCount( ), 213
GetInstancesCount( ), 106, 107
GetLoc( ), 163, 164
GetLongDesc( ), 159–160
GetMessager( ), 235–236
GetNextModified( ), 123, 124–125
GetObjectInfo( ), 236
GetObjectList( ), 107–108
GetObjectName( ), 163
GetPrimaryKey( ), 117, 118
GetReturnValues( ), 159–160
GetShortDesc( ), 159–160
Go( )
navigation tasks, 202–203, 204
primary key value, 200
retrieving data, 128–129, 133–134
GO BOTTOM command, 200
GO command, 200
goal statement, steering document, 9–10
GoBottom( ), 201, 203, 204–205
GoFirst( ), 103
GoNext( ), 103, 201, 203
GoPrevious( ), 201, 203
GoTop( ), 201, 203
grid classes, 74
grid column, 228
grids, 73–74
Griver, Yair Alan, 76
groups, adding, 299–300, 302

H

HandleError( ), 230–232
hard coding, avoiding, 31–32
hard confirmation, 193
HeaderClass property, 73–74
HeaderClassLibrary property, 73–74
highlighting, nodes, 71
HitTest( ), 71
hot keys, for command buttons, 67
Hsia, Calvin, 44

I

IF statement, 133–134
IN clause, 121
include files, with relative paths, 46
inheritance hierarchy, 161
Init( )
with aBuilderForm, 265
calling LoadApp( ), 51

calling SetLocations( ), 220
calling Unbind( ), 69
data entry methods, 195–196
Environment class hierarchy, 86
Meta Data approach, 78
with MySplash, 170
with SetSecurity( ), 246
INSERT command, 120
INSERT INTO command, 119
instantiating
framework services, 165–167
toolbars, 217
IsAuthorized( ), 244
IsBaseClass( ), 258, 259
IsDone( ), 103
IsOKToClose( ), 175–176
IsOKToModify( ), 260
IsTag( ), 74
IsType( ), 145–146
IsValid( ), 124–125, 146

J

JUSTPATH( ), 53
JustTable( ), 118

K

KEYBOARD command, 262
KEYBOARD '{F8}', 231
KeyFieldList property, 132

L

labels, 60–62, 76
Last In First Out (LIFO), 97
lblTitle property, 170
LIFO (Last In First Out), 97
LineDesc field, 137–138
lines of code, capturing, 235–236
List Box control, 203–205, 296
Load( ), 173, 174, 177–178
LoadApp( ), 50–51, 171
LoadComponents( ), 217–218
loading
files, 47–48
tables, 142–144
toolbars, 214–215, 217
LoadProxy( ), 38
Local Area Network (LAN), 129
LOCATE command, 126–127, 134, 275
LocateObject( ), 163, 164
location manager, 28
locking forms, 193–194
LockScreen property, 173
LogError( ), 234
logical .NOT. operator “!”, 51
login screen, Security Manager, 250–252

M

Main_Frame.prg, 53–54, 91, 277
Main.prg, 54–56, 274
MakeConnection( ), 150
MakeDecimals property, 87
MakeSession command, 143
Make<<Setting>> properties, 88
MakeToolbar( ), 213
Mapping tab, 294–295
mediator, 35–36
MemberClass property, 72
MemberClassLibrary property, 74
Message( ), 235
MessageBox( ), 46, 153, 227
messaging service, 28
Meta classes approach, 79–80
methods
for cursor classes, 114–115
data entry methods, 195–200
error handling, 230–233
forms, 193–194
prefixes, 21–22
writing method code, 261–262
middle-tier objects, 141
MKDIR command, 273
modal forms, 182–185
ModiClasses( ), 259
MODIFY CLASS command, 257, 259
modular construction, 13
MousePointer property, 194
multi-user environments, 162
multiple data sources, 12
multiple records, editing, 189
MyApp.h, 46
MyAppVars, 206
MyBuilder, 263
MyBuilderForms, 267
MyCopyClass.scx, 275
MyCursor.Go( ), 134
MyCursor
buffer mode, 125–126
builder, 267
capabilities, 113–114
checking for unsaved changes, 123–124
design considerations, 115
filters, 127
methods, 114–115

naming toolbars, 214–215
properties, 114
records
adding, 119–120
deleting, 120–121
navigating, 126–127
saving, 122
reverting pending edits (undo), 122
tables, 115–119
updating remote data, 130–131
validating data, 124–125, 147
MyCursorAdapter
capabilities, 133–134
validating data, 147
MyCursorAdapter.Go( ), 134
MyCursor.GetCurrentRecord( ), 127–128
MyCursorGo( ), 127–128
MyDataEnvironment. *See also* data entry forms
capabilities, 141–142
class definitions, 142
editing forms, 197
goals, 140–141
preferences, 165
saving data, 144
validation errors, 146–147
MyDataFrm, 93, 191–192, 198, 202–204
MyErrorHandler, 229–230
MyErrorLog form, 240–241
MyErrorMsg_Dialog, 233
MyFrame, collections, 98–108
MyFrame.vcx, 46
MyLocations, 163
MyMessages, 154–156
MyMessagesValid( ), 155
MyModalFrm, 183–184
MyNavList, 203–204
MyNavList.Init( ), 205
MyPreferences, 165
MyProjectInfo.scx, 271, 273
MySecurityManager, 248–250
MySplash.scx, 169–172
MyStack, 176
MySysCodesCbo, 160
MyToolbars, 212, 216–217

N

naming conventions, 18–22
National Institute of Standards and Technology (NIST), 96–97
Navigate( ), 204
Navigate event, 204
navigation cursor, 317
nested TRY/CATCH blocks, 225
New( ), 140, 195–197
NewClass, 269
NewProject form, 280
NODEFAULT command, 90
nodes, highlighting, 71
normalization process, 60
NOWAIT clause, 259

O

object hierarchy, navigating, 107–108
object properties, capturing, 236–237
ObjectError( ), 238–240
objects, error handling, 224
ODBC-compliant data, 130
ODBC connection string, 289
ODBC.INI, 130
OLE control, treeview, 70
OLEBOUNDCONTROL class, 254–255
OLECONTROL, 253
ON ERROR command, 224, 229
ON SEEK command, 119
ON SHUTDOWN command, 55
OnAfterSave( ), 122, 146, 310–311
OnBeforeSave( ), 122, 144
OnChangesVerified( ), 196–197
OnCheckMachine( ), 51
OnClose( ), 175–176
OnConfirmNew( ), 196–197
OnDelete( ), 36
one-to-many entry form, 190, 305–312
OnLaunchApp( ), 56
OnLoadAppMenu( ), 51
OnLoadSecurity( ), 252
OnNavigationFailed( ), 202–203
OnNavigationSucceeded( ), 202–203
OnNew( ), 147–148
OnReadProperties( ), 266, 268
OnSetControlFocus( ), 194
OnSetDefaultValues, 120
OnSetDevEnvironment( ), 92
OnSetForeignKeys( ), 307
OnShowSplash( ), 171
OnShutDown( ), 51, 288
OnWriteProperties( ), 268
Open( ), 116–117, 118, 134
OPEN DATABASE command, 48
OpenTables( ), 125, 142
“Operator/operand type mismatch” error, 235
optimistic buffering, 112
OptionButton class, 74
OptionGroup class, 72–73

OTHERWISE clause, 224
overriding methods, 48
overwriting defaults, 56

P

page frame, 73
PageFrame class, 72–73
Pane Customization dialog, 281
parent, changing, 268–269
passwords, Security Manager, 250
PEM Adder tool, 256, 257
PemAdder class, 258
PEMStatus( ), 236
PEMSTATUS, zero (0) value, 229
pending edits, reverting, 122
performance issues, 91
pessimistic locking (buffering), 112
Pop( ), 103, 106
POP operations, 96
Populate( ), 160, 161–162
portability, 13
predefined constants, 32, 46
preferences manager, 28
preferences, saving, 164–165, 180
prefixes, 19, 20
Prefs.dbf, 165
Primary( ), 117
primary key tag, 118
primary keys, 119, 156, 159–160
PRINT WHEN clause, 250
PRINT WHEN command, 244
“Printer Not Ready” error, 231
PrivateDatasession property, 143–144
procedural errors, 224, 229
procedures
  generic, 44–45
  specific, 45–46
Program( ), 237
Project Details form, 289–290
project manager, 44
prompting users, 193
properties
  adding and removing, 261
  CamelCase convention, 21
  hard coding, 32
  saving, 164–165
  security, 246–250
  setting, 114
proxy class, 38–39
public property, 62
Push( ), 102
PUSH operations, 96

Q

queries, executing, 130–131
QueryUnload( ), 175
QueryUnload event, 175
queues, 96–97
quitting FoxPro, 288

R

ReadMethod( ), 256
ReadProperties( ), 266
Rebuild approach, 78
record-based navigation, 202
record pointer, 190
records. *See also* data entry forms; data objects; tables
  adding, 119–120, 147–148, 196–197
  buffered, 111–112, 124–125
  deleting, 120–121
  editing, 187–190
  navigating, 126–127
  saving, 122
  writing data object contents to, 128
Refresh( ), 202–203
refreshing
  security code, 249–250
  toolbars, 214, 220
relative paths, with include files, 46
Release( ), 175
ReleaseSplash( ), 171
ReleaseToolbar( ), 219
remote data
  accessing with SQL pass through, 130–134
  concepts, 129–130
  connecting to, 130, 148–150
  sample application, 313–319, 318–319
  updating, 130–131
Remove( ), 100–101
RemoveFromFormsCollection( ), 181–182
RemoveProperty( ), 261
REPLACE statements, 234
reports manager, 29
Requery( ), 133–134
requirements list, 14
Resize( ), 100, 178–179
resizing forms, 176, 178–179
RestoreEnvironment( ), 92
restoring environments, 84, 90
RETRY command, 231
reusable frameworks, 11
run time, error handling, 233

S

SampleForm class, 37
Save( ), 18, 140, 144, 195–197
SaveAsClass( ), 253, 255
SaveLoc( ), 163
saving
  environment variables, 84
  environments, 90
  preferences, 164–165, 180
  property values, 164–165
SCATTER NAME command, 127–128
scope, prefixes, 19
_Screen.ActiveForm property, 173–174
SCREEN.ADDOBJECT( ), 94
Search Form control, 203–205
security
  applying, 246–250
  form controls, 247–248
  non-object-oriented controls, 250
  sample application, 298–303
  strategies, 243
Security class, 246–247
Security Manager, logging in, 250–252
security module, 28, 244–246
SEEK command, 119, 126–127, 134
SelectCmd property, 132, 133–134
SendUpdates property, 131
session class, 93–94
SET ASSERTS ON, 82–83
set-based navigation, 202
SET CLASSLIB command, 276–277
SET commands, 83–84, 88, 91
SET DATABASE EVENTS ON command, 48
SET DELETED command, 119, 120–121, 143
SET DEVELOPMENT ON command, 288
SET FILTER TO command, 127
SET PATH statement, 55
SET PROCEDURE TO command, 48
SET SAFETY command, 87
SET STEP ON command, 231
SET TALK command, 87, 143
SetBufferMode( ), 125
SetCurrentRecord( ), 127–129
SetEditMode( ), 191–192
SetEnv( ), 90
SetErrorState( ), 260
SetFilter( ), 127, 133–134
SetFocus( ), 65, 194, 219
SetLocation( ), 177–178
SetOrder( ), 133–134
SetPrefs( ), 216–217
SetPrimaryKey( ), 118
SetSecurity( ), 244, 247–248, 250
SetStatus( ), 170
shapes, 71–72
ShortDesc field, 159–160
Show( ), 182
ShowDevTools( ), 62
ShowMessage( ), 155–156
ShowTips property, 179–180
shutdown sequence, 46–48, 199
single point accountability, 11
SKIP clause, 244
SKIP command, 200
SKIP FOR clause, 250
soft confirmation, 193
source code, 273, 288
splash screens, 169–172
SQL pass through, 130–134
SQL server, security, 243
SQLConnect( ), 132
SQLEXEC( ), 130
SQLStringConnect( ), 130, 132, 148–150
stacks, 96–97
startup sequence, 46–48
steering document, 9–13
STEP INTO command, 231
Str2Expr( ), 157
StrExp( ), 69
StrExtract( ), 218
string values, in permanent data stores, 32
StrToFile( ), 238
StrTran( ), 266
Style object, 35
styles, 34–35
surrogate primary keys, 112–113, 119
swapping objects, 32
synchronizing tools and toolbars, 218–220
SysCodes, 132
system codes, 29, 158–162
System Codes Management screen, 203–204, 309

T

TabIndex, 194
Table Designer, 111, 292
table names, 116
TABLEREVERT command, 122
tables. *See also* records
  adding manually, 296
  Albums table, 190
  closing, 119
  creating, 273

FramePems, 259
loading, 142–144
opening, 115–118
prefixes, 21
returning records from, 127–128
sample application, 292–294
security module, 245–246
SysCodes, 158–159
updating, 316
Tables property, 132
TableUpdate( ), 112, 121–122
task list, 14
template method, 34
test plan, 14
text boxes, 67–68
Textbox classes, Meta Data approach, 78–79
three-tier architecture, 138
THROW error handling, 225
tiers, 23–27
Timer( ), 259–260
Timer class, 259–260
Timer1 property, 171
Toolbar class, 220–221
Toolbar Manager
application shutdown, 217–218
creating, 212–213, 217
instantiating, 217
loading toolbars, 214–215
toolbars
adding, 215
capabilities, 211
collection, 29, 212
data entry and, 219
docking status, 163
expected behaviors, 211–212
individual controls, 221
loading, 214–215, 217
refreshing, 214, 220
releasing, 213–214
synchronizing, 218–220
user selection, 215–217
Toolbars table, 214
ToolboxEngine.prg, 279
TOP command, 200
transparent containers, 68–69
TrapError.prg, 224
trapping errors, 227, 240
treeview control, 70–71, 203–205
True value, reversing, 51
TRY/CATCH error handling, 225

U

Unbind( ), 69
Undo( ), 140, 195–196, 219
Unload( ), 183, 184
UpdatableFieldList property, 131
Update( ), 140
UpdateNameList code, 131
UpdateNameList property, 132
UpdateTableBuffer( ), 18, 19
USE command, 115–116
users, adding, 299

V

Validate( ), 124–125
validating data, 124–125, 155, 297–298
Validation Error form, 208
validation errors, 146–147, 207–208
variables, prefixes, 19, 21
VCX file, 258
VCX form support, 181
verifying changes, 197
VFP Projects collection, 95
vfps.application handler, 281
Visible property, 182, 248
Visual FoxPro. *See also* framework development
aApplication, 49–50, 91
buffered records, 111–112
commands for configuring application environment, 46
creating main programs, 53–56
definition, 1
examples, 1–3
field mapping, 294–295
FoxPro error dialog, 226
quitting, 288
redirecting FoxPro’s builder, 262–265
startup and shutdown sequence, 46–48

W

WindowState property, 184
WindowType property, 183
wizards, 10, 318. *See also* remote data
wrappers, flexibility, 39
WriteMethod( ), 256

Z

zero (0) value, PEMSTATUS, 229
zNewProject program, 281, 289–290
zReadMe( ), 265, 266
zVisualBaseClassConfigurator.prg, 256–259