

INDEX

SYMBOLS

_ (underscore character), 942
16-bit Delphi, capturing exceptions in, 271-272
32-bit integers, 166-167
3G (third generation) mobile networks, 1120
64-bit integers, 168
802.11 technology, 1120-1121

A

abstraction, example code, 1023-1026

access methods

GetMaxLength() method, 388
SendMessage() method, 389
SetMaxLength() method, 388

accessing

field values, 314-315
Project Manager, 32
WebApp Debugger, 1082
WinHelp, 22

accessories properties, 105

actions

raAbort, 1015
raCancel, 1015
raCorrect, 1015
raMerge, 1015
raRefresh, 1015
raSkip, 1014

Active property, 301

Active Server Objects

ASP (Active Server Pages), 1050
in-process, 1060-1061
out-of-process, 1060-1061
overview, 1050-1051
Request object, 1051, 1061-1062
Response object, 1051-1052, 1059-1060
running in response to HTML input, 1063-1064
creating, 1052
CoClass Name, 1055
Instanting options, 1054-1055
New Active Server Object dialog box, 1054
Object Context option, 1055
Start In edit box, 1053
Threading Model choices, 1054-1055
debugging, 1071
using Windows 2000, 1074-1076
using Windows NT 4, 1073-1074
with MTS, 1071-1073

- NetCLX support and,
 - 1069-1070
 - overview, 1050
- Page-Level Event Methods
 - option, 1056
- recompiling, 1062-1063
- Type Library Editor, 1055-1059
- Active Server Pages.** See **ASP**
- ActiveIconHandle() method, 751**
- ActiveX, 655-656**
- ActiveX Data Objects.**
 - See **ADO**
- Adapter Dispatcher component, 1084**
- Add Web Component, 1036**
- Add() method, 403**
- AddActivity() method, 910**
- adding fields, Fields Editor, 319**
- AddInts() method, 38**
- AddObject() method, 403**
- AddRef() method, 661-665**
- AddText() method, 704**
- ADO (ActiveX Data Objects), 298, 364**
 - datasets, 300
 - dbGo for ADO, 364, 367
 - OLE DB provider for ODBC, establishing, 365-366*
 - TADOCCommand component, 372*
 - TADOCConnection component, 368, 370, 375-376*
 - Login prompt, bypassing, 370, 372
 - TADODataset component, 373*
 - TADOQuery component, 375*
 - TADOStoredProc component, 375*
 - TADOTable component, 373-375*
- ADTs (advanced data types), 958**
 - ADT client side listing, 963-965
 - ADT I.pas file listing, 959-961
 - ADT.idl listing, 958-959
 - implementation file for servers
 - listing, 961-963
- advertising services, server setup, 1004**
- Advise() method, 703**
- Aggregation, 672-673**
- aliases**
 - IDL, 944
 - Object Pascal, 86
- Align property, 436**
- AlignControls() method, 399**
- AllocateHWND() function, 751**
- Alpha CPU data structure, spinlock, 200**
- animated components, marquee components, 494-495**
 - animating, 498-510
 - destructor, 500
 - scrolling control, 501-502
 - writing, 495-497
- AnsiString type, 51**
 - migrating from Delphi 1, 57
 - Object Pascal, 52-56
- apartment option (Threading Model option), 1054**
- Apartment threading choice, 1003**
- API, tray notification icon, 748-750**
- AppBars (application desktop toolbars), 764-766**
 - component code, example
 - code, 768-775
 - TAppBar, 776-779
 - VCL form encapsulation, 766-775
- AppBrowser navigation, 30-31**
- Application Adapter component, 1083**
- application desktop toolbars.**
 - See **AppBars**
- Application Module Page Options dialog box, 1085**
- Application Partitioning scenario, package creation, 635**
- application servers**
 - building, 1007-1009
 - techniques, 1018
- Application.Initialize() method, 672**
- Application.ProcessMessage() method, 147**
- applications**
 - back end, 29
 - client/server, 998
 - COM+, 908
 - debugging, 934*
 - execution context, 908*
 - lifetime management, 910*
 - organization, 910-911*
 - resources, 912*
 - stateful versus stateless, 909-910*
 - system design, 908*
 - transactions, 911*
 - developers, 382
 - DLLs and, 250
- multitier, 998-999
 - benefits, 999-1000*
 - DataSnap, 1007-1046*
 - DataSnap architecture, 1001-1006*
- standalone
 - DLLs to deploy with, 361*
 - units required for, 360*
- WebSnap, designing, 1078
 - ApplicationTitle property, 1089*
 - component choices, 1083-1084*
 - converting to ISAPI DLL, 1107*
 - custom components, 1112-1114*
 - custom templates, 1111-1112*
 - data, displaying, 1103-1107*
 - dispatching methods, 1079*
 - file uploading, 1109-1111*
 - file uploading services, 1080*
 - HTML management, 1080*
 - image handling, 1101-1103*
 - LocateFileServices, 1108-1109*
 - logging in, 1092-1094*
 - login services, 1079-1080*
 - naming, 1085-1088*
 - navigation menu bar, 1089-1092*
 - page producer components, 1079*
 - preference data between sessions, 1099-1101*
 - server choices, 1081-1083*
 - server-side scripting, 1078*
 - session management, 1079*
 - Tadapter components, 1078-1079*
 - toolbar, adding to IDE window, 1080*
 - user preference data, managing, 1095-1098*
 - user tracking, 1080*
 - Web App Debugger option, 1083*
 - Web modules, multiple, 1078*
- ApplyRange() method, 336**
- AppSpecific moniker parameter, 895**
- AProc() method, 666**
- architecture**
 - CLX, 565-568
 - COM+ event system, 899
 - CORBA, 939-941
 - interfaces, 941-942*
 - OSAGent, 941*

database architecture, 14-15
dbExpress, 350

arguments, method arguments, 945

arithmetic operators, Object Pascal, 45-46

arrays
collections and, 714
dynamic, Object Pascal, 77-78
IDL, 944
multidimensional, 77
properties, 390, 441-443
variable type creation, 76-77
variant arrays, 729-730
Variant type, 71
 initializing, 73
 support functions, 73-75
 VarArrayCreate(), 72
 VarArrayOff(), 72

as operator, 404

ASP (Active Server Pages), 1050
in process, 1060-1061
out-of process, 1060-1061
overview, 1050-1051
Request object, 1051, 1061-1062
Response object, 1051-1052
 welcome message example, 1059-1060
 Write object, 1059
running in response to HTML input, 1063-1064

Assign() method, 395
TSomeObject property, 440
TStrings class and, 403

assignment operators, Object Pascal, 43

AssignPrn() standard procedure, 218

AssignTo() method, 395

attribute specification, property editors, 521-522

authentication levels, role-based security configuration, 885

AuthLevel moniker parameter, 896

Automation
binary data, exchanging, 724-727
collections
 arrays, 714
 creation requirements, 715
 Delphi implementation of, 715-723
 overview, 713
 rules of, 714-715
early binding, 676
events, 700-701
 client events, 705-708
 events interface, 702-703
 method pointers, 701

multicasting, 702
 overview, 701
 server events, 703-705
 with sinks, multiple, 708-712

IDispatch interface, 674-675
in-process server
 controlling, 698-700
 creating, 687-692
language support (COM), 727-728
 interfaces, 731-732
 late binding, 730
 variant arrays, 728-730
 WideString data type, 730-731
late binding, 676
out-of-process server
 controlling, 692-697
 creating, 677-687
overview, 673-674
registration, 676
servers, 170
type libraries, new interface types, 723-724

Automation Object Wizard, 912-913

Automation Object Wizard dialog box
Class Name field, 678
Instancing combo box, 678

B

back end, 29

Bank Example (CORBA)
balance method, 946
Bank C.pas file, 948-952
Bank I.pas, 948
bank server, implementation
 class for, 953-954
Bank.idl, 946-947
client source listing, 956-958
client-side stub class, 955
deposit method, 946
stub class deposit method, 955
withdraw, 946

base class hierarchy, CLX, 565-568

BaseCLX, 564
portable code compatibility and, 161

Basic Object Adaptor. See BOA

BDE (Borland Database Engine), 298
datasets, 299
dbExpress versus, 350-351

multithreaded graphics, 233-238
multithreading access, 227-233

binary data, exchanging, 724-727

Binary Large Object fields. See BLOB fields

bitmaps
component, CLX, 622
wireless, 1128-1131

bitwise operators, Object Pascal, 46

BLOB (Binary Large Object) fields, 324
sample application building, 325-329
TBlobField descendant, 324-325

Bluetooth radio technology, 1120

BOA (Basic Object Adaptor), 940

BOF property, 305-306

Bookmark property, 347

Borland Database Engine. See BDE

Borland Web site, 965

both option (Threading Model option), 1054

Both threading choice, 1004

Break procedure, Object Pascal, 92

briefcase model, client optimization techniques, 1017

Broadcast() method, 146

btnFreeLibClick() method, 271

btnLoadLibClick() method, 270

building application servers, DataSnap applications
providers, 1008
registering, 1009

business tier, 998

button notification messages, 142

buttons
adding to forms, sample application, 26-27
MyOnClickEvent() method, 391
TMouseEvent, 392
toolbar buttons, 20-21

C

calculated fields, Fields Editor, 321-322

callback function, 273
calling from, 277-279
defined, 132
EnumWindows() API method, 273-275

- TListBox event, 276
- TWindowInfo() class, 276
- CanBePooled() method, 897-898**
- Cancel() method, dataset manipulation, 314**
- CancelRange() method, 336**
- CanFocus() method, 399**
- CanModify property, 318**
- canvas-locking rules, 238**
- Canvas.Start() method, 584**
- Canvas.Stop() method, 584**
- case sensitivity**
 - CLX components, porting, 569
 - Object Pascal, 39
- case statement, Object Pascal, 89-90**
- categories, property, 538**
 - classes, 539-540
 - custom, 540-543
 - registering, 538
- CDMA technology, 1119**
- CDPD (Cellular Digital Packet Data), 1119**
- cellular phones. *See* mobile phones
- CGI (common gateway interface), 1050**
- change control methods, 1011**
- character types, Object Pascal, 50**
- ciInternal instancing choice, 1003**
- ciMultiInstance instancing choices, 1002**
- circular unit reference, 101**
- ciSingleInstance instancing choice, 1003**
- class completion feature, 30**
- class factories**
 - aggregation, 672-673
 - CreateInstance() method, 667
 - defined, 667
 - in-process COM servers, 669
 - creating an instance of, 670-671
 - DllCanUnloadNow() function, 670
 - DllGetClassObject() function, 670
 - DllRegisterServer() function, 669
 - DllUnregisterServer() function, 669
 - out-of-process COM servers, 672
 - creating instances of, 672
 - registration, 672
 - TComObject, 667-668
 - TComObjectFactory, 667-668
- class ID (CLSID), 660**
- class libraries, Qt, 565**
- Class Name field (Automation Object Wizard dialog box), 678**
- class reference, 644**
- classes**
 - process priority, 187-188
 - property categories, 539-540
 - TCanvas, 568
 - TCollection, 544-546
 - TCollectionItem, 544-545
 - defining, 546
 - editing components, 555-561
 - TForm1, 27
 - TFrameControl, 565
 - TWidgetControl, 565
 - TWinControl, 566
 - TWindowInfo, 276
 - versus components, 387
- ClassInfo() method, 405**
- ClassName() method, 405**
- ClassParent() method, 405**
- ClassType() method, 405**
- Clear() method, 403, 704**
- client application, COM+, 929-934**
- client connecting choices**
 - TCORBAConnection, 1005
 - TDCOMConnection, 1005
 - TdispatchConnection, 1005
 - TSOAPConnection, 1006
 - TSocketConnection, 1005
 - TWebConnection, 1005
- client dataset features**
 - DataSnap applications, 1039
 - two-tier applications, 1039-1041
- client events, 705-708**
- client optimization techniques, 1015**
 - briefcase model, 1017
 - limiting data packet, 1015-1017
 - sending dynamic SQL to server, 1017
- client setup DataSnap applications, 1004**
 - connection choices, 1005-1006
 - server connection, 1006
- client tasks, DataSnap applications, 1009**
 - client-side transactions, 1011-1012
 - editing data, 1010-1011
 - Error Reconciliation dialog box, 1014-1015
 - reconciling data, 1012-1013
 - retrieving data, 1009-1010
 - reverting to original version, 1011
 - undoing changes, 1011
- client-side linking, 1021-1026**
- client-side transactions, 1011-1012**
- client/server applications, 998**
- clients**
 - fat, 1000
 - queued components, 893-896
- ClientToScreen() method, 253**
- Close method(), 301**
- CloseSharedData() method, 282, 284**
- closing**
 - datasets, 301
 - mutexes, 204
- CLX (Component Library for Cross-Platform)**
 - architecture, 565-568
 - component bitmaps, 622
 - components
 - creating, 569
 - identifying on the Component Palette, 622
 - porting, 568-569
 - DataCLX, 383
 - design editors, 608
 - TddgDefaultEditor component editor, 611-612
 - TddgRadioGroupEditor custom component editor, 608-610
 - hierarchy, 384
 - NetCLX, 383
 - overview, 382, 564
 - BaseCLX, 564
 - DataCLX, 564
 - NetCLX, 564
 - VisualCLX, 564-565
 - packages, 613
 - design-time, 618-620
 - Linux packages, 614
 - runtime, 615-617
 - Windows packages, 613-614
 - registration units, 621
 - RTL, 383
 - VisualCLX, 383
- ClxDesignWindows unit, 838**
- ClxEditors unit, 838**
- ClxSprigs unit, 838**
- CM (component messages), 143-144**
- CM MOUSEENTER message, 143**
- CM MOUSELEAVE message, 143**

- CN (component notification), 143-144**
- Coclass, 655**
- CoCreateGUID() API function, 660**
- CoCreateInstance(), 671**
- CoCreateInstanceEx() COM API function, 673**
- code base, 635**
- Code Editor**
 - overview, 22
 - windows, viewing multiple, 23
- Code Explorer, 23**
- Code Insight, 33**
- code reduction, 626**
- code sharing, 252**
- Codelnsight technologies, 11**
- CoGetClassObject() function, 672**
- CoGetObject() method, 895**
- Colnitialize() function, 671**
- ColnitializeSecurity() method, 886**
- collections, Automation**
 - arrays, 714
 - creation requirements, 715
 - Delphi implementation of, 715-723
 - overview, 713
 - rules of, 714-715
- columns, dataset, 300**
- COM (Component Object Model) development, 654**
 - ActiveX, 656
 - Automation
 - binary data, exchanging, 724-727
 - collections, 713-723
 - early binding, 676
 - events, 700-712
 - IDispatch interface, 674-675
 - in-process servers, controlling, 698-700
 - in-process servers, creating, 687-692
 - language support, 727-732
 - late binding, 676
 - out-of-process servers, controlling, 692-697
 - out-of-process servers, creating, 677-687
 - overview, 673-674
 - registration, 676
 - type libraries, new interface types, 723-724
 - basic concepts, 654-655
 - Component Object Model, 654
 - marshaling, 655
 - terminologies, 655-656
 - class factories
 - aggregation, 672-673
 - CreateInstance() method, 667
 - defined, 667
 - in-process COM servers, 669-671
 - out-of-process COM servers, 672
 - TComObject, 667-668
 - TComObjectFactory, 667-668
 - COM+, goal of, 658
 - DCOM (Distributed COM), security features, 673
 - Object Pascal Language and, 658
 - HResult return type, 666
 - interfaces, 658-666
 - OLE 1 versus OLE 2, 657
 - overview, 654
 - threading models, list of, 657-658
 - ToleContainer class, sample application, 733-746
- COM library, initializing, 781**
- COM object wizard, 801**
- COM+**
 - advantages, 880-881
 - applications, creating, 908
 - execution context, 908
 - lifetime management, 910
 - organization, 910-911
 - resources, 912
 - stateful versus stateless, 909-910
 - system design, 908
 - transactions, 911
 - applications, debugging, 934
 - Automation Object Wizard, 912-913
 - client application, 929-934
 - defined, 880
 - DTC (Distributed Transaction Coordinator), 881-882
 - events, 898
 - creating, 900
 - event class servers, creating, 900-901
 - event class servers, registering, 901-902
 - event publishers, 899
 - event subscribers, 899
 - LCE (loosely coupled events), 899
 - parameter filters, 906
 - publisher filters, 906
 - publishing, 904-905
 - subscriber servers, creating, 902-903
 - subscriber servers, registering, 903-904
 - system architecture, 899
 - TCE (tightly coupled events), 898
 - framework, 913-914
 - IObjectContext methods, 915
 - OnActivate() method, 914
 - OnDeactivate() method, 914
 - goal of, 658
 - JIT (Just-In-Time), 888
 - object pooling, 897-898
 - queued components, 888
 - advantages, 889-890
 - clients, creating, 893-896
 - MSMQ technology, 888-889
 - servers, creating, 890-893
 - servers, running, 896-897
 - runtime, 906-907
 - components, configured, 907
 - contexts, 907
 - neutral threading, 907
 - registration database, 907
 - security, 882
 - multitier performance, 887
 - programmatic, 887-888
 - role-based configuration, 882-887
 - servers, installing, 928-929
 - services, list of, 881
 - tic-tac-toe sample application, 916-928
 - Transactional Data Module Wizard, 912
- combo box notification messages, 142-143**
- ComCtl132.dll, 564**
- commands**
 - File menu
 - New, 1052
 - New Application, 24
 - New Project, 968
 - Run menu
 - Install MTS Objects, 1072
 - Register ActiveX server, 692
 - Tools menu
 - EJB Deployment, 970
 - Enterprise Setup, 968
 - IDE Options, 971
 - View menu
 - New Edit Window, 23
 - Object Treeview, 1103
 - Project Manager, 32
 - To Do List, 32
- CommandText property**
 - ctQuery value, 355
 - result sets, extracting, 356
- CommandType property, 355**

comments supported, Object Pascal, 36-37**common gateway interface.**
See CGI**Common Object Request Broker Architecture.**
See CORBA**comparison operators, Object Pascal, 43-44****compatibility issues, portable code, 158**

- components, 160
- conditional defines, 158-159
- Delphi-Kylix compatibility, 161-163
- IDE issues, 160-161
- packages, 160
- units, 160

compilation, Active Server Objects, 1062-1063**compilation speed, 12-13****Compiled unit package file type, 628****compiler directives, packages, 635-637****compilers**

- Delphi 6, new features, 163
- \$IF directives*, 164
- binary DFM incompatibility*, 164
- enumeration values*, 163
- variants*, 163
- portable codes
 - compatibility issues*, 158-159
 - components*, 160
 - Delphi-Kylix compatibility*, 161-163
 - IDE issues*, 160-161
 - packages*, 160
 - units*, 160

component bitmaps, 622**component class, 655****component editors, 522**

- example code, 525
- methods,
 - Edit()*, 523
 - ExecuteVerb()*, 524
 - GetVerb()*, 524
 - GetVerbCount()*, 524
 - Paste()*, 524
- registering, 526-527
- TComponentEditor, 523
- TDefaultEditor, 524
- TddgDefaultEditor, 611-612

Component Library for**Cross-Platform. See CLX****Component Object Model. See COM development****Component Palette, 298**

- adding a marquee component, 510
- overview, 21

component security, 634**ComponentCount property, 396****ComponentIndex property, 396****components**

- ancestor classes, 432-433
- animated, marquee, 494-510
- building
 - overview*, 430
 - writing decisions*, 430-431
 - writing steps*, 431-432

CLX (Component Library for Cross Platform)

- creating*, 569
- DataCLX*, 383
- hierarchy*, 384
- NetCLX*, 383
- overview*, 382
- registering*, 621
- RTL*, 383
- VisualCLX*, 383

complexity of, 383

connection, 299-300

constructors

- Component State values*, 453
- design-time behavior*, 453
- overriding*, 452

defined, 383

destructors, overriding, 454

events and, 390

- assigning at runtime*, 391-392
- event handler*, 445
- event property*, 445
- event-dispatching method*, 445-446
- OnChange*, 390
- OnClick*, 390-391
- OnDblClick*, 390
- properties, defining*, 446-450

extensible, 29-30

graphical controls, 387

icons, 458-459

lists, 543

methods and, 390, 451

nonvisual, 385

OWL (Object Windows Library), 382

ownership, 393-394

Parent property, 394

portable code compatibility

issues, 160

porting, 268-269

ProcessExecute() method, 475-476

properties, 388

- access methods*, 388-389
- array*, 390
- enumerated*, 390
- object*, 390
- set*, 390
- simple*, 390
- properties, adding, 435
 - array properties*, 441-443
 - default array properties*, 445
 - default values*, 444-445
 - enumerated properties*, 436
 - object properties*, 438-441
 - set properties*, 437
 - simple properties*, 435

pseudo-visual, extending hints, 490-494

registering, 454-455

RTTI (Runtime Type Information), 382

SetCommandLine() method, 476-477

streaming, 392

TADOQuery, 301

TADOTable, 301

TAppBar, 766-776

TCanvas class, 403

TclientDataset, control options, 1039

Tcomponent class, 395

- methods*, 396-397
- properties, list of*, 396

TControl, 397

TCustom class, 400

TCustomConnection, 299

TDatabase, 302

TDataModule, 336-337

TDataSet, 300, 305, 315

TddgButtonEdit container

- design decisions*, 477-478
- forms, adding*, 485-488
- surfacing event*, 478
- TddgDigitalCloc*, 481-485
- Text property*, 478
- TSpeedButton control*, 478-481

TddgButtonEdit container , 477

TddgRunButton example, 470-475

TddgWaveFile, 530-537

TDispatchConnection, 1009

testing, 456-458

TgraphicControl class, 399-400

- TIBQuery, 301
- TIBTable, 301
- TListBox example, 463-470
- TMemo example, 459-463
- TObject class, 394
 - Create()* method, 394
 - Destroy()* method, 394
 - TPersistent* class, 395
- TQuery, 301
- TSQLQuery, 301
- TSQLTable, 301
- TStrings class, 400-403
- TTable, 301
- TTimer, 498
- TWidgetControl class, 398-399
- TwinControl class, 398
 - events, keyboard interaction,* 399
 - methods, types of,* 399
 - properties, types of,* 398-399
- units, creating, 433-435
- VCL (Visual Component Library)
 - hierarchies,* 384
 - overview,* 382
 - versus classes,* 387
 - versus controls,* 386
 - visual,* 385-386
 - TGraphicControl,* 387
 - TWidgetControl,* 386-387
 - TWinControl,* 386-387
 - writers,* 382
- ComponentState** property, 396, 453
- ComponentStyle** property, 396
- compound document,** 656
- ComputerName** moniker parameter, 895
- Concat()** function, 54-55
- conditional defines, for compilers,** 158-159
- configuring**
 - JBuilder, 967-968
 - security, role-based, 883-887
 - authentication levels, 885
- Connected property,** 353
- connection components**
 - TADOConnection connection component, 300
 - TDatabase connection component, 299
 - TIBDatabase connection component, 300
 - TSQLConnection connection component, 300
- ConnectionString** property, 352-353
- connections**
 - Connection Editor, 353
 - TSQLConnection component
 - Connected property,* 353
 - ConnectionString component,* 352-353
 - dbxconnections.ini configuration file,* 351-352
 - dbxdrivers.ini configuration file,* 351-352
 - LoginPrompt property,* 353-354
 - Params property,* 354
- ConnectionStringPropertyEditor,** 368
 - Microsoft OLE DB Provider For ODBC Drivers, 368
 - Text Connection, 368, 370
- constant parameters,** 95
- constants**
 - declaration, Object Pascal, 43
 - Object Pascal, 41-43
 - space allocation, Object Pascal, 41
- constructors**
 - Component State values, 453
 - design-time behavior, 453
 - objects, 105
 - overriding, 452, 501
- containers.** See **collections**
- context menu handlers,** 800, 808-811
 - registering, 812-818
- contexts, runtime,** 907
- Continue()** procedure, 92
- contract-free programming,** 28-29
- controls**
 - graphical, 387
 - versus components,* 386
- converting WebSnap applications,** 1107
- ConvertString()** method, 289
- copy hook handlers,** 800-807
- Copy()** method, 739
- CopyCallback()** method, 802-804
- CORBA (Common Object Request Broker Architecture)**
 - architecture, 939-941
 - interfaces,* 941-942
 - OSAgent,* 941
 - bank example
 - balance method,* 946
 - Bank.C.pas file,* 948-952
 - Bank.I.pas,* 948
 - bank server, implementation class for,* 953-954
 - Bank.idl,* 946-947
 - client source listing,* 956-958
 - client-side stub class,* 955
 - deposit method,* 946
 - stub class deposit method,* 955
 - withdraw method,* 946
- EJB connection**
 - APIs, predefined,* 966
 - components and,* 966
 - EJB containers,* 966
 - entity beans,* 967
 - Hello, world example,* 968-973
 - Home interfaces,* 966-967
 - Jbuilder, configuring,* 967-968
 - overview,* 965-966
 - Remote interfaces,* 966-967
 - session beans,* 967
- features of, 938-939
- IDL (Interface Definition Language), 942-943
 - ADTs (advanced data types),* 958-965
 - aliases,* 944
 - arrays,* 944
 - enumerations,* 944
 - method arguments,* 945
 - modules,* 945-946
 - sequences,* 944-945
 - structures,* 944
 - types of, mapped to Object Pascal,* 943
 - user defined types,* 944
- overview, 938
- Web services
 - CORBA client code, adding,* 978-981
 - creating,* 975-976
 - example architecture,* 975
 - SOAP client application, creating,* 977-978
- CoSetProxyBlanket()** method, 885-886
- Create New Data Source dialog box,** 366
- Create()** method, 499
 - example code, 500
 - Owner property, 393
 - TObject class and, 394
- CreateComObject()** method, 671
- CreateFileMapping()** method, 284
- CreateInstance()** method
 - class factories and, 667
 - IObjectContext, 915

CreateLinkToFile() method, 735
CreateMDIChild() method, 737
creating

DataSnap applications, 1007
application server building, 1007-1009
client tasks, 1009-1015
 descendant property editor
 object, 511-513
 IShellLink instance, 781
 mutexes, 204
 object instance, 105
 packages, 630
Package Editor, 630-631
scenarios to consider, 631-635
 shell links, 784-785
 THintWindow descendant,
 490-492

critical sections, 199-200

example code, 200-202
 versus mutexes, 203

cross-platform development, 351

csDesigning state behavior, 453

ctQuery value, 355

ctStoredProc value, 356-357

ctTable value, 355

Currency type, Object Pascal, 75

CurrToFMTBCD() function, 166

custom component editors, CLX, 608-610

D

data

displaying, WebSnap application
 design, 1103-1107
 editing DataSnap applications,
 1010-1011

Data Link Properties dialog box

Microsoft OLE DB Provider
 For ODBC Drivers, 368
 Text Connection, 368, 370

data modules

datasets, 336
sample application building, 337-347
TDataModule component, 336-337
 filter form
example code, 345-347
sample application building, 343-344

key search form

example code, 341-343
sample application building, 341

data object handlers, 800

data reconciliation, 1012-1013

data sharing, 252

across processes, 279-280
CloseSharedData() method, 282, 284
CreateFileMapping() method, 284
GlobalData property, 280-282
MapViewOfFile() method, 283
OpenSharedData() method, 282

Data Source Name. See DSN

data structure, Variant type, 64-67

data tier, 998

data-access choices, 1004

database architecture, Delphi 6

database connectivity
datasets, 300-347
overview, 299-300
 database types supported,
 298-299

database connectivity,

datasets, 300-301

bookmarks, 347
 closing, 301-305
 data modules, 336-347
 field values, accessing, 315-316
 field values, determining type,
 316
 field values, editing, 317-329
 field values, names, 317
 field values, numbers, 317
 filtering, 330-331
 manipulating, 310-314
 navigating, 305-310
 opening, 301-305
 searching, 332-336
 states, 314
 overview, 299-300

databases, supported types by Delphi 6, 298-299

DataCLX, 564

defined, 383
 portable code compatibility
 and, 161

datasets, 300-301

bookmarks, 347
 closing, 301-305

data modules, 336

sample application building, 337-347
TDataModule component, 336-337

defined, 301

field values

accessing, 315-316
 determining type, 316
 editing, 317-329
 names, 317
 numbers, 317

filtering, 330-331

manipulating, 310

example code, 310-313
methods, 314

navigating, 305-310

opening, 301-305

ranges, 335-340

searching, 332-336

states, 314

types, 301

unidirectional, limitations, 350

DataSnap applications

architecture, 1001
 connection choices, 1005-1006
 creating, 1007-1015
 deploying, 1041
DCOM configuration, 1042-1043
files to deploy, 1043-1044
internet considerations, 1044-1046
licensing issues, 1042
 examples, 1027
client dataset features, 1039-1041
joins, 1027-1028
multitable updates, 1028-1029
Web, 1030-1037
 master/detail relationships,
 1020-1026
 mistakes, 1041
 options, 1015
application server techniques, 1018
client optimization techniques, 1015-1017
server, miscellaneous, 1019

DataSnap Reconciliation Error dialog box, 1037

DataType property, 316

dBase tables, 298

record searching, 333

dbExpress, 298

applications, standalone
DLLs to deploy with, 361
units required for, 360

- architecture, 350
- cross-platform development, 351
- datasets, 300
- DBE (Borland Database Engine) versus, 350-351
- overview, 350
- TSQL Monitor component, 358-359
- TSQLClientDataset component, 359-360
- TSQLConnection component, 351
 - Connected property*, 353
 - ConnectionString*, 352-353
 - dbxconnections.ini configuration file*, 351-352
 - dbxdrivers.ini configuration file*, 351-352
 - LoginPrompt property*, 353-354
 - Params property*, 354
- TSQLDataset component, 354
 - CommandText property*, 355-356
 - CommandType property*, 355
 - ciStoredProc value*, 356-357
 - SetSchemaInfo procedure*, 357-358
- TSQLQuery, 358
- TSQLTable component, 358
- TSWLStoredProc component, 358
- unidirectional datasets, limitations, 350
- dbGo for ADO, 364-365, 367**
- OLE DB provider for ODBC, establishing, 365-366
- TADOCCommand component, 372
- TADOCConnection component, 368
 - ConnectionString Property Editor*, 368, 370
 - Login prompt, bypassing*, 370, 372
 - transaction processing*, 375-376
- TADODataset component, 373
- TADOQuery component, 375
- TADOStoredProc component, 375
- TADOTable component, 373-375
- dbxconnections.ini configuration file, 351-352**
- dbxdrivers.ini configuration file, 351-352**
- DCOM (Distributed COM), 655**
 - configuration, 1042-1043
 - security features, 673
- DCOM page Web site, 1043**
- DCOMCNFG utility, 1042**
- DCUs (Delphi compiled units), 253**
- debugging**
 - Active Server Objects, 1071
 - using Windows 2000*, 1074-1076
 - using Windows NT 4*, 1073-1074
 - with MTS*, 1071-1073
 - COM+ application, 934
 - shell extensions, 800-801
- declaration objects, 105-107**
- default thread, 174**
- default value parameters, Object Pascal, 38**
- Default() method, message routing process, 147**
- DefineBinaryProperty(), 530-537**
- DefineProperties() method, 395**
- DefineProperty() function, example code, 529-530**
- DefineProperty() method, 528**
- defining interfaces, 114-115**
- Delete() method**
 - dataset manipulation, 314
 - TStrings class and, 403
- Delivery moniker parameter, 896**
- Delphi**
 - contract-free programming, 28-29
 - developing EJB client in, 971-973
 - history, 15
 - Delphi 1*, 16
 - Delphi 2*, 16-17
 - Delphi 3*, 17-18
 - Delphi 4*, 18
 - Delphi 5*, 18-19
 - Delphi 6*, 19
 - IDE, 19-20
 - AppBrowser navigation*, 30-31
 - class completion feature*, 30
 - Code Editor*, 22-23
 - Code Explorer*, 23
 - Code Insight*, 33
 - Component Palette*, 21
 - docking bays*, 31
 - FormDesigner*, 22
 - interface/implementation navigation*, 31
 - main menu*, 20
 - Object Browser*, 31
 - Object Inspector*, 22
 - Object TreeView*, 23-24
 - Project Manager*, 32
 - syntax highlighting*, 32
 - To Do List*, 32
 - toolbars*, 20-21
 - productivity, 10-11
 - compilation speed*, 12-13
 - database architecture*, 14-15
 - language features*, 13-14
 - software design*, 15
 - visual development environment*, 11-12
 - products, 8
 - Delphi 6 Enterprise*, 10
 - Delphi 6 Personal*, 8-9
 - Delphi 6 Professional*, 9
 - prototyping, 29
- Delphi 1**
 - migration from, 171
 - porting from, 50, 57
- Delphi 2, migration from, 168**
 - automation servers, 170
 - Boolean types, changes to, 168-169
 - GetChildren() method, 170
 - ResourceStrings, 169
 - RTL changes, 169
 - TCustomForm property, 169-170
- Delphi 3, migration from**
 - 64-bit integers, 168
 - Real type command, 168
 - unsigned 32-bit integers, 166-167
- Delphi 4, migration from, 165**
 - database issues, 166
 - Internet development issues, 165-166
 - RTL issues, 165
 - VCL issues, 165
- Delphi 5, migration from, 164**
 - cardinal unary negation, 164-165
 - writable types constants, 164
- Delphi 6**
 - database architecture
 - database connectivity*, 299-300
 - database types supported*, 298-299
 - datasets*, 300-347
 - Object Browser, 567
- Delphi 6 Developer's Guide**
 - introduction to, 1-3
 - Web site, 2

Delphi applications, threads

- fibers, 238-244
- misuse of, 175-176
- multiple, 175, 192-238
- multithreading, 185-186
- priorities, 187-189
- resuming dynamically, 190
- single-threaded user interface, advantages of, 182
- suspending dynamically, 190
- timing, 190-192
- TThread object, 176-184

Delphi compiled units.

See DCUs

Delphi online help, Using Databases, 298**DEPLOY.TXT document, 1042****deploying**

- DataSnap applications, 1041
- DCOM configuration, 1042-1043
- files, 1043-1044
- internet considerations, 1044-1046
- licensing issues, 1042
- THintWindow descendant, 494

descendant property editor object, creating, 511-513**descendants, TBlobStream, 325****Design and Runtime Packages for Components scenario, package creation, 631-634****design editors, CLX, 608**

- TddgDefaultEditor component editor, 611-612
- TddgRadioGroupEditor custom component editor, 608-610

Design Features Only (No Components) IDE**Enhancements scenario, 635****Design Package Only for Components scenario, 634****design package type, 628****design-time packages**

- CLX, 618-620
- data-aware, 619-620
- nondata-aware, 618-619

DesignConst unit, Open Tools API, 837**DesignEditors unit, Open Tools API, 837****designide packages, CLX, 618****DesignInfo property, 396****DesignIntf unit, Open Tools API, 837****DesignMenus unit, Open Tools API, 837****DesignWindows unit, Open Tools API, 837****Destroy() method, 106**

- example code, 500
- TObject class and, 394

destruction objects, 106-107**destructor method, 106****destructors, overriding, 454**
determining type, field values, 316**dialog boxes**

- Thread object item in New Items dialog box, 178
- Active Server Objects, 1056
- Add Web Component, using Web Page Editor, 1036
- Application Module Page Options, 1085
- Automation Object Wizard, 678
- Create New Data Source, 366
- Customize toolbar, 20-21
- Data Link Properties, 368
 - Microsoft OLE DB Provider For ODBC Drivers, 368*
 - Text Connection, 368, 370*
- DataSnap Reconciliation Error, 1037
- Edit Configuration, 968
- Environment Options, 23
- Error Reconciliation, 1014-1015
- Insert Object, 733
- Install COM+ Objects, 928
- New Active Server Object, 1054
- New Field, 321-322
- New Items, 677
- New Remote Data Module, 1001
- New WebSnap Application
 - Data Module option, 1083*
 - Page Module option, 1083*
 - server choices, 1081-1083*
 - Web App Debugger option, 1083*

Project Options, 854

Run Parameters, 935, 1074

Thread object item in New Items dialog box, 178

Web App Components 1083-1084

dialog property editor, editing, 555-561**difference set, 81****DisableAlign() method, 399****Dispatch() method, sending messages, 141****Dispatcher Actions component, 1084****dispatching methods, 1079****DispatchMessage() method, message routing process, 147****Distributed COM. See DCOM****Distributed Transaction Coordinator. See DTC****DllCanUnloadNow() function, 670****DllRegisterServer() function, 669-670****DLLs (Dynamic Linked Libraries)**

- advantages of, 248
- applications, 250
- callback function, 273
 - calling from, 277-279*
 - EnumWindows() API method, 273-275*
 - TListBox event, 276*
 - TWindowInfo class, 276*
- ClientTo Screen() method, 253
- code sharing, 252
- data sharing, 252
- data sharing across processes, 279-280
 - CloseSharedData() method, 282, 284*
 - CreateFileMapping() method, 284*
 - GlobalData property, 280-282*
 - MapViewOfFile() method, 283*
 - OpenSharedData() method, 282*

dynamic linking, 248

entry/exit events, 266

- sample code, 268-270*
- source code, 267*

exceptions

- capturing in 16-bit Delphi, 271-272*
- Safecall directive and, 272*

executable code, 250

explicitly loading, 263-264

- FreeLibrary() method, 265*
- GetProcAddress() method, 265-266*

LoadLibrary() method, 265

exporting objects from, 287

- ConvertString() method, 289*
- example code, 288*
- limitations, 287*

STRINGCONVERTLIB directive, 290-292

- TStringConvert object, 289-290*

FreeLibrary() method, 271

hiding implementation, 252

instance, 250

linking, dynamic *versus* static, 250-252

- LoadLibrary() method, 270
 - memory-mapped files, 249, 282
 - CloseSharedData()* method, 284
 - CreateFileMapping()* method, 284
 - MapViewOfFile()* method, 283
 - modal forms, displaying, 256-258
 - AHandle* parameter, 258
 - ShareMem* property, 258-259
 - modeless forms, displaying, 259-260
 - modules, 250
 - overview, 248-249
 - packages, 248
 - PenniesToCoins() method
 - example, 253-254
 - exports* clause, 254
 - functions*, importing, 256
 - interface* units, defining, 254-255
 - TcoinsRec* variable, 255
 - preferred base address, 249
 - related terms, list of, 250
 - resource sharing, 252
 - shared memory and,
 - example code*, 284-286
 - source code*, 286-287
 - tasks, 250
 - TButton component, 261-262
 - TLabel component, 261-262
 - TMaskEdit component, 261-262
 - versus* packages, 627
 - when to deploy, 361
 - DllUnrefsterServer()** function, 669
 - do-and-assign operators**, Object Pascal, 47
 - docking bays**, 31
 - DoCoMo** Web site, 1132
 - documentation, wizards**, 839
 - documents**
 - ActiveX, 656
 - compound, 656
 - DEPLOY.TXT, 1042
 - DoHalfMinute()** method, 450
 - DoSearch()** method, 224
 - DoTimerOnTimer** method, *example code*, 498
 - drag-and-drop fields**, Fields Editor, 323
 - drag-and-drop handlers**, 800
 - drawing offscreen bitmap**, 495-496
 - drop target handlers**, 800
 - DSN (Data Source Name)**, 365
 - Create New Data Source dialog box, 366
 - File DSN, 365
 - System DSN, 365
 - User DSN, 365
 - DTC (Distributed Transaction Coordinator)**, 881-882
 - dual interface**, 676
 - dynamic arrays**, 77-78
 - dynamic link libraries**. See **DLLs**
 - dynamic linking**, 248-252
 - Dynamic Method Table (DMT)**, 109
 - dynamic methods**, 108
 - dynamic SQL, sending to the server**, 1017
- E**
- early binding**, 676
 - Edit Configuration dialog box**, 968
 - edit notification messages**, 143
 - Edit()** method, 523
 - dataset manipulation, 314
 - editing**
 - field values, datasets, 317-329
 - property as text, 513-517
 - TcollectionItem components, 555
 - editing with dialog, property editors**, 519-521
 - Editor, Fields**, 318
 - adding fields, 319
 - calculated fields, 321-322
 - drag-and-drop fields, 323
 - lookup fields, 322
 - Object Inspector, 321
 - TField object descendants, 319-320
 - editors**
 - component, 522
 - example code*, 525
 - methods*, 523-524
 - registering*, 526
 - TComponentEditor*, 523
 - TDefaultEditor*, 524
 - dialog property, TcollectionItem components, 555-561
 - property, 510-522
 - EJB (Enterprise Java Bean)**, 880
 - APIs, predefined, 966
 - components and, 966
 - EJB containers, 966
 - entity beans, 967
 - Hello, world example, 968-969
 - AppServer*, deploying, 970-971
 - client*, building, 970
 - EJB client*, developing in Delphi, 971-973
 - JBuilder*, client test application in, 969-970
 - SIDL* file, generating, 971
 - Home interfaces, 966-967
 - JBuilder, configuring, 967-968
 - overview, 965-966
 - Remote interfaces, 966-967
 - session beans, 967
 - elliptical hint creation**, 490-493
 - EnableAlign()** method, 399
 - EnableCommit()** method, 915
 - encapsulation**, 103
 - EncryptAlgorithm** moniker parameter, 896
 - End User Adapter component**, 1084
 - Enterprise Java Bean**. See **EJB**
 - Enterprise Setup command (Tools menu)**, 968
 - entity beans**, 967
 - entry/exit events**, 266
 - sample code*, 268-270
 - source code*, 267
 - EnumConnectionPoints()** method, 702
 - enumerated properties**, 390
 - adding to components, 436
 - enumerated types, obtaining RTTI on**, 422-423
 - enumeration values, compiler issues**, 163
 - enumerations, IDL**, 944
 - EnumWindows()** API method, 273-275
 - Environment Options dialog box**, 23
 - environments, object-based**, 105
 - EOF property, TDataSet component**, 305-306
 - error handling, Object Pascal**, 118-121
 - Error Reconciliation dialog box**, 1014-1015
 - error reporting, WAP (Wireless Application Protocol) sample application**, 1127
 - events**
 - assigning values, rules, 498
 - Automation, 700-701
 - client events*, 705-708
 - events interface*, 702-703

- method pointers*, 701
- multicasting*, 702
- overview*, 701
- server events*, 703-705
- with sinks*, *multiple*, 708-712
- COM+, 898
 - creating*, 900
 - event class servers*, *creating*, 900-901
 - event class servers*, *registering*, 901-902
 - event publishers*, 899
 - event subscribers*, 899
 - LCE (loosely coupled events)*, 899
 - parameter filters*, 906
 - publisher filters*, 906
 - publishing*, 904-905
 - subscriber servers*, *creating*, 902-903
 - subscriber servers*, *registering*, 903-904
 - system architecture*, 899
 - TCE (tightly coupled events)*, 898
- component structure and, 390
 - assigning at runtime*, 391-392
 - OnChange*, 390
 - OnClick*, 390-391
 - OnDb1Click*, 390
- contract-free programming, 28-29
- DoHalfMinute method, 450
- event handler, 445
- event property, 445
- event-dispatching method, 445-446
- exit/entry, 266
 - sample code*, 268-270
 - source code*, 267
- IEventObj.MyEvent() method, 904
- messaging relationships, 154
- OnClick, 445
- OnFilterRecord, 331
- OnHalfMinute, 447
- OnKeyDown, 28
- OnMouseDown, 28
- overview, 22, 28
- properties, defining, 446-450
- TListBox, 276
- TNotifyEvent property, 446
- TwinControl, keyboard interaction, 399
- subscriber servers*, *implementation unit*, 902-903
- tic-tac-toe sample application*, 916-928
- component editors, 525
 - registering*, 526-527
 - TcomponentEditor*, 523
- components
 - creating*, 434-435
 - registering*, 454-455
 - TddgRunButton*, 470-475
 - testing*, 456-458
 - TListBox*, 463-470
 - TMemo*, 459-463
- context menu handlers, 813-818
- constructors, overriding, 452
- copy hook handlers, 805-807
- Create() method, 500
- critical sections, 200-202
- data modules
 - filter form*, 345-347
 - key search form*, 341-343
- data reconciling, 1013
- datasets
 - manipulating*, 310-313
 - navigating*, 306-310
 - opening and closing*, 302-305
 - ranges*, 339-340
- DefineProperty() function, 529-530
- Destroy() method, 500
- DLLs
 - data sharing across processes*, 280-282
 - displaying modal forms from*, 257-258
 - displaying modeless forms from*, 259-260
 - explicitly loading*, 263-264
 - exporting objects from*, 288
 - shared memory and*, 284-286
 - TButton component*, 261-262
 - TLabel component*, 261-262
 - TMaskEdit component*, 261-262
- DoTimerOnTimer() method, 498
- EJB (Enterprise Java Bean)
 - EJB ClientMain Application file listing*, 972-973
 - Hello, world example*, 969-971
- example codes**
 - Active Server Objects
 - NetCLX support and*, 1070
 - Response object*, 1051
 - running in response to HTML input*, 1063-1064
 - source code*, 1057-1058
 - ADTs (advanced data types)
 - ADT client side listing*, 963-965
 - ADT I.pas file listing*, 959-961
 - ADT implementation file for servers listing*, 961-963
 - ADT.idl listing*, 958-959
 - AppBars
 - component code*, 768-775
 - TAppBar*, 776-779
 - Bank Example (CORBA)
 - Bank C.pas file*, 948-952
 - Bank I.pas*, 948
 - bank server*, *implementation class for*, 953-954
 - Bank.idl*, 946
 - client source listing*, 956-957
 - client-side stub class*, 955
 - stub class deposit method*, 955
 - BLOB (Binary Large Object) fields, 327-329
 - client updates to master/detail, 1022
 - CLX
 - custom spinner*, 571-583
 - custom spinner*, *displaying images*, 592-598
 - custom spinner*, *handling mouse events*, 585-591
 - design-time packages*, *data-aware*, 619-620
 - design-time packages*, *nondata-aware*, 618-619
 - registration units*, *nondata-aware components*, 621
 - runtime packages*, *data-aware*, 616-617
 - runtime packages*, *nondata-aware*, 615
 - COM+
 - applications*, *stateful versus stateless*, 909-910
 - client application*, 929-934
 - event class servers*, *creating*, 900-901
 - events*, *publishing*, 904-905
 - framework*, 914

- events
 - MyOnClickEvent()* method, 391
 - OnClick* event, 391
 - properties, defining, 447-450
 - TMouseEvent*, 392
- exception classes, 121-122
- exceptions, execution flow, 124-125
- exporting functions from packages, 644-645
- fiber creation, 241-244
- fiber unit, 240
- functions and procedures, 93-94
- generating add-in forms, 638-643
- GetValue()* method
 - implementation, 513
- HTML, DataSnap applications, 1030-1034
- icon handlers, 822-827
- implementing multiple interfaces, 116
- IncLine()* method, 499
- InfoTip* handlers, implementing, 828-832
- in-process server, Automation
 - controlling, 698-700
 - creating, 687-692
- invoking functions from packages, 646-648
- IShellLink* interface, 780
- joins, DataSnap applications, 1028
- marquee components, 502-508
 - painting, 497
 - PaintLine()* method, 496
 - testing, 508-510
- messaging
 - CaichI* message, 148-149
 - CIMain.PAS*, source code, 149-153
 - CM MOUSE LEAVE*, 143
 - CM MOUSEENTER*, 143
 - GetMess* message handling example, 137
 - mouse button click example, 130
 - OnMessage* event, 140
 - user-defined, *Broadcast()* method, 146
 - user-defined, within applications, 144-145
 - using *Messages* unit, 134-135
 - WM CTLCOLOR* message, 139
 - WM PAINT* message, 135-136
- multitable update, 1029
- multithreaded BDE access, part 1, 228-230
- multithreaded BDE access, part 2, 230-233
- multithreaded graphics, 234-237
- multithreading, 186
- mutexes, 204-206
- OnButtonTimer()* method, 754
- overloading methods, 109
- PQA (Palm Query Applications)
 - client sample document, 1133-1134
 - server application sample, 1134-1135
- property categories, custom, 540-543
- property editors, 515-516
 - editing with dialog, 520
 - registering, 518
- queued components
 - clients, creating, 893-896
 - servers, creating, 891-893
 - servers, running, 896-897
- reintroducing method names, 110
- Remote *DataModule* creation, 1007
- RTTI
 - obtaining for enumerated types, 422-423
 - obtaining for integers, 420-421
 - obtaining for methods, 416-420
 - obtaining for set types, 423-425
 - obtaining on objects, 407-411
 - TTypeData Structure* example, 406-407
- scope, 98
- search thread, multithreaded application, 219-223
- search thread, priority adjustment, 225-227
- security
 - GetObjectContext()* method, 887
 - IsCallerInRole()* method, 887
- semaphores, 207-209
- server updates to master/detail, 1022
- SetActive()* method, 501-502
- SetValue()* method
 - implementation, 514
- shell extensions, initializing, 809
- shell links
 - creating, 785
 - information, getting and setting, 786-789
 - sample application, main unit, 791-796
 - sample application, supporting unit 1, 796-797
 - sample application, supporting unit 2, 798-799
- ShowWindow()* procedure, 755
- Synchronize()* method, *ThrdU.PAS* Unit, 183-184
- TADOTable component, 374
- TddgDefaultEditor* component editor, 611-612
- TddgRadioGroupEditor* custom component editor, 609-610
- TddgWaveFile* component
 - DefineProperties()* method, 531
 - LoadFromStream()* method, 532
 - ReadData()* method, 531
 - SaveToStream()* method, 532
 - WriteData()* method, 531
- TDefaultEditor*, 524
- thread-local storage, 194-196
- TObject, 113
- tray notification icon, 762-764
- TrayWndProc()* method, 753-754
- try..except Exception Handling Block, 120
- try..except..else Exception Handling Block, 120
- TThread object, 176-177
- TTimer* component, initializing, 498
- TTrayNotifyIcon* component, 755-762
- two-tier applications, 1039
- Type Library Editor, 1056
- unit providing utility routines and abstraction, 1023-1026
- user interface, multithreaded application, 211-218
- WAP (Wireless Application Protocol)
 - phone display example, 1128-1131
 - sample application, 1125-1127
- Wave File component, 533-537
- Web Services, creating
 - invokable interface, defining, 986-987, 989
 - invokable interface, implementing, 988
 - testing, 989-991

- WndProc() method, 767-768
 - writing DDG Search wizards
 - main form, bringing up files in the IDE Code Editor*, 859-860
 - main form, complete*, 860-867
 - exception classes**
 - execution flow, 123-125
 - Object Pascal, 121-123
 - exceptions**
 - DLLs
 - capturing in 16-bit Delphi*, 271-272
 - Safecall directive and*, 272
 - invalid variant type conversion, 70
 - executable code, DLLs and**, 250
 - Execute() method**, 181
 - ExecuteVerb method**, 524
 - exit/entry events**, 266
 - sample code, 268-270
 - source code, 267
 - experts**, 30
 - explicitly loading, DLLs**, 263-264
 - FreeLibrary() method, 265
 - GetProcAddress() method, 265-266
 - LoadLibrary() method, 265
 - Explorer tab (Environment Options dialog box)**, 23
 - exporting**
 - functions from packages, 644-648
 - objects from DLLs, 287
 - ConvertString() method*, 289
 - example code*, 288
 - limitations*, 287
 - STRINGCONVERTLIB directive*, 290-292
 - TStringConvert object*, 289-290
 - expressions, Variant type**, 69-70
 - extending hints**, 490-494
 - extensible applications, generating add-in forms**, 637-644
 - extensible components**, 29-30
- F**
- fat client**, 1000
 - fault tolerance**, 1000
 - features, Object Pascal**
 - constants, 41-43
 - default value parameters, 38
 - error handling, 118-121
 - functions, 93-103, 107-110, 114-115
 - interfaces, 114-118
 - objects, 110-113, 121-127
 - OOP, 103-107
 - operators, 43-47
 - overloading, 37-38
 - parentheses, 37
 - procedures, 93-103, 107-110, 114
 - types, 48-92
 - variables, 39-41
 - fiber creation, example code**, 241-244
 - fiber unit, example code**, 240
 - fibers**, 238-244
 - field definitions. See fields**
 - FieldCount property**, 317
 - FieldList property**, 317
 - FieldNo property**, 317
 - fields**
 - accessing, 105
 - BLOB (Binary Large Object) fields, 324
 - sample application building*, 325-329
 - TBlobField descendant*, 324-325
 - calculated, Fields Editor, 321-322
 - drag-and-drop, Fields Editor, 323
 - lookup, Fields Editor, 322
 - objects, 104
 - Fields Editor**, 318
 - adding fields, 319
 - calculated fields, 321-322
 - drag-and-drop fields, 323
 - lookup fields, 322
 - Object Inspector, 321
 - TField object descendants, 319-320
 - FieldValues() property**, 315
 - File DSN**, 365
 - file locations, package types**, 629-630
 - File menu commands**
 - New, 1052
 - New Application, 24
 - New Project, 968
 - file uploading**
 - services, 1109-1111
 - WebSnap applications, 1109-1111
 - files**
 - deploying, 1043-1044
 - memory-mapped, 282-283
 - CloseSharedData() method*, 284
 - CreateFileMapping() method*, 284
 - MapViewOfFile() method*, 283
 - FileSaveAsItemClick() method**, 738
 - Filtered property**, 330
 - filtering datasets**, 330-331
 - filters**, 906
 - FindAllFiles() method**, 223
 - FindConnectionPoint() method**, 702
 - FindFirst() method**, 332
 - FindKey() method**, 333
 - FindLast() method**, 332
 - FindNearest() method**, 334
 - FindNext() method**, 332
 - FindPrior() method**, 332
 - Finterger property**, 444
 - firewalls**, 1044-1046
 - First() method**
 - TDataSet component, 305
 - unidirectional datasets and, 350
 - flags, TPropertyAttribute**, 521-522
 - FMTBCDToCurr() function**, 166
 - Focused() method**, 399
 - folders**, 781
 - for loop**, 90-91
 - Form Designer**, 22
 - form wizards**, 868-869
 - Format moniker parameter**, 895
 - forms**
 - adding buttons to, sample application, 26-27
 - adding to Component Palette, 485-488
 - empty, source code, 24-26
 - modal forms, displaying from DLLs, 256-259
 - modeless forms, displaying from DLLs, 259-260
 - framework, COM+, 913-914**
 - IObjectContext methods, 915
 - OnActivate() method, 914
 - OnDeactivate() method, 914
 - free option (Threading Model option)**, 1054
 - Free threading choice**, 1004
 - Free() method**, 106
 - FreeBookmark() property**, 348
 - FreeLibrary() method**, 265, 271
 - FSomeObject property**, 440

functions

AllocateHWND(), 751
 Concat(), 55
 DLLs, importing, 256
 example code, 93-94
 exporting from packages, 644-648
 GetDoubleClickTime, 753
 GetTextMetrics, 495
 GetThreadTimes(), 190, 192
 HandleReconcileError(), 1014
 InitWizard(), 855
 invoking from packages, example code, 646-648
 Object Pascal, 93-94
 methods, 107-110
 RegisterPropertyInCategory(), 538
 Result local variable, 94
 SHAppBarMessage(), 764-766
 Shell NotifyIcon(), 748-750
 VarArrayCreate(), 72
 VarArrayLock(), 74
 VarArrayOf(), 72
 VarAsType(), 75
 VarFromDateTime(), 75
 VarIsEmpty(), 75
 VarIsNull(), 75
 VarToDateTime(), 75
 VarToStr(), 75
 VarType(), 75

G

General Packet Radio Service (GPRS), 1120
generating add-in forms, 637-643
GetBaseClassInfo() method, 412
GetBookmark() property, 348
GetChildren() method, 170
GetClassAncestry() method, 413
GetClassProperties() method, 413-414
GetCommandString() method, 810
GetConnectionInterface() method, 703
GetDoubleClickTime() function, 753
GetIDsOfName() method, 675
GetInfoTip() method, 828
GetMaxLength() access method, 388
GetObjectContext() method, 887, 910
GetPackageInfo() method, 819

GetProcAddress() method, 265-266
GetPropInfos() method, 414
GetText() method, 478
GetTextMetrics function, 495
GetThreadTimes() function, 190, 192
GetTypeData() method, 412
GetValue() method, 513-514, 517
GetVerb method, 524
GetVerbCount method, 524
GlobalData property, 280-282
globally unique identifiers. See GUID
GotoBookmark() property, 348
GotoKey() method, 334
GPRS (General Packet Radio Service), 1120
graphical controls, 387
GSM technology, 1119
GUID (globally unique identifiers)
 CLSID (class ID), 660
 CoCreateGUID() API function, 660
 generating new, 660
 IID (interface ID), 660
 overview, 659-660

H

Handle property, 386-387
HandleReconcileError() function, 1014
handlers
 context menu, 800, 808-818
 copy hook, 800-807
 data object, 800
 drag-and-drop, 800
 drop object, 800
 drop target, 800
 icons, 800, 818
 interfaces, 819-821
 package flags, 818-819
 registering, 822-827
 InfoTip, 827-832
 interfaces, 827-828
 registration, 833
 property sheet, 800
handling messages
 inherited procedure, 138
 OnMessage event, 139-140
 procedure requirements, 135
 GetMess message handling example, 137
 MessageBeep() method, 137-138

naming, 136

WM_PAINT message

example, 136

result values, 139

WM_KILLFOCUS message

example, 138

WM_SYSCOMMAND, 139

HashAlgorithm moniker

parameter, 896

help system, WinHelp, 22

hiding applications, tray

notification icon, 755-762

hiding implementation, 252

highlighting, syntax

highlighting, 32

Hint property, 752

hints, surfacing, 752

Home interfaces, 966-967

HResult return type, 666

HTML, DataSnap applications, 1030-1034

hwnd field, messaging system and, 131

I

I-mode, 1132

IAmAMessage method, 109

IAmDynamic method, 108

IAmStatic method, 108

IAmVirtual method, 108

IAutoTestDisp automation object, 686

IClassFactory interface, 667

icon handlers, 800, 818

interfaces, 819-821

package flags, 818-819

registering, example code, 822-827

Icon property, write method, 752

icons

components, 458-459

surfacing, tray notification

icon, 752

tray notification, 748

API, 748-750

message handling, 751

IContextMenu interface, 810-811

IContextMenu.QueryContext Menu() method, 810

IDE, 19

AppBrowser navigation, 30-31

class completion feature, 30

Code Editor

overview, 22

windows, viewing multiple,

23

- Code Explorer, 23
- Code Insight, 33
- Component Palette, 21
- docking bays, 31
- Form Designer, 22
- interface/implementation navigation, 31
- main window,
 - main menu*, 20
 - toolbars*, 20-21
- Object Browser, 31
- Object Inspector, 22
- ObjectTreeView, 23-24
- Options command (Tools menu), 971
- Project Manager, accessing, 32
- syntax highlighting, 32
- To Do List, 32
- IDispatch**
 - collections and, 714-715
 - interface, 674-675
- IDL (Interface Definition Language), 942**
 - ADTs (advanced data types), 958
 - ADT client side listing*, 963-965
 - ADT L.pas file listing*, 959-961
 - ADT implementation file for servers listing*, 961-963
 - ADT.idl listing*, 958-959
 - aliases, 944
 - arrays, 944
 - bank example, 946
 - Bank C.pas file*, 948-952
 - Bank L.pas*, 948
 - bank server, implementation class for*, 953-954
 - Bank.idl*, 946-947
 - balance method, 946
 - deposit method, 946
 - enumerations, 944
 - method arguments, 945
 - modules, 945-946
 - overview, 942-943
 - sequences, 944-945
 - structures, 944
 - types of, mapped to Object Pascal, 943
 - user defined types, 944
 - withdraw method, 946
 - client source listing*, 956-958
 - client-side stub class*, 955
 - stub class deposit method*, 955
- IEnumVARIANT methods, list of, 718**
- IExtractIcon.GetIconLocation() method, 820-821**
- if statement, 88-89**
- IID (interface ID),**
 - defined, 660
 - interfaces and, 665
- IIOOP (Internet Inter-ORB Protocol), 939**
- IIS (Internet Information Server), 1050**
- image handling, WebSnap application design, 1101-1103**
- impersonation levels, role-base security configuration, 885-887**
- implementing**
 - context menu handlers, example code, 813-818
 - icon handlers, example code, 822-827
 - InfoTip handlers, example code, 828-832
 - interfaces, example code, 115-116
- implements directive, 116-117**
- in-place activation, 657**
- in-process COM servers, 669**
 - creating of instance of, 670-671
 - DllCanUnloadNow() function, 670
 - DllGetClassObject() function, 670
 - DllRegisterServer() function, 669
 - DllUnregisterServer() function, 669
- InLine() method, 499**
- increment and decrement procedures operators, Object Pascal, 46-47**
- IndexName property, switching indexes, 335**
- IndexOf() method, Exchange class and, 403**
- InfoTip handlers, 827**
 - Implementing, example code, 828-832
 - interfaces, 827-828
 - registering, 833
- inheritance, 103-104**
- inherited function, messaging system and, 138**
- InheritsFrom() method, RTTI and, 405**
- initialization/finalization code, 100**
- Initialize() method, 808, 833**
- InitializeControl() method, 722**
- initializing**
 - arrays, variant, 73
 - COM library, 781
 - shell extensions, 808-811
- InitWizard() function, 855**
- Inprise, 18**
- InputQuery() method, 354**
- Insert Object dialog box, 733**
- Insert() method**
 - dataset manipulation, 314
 - Exchange class and, 403
- InsertObjectDialog() method, 737**
- Install COM+ Objects dialog box, 928**
- Install MTS Objects command (Run menu), 1072**
- installing**
 - COM+ server, 928-929
 - packages into Delphi IDE , 629-630
- instance, DLLS and, 250**
- InstanceSize() method, 405**
- instancing choices**
 - ciInternal, 1003
 - ciMultiInstance, 1002
 - ciSingleInstance, 1003
 - server setup, DataSnap applications, 1002-1003
- Instancing combo box (Automation Object Wizard dialog box), 678**
- Instancing option, Active Server Object creation, 1055**
- Instantiation, 106**
- Integers, obtaining RTTI on, 420-421**
- InterBase Express, 298-300**
- interface ID. See IID**
- interface objects, Open Tools API, 836**
- interface units, 254-255**
- interface/implementation navigation, 31**
- InterfaceConnect() method, 708**
- interfaces**
 - Corba architecture and, 941-942
 - defining, 114-115
 - dual interface, 676
 - Home, 966-967
 - IClassFactory, 667
 - icon handlers, 819-821
 - IContextmenu, 810-811
 - IIDs and, 665
 - Implementing, 115-117
 - InfoTip handlers, 827-828
 - IPersistFile, 784-785

IShellExtInit, 808-809
 IShellLink, 780, 784-790
 example code, 780
 instance creation, 781
 using, 781-783
 IUnknown, 659
 AddRef() method, 661
 declarations, 661
 language support, 731-732
 method aliasing, 665-666
 Object Pascal, 114
 overview, 658-659
 Remote, 966-967
 rules for using, 117-118
 VMTs (virtual method tables), 658
 tables, 658

IntergerProp property, 444
Internal Instance option, 1055
internet considerations,
deploying, 1044-1046
Internet Information Server.
 See IIS
Internet Inter-ORB Protocol.
 See IIOF
Internet Server API/Netscape
Server API. See ISAPI/NSAPI
InternetExpress, 1034-1037
intersection calculation sets,
82
invalid variant type conversion
 exception, **Variant types, 70**
Invoke() method, 674, 955
InvokeCommand() method, 811
ObjectContext methods, 915
IPersistFile interface, 784-785
IPersistFile.Load() method, 820
is operator, 404
ISAPI/NSAPI (Internet Server
API/Netscape Server API),
1050
IsCallerInRole() method, 887
 IObjectContext, 915
IsExecutableFile() method, 475
IShellExtInit interface, 808-809
IShellLink interface, 780,
784-790
 example code, 780
 instance creation, 781
 using, 781-783
IsInTransaction() method, 915
IsSecurityEnabled() method,
915
Item() method, 715
IUnknown, 659
 AddRef() method, 661
 declaration, 661
 QueryInterface() method, 666

J-K

JBuilder
 client test application, building,
 969-970
 configuring, 967-968
JIT (Just-In-Time), 888
joins, DataSnap applications,
1027-1028
Journal moniker parameter,
896
Just-In-Time. See JIT, 888

KDE Window Manager, 565
keyboard events, 399
Kylix, portable code
compatibility, 161-162
 calling conventions, 163
 linux conditional language, 162
 PIC format, 162-163
 platform issues, 163

L

language features, 13-14
language support for COM,
727-728
 interfaces, 731-732
 late binding, 730
 variant arrays, 729-730
 variants, 728-729
 WideString data type, 730-731
Last() method, 305
late binding, 676, 730
LCE (loosely coupled events),
899
licensing issues, 1042
lifetime-managed types, 53-54
Linux packages, CLX, 614
list box notification messages,
143
listings. See **example codes**
Listitem object, 716
Lists, components, 543
 TCollection class, 544-555
 TCollectionItem class, 544-546,
 555-561
load balancing, 1000
load scheduling, 890
Loaded() method, 458
LoadFromFile() method, 403,
738
LoadFromStream() method, 738
LoadLibrary() method, 265, 270
Local File Service component,
1084
Locate() method, 332-333
LocateFileServices compo-
 nent, 1108-1109

LockServer() method, 667
Logic, range checking, 59
logical operators, Object
Pascal, 44
login services, 1079-1080
LoginPrompt property, 370,
372
 TSQL Connection component,
 353-354
lookup fields, Fields Editor,
322
loops, Object Pascal, 90
 Break procedure, 92
 Continue() procedure, 92
 for loop, 90-91
 repeat..until loop, 92
 while loop, 91-92
loosely coupled events.
 See LCE
IParam field, **messaging**
systems and, 131

M

m-commerce systems, 1138
main window (IDE), 20-21
MainWndProc() method, 147
manipulating
 datasets, 310-313
 methods, 314
MapViewOfFile() method, 283
marquee components, 494-495
 example code, 502-508
 painting, example code, 497
 PaintLine() method, 496
 testing, example code, 508-510
 writing, 495
 animating, 498-508
 Component Palette addition,
 510
 offscreen bitmap, drawing,
 495-496
 painting, 497
 testing, 508-510
marshaling, 655, 940
master/detail relationships
 client updates, example code,
 1022
 client-side linking, 1021-1026
 nested datasets, 1020-1021
 server updates, example code,
 1022
MDAC (Microsoft Data Access
Components), 364
membership set, 81
memory
 allocation, 62-68
 shared memory, DLLs and,
 284-287

memory-mapped files, 249, 282-283

- CloseSharedData() method, 284
- CreateFileMapping() method, 284
- MapViewOfFile() method, 283

message cracking, 28**message handling**

- methods, 109
- tray notification icon, 751

MessageBeep() procedure, 137-138**messaging, 130**

- callback function, 132
- CatchIt message example
 - codes, 148-149
- CIMain.PAS source code example, 149-153
- CLX components, porting, 569
- CM (component messages), 143-144
- CN (component notification), 143-144
- event relationships, 154
- hwnd field, 131
- lParam field, 131
- message field, 131
- Message loop component, 132
- Message queue component, 132
- message specific records, 134-135
- Messages unit, 134-135
- mouse button click example, 130
- notification messages, 142
 - button notification, 142
 - combo box notification, 142-143
 - edit notification, 143
 - list box notification, 143
- process procedure requirements, 136
- processing, 133
 - GetMess* message handling example, 137
 - inherited elements*, 138
 - MessageBeep()* method, 137-138
 - OnMessage* event, 139-140
 - procedure requirements, 135-136
 - result values, 139
 - WM_SYSCOMMAND*, 139
- routing process, 147
- sending, 140
 - Dispatch()* method, 141
 - Perform()* method, 140-141
 - Post()* method, 141
 - SendMessage()* method, 141

- system overview, 130
- TMessage record, 133-134
- TMsg record, 133-134
- user-defined, 144
 - Broadcast()* method, 146
 - messages between applications*, 145-146
 - messages within applications*, 144-145
- Window procedure component, 132
- Windows Message, types of, 131-132
- WM_KILLFOCUS message example, 138
- wParam field, 131

metadata representation, 357-358**method aliasing, 665-666****method arguments, 945****method pointers, 701****methods**

- ActiveIconHandle(), 751
- Add(), 403
- AddActivity(), 910
- AddInts(), 38
- AddObject(), 403
- AddRef(), 661-665
- AddText(), 704
- Advise(), 703
- AlignControls(), 399
- Application.Initialize, 672
- Application.ProcessMessage, 147
- ApplyRange(), 336
- AProc, 666
- Assign(), 395
 - TSomeObject* property, 440
 - TStrings* class and, 403
- AssignTo(), 395
- Broadcast(), 146
- CanBePooled(), 897-898
- Cancel(), 314
- CancelRange(), 336
- CanFocus(), 399
- Canvas.Start(), 584
- Canvas.Stop(), 584
- change control, 1011
- ClassInfo(), 405
- ClassName(), 405
- ClassParent(), 405
- ClassType(), 405
- Clear(), 704
- ClientToScreen(), 253
- Close(), 301
- CloseSharedData(), 282, 284
- CoCreateInstance(), 671
- CoGetClassObject(), 672
- CoGetObject(), 895

- CoInitialize(), 671
- CoInitializeSecurity(), 885
- component editors,
 - Edit()*, 523
 - ExecuteVerb()*, 524
 - GetVerb()*, 524
 - GetVerbCount()*, 524
 - Paste()*, 524
- component structure and, 390
- ConvertString(), 289
- Copy(), 739
- CopyCallback(), 802-804
- CoSetProxyBlanket(), 885-886
- Create(), 499
 - Owner* property, 393
 - Object* class and, 394
- CreateComObject(), 671
- CreateFileMapping(), 284
- CreateInstance()
 - class factories* and, 667
 - IObjectContext*, 915
- CreateLinkToFile, 735
- CreateMDIChild(), 737
- creating, 107
- DefaultHandler(), 147
- defined, 107
- DefineProperties(), 395
- Delete()
 - dataset manipulation*, 314
 - TStrings* class and, 403
- Destroy(), 106, 394
- DisableAlign(), 399
- DisableCommit(), 915
- Dispatch(), 141
- DispatchMessage(), 147
- DllCanUnloadNow(), 670
- DllGetClassObject(), 670
- DllRegisterServer(), 669
- DllUnregisterServer(), 669
- DoHalfMinute(), 450
- DoSearch(), 224
- Edit(), 314
- EnableAlign(), 399
- EnableCommit(), 915
- EnumConnectionPoints(), 702
- EnumWindows() API, 273-275
- event-dispatching, 445-446
- example code, creating a method, 107
- Exchange(), 403
- Execute(), 181
- FindAllFiles, 223
- FindConnectionPoint(), 702
- FindFirst(), 332
- FindKey(), 333
- FindLast(), 332
- FindNearest(), 334
- FindNext(), 332
- FindPrior(), 332

- First()
 - TDataSet* component, 305
 - unidirectional datasets and, 350
 - Focused(), 399
 - Free(), 106
 - FreeLibrary(), 265, 271
 - GetBaseClassInfo(), 412
 - GetChildren(), 170
 - GetClassAncestry(), 413
 - GetClassProperties(), 413-414
 - GetCommandString(), 810
 - GetConnectionInterface, 703
 - GetIDsOfName(), 675
 - GetInfoTip(), 828
 - GetMaxLength(), 388
 - GetObjectContext, 910
 - GetObjectContext(), 887
 - GetPackageInfo(), 819
 - GetProcAddress(), 265-266
 - GetPropInfos(), 414
 - getter methods, 426
 - GetText(), 478
 - GetTypeData(), 412
 - GetValue(), 514, 517
 - GotoKey(), 334
 - IAmAMessage, 109
 - IAmDynamic, 108
 - IAmStatic, 108
 - IAmVirtual, 108
 - IContextMenu.QueryContextMenu(), 810
 - IEventObj.MyEvent() method, 904
 - IExtractIcon.GetIconLocation(), 820-821
 - IncLine(), 499
 - IndexOf(), 403
 - InheritsFrom(), 405
 - Initialize(), 808, 833
 - InitializeControl(), 722
 - InputQuery, 354
 - Insert()
 - dataset manipulation*, 314
 - TStrings* class and, 403
 - InsertObjectDialog(), 737
 - InstanceSize(), 405
 - interdependencies, 451
 - InterfaceConnect(), 708
 - Invoke(), 674, 955
 - InvokeCommand(), 811
 - IPersistFile.Load(), 820
 - IsCallerInRole(), 887, 915
 - IsExecutableFile(), 475
 - IsInTransaction(), 915
 - IsSecurityEnabled(), 915
 - Item(), 715
 - Last(), 305
 - Loaded(), 458
 - LoadFromFile(), 403, 738
 - LoadFromStream(), 738
 - LoadLibrary(), 265, 270
 - Locate(), 332-333
 - LockServer(), 667
 - MainWndProc(), 147
 - MapViewOfFile(), 283
 - Move(), 403
 - MoveBy(), 305
 - MyOnClickEvent(), 391
 - Next(), 721
 - TDataSet* component, 305
 - unidirectional datasets and, 350
 - objects, 104
 - obtaining RTTI on, 416-420
 - OnActivate(), 914
 - OnClear(), 707
 - OnDeactivate(), 914
 - OnEndPage, 1056
 - OnServerMemoChanged(), 707
 - OnStartPage, 1056
 - Open(), 301
 - OpenSharedData(), 282
 - overloading, 109
 - overriding, 109
 - Paint(), 238, 497, 584
 - PasteSpecialDialog(), 739
 - Perform(), 140-141
 - Post()
 - dataset manipulation*, 314
 - sending messages*, 141
 - Prior(), 305
 - private exposure, 451
 - ProcessExecute(), 470, 475-476
 - ProcessMessage(), 147
 - protected exposure, 451
 - public, 451
 - published exposure, 451
 - QueryInterface(), 666
 - ReAlign(), 399
 - Register(), 454-455
 - RegisterActiveObject(), 710
 - RegisterInterface(), 987
 - RegisterNonActive(), 715
 - RegisterWindowMessage(), 147
 - reintroducing names, 110
 - Release(), 661-665
 - Remove(), 715
 - Resolve(), 790
 - SaveToFile(), 403, 738
 - SaveToStream(), 738
 - SearchStr, 278
 - Self variable, 110
 - SendMessage(), 141, 389
 - SendText(), 893
 - SendTrayMessage(), 750
 - SetAbort()
 - applications, creating*, 910
 - IObjectContext*, 915
 - object deactivation*, 888
 - SetAppBarEdge, 766
 - SetAppBarPos, 766
 - SetCommandLine(), 475-477
 - SetComplete()
 - applications, creating*, 910
 - IObjectContext*, 915
 - object deactivation*, 888
 - SetKey(), 333
 - SetKey.GotoNearest(), 334
 - SetMaxLength(), 388
 - SetRange(), 335-336
 - SetRangeEnd(), 336
 - SetRangeStart(), 336
 - SetSomeProp(), 441
 - setter methods, 426
 - SetText, 478
 - SetValue(), 514, 517
 - ShellExecute(), 470
 - SHGetSpecialFolderPath(), 781-783
 - ShowModal, 260
 - StrPosProc(), 279
 - Synchronize(), 182-184
 - Tcomponent class, types of, 396-397
 - TControl.Click(), 446
 - TerminateThread(), 267
 - TrayWndProc(), 751, 753
 - TSQLDataSet.ExecSQL(), 355
 - TwinControl, 399
 - types, 108
 - dynamic*, 108
 - message-handling*, 109
 - static*, 108
 - virtual*, 108
 - wizard initialization, 847
 - WndProc(), 767-768
 - Write, 1059
- Microsoft OLE DB Provider For ODBC Drivers (Data Link Properties dialog box), 368**
- Microsoft Transaction Server. See MTS**
- migration**
- from Delphi 1, 171
 - from Delphi 2, 168
 - automation servers*, 170
 - Boolean types, changes to*, 168-169
 - GetChildren()* method, 170
 - ResourceStrings*, 169
 - RTL changes*, 169
 - TCustomForm* property, 169-170

- from Delphi 3
 - 64-bit integers*, 168
 - Real type command*, 168
 - unsigned 32-bit integers*, 166-167
 - from Delphi 4, 165
 - database issues*, 166
 - Internet development issues*, 165-166
 - RTL issues*, 165
 - VCL issues*, 165
 - from Delphi 5, 164
 - cardinal unary negation*, 164-165
 - writable types constants*, 164
 - mistakes, DataSnap applications**, 1041
 - mobile phones**, 1118
 - mobile wireless devices**, 1118
 - mobile phones, 1118
 - PalmOS devices, 1118
 - Pocket PCs, 1119
 - RIM BlackBerry, 1119
 - modal forms, from DLLs, displaying**, 256-258
 - AHandle parameter, 258
 - ShareMem property, 258-259
 - modeless forms, from DLLs, displaying**, 259-260
 - models, briefcase**, 1017
 - modules**
 - DLLs and, 250
 - IDL, 945-946
 - mouse**
 - clicks, 752-754
 - TMouseEvent procedure, 392
 - Move() method**, 403
 - MoveBy() method**, 305
 - MSMQ (Microsoft Message Queue)**, 880
 - queued components, 888-889
 - advantages*, 889-890
 - clients, creating*, 893-896
 - servers, creating*, 890-893
 - servers, running*, 896-897
 - MTA (multithreaded apartment)**, 658
 - MTS (Microsoft Transaction Server)**, 880, 1071-1073
 - multicasting**, 702
 - multidimensional arrays**, 77-78
 - multiple inheritance**, 103-104
 - Multiple Instance option**, 1055
 - multiple threads**, 175
 - BDE access, 227-233
 - graphics, 233-238
 - managing, 192
 - storage, thread-local*, 192-196
 - synchronization*, 196-210
 - non-UI VCL, 175
 - sample application, 210-211
 - search thread
 - priority adjustment*, 224-227
 - sample application*, 219-224
 - UI VCL, 175
 - user interface, sample application, 211-218
 - multithreaded apartment**. See **MTA**
 - multithreaded BDE access**
 - example code, part 1, 228-230
 - example code, part 2, 230-233
 - multithreaded graphics**, 233-238
 - multithreading**, 185-186
 - multithreading BDE access**, 227-233
 - multitier applications**, 998-999
 - benefits, 999-1000
 - creating, 1007-1015
 - DataSnap architecture, 1001
 - client setup*, 1004-1006
 - RDM*, 1001
 - server setup*, 1001-1004
 - deploying, 1041-1046
 - mistakes, 1041
 - options, 1015-1026
 - multitier performance security**, 887
 - mutexes**, 202, 206
 - closing, 204
 - creating, 203-204
 - example code, 204-206
 - signaled, 206
 - versus* critical sections, 203
 - MyOnClickEvent() method**, 391
- ## N
- Name property**, 396
 - names, field values**, 317
 - naming conventions**, 637
 - navigation**
 - AppBrowser, 30-31
 - datasets, 305-306
 - interface/implementation, 31
 - neither runtime and design package type**, 628
 - nested datasets**, 1020-1021
 - NetCLX**, 564
 - Active Server Objects and, 1069-1070
 - defined, 383
 - portable code compatibility and, 161
 - neutral option (Threading Model option)**, 1055
 - neutral threading**, 907
 - New Active Server Object dialog box**, 1054
 - New Application command (File menu)**, 24
 - New command (File menu)**, 1052
 - New Edit Window command (View menu)**, 23
 - New Field dialog box**, 321-322
 - New Items dialog box**, 677
 - New Project command (File menu)**, 968
 - New Remote Data Module dialog box**, 1001
 - New WebSnap Application dialog box**
 - Data Module option, 1083
 - Page Module option, 1083
 - server choices, 1081-1083
 - Web App Debugger option, 1083
 - Next() method**, 721
 - TDataSet component, 305
 - unidirectional datasets and, 350
 - non-UI VCL**, 175
 - nonvisual components**, 385
 - notification messages**
 - button notification, 142
 - combo box notification, 142-143
 - edit notification, 143
 - list box notification, 143
 - null-terminated string types**, 61-63
 - numbers, field values**, 317
- ## O
- Object Browser**, 31, 567
 - Object Inspector**
 - categories, viewing, 22
 - contents, arranging, 22
 - Fields Editor, 321
 - help system, 22
 - overview, 22
 - object linking and embedding**. See **OLE**
 - Object Management Group**. See **OMG**
 - object oriented programming**. See **OOP**

Object Pascal, 330

case sensitivity, 39
 comments supported, 36-37
 constants,
 declaration, 43
 space allocation, 41
 features
 constants, 41-43
 default value parameters, 38
 error handling, 118-121
 functions, 93-103, 107-110, 114
 interfaces, 114-118
 objects, 110-113, 121-127
 OOP, 103-107
 operators, 43-47
 overloading, 37-38
 parentheses, 37
 procedures, 93-103, 107-110, 114
 types, 48-92
 variables, 39-41
 language, COM development
 and, 658-666
 operators

arithmetic, 45-46
 assignment, 43
 bitwise, 46
 comparison, 43-44
 do-and-assign, 47
 gets, 43
 increment and decrement procedures, 46-47
 logical, 44
 variables, *declaration, 40-41*

object pooling, 897-898**object properties, 390**

 adding to components, 438-441

object reference, 940**Object Repository, 337****Object Request Brokers.**

 See **ORBs**

Object TreeView

 commands (View menu), 1103
 overview, 23-24

Object Windows Library.

 See **OWL**

object-based environments, 105**object-oriented techniques, 15****objects**

 Active Server Objects, creating, 1052
 CoClass Name, 1055
 Instancing options, 1054-1055
 New Active Server Object dialog box, 1054

-np command line option, 1053

Object Context option, 1055
 Start In editbox, 1053
 Threading Model choices, 1054-1055

 automation, 1001

 exporting from DLLs, 287

ConvertString() method, 289

example code, 288

limitations, 287

STRINGCONVERTLIB

directive, 290-292

TstringConvert object,

289-290

 FSomeObject property, 440

 Object Pascal, 82-83

friends, 112

properties, 110-111

TObject, 113

visibility specifiers, 111-112

 obtaining RTTI on, 407-415

 OOP, 103-105

 SetSomeObject property, 439

 SetSomeProp() method, 441

 signaled, 206-207

 SomeObject property, 440

 TCopyHook, 804

 TSomeObject property, 438-439

obtaining information about a package, 648-651**ODBC**

 DSN (Data Source Name), 365-366

 overview, 364

OLE (object linking and embedding), 655**OLE 1 versus OLE 2, 657****OLE containers, 655****OLE DB, 364****OLE objects, 655****OleVariant type, 75****OMG (Object Management Group), 938****OnActivate() method, 914****OnButtonTimer() method, 754****OnChange event, 390****OnClear() method, 707****OnClick event, 390-391, 445****OnDbClick event, 390****OnDeactivate() method, 914****OnEndPage method, 1056****OnFilterRecord event, 331****OnHalfMinute event, 447****OnKeyDown event, 28****online help, Delphi, 298****OnMessage event, 139-140, 147****OnMouseDown event, 28****OnServerMemoChanged() method, 707****OnStartPage method, 1056****OOP (object oriented programming),**

 encapsulation, 103

 inheritance, 103-104

 objects

creating an instance, 105

declaration, 105-107

destruction, 106-107

field access, 105

fields, 104

instantiation, 106

methods, 104

properties, 104-105

open array parameters, 95-98**Open Tools API,**

 Delphi supported versions, 836

 interface objects, 836

 sample application building

writing DDG Search

wizards, 855-867

writing DLL-based wizards,

843-854

writing form wizards,

868-876

writing simple wizards,

839-843

 units, 836

ClxDesignWindows, 838

ClxEditors, 838

ClxSprigs, 838

DesignConst, 837

DesignEditors, 837

DesignIntf, 837

Designmenus, 837

DesignWindows, 837

obsolete, 838-839

PropertyCategories, 837

ToolsAPI, 837

TreeIntf, 837

VCLEditors, 838

VCLSprigs, 837

VCSIntf, 837

Open() method, 301**OpenSharedData() method, 282****operators**

 arithmetic, 45-46

 assignment, 43

 bitwise, 46

 comparison, 43-44

 gets, 43

 increment and decrement

 procedures, 46-47

 logical, 44

 Object Pascal, 43

 sets, 81-82

ORBs (Object Request Brokers), 938-940
OSAgent, 941
out-of-process COM servers, 672
out-of-process server, 692-697
overloading methods, 109
overloading, Object Pascal, 37-38
overriding
 constructors, 501
 methods, 109
OWL (Objects Windows Library), 382
Owner property
 Create() method, 393
 TComponent class, 396
ownership, component structure and, 393-394

P

Package Editor, 630-631
package files
 Compiled unit, 628
 Runtime/Design, 629
 Runtime/Design package symbol files, 628
 Types, package source, 628
package flags, 818-819
package source package file type, 628
package syntax, 102-103
packages
 CLX, 613
design-time, 618-620
designide, 618
Linux packages, 614
runtime, 615-617
Windows packages, 613-614
 compiler directives, 635
 {\$WEAKPACKAGEUNIT}, 636-637
 {\$DESIGNONLY ON}, 636
 {\$G}, 636
 {\$SIMPLICITBUILD OFF}, 636
 {\$RUNONLY ON}, 636
 {\$IMPORTEDDATA OFF}, 636
 creating
 Package Editor, 630-631
 scenarios to consider, 631-635
 DLLs, 248
 exporting functions, 644-648
 extensible applications using runtime (add-in), 637-644
 naming conventions, 637

obtaining information about, 648-651
 overview, 626
 package files
Compiled unit type, 628
package source type, 628
Runtime/Design package symbol type, 628
Runtime/Design type, 629
 portable code compatibility issues, 160
 reasons for using
code reduction, 626
distribution size, 626
partitioning, 627
third-party component distribution, 627
 reasons not to use, 627
 types, 628
 versioning, 635
 versus DLLs, 627
 units, 101-103

paDialog attribute, 522

page producer components, 1079

Page-Level Event Methods option (New Active Server Object dialog box), 1056

Paint() method, 238, 497, 584

painting, 497

Palm Query Applications.

See PQA

PalmOS devices, 1118

PAnsiChar string type, 51

Paradox tables, 298, 333

parameter filters, 906

parameters

constant, 95
 open array, 95-98
 reference, 95
 Value, 94
 variable, 95

Params property, 354

Parent property, 394

parentheses, Object Pascal, 37

partitioning packages, 627

passing parameters to functions and procedures, 94-98

Paste method, 524

PasteSpecialDialog() method, 739

PathName moniker parameter, 895

PChar null-terminated string type, 61-63

PChar string type, 51

Perform() method, 140-141

Personal Web Server. See PWS

phones. See mobile phones

PIC (Position Independent Code), 162

platforms, cross platform development, 351

POA (Portable Object Adaptor), 940

Pocket PCs, 1119

pointers, 83-85

polymorphism, 103

portable code

Delphi 6, new features, 163

\$IF directives, 164

Binary DFM incompatibility, 164

enumeration values, 163

variants, 163

writing, 158-161

Portable Object Adaptor.

See POA

porting from Delphi 1, 50

Position Independent Code (PIC), 162

positioning methods, 399

Post() method

dataset manipulation, 314

sending messages, 141

PQA (Palm Query Applications)

client sample document,

1133-1134

overview, 1132-1133

server application sample,

1134-1135

preemptive multitasking, 174

preferred base address, 249

presentation tier, 998

primary thread, 174

Prior() method, 305

priority class, 187-188

Priority moniker parameter, 896

priority, relative, 187-189

PrivLevel moniker parameter, 896

procedures

Break, 92

Continue(), 92

example code, 93-94

Object Pascal, 93-94

methods, 107-110

passing parameters, 94-98

scope, 98-99

units, 99-103

Register(), 569

SetLength(), 55-56

ShellExecute(), 218-219

ShowWindow(), 755

SizeOf(), 50

- VarArrayUnlock(), 74
 - VarCast(), 75
 - VarClear(), 75
 - VarCopy(), 75
 - ProcessExecute() method, 470, 475-476**
 - ProcessMessage() method, 147**
 - productivity, Delphi, 10-11**
 - compilation speed, 12-13
 - database architecture, 14-15
 - language features, 13-14
 - software design, 15
 - visual development environment, 11-12
 - programmatically security, 887-888**
 - programming, contract-free, 28-29**
 - Project Manager, accessing, 32**
 - project Options dialog box, 854**
 - properties**
 - Active, 301
 - adding to components, 435
 - array properties, 441-443*
 - default array properties, 445*
 - default values, 444-445*
 - enumerated properties, 436*
 - object properties, 438-441*
 - set properties, 437*
 - simple properties, 435*
 - assigning values through RTTI, 426-428
 - BOF, 305-306
 - Bookmark, 347
 - CanModify, 318
 - component structure and, 388
 - access methods, 388-389*
 - array, 390*
 - enumerated, 390*
 - object, 390*
 - set, 390*
 - simple, 390*
 - ComponentState, 453
 - default value assignment, 508
 - defined, 22
 - EOF, 305-306
 - FieldCount, 317
 - FieldList, 317
 - FieldNo, 317
 - FieldValues, 315
 - Filtered, 330
 - FreeBookmark, 348
 - GetBookmark, 348
 - GotoBookmark, 348
 - Hint, 752
 - Icon, 752
 - IncludePathURL, 1036
 - IndexName, 335
 - objects, 104-105, 110-111
 - TCanvas.Handle, 568
 - TDataSet.State, 314
 - TField object, 316
 - TField.FieldName, 317
 - property categories, 538**
 - classes, 539-540
 - custom, example code, 540-543
 - registering, 538
 - property definition, streaming nonpublished component data, 528-537**
 - property editors, 510, 513**
 - editing with dialog, example code, 520
 - example code, 515-516
 - registering, example code, 518
 - writing, 511
 - attribute specification, 521-522*
 - descendant property editor object, 511-513*
 - editing the property as text, 513-517*
 - editing with dialog, 519-521*
 - registering, 517-519, 522*
 - property sheet handlers, 800**
 - PropertyCategories unit, Open Tools API, 837**
 - prototyping, 29**
 - providers, 1008**
 - pseudo-visual components, 490-494**
 - publishing events**
 - example code, 904-905
 - IEventObj.MyEvent() method, 904
 - parameter filters, 906
 - publisher filters, 906
 - PWideChar string type, 51**
 - PWS (Personal Web Server), 1050**
- ## Q-R
- Qt class library, 565**
 - Query, 301**
 - QueryInterface() method, 666**
 - queued components**
 - advantages, 889-890
 - clients, creating, 893-896
 - MSMQ technology, 888-889
 - servers, creating, 890-893
 - servers, running, 896-897
 - QueueName moniker parameter, 895**
 - raAbort action, 1015**
 - raCancel action, 1015**
 - raCorrect action, 1015**
 - RAD (Rapid Application Development)**
 - components, 11
 - overview, 1078
 - radio technologies**
 - 3G (third generation), 1120
 - 802.11, 1120-1121
 - Bluetooth, 1120
 - CDMA, 1119
 - CDPD (Cellular Digital Packet Data), 1119
 - GPRS (General Packet Radio Service), 1120
 - GSM, 1119
 - TDMA, 1119
 - raMerge action, 1015**
 - range checking logic, 59**
 - ranges, TTable component, 335-336**
 - raRefresh action, 1015**
 - raSkip action, 1014**
 - RDM (Remote Data Model)**
 - building application servers, 1007-1008
 - creating, 1001
 - ReAlign() method, 399**
 - reconciling data**
 - DataSnap applications, 1012-1013
 - example code, 1013
 - record contention, resolving, 1018**
 - record searching, TTable component**
 - FindKey() method, 333
 - FindNearest() method, 334
 - GotoKey() method, 334
 - indexes, secondary, 335
 - ranges, 335-336
 - SetKey() method, 333
 - SetKey..GotoNearest() method, 334
 - records**
 - Object Pascal, 78-80
 - variant, 66
 - recursion algorithm, 224**
 - Red Dispatcher component, 1084**
 - reference class, 644**
 - reference parameters, 95**
 - region, 493**
 - Register ActiveX server command (Run menu), 692**
 - Register() method, 454-455**
 - Register() procedure, 569**

RegisterActiveObject() COM

API function, 710

registering

- building application servers, 1009
- component editors, 526-527
- context menu handlers, 812-818
- copy hook handlers, 805-807
- icon handlers, 822-827
- InfoTip handlers, 833
- property categories, 538
- property editors, 517-519, 522

RegisterInterface() method, 987**RegisterNonActiveX() function, 715****RegisterPropertyInCategory() function, 538****RegisterWindowMessage() API function, 145-147****registration**

- Automation objects, 676
- out-of-process servers, 672

registration database, 907**registration unit for Delphi 6****Developers Guide****components, 633****reintroducing method names, 110****relative priority, 187-189****Release() method, 661-665****Remote Data Model. See RDM****Remote DataModule creation, 1007****Remote interfaces, 966-967****Remove() method, 715****repeat..until loop, 92****Repository, Object, 337****Request object, 1051, 1061-1062****Requires folder, 631****Resolve() method, 790****resource sharing, 252****resources, application creation, 912****Response object, 1051-1052**

- Response.Write method, 1061
- welcome message example, 1059-1060
- Write method, 1059

Result local variable, functions, 94**result sets, extracting, 356****result values, messaging handling, 139****retrieving data, 1009-1010****reverting to original version, 1011****RIM BlackBerry, 1119****role-based security, 882-883**

- authentication levels, 885
- impersonation levels, 885-887

rows, dataset, 300**RTL (runtime library)**

- CLX, components, porting, 568
- defined, 383

RTTI (Runtime Type Information), 14, 126-127, 382, 451

- as operator and, 404
- assigning values to properties through, 426-428
- is operator and, 404
- obtaining

*ancestry for objects, 413**existence of object**properties, 414-415**for enumerated types, 422-423**for integers, 420-421**for object properties, 413-414**for objects, 412-413**for set types, 423-425**on method pointers, 416-420**on objects, 407-411*

overview, 403

TObject class, 405

TTypeData structure example, 406-407

typesafe programming, 404

rules

canvas-locking, 238

interfaces, 117-118

placing packages on Requires folder, 631

placing units into Contains folder, 630-631

Run menu commands

Register ActiveX server, 692

Install MTS Objects, 1072

Run Parameters dialog box, 935, 1074**runtime**

COM+, 906-907

*components, configured, 907**contexts, 907**neutral threading, 907**registration database, 907*

design package type and, 628

runtime packages

CLX, 615-617

data-aware, example code, 616-617

nondata-aware, example code, 615

Runtime Type Information.

See RTTI

Runtime/Design package file type, 628-629**S****Safecall**

- calling convention, 163
- defined, 686
- directives, DLLs and, 272

SavePoint, client-side transactions, 1011-1012**SaveToFile() method, 403, 738****SaveToStream() method, 738****Scalability, 890****scope, 98-99****scrolling control, 501-502****search thread, multithreaded application, example code, 219-224****search thread, priority adjustment, example code, 225-227****searching datasets, 332-336****SearchStr() method, 278****security**

CoInitializeSecurity() method, 885-886

COM+, 882

*multitier, 887**programmatic, 887-888**role-based, 882-883*

configuration, 883-887

component, 634

CoSetProxyBlanket, 886

CoSetProxyBlanket() method, 885

DCOM features, 673

GetObjectContext() method, 887

IsCallerInRole() method, 887

WTLS (Wireless Transport Layer Security), 1124

Self variable methods, 110**semaphores**

example code, 207-209

thread synchronization, 207

sending messages, 133, 140

Dispatch() method, 141

Perform() method, 140-141

Post() method, 141

SendMessage() method, 141

SendMessage() access**method, 389****SendMessage() method, 141****SendText() method, 893****SendTrayMessage() method, 750**

- server connection, client setup, 1006**
- server events, 703-705**
- server setup**
 - advertising services, 1004
 - data-access choices, 1004
 - DataSnap applications, 1001
 - instancng choices, 1002-1003
 - threading choices, 1003-1004
- server-based wireless data technologies, 1121**
 - I-mode, 1132
 - PQA (Palm Query Applications)
 - client sample document, 1133-1134*
 - overview, 1132-1133*
 - server application sample, 1134-1135*
 - SMS (Short Message Service), 1121
 - WAP (Wireless Application Protocol)
 - advantages of, 1122-1123*
 - architecture, 1121-1122*
 - bitmaps, 1128-1131*
 - error-reporting, 1127*
 - sample application, 1125-1127*
 - WML (wireless markup language), 1123-1124*
 - WTLS (Wireless Transport Layer Security), 1124*
 - WTLS (Wireless Transport Security), 1124*
- server-side scripting, 1078**
- servers**
 - application, building, 1007-1009
 - automation, 170
 - COM+, installing, 928-929
 - event class, 900-902
 - in-process, 669
 - miscellaneous options, 1019
 - object reference, 940
 - queued components, 890-893, 896-897
 - SendText() method, 893
 - subscriber
 - creating, 902-903*
 - registering, 903-904*
- session beans, 967**
- session management, 1079**
- SessionsService component, 1084**
- set properties, 390**
 - adding to components, 437
- set types, obtaining RTTI on, 423-425**
- Set8087CW() function, 165**
- SetAbort() method**
 - applications, creating, 910
 - IObjectContext, 915
 - object deactivation, 888
- SetActive() method, example code, 501-502**
- SetAppBarEdge() method, 766**
- SetAppBarPos() method, 766**
- SetCommandLine() method, 475-477**
- SetComplete() method**
 - applications, creating, 910
 - IObjectContext, 915
 - object deactivation, 888
- SetKey() method, 333**
- SetKey..GotoNearest() method, 334**
- SetLength() procedure, 55-56**
- SetMaxLength() access method, 388**
- SetRange() method, 335-336**
- SetRangeEnd() method, 336**
- SetRangeStart() method, 336**
- sets, Object Pascal**
 - operators, 81-82*
 - overview, 80*
 - using, 80*
- SetSchemaInfo() method, 3 57-358**
- SetSomeObject property, 439**
- SetSomeProp() method, 441**
- setter methods, 426**
- SetText() method, 478**
- SetValue() method, 514, 517**
- ShapeType property, 680**
- shared memory, DLLs and, 284, 286**
 - example code, 284-286
 - source code, 286-287
- ShareMem property, 258-259**
- sharing data across processes, 279-280**
 - CloseSharedData() method, 282, 284
 - CreateFileMapping() method, 284
 - GlobalData property, 280-282
 - MapViewOfFile() method, 283
 - OpenSharedData() method, 282
- shell extensions, 799**
 - COM object wizard, 801
 - debugging, 800-801
 - handlers, 800
 - context menu, 800, 808-818*
 - copy hook, 800-807*
 - drag-and-drop, 800*
 - drop target, 800*
 - icon, 800, 818-827*
 - InfoTip, 827-833*
 - property sheet, 800*
 - types, 800-833
- shell folders, 781**
- shell links**
 - creating, 784-785
 - information, getting and setting, 785-790
 - IShellLink interface, 780, 784-790
 - example code, 780*
 - instance creation, 781*
 - using, 781-783*
 - sample application, 790
 - main unit, 791-796*
 - supporting unit 1, 796-797*
 - supporting unit 2, 798-799*
- Shell Notiflycon() function, 748-750**
- shell programming**
 - AppBar, 764
 - API, 764-766*
 - TAppBar, 776-779*
 - VCL form encapsulation, 766-775*
 - shell extensions, 799
 - COM object wizard, 801*
 - debugging, 800-801*
 - handlers, 800-833*
 - initializing, 808-811*
 - types, 800-833*
 - shell links
 - creating, 784-785*
 - information, getting and setting, 785-790*
 - IShellLink interface, 780-790*
 - sample application, 790-799*
 - tray notification icon, 748, 762-764
 - API, 748-750*
 - hiding the application, 755-762*
 - hints surfacing, 752*
 - icon surfacing, 752*
 - message handling, 751*
 - mouse clicks, 752-754*
- ShellExecute() method, 470**
- ShellExecute() procedure, 218-219**
- SHGetSpecialFolderPath() method, 781-783**
- Short Message Service (SMS), 1118, 1121**
- ShortString string type, 51**
- ShowModal() method, 260**
- ShowWindow() procedure, 755**

SIDL (simplified IDL) file,
 generating, 971
signaled objects, 206-207
Simple Object Access Protocol.
 See **SOAP**
simple properties, 390, 435
simplified IDL. See **SIDL**
Single Instance option, 1055
**single option (Threading
 Model option)**, 1054
single thread, 657
Single threading choice, 1003
single-threaded apartment.
 See **STA**
single-threaded user interface,
 advantages, 182
sinks, multiple, events with,
 708-712
SizeOf() procedure, 50
smart linker, 251
SMS (Short Message Service),
 1118, 1121
**SOAP (Simple Object Access
 Protocol)**, 984
software design, 15
space allocation constants, 41
special folders, 781
speed, compilation speed,
 12-13
spinlock, Alpha-CPU data
 structure, 200
SQL tables, 333
**STA (single-threaded
 apartment)**, 658
standard procedures, 218
states, datasets, 314
static linking, 250-252
static methods, 108
stdcall calling convention, 163
stored procedures
 executing, 356-357
 results, displaying, 356
**streaming, component
 structure and**, 392
**streaming nonpublished
 component data**, 527-530
string operations, 54-55
string resources, 88
string types, Object Pascal, 51
 AnsiString, 52-57
 null-terminated, 61-63
 operations, 54-55
 ShortString, 58-59
 WideString, 60-61
 Win32 compatibility, 56-57
STRINGCONVERTLIB directive,
 290-292
StrPosProc() method, 279

**structured exception handling
 (SEH)**, 118-121
structured storage, 657
subscriber servers
 creating, 902-903
 registering, 903-904
support
 COM, 727-728
 interfaces, 731-732
 late binding, 730
 variant arrays, 729-730
 variants, 728-729
 WideString data type,
 730-731
 functions, arrays, variant, 73-75
synchronization
 message usage threads, 184
 multiple threads, 196-199
 critical sections, 199-203
 mutexes, 202-206
 semaphores, 207-210
 threads, 182-184
 VCL, 182
Synchronize() method, 182-184
syntax, packages, 102-103
syntax highlighting, 32
System DSN, 365
system scalability, 890

T

table data, retrieving, 355
tables, 301
Tadapter components,
 1078-1079
**TAdapterPageProducer
 component**, 1112-1114
TADOCommand component,
 372
TADOConnection component,
 368
 ConnectionString Property
 Editor, 368, 370
 Login prompt, bypassing, 370,
 372
 transaction processing, 375-376
**TADOConnection connection
 component**, 300
TADODataset component, 373
TADOQuery component, 301,
 375
TADOSorted component, 375
TADOTable component, 301,
 373-375
Tag property, 396
TAppBar component, 250,
 766-779
TAutoTest class, 679
TBevel control, 387
TBlobField descendant, 324-325
TBlobStream descendant,
 325-327
TButton component, 261-262
TCanvas class, 403, 568
TCanvas.Handle property, 568
TCE (tightly coupled events),
 898
TClientDataset component,
 1039
TCollection class, 544-545
TCollectionItem class, 544-545
 defining, 546
 editing components, 555
TComObject, 667-668
TComObjectFactory, 667-668
TComponent class, 395-397
**TComponentEditor
 component editor**, 523
TControl class, 397
TControl.Click() method, 446
TCopyHook object, 804
**TCORBAConnection client
 connection choice**, 1005
TCustom class, 400
**TCustomConnection
 component**, 299
TCustomControl property, 433
TCustomForm property,
 169-170
TDatabase component, 302
TDataSet component, 300,
 305, 315
TDataSet.State property, 314
**TDCOMConnection client
 connecting choice**, 1005
**TddgButtonEdit container
 components**
 design decisions, 477-478
 forms, adding, 485-488
 surfacing events, 478
 TddgDigitalClock, 481-485
 Text property, 478
 TSpeedButton control, 478-481
**TddgDefaultEditor component
 editor**, 611-612
**TddgRadioGroupEditor custom
 component editor**, 608-610
**TddgRunButton component
 example**, 470-475
TddgWaveFile component,
 531-532
TddGWorthless component,
 adding properties to, 435
 array properties, 441-443
 default array properties, 445
 default values, 444-445

- enumerated properties, 436
- object properties, 438-441
- set properties, 437
- simple properties, 435
- TddWaveFile component, 530**
- TDefaultEditor, 524**
- TDispatchConnection component, 1005, 1009**
- TDMA technology, 1119**
- templates, adding to WebSnap applications, 1111-1112**
- TerminateThread() method, 267**
- termination, TThread object, 180-182**
- testing**
 - components, 456-458
 - marquee components, 508-510
 - Web Services, 989-991
- testing conditions, 88-90**
- Text Connection (Data Link Properties dialog box), 368, 370**
- Text property, 478**
- TField object, 319-320**
- TField.FieldName property, 317**
- TForm1 class, 27**
- TFrameControl class, 565**
- TGraphicControl class, 399-400**
- TgraphicControl component, 387**
- THintWindow descendant**
 - creation, 490-492
 - deploying, 494
 - enabling, 494
- thread instances, 180**
- Thread Model, 1054-1055**
- Thread Neutral Apartment.**
See **TNA**
- Thread object item in New Items dialog box, 178**
- thread-local storage, 192**
 - example code, 194-196
 - TThread storage, 193
- threading choices**
 - Apartment, 1003
 - Both, 1004
 - Free, 1004
 - server setup, 1003-1004
 - Single, 1003
- threading models, list of, 657-658**
- threads**
 - explained, 174-176
 - fibers, 238-244
 - misuse of, 175-176
 - multiple, 175
 - BDE access, 227-233*
 - graphics, 233-238*
 - managing, 192-210*
 - non-UI VCL, 175*
 - sample application, 210-227*
 - UI VCL, 175*
- relative priority, 187
- resuming dynamically, 190
- suspending dynamically, 190
- synchronization, 182-184
- TerminateThread() method, 267
- timing, 190-192
- TThread object, 176-192
- threadvar clause, 193-196**
- THTTTPRIO component, 993-995**
- THTTTPSoapDispatcher component, 985**
- THTTTPSoapPascalInvoker component, 986**
- TIBDatabase connection component, 300**
- TIBQuery component, 301**
- TIBTable component, 301**
- tiers, 998
- tightly coupled events.** See **TCE**
- TImage control, 387**
- timing threads, 190-192**
- TinetXPageProducer, 1036**
- TInvoicableClass instance, 989**
- Tlabel component, 261-262**
- TListBox component example, 463-470**
- TListBox event, 276**
- TMask component, 261-262**
- TMemo component example, 459-463**
- TMessage record, 133-134**
- TMouseEvent, 392**
- TMsg record, 133-134**
- TNA (Thread Neutral Apartment), 907**
- TnotifyEvent property, 446**
- To Do List, viewing, 32**
- TObject class, 113, 394**
 - Create() method, 394
 - Destroy() method, 394
 - RTTI and, 405
 - TPersistent class, 395
- TOLEContainer class, sample application, 733-737**
 - child form, creating, 737
 - Clipboard, using to copy and paste, 739, 741-746
 - files, saving to and reading from, 738
 - OLE file, embedding or linking, 735
 - OLE object, embedding new, 733-734
- toolbars**
 - buttons, 20-21
 - Customize toolbar dialog box, 20-21
 - tooltips, 20
- Tools menu commands**
 - Enterprise Setup, 968
 - IDE Options, 971
- ToolsAPI unit, Open Tools API, 837**
- tooltips, 20, 827**
- Toplevel dialog component, 385**
- TPaintBox control, 387**
- TPersistent class, 395**
- TPropertyAttribute flags, 521-522**
- TpropertyEditor, 514**
- Tprovider component, 359-360**
- TQuery component, 301**
- Trace moniker parameter, 896**
- Tracelist property, 358-359**
- transaction processing, 375-376**
- Transactional Data Module Wizard, 912**
- transactions, application creation, 911**
- tray notification icon, 748**
 - API, 748-750
 - demo application, 762-764
 - hiding the application, 755-762
 - hints surfacing, 752
 - icon surfacing, 752
 - message handling, 751
 - mouse clicks, 752-754
- TrayWndProc() method, 751-754**
- Treelntf unit, Open Tools API, 837**
- Troll Tech, 565**
- try..except Exception Handling Block, 120**
- try..except..else Exception Handling Block, 120**
- TsetPropOptions property, 437**
- TShape control, 387**
- TSOAPConnection client connection choice, 1006**
- TSocketConnection, 1005, 1008**
- TSomeObject property, 438-440**
- TSQL Monitor component, 358-359**
- TSQLClientDataset component, 359-360**

TSQLConnection component, 351

- Connected property, 353
- ConnectionString, 352-353
- dbxconnections.ini configuration file, 351-352
- dbxdrivers.ini configuration file, 351-352
- LoginPrompt property, 353-354
- Params property, 354

TSQLConnection connection component, 300**TSQLDataset component, 354**

- CommandText property, 355
 - ctQuery* value, 355
 - result sets, extracting*, 356
 - table data, retrieving*, 355
- CommandType property, 355
 - ctQuery* value, 355
 - ctTable* value, 355
- ctStoredProc value, 356-357
- SetSchemaInfo procedure, 357-358

TSQLDataSet.ExecSQL() method, 355**TSQLQuery component, 301, 358****TSQLStoredProc component, 358****TSQLTable component, 301, 358****TStringConvert object, 289-290****TStrings Class, 400-403****TStringsList class, 402****TTable component, 301, 385**

- record searching, 333
 - FindKey()* method, 333
 - FindNearest()* method, 334
 - GotoKey()* method, 334
 - indexes, secondary*, 335
 - ranges*, 335-336
 - SetKey()* method, 333
 - SetKey..GotoNearest()* method, 334

TThread object, 176

- basic concepts, 178-179
 - instances*, 180
 - termination*, 180-182
- example code, 176-177
- synchronization, 184
- Synchronize()* method, 182-184

TThread storage, 193**TTimer component, 385, 498****TTimer object, 450****TTrayNotifyIcon component, 755-762****TWebAppComponents control, 1086****TWebConnection client connection choice, 1005****TwebModule, 985****TWidgetControl class, 398-399, 565**

- TwinControl class, 398, 566**
 - events, keyboard interaction, 399
 - methods, types of, 399
 - properties, types of, 398-399

TWindowInfo class, 276**two-tier applications, 1039-1041****TWSDLHTMLPublish component, 986****TWwidgetControl component, 386-387****type libraries, 675, 723-724****Type Library Editor, 1055-1059****typecasting, 87****typecasting expressions, 68-69****types**

- Object Pascal, 48-50
 - aliases*, 86
 - AnsiChar*, 50
 - arrays*, 76-78
 - case statement*, 89-90
 - Char*, 50
 - Currency*, 75
 - if statement*, 88-89
 - lifetime-managed*, 53-54
 - loops*, 90-92
 - null-terminated strings*, 61-63
 - objects*, 82-83
 - OleVariant*, 75
 - pointers*, 83-85
 - records*, 78-80
 - sets*, 80-82
 - ShortString*, 58-59
 - string resources*, 88
 - strings*, 51-57
 - typecasting, 87
 - user-defined*, 75
 - Variant*, 63-75
 - WideChar*, 50
 - WideString*, 60-61
- packages
 - design package*, 628
 - neither runtime and design package*, 628
 - runtime package*, 628
- typesafe programming, 404**

U**UDT (uniform data transfer), 657****UI (user interface), 29****UI VCL, 175****underscore character, 942****unidirectional datasets, 350****uniform data transfer. See UDT****union set, 81****unit source files, Win32 API, 253****units, 99-100**

- circular unit reference, 101
- initialization/finalization code, 100

Open Tools API, 836*ClxDesignWindows*, 838*ClxEditors*, 838*ClxSprigs*, 838*DesignConst*, 837*DesignEditors*, 837*DesignIntf*, 837*DesignMenus*, 837*DesignWindows*, 837*obsolete*, 838-839*PropertyCategories*, 837*ToolsAPI*, 837*TreeIntf*, 837*VCLEditors*, 838*VCLSprigs*, 837*VCSIntf*, 837

packages, 101-103

- portable code compatibility issues, 160

uses clause, 100-101

untyped pointers, 83**updates, multitable, 1028-1029****uploading**

- files, 1009-1011

- services, 1080

upWhereAll setting, 1018**upWhereChanged setting, 1018****upWhereKeyOnly setting, 1018****user defined types, 944****User DSN, 365****User List Service component, 1084****user tracking, 1080****user-defined messaging, 144**

- between applications, 145-146

- Broadcast() method, 146

- within applications, 144-145

user-defined types, 75**User32.dll, 564**

uses clause, 100-101

utilities, DCOMCNFG, 1042

V

Value parameter, 94

values, assigning to properties through RTTI, 426-428

VarArrayCreate() function, 72

VarArrayLock() function, 74

VarArrayOf() function, 72

VarArrayUnlock() procedure, 74

VarAsType() function, 75

VarCast() procedure, 75

VarClear() procedure, 75

VarCopy() procedure, 75

VarFromDateTime() function, 75

variable parameters, 95

variables, declaration, 39-41

variant arrays, 729-730

initializing, 73

support functions, 73-75

variant records, 66

Variant type, Object Pascal, 63

arrays, 71-75

changing, 64

data structure, 64-67

expressions, using in, 69-70

memory allocation, 67-68

typecasting expressions, 68-69

VType values, 70-71

variants, 163, 728-729

VarIsEmpty() function, 75

VarIsNull() function, 75

VarToDateTime() function, 75

VarToStr() function, 75

VarType() function, 75

VCL (Visual Component

Library) component building

Component Expert, 433

component units, 433-435

components

icons, 458-459

ProcessExecute() method,

475-476

registering, 454-455

SetCommandLine() method,

476-477

TddgButtonEdit container

component, 477-488

TddgRunButton component

example, 470-475

testing, 456-458

TListBox component

example, 463-470

TMemo component

example, 459-463

constructors

Component State values, 453

design-time behavior, 453

overriding, 452

custom control, writing

custom controls, types of,

432-433

decisions regarding, 430-431

writing steps, 431-432

destructors, overriding, 454

events

event handler, 445

event property, 445

event-dispatching method,

445-446

properties, defined, 446-450

hierarchies, 384

methods, 451

overview, 430

properties, adding, 435

array properties, 441-443

default array properties,

445

default values, 444-445

enumerated properties, 436

object properties, 438-441

set properties, 437

simple properties, 435

TCustomControl property, 433

VCL synchronization, 182

VCLEditors unit, Open Tools

API, 838

VCLSprigs unit, Open Tools

API, 837

VCSIntf unit, Open Tools API,

837

versioning packages, 635

VFI (visual form inheritance), 12

View menu commands

New Edit Window, 23

Object Treeview, 1103

Project Manager, 32

To Do List, 32

Virtual Method Table. See VMT

virtual methods, 108

visibility specifiers, 111-112

Visual Component Library.

See **VCL**

visual components, 385-386

TGraphicControl, 387

TWidgetControl,

characteristics, 386

Handle property, 386-387

TwinControl

characteristics of, 386

Handle property, 386-387

visual development

environment, 11-12

visual editing, 657

visual form inheritance. See

VFI

VisualCLX, 564-565

defined, 383

portable code compatibility

and, 161

VMT (Virtual Method Table)

ConvertString() method, 289

overview, 658

vtables, 658

VType values, Variant type,

70-71

W-Z

WAP (Wireless Application Protocol), 1118

advantages of, 1122

architecture, 1121-1122

bitmaps, 1128-1131

disadvantages of, 1123

error reporting, 1127

sample application, 1125-1127

WML (wireless markup

language, 1123-1124

WTLS (Wireless Transport

Layer Security), 1124

Wave File component, 533-537

Web App Components dialog

box, 1083-1084

Web Debugger, accessing, 1082

Web Services

CORBA and

CORBA client code,

adding, 978-981

creating, 975-976

example architecture, 975

SOAP client, creating,

977-978

invoking from client, 991-993

import unit, generating for

remote invokable object,

993-994

using THTTTPRIO

component, 994-995

overview, 984

SOAP (Simple Object Access

Protocol), 984

writing, 985

invokable interface, defining,

986-987

invokable interface,

implementing, 987-989

testing, 989-991

TWebModule, 985-986

WSDL (Web Services

Description Language), 984

Web sites

Borland, 965
 DCOM page, 1043
 DoCoMo, 1132
 OMG (Object Management Group), 938

WebAppComponents component, 1086**WebBroker technology, 564, 1034****WebSnap applications, 1034, 1078**

designing
ApplicationTitle property, 1089
component choices, 1083-1084
 converting to ISAPI DLL, 1107
custom components, 1112-1114
custom templates, 1111-1112
 data, displaying, 1103-1105, 1107
 file uploading, 1109-1111
 image handling, 1101-1103
LocateFileServices, 1108-1109
 logging in, 1092-1094
 naming, 1085-1088
 navigation menu bar, 1089-1092
preference data between sessions, 1099-1101
server choices, 1081-1083
user preference data, managing, 1095-1098
Web App Debugger option, 1083
WebSnap toolbar, adding to IDE window, 1080

features
dispatching methods, 1079
file uploading services, 1080
 HTML management, 1080
login services, 1079-1080
page producer components, 1079
server-side scripting, 1078
session management, 1079
Tadapter components, 1078-1079
user tracking, 1080
Web modules, multiple, 1078

while loop, 91-92**WideString data type, 730-731****WideString string type, 51, 60-61****widgets, user interface, 565****Win32**

API, unit source files, 253
 compatibility, 56-57
 handles, 387
 preemptive multitasking, 174

Windows, shell programming

AppBars, 764-779
 shell extensions, 799-833
 shell links, 780-799
 tray notification icon, 748-764

Windows 2000, Active Server Objects, debugging, 1074-1076**Windows Messages. See WM****Windows NT 4, Active Server Objects, debugging, 1073-1074****Windows packages, CLX, 613-614****WinHelp, accessing, 22****Wireless Application Protocol (WAP), 1118**

advantages of, 1122
 architecture, 1121-1122
 bitmaps, 1128-1131
 disadvantages of, 1123
 error reporting, 1127
 sample application, 1125-1127
 WML (wireless markup language), 1123-1124
 WTLS (Wireless Transport Layer Security), 1124

wireless development, 1116

history of
 1980s-late, 1117
 1980s-pre, 1116
 1990s-early, 1117
 1990s-late, 1117
 2000s, 1117
 mobile wireless devices,
mobile phones, 1118
PalmOs devices, 1118
Pocket PCs, 1119
RIM BlackBerry, 1119
 overview, 1116
 radio technologies
 3G (third generation), 1120
 802.11, 1120-1121
 Bluetooth, 1120
 CDMA, 1119

CDPD (Cellular Digital

Packet Data, 1119
GPRS (General Packet Radio Service), 1120
GSM, 1119
TDMA, 1119
 server-based, 1121
I-mode, 1132
PQA (Palm Query Applications), 1132-1135
SMS (Short Message Service), 1121
WAP (Wireless Application Protocol), 1121-1131
 user experience, 1136
circuit switched versus *packet switched* networks, 1137
form factors, 1137
m-commerce, 1138
misconceptions, 1137
navigation techniques, 1137

wireless markup language (WML), 1123-1124**wizard initialization method, 847****wizards**

Automation Object, 912-913
 COM object, 801
 form, steps to create, 868-869
 Transactional Data Module, 912
 writing DDG Search
main form, bringing up files in the IDE Code Editor, 859,860
main form, complete, 860-867
Open Tools API, 855-867
project file, 858-859
 wizard logic, 855-857
 writing DLL-based
 adding and modifying
 wizard entries in registry, 852-853
main project file, 854
main unit, 848-852
Open Tools API, 843-854
 wizard class, 845-847
 writing simple, 839-842
 writing without packages, 855

WM (Windows Message), 131-132**WM CTLCOLOR message, 139****WM KILLFOCUS message, 138**

WM SYSCOMMAND, 139

WML (wireless markup language), 1123-1124

WndProc() method, 767-768

wParam field, 131

Write method, 1059

writing

components

Component Expert, 433

component units, 433

*custom controls, types of,
432-433*

*decisions regarding,
430-431*

writing steps, 431-432

property editors, 511

*attribute specification,
521-522*

*descendant property editor
object creation, 511-513*

*editing the property as text,
513-517*

editing with dialog, 519-521

registering, 517-519

registering, 522

Web Services, 985

*invokable interface, defining,
986-987*

*invokable interface,
implementing, 987-989*

testing, 989-991

TWebModule, 985-986

WTLS (Wireless Transport Layer Security), 1124