

Contents

Foreword	v
Preface	vii

Part 1. Client/Server With Distributed Objects 1



Chapter 1. Objects: The Next Client/Server Revolution 3

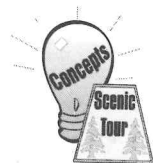
Intergalactic Client/Server Computing	4
What Client/Server Software Is Needed?	5
Who Will Lead the Next Client/Server Revolution?	6
Client/Server With SQL Databases	7
Client/Server With TP Monitors	9
Client/Server With Groupware	12
Client/Server With Distributed Objects	15



Chapter 2. From Distributed Objects To Smart Components 21

Objects and Distributed Objects	22
Distributed Objects	23
Object Magic	23
Encapsulation	24
Components: The Grand Prize of Objects	28
The Driving Force Behind Components	29
Components to the Rescue	30
When Can We Expect These Components?	32
So, What Exactly Is a Component?	34
So, What Is a Supercomponent?	36
Business Objects: The Ultimate Components	38
Your Guide to the Symbols Used in This Survival Guide	39
The Component Road Map	42

Part 2. CORBA: The Distributed Object Bus 43



Chapter 3. CORBA: A Bird's Eye View 47

Distributed Objects, CORBA Style 49

 What Is a CORBA Distributed Object? 49

 Everything Is in IDL 50

 CORBA Components: From System Objects To Business Objects 52

 OMG's Object Management Architecture 53

 The Object Request Broker (ORB) 54

 Object Services 57

 Object Services: Build-to-Order Middleware. 59

 Object Services: The Roll-Out Road Map. 60

 Common Facilities. 61

 Application/Business Objects 62



Chapter 4. CORBA 2.0: The Intergalactic Object Bus 67

What Exactly Is a CORBA 2.0 ORB? 68

 The Anatomy of a CORBA 2.0 ORB 69

 CORBA Method Invocations: Static Versus Dynamic. 72

 CORBA Static Method Invocations: From IDL to Interface Stubs. 74

 CORBA Dynamic Method Invocations: A Step-By-Step Guide. 76

 The Server Side of CORBA 78

 What's an Object Adapter? 78

 BOA and Other Object Adapters 80

 BOA Shared Server 81

 BOA Unshared Server 81

 BOA Server-per-Method 82

 BOA Persistent Server. 83

 CORBA 2.0 Initialization—Or How Does a Component Find Its ORB? 83

 CORBA 2.0: The Intergalactic ORB. 85

 CORBA 2.0: The Inter-ORB Architecture 85

 CORBA 2.0: ORB-to-ORB Bridging. 87

 Federated ORBs 88



Chapter 5. CORBA Metadata: IDL and Interface Repository 91

The CORBA IDL: A Closer Look 93

 What Does an IDL Contract Cover? 93

The Structure of the CORBA IDL 94

An IDL Example 96

Type Codes: CORBA's Self-Describing Data 97

The CORBA 2.0 Interface Repository 98

 What's an Interface Repository? 98

 Why Is an Interface Repository Needed Anyway? 99

 Interface Repository Classes: The Containment Hierarchy 99

 The Interface Repository Class Hierarchy 100

 Federated Interface Repositories 104

 What Does a Global Repository ID Look Like? 105

 Conclusion. 107



Chapter 6. CORBA Services: Naming, Events, and Life Cycle . . . 109

The CORBA Object Naming Service 110

 What's in a CORBA Object Name? 111

 How Does It Work? 112

The CORBA Object Trader Service. 113

The CORBA Object Life Cycle Service 114

 A Compound Life Cycle Example. 115

 The Life Cycle Interfaces 115

 The Compound Life Cycle Interfaces 117

The CORBA Event Service 119

 Suppliers and Consumers of Events. 119

 The Event Channel. 121

 Conclusion. 121



Chapter 7. CORBA Services: Transactions and Concurrency . . . 123

The CORBA Object Transaction Service 124

 What Is a Transaction? 124

 Object Transaction Service: Features. 127

 The Elements of the Object Transaction Service 128

 The OTS Interfaces 129

 An Object Transaction Scenario 132

The CORBA Concurrency Control Service 134

 The Concurrency Control Service and Transactions 134

 Locks 135

 Locksets 135

 Nested Transactions and Locking 135

 The Concurrency Control Interfaces. 136

 Conclusion. 137



Chapter 8. CORBA Services: Persistence and Object Databases . 139

The CORBA Persistent Object Service (POS) 140

 What Is POS? 140

 Single-Level Stores Versus Two-Level Stores 142

 POS: The Client's View 142

 POS: The Persistent Object's View 143

 The Elements of POS 144

 POS Protocols: The Object-PDS Conspiracy 146

 The POS Interfaces 149

 The POS CLI Interfaces 150

Object Database Management Systems 151

 What's an ODBMS? 152

 What's an ODBMS Good For? 154

 ODBMS Client/Server Implementations 157

 ODMG-93: The Lingua Franca for ODBMS 159



Chapter 9. CORBA Services: Query and Relationships. 165

The CORBA Query Service 165

 Federated Queries 166

 Collections for Manipulating Query Results 167

 Query Service: The Collection Interfaces 167

 Query Service: The Query Interfaces 168

 A Simple Query Scenario 169

 A More Complex Query Scenario 171

The CORBA Collection Service 172

The CORBA Relationship Service 173

 Why a Relationship Service? 173

 What Exactly Is a Relationship? 174

 Levels of Relationship Service 176

 Relationship Service: The Base Interfaces 176

 Relationship Service: Graphs of Related Objects 178

 Relationship Service: The Containment and Reference Relationships 179

 Conclusion 181



Chapter 10. CORBA Services: System Management and Security 183

The CORBA Externalization Service 184

 Stream Power 184

 Externalization Service: The Base Interfaces 185

A Stream Scenario 186

The CORBA Object Licensing Service 188

 What Does the Licensing Service Do? 189

 Licensing Service Interfaces 189

 A Licensing Scenario 191

The CORBA Object Property Service 192

 Property Service Interfaces 192

The CORBA Object Time Service 194

The CORBA Object Security Service 195

 What Is ORB-Based Security? 198

 Authentication: Are You Who You Claim to Be? 198

 Authorization: Are You Allowed to Use This Resource? 199

 Audit Trails: Where Have You Been? 200

 Non-Repudiation: Was This Message Tampered With? 200

 Other Security Issues 200

The CORBA Object Change Management Service 201

 Conclusion 202



Chapter 11. CORBA Commercial ORBs 203

Iona's Orbix 204

 Orbix Architecture 205

 Orbix 2.0's Current CORBA Support 205

 Orbix's CORBA Support in Early 1996 205

 Orbix's CORBA Extensions 206

Digital's ObjectBroker 206

 ObjectBroker Architecture 207

 ObjectBroker 2.5's Current CORBA Support 207

 ObjectBroker's CORBA Support in 1996 207

IBM's SOM 208

 SOM Architecture 208

 SOM's Object Model Extensions and Other Features 209

 SOM 2.1's Current CORBA Support 209

 SOM's CORBA Support in 1996 210

Expersoft's XShell 210

 XShell 3.5's Current CORBA Support 211

 XShell (or PowerBroker): CORBA Support in 1996 211

Sun's Distributed Objects Everywhere (DOE) 211

 Sun's Current CORBA Support 212

 Sun's CORBA Support in 1996 212

HP's ORB Plus 212

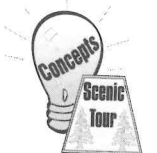
HP's CORBA Support in 1996..... 212
 Who Did We Leave Out? 213
 Conclusion and Some Parting Comments 213

Part 3. Frameworks for Business Objects and Components 217



Chapter 12. Object Frameworks: An Overview..... 221

Frameworks Overview 222
 What Are Object Frameworks?..... 223
 How Frameworks Work 224
 Key Benefits of Object Frameworks 226
 Frameworks, APIs, or Class Libraries?..... 227
 Procedural Frameworks: Look Ma, No Objects 229
 Frameworks for Business Objects and Components 231
 Meet the Players 231
 Types of Frameworks 231
 Frameworks and Design Patterns 233
 Quiz: Framework or White Box Component? 235
 Class Libraries, Frameworks, and Components 236
 Conclusion..... 237



Chapter 13. CORBA's Frameworks and Business Objects 239

Business Objects 240
 So What Is a Business Object Anyway?..... 240
 Cooperating Business Objects 242
 The Anatomy of a Business Object 243
 The Anatomy of a Client/Server Business Object..... 245
 CORBA's Common Facilities..... 246
 What Are CORBA Common Facilities?..... 247
 The CORBA User Interface Common Facility..... 248
 The CORBA Information Management Common Facility..... 249
 The CORBA System Management Common Facility..... 250
 The CORBA Task Management Common Facility 255
 The CORBA Vertical Market Common Facilities..... 258
 When Can You Expect These Common Facilities?..... 259
 Conclusion..... 260

Chapter 14. Compound Documents: The Desktop Framework ... 261



Compound Documents: Why All the Fuss?..... 262
 The Borderless Desktop 262
 Documents Come in All Shapes 262
 A Home for All Data Types 263
 In-Place Editing 264
 Shippable Documents..... 264
 The Compound Document Framework 266
 Document Layout 267
 Structured Storage 267
 Scripting and Automation 268
 Uniform Data Transfer 269
 Conclusion..... 269



Chapter 15. The OpenDoc Component Model 271

Parts: Components, OpenDoc Style 272
 OpenDoc Meets CORBA..... 272
 OpenDoc's Constituent Technologies 272
 SOM 273
 Bento..... 274
 Uniform Data Transfer 274
 Compound Document Management..... 275
 Open Scripting Architecture 275
 OpenDoc: Who Ships What?..... 277
 ComponentGlue: OpenDoc Becomes OLE and Vice Versa..... 278
 What OpenDoc Does for Client/Server Systems 279
 Client/Server, OpenDoc Style 279
 How OpenDoc Enhances CORBA..... 280
 Conclusion..... 281



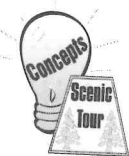
Chapter 16. OLE/COM: The Other Component Standard 283

OLE for the Enterprise 284
 What Is OLE?..... 284
 OLE: A Short History 285
 OLE Is OLE 285
 The Vision: The Framework Is Windows 286
 OLE: Interfaces Everywhere..... 286
 So, What Is an OLE Component? 287



HP's CORBA Support in 1996	212
Who Did We Leave Out?	213
Conclusion and Some Parting Comments	213

Part 3. Frameworks for Business Objects and Components 217



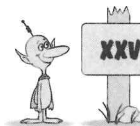
Chapter 12. Object Frameworks: An Overview 221

Frameworks Overview	222
What Are Object Frameworks?	223
How Frameworks Work	224
Key Benefits of Object Frameworks	226
Frameworks, APIs, or Class Libraries?	227
Procedural Frameworks: Look Ma, No Objects	229
Frameworks for Business Objects and Components	231
Meet the Players	231
Types of Frameworks	231
Frameworks and Design Patterns	233
Quiz: Framework or White Box Component?	235
Class Libraries, Frameworks, and Components	236
Conclusion	237



Chapter 13. CORBA's Frameworks and Business Objects 239

Business Objects	240
So What Is a Business Object Anyway?	240
Cooperating Business Objects	242
The Anatomy of a Business Object	243
The Anatomy of a Client/Server Business Object	245
CORBA's Common Facilities	246
What Are CORBA Common Facilities?	247
The CORBA User Interface Common Facility	248
The CORBA Information Management Common Facility	249
The CORBA System Management Common Facility	250
The CORBA Task Management Common Facility	255
The CORBA Vertical Market Common Facilities	258
When Can You Expect These Common Facilities?	259
Conclusion	260



Chapter 14. Compound Documents: The Desktop Framework ... 261



Compound Documents: Why All the Fuss?	262
The Borderless Desktop	262
Documents Come in All Shapes	262
A Home for All Data Types	263
In-Place Editing	264
Shippable Documents	264
The Compound Document Framework	266
Document Layout	267
Structured Storage	267
Scripting and Automation	268
Uniform Data Transfer	269
Conclusion	269



Chapter 15. The OpenDoc Component Model 271

Parts: Components, OpenDoc Style	272
OpenDoc Meets CORBA	272
OpenDoc's Constituent Technologies	272
SOM	273
Bento	274
Uniform Data Transfer	274
Compound Document Management	275
Open Scripting Architecture	275
OpenDoc: Who Ships What?	277
ComponentGlue: OpenDoc Becomes OLE and Vice Versa	278
What OpenDoc Does for Client/Server Systems	279
Client/Server, OpenDoc Style	279
How OpenDoc Enhances CORBA	280
Conclusion	281



Chapter 16. OLE/COM: The Other Component Standard 283

OLE for the Enterprise	284
What Is OLE?	284
OLE: A Short History	285
OLE Is OLE	285
The Vision: The Framework Is Windows	286
OLE: Interfaces Everywhere	286
So, What Is an OLE Component?	287

- OLE's Constituent Technologies 288
 - The Component Object Model (COM) 289
 - OLE's Automation and Scripting Services 290
 - OLE's Uniform Data Transfer 291
 - OLE's Structured Storage and Persistence Services 291
 - OLE's Compound Document Service 292
- OLE Tools and Foundation Classes 293
 - Microsoft Foundation Classes (MFC) 293
 - Microsoft Visual C++ 294
 - Conclusion 294



Chapter 17. Taligent's CommonPoint: Frameworks Everywhere. . 297

- Integrated Frameworks 298
 - So What's an Application System Anyway? 299
 - Frameworks Everywhere 300
- People, Places, and Things 302
 - How the New Metaphor Works 302
 - Places 303
 - People 305
 - Things 306
- Compound Documents, CommonPoint Style 307
 - Shared Documents 307
 - The Taligent Compound Document Model 308
 - Taligent/OpenDoc Interoperability 308
 - Conclusion 312



Chapter 18. NeXT's OpenStep and Portable Distributed Objects . 313

- Who Is Doing What With OpenStep? 314
 - OpenStep in the Unix World 314
 - OpenStep in the Windows World 314
- What Is OpenStep? 315
 - PDO and Objective C 315
 - The Application Framework 319
 - Display PostScript 319
 - Application Development Tools 319
 - The Enterprise Objects Framework 320

Chapter 19. Newi: Cooperative Business Objects 325



- Newi's Business Objects 326
 - What Is a CBO? 327
 - CBOs Versus Compound Document Components 327
- The Newi ORB and Business Object Framework 329
 - The Newi ORB: Message-Time Binding 329
 - Newi's Dynamic Object Hierarchies 331
 - Newi's Model/View Separation 332
 - View Layout Files 334
 - Other Newi Frameworks and Tools 335
 - Newi, OpenDoc, and SOM/CORBA 335
 - Conclusion 337

Part 4. OpenDoc Under the Hood 339



Chapter 20. OpenDoc and SOM: The Object Model 343

- What SOM Brings to the Party 344
 - OpenDoc and SOM 344
 - SOM: A Technology for Packaging Parts 345
- The OpenDoc Programming Model 346
 - ODPart: The Part Editor Class 346
 - The OpenDoc Class Hierarchy 347
 - How Part Editors Find OpenDoc Objects 350
 - How Intrusive Is the OpenDoc Programming Model? 351
 - OpenDoc Frameworks and Tools 352



Chapter 21. OpenDoc: The Compound Document Model 357

- Binding: Creating the OpenDoc Environment 358
 - The Document Shell: OpenDoc's Run-Time Manager 359
 - Factories and Reference-Counted Objects 359
 - Binding: How Part Data Finds Part Editor 360
- Layout and Geometry Management 361
 - OpenDoc Container Parts 361
 - Frames, Facets, and Canvases 362
 - Drawing, OpenDoc Style 365
 - Views: A Part Is a Part 366

Part Windows	366
Controls	367
Event Distribution and Arbitration	368
Arbitration With Focus Sets	368
Inside-Out Part Activation	369
Sharing Menus	370



Chapter 22. OpenDoc: Bento and Storage Units 373

The Bento Storage System	375
The Storage Containment Hierarchy	376
OpenDoc Storage Units	377
Properties With Multiple Values	377
Just What Is a Storage Unit Anyway?	377
Navigating Through Storage Units	378
Persistent References and Cloning	380
The Persistent Elements	382
Standard Properties	382
A Day in the Life of a Storage Unit	383
Conclusion	385



Chapter 23. OpenDoc: Uniform Data Transfer 387

Drag-and-Drop Transfers	389
Clipboard Transfers	391
Linked Data Transfers	394
OpenDoc's Link-Related Objects	394
Advertising a Link	396
Scenario: Data Interchange Via Links	396
Conclusion	398



Chapter 24. OpenDoc: Automation and Semantic Events 399

The Brave New World of OpenDoc Scripting	401
New Age Scripting	401
Roaming Agents	402
Disposable Applications	405
Client/Server Component Suites	406
The OpenDoc Extension Mechanism	407
Why Are Extensions Needed?	407

How Extensions Work	408
Semantic Events	410
What's a Semantic Event?	410
An Example	411
Object Specifiers Unleashed	411
How Semantic Events Work	412
Setting Up the Environment	415
Scenario: A Semantic Message At Work	416
A Quick Review	418
Scripting and Automation	418
Beyond Just Scriptability	420
Scriptability	420
Tinkerability	421
Recordability	421
Scripting Systems	423
Conclusion	424

Part 5. OLE/COM Under the Hood 425



Chapter 25. COM: OLE's Object Bus 429

COM: The Object Bus	431
Looking at COM Through CORBA Eyes	431
COM Style Interfaces	432
So, What's a COM Object?	433
What's a COM Server?	436
Server Flavors: In-Process and Out-Of-Process	437
Creating Custom Interfaces	439
COM Object Services	441
The Ubiquitous IUnknown Interface	442
Interface Negotiations Using QueryInterface	442
Life Cycle Management With Reference Counts	444
IClassFactory2: Object Creation and Licensing	445
A COM Object Creation Scenario	446
Connectable Objects: COM's Event Service	448
COM Style Inheritance: Aggregation and Containment	449



Chapter 26. OLE: Automation, Scripting, and Type Libraries . . . 453

Automation, OLE Style . . . 454
How the Pieces Work Together. . . 455
What's an OLE Automation Server? . . . 455
OLE Guidelines for Automation Objects . . . 456
The Structure of an OLE Automation Server . . . 457
Collection Objects. . . 458
Automation Controllers: Visual Basic and DispTest . . . 460
Building OLE Automation Servers . . . 461
What's a Dispinterface? . . . 461
The IDispatch Interface . . . 462
What Does an IDispatch Client Do? . . . 463
How OLE Creates and Manages Type Information . . . 464
The Object Description Language . . . 465
Building Type Libraries: The Hard Way . . . 467
Building Type Libraries: The Easy Way . . . 468
Registering a Type Library . . . 469
Finding and Loading a Type Library . . . 470
So, How Do I Get Information Out of a Type Library? . . . 470
A Type Library Navigation Scenario. . . 471
Getting Information From IProvideClassInfo . . . 472
Conclusion. . . 473



Chapter 27. OLE: Uniform Data Transfer . . . 475

OLE's Data Transfer Model. . . 476
Data Transfer: Formats and Structures. . . 477
The IDataObject Interface. . . 477
Clipboard Transfers, OLE Style. . . 479
Delayed Transfers . . . 479
A Clipboard Data Transfer Scenario. . . 479
Drag-and-Drop Transfers . . . 481
Drag-and-Drop: Who Does What? . . . 481
A Drag-and-Drop Data Transfer Scenario . . . 482
Linked Data Transfers . . . 484
What's an Advise Sink? . . . 485
A Parting Scenario . . . 486
Conclusion . . . 487



Chapter 28. OLE: Structured Storage and Monikers . . . 489

OLE's Structured Storage: Compound Files . . . 490
The Structure of a Compound File. . . 491
The IStorage Interface . . . 492
Transactional Storage . . . 493
The IStream Interface . . . 494
A Storage Access Scenario . . . 495
Persistent Objects . . . 497
How Persistent Is Your Object? . . . 497
The IPersist Interfaces. . . 498
A Persistent Object Scenario. . . 499
Monikers: Persistent, Intelligent Names . . . 501
So, What Exactly Is a Moniker? . . . 501
Monikers As Persistent Objects . . . 502
The IMoniker Interface. . . 502
Types of Monikers. . . 503



Chapter 29. OLE: Compound Documents and OCXs. . . 507

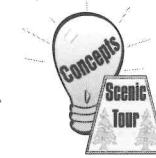
The OLE Compound Document Model . . . 508
Will the Real Container/Server Please Stand Up? . . . 509
Linking versus Embedding. . . 509
The Minimalist Container/Server . . . 511
The Minimalist Container . . . 512
The Minimalist Server . . . 513
A Container/Server Scenario. . . 515
The Maximalist Container/Server . . . 516
Linking and Embedding Interfaces. . . 518
So, How Does Linking Work? . . . 520
In-Place Editing . . . 520
Miscellaneous Interfaces . . . 521
OLE Custom Controls (OCXs). . . 523
So What's an OCX? . . . 524
Container Meets OCX . . . 524
How OCXs and Containers Use Automation . . . 527
OCX-Generated Events. . . 527
Connectable Objects: COM's Event Service. . . 528
An OCX Connectable Object Scenario . . . 530
Conclusion. . . 532

Part 6. Component Nirvana: Client/Server With Parts . . . 533



Chapter 30. Which Component Model? 535

Should You Bet on COM/OLE or CORBA/OpenDoc? 536
OLE/COM Versus CORBA/OpenDoc: The Object Models 536
OLE/COM Versus CORBA/OpenDoc: The Object Buses 538
OLE/COM Versus CORBA/OpenDoc: Higher-Level Language Bindings 539
OLE/COM Versus CORBA/OpenDoc: The System Services 540
OLE/COM Versus CORBA/OpenDoc: Compound Document Frameworks 541
So, the Winner Is... 543
COM/OLE and CORBA/OpenDoc: Can We Interoperate? 543
Gateways Come in All Shapes 543
ComponentGlue: The Deep Gateway Approach 544
The COM/CORBA Interworking RFP: The Generic Gateway Approach 546
Conclusion 549



Chapter 31. Client/Server With Component Suites 551

Client/Server With Component Suites 552
Suites: The Sum Is More Than the Total of the Parts 552
Suites of Pluggable Places 552
Suites of Client/Server Parts 553
Cafeteria-Style Suites 553
The Component Market Infrastructure 554
The Open Component Market Vision 554
Getting Ready for Client/Server With Components 559
It's Time to Say Farewell 561

Where to Go for More Information 563

Trademarks 573

Index 575