INDEX

A
Aborted transactions, 6–7, 41
Abstract component/service, 50–51
ACID properties, xxvii–xxviii, 6, 15, 27, 66, 82, 256, 257–258
ACID transactions, 281–285
Active replication, 52–53
Active state, business processes, 342
ActiveStatus, 352
Activities, 58
Activity coordinator, 7, 18, 21–22, 288, 292, 294–295
ADAPTS (OTSPolicy), 108–109
Additional Structuring Mechanisms for the OTS, version 1.C, 364
Advanced transaction concepts, 259–280
afterCompletion() method:
  synchronization event, 90
  Synchronization interface, 79
Application server embedded implementations, 374
Architecture, 86–94
  failure recovery, 86–88
  transaction interoperability, 89–91
  tree of process model, 88–89
Arjuna Technologies, 187
  Transaction Service, 8, 33, 148, 260, 277, 373–374
Aspect Oriented Programming (AOP) techniques, 209
Asynchronous invocations, 45–46
Atomic operations:
  defined, 4
Atomic programs with respect to concurrency, 10
Atomic transactions, 324–328
  defined, 4
  properties of, 6
  top-level, 7
Atomicity, 6, 7–8
AUTO__ACKNOWLEDGE mode, 190–191

B
Basic language, 260
BEA, 309
  and OMG standard, 96
  Tuxedo, 65, 260, 374, 379
  WebLogic Server v 6.1, 200
Bean-managed persistence (BMP) model, 209
Bean-managed transactions (BMTs), 215–217, 229–230
  acknowledgment of, 230
  CMTs compared to, 217
  UserTransaction interface, 229–230
beforeCompletion() method, 90–91
  Synchronization interface, 79
Begin command, transaction API, 41
begin method, Current interface, 116
begin() method, TransactionManager interface, 72
Billing and accounting resource usage, 283
Blocked transactions, 13
BTP Committee Specification, 364
Bulletin board, 283
Business domains, 340–341, 343
Business Process Execution Language for Web Services Version 1.1, 364
Business process interposition: example of, 276, 344
Business process state transitions, 343
Business Transactions Protocol (BTP), 309–321
atoms, 310–311
begin message, 312
cohesion, 310–311
composers, 313–314
open-top coordination, 309–310
optimizations, 315–317
autonomous decision by a participant, 316–317
one-shot, 315–316
resignation by a participant, 316
spontaneous prepare, 316
participants, 313
qualifiers, 317
roles in, 314–315
using, 317–321
XML message sets and carrier bindings, 311–313
BusinessAgreement protocol, 329–330
BusinessAgreementWithComplete protocol, 330
Byzantine outcomes, 82

C
Caching of an object’s state, 19
Call Level Interfaces (CLI), 67
CancelCompletionStatus, 351
Cancelled state, business processes, 342–343
CancelOnlyCompletionStatus, 351
Cascade roll back problem, 11
CCI, See Common Client Interface (CCI)
Checked transactions, 43–47
behavior, 131–132
CICS (IBM), 260, 373, 379
clearBatch() method, 166
CLIENT_ACKNOWLEDGE mode, 191
Closed-top commit protocol, 310
Clustering, 56
Cohesion composer, 311, 314
Commercial application servers:
availability measures in, 55–57
Commercial transaction system implementations,
373–374
application server embedded implementations, 374
stand-alone implementations, 373–374
Commercial transaction systems:
arbitrary participants, support for, 370
cost, 372
development background, 368
distributed transactions, 371
fault tolerance, 369
management support, 371
multi-threading, support for, 372
nested transactions, support for, 370
openness, 372
open-source vs., 367–368
pedigree, 371
performance characteristics of, 368–369
quality assurance, 370
reliability characteristics, 369–370
summary, 372
two-phase commit protocol, 369
Web services transactions, support for, 371
Commit, 7
two-phase, 7–8
Commit check, 132
Commit command, transaction API, 41
commit method:
 Current interface, 116
 Resource interface, 121
Index

commit() method, TransactionManager interface, 72–73
Commit options, 226
commit_one_phase method Resource interface, 121
Committed transactions, 6–7
Common Application Environment (CAE), 64
Common Client Interface (CCI), 240–243
javax.resource.cci.Connection, 240–241
javax.resource.cci.ConnectionFactory, 241
javax.resource.cci.Interaction, 241–242
javax.resource.cci.InteractionSpec, 242
javax.resource.cci.Record, 241
Common Facilities, 96
Common Object Request Broker Architecture (CORBA), See CORBA
Common Secure Interoperability, version 2 (CSIv2), 232
Communications Resource Manager (CRM), 89
Compensator, 338–339
Compiled versus interpreted applications, 260–261
CompletedCancelStatus, 352
CompletedConfirmStatus, 352
CompletingCancelStatus, 352
CompletingConfirmStatus, 352
Completion states, 351
CompletionStatus state diagram, 351
Concurrency control:
  optimistic vs. pessimistic, 11–12
  two-phase, 10–11
  type-specific, 12–13
Concurrent transactions, 35
ConfirmCompletionStatus, 351, 353
Confirm-set, 311
Connection interface, 240–241
ConnectionEventListener interface, 248
ConnectionFactory interface, 241, 245
ConnectionManager class, 245–246
Connectivity, and interposition, 40
Consensus group, 306–307
Consistency, 6, 9, 15, 26
  strong, 52
Container-managed persistence (CMP) model, 209–210, 223
Container-managed transactions (CMTs), 230–231
  BMTs compared to, 217
  client side container behavior, 233
  CosTransactions::ADAPTS, 233
  CosTransactions::Coordinator, 233
  CosTransactions::EITHER, 233
  CosTransactions::InvocationPolicy, 233
  CosTransactions::OTSPolicy, 233
  CosTransactions::PropagationContext, 232–233
  CosTransactions::SHARED, 233
  CosTransactions::Terminator, 233
  Mandatory attribute, 213
  Never attribute, 213
  NotSupported transaction attribute, 211–212, 230
  Required transaction attribute, 212, 230–231
  RequiresNew attribute, 212–213
  server side container behavior, 232–233
  specifying, 214–215
  Supports attribute, 212
  transaction interoperability, 233–234
Control interface, 111
CoordinationInformation interface, 294–295
Coordinator, 7, 18, 21–22, 288, 292, 294–295
Coordinator interface, 113–115
Coordinator-to-application heuristic interaction, 21–22
CORBA, 2, 185, 208, 231–232, 271–272, 301, 381
  IIOP Service Context Information, 37
Object Transaction Service, 67, 92
  specification, 22
Portable Interceptors, 41
create_subtransaction:
    Coordinator interface, 115
Current interface, 115–118
Customer Relationship Management systems, 237

D
DataDirect JDBC drivers, 159
Dataflow dependency, 59
DataSource interface, 157–158
Deadlock detection and prevention, 13–14
delistResource() method, 78, 83
Digital cash tokens (e-cash), 267
Direct context management and explicit propagation, 135
Direct transaction management, 47
Dirty reads, 161–162
Distributed systems, 4
Distributed TP: Reference Model, Version 3, 364
Distributed TP: The TX (Transaction Demarcation) Specification, 364
Distributed TP: The XA Specification, 364
Distributed Transaction Processing (DTP) model, See DTP:
Distributed transactions, 36–40
    interaction diagram, 252
    interposition, 38–40
JDBC, 167–182
    connection management, 173–182
    XAConnection interface, 169–170
    XDataSource interface, 169–170
    XAResource interface, 170–173
    transaction context, 37–38
DriverManager class, 156–157
DTP, xxvi, 1–2, 63–68
    application program, 64–65
    defined, 64
    -JTA discrepancies, 79–80
    resource manager, 65–66
    transaction manager, 64–65
    TX Specification, 66
    XA Specification, 66–68
    DUPS_OK_ACKNOWLEDGE mode, 191
Durability, 6, 10, 14
    Durable storage media, 19

E
E-cash, 267
E-commerce:
    and Java, 262
    EJBException, 221
    Encina, 33, 373, 380
    Enclosing transactions, 32
    end() method, XAResource, 78
End-to-end transactionality integrity, 265–273
    application servers and thin clients, 273
    CORBA effect, 272
    J2EE effect, 272–273
    and the new generation of transaction systems, 268
    online transaction processing versus object-oriented transaction processing, 270–271
    and OTS, 271–272
    transactional participants, 268–270
    enlistResource() method, 78, 83
    Enterprise Application Integration (EAI), 238
    Enterprise Archive (EAR) file, 239
    Enterprise Information Systems, 255
    Enterprise Integration Systems (EIS), 237
    Enterprise JavaBeans (EJBs), 65
        defined, 207–234
        message-driven beans, 227–229
        transaction support, 210–227
            bean types and transactions, 221–223
            bean-managed transactions (BMTs), 215–217
            BMTs vs. CMTs, 217
            commit options, 226
            container-managed transactions (CMTs), 211–215
            error handling, 217–220
            isolation levels, 226–227
            optimistic locking, 223–224
            pessimistic locking, 225–226
            specification 2.1, 362
        and transactions, 207–235
        types of, 208–209
### Index

*Enterprise JavaBeans* (Monson-Haefel), 209
Enterprise Resource Planning systems, 237
Enterprise-level transaction systems, 26
Entity beans, 208–209
Exactly-once semantics, 37
Exceptional circumstances, 8
executeBatch () method, 166
Extended transaction models, 256, 282–283, 296–301
   nested top-level transactions with compensations, 297–299
two-phase commit, 296–297
workflow coordination, 299–301
Extended transactions:
   spectrum of, 302
types of, 310–311

**F**
Failover:
   and clustering, 56
Failure recovery, 26–30, 46, 86–88
   and heuristics, 29–30
   implementing, 28–29
   and transaction log, 27
*Failure* state, business processes, 342
Fault-isolation:
   and nested scopes, 329
   nested transactions, 32
Fault-tolerant services:
   constructing, 50
File-level locking, 263
File-segment locking, 263
Fiorano Software, 187
FORBIDS (OTSPolicy), 108–109
forget method:
   Resource interface, 121

**G**
Garbage collection, 263–264
get_control method, Current interface, 116–117
get_parent_status, Coordinator interface, 113–114
get_status, Coordinator interface, 113–114
get_timeout method, Current interface, 116
get_top_level_status, Coordinator interface, 113–114
getLocalTransaction() method, 244
getStatus() method, 75
getTransaction() method, 75
TransactionManager interface, 72
getXAResource () method, 244
Global transactions, Java Message Service (JMS), 195–199
   XAConnectionFactory interface, 196–197
   XASession interface, 197–199
Glued transactions, 36
Graph-based technique for deadlock detection, 14
Guarantee, 8

**H**
Heuristic commit, 24
Heuristic decisions, 21
Heuristic hazard, 24
Heuristic mixed, 24
Heuristic outcomes, 8, 19, 21–24
   example of, 22–23
Heuristic rollback, 24
Heuristic state, 21
Heuristic transactions, 21–24
   levels of, 21–22
Heuristics, 316
Hewlett Packard (HP):
   and OMG standard, 96
Hewlett-Packard, 309
   Transaction Service (HP-TS), 379–380
Hewlett-Packard/Arjuna Technologies Transaction Service, 148, 260, 277, 373–374
High performance transaction systems:
   and Java, 259–265
High-Level Services (HLS), 290–296
   Action interface, 290–292
   Signal class, 290–292
   SignalSet interface, 290, 292–295
Hitachi Transaction Service, 374
Hotspot Just In Time (JIT) compiler, 261

IBM, and OMG standard, 96
IIOP (Internet Inter-ORB Protocol), 96
Independent top-level transactions, 34–35
Indirect context management and implicit propagation, 134
Input sets, 59
Instance variables, 51
Integration servers, 238
Interaction interface, 241–242
InteractionSpec interface, 242
Interdependency tasks, 60
Interference-free execution order, 9
Interposition, 38–40, 46
business process interposition:
example of, 276
Object Transaction Service (OTS), 130
universal adapter, 274–277
usefulness of, 274–277
Interpreted language, 260–261
INVALID_TRANSACTION exception, 109
IONA Technologies, Encina transaction service, 33
is_same_transaction:
Coordinator interface, 113–114
Isolation, 6, 9–14, 15
Isolation levels, 226–227

J
J2EE Activity Service, 277, 281–303
activities, 287–288
activity coordinator, 288
coordinating activities, 289–290
coordination, 287–288
defined, 285–286
High-Level Services (HLS), 290–296
implementing, 301
J2EE architecture, 288–289
and nested top-level transactions with compensations, 297–299
participants, 287–288
protocol managers, 288
signals, 288
and two-phase commit protocol, 296–297
and workflow coordination, 299–301
J2EE Activity Service for Extended Transactions, 362
J2EE Connector Architecture:
Common Client Interface (CCI), 240–243
components, 239–240
connection management, 245–246
defined, 237
distributed transaction interaction diagram., 251–252
distributed transactions, 251–253
interactions, 251–252
future of, 253–255
integration servers, 238
resource adapters, 239–240
Service Provider Interface (SPI), 243–244
transaction management, 246–253
CCI LocalTransaction interface, 248–250
local transaction, 247–251
local transaction interaction diagram, 251
NoTransaction support option, 247
SPI LocalTransaction interface, 247–248
transactions, 237–255
Web services, 239
J2EE Connector Architecture (J2C), Enterprise Application Integration (EAI), 238
J2EE Connector (JCA) Specification 1.5, 362
J2EE Reference Implementation (Sun Microsystems), 374
J2EE Test Compatibility Kit (TCK), 239
Java:
and e-commerce, 262
and high performance transaction systems, 259–265
Java 2 Platform, Enterprise Edition (J2EE), xxix–xxxii, 361
Java API for XML Transactions (JAXTX), 346–359, 362
API, 350–359
controlling server-side transactions, 356–357
current transaction status, 352–353
demarcating client-side transactions, 354–356
result of a transaction, 353–354
transaction model-specific APIs, 357–359
transaction termination state, 350–351
architecture, 347–348
client interfaces, 349
components, 346, 348–349
configuration schema, 350
container interfaces, 349
context, 347
implementation interfaces, 349
mandatory interfaces, 349
optional interfaces, 349
Outcome interface, 353–354
participants, 347
Transaction API, 347
and transaction protocols, 350
transaction service, 347
XML configuration, 349–350
Java Community Process (JCP), 208
Java Connector Architecture, 92, 185
Java Database Connectivity (JDBC) API, See JDBC
Java Message Service (JMS):
application server integration, 199–202
defined, 188
two example applications, 202–205
function of, 188–190
global transactions, 195–199
XAConnectionFactory interface, 196–197
XADataSource interface, 197–199
local transactions, 192–195
and boundaries, 193
JMS 1.1 improvements, 195
message acknowledgement, 190–191
AUTO_ACKNOWLEDGE mode, 190–191
CLIENT_ACKNOWLEDGE mode, 191
DUPS_OK_ACKNOWLEDGE mode, 191
message-oriented middleware, 185–188
point-to-point interaction, 188
specification 1.1, 362
and transactions, 185–206
Java Native Interface, 262, 367
Java Transaction API (JTA), xxvi, 41, 63–94, 67–78, 95, 273
application level transaction management in Java, 67
-DTP discrepancies, 79–80
general procedure in transaction management scenario, 84
specification 1.0.1B, 361
TransactionManager interface, 72–75
UserTransaction interface, 69–72
Java Transaction Service (JTS), xxvi, 41–42, 95–152, 361
java.sql.Connection, 154–158
JBoss, 56
Transaction Manager, 375
JDBC, 238
basic model, 154–160
java.sql.Connection, 154–158
batch updates, 166–167
Connection interface:
 isolation levels, 160
distributed transactions, 167–182
connection management, 173–182
XAConnection interface, 169–170
XDataSource interface, 169–170
XAResource interface, 170–173
isolation constraints, 162–164
TRANSACTION_NONE, 163
TRANSACTION_REPEATABLE_READ, 163
TRANSACTI
TRANSACTION_SERIALIZEABLE, 163
local transactions, 164–166
relaxed isolation, implications of, 161–162
dirty reads, 161–162
non-repeatable reads, 162
phantom reads, 162
supported isolation levels, 161
transaction isolation levels, 160–164
and transactions, 153–183
types of JDBC drivers, 159–160
updateable result sets, 167
version 2.0 advanced features, 166–167
JDBC Specifications, 3.0.2.1, and Optional
Package API 2.0, 362
JGroups, 56
JOTM, 375

L
Last resource commit, 92–94
optimization, 17
Lisp, 260
Load balancing:
and clustering, 56
Local transactions:
interaction diagram, 251
Java Message Service (JMS), 192–195
and boundaries, 193
JMS 1.1 improvements, 195
JDBC, 164–166
LocalTransaction interface, 243–244
Logical transaction flow, 84–86
Long-running transactions, 282
Loosely coupled distribution applications, 47

M
ManagedConnection class, 244
ManagedConnectionFactory, 244
Mandatory attribute:
container-managed transactions, 213
MarkedCancelOnlyStatus, 352
Meeting arrangements, 283–284
Memory mapped files, 263
Message acknowledgement, 190–191
AUTO_ACKNOWLEDGE mode, 190–191
CLIENT_ACKNOWLEDGE mode, 191
DUPS_OK_ACKNOWLEDGE mode, 191
Message-driven beans, 227–229
bean-managed transactions (BMTs), 229–230
container-managed transactions (CMTs), 230–231
transaction context propagation, 231–234
MessageListener, 190
Message-oriented middleware, 185–188
broadcast messaging, 187
local/global transactions, support of, 187
messaging products, 187
scalability, 186
Methods, 51
Middleware, 185–188
Mixed transactional client, 110
Modularity:
and nested scopes, 329
nested transactions, 32–33

N
Name server access, 283
Nested top-level transactions, 34–35
with compensations, 297–299
Nested transactions, 32–34
compared to top-level transactions, 32
fault-isolation, 32
modularity, 32–33
Network address translation (NAT), 56
Never attribute, container-managed transactions, 213
New state, use of term, 25
Non-repeatable reads, 162
Notification (temporal) dependency, 59
NoTransactionStatus, 352
NotSupportedException attribute, container-managed
transactions, 211–212
OASIS Business Transactions Protocol (BTP), See Business Transactions Protocol (BTP)
Object activation and access, 54
Object binding, 54
Object decomposition, 277–279
Object Management Group (OMG), 96, 208, 285, 381
Object passivation, 54
Object Relational mapping runtime engines, 153
Object Request Broker (ORB), 96, 381
Object Services, 96
Object Transaction Monitors (OTM), 208
Object Transaction Service (OTS), 2, 22, 96–98, 98, 130, 381
application programming models, 100–102
architecture, 99–131
checked transaction behavior, 131–132
choosing an implementation, 147–149
Control interface, 100, 111
Coordinator interface, 100, 113–115
Current interface, 100, 115–118
heuristics, handling, 110–111
interposition, 130
nested transactions, 106–107
participant relationships, 125–126
participating in a transaction, 116–128
Resource interface, 116–121
recoverable server, implementing, 135–141
RecoveryCoordinator interface, 128–130
relationship to other transaction standards, 98–99
Resource/SubtransactionAwareResource interfaces, 100
shared and unshared transactions, 131
SubtransactionAwareResource interface, 122–124
Synchronization interface, 124–125
Terminator interface, 100, 112–113
transaction context management, 103–106, 134–135
transaction controls, 111–118
transaction interoperability, 132–133
transaction propagation, 107–109
TransactionFactory interface, 102–103
transactions and registered resources, 126–127
worked example, 141–147
client implementation, 145–146
Resource implementation class, 142
sequence diagram, 146–147
server implementation, 143–144
transactional implementation, 143
Object Transaction Service (OTS) Specification 1.3, 364
and end-to-end transactionality integrity, 271–272
Object-oriented transaction systems (OO-TP), 270–271, 274, 279
ODBC bridge, 159
ODMG standard, 133
100% solution, 262–265
One-phase optimization, 17
Online Transaction Processing systems (OLTP), 268, 270–271, 274, 279
onMessage() method, MessageListener, 200
Open Group, 63–64, 66, 67
Open Group Distributed Transaction Model (DTP), See DTP
Open-top coordination, 309–310
Operations, 51
Optimistic concurrency control, assumptions of, 12
Optimistic locking, 223–224
Optimistic vs. pessimistic concurrency control, 11–12
Oracle, 237, 309
Oracle 10g RDBMS, 65
Oracle Advanced Queuing, 65, 187
Oracle Application Server, 56, 200, 225
Oracle Technology Network, 222
Oracle Transaction Service, t, 374
ORBs, 272
Original state, use of term, 25
OSI TP protocol, 132
OTS interfaces, writing applications using, 133–147
Outcome interface, JAXTX, 353–354
outcome Signal, 300
outcome_ack Signal, 300
Output sets, 59

P
Participant log, 26
interactions, 25–26
Participants, 311
Synchronization, 19
transactional, 14–15
Participant-to-coordinator heuristic interaction, 21–22
Passivation, 222–223
Passive replication, 52–53
Performance, and interposition, 40
Persistence, 10, 14
Persistence/recovery service, 97
Persistent objects, maintaining information on, 51–52
Personal diary object, 283
Pessimistic concurrency control mechanisms, 11–12
Pessimistic locking, 225–226
Phantom reads, 162
Portable Object Adapters (POA), 97, 107, 232
Posix Threads, 263
prepare method, Resource interface, 120–121
Presumed abort, 16–17
Principles of Transaction Processing (Bernstein/
Newcomer), 3
Process order application example, 60–61
Protocol bridges, 92
Protocol engine, 97
Proxy coordinators, 39–40

Q
QueueConnection, 189
QueueConnectionFactory, 189
Queues, and transactions, 47–50
QueueSession, 189

R
Read lock, 10
Read-only optimization, 17
Record interface, 241
recover() method, 85
Recoverable objects, 97, 381
Recoverable server:
example of, 137–140
implementing, 135–141
resource object, 137
transactional object, 136
example of, 140–141
RecoveryCoordinator interface, 128–130
register_resource, Coordinator interface, 114
register_resource method,
  SubtractionAwareResource
  interface, 122–123
register_subtran_aware method:
  Coordinator interface, 114
  SubtractionAwareResource
  interface, 122–123
register_synchronization,
  Coordinator interface, 115
Relational database management systems, 237–238
Relaxed isolation, implications:
dirty reads, 161–162
non-repeatable reads, 162
phantom reads, 162
Remote Method Invocation (RMI), 96
Remote Procedure Calls (RPC), 185
Remote transactional invocation, 43–44
RemoteException, 220–221
Replica group technology, 54
Replica-consistency protocol, 50
Replicated computations, 51
Replication:
active, 52–53
groups, 54
integrating transactions and, 54–55
managing, 54
passive, 52–53
protocols, 52–54
total order property, 54
and transactions, 50–57
Reply check, 132
Request and response queues, 50
Request-insertion end, 48
Request-removal end, 48
Required attribute, container-managed transactions, 212
REQUIRES (OTSPolicy), 108–109
RequiresNew attribute, container-managed transactions, 212–213
Resource Adapter Archive (RAR) file, 239
Resource CORBA interface, 97
Resource interface:
commit method, 121
commit_one_phase method, 121
forget method, 121
prepare method, 120–121
rollback method, 121
Resource manager, and protocol bridges, 92
Resources, 361–364
ResultSet interface, 243
Resume check, 132
resume method, Current interface, 117
Resuming transactionality, 47
Rollback, 7
Rollback command:
transaction API, 41
rollback method, Resource interface, 121
rollback() method,
TransactionManager interface, 73
rollback_only method:
Coordinator interface, 115
Current interface, 116
RPC-based middleware technologies, 185–186
Scalability:
and clustering, 56
and RPC-based middleware technologies, 186
Scatter-gather buffers, 264
Scopes of work, 307
Security, and interposition, 40
Seibel, 237
Separation of concerns, and interposition, 40
Serializability, 9–14
maintaining, 10
Serializable execution order, 9
Service Provider Interface (SPI), 243–244
javax.resource.spi.Managed Connection, 244
javax.resource.spi.ManagedConnectionFactory, 244
Services, 14–15
SessionSynchronization interface, 79, 223
set_timeout method, Current interface, 116
Shared memory, 263
Shared transaction model, 131
sid (system identifier), 157
Simple Object Access Protocol (SOAP), RPC facility, 185
Single-threaded transaction systems, 44
SNA LU 6.2 protocol, 133
Socket-level programming, 185–186
Sonic Software, 187, 200–201
Sparse files, 265
Stand-alone transaction system implementations, 373–374
start Signal, 299
start_ack Signal, 300
Stateful session beans, 208, 216
Stateless session beans, 208, 216
Status interface, 76–77
Status values, Coordinator interface, 113–114
Strong consistency, 52
Subordinate coordinators, 39–40
SubtransactionAwareResource interface, 97, 122–124, 126–127
SAP, 237
Sapphire Integration Modules (SIMs), 237
Subtransactions, 32–34
fault-isolation, 32
modularity, 32–33
Supports attribute, container-managed
transactions, 212
suspend method, Current interface, 117
Suspending transactionality, 47
Synchronization interface, 78–79, 124–125
afterCompletion() method, 79
beforeCompletion() method, 79
Synchronization participants, 19
Synchronizations, 19–21
synchronized keyword, 263
Synchronous checked transactions, 45
System exceptions, 220
SystemException, 79–80

T
Tasks, 58
Terminator interface, 112–113
Thread synchronization, 263
Tibco Rendezvous, 187
TID, 67–68
Timeout-based technique for deadlock detection, 13–14
TMFAIL, 78
TMSUCCESS, 78
TMSUSPEND, 78
Top-level transaction commit, 128
Top-level transactions, 7, 32, 58, 277, 281
Transaction attribute summary, 214
Transaction context propagation, 231–234
Transaction inflow sequence diagram, 254
Transaction interface, 75–78
resource enlistment, 77–78
transaction status, 76–77
Transaction interoperability, 132–133
Transaction isolation levels, 160–164
Transaction log, 24–25, 24–26
Transaction manager factory, 7
Transaction processing, xxvi–xxvii
history of, xxvii

Transaction Processing: Concepts and Techniques
(Gray), 3
Transaction processing monitors (TPM), 57–58
Transaction processing systems, 57
interceptors, 37
principles of, 3
Transaction propagation, 107–109
OTSPolicy values, 108–109
TransactionalObject interface, 108
Transaction service:
build vs. buy, 375–376
choosing, 375–378
making a wrong choice, 377–378
underestimating your requirements, 376–377
Transaction synchronization, 78–79
Transaction system implementations, catalog of,
372–375
commercial implementations, 373–374
application server embedded
implementations, 374
stand-alone implementations, 373–374
Transaction systems:
CICS, 373, 379
Encina, 33, 373, 380
Hewlett-Packard Transaction Service (HP-
TS), 379–380
Object Transaction Service (OTS), 381
Tuxedo (BEA), 65, 260, 374, 379
Web-effect on, 261–265
X/Open, 380
TRANSACTION_NONE, 163
TRANSACTION_READ_COMMITED, 163
TRANSACTION_READ_UNCOMMITTED, 163
TRANSACTION_REPEATABLE_READ, 163
TRANSACTION_SERIALIZABLE, 163
Transactional API (Tx API), 41–44
Transactional client using direct and explicit
transaction propagation (example), 109
Transactional client using indirect and explicit
transaction propagation (example), 109–110
Transactional client using indirect and implicit
transaction propagation (example), 110
Index

Transactional interaction diagram, 198–199
Transactional objects, 14–15, 97, 381
Transactional participants, 14–15
Transactional queue, 48
Transactional remote procedure (TxRPC), 37
Transactional request dequeue, 48–49
Transactional request enqueue, 48–49
Transactional web services specifications, 363–364
Transactionality, suspending/resuming, 47
TransactionalObject interface, 108
Transaction-definition schema, 349
TransactionFactory interface, 100
TransactionManager interface:
  begin() method, 72
  commit() method, 72–73
  getTransaction() method, 72
  resume() method, 73–75
  rollback() method, 73
  suspend() method, 73
  suspending/resuming a transaction, 73–75
Transactions:
  aborted, 6–7
  advanced concepts, 259–280
  checked, 43–47
  committed, 6–7
  concurrent, 35
  controlling, 40–44
  defined, 4
  distributed, 36–40
  enclosing, 32
  glued, 36
  independent top-level, 34–35
  integrating replication and, 54–55
  J2EE Connector Architecture, 237–255
  and Java Message Service (JMS), 185–206
  nested, 32–34
  properties of, 6
  and queues, 47–50
  and replication, 50–57
  termination of, 6–7
  and threads, 43–47
  types of, 32–36
  and Web services, 305–360
  Transparent failover, 56–57
  Trust, 30–31
  and interposition, 40
  Tuxedo (BEA), 65, 260, 374, 379
  Two-phase commit, xxviii–xxix
  Two-phase commit optimizations, 16–19
    last resource commit optimization, 17
    one-phase optimization, 17
    presumed abort, 16–17
    read-only optimization, 17
    Two-phase commit protocol, 7–8, 19
    Two-phase concurrency control, 10–11
    Two-phase locking policy, 11
    TX Specification, 66
    TxRPC, 37
    Type-specific concurrency control, 12–13
    Tyrex, 375
U
  Universal adapter, 274–277
  Unshared transaction model, 131
  UserTransaction interface, 79, 229–230
  util.concurrent library, 263
V
  Virtually synchronous, 56
W
  Web services, 239
  Web Services Choreography (WS-C), 363
  Web Services Composite Application Framework, 332–344
    context service, 333–335
    coordination framework, 335
    transaction models, 336–344
      ACID transaction, 336
      business process (BP) model, 340–344
      business process transaction, 336
      long running action, 336
      long-running activities (LRA), 336–340
Web Services Context (WS-Context), 332
Web Services Coordination Framework (WS-CF), 333
Web Services Transaction Management (WS-TXM), 333, 344–345
Web Services Composite Application Framework (WS-CAF) 1.0, 363
Web Services Coordination (WS-C) specifications, 321–322, 363
context schema fragment, 323
main components involved in using/defining, 322
WS-Transaction dependency on, 323
Web services transactions, 305–360
common features of, 306–309
consensus, obtaining, 306–307
general architecture, 307–309
Web services coordination and transactions, 321–332
WebLogic (BEA), 56
  Transaction Service, 374
WebSphere (IBM), 56
WebSphere MQ (IBM), 65, 187
WebSphere Transaction Service, 374
Web-years, 262
Workflow applications, 58
Workflow Management Coalition (WFMC), 58–59
Workflow Reference Model, 58
Workflow schema, 58
Workflow script, 58
Workflow systems, 58–61
  function of, 60
Workflows, 58
Working state, business processes, 342
Write lock, 10
“Write Once, Run Anywhere” transactional applications, 66
WS-AtomicTransaction, 321
WS-BusinessActivity, 321
WS-Transaction models, 324–332
  API, 332
  atomic transaction, 324–328
  business activities, 328–330
  example of using, 331–332

XA Specification, 66–68
XAConnection interface, 169–170
XADatasource interface, 169–170
XAException, 81
XAResource interface, 15, 75, 81–84, 93, 170–173, 255
  methods, 81
  resource manager, opening/closing, 84
  sequence diagram, 172
  thread of control, 82–83
  transaction association, 83
XA Terminator interface, 255
XID interface, 80
X/Open, 380
  transaction service model, 131–132
  TX interface, 132
  XA interface, 132
  XA transactions, 33