

Index

A

Above normal process and thread priority levels, 65
 Accept API function, 312
 Access tokens, 27
 ACID (atomic, consistent, isolated and durable) test, 499–501
 atomicity, 500
 consistency, 500
 durability, 501
 isolation, 500–501
 ActiveX
 COM, 361
 DTS, 845–846
 implementing conditional execution, 857–859
 looping with ExecutionStatus property, 853–857
 looping with separate packages, 852–853
 GUIDs (interface IDs), 366
 AddRef I Unknown method, 366–367
 Address spaces, processes, 25
 address translation from physical to virtual memory, 110, 124–127
 PDEs (page directory entries), 125, 127
 PTEs (page table entries), 125, 127
 memory, 112–113, 115
 private, 118
 TB (Translation Buffer), 126
 TLB (Translation Look-aside Buffer), 126
 virtual memory, 26, 113
 Address Windowing Extensions (AWE)
 memory
 basics, 120–122
 definition, 111

physical memory, 121–122
 physical memory region with
 AWE support, 436
 basics, 442
 with BPool and MemToLeave regions, 438–439
 sizing, 436
 /3GB boot option, 123–124
 virtual memory, 122
 Advapi32.DLL, 17, 415
 Affinity of processors
 basics, 68–70
 functions
 GetProcessAffinityMask API, 61
 GetThreadAffinityMask API, 61
 SetProcessAffinityMask API, 61
 thread scheduling, 60
 types, 70–72
 Aggregation, COM, 368
 AllocateUserPhysicalPages API
 function, 111, 121
 Allocation granularities
 memory, 110, 133–134
 virtual memory, 140
 Allow Post option, 713–714
 Allow XPath option, 713
 ALTER DATABASE command, 522
 ALTER TABLE command, 506
 Annotated XDR (XML-Data Reduced) mapping schemas, 737–738
 ANSI/ISO SQL-92 specification, automatic cursor closing, 561–564
 ANSI_NULLS setting, 475
 ANSI_PADDING setting, 475
 ANSI_WARNINGS setting, 475
 Apartments (single-threaded and multithreaded) COM models, 369–371, 574–576

APCs (asynchronous procedure calls), 201
 API functions
 asynchronous I/O, 201
 I/O, 193, 412
 I/O completion ports, 253
 I/O memory mapped files, 289
 memory, 111
 heaps, 165
 networking, 312–313
 processes, 27
 scatter-gather I/O, 235
 shared memory, 176
 thread scheduling, 61
 thread synchronization, 85
 threads, 40
 virtual memory, 141
 Application layer, OSI reference model, 314
 networking components, 316
 Architecture
 COM
 aggregation, 368
 interfaces, 366
 marshaling, 367–368
 QueryInterface method, 367
 reference counting, 366–367
 Microsoft Search full-text searches, 632–633
 ARITHABORT setting, 475
 Asynchronous I/O
 APCs (asynchronous procedure calls), 200–201
 asynchronous RPCs (Remote Procedure Calls), 358
 clusters, 200
 networking, 346
 overlapped I/O, 193
 overview, 202–204
 sectors, 200
 Atomic access, thread synchronization, 85

1000 Index

AUTO mode queries
 basics, 682–684
 ELEMENTS option, 684–685

Auto-parameterization, query processing, 465–466

Auto-reset events, 95

Automation
 of COM objects
 easy binding *versus* late binding, 576–577
 .NET Framework classes with COM interoperability, 588–591
 threading models, 574–576
 in user-defined functions, 591–594
 VB array functions, 619–630
 of custom objects, 614–619

ODSOLE
 sp_checkspelling, 582
 sp_OA procedures with other procedures, 582
 sp_vbscript_reg_ex, 585–588

sp_OA procedures
 dot notation for object name
 transversal, 580–581
 named parameters, 581
 sp_OACreate, 578–579
 sp_OADestroy, 580
 sp_OAGetErrorInfo, 580
 sp_OAGGetProperty, 580
 sp_OAMethod, 579–580
 sp_OA SetProperty, 580
 sp_OAStop, 580

of SQL-DMO COM objects
 sp_exporttable, 594–600
 sp_generate_script, 600–614

Avg. Bytes Free per Page field, 473

Avg. Disk Queue Length counter, 194

Avg. Page Density key indicator, 471

AWE (Address Windowing Extensions)
 memory
 basics, 120–122
 definition, 111
 physical memory, 121–122

physical memory region with
 AWE support, 436
 basics, 442
 with BPool and MemToLeave regions, 438–439
 sizing, 436
/3GB boot option, 123–124

virtual memory, 122

B

B-trees, clustered and nonclustered indexes, 467–469

BACKUP LOG command, 522

BaseProcessStart API function, 30, 44

BEGIN DISTRIBUTED TRANSACTION command, 504

BEGIN TRAN command, 502
 nested transactions, 514–519
 user-defined transactions, 503–504

_beginthreadex, 46–47

Below normal process and thread priority levels, 65

Berkeley Software Distribution (BSD) sockets, 315–317
 versus Winsock, 316, 334

BETWEEN predicate, partitioned view queries, 665–667

Binding in object automation, 576–577

Bosak, Jon, 376

Boundary partitions, 119–120

BPool (buffer pool) memory region, 436

arrays
 BUF, 442, 444–446
 page, 441
 basics, 438–439

Commit Bitmap option, 442

finding BPool with WinDbg, 439–441

hashing pages, 439

management with lazywriter basics, 443
 checkpoint process, 446
 computing physical memory, 443–444
 flushing and freeing pages, 444–446
 partitioning BPool free list, 446
 sizing, 436–438

Bray, Tim, 376

BSD (Berkeley Software Distribution) sockets, 315–317
 versus Winsock, 316, 334

BUF arrays, BPool memory management, 442, 444–446

Buffer pool (BPool) memory region, 436

arrays
 BUF, 442, 444–446
 page, 441
 basics, 438–439

Commit Bitmap option, 442

finding BPool with WinDbg, 439–441

hashing pages, 439

management with lazywriter basics, 443
 checkpoint process, 446
 computing physical memory, 443–444
 flushing and freeing pages, 444–446
 partitioning BPool free list, 446
 sizing, 436–438

BUF_STATE_INACTIVE, 268

BUF_STATE_READING, 268

BUF_STATE_READY, 268

BUF_STATE_SEARCHING, 268

BULK INSERT command, 504, 842–843

Bulk Insert task, 842–843

C

C++, xxxv–xxxvi

Cache counters, 194

CancelWaitableTimer API function, 96

Cardinality, 456, 479

Cdata directive, 688–690

EXPLICIT mode queries, 693–694

CDATA sections, EXPLICIT mode queries, 693–694

Cdb.exe command-line debugger, 21

CheckConstraints property, 761

Class libraries, 18

Client IDs, 27

Client side templates, 731–733

Clock intervals, thread scheduling, 60, 63

- CLOSE command, cursor variables, 566–568
 Close connection on completion property, 849
 CloseHandle API function, 29, 193
 Clusters, I/O, 200
 scatter-gather I/O, 234
 CMD files, Notification Services, 806, 807
 CoInitialize COM function, 369–370
 CoInitializeEx COM function, 369–370
 Column statistics, 481–482
 COLUMNPROPERTY function, 638
 Columns
 full-text searches, 637–642
 partitioning with CHECK constraint, 658–665
 COM (Component Object Model)
 ActiveX, 361
 architecture
 aggregation, 368
 interfaces, 366
 marshaling, 367–368
 QueryInterface method, 367
 reference counting, 366–367
 automation of objects
 easy binding *versus* late binding, 576–577
 .NET Framework classes with
 COM interoperability, 588–591
 threading models, 574–576
 in user-defined functions, 591–594
 VB array functions, 619–630
 history, 362–364
 origin of COM, 364–365
 IIDs (interface IDs), 366
 IMalloc interface, 449
 OLE (Object Linking and Embedding), 361
 DDE (Dynamic Data Exchange), 361
 OleInitialize COM function, 369–370
 overview, 361–362
 SQL Server, 371–372
 threading models, 369–371
 using, 368–369
- Command-line cdb.exe debugger, 21
 Commit Bitmap option, BPool memory management, 442
 Commit Limit counter, memory, 112
 COMMIT TRAN command, 502, 506, 514–519
 user-defined transactions, 503–504
 % Committed Bytes in Use counter, memory, 112
 Committed counter, memory, 112
 Committed memory, 145–146
 virtual memory
 basics, 145–146
 definition, 141
 Completion ports (I/O), 98–99, 412
 API functions, 253
 basics, 253–257
 Component Object Model (COM)
 ActiveX, 361
 architecture
 aggregation, 368
 interfaces, 366
 marshaling, 367–368
 QueryInterface method, 367
 reference counting, 366–367
 automation of objects
 easy binding *versus* late binding, 576–577
 .NET Framework classes with
 COM interoperability, 588–591
 threading models, 574–576
 in user-defined functions, 591–594
 VB array functions, 619–630
 history, 362–364
 origin of COM, 364–365
 IIDs (interface IDs), 366
 IMalloc interface, 449
 OLE (Object Linking and Embedding), 361
 DDE (Dynamic Data Exchange), 361
 OleInitialize COM function, 369–370
 overview, 361–362
 SQL Server, 371–372
 threading models, 369–371
 using, 368–369
- Components, networking, 315–317
- CONCAT_NULL_YIELDS_NULL setting, 475
 Concurrent snapshot processing, 930–932
 Configuration files, 793–794
 Connect API function, 312
 Connection memory manager, 447
 Connection-oriented Winsock applications, 312, 335
 Connectionless Winsock applications, 312, 343–344
 ConnectNamedPipe API function, 312, 318
 Connolly, Dan, 376
 CONTAINS predicate, 643–647
 Contenttype URL query parameter, 720–722
 Context structure, threads, 42
 Context Switches/sec counter, 41
 Context switching
 thread scheduling, 15, 60, 68
 threads, 39
 Conversion stage, performance optimization, 461
 Cooperative tasking *versus* preemptive tasking, UMS (User Mode Scheduler), 421
 Copy-on-write memory, virtual memory
 basics, 143–144
 definition, 141
 CPU registers
 thread internal structures, 41–42
 threads, 15
 CRCs (cyclic redundancy checks), 935–938
 CREATE command, 506
 CREATE DATABASE command, 522
 Create Dynamic Snapshot Job Wizard, 952–953
 CREATE INDEX command, 504
 Create Publication Wizard, 950
 CREATE STATISTICS command, 483–484
 CreateEvent API function, 85
 CreateFile API function, 193
 memory-mapped files, 178
 CreateFileMapping API function, 289–290
 shared memory, 176
 memory-mapped files, 178

1002 Index

CreateI/OCompletionPort API
 function, 98–99, 253–255
CreateMutes API function, 85
CreateNamedPipe API function,
 312, 317–318
CreateProcess API function, 27, 29
 memory protection, 115
CreateSemaphore API function, 85
CREATE_SUSPENDED flag, 72
CreateThread API function, 40, 44
 _beginthreadex, 46–47
CreateWaitableTimer API function,
 85, 95–96
Creating application databases,
 817–818
Creating instance databases, 817–818
Creating Process counter, 28
Critical sections, 89–91
CSRSS (Win32 subsystem), 30, 42
Csrss.exe environmental subsystem
 process, 17
Cursors
 appropriate use, 539–540
 asynchronous, 559–561
 automatic closing, 561–564
 basics, 533
 CLOSE command, 546, 558
 configuring, 558–559
 @@CURSOR_ROWS command,
 546, 552–553
 CURSOR_STATUS command, 546
 DEALLOCATE command, 546,
 558, 566–568
 DECLARE CURSOR command,
 546, 550
 global cursors, 551–552
 local cursors, 551–552
 defaulting to global/local cursors,
 564
 DYNAMIC command, 533,
 535–536
 dynamic queries, 540–543
 FETCH command, 546, 553–558
 @@FETCH_STATUS command,
 546, 553
 FORWARD_ONLY command,
 533–534
 ISAM databases, 530–532
 KEYSET command, 533, 537–539
 OPEN command, 546, 552–553
 overview, 529–530
 performance optimization,
 569–571

row-oriented operations, 543–546
scrollable forms, 546
sp_cursor_list procedure, 569
sp_describe_cursor procedure,
 569
sp_describe_cursor_columns
 procedure, 569
sp_describe_cursor_tables proce-
 dure, 569
STATIC, 533, 536–537
stored procedures, 568–569
types, 533–539
 updating, 565–566
variables, 566–568
CURSOR_STATUS function,
 567–568
Custom heaps
 basics, 167–168
 definition, 165
Custom Task interface, DTS (Data
 Transformation Services), 836,
 864–873
Custom tasks
 creating, 864–873
 based on sample tasks, 874–882
 debugging, with VB application,
 885
Cyclic redundancy checks (CRCs),
 transactional replication,
 935–938

D

Data Driven Query task, 839, 843–845
Data-link layer, OSI reference
 model, 314
 networking components, 316
Data pump, multiphase data pump,
 836–837, 839–842
Data relationships, EXPLICIT mode
 queries, 690–692
Data Transformation Services
 (DTS), 835
 ActiveX transformations, 845–846
 automation application, 886–888
 Bulk Insert task, 842–843
 Custom Task interface, 836
 custom tasks
 creating, 864–873
 creating, based on sample
 tasks, 874–882
 debugging, with VB applica-
 tion, 885
 Data Driven Query task, 843–845
DEALLOCATE command, cursor
 variables, 566–568

- Debugging tools
 command-line, *cdb.exe*, 21
 debugging symbols, 21–22
 transactions, 522–524
- WinDbg
 basics, xxxii–xxxiii
 BPool memory management, 439–441
 source and symbol paths, 22
- Decision Support System (DSS)
 applications, 505
- DECLARE CURSOR command, 551–552
- Default heaps, 165
- DELETE command, 506
 WHERE CURRENT OF clause, 565–566
- Delivery of notifications, 803–804
- Density, 456, 479, 480–481
- DHTML (Dynamic HTML), 375, 376
- Digest notifications, 804
- Directives, EXPLICIT mode queries
 cdata, 688–690
 element, 688–690
 hide, 688–690
 id, 688–690
 idref, 688–690
 idrefs, 688–690
 xml, 688–690
 xmldtext, 688–690
- DISK INIT command, 522
- % Disk Read Time counter, 194
- Disk reads/sec counter, 194
- Disk Write Time counter, 194
- Disk writes/sec, 194
- Distributed COM (DCOM), 368
- Distributed Computing Environment (DCE), 356
- Distributed Transaction Coordinator (DTC), 504
- Distribution Agent
 snapshot replication
 basics, 908
 tasks, 915–916
 transactional replication, 923
 skipping errors, 938
- Distribution clean up task, 939
- Distributors, Notification Services
 basics, 802–803
 delivery notifications, 803–804
 message formatting, 803
 multicast digest notifications, 804
- DLLs (Dynamic Link Libraries), 13, 17–20
 Advapi32.DLL, 17, 415
 files, instance handles, 27
 finding undocumented SQL features, 968–969
- GDI32.DLL, 13, 17
- history of COM, 363–364
- Kernel32.DLL, 13, 17
 static importing, 18
 static linking, 415
- .LIB files, 18
- loading processes implicitly or explicitly, 18
- mapped code, 27
- Network APIs, 316Msvcrt.dll and Msvcrtdll, 415
- NTDLL.DLL, 17
- Opends60.dll, 415
- paging files, 19
- physical memory, 19
- Rpcrt.dll, 415
- SQL Server, 414–415
- Sqsort.dll, 415
- subsystems, 17
- Ums.dll, 415
- User32.DLL, 13, 17–18
 static linking, 415
 statically importing, 18
- Windows, 17–20
- .DOC (Microsoft Word), SQL Server support, 636
- DOM (Document Object Model)
 MSXML parser, 673–675
 processing XML with MSXML parser, 398–400
- XML (Extensible Markup Language), 397–398
- Dot notation for object name traversal, automation, 580–581
- DPVs (distributed partitioned views), 658, 668–669
- DROP command, 506
- DROP DATABASE command, 522
- DSO Rowset provider, 861–862
- DSS (Decision Support System) applications, 505
- DTC (Distributed Transaction Coordinator), 504
- DTDs (Document Type Definitions), 383–385
- XML Schemas, 386–388
- DTS (Data Transformation Services), 835
- ActiveX transformations, 845–846
- automation application, 886–888
- Bulk Insert task, 842–843
- Custom Task interface, 836
- custom tasks
 creating, 864–873
 creating, based on sample tasks, 874–882
 debugging, with VB application, 885
- Data Driven Query task, 843–845
- DTS Designer, creating packages, 837–838
- DTSPKGuru application, 888–904
- Import/Export Wizard, 837
 creating packages, 837–838
- lookup queries, 847–849
- multiphase data pump, 836–837, 839–842
- DataDriven Query task, 839
- Parallel Data Pump task, 839
- Transform Data task, 839
- overview, 836–837
- packages, 837–838
 connections, 838
 controlling workflow through scripting, 852–859
- DSO Rowset provider, 861–862
- parameterized packages, 859–861
- tasks, 838–839
- workflow properties, 849–851
- Query Designer, 837
- Read File task, 847
- replication subscriptions, 862–864
- WriteFile transportation, 846–847
- DTSDIAG utility
 capabilities, 971–972
 configuration file, 973–974
 Execute Package method, 974–976
 packages utilized, 980–983
 source code, 972–973
 TranslateVars function, 976–980
- DTSPKGuru application, 888–904
- DUMP TRANSACTION command, 522
- DuplicateHandle API function, 195
- DYNAMIC cursors, 533, 535–536

1004 Index

Dynamic Data Exchange (DDE), 361
 Dynamic HTML (DHTML), 375, 376
 Dynamic Link Libraries (DLLs), 13, 17–20
 Advapi32.DLL, 17, 415
 files, instance handles, 27
 finding undocumented SQL features, 968–969
 GDI32.DLL, 13, 17
 history of COM, 363–364
 Kernel32.DLL, 13, 17
 static importing, 18
 static linking, 415
 LIB files, 18
 loading processes implicitly or explicitly, 18
 mapped code, 27
 Msvcrt.dll and Msver.dll, 415
 Network APIs, 316
 NTDLL.DLL, 17
 Opend60.dll, 415
 paging files, 19
 physical memory, 19
 Rpct.dll, 415
 SQL Server, 414–415
 Sqlsort.dll, 415
 subsystems, 17
 Ums.dll, 415
 User32.DLL, 13, 17–18
 static linking, 415
 statically importing, 18
 Windows, 17–20

E

ECMAScript European standard JavaScript, 397
 Edge table format, 706–707
 8759 trace flag, 458, 466
 Elapsed Time counter
 processes, 28
 threads, 41
 Element directive, 688–690
 ELEMENTS option, FOR XML
 AUTO clause, 684–685
 _endthreadex, 46–47
 EnterCriticalSection API function, 85, 89–90
 Enterprise Manager
 bypassing to script undocumented and system objects, 966–967
 COM and SQL Server, 371

Entry-point functions
 processes, 26
 threads, 39
 Environment subsystems, 16–17
 EPROCESS (executive process)
 block, 29, 30
 EPROCESS (executive process)
 blocks, 29, 30
 address translation, 125
 ErrorLogFile property, 766
 Ethernet networking component, 316
 IrDa, 316
 ETHREAD (executive thread)
 blocks, 29
 Event objects, 94, 95
 Event providers, Notification Services, 796–797
 Exception handling, threads, 48–49
 EXE files, instance handles, 27
 Execute on main package thread property, 849–850
 ExitProcess API function, 27, 30
 ExitThread API function, 40, 45
 Explicit mapping, 740–741
 EXPLICIT mode queries
 cdata directive, 693–694
 data relationships, 690–692
 directives, 688–690
 cdata, 688–690
 element, 688–690
 hide, 688–690, 692–693
 id, 688–690
 idref, 688–690
 idrefs, 688–690
 xml, 688–690
 xmltext, 688–690
 id directive, 694–696
 idref directive, 694–696
 idrefs directive, 694–696
 XMLDATA option, 694–696
 Expose runtime errors as HTTP error option, 714
 Extended procedures, undocumented SQL features, 964–965
 Extensible Markup Language (XML)
 DOM, 397–398
 DTDs, 383–385
 fragment property, 759–760
 history, 376–377

HTML (HyperText Markup Language)
 advantages/disadvantages, 377–380
 converting XML to HTML, 389–397
 disadvantages, 375–376
 terminology, 380–382
 overview, 374–375
 well-formed and valid documents, 382–383

XML Schemas, 386–389

XPath, 391
 Extensible Stylesheet Language Transformations (XSLT)

 converting XML to HTML, 389–397
 embedded scripting, 397

 style sheets, URLs, 719–720
 xsl:attribute construct, 397
 xsl:choose construct, 397
 xsl:if construct, 397
 xsl:sort construct, 397

Extent Switches field, 473

Extents Scanned field, 473

F

FAT16 file system, 192
 FAT32 file system, 192
 FETCH command, 506
 cursor variables, 566–568
 Fiber mode, 50–51
 threads, 39
 UMS (User Mode Scheduler), 430
 Fibers, *versus* threads, 50–51
 File objects, 193
 File systems, 192–193
 FILE_FLAG_NOBUFFERING flag, 212
 FILE_FLAG_OVERLAPPED flag, 416
 Filters
 full text searches, 635–636
 merge replication
 basics, 949–950
 dynamic filters, 951
 dynamic snapshots, 952–953
 join filters, 951–952
 synchronization, 950–951
 Fish!, 987–989
 Flags
 CREATE_SUSPENDED, 72
 8759 trace, 458, 466

- FILE_FLAG_NOBUFFERING, 212
 FILE_FLAG_OVERLAPPED, 416
 HEAP_GENERATE_EXCEPTI ONS, 166–168
 HEAP_NO_SERIALIZE, 166–168
 HEAP_REALLOC_IN_PLACE, 166
 HEAP_ZERO_MEMORY, 166
 IMAGE_LARGE_FILE_ADDR ESS_AWARE, 111, 123
 INVALID_HANDLE_VALUE, 178
 MEM_COMMIT, 146
 MEM_RESERVE, 146
 MEM_RESET, 147
 parameters, 704–705
 #PCDATA, 384
 #REQUIRED, 384
 ROWGUID_COL, 945
 SEM_FAILCRITICALERRORS, 31
 SEM_NOALIGNMENTFAULT EXCEPT, 31
 SEM_NOGPFAULTERRORBO X, 31
 SEM_NOOPENFILEERRORB OX, 31
 STATUS_ACCESS_VIOLATIO, 166
 STATUS_NO_MEMORY, 166
 FlushViewOfFile API function, 176, 178, 289
 initializing and modifying, 146–147
 FOR XML extension
 AUTO mode queries
 basics, 682–684
 ELEMENTS option, 684–685
 CDATA sections, EXPLICIT mode queries, 693–694
 EXPLICIT mode queries
 basics, 685–688
 cdata directive, 693–694
 data relationships, 690–692
 directives, 688–690
 hide directive, 692–693
 id directive, 694–696
 idref directive, 694–696
 idrefs directive, 694–696
 XMLDATA option, 694–696
- RAW mode queries, 681–682
 SELECT...FOR XML queries
 client-side, 696–698
 server side, 680–681
 SQL Server XML feature, 672
 ForceTableLock property, 764
 FORWARD_ONLY (default)
 cursors, 533–534
 FREETEXT predicate, 647
 FREETEXTTABLE rowset function, 636, 651–652
 Full-text indexes, 636–642
 Full-text predicates
 basics, 642–643
 CONTAINS, 643–647
 FREETEXT, 647
 Full-text searches
 binary data, 635–636
 columns, 637–642
 DBCC CALLFULLTEXT
 command, 633
 functions, 634–635
 filters, 635–636
 functions, DBCC CALLFULL TEXT command, 634–635
 Microsoft Indexing Service, overview, 632
 Microsoft Search
 architecture, 632–633
 overview, 632
 non-SQL Server data, 635
 rowset functions
 basics, 646
 CONTAINSTABLE rowset function, 648–651
 FREETEXTTABLE rowset function, 651–652
 FULLTEXTCATALOGPROP ERTY function, 638
 Functions
 Accept, 312
 AllocateUserPhysicalPages, 111
 BaseProcessStart, 30, 44
 CancelWaitableTimer, 96
 CloseHandle, 29, 193
 CoInitialize, 369–370
 CoInitializeEx, 369–370
 COLUMNPROPERTY, 638
 Connect, 312
 ConnectNamedPipe, 312, 318
 CreateEvent, 85
 CreateFile, 193
 memory-mapped files, 178
 CreateFileMapping, 289–290
 shared memory, 176
 memory-mapped files, 178
 CreateI/OCompletionPort, 98–99, 253–255
 CreateMutes, 85
 CreateNamedPipe, 312, 317–318
 CreateProcess, 27, 29
 memory protection, 115
 CreateSemaphore, 85
 CreateThread, 40, 44
 _beginthreadex, 46–47
 CreateWaitableTimer, 85, 95–96
 CURSOR_STATUS, 567–568
 DuplicateHandle, 195
 EnterCriticalSection, 85, 89–90
 ExitProcess, 27, 30
 ExitThread, 40, 45
 FlushViewOfFile, 176, 178, 289
 initializing and modifying, 146–147
 FREETEXTTABLE rowset, 636, 651–652
 GetDiskFreeSpace, 200–201
 GetExitCodeProcess, 29, 31
 GetExitCodeThread, 40
 GetIDsOfNames, 368
 GetLastError, 49–50, 204
 GetOverlappedResult, 201–204
 GetPriorityClass, 61
 GetProcAddress, Load Library API, 18
 GetProcessAffinityMask, 61
 GetProcessHeap, 165
 GetProcessPriorityBoost, 61
 GetQueuedCompletionStatus, 98–99, 253, 255
 GetSystemInfo, 111, 141
 memory allocation
 granularities, 114
 page sizes, 114
 GetThreadPriority, 61
 GetThreadPriorityBoost, 61
 GlobalMemoryStatusEx, 443–444
 HasOverlappedIoCompleted, 201–202
 HeapAlloc, 165
 allocating heap memory, 166–167
 HeapCreate, 165
 HeapDestroy, 165

1006 Index

Functions, *continued*
 HeapFree, 165
 allocating heap memory,
 166–167
 ImpersonateNamedPipeClient,
 317–318
 InitializeCriticalSection, 89
 InitializeCriticalSectionAndSpin-
 Count, 90
 InterlockedCompareExchange-
 Pointer, 89
 InterlockedDecrement, 89
 InterlockedExchange, 85, 87
 InterlockedExchangeAdd, 85, 89
 InterlockedExchangePointer, 89
 InterlockedIncrements, 89
 Invoke, 368
 ISABOUT, 649
 LeaveCriticalSection, 85, 90
 Listen, 312
 MapUserPhysicalPages, 111, 121
 MapUserPhysicalPagesScatter,
 121
 MapViewOfFile, 176, 289–290
 MessageBeep Win32, 49
 OBJECTPROPERTY, 638
 OleInitialize, 369–370
 OpenFile, 194–195
 OpenProcess, 27
 OPENXML
 basics, 698–700
 edge table format, 706–707
 flag parameters, 704–705
 inserting data, 707–711
 SQL Server XML feature,
 672
 using, 700–704
 PostQueuedCompletionStatus,
 99, 253
 ReadFile, 193, 312
 ReadFileEx, 201–204
 ReadFileScatter, 235
 ReadProcessMemory, 29, 111
 memory protection, 115
 Recv, 312
 ReleaseMutes, 85
 ReleaseSemaphore, 85
 ResetEvent, 85, 95
 ResumeThread, 72–73
 Send, 312
 SetCriticalSectionSpinControl, 90
 SetErrorMode, 31–32
 SetEvent, 85, 95, 98

SetEvent/ResetEvent, 85
 SetLastError, 49
 SetPriorityClass, 61
 SetProcessAffinityMask, 61
 SetProcessPriorityBoost, 61
 SetThreadAffinityMask, 61, 70
 SetThreadIdealProcessor, 61, 70
 SetThreadPriority, 61
 SetThreadPriorityBoost, 61
 SetWaitableTimer, 85, 95–96
 SetWorkingSetSize, 141
 Sleep, 61, 73
 SleepEx, 61, 73
 Socket, 312
 SuspendThread, 61, 72–73
 SwitchToThread, 61, 74
 TerminateProcess, 27, 30
 TerminateThread, 40, 45
 TlsAlloc, 39
 TlsGetValue, 39
 TlsSetValue, 39
 TryEnterCriticalSection, 90
 VirtualAlloc, 111, 140, 141
 committed memory, 145–146
 page protection attributes, 142
 VirtualAllocEx, 140, 141
 VirtualFree, 141
 freeing memory, 147
 VirtualLock, 141
 freeing memory, 147
 locking pages, 148
 VirtualProtect, 141
 freeing memory, 147
 VirtualQuery(Ex), 141
 VirtualUnlock, 141
 WaitForMultipleObjects, 85,
 94–95, 98
 WaitForSingleObject, 30, 85,
 94–95, 177
 WaitForSingleObjectEx, 201
 WriteFile, 193, 312
 WriteFileEx, 201–204
 WriteFileScatter, 235
 WriteProcessMemory, 111
 memory protection, 115
 WriteProcessMemorys, 29
 WSASocket, 312
 WSAStartup, 312

G

4GB virtual memory, 15–16
 GDI32.DLL, 13, 17
 General memory manager, 447

Generator, Notification Services
 match rules, 798
 QuantumDuration node, 798
 GetDiskFreeSpace API function,
 200–201
 GetExitCodeProcess API function,
 29, 31
 GetExitCodeThread API function,
 40
 GetIDsOfNames COM function,
 368
 GetLastError API function, 49–50,
 204
 GetOverlappedResult API function,
 201–204
 GetPriorityClass API function, 61
 GetProcAddress API function, Load
 Library API, 18
 GetProcessAffinityMask API func-
 tion, 61
 GetProcessHeap API function, 165
 GetProcessPriorityBoost API
 function, 61
 GetQueuedCompletionStatus API
 function, 98–99, 253, 255
 GetSystemInfo API function, 111,
 141
 memory allocation granularities,
 114
 page sizes, 114
 GetThreadPriority API function, 61
 GetThreadPriorityBoost API
 function, 61
 Global cursors
 defaulting to, 564
 versus local cursors, 551–552
 GlobalMemoryStatusEx API
 function, 443–444
 GRANT command, 506
 Guard pages, virtual memory, 141
 GUIDs (Globally Unique Identifi-
 ers), 758

H

HAL (hardware abstraction layer),
 63, 316
 Handle Count counter, 28
 handles pseudohandle, 40
 Handling exceptions. *See* Exception
 handling
 Hard processor affinity, 70–72
 Hardware Abstraction Layer (HAL),
 63, 316

- Hash pages, BPool memory management, 439
- HasOverlappedIoCompleted API function, 201–202
- HeapAlloc API function, 165
allocating heap memory, 166–167
- HeapCreate API function, 165
- HeapDestroy API function, 165
- HeapFree API function, 165
allocating heap memory, 166–167
- HEAP_GENERATE_EXCEPTIONS** flag, 166–168
- HEAP_NO_SERIALIZE** flag, 166–168
- HEAP_REALLOC_IN_PLACE** flag, 166
- Heaps (memory)
allocating memory, 166–167
- API functions, 165
- custom heaps
basics, 167–168
definition, 165
- default heaps, 165
- serializing
basics, 168–169
definition, 165
- HEAP_ZERO_MEMORY** flag, 166
- Hidden schedulers, UMS (User Mode Scheduler), 431
- Hide directive, EXPLICIT mode queries, 688–690, 692–693
- High/Highest process and thread priority levels, 65
- .HTM (Hypertext Markup Language) SQL Server support, 636
- HTML (HyperText Markup Language)
disadvantages, 376–377
- XML (Extensible Markup Language), 374–375
advantages/disadvantages, 377–380
converting XML to HTML, 389–397
terminology, 380–382
- HTTP SQL Server access
basics, 712–714
virtual directory configuration, 712–714
- I**
- I/O
API functions, 193, 412
asynchronous, 200
- APCs (asynchronous procedure calls), 200
- clusters, 200
- overlapped I/O, 193
- overview, 202–204
- sectors, 200
- completion ports, 98–99, 412
- API functions, 253
basics, 253–257
- file systems, 192–193
- I/O list, UMS (User Mode Scheduler), 426–428
- memory-mapped files
API functions, 289
basics, 289–291
- monitoring tools, 193
- Perfmon counters, 194
- nonbuffered, 200, 211–213
- overview, 194–195
- scatter-gather, 233–234
- API functions, 235
- clusters, 234
- overview, 235–236
- sectors, 234
- states, 268
- synchronous, 195
- IAM (Index Allocation Map), 467
- Id directive, EXPLICIT mode queries, 688–690, 694–696
- Id Process counter, 28, 41
- Id Thread, 41
- Ideal processors
affinity, 70–72
thread scheduling, 60
- Identity column values, 756–758
- Identity range management, merge replication, 953–955
- Idle loops, UMS (User Mode Scheduler), 428–429
- Idle process and thread priority levels, 65
- Idref and idrefs directives, EXPLICIT mode queries, 688–690, 694–696
- IgnoreDuplicateKeys property, 761–762
- IDs (interface IDs), COM, 366
- Image-mapped files, 178–179
- IMAGE_LARGE_FILE_ADDRESS_AWARE** flag, 111, 123
- IMalloc interface (COM), 449
- Immediate updating subscribers
with queued updating as failover, 935
- snapshot replication, 916
with queued updating as failover, 919
- transaction replication, 933–934
- ImpersonateNamedPipeClient API function, 317–318
- IMPLICIT TRANSACTIONS** command, 506
- Index Tuning Wizard, undocumented SQL features, 961, 967
- Indexable expressions
basics, 486
- folding, 488–489
- operator translations, 486–488
- Indexes
B-trees, clustered and nonclustered indexes, 467–469
- B-trees, nonclustered indexes, 467–469
- on computed columns, 474–478
- required settings, 475
- required views, 474–478
- covering, 469
- fragmentation, 470–473
- Index Allocation Map, 467
- intersection, 470
- locking, 478–479
- performance issues, 469–470
- query processing, 466
- row identifiers, 469
- locking, 478
- storage
clustered indexes, 467–469
- nonclustered indexes, 467–469
- type selection, 489–493
- on views, 474–478
- required settings, 475
- INFINITE** constant, 73
- InitializeCriticalSection API function, 89
- InitializeCriticalSectionAndSpinCount API function, 90
- Initialized state, 64
- Input/output. *See* I/O
- INSERT command, 506
- INSERT...SELECT FROM OPENXML clause, 707–711
- Installation scripts, finding undocumented SQL features, 968

1008 Index

Instance database, 793
 Instance handles, 27
 Intel x86 family of processors, 14–15
 Interlocked functions, thread synchronization, 84, 88–89
 InterlockedCompareExchange—Pointer API function, 89
 InterlockedDecrement API function, 89
 InterlockedExchange API function, 85, 87
 InterlockedExchangeAdd API function, 85, 89
 InterlockedExchangePointer API function, 89
 InterlockedIncrement API functions, 89
 Internal structures, 62 processes, 29–30
 Invalid syntax transactions, 522
 INVALID_HANDLE_VALUE flag, 178
 Invoke COM function, 368
 IrA networking component, 316
 ISABOUT function, 649
 ISAM databases, 530–532
 IUnknown methods, reference counting, 366–367

J
 Join order, 489–493 alternatives, 493–494 hashes, 489, 492–493 merges, 489, 492 nested loops, 489, 491
 Join transaction if present workflow property, 850–851

K
 KeepIdentity property, 762–763
 KeepNulls property, 763–764
 Kernel processor mode, 26 executing stacks, 41–42 memory partitions, 118 partitions, 16 process components, 26 thread synchronization, 84 basics, 93 event objects, 94, 95 I/O completion ports, 98–99 mutexes, 94, 97–98 semaphores, 94, 96–97 signaled or unsignaled state, 93–95

wait functions, 94–95
 waitable timers, 94, 95–96
versus user processor mode, 14–15
 UMS scheduling, 420–421
 user timers, 96

Kernel32.DLL, 13, 17 static importing, 18 static linking, 415
 KEYSET cursors, 533, 537–539
 KPROCESS blocks, address translation, 125

L

Language processing and executions (LPEs), 417

Large-address-aware applications, 111, 122–123

Last processor affinity, 70–72

Layers, OSI reference model, 314 networking components, 316

Lazy Write Flushes/sec counter, 194

Lazy Write Pages/sec counter, 194

Lazywriter memory management basics, 443 checkpoint process, 446 computing physical memory, 443–444 flushing and freeing pages, 444–446

partitioning BPool free list, 446

LeaveCriticalSection API function, 85, 90

.LIB files, DLLs (Dynamic Link Libraries), 18

Lightweight threads, 50–51

Listen API function, 312

LOAD DATABASE command, 522

LOAD TRANSACTION command, 522

Local cursors

defaulting to, 564
versus global cursors, 551–552

Local partitioned views (LPVs), 658, 660

Locks, indexes, 478–479

RIDs (row identifiers), 478

Log Reader Agent, transactional replication, 923

MSrep1_command table, 924–925

procedures

sp_replcmds, 925–928

sp_repldone, 928

updated stored, 928–930

skipping errors, 938

Logical operators basics, 494–495
 DISTINCT, 495
 GROUP BY, 495
 ORDER BY, 496
 Spooling, 496

Logical Scan Fragmentation key indicator, 471

Lookup queries, 847–849

Low memory manager, 448

Lowest thread priority level, 65

LPEs (language processing and executions), 417

LPVs (local partitioned views), 658, 660

M

Main STAs (single-threaded apartments) COM models, 371

Main threads, 43 processes, 26

Managed classes, .NET Framework

SQL Server XML feature, 672

SQLXML, 769–772

Manual-reset events, 95

Mapped code, DLL and EXE files, 27

Mapping schemas

basics, 733–734

updategrams

GUIDs, 758

identity column values, 756–758

multiple rows, 754–756

nulls, 752

parameters, 752–754

XDR and XSD mapping schemas, 750–752

XDR

annotated, 737–738

basics, 734–737

XSD

basics, 738–739

default mapping, 739

explicit mapping, 740–741

sql:inverse annotation, 742–743

sql:key-fields annotation, 746–748

sql:limit-field annotation, 745–746

sql:limit-value annotation, 745–746

- sql:mapped annotation, 743–745
 sql:relationship annotation, 741–742
 MapUserPhysicalPages API function, 111, 121
 MapUserPhysicalPagesScatter API function, 121
 MapViewOfFile API function, 176, 289–290
 Marshaling
 COM, 367–368
 RPCs, 357
 Match rules, 798
 dynamic, 828–832
 MEM_COMMIT flag, 146
 Memory. *See also* Memory management, SQL Server; Memory management, Windows
 allocation granularities, 110, 133–134
 API functions, 111
 application memory tuning, 111, 122–124
 AWE, 111, 120–122
 basics, 109–110
 committed memory, 141
 heaps
 allocating memory, 166–167
 API functions, 165
 basics, 164–165
 large-address-aware applications, 111, 122–123
 memory management, 133
 memory-mapped files
 API functions, 289
 basics, 178, 289–291
 definition, 176
 PAE (Physical Address Extensions), 127
 page sizes, 110
 paging files, 110, 120
 partitions, 115–116
 boundary partitions, 119–120
 NULL pointer assignment, 111, 116–118
 PEBs (process environment blocks), 118–119
 private address space, 118
 TEBs (thread environment blocks), 119
 user data pages, 119
- physical
 address translation to virtual memory, 110, 124–127
 DLLs, 19
 management with AWE support, 436–438
 management with lazywriter, 443–446
 versus virtual memory, 15–16
 process address spaces, 110, 112–113, 115
 protecting, 114–115
 shared memory
 API functions, 176
 basics, 176
 image-mapped files, 178–179
 memory-mapped files, 176, 178
 section objects, 176, 177–178
 thrashing, 110
 threads, 15
 tools, 111–112
 Perfmon counters, 112
 /USEVA and 3GB boot options, 122–123
- virtual
 address spaces, 110
 address translation from physical memory, 110, 124–127
 allocation granularities, 133–134, 140
 API functions, 141
 AWE, 122
 basics, 140
 committed memory, 141, 145–146
 copy-on-write memory, 141
 definition, 141
 freeing, 147
 guard pages, 141–143
 locking pages, 148
 memory management, 133
 page sizes, 140
 pages, initializing and modifying, 146–147
 versus physical memory, 15–16
 physical storage of address ranges, 19
 reserved memory, 140, 144–145
- Memory management, SQL Server
 See also Memory
 BPool (buffer pool) memory region, 436
 with AWE and MemToLeave regions, 438–439
 BUF arrays, 442, 444–446
 finding BPool with WinDbg, 439–441
 hashing pages, 439
 page arrays, 441
 sizing, 436–438
 COM IMalloc interface, 449
 Connection memory manager, 447
 General memory manager, 447
 lazywriter
 basics, 443
 checkpoint process, 446
 computing physical memory, 443–444
 flushing and freeing pages, 444–446
 partitioning BPool free list, 446
 low memory manager, 448
 management process steps, 449–452
 MemToLeave (memory to leave) region, 436
 sizing, 436–438
 Optimizer memory manager, 447
 OS memory manager, 448
 physical memory region with AWE support
 basics, 442
 sizing, 436
 Query Plan memory manager, 447
 Utility memory manager, 447
- Memory management, Windows.
 See also Memory
 physical and virtual memory, 133
 SQL Server, memory regions, 436
- MEM_RESERVE flag, 146
 MEM_RESET flag, 147
 MemToLeave (memory to leave) region, 436
 sizing, 436–438
- Merge Agent, merge replication
 conflict resolution, 946–947
 filtering, 949–953
 generations, 947–949
 overview, 944

1010 Index

Merge replication
 identity range management, 953–955
Merge Agent
 conflict resolution, 946–947
 filtering, 949–953
 generations, 947–949
 overview, 944
NOT FOR REPLICATION
 option, 953–955
Snapshot Agent, overview, 944–946
Message formatting, 803
MessageBeep Win32 API function, 49
Microsoft Distributed Transaction Coordinator, 852
 updatable subscriptions
 snapshot replication, 916–918
 transaction replication, 933–934
Microsoft Indexing Service, overview, 632
Microsoft Search
 architecture, 632–633
 binary data, 635–636
 non-SQL Server data, 635
 overview, 632
Minimal perfect hashing functions, 493
MSrepl_command table, 924–925
Msvcrt.dll and Msvcrtdll, 415
MSXML (Microsoft XML) parser
 COM and SQL Server, 371
 finding server-side, 679–680
 processing XML
 with DOM APIs, 398–400
 with SAX (Simple API for XML) APIs, 400–404
MTAs (multithreaded apartments)
 COM models
 basics, 369–371
 COM objects and automation, 574–576
Multicast notifications, 804
Multiphase data pump
 DataDriven Query task, 839
 Parallel Data Pump task, 839
 Transform Data task, 839
Multiphase data pump, DTS (Data Transformation Services), 836–837

Multithreaded apartments (MTAs)
 COM models
 basics, 369–371
 COM objects and automation, 574–576
Multithreaded RTLs (runtime libraries), 46
Multithreading, 44
Mutexes, 94, 97–98

N

Name resolution, 312
Named pipes, 311
 networking, 317–319
 UNC (Universal Naming Convention), 318
 versus Winsock, 346–347
Native API functions, 17
Ndx Allocation Map (IAM), 467
NDIS library and miniport, 317
NDIS protocol drivers, 316
Nested transactions, 514–519
.NET Framework classes
 automation with COM interoperability, 588–591
 managed, SQL Server XML feature, 672
Net-Library (SQL Server), 18
 component, 417
 DDLS, 18
 networking, 412–414
 shared memory, 177
Network API DLLs (Dynamic Link Libraries), 316
Network API drivers, 316
Network application, 316
Network layer, OSI reference model, 314
 networking components, 316
Network stacks, 312
Networking
 API functions, 312–313
 components, 315–317
 name resolution, 312
 named pipes, 311, 317–319
 versus Winsock, 346–347
 network stacks, 312
 OSI reference model, 313–315
 RPCs, 312
 asynchronous, 355
 basics, 356–357
 marshaling, 357
 runtime, 358
 synchronous, 358
and SQL Server, 411–414
Winsock, 334, 344
 connection-oriented applications, 312, 335
 connectionless applications, 312, 343–344
 extensions, 346
 versus named pipes, 346–347
 sockets, definition, 311
Non XML query results, 722–723
Nonbuffered I/O, 200
 sectors, 200
Nonindexable expressions, row estimates, 488
Normal process priority level, 65
Normal thread priority level, 65
NOT FOR REPLICATION option, 953–955
Notification Services
 application development, 806, 808
 configuration files, 808–815
 creating custom database, 815–817
 creating instance and application databases, 817–818
 improvements, 828–832
 registering and enabling instance databases, 818–819
 starting service, 819
 subscription management, 819–827
 testing notifications, 827–828
 basics, 791–792
 configuration files, 793–794
 distributors
 basics, 802–803
 delivery of notifications, 803–804
 message formatting, 803
 multicast digest notifications, 804
 event providers, 796–797
 functions
 BNSInfoNotify function, 799
 NSInsertNotificationN stored procedure, 798–802
 generator
 basics, 797
 match rules, 798
 QuantumDuration node, 798

instance and application databases, 793
N
 NSControl utility, 792–793
 NSService.exe file, 794–796
 subscribers and subscriptions basics, 804–805
 CMD files, 806, 807
 sample applications, 805
 NSControl utility, 792–793
 application development configuration files, 808–815
 creating custom database, 815–817
 creating instance and application databases, 817–818
 registering and enabling instance databases, 818–819
 starting service, 819
 CMD files, 806, 807
 NSService.exe file, 794–796
 NTDLL.DLL, 17
 NTFS file system, 192–193
 NTOSKRNL.EXE file, 13, 17
 NULL pointer assignment partitions, 111, 116–118
 NULL values, updategrams, 763–764
 Nulls, 752
 NUMERIC_ROUNDABORT setting, 475

O

OBJECTPROPERTY function, 638
ODSOLE (Open Data Services Object Linking and Embedding) automation
 of COM objects COM objects in user-defined functions, 591–594
 easy binding *versus* late binding, 576–577
 .NET Framework classes with COM interoperability, 588–591
 threading models, 574–576
 VB array functions, 619–630
 of custom objects, 614–619
 overview, 573–574
 sp_OA procedures, 577–578
 dot notation for object name transversal, 580–581
 named parameters, 581
 sp_OACreate, 578–579

sp_OADestroy, 580
 sp_OAGetErrorInfo, 580
 sp_OAGGetProperty, 580
 sp_OAMethod, 579–580
 sp_OASetProperty, 580
 sp_OAStop, 580
 sp_OA procedures with other procedures, 582
 sp_checkspelling, 582
 sp_vbscript_reg_ex, 585–588
 of SQL-DMO COM objects sp_exporttable, 594–600
OLE (Object Linking and Embedding) sp_generate_script, 600–614
 COM, 361
 IIDs (interface IDs), 366
OleInitialize COM function, 369–370
OPEN command, 506 cursor variables, 566–568
 Open Group, 356
 Open Software Foundation, 356
 Opensd60.dll, 415
 OpenFile API function, 194–195
 OpenProcess API function, 27
OPENROWSET command, Transact-SQL, 861
OPENXML function basics, 698–700
 edge table format, 706–707
 flag parameters, 704–705
 inserting data, 707–711
 SQL Server XML feature, 672 using, 700–704
Optimize Synchronization, 950
Optimizer memory manager, 447
 Optimizing performance basics, 122–124
 definition, 111
 query processing basics, 457
 full optimization, 459–461
 optimization limits, 462
 simplification, 459
OS/2 Presentation Manager API, 13
OS/2 subsystem, 16–17
 OS kernel, 13
 OS memory manager, 448
OSI (Open Systems Interconnection), 313–315
 Out of order pages field, 473
 Overlapped I/O, 200

P

Packages, DTS (Data Transformation Services)
 connections, OLE DB data sources, 838
 creating, 837–838
 DSO Rowset provider, 861–862
 parameterized packages, 859–861
 tasks, 838–839
 workflow properties, 849–851
PAE (Physical Address Extensions), 127
 Page directory entries (PDEs), address translation, 125, 127
Page File Bytes counter, 112
Page File Peak counter, 112
Page frame numbers (PFNs), 126
Page table entries (PTEs) address translation from physical to virtual memory, 125, 127
 PFNs, 126
PAGE_EXECUTE protection attribute, 142–143
 shared memory, 179
PAGE_EXECUTE_READ protection attribute, 142
PAGE_EXECUTE_WRITECOPY protection attribute, 142–144
PAGE_GUARD protection attribute, 142–143, 155–157
PAGE_READONLY protection attribute, 122, 142
Pages BPool memory management arrays, 441
 page sizes, 110
 GetSystemInfo API function, 114
 memory, 110
 virtual memory, 140
 virtual memory initializing and modifying, 146–147
 page protection attributes, 141–143
Pages Scanned field, 473
PAGE_WRITECOPY protection attribute, 142, 144
 shared memory, 179
Paging files, 110, 120
 DDLs, 19
Paoli, Jean, 376
Parallel Data Pump task, 839

1012 Index

Parallel performance optimization, 461
 Parameter sniffing, query processing, 463–465
 Partitioned views
 DPV (distributed partitioned views), 668–669
 queries with BETWEEN predicate, 665–667
 Partitions, 115–116
 boundary, 119–120
 boundary partitions, 119–120
 NULL pointer assignment, 111, 116–118
 partitioning columns with CHECK constraint, 658–665
 PEBs, 118–119
 private address space, 118
 TEBs, 119
 user data pages, 119
 #PCDATA flag, 384
 PDB (Program Database) format, symbol files, 22
 PDEs (page directory entries), address translation, 125, 127
 PEBs (process environment blocks), 29, 30, 118–119
 Perfect hashing functions, 493
 Perfmon tool
 counters, 28
 DTSDIAG utility, 971–972, 981–983
 I/O, 194
 memory, 112
 processes, 28
 threads, 41
 I/O, 193
 memory, 112
 processes, 21, 28
 thread diagnostics, 21, 40
 thread scheduling, 62
 thread synchronization, 21, 86
 Performance optimization
 basics, 122–124
 definition, 111
 query processing
 basics, 457
 full optimization, 459–461
 optimization limits, 462
 simplification, 459
 trivial plan optimization, 458–459

PFNs (page frame numbers), 126
 Physical Address Extensions (PAE), 127
 Physical layer, OSI reference model, 314
 networking components, 316
 Physical memory. *See also* Memory management, SQL; Memory management, Windows; Virtual memory
 address translation to virtual memory, 110, 124–127
 AWE (Address Windowing Extensions), 121–122
 DLLs (Dynamic Link Libraries), 19
 management, 133
 management with AWE support, 436
 sizing, 436–438
 management with lazywriter
 basics, 443
 checkpoint process, 446
 computing physical memory, 443–444
 flushing and freeing pages, 444–446
 partitioning BPool free list, 446
 memory management, 133
 paging files, 110
 versus virtual memory, 15–16
 Physical operators
 basics, 494–495
 DISTINCT operator, 495
 GROUP BY operator, 495
 ORDER BY operator, 496
 Spooling operator, 496
 Physical storage, virtual memory
 address ranges, 19
 Pmon tool, 112
 POSIX subsystem, 16–17
 PostQueuedCompletionStatus API function, 99, 253
 .PPT (Microsoft PowerPoint) SQL Server support, 636
 Predicates, full-text, 456
 BETWEEN, 665–667
 basics, 642–643
 CONTAINS, 643–647
 FREETEXT, 647
 Preemptions, thread scheduling, 60
 Preemptive *versus* cooperative tasking, UMS (User Mode Scheduler), 421
 Presentation layer, OSI reference model, 314
 networking components, 316
 Primary threads, 43
 Priority Base counter
 processes, 41
 threads, 41
 Priority Current counter, 41
 Priority levels, thread scheduling, 64–66
 Private address space, 118
 Private Bytes counter, 112
 Private heaps. *See* Custom heaps
 % Privileged Time counter
 processes, 28
 threads, 41
 Processes, 25. *See also* QPs (query processors)
 address spaces, 25, 112–113, 115
 memory, 110
 affinity, 68–70
 thread scheduling, 60
 types, 70–72
 API functions, 27
 components, virtual address space, 26
 entry-point functions, 26
 IDs, 27
 initialization, 30
 internal structures, 29–30
 kernel mode, 26
 loading DLLs implicitly or explicitly, 18
 main threads, 26
 overview, 26–27
 PEBs, 29, 30, 118–119
 priority levels, 66
 changing base priorities, 66
 thread scheduling, 60
 processor modes, user *versus* kernel mode, 14–15
 SetErrorMode API function, 31–32
 terminating, 30
 threads, 15, 29
 versus threads, 43–44
 tools, 20–21, 27–28
 Perfmon counters, 28
 tweaking, foreground applications, 66

- user mode, 26
- VADS, 26
- % Processor Time counter
 - processes, 28
 - threads, 41
- Profiler, undocumented SQL features, 967–968
- Program Database (PDB) format, symbol files, 22
- Protocol drivers, 316
- Pseudo-Techie Tactics, 991–997
- Pseudohandle, 40
- PSS (process status structure), 447
- Pstat tool
 - memory, 112
 - processes, 28
 - thread diagnostics, 40
 - thread scheduling, 62
- PTEs (page table entries)
 - address translation from physical to virtual memory, 125, 127
 - PFNs, 126
- Pview tool
 - memory, 112
 - thread synchronization, 20, 86
- Pviewer tool
 - processes, 28
 - thread diagnostics, 20, 40
 - thread scheduling, 62
- Q**
 - QP_s (query processors). *See also* Processes
 - auto-parameterization, 465–466
 - cardinality, 456
 - density, 456
 - indexable expressions
 - basics, 486
 - folding, 488–489
 - operator translations, 486–488
 - indexes
 - basics, 466
 - on computed columns, 474–478
 - covering, 469
 - fragmentation, 470–473
 - intersection, 470
 - locking, 478–479
 - performance issues, 469–470
 - storage, 467–469
 - type selection, 489–493
 - on views, 474–478
- QueryInterface IUnknown COM method, 367
- Queue Reader Agent, 918
- Queued updating subscribers
 - snapshot replication, 918–919
 - transaction replication, 934
- Queues, 67–68
- Quick plan, performance optimization, 460
- QUOTED_IDENTIFIER setting, 475
- join order, 489–493
 - alternatives, 493–494
- logical operators, 494–495
 - DISTINCT operator, 495
 - GROUP BY operator, 495
 - ORDER BY operator, 496
 - Spooling operator, 496
- nonindexable expressions, row estimates, 488
- parameter sniffing, 463–465
- parsing, 456
- performance optimization
 - basics, 457
 - full optimization, 459–461
 - limits, 462
 - simplification, 459
 - trivial plan optimization, 458–459
- physical operators, 494–495
- predicates, 456
- selectivity, 456
- SQL Server component, 417
- statistics, 479–481
 - cardinality, 479
 - columns, 481–482
 - density, 479, 480–481
 - listing, 482
 - selectivity, 479–480
 - selectivity of rows, estimation, 485
 - storage, 481
 - updating, 482–485
- subqueries, 493–494
- Oslice tool
 - processes, 28
 - thread diagnostics, 40
- Quantum units, thread scheduling, 60, 62–63
- QuantumDuration node, 798
- Query Plan memory manager, 447
- Replication, snapshot cleanup, 920
- Distribution Agent
 - basics, 908
 - remote agent activation, 919–920
 - tasks, 915–916
- Snapshot Agent
 - basics, 908
 - tasks, 912–915
 - using, 908–912
- updatable subscriptions
 - immediate updating, 916–918
 - immediate updating subscribers with queued updating as failover, 919
 - queued updating, 918–919
- Replication, transactions
 - cleanup, 939
 - concurrent snapshot processing, 930–932
 - Distribution Agent, 923
 - skipping errors, 938

1014 Index

Replication, snapshot, *continued*
 Log Reader Agent, 923
 MSrepl_commands table,
 924–925
 skipping errors, 938
 sp_replcmds procedure,
 925–928
 sp_repldone procedure, 928
 updated stored procedures,
 928–930
 Snapshot Agent, 923
 undocumented SQL features, 961
 updatable subscriptions, 932
 immediate updating, 933–934
 immediate updating with
 queued updating as
 failover, 935
 queued updating, 934
 validating replicated data,
 935–938
 Replication subscriptions, 862–864
 #REQUIRED flag, 384
 Reserved memory
 basics, 144–145
 definition, 140
 Reserved memory manager, 448
 ResetEvent API function, 85, 95
 RESTORE DATABASE command,
 522
 RESTORE LOG command, 522
 ResumeThread API function, 72–73
 REVOKE command, 506
 RIDs (row identifiers), indexes, 469
 locking, 478
 Role reversals, 493
 ROLLBACK TRAN command, 506,
 514–522
 accidental rollbacks, 520
 user-defined transactions,
 503–504
 Row identifiers (RIDs), indexes, 469
 locking, 478
 Row-oriented operations, 543–546
 ROWGUID_COL flag, 945
 Rowset functions
 basics, 648
 CONTAINSTABLE, 648–651
 FREETEXTTABLE, 651–652
 Rpct.dll, 415
 RPCs (Remote Procedure Calls),
 312
 asynchronous, 358
 marshaling, 357

networking, basics, 356–357
 runtime, 358
 synchronous, 358
 RTLs (runtime libraries)
 multi-threaded, 46
 process termination, 30
 single-threaded, 46
 Run on client option, 714
 Runnable list, UMS (User Mode
 Scheduler), 425–426
 Running state, 64
 Runtime libraries. *See* RTLs
 Runtime RPCs (Remote Procedure
 Calls), 358

S

SARGs (search arguments in que-
 ries), 486
 SAVE TRAN command, 519–520
 save points, 519–520
 SAX (Simple API for XML) APIs
 MSXML parser, 673–675
 processing XML with MSXML
 parser, 400–404
 Scatter-gather I/O, 233–234
 API functions, 235
 clusters, 234
 overview, 235–236
 sectors, 234
 Scheduling threads, 67. *See also*
 UMS (User Mode Scheduler)
 API functions, 61
 basics, 59–60
 clock intervals, 60, 63
 context switching, 15, 60, 68
 counters, Thread State, 60,
 63–64
 values, 64
 diagnostic tools, 20–21, 62
 ideal processors, 60
 internal structures
 basics, 62
 quantum units, 62–63
 preemptions, 60
 process affinity, 60, 68–70
 process priority levels, 64–66
 putting threads to sleep, 73–74
 quantum units, 60
 real-time threads, 67
 selecting threads, 72
 suspending/resuming threads,
 72–73
 thread priority levels, 60

thread starvation, 60
 tools, 20–21, 62
 SchemaGen property, 766–769
 Searches, full-text
 binary data, 635–636
 columns, 637–642
 DBCC CALLFULLTEXT
 command, 633
 functions, 634–635
 filters, 635–636
 functions, DBCC CALLFULL-
 TEXT command,
 634–635
 Microsoft Indexing Service,
 overview, 632
 Microsoft Search
 architecture, 632–633
 overview, 632
 non-SQL Server data, 635
 rowset functions
 basics, 646
 CONTAINSTABLE rowset
 function, 648–651
 FREETEXTTABLE rowset
 function, 651–652
 Section objects, shared memory,
 basics, 177–178
 Sectors, I/O, 200
 scatter-gather I/O, 234
 SEH (structured exception han-
 dling), 48–49
 threads, 39
 SELECT command, 506
 SELECT INTO command, 504
 SELECT...FOR XML queries
 client-side, 696–698
 server side, 680–681
 Selectivity, query processing, 456,
 479–480
 of rows, estimation, 485
 Semaphores, 94, 96–97
 SEM_FAILCRITICALERRORS
 flag, 31
 SEM_NOALIGNMENTFAULTEX
 CEPT flag, 31
 SEM_NOGPFAULTERRORBOX
 flag, 31
 SEM_NOOPENFILEERRORBOX
 flag, 31
 Send API function, 312
 SERIALIZABLE TIL (transaction
 isolation level) control,
 512–513

- Serialization of memory heaps
 basics, 168–169
 definition, 165
- SEs (storage engines), SQL Server
 component, 417
- Session layer, OSI reference model,
 314
 networking components, 316
- SET IMPLICIT_TRANSACTIONS
 command, 502–503
- SET XACT_ABORT command,
 506–508
- SetCriticalSectionSpinControl
 function, 90
- SetErrorMode API function,
 31–32
- SetEvent API function, 85, 95, 98
- SetEvent/ResetEvent API function,
 85
- SetLastError API function, 49
- SetPriorityClass API function, 61
- SetProcessAffinityMask API
 function, 61
- SetProcessPriorityBoost API
 function, 61
- SetThreadAffinityMask API
 function, 61, 70
- SetThreadIdealProcessor API
 function, 61, 70
- SetThreadPriority API function,
 61
- SetThreadPriorityBoost API
 function, 61
- SetWaitableTimer API function, 85,
 95–96
- SetWorkingSetSize API function,
 141
- Shared memory
 API functions, 176
 basics, 176
 image-mapped files, 178–179
 memory-mapped files
 basics, 178
 definition, 176
- Net-Library, 177
- section objects
 basics, 177–178
 definition, 176
- SQL Server, 177
 viewing objects with Super-
 Recorder, 184–188
- Sigaled state, 94–95
 thread synchronization, 84
- Simple API for XML (SAX) APIs
 MSXML parser, 673–675
 processing XML with MSXML
 parser, 400–404
- Simplification stage, performance
 optimization, query processing,
 459
- Single-threaded RTLs (runtime
 libraries), 46
- Single-user database, transactions,
 505–506
- Sleep API function, 61, 73
- SleepEx API function, 61, 73
 basics, 908Snapshot Agent
 merge replication, overview,
 944–946
- snapshot replication
 basics, 908
 tasks, 912–915
 using, 908–912
- transactional replication, 923
- Snapshot replication
 cleanup, 920
- Distribution Agent
 basics, 908
 remote agent activation,
 919–920
 tasks, 915–916
- Snapshot Agent
 basics, 908
 tasks, 912–915
 using, 908–912
- updatable subscriptions
 immediate updating, 916–918
 immediate updating subscrib-
 ers with queued updating
 as failover, 919
 queued updating, 918–919
- SOAP support, 772–773
 SQL Server XML feature, 672
- Socket API function, 312
- Sockets. *See* Winsock
- Soft processor affinity. *See* Last
 processor affinity
- Source paths, WinDbg, 22
- Sp_checksplelling procedure, 582
- Special characters, URL (Uniform
 Resource Locator), 719–720
- Sperberg-McQueen, C.M., 376
- Sp_exporttable procedure,
 594–600
- Sp_generate_script procedure,
 600–614
- Spinlock constructs, 86–87
 CPU usage, 87–88
 user mode, 84
- Sp_OA procedures, automation
 dot notation for object name
 transversal, 580–581
 named parameters, 581
 with other procedures, 582
- sp_OACreate, 578–579
- sp_OADestroy, 580
- sp_OAGetErrorInfo, 580
- sp_OAGetProperty, 580
- sp_OAMethod, 579–580
- sp_OASetProperty, 580
- sp_OAStop, 580
- Sp_replcmds procedure, 925–928
- Sp_repldone procedure, 928
- Sp_run_xml_proc procedure,
 782–788
- Sp_vbscript_reg_ex, 585–588
- Sp_xml_concat procedure, 778–782
- Spy++ tool, thread synchronization,
 86
- SQL-DMO (Distributed Manage-
 ment Objects) and COM
 objects
- basis for SQL-DMO, 371
- ODSOLE automation
 sp_exporttable procedure,
 594–600
 sp_generate_script procedure,
 600–614
- SQL Server
 COM, 371–372
 components, 417
 DLLs (Dynamic Link Libraries),
 414–415
- Net-Libraries, 18
- undocumented features
 basics, 961–963
 import tables, 968–969
 investigating installation
 scripts, 968
 recording stored procedure ex-
 ecutions and batches with
 Profiler, 967–968
- scripting undocumented and
 system objects, 966–967
- syscomments system table,
 963–964
- sysobjects system table,
 964–966
- versions, xxxvii

1016 Index

SQL Server federation
 DPVs (distributed partitioned views), 668–669
 partitioned views, 657–665
 DPVs (distributed partitioned views), 668–669
 queries with BETWEEN predicate, 665–667
 SQLDIAG file, DTSDIAG utility, 971, 980–983
 Sql:inverse annotation, 742–743
 SQLISAPI, URL (Uniform Resource Locator) queries
 basics, 714–715
 contenttype URL query parameter, 720–722
 non XML query results, 722–723
 special characters, 718–719
 stored procedures, 723–724
 using, 715–718
 xsl style sheets, 719–720
 Sql:key-fields annotation, 746–748
 Sql:limit-field annotation, 745–746
 Sql:limit-value annotation, 745–746
 Sql:mapped annotation, 743–745
 Sql:relationship annotation, 741–742
 Sqldsort.dll, 415
 SQLXML
 FOR XML extension
 AUTO mode, 682–685
 basics, 677–679
 EXPLICIT mode, 685–696
 RAW mode, 681–682
 SELECT...FOR XML (client side), 696–698
 SELECT...FOR XML (server side), 680–681
 HTTP SQL Server access
 basics, 712–714
 virtual directory configuration, 712–714
 limitations
 sp_run_xml_proc procedure, 782–788
 sp_xml_concat procedure, 778–782
 managed classes, 672, 769–772
 mapping schemas
 basics, 733–734
 XDR (XML-Data Reduced), 734–737
 XSD (XML Schema Definition), 738–739

MSXML parser, 673–675
 finding server-side, 679–680
 OPENXML function, 672
 basics, 698–700
 edge table format, 706–707
 flag parameters, 704–705
 inserting data, 707–711
 using, 700–704
 overview, 672–673
 SOAP support, 672
 template queries
 applying style sheets on clients, 730–731
 basics, 725–726
 client side templates, 731–733
 parameterized templates, 726–727
 style sheets, 727–730
 updategrams, 672
 basics, 748–750
 URL queries, 714–721
 basics, 714–715
 contenttype URL query parameter, 720–722
 non-XML query results, 722–723
 special characters, 718–719
 stored procedures, 723–724
 using, 715–718
 xsl style sheets, 719–720
 Web Services (SOAP) support, 772–773
 XML Bulk Load component, 672
 basics, 759
 CheckConstraints property, 761
 ErrorLogFile property, 766
 ForceTableLock property, 764
 IgnoreDuplicateKeys property, 761–762
 KeepIdentity property, 762–763
 KeepNulls property, 763–764
 Null values, 763–764
 SchemaGen property, 766–769
 Transaction property, 764–766
 using, 759–760
 XML fragment property, 759–760
 SQLXMLOLEDB, COM and SQL Server, 371–372
 SRVPROC structure, 447
 Stacks, executing in kernel and user processor modes, 41–42
 Standby state, 64
 STAs (single-threaded apartments)
 COM models
 basics, 369–371
 COM objects and automation, 574–576
 STATIC cursors, 533, 536–537
 Statistics, query processing
 cardinality, 479
 columns, 481–482
 density, 479, 480–481
 listing, 482–485
 query processing, 479–481
 selectivity, 479–480
 of rows, estimation, 485
 storage, 481
 updating, 482–485
 STATUS_ACCESS_VIOLATION flag, 166
 STATUS_NO_MEMORY flag, 166
 Step priority property, 850
 Storage
 indexes, 467–469
 clustered indexes, 467–469
 nonclustered indexes, 467–469
 physical storage, virtual memory address ranges, 19
 query processing statistics, 481
 threads, TLS (thread local storage) area, 39
 Storage engines (SEs), SQL Server component, 417
 Stored procedures, 723–724
 Structured exception handling (SEH), 48–49
 threads, 39
 Style sheets, 727–730
 applying on clients, 730–731
 XSLT, converting XML to HTML, 389–397
 Subqueries, 493–494
 Subscribers and subscriptions
 basics, 804–805
 CMD files, 806, 807
 sample applications, 805
 Subsystems
 components, 17
 DLLs (Dynamic Link Libraries), 17
 SuperRecorder application, 184–188

- SuspendThread API function, 61, 72–73
 SwitchToThread API function, 61, 74
 Symbol files, SQL Server, 22
 Symbol paths, WinDbg, 22
 Synchronizing threads
 API functions, 85
 atomic access, 85
 basics, 84, 86
 critical section, 89–91
 deadlocks, 84
 diagnostic tools, 86
 event objects, 94, 95
 I/O completion ports, 98–99
 interlocked functions, 84, 88–89
 kernel synchronization objects, 84
 mutexes, 94, 97–98
 semaphores, 94, 96–97
 signaled state, 84, 93–95
 spinlock constructs, 86–87
 CPU usage, 87–88
 user mode, 84
 thread deadlocks, 91–93
 thread-safe code, 85
 tools, 20–21, 85–86
 unsignaled state, 84, 93–95
 user timers, 96
 wait functions, 84, 94–95
 wait states, 91
 waitable timers, 94, 95–96
 Synchronous I/O, 193, 195
 Synchronous I/O, 193, 195
 networking, 346
 Synchronous RPCs (Remote Procedure Calls), 358
- T**
 Task Manager tool
 I/O, 193
 memory, 110–111
 process tweaking, 66
 processes, 28
 setting processor affinity, 70
 thread scheduling, 62
 Tasks, DTS (Data Transformation Services), 838–839
 TB (Translation Buffer), 126
 TEBs (thread environment blocks), 42, 119
 Template queries
 applying style sheets on clients, 730–731
 basics, 725–726
 client side templates, 731–733
 parameterized templates, 726–727
 style sheets, 727–730
 Templates, converting XML to HTML, 389–397
 Terminated state, 64
 TerminateProcess API function, 27, 30
 TerminateThread API function, 40, 45
 32-bit Windows, 13
 Thrashing, 110
 memory, 110
 Thread environment blocks (TEBs), 42
 Thread scheduling, 67
 API functions, 61
 basics, 59–60
 clock intervals, 60, 63
 context switching, 15, 60, 68
 counters, Thread State, 60, 63–64
 values, 64
 diagnostic tools, 20–21, 62
 ideal processors, 60
 internal structures
 basics, 62
 quantum units, 62–63
 preemptions, 60
 process affinity, 60, 68–70
 process priority levels, 64–66
 putting threads to sleep, 73–74
 quantum units, 60
 real-time threads, 67
 selecting threads, 72
 suspending/resuming threads, 72–73
 thread priority levels, 60
 thread starvation, 60
 tools, 20–21, 62
 Thread synchronization
 API functions, 85
 atomic access, 85
 basics, 84, 86
 critical section, 89–91
 deadlocks, 84
 diagnostic tools, 86
 event objects, 94, 95
 I/O completion ports, 98–99
 interlocked functions, 84, 88–89
 kernel synchronization objects, 84
 mutexes, 94, 97–98
 semaphores, 94, 96–97
 signaled or unsignaled state, 93–95
 signaled state, 84
 spinlock constructs, 86–87
 CPU usage, 87–88
 user mode, 84
 thread deadlocks, 91–93
 thread-safe code, 85
 tools, 20–21, 85–86
 unsignaled state, 84
 user timers, 96
 wait functions, 84, 94–95
 wait states, 91
 waitable timers, 94, 95–96
 Thread Wait Reason counter, 41
 Threads
 API functions, 40
 context structure, 42
 context switching, 39
 counters
 Thread Count, 28
 Thread State, 60
 thread scheduling, 63–64
 threads, 41
 creating/destroying, 44–45
 deadlocks, 91
 diagnostic tools, 20–21, 40
 entry-point functions, 39
 exception handling, 48–49
 fiber mode, 39, 50–51
 versus fibers, 50–51
 functions, 47–48
 GetLastError API function, 49–50
 IDs, 42
 initializing stacks, 44
 internal structure
 TLS area, 39
 tools, 40
 Perfmon counters, 41
 virtual memory, 15
 volatile registers, 15
 internal structures
 CPU registers, 41
 stack execution in kernel or user processor modes, 41–42
 TLS area, 42
 main, 26
 MTAs COM models
 basics, 369–371
 COM objects and automation, 574–576

1018 Index

Threads, *continued*
 object handles, 45
 Perfmon tool counter, 41
 primary threads, 43
versus processes, 43–44
 SEH, 39
 STAs COM models
 basics, 369–371
 COM objects and automation, 574–576
 terminating, 45–46
 thread starvation, thread scheduling, 60
 threading models
 COM, 369–371
 COM objects and ODSOLE automation, 574–576
/3GB boot option, application memory tuning, 122–123
AWE (Address Windowing Extensions), 123–124
TILs (transaction isolation levels)
 controls, 508–510
Time-critical thread priority level, 65
Timer list, UMS (User Mode Scheduler), 428
TLB (Translation Look-aside Buffer), 126
TList tool
 memory, 112
 processes, 20, 28
 thread diagnostics, 20, 40
TLS (thread local storage), 39
TlsAlloc API function, 39
TlsGetValue API function, 39
TlsSetValue API function, 39
Transact-SQL, 533
 COLUMNPROPERTY function, 638
 commands, 514–522
 cursors
 appropriate use, 539–540
 asynchronous, 559–561
 automatic closing, 561–564
 CLOSE command, 546, 555
 configuring, 558–559
 @@CURSOR_ROWS command, 546, 552–553
 CURSOR_STATUS command, 546
 DEALLOCATE command, 546, 558, 566–568

DECLARE CURSOR command, 546, 550, 551–552
 defaulting to global/local cursors, 564
 DYNAMIC, 533, 535–536
 dynamic queries, 540–543
 FETCH command, 546, 553–558
 @@FETCH_STATUS command, 546, 553
 FORWARD_ONLY, 533–534
 KEYSET, 533, 537–539
 OPEN command, 546, 552–553
 performance optimization, 569–571
 row-oriented operations, 543–546
 scrollable forms, 546
 sp_cursor_list procedure, 569
 sp_describe_cursor procedure, 569
 sp_describe_cursor_columns procedure, 569
 sp_describe_cursor_tables procedure, 569
 STATIC, 533, 536–537
 stored procedures, 568–569
 types, 533–539
 updating, 565–566
 variables, 566–568
 data relationship unions, 690–692
 full-text predicates
 basics, 642–643
 CONTAINS predicate, 643–647
 FREETEXT predicate, 647
 full-text searches
 CONTAINSTABLE rowset function, 648–651
 FREETEXTTABLE rowset function, 651–652
 rowset functions, 648
 FULLTEXTCATALOGPROPERTY function, 638
 OBJECTPROPERTY function, 638
 ODSOLE automation
 sp_OA procedures, 577–582
 sp_OA procedures with other procedures, 582–585
 of SQL-DMO COM objects, 594–614
 ODSOLE automation of COM objects
 COM objects in user-defined functions, 591–594
 easy binding *versus* late binding, 576–577
 .NET Framework classes with COM interoperability, 588–591
 threading models, 574–576
 VB array functions, 619–630
 ODSOLE automation of custom objects, 614–619
 OPENROWSET command, 861
 OPENXML function
 basics, 698–700
 edge table format, 706–707
 flag parameters, 704–705
 inserting data, 707–711
 using, 700–704
 transactions
 ACID (atomic, consistent, isolated, and durable) test, 499–501
 automatic, 502
 debugging, 522–524
 distributed, 504
 Distributed Transaction Coordinator, 504
 illegal inside transactions, 522
 implicit, 502–503
 invalid, 522
 minimizing logging, 504–505
 nested, 514–519
 optimizing code, 524–526
 performance optimization, 459
 processes, 501–502
 read-only database, 505–506
 recovery models, 505
 single-user database, 505–506
 syntax, 514–522
 transaction management, 506–508
 user-defined, 503–504
 undocumented SQL features, 962
 Transaction isolation levels (TILs)
 controls, 508–510
 Transaction property, 764–766
 Transactional replication
 cleanup, 939
 concurrent snapshot processing, 930–932

- Distribution Agent, 923
 skipping errors, 938
- Log Reader Agent, 923
 MSrepl_commands table, 924–925
 skipping errors, 938
 sp_replcmds procedure, 925–928
 sp_repldone procedure, 928
 updated stored procedures, 928–930
- Snapshot Agent, 923
- undocumented SQL features, 961
- updatable subscriptions, 932
 immediate updating, 933–934
 immediate updating with queued updating as failover, 935
 queued updating, 934
 validating replicated data, 935–938
- Transactions
 ACID (atomic, consistent, isolated, and durable) test, 499–501
 automatic, 502
 commands, 514–522
 debugging, 522–524
 distributed, 504
 Distributed Transaction Coordinator, 504
 illegal inside transactions, 522
 implicit, 502–503
 minimizing logging, 504–505
 nested, 514–519
 optimizing code, 524–526
 performance optimization, 459
 read-only database, 505–506
 recovery models, 505
 single-user database, 505–506
 syntax, 514–522
 invalid, 522
 transaction management, 506–508
 user-defined, 503–504
- Transform Data task, 839
- Transform Published Data wizard, 862–864
- Transform-SQL, 842–843
 BULK INSERT command, 842–843
- Transformers, XSLT, 389–397
- Transition state, 64
- Translation Buffer (TB), 126
- Translations of process address spaces, 110, 124–127
- PDEs (page directory entries), 125, 127
- PTEs (page table entries), 125, 127
- TB (Translation Buffer), 126
- Transport Driver Interface, 316TLB (Translation Look-aside Buffer), 126
- Transport layer, OSI reference model, 314
 networking components, 316
- Trivial plan optimization, query processing, 458–459
- TRUNCATE TABLE command, 504, 506
- TryEnterCriticalSection API function, 90
- .TXT (text files) SQL Server support, 636
- U**
- UMS (User Mode Scheduler). *See also* Thread scheduling
 DBCC SQLPERF (umsstats), 430, 431–432
 design goals, 420
 fiber mode, 430
 hidden schedulers, 431
 I/O list, 426–428
 Idle loops, 428–429
 preemptive *versus* cooperative tasking, 421
 runnable list, 425–426
 scheduling process, 421
 SQL Server component, 417
 timer list, 428
 user *versus* kernel process mode scheduling, 420–421
 waiter list, 426
 worker list, 423–425
- Ums.dll, 415
- Umsstats (DBCC SQLPERF), UMS (User Mode Scheduler), 430, 431–432
- UNC (Universal Naming Convention), named pipes, 318
- Unix server applications, fiber mode, 50–51
- Unknown state, 64
- Unsignaled state, 94–95
 thread synchronization, 84
- Updatable subscriptions
 snapshot replication
 immediate updating, 916–918
 immediate updating subscribers with queued updating as failover, 919
 queued updating, 918–919
 transactional replication, 932
 immediate updating, 933–934
 immediate updating with queued updating as failover, 935
 queued updating, 934
- UPDATE command, 506, 565–566
 WHERE CURRENT OF clause, 565–566
- UPDATE STATISTICS command, 483–484, 522
- Updated stored procedures, 928–930
- Updategrams
 basics, 748–750
 GUIDs (Globally Unique Identifiers), 758
 identity column values, 756–758
 multiple rows, 754–756
 nulls, 752
 parameters, 752–754
 SQL Server XML feature, 672
- XDR (XML Data-Reduced) and XSD (XML Schema Definition) mapping schemas, 750–752
- UPDATETEXT command, 504
- URL (Uniform Resource Locator)
 queries, 713
 basics, 714–715
 contenttype URL query parameter, 720–722
 non XML query results, 722–723
 special characters, 718–719
 stored procedures, 723–724
 using, 715–718
 xsl style sheets, 719–720
- Use transactions package property, 850–851
- User data pages, 119
- User-defined functions, automation of COM objects, 591–594
- User processor mode, 26
 executing stacks, 41–42
 versus kernel process mode, UMS scheduling, 420–421

1020 Index

User processor mode, *continued versus* kernel processor mode, 14–15
 memory partitions, 115–116
 boundary partitions, 119–120
 NULL pointer assignment, 116–118
 PEBs (process environment blocks), 118–119
 private address space, 118
 TEBs (thread environment blocks), 119
 user data pages, 119
 thread synchronization
 basics, 86
 critical section, 89–91
 interlocked functions, 88–89
 spinlock constructs, 86–87
 spinlock constructs and CPU use, 87–88
 thread deadlocks, 91–93
 wait states, 91
% User Time counter
 processes, 28
 threads, 41
 User timers, 96
 User32.DLL, 13, 17–18
 static linking, 415
 statically importing, 18
//USEVA boot options, 122–123
 application memory tuning, 122
 Utility memory manager, 447

V
 VADs (virtual address descriptors), 26, 144
 Valid XML documents, 382–383
 VBinStrRev function, 618–619
 VBODSOLELib file, 619–620
 Versions, SQL Server, xxvii
 Virtual Bytes counter, 112
 Virtual directory configuration options, 712–714
 Allow Post, 713–714
 Allow XPath, 713
 Expose runtime errors as HTTP error, 714
 Run on client, 714
 URL (Uniform Resource Locator) queries, 713

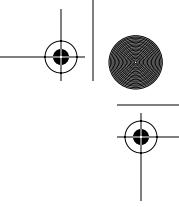
Virtual memory. *See also* Memory management; Memory management, Windows; Physical memory; SQL Server
 address spaces, 110
 VADs (virtual address descriptors), 26, 144
 address translation from physical memory, 110, 124–127
 allocation granularities, 133–134, 140
 API functions, 141
 AWE (Address Windowing Extensions), 122
 basics, 140
 committed memory
 basics, 145–146
 definition, 141
 copy-on-write memory, definition, 141
 freeing, 147
 guard pages
 definition, 141
 page protection attributes, 141–143
 locking pages, 148
 memory management, 133
 pages
 initializing and modifying, 146–147
 sizes, 140
versus physical memory, 15–16
 physical storage of address ranges, 19
 reserved memory, 140
 basics, 144–145
 definition, 140
 threads, 15
 VirtualAlloc API function, 111, 140, 141
 committed memory, 145–146
 page protection attributes, 142
 VirtualAllocEx API function, 140, 141
 VirtualFree API function, 141
 freeing memory, 147
 VirtualLock API function, 141
 freeing memory, 147
 locking pages, 148
 VirtualProtect API function, 141
 freeing memory, 147
 VirtualQuery(Ex) API function, 141

VirtualUnlock API function, 141
 Visual C++ 6.0, xxvi
 Volatile registers, 15
W
 Wait functions, 94–95
 thread synchronization, 84
 Wait states, 64, 91
 Waitable timers, 94, 95–96
 Waiter list, UMS (User Mode Scheduler), 426
 WaitForMultipleObjects API function, 85, 94–95, 98
 WaitForSingleObject API function, 30, 85, 94–95, 177
 WaitForSingleObjectEx API function, 201
 Well-formed XML documents, 382–383
 Win32 API, genesis, 14
 Win32 subsystem, 16–17
 WinDbg tool, 21–23
 basics, xxxii–xxxiii
 BPool memory management, 439–441
 source and symbol paths, 22
 Windows
 DLLs (Dynamic Link Libraries), 17–20
 kernel processor mode, *versus* user processor mode, 14–15
 not windows NT family
 Windows ME, 16
 Windows 9x, 16
 OS/2 subsystem, 16–17
 processes, 15
 threads, 15
 tools, 20–23
 virtual memory, *versus* physical memory, 26
 Win32 API, 13–14
 Win32 subsystem, 16–17
 Windows NT family, BOOT.INI switches, 16
 Win32k.sys kernel mode device driver, 17, 30
 Winsock, 344
 versus BSD (Berkeley Software Distribution), 316, 334
 connection-oriented applications, 335

- connectionless applications, 343–344
- extensions
 - asynchronous I/O, 346
 - synchronous I/O, 346
- versus* named pipes, 346–347
- sockets, definition, 311
- Worker list, UMS (User Mode Scheduler), 423–425
- WRITE TEXT command, 504
- WriteFile API function, 193, 312
- WriteFile transformation, 846–847
- WriteFileEx API function, 201–204
- WriteFileScatter API function, 235
- WriteProcessMemory API function, 111
 - memory protection, 115
- WriteProcessMemory API functions, 29
- WSASocket API function, 312
- WSAStartup API function, 312
- W32THREAD structure, 42
- X**
 - XDR (XML-Data Reduced) mapping schemas
 - annotated, 737–738
 - basics, 734–737
 - updategrams, 750–752
 - XLS (Microsoft Excel) SQL Server support, 636
 - XML Bulk Load component basics, 759
 - CheckConstraints property, 761
 - ErrorLogFile property, 766
 - ForceTableLock property, 764
 - IgnoreDuplicateKeys property, 761–762
- KeepIdentity property, 762–763
- KeepNulls property, 763–764
- Null values, 763–764
- SchemaGen property, 766–769
- SQL Server XML feature, 672
- Transaction property, 764–766
 - using, 759–760
- XML fragment property, 759–760
- XML-Data Reduced (XDR) mapping schemas
 - annotated, 737–738
 - basics, 734–737
 - updategrams, 750–752
- Xml directive, 688–690
- XML (Extensible Markup Language)
 - DOM, 397–398
 - DTDs, 383–385
 - fragment property, 759–760
 - history, 376–377
 - HTML (HyperText Markup Language)
 - advantages/disadvantages, 377–380
 - converting XML to HTML, 389–397
 - disadvantages, 375–376
 - terminology, 380–382
 - overview, 374–375
 - well-formed and valid documents, 382–383
- XML Schemas, 386–389
- XPath, 391
- XML Schema Definition (XSD)
 - mapping schemas
 - annotations
 - sql:inverse, 742–743
 - sql:key-fields, 746–748
- sql:inverse, 742–743
- sql:key-fields, 746–748
- sql:limit-field, 745–746
- sql:limit-value, 745–746
- sql:mapped, 743–745
- sql:relationship, 741–742
- basics, 738–739
- mapping
 - default, 739
 - explicit, 740–741
 - updategrams, 750–752
- XML Schemas, 386–389
- XMLDATA option, FOR EXPLICIT clause, 694–696
- XmItext directive, 688–690
- XPath (XML Path), 391
- XSD (XML Schema Definition)
 - mapping schemas
 - annotations
 - sql:inverse, 742–743
 - sql:key-fields, 746–748
 - sql:limit-field, 745–746
 - sql:limit-value, 745–746
 - sql:mapped, 743–745
 - sql:relationship, 741–742
 - basics, 738–739
 - mapping
 - default, 739
 - explicit, 740–741
 - updategrams, 750–752
- XSLT (Extensible Stylesheet Language Transformations)
 - converting XML to HTML, 389–397
 - embedded scripting, 397
 - style sheets, URLs, 719–720
 - xsl:attribute construct, 397
 - xsl:choose construct, 397
 - xsl:if construct, 397
 - xsl:sort construct, 397

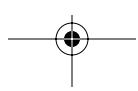
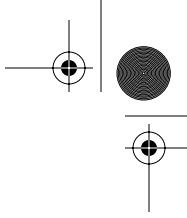


Henderson_book.fm Page 1022 Thursday, September 25, 2003 5:23 AM



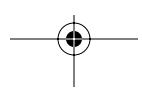
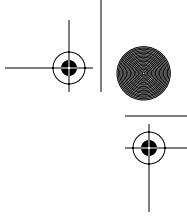


Henderson_book.fm Page 1023 Thursday, September 25, 2003 5:23 AM





Henderson_book.fm Page 1024 Thursday, September 25, 2003 5:23 AM





Henderson_book.fm Page 1025 Thursday, September 25, 2003 5:23 AM

