Glossary

(from *Windows Internals,* 4th edition by Mark E. Russinovich and David A. Solomon, Microsoft Press, 2004)

access-control list (ACL)
The part of a security descriptor that enumerates who has what access to an object. The owner of an object can change the object's ACL to allow or disallow others access to the object. An ACL is made up of an ACL header and zero or more access-control entry (ACE) structures. An ACL with zero ACEs is called a null ACL and indicates that no user has access to the object.

access token
A data structure that contains the security identification of a process or a thread, which includes its security ID (SID), the list of groups that the user is a member of, and the list of privileges that are enabled and disabled. Each process has a primary access token that it inherits by default from its creating process.

add-device routine
A routine implemented by drivers that supports Plug and Play. The Plug and Play manager sends a driver notification via this routine whenever a device for which the driver is responsible is detected. In this routine, a driver typically allocates a device object to represent the device.

Address Windowing Extensions (AWE)
A mechanism in Windows that allows a 32-bit application to allocate up to 128 GB of physical memory and then map views, or windows, into its 2-GB virtual address space. Using AWE puts the burden of managing mappings of virtual-to-physical memory on the programmer but solves the immediate need of being able to directly access more physical memory than can be mapped at any one time in a 32-bit process address space.

affinity mask
A bitmask that specifies the processors on which the thread is allowed to run. The initial thread affinity mask is inherited from the process affinity mask.

aging
A process performed on a page that increments a count indicating that the page hasn't been referenced since the last working set trim scan. On a single-processor system, the working set manager tries to remove pages that haven't been accessed recently. It does this by first clearing the accessed bit in the hardware page table entry (PTE) and then later checking the bit to see whether the page has been accessed. If the bit remains clear, the page wasn't accessed between scans and is aged. Later, the age of pages is used to locate candidate pages to remove from the working set.

alertable wait state
A thread state that the thread enters either by waiting on an object handle and specifying that its wait is alertable (with the Windows WaitForMultipleObjectsEx function) or by testing directly whether it has a pending APC (using SleepEx). In both cases, if a user-mode APC is pending, the kernel interrupts (alerts) the thread, transfers control to the APC routine, and resumes the thread's execution when the APC routine completes. User-mode APCs are delivered to a thread only when it's in an alertable wait state.

allocation granularity
The granularity with which virtual memory is allocated. Windows aligns each region of reserved process address space to begin on an integral boundary defined by the system allocation granularity value, which can be retrieved from the Windows *GetSystemInfo* function. Currently, this value is 64 KB.

This size was chosen so that if support were added for future processors with large page sizes (for example, up to 64 KB), the risk of requiring changes to applications that made assumptions about allocation alignment would be reduced. (Windows kernel-mode code isn't subject to the same restrictions; it can reserve memory on a single-page granularity.)

alternative hive

A hive that acts as a backup to the crucial SYSTEM hive. The alternate hive is stored in \Winnt\System32\Config as System.alt. Whenever a hive sync flushes dirty sectors to the SYSTEM hive, the hive sync also updates the System.alt hive. If the configuration manager detects that the SYSTEM hive is corrupt when the system boots, the configuration manager attempts to load the alternate hive. If that hive is usable, it then uses that alternate to update the original SYSTEM hive.

APC queue

A queue in which asynchronous procedure calls (APCs) waiting to execute reside. The APC queues (one for user mode and one for kernel mode) are thread-specific—each thread has its own APC queues (unlike the DPC queue, which is processor-wide).

asymmetric multiprocessing (ASMP)

A system of processing on a multiprocessor system that typically selects one processor to execute operating system code while other processors run only user code.

asynchronous I/O

An I/O model that allows an application to issue an I/O request and then continue executing while the request is completed. This type of I/O can improve an application's throughput because it allows the application to continue with other work while an I/O operation is in progress.

asynchronous procedure call (APC)

A function that provides a way for user programs and system code to execute code in the context of a particular user thread (and hence a particular process address space). An APC can be either kernel mode or user mode. (Kernel-mode APCs don't require "permission" from a target thread to run in that thread's context, as user-mode APCs do.)

asynchronous read-ahead with history

A method in which the cache manager maintains a history of the last two read requests in the private cache map for the file handle being accessed.

atomic transaction

A technique for handling modifications to a database so that system failures don't affect the correctness or integrity of the database. The basic tenet of atomic transactions is that some database operations, called transactions, are all-or-nothing propositions. The separate disk updates that make up the transaction must be executed atomically; that is, once the transaction begins to execute, all its disk updates must be completed. If a system failure interrupts the transaction, the part that has been completed must be undone, or rolled back. The rollback operation returns the database to a previously known and consistent state, as if the transaction had never occurred. *See also* transaction.

attribute list

A special kind of file attribute in an NTFS file header that contains additional attributes. The attribute list is created if a particular file has too many attributes to fit in the master file table (MFT) record. The attribute list attribute contains the name and type code of each of the file's attributes and the file reference of the MFT record where the attribute is located.

authentication packages

Dynamic-link libraries (DLLs) that run in the context of the LSASS process and that implement Windows authentication policy. An authentication DLL is responsible for checking whether a given username and password match, and if so, returning to LSASS information detailing the user's security identity.

Windows authentication packages include Kerberos and MSV1_0.

automatic working set trimming
A technique the memory manager uses when physical memory runs low to increase the amount of free memory available in the system.

bad-cluster file
A system file (filename $BadClus) that records any bad spots on the disk volume.

balance set manager
A system thread that wakes up once per second to check and possibly initiate various scheduling and memory management–related events.

basic disk
A disk that relies on the MS-DOS-style partitioning scheme. *See also* dynamic disk.

bitmap file
A system file (filename $Bitmap) in which NTFS records the allocation state of the volume. The data attribute for the bitmap file contains a bitmap, each of whose bits represents a cluster on the volume, identifying whether the cluster is free or has been allocated to a file.

boot code
Instructions executed when a system is booted.

boot device drivers
Device drivers necessary to boot the system.

boot file
A system file (filename $Boot) that stores the Windows bootstrap code.

boot partition
The partition that contains core operating system files. The boot partition is identified by the system at startup. The code in a master boot record (MBR) scans the primary partition table until it locates a partition containing a flag that signals the partition is bootable. When the MBR finds at least one such flag, it reads the first sector from the flagged partition into memory and transfers control to code within the partition.

boot sector
The first sector of the partition marked as active and from which the MBR boots. The boot sector contains information identifying the partition's file system format and structure.

boot volume
The volume that contains the Windows operating system and its support files. The boot volume can be, but does not have to be, the same as the system volume.

bus driver
Driver that services a bus controller, adapter, bridge, or any device that has child devices. Bus drivers are required drivers, and Microsoft generally provides them; each type of bus (such as PCI, PCMCIA, and USB) on a system has one bus driver.

cache manager
The component of the Windows executive that provides systemwide caching services for NTFS and other file system drivers, including network file system drivers (servers and redirectors).

careful write
A technique for constructing a file system's I/O and caching support. *See also* write-through.

change journal
An internal file where an NTFS file system can record information that allows applications to efficiently

monitor file and directory changes. A change journal is usually large enough to virtually guarantee that applications get a chance to process changes without missing any.

checked build
A special debug version of Windows that is available only as part of the MSDN Professional (or Universal) subscription. The checked build is created by compiling the Windows sources with the compile-time flag DEBUG defined.

checkpoint record
A record that helps NTFS determine what processing would be needed to recover a volume if a crash were to occur immediately. This record also includes redo and undo information.

class driver
A type of kernel-mode device driver that implements the I/O processing for a particular class of devices, such as disk, tape, or CD-ROM.

clock algorithm
A working set page replacement policy implemented on uniprocessor systems, similar to a least recently used policy (as implemented in most versions of UNIX).

clock interrupt handler
A system routine that updates the system time and then decrements a counter that tracks how long the current thread has run.

cluster factor
The cluster size on a volume, which is established when a user formats the volume with either the format command or the Disk Management Microsoft Management Console (MMC) snap-in.

cluster remapping
A process in which NTFS dynamically retrieves good data from a cluster with a bad sector, allocates a new cluster, and copies the data to the new cluster.

clustering
A method by which the memory manager resolves a page fault by reading into memory several pages near the page explicitly accessed.

clusters
Same-size allocation units into which a volume is divided. Each cluster must be uniquely numbered using 16 bits.

collided page fault
A fault that occurs when another thread or process faults a page that is currently being in-paged.

commitment
The process by which the memory manager keeps track of private committed memory usage on a global basis.

common model
A set of classes in the Common Information Model (CIM) that represent objects that are specific to management areas of a system but independent of a particular implementation. These classes are considered an extension of the CIM core model. *See also* core model.

complete memory dump
A memory dump that contains all of physical memory at the time of the crash. This type of dump requires that a page file be at least the size of physical memory. Because it can require an inordinately large page file on large memory systems, this type of dump file is the least common. Windows NT 4 supported only this type of crash dump file.

completion port

A mechanism to deliver I/O completion notification to threads. Once a file is associated with a completion port, any asynchronous I/O operations that complete on the file result in a completion packet being queued to the completion port. A thread can wait for any outstanding I/Os to complete on multiple files simply by waiting for a completion packet to be queued to the completion port. With completion ports, concurrency, or the number of threads that an application has actively servicing client requests, is controlled with the aid of the system.

configuration manager

A major component of the executive that's responsible for implementing and managing the system registry.

container object

A namespace object that can hold other objects, including other container objects. Examples of containers are directories in the file system namespace and keys in the registry namespace.

context switch

The procedure of saving the volatile machine state associated with a running thread, loading another thread's volatile state, and starting the new thread's execution.

control objects

A set of kernel objects that establishes semantics for controlling various operating system functions. This set includes the kernel process object, the asynchronous procedure call (APC) object, the deferred procedure call (DPC) object, and several objects the I/O system uses, such as the interrupt object.

core model

A set of classes in the Common Information Model (CIM) provided as part of the WBEM standard. These classes are CIM's basic language and represent objects that apply to all areas of management. *See also* common model.

crash dump

A record of system memory at the time of a crash that can help you figure out which component caused the crash.

critical sections

An intra-process mutual exclusion synchronization primitive.

deadlock detection

A Driver Verifier option in Windows XP and Windows Server 2003 that monitors the use of spin locks, fast mutexes, and mutexes, looking for patterns that could result in deadlock.

deferred procedure call (DPC)

A routine that performs most of the work involved in handling a device interrupt after the interrupt service routine (ISR) executes. The DPC routine executes at an interrupt request level (IRQL) that is lower than that of the ISR to avoid blocking other interrupts unnecessarily. A DPC routine initiates I/O completion and starts the next queued I/O operation on a device.

deferred procedure call (DPC) object

A kernel control object that describes a request to defer interrupt processing to DPC/dispatch level. (*See* interrupt request levels [irqls].) This object isn't visible to user-mode programs but is visible to device drivers and other system code. The most important piece of information the DPC object contains is the address of the system function that the kernel will call when it processes the DPC interrupt.

Deferred ready

A state used for threads that have been selected to run on a specific processor but have not yet been scheduled. This new state in Windows Server 2003 exists so that the kernel can minimize the

amount of time the systemwide lock on the scheduling database is held.

demand-paging policies

A fetch policy that loads a page into physical memory only when a page fault occurs. In a demand-paging system, a process incurs many page faults when its threads first begin executing because the threads reference the initial set of pages they need to get going. Once this set of pages is loaded into memory, the paging activity of the process decreases.

desired access rights

The accesses desired by a thread opening an object.

device drivers

Loadable kernel-mode modules (typically ending in .sys) that interface between the I/O system and the relevant hardware. Device drivers on Windows don't manipulate hardware devices directly, but rather they call parts of the hardware application layer (HAL) to interface with the hardware.

device ID

A device identifier reported to the Plug and Play manager. The identifiers are bus-specific; for a USB bus, an identifier consists of a vendor ID (VID) for the hardware vendor that made the device and a product ID (PID) that the vendor assigned to the device.

device instance ID (DIID)

An identifier consisting of a device ID and an instance ID that the Plug and Play manager uses to locate the device's key in the enumeration branch of the registry (HKLM\SYSTEM\CurrentControlSet\Enum).

device object

A data structure that represents a physical, logical, or virtual device on the system and describes its characteristics, such as the alignment it requires for buffers and the location of its device queue to hold incoming I/O request packets.

device-specific module (DSM)

Third-party drivers used to manage details of the path management, such as load balancing policies that choose which path to route requests and error detection mechanisms to inform Windows when a path fails.

device tree

An internal tree the Plug and Play manager creates that represents the relationships between devices. Nodes in the tree are called devnodes. *See also* devnode.

devnode

A node in a device tree. A devnode contains information about the device objects that represent the device as well as other Plug and Play–related information the Plug and Play manager stores. *See also* device tree.

direct memory access (DMA)

A third interface provided by the cache manager to cached data. The DMA functions are used to read from or write to cache pages without intervening buffers, such as when a network file system is doing a transfer over the network.

dirty page threshold

The number of pages that the system cache keeps in memory before waking up the cache manager's lazy writer system thread to write out pages back to the disk. This value is computed at system initialization time and depends on physical memory size and the value of the registry value HKLM\SYSTEM\CurrentControlSet\
Control\Session Manager\Memory Management\LargeSystemCache.

discretionary access control

Allows the owner of a resource to determine who can access the resource and what they can do with it. The owner grants rights that permit various kinds of access to a user or to a group.

disk group

Dynamic disks that share a common database. VERITAS's commercial volume-management software for Windows includes disk groups, but the Windows Logical Disk Manager (LDM) implementation includes only one disk group.

dispatch code

Instructions of assembly language code stored in an interrupt object when it is initialized. When an interrupt occurs, this code is executed.

dispatch routines

The main functions that a device driver provides. Some examples of dispatch routines are open, close, read, and write, and any other capabilities the device, file system, or network supports. When called on to perform an I/O operation, the I/O manager generates an IRP and calls a driver through one of the driver's dispatch routines.

dispatcher

A set of routines in the kernel that implement Windows scheduling. Windows doesn't have a single "scheduler" module or routine—the code is spread throughout the kernel in which scheduling-related events occur.

dispatcher database

A set of data structures the kernel maintains to make thread-scheduling decisions. The dispatcher database keeps track of which threads are waiting to execute and which processors are executing which threads. *See also* dispatcher ready queue.

dispatcher header

A data structure that contains the object type, the signaled state, and a list of the threads waiting on that object.

dispatcher objects

A set of kernel objects that incorporate synchronization capabilities and alter or affect thread scheduling. The dispatcher objects include the kernel thread, mutex (called mutant internally), event, kernel event pair, semaphore, timer, and waitable timer.

dispatcher ready queue

The most important structure in the dispatcher database (located at KiDispatcherReadyListHead). The dispatcher ready queue is really a series of queues, one queue for each scheduling priority. The queues contain threads that are in the ready state, waiting to be scheduled for execution.

display driver

Driver that translates device-independent graphics requests into device-specific requests. The device-specific requests are then paired with a kernel-mode video miniport driver to complete video display support. A display driver is responsible for implementing drawing operations, either by writing directly to the frame buffer or by communicating with the graphics accelerator chip on the controller.

driver object

Data structure that represents an individual driver in the system and records for the I/O manager the address of each of the driver's dispatch routines (entry points).

driver support routines

Routines that device drivers call to accomplish their I/O requests.

dynamic disk

A disk that supports multipartition volumes, providing a more flexible partitioning scheme than that of a

basic disk. *See also* basic disk.

dynamic-link library (DLL)
A set of callable subroutines linked as a binary image that can be dynamically loaded by applications that use them.

environment subsystems
User processes and DLLs that expose the native operating system services to user applications, thus providing an operating system environment, or personality. Windows 2000 ships with two environment subsystems: Windows and POSIX (Windows NT 4.0 had one for OS/2 1.2). Windows XP and later only ship with Windows, but an enhanced POSIX subsystem is included with the free Services for Unix product from Microsoft.

event
An object with a persistent state (signaled or not signaled) that can be used for synchronization; also, a system occurrence that triggers an action.

exception
A synchronous condition that results from the execution of a particular instruction. Running a single program with the same data under the same conditions can reproduce exceptions.

exception dispatcher
A kernel module that services all exceptions, except those simple enough to be resolved by the trap handler. The exception dispatcher's job is to find an exception handler that can "dispose of" the exception.

executive
The upper layer of Ntoskrnl.exe. (The kernel is the lower layer.) The executive contains the base operating system services, such as the process and thread manager, the virtual memory manager, the memory manager, the security reference monitor, the I/O system, and the cache manager. *See also* kernel.

executive objects
Objects implemented by various components of the executive (such as the process manager, memory manager, I/O subsystem, and so on). The executive objects and object services are primitives that the environment subsystems use to construct their own versions of objects and other resources. Because executive objects are typically created either by an environment subsystem on behalf of a user application or by various components of the operating system as part of their normal operation, many of them contain (encapsulate) one or more kernel objects. *See also* kernel objects.

executive resources
Resources that provide both exclusive access (such as a mutex) as well as shared read access (multiple readers sharing read-only access to a structure). Because executive resources are available only to kernel-mode code, they aren't accessible from the Windows API.

executive support routines
Functions in Ntoskrnl.exe that provide services to device drivers.

extended partition
A special partition type that contains a master boot record (MBR) with its own partition table. By using extended partitions, Microsoft's operating systems overcome the apparent limit of four partitions per disk. In general, the recursion that extended partitions permit can continue indefinitely, which means that no upper limit exists to the number of possible partitions on a disk. *See also* partition.

fast I/O
A means of reading or writing a cached file without going through the work of generating an I/O request packet (IRP).

fast LPC
A special interprocess communication facility used to send messages between threads.

file mapping objects
Windows APO underlying primitives in the memory manager that are used to implement shared memory (called section objects internally). *See also* section object.

file reference
A 64-bit value that identifies a file on an NTFS volume. The file reference consists of a file number and a sequence number. The file number corresponds to the position of the file's file record in the master file table minus 1 (or to the position of the base file record minus 1 if the file has more than one file record).

file system driver (FSD)
A type of kernel-mode device driver that accepts I/O requests to files and satisfies the requests by issuing its own, more explicit, requests to physical device drivers. *See also* local file system driver (FSD), network file system driver (FSD).

file system filter driver
A type of kernel-mode device driver that intercepts I/O requests, performs additional processing, and passes them on to lower-level drivers.

file system format
Defines the way that file data is stored on storage media and impacts a file system's features. For example, a format that doesn't allow user permissions to be associated with files and directories can't support security. A file system format can also impose limits on the sizes of files and storage devices that the file system supports.

filter device object (FiDO)

A device object that can be part of a devnode. One or more optional FiDOs can layer either between the physical device object (PDO) and the functional device object (FDO) or above the FDO. *See also* devnode, functional device object (FDO), physical device object (PDO).

filter driver
See file system filter driver.

frame-based exception handlers
The mechanism that permits each stack frame in a call stack to have its own exception handler declared.

foreground application
The process that owns the thread that owns the window that's in focus.

free build
The version of the Windows system that can be purchased as a retail product. It is built with full compiler optimizations turned on and has internal symbol table information stripped out from the images. *See also* checked build.

function driver
The main device driver that provides the operational interface for its device. It is a required driver unless the device is used raw (an implementation in which I/O is done by the bus driver and any bus filter drivers, such as SCSI PassThru). A function driver is the driver that knows the most about a particular device and is usually the only driver that accesses device-specific registers.

functional device object (FDO)
A device object that is a required part of a devnode. The function driver that the Plug and Play manager

loads to manage a detected device creates the FDO. An FDO represents the logical interface to a device. *See also* devnode, filter device object (FiDO), physical device object (PDO).

granted access rights
The accesses granted to a thread by the security reference monitor as the result of a successful object open.

Graphical Identification and Authentication (GINA)
A user-mode DLL that runs in the Winlogon process and that Winlogon uses to obtain a user's name and password or smart card PIN. The standard GINA is \Winnt\System32\Msgina.dll.

handle
An object identifier. A process receives a handle to an object when it creates or opens an object by name. Referring to an object by its handle is faster than using its name because the object manager can skip the name lookup and find the object directly.

handle table
A table that contains pointers to all the objects that the process has opened a handle to. Handle tables are implemented as a three-level scheme, similar to the way that the x86 memory management unit implements virtual-to-physical address translation.

hardware abstraction layer (HAL)
A loadable kernel-mode module (Hal.dll) that provides the low-level interface to the hardware platform on which Windows is running. The HAL hides hardware-dependent details such as I/O interfaces, interrupt controllers, and multiprocessor communication mechanisms—any functions that are architecture-specific and machine-dependent.

hardware device drivers
Device drivers that manipulate hardware to write output to or retrieve input from a physical device or network. There are many types of hardware device drivers, such as bus drivers, human interface drivers, mass storage drivers, and so on.

hash
A statistically unique value that is generated from a block of data (for example, a file) using cryptographic algorithms. Because different data results in different hashes, hashes can be used to detect changes to data from corruption or tampering. The Windows driver signing facility uses hashes.

heap
A region of one or more pages that can be subdivided and allocated in smaller chunks by a set of functions provided by the heap manager.

heap manager
A set of functions that allocate and deallocate variable amounts of memory (not on a page-size granularity). The heap manager functions exist in two places: Ntdll.dll and Ntoskrnl.exe. The subsystem APIs (such as the Windows heap APIs) use the copy in Ntdll, and various executive components and device drivers use the copy in Ntoskrnl.

hive
One of a number of files stored on disk that contain registry information. Each hive contains a registry tree, which has a key that serves as the root or starting point of the tree.

hyperspace
A special region used to map the process working set list and to temporarily map other physical pages for such operations as zeroing a page on the free list (when the zero list is empty and a zero page is needed), invalidating page table entries in other page tables (such as when a page is removed from the standby list), and on process creation setting up a new process's address space.

I/O completion routine
A routine implemented by a layered driver that will notify the driver when a lower-level driver finishes processing an I/O request packet (IRP). For example, the I/O manager calls a file system driver's I/O completion routine after a device driver finishes transferring data to or from a file. The completion routine notifies the file system driver about the operation's success, failure, or cancellation, and it allows the file system driver to perform cleanup operations.

I/O request packet (IRP)
A data structure that controls how the I/O operation is processed at each stage. Most I/O requests are represented by an IRP, which travels from one I/O system component to another.

I/O subsystem API
The internal executive system services (such as NtReadFile and NtWriteFile) that subsystem DLLs call to implement a subsystem's documented I/O functions.

I/O system
The Windows executive component that accepts I/O requests (from both user-mode and kernel-mode callers) and delivers them, in a different form, to I/O devices.

ideal processor
The preferred processor that a particular thread should run on.

idle summary
A bitmask (KiIdleSummary) in which each set bit represents an idle processor.

impersonation
A capability that allows threads to have a different access token than that of the process.

initialization routine
A driver routine that the I/O manager executes when it loads the driver into the operating system. The initialization routine creates system objects that the I/O manager uses to recognize and access the driver.

in-paging I/O
A condition that occurs when a read operation must be issued to a file (paging or mapped) to satisfy a page fault. The in-page I/O operation is synchronous—the thread waits on an event until the I/O completes—and isn't interruptible by asynchronous procedure call (APC) delivery.

instancing
The term for making separate copies of the same parts of a namespace. Instancing \DosDevices makes it possible for each user to have different drive letters and Windows objects such as serial ports.

intelligent file read-ahead
A technique that predicts what data the calling thread is likely to read next based on the data it's currently reading.

inter-processor interrupt (IPI)
An interrupt the kernel issues to request that another processor perform an action, such as dispatching a particular thread for execution or updating its translation look-aside buffer cache.

interrupt
An asynchronous event (one that can occur at any time) that is unrelated to what the processor is executing. Interrupts are generated primarily by I/O devices, processor clocks, or timers, and they can be enabled or disabled.

interrupt dispatch table (IDT)
A data structure that Windows uses to locate the routine that will handle a particular interrupt. The interrupt request level (IRQL) of the interrupting source serves as a table index, and table entries

point to the interrupt-handling routines.

interrupt dispatcher
A submodule of the kernel's trap handler that responds to interrupts.

interrupt object
A kernel control object that allows device drivers to register interrupt service routines (ISRs) for their devices. An interrupt object contains all the information the kernel needs to associate a device ISR with a particular level of interrupt, including the address of the ISR, the interrupt request level (IRQL) at which the device interrupts, and the entry in the kernel's interrupt dispatch table with which the ISR should be associated.

interrupt request (IRQ)
A value identifying an interrupt. On x86 systems, external I/O interrupts come into one of the lines on an interrupt controller. The controller in turn interrupts the processor on a single line. Once the processor is interrupted, it queries the controller to get the interrupt request (IRQ). The interrupt controller translates the IRQ to an interrupt number, uses this number as an index into the interrupt dispatch table (IDT), and transfers control to the appropriate interrupt dispatch routine. At system boot time, Windows fills in the IDT with pointers to the kernel routines that handle each interrupt and exception.

interrupt request levels (IRQLs)
An interrupt priority scheme imposed by Windows. The kernel represents IRQLs internally as a number from 0 through 31 (0 to 15 on 64-bit Windows), with higher numbers representing higher-priority interrupts. Although the kernel defines the standard set of IRQLs for software interrupts, the HAL maps hardware-interrupt numbers to the IRQLs.

interrupt service routine (ISR)
A device driver routine that the kernel's interrupt dispatcher transfers control to when a device issues an interrupt. In the Windows I/O model, ISRs run at a high device interrupt request level (IRQL), so they perform as little work as possible to avoid blocking lower-level interrupts unnecessarily. An ISR queues a deferred procedure call (DPC), which runs at a lower IRQL, to execute the remainder of interrupt processing. Only drivers for interrupt-driven devices have ISRs; a file system, for example, doesn't have one.

job object
A nameable, securable, shareable object in Windows that controls certain attributes of processes associated with the job. A job object's basic function is to allow groups of processes to be managed and manipulated as a unit. The job object also records basic accounting information for all processes associated with the job and for all processes that were associated with the job but have since terminated.

journaling
A logging technique originally developed for transaction processing that a recoverable file system such as NTFS uses to ensure volume consistency.

kernel
The lowest layer in Ntoskrnl.exe. The kernel, a component of the executive, determines how the operating system uses the processor or processors and ensures that they are used prudently. The kernel provides thread scheduling and dispatching, trap handling and exception dispatching, interrupt handling and dispatching, and multiprocessor synchronization. *See also* executive.

kernel debugger
A tool used to debug drivers, troubleshoot hung systems, and examine crash dumps. It is also a useful tool for investigating Windows internals because it can display internal Windows system information not visible through any standard utility. (The LiveKd tool on the companion CD allows the use of the

standard kernel debuggers on a live system.)

kernel handle table
A table (referenced internally with the name ObpKernelHandleTable) whose handles are accessible only from kernel mode and in any process context. A kernel-mode function can reference the handles in this table in any process context with no performance impact.

kernel memory dump
A type of memory dump (the default on Windows Server systems) that contains only the kernel-mode read/write pages present in physical memory at the time of the crash. A kernel memory dump doesn't contain pages belonging to user processes. Because only kernel-mode code can directly cause Windows to crash, however, it's unlikely that user process pages are necessary to debug a crash. There is no way to predict the size of a kernel memory dump because its size depends on the amount of kernel-mode memory allocated by the operating system and drivers present on the machine.

kernel mode
A privileged mode of code execution in a processor in which all memory is totally accessible and all CPU instructions can be issued. Operating system code (such as system services and device drivers) runs in kernel mode. *See also* user mode.

kernel objects
A primitive set of objects implemented by the Windows kernel. These objects aren't visible to user-mode code but are created and used only within the executive. Kernel objects provide fundamental capabilities, such as synchronization, on which executive objects are built. *See also* executive objects.

kernel streaming filter drivers
Kernel-mode drivers chained together to perform signal processing on data streams, such as recording or displaying audio and video.

kernel-mode device driver
The only type of driver that can directly control and access hardware devices.

kernel-mode graphics driver
A Windows subsystem display or print device driver that translates device-independent graphics (GDI) requests into device-specific requests.

key
A mechanism to refer to data in the registry. Although keys appear in the object manager namespace, the registry manages them in a way similar to how it manages file objects. Zero or more key values are associated with a key object; key values contain data about the key.

key control block
A structure that stores the full pathname of a registry key, includes the cell index of the key node that the control block refers to, and contains a flag that notes whether the configuration manager needs to delete the key cell that the key control block refers to when the last handle for the key closes. In Windows, all key control blocks are in an alphabetized binary tree to enable quick searches for existing key control blocks by name. A key object points to its corresponding key control block, so if two applications open the same registry key, each will receive a key object and both key objects will point to a common key control block.

key object
An object type the configuration manager defines to integrate the registry's namespace with the kernel's general namespace.

keyed event

A synchronization object new to Windows XP.

last known good control set

A copy of the critical boot-time information under the registry key HKLM\SYSTEM\CurrentControlSet, made when a user successfully logs on. The last known good control set can be selected at boot time in case configuration changes made to the registry result in the system not being able to boot successfully.

lazy IRQL

A performance optimization that avoids Programmable Interrupt Controller (PIC) accesses. When the interrupt request level (IRQL) is raised, the HAL notes the new IRQL internally instead of changing the interrupt mask. If a lower-priority interrupt subsequently occurs, the HAL sets the interrupt mask to the settings appropriate for the first interrupt and postpones the lower-priority interrupt until the IRQL is lowered. Thus, if no lower-priority interrupts occur while the IRQL is raised, the HAL doesn't need to modify the PIC.

lazy writer

A set of system threads that call the memory manager to flush cache contents to disk as a background activity (asynchronous disk writing). The cache manager optimizes disk I/O by using its lazy writer.

legacy drivers

Device drivers written for Microsoft Windows NT but that run unchanged on Windows 2000 and later. They are differentiated from other Windows drivers in that they don't support power management or work with the Windows Plug and Play manager.

local file system driver (FSD)

A driver that manages volumes directly connected to the computer. *See also* file system driver.

local kernel debugging

The ability to connect the kernel debugger to the local running system to view internal state (as opposed to connecting to a target system booted in debugging mode using a null modem or 1394 cable).

local procedure call (LPC)

An interprocess communication facility for high-speed message passing (not available through the Windows API but rather through an internal mechanism available only to Windows operating system components). LPCs are typically used between a server process and one or more client processes of that server. An LPC connection can be established between two user-mode processes or between a kernel-mode component and a user-mode process.

local security authority (LSA) server

A user-mode process running the image \Winnt\ System32\LSASS.exe that is responsible for the local system security policy (such as which users are allowed to log on to the machine, password policies, privileges granted to users and groups, and the system security auditing settings), user authentication, and sending security audit messages to the Event Log. The LSA service (Lsasrv - \Winnt\System32\ Lsasrv.dll), a library that LSASS loads, implements most of this functionality.

local security authority (LSA) server policy database

A database (stored in the registry under HKLM\SECURITY) that contains the system security policy settings. This database includes such information as what domains are trusted to authenticate logon attempts, who has permission to access the system and how (interactive, network, and service logons), who is assigned which privileges, and what kind of security auditing is to be performed.

Local Security Authority Subsystem (LSASS)

The system user-mode process responsible for authentication of accounts accessing a computer.

Local System account
A predefined local account that is used to start a service and provide the security context for that service. The name of the account is NT AUTHORITY\System. This account does not have a password, and any password information that you supply is ignored. The Local System account has full access to the system, including the directory service on domain controllers. Because the Local System account acts as a computer on the network, it has access to network resources.

log file
A metadata file (filename $LogFile) NTFS uses to record all operations that affect the NTFS volume structure, including file creation or any commands, such as Copy, that alter the directory structure. The log file is used to recover an NTFS volume after a system failure.

log hive
A registry hive the configuration manager uses to make sure that a nonvolatile registry hive (one with an on-disk file) is always in a recoverable state. Each nonvolatile hive has an associated log hive, which is a hidden file with the same base name as the hive and a .log extension. *See also* hive.

logging
A transaction-processing technique NTFS uses to maintain file system integrity in case of system crashes or other failures. In NTFS logging, the suboperations of any transaction that alters important file system data structures are recorded in a log file before they are carried through on the disk so that if the system crashes, partially completed transactions can be redone or undone when the system comes back on line.

logical cluster numbers (LCNs)
The numbers of all clusters from the beginning of the volume to the end with which NTFS refers to physical locations on a disk. To convert an LCN to a physical disk address, NTFS multiplies the LCN by the cluster factor to get the physical byte offset on the volume, as the disk driver interface requires.

logical prefetcher
The kernel component that monitors boot and application startup file access and that preemptively reads that data to speed up subsequent booting and application startups.

logical sequence numbers (LSNs)
The numbers that NTFS uses to identify records written to the log file.

logon process
A user-mode process running Winlogon.exe that is responsible for capturing the username and password, sending them to the local security authority server for verification, and creating the initial process in the user's session.

look-aside list
A fast memory allocation mechanism that contains only fixed-sized blocks. Look-aside lists can be either pageable or nonpageable, so they are allocated from paged or nonpaged pool.

LPC facility
Local procedure call interprocess communication support in the kernel.

mapped file I/O
The ability to view a file residing on disk as part of a process's virtual memory. A program can access the file as a large array without buffering data or performing disk I/O. The program accesses memory, and the memory manager uses its paging mechanism to load the correct page from the disk file. If the application writes to its virtual address space, the memory manager writes the changes back to the file as part of normal paging.

mask
The process whereby interrupts wait for an executing thread to lower the IRQL before the interrupt is

processed. Interrupts from a source with an IRQL above the current level interrupt the processor, whereas interrupts from sources with IRQLs equal to or below the current level are masked until an executing thread lowers the IRQL.

master file table (MFT)
The heart of the NTFS volume structure. The MFT is implemented as an array of file records. The size of each file record is fixed at 1 KB, regardless of cluster size.

memory manager
The Windows executive component that implements demand-paged virtual memory, giving each process the illusion that it has a large virtual address space (while mapping a subset of that address space to physical memory).

metadata
Data that describes the files on a disk; also called volume structure data.

metadata files
A set of files in each NTFS volume that contains the information used to implement the file system structure.

MFT mirror
An NTFS metadata file (filename $MFTMirr) located in the middle of the disk called that contains a copy of the first few rows of the master file table.

miniport driver
A type of kernel-mode device driver that maps a generic I/O request to a type of port into an adapter type, such as a specific SCSI adapter.

mirrored volume
A volume on which the contents of a partition on one disk are duplicated in an equal-sized partition on another disk. Mirrored volumes are sometimes referred to as RAID level 1 (RAID-1).

mirror set
A technique by which the contents of a partition on one disk are duplicated in an equal-size partition on another disk.

modified page writer
A thread in the virtual memory manager that is responsible for limiting the size of the modified page list by writing pages to their backing store locations when the list becomes too big. The modified page writer consists of two system threads: one to write out modified pages (MiModifiedPageWriter) to the paging file and a second one to write modified pages to mapped files (MiMappedPageWriter).

mount
A technique NTFS uses when it first accesses a volume; in this context, to mount means to prepare the volume for use. To mount the volume, NTFS looks in the boot file to find the physical disk address of the master file table.

mount points
A mechanism that permits the linking of volumes through directories on NTFS volumes, which makes volumes accessible with no drive-letter assignment. Reparse points in NTFS make mount points possible.

MSDN
Microsoft Developer Network, Microsoft's support program for developers. MSDN offers three CD-ROM subscription programs: MSDN Library, Professional, and Universal. For more information, see msdn.microsoft.com.

multipartition volumes
Objects that represent sectors from multiple partitions and that file system drivers manage as a single unit. Multipartition volumes offer performance, reliability, and sizing features that simple volumes don't.

multipathing
Where more than one set of hardware exists between the computer and a disk so that if a path fails, the system can still access the disk via an alternate path.

mutant
Internal name for a mutex.

mutex
A synchronization mechanism used to serialize access to a resource.

mutual exclusion
A means whereby only one thread or processor is allowed access to a resource at any given time.

name retention
The first phase of object retention, which the object manager implements. Name retention is controlled by the number of open handles to an object that exist. Every time a process opens a handle to an object, the object manager increments the open handle counter in the object's header. As processes finish using the object and close their handles to it, the object manager decrements the open handle counter. When the counter drops to 0, the object manager deletes the object's name from its global namespace. This deletion prevents new processes from opening a handle to the object.

native application
An application that uses only system service APIs provided by Ntdll and that isn't a client of the Windows subsystem. Smss (Session Manager) is an example of a native application.

network file system driver (FSD)
A driver that allows users to access data volumes connected to remote computers. *See also* file system driver.

network logon service
A user-mode service inside the Services.exe process that responds to network logon requests. Authentication is handled as local logons are, by sending them to the LSASS process for verification.

network redirectors and servers
File system drivers that transmit remote I/O requests to a machine on the network and receive such requests, respectively.

nonpaged pool
Memory pool that consists of ranges of system virtual addresses that are guaranteed to be resident in physical memory at all times and thus can be accessed from any address space without incurring paging I/O. Nonpaged pool is created at system initialization and is used by kernel-mode components to allocate system memory.

Ntdll.dll
A special system-support library primarily for the use of subsystem DLLs that contains system service dispatch stubs to Windows executive system services and internal support functions used by subsystems, subsystem DLLs, and other native images.

Ntkrnlmp.exe
The executive and kernel for multiprocessor systems.

Ntoskrnl.exe
The executive and kernel for uniprocessor systems.

object
In the Windows executive, a single, run-time instance of a statically defined object type.

object attribute
A field of data in an object that partially defines the object's state.

object directory
A container object for other objects. The object directory is used to implement the hierarchical namespace within which other object types are stored.

object handle
An index into a process-specific handle table, pointed to by the executive process (EPROCESS) block.

object manager
The Windows executive component responsible for creating, deleting, protecting, and tracking objects. The object manager centralizes resource control operations that would otherwise be scattered throughout the operating system.

object methods
The means for manipulating objects, usually to read or change the object attributes. For example, the open method for a process would accept a process identifier as input and return a pointer to the object as output.

object reuse protection
A means of preventing users from seeing data that another user has deleted or from accessing memory that another user previously used and then released. Object reuse protection prevents potential security holes by initializing all objects, including files and memory, before they are allocated to a user.

object type
A system-defined data type, including services that operate on instances of the data type and a set of object attributes; sometimes called an *object class*.

page directory
A page the memory manager creates to map the location of all page tables for that process. Each process has a single page directory.

page directory entries (PDEs)
The page directory is composed of PDEs, each of which is 4 bytes long and describes the state and location of all the possible page tables for that process.

page fault
A reference to an invalid page. The kernel trap handler dispatches this kind of fault to the memory manager fault handler (MmAccessFault) to resolve.

page file backed section
A section object that is mapped to committed memory.

page file quota
A limit on the number of committed pages a process can consume—not necessarily page file space.

page frame database
A database that describes the state of each page in physical memory. Pages are in one of eight states: active (also called valid), transition, standby, modified, modified-no-write, free, zeroed, or bad.

page frame number (PFN) database
Describes the state of each page in physical memory.

page table
A page of mapping information (made up of an array of page table entries) the operating system

constructs that describes the location of the virtual pages in a process address space. Because Windows provides a private address space for each process, each process has its own set of process page tables to map that private address space because the mappings will be different for each process. The page tables that describe system space are shared among all processes.

page table entry (PTE)
An entry in a process's page table that contains the address to which the virtual address is mapped. The page can be in physical memory or it can be on disk.

paged pool
A region of virtual memory in system space that can be paged in and out of the system process's working set. Paged pool is created at system initialization and is used by kernel-mode components to allocate system memory. Uniprocessor systems have two paged pools; multiprocessor systems have four. Having more than one paged pool reduces the frequency of system code blocking on simultaneous calls to pool routines.

paging
The process of moving memory contents to disk, freeing physical memory so that it can be used for other processes or for the operating system itself. Because most systems have much less physical memory than the total virtual memory in use by the running processes (2 GB or 3 GB for each process), the memory manager transfers, or pages, some of the memory contents to disk.

partition
A discrete area of the hard disk in Microsoft operating systems. File systems (such as FAT and NTFS) format each partition into a volume. A hard disk can contain up to four primary partitions. *See also* extended partition.

partition table
A part of the master boot record (MBR) that consists of four entries that define the locations of as many as four primary partitions on a disk. The partition table also records a partition's type. Numerous predefined partition types exist, and a partition's type specifies which file system the partition includes.

Physical Address Extension (PAE)
A memory-mapping mode included in all Intel x86 processors since the Pentium Pro. With the proper chipset, the PAE mode allows access to up to 64 GB of physical memory. When the x86 executes in PAE mode, the memory management unit (MMU) divides virtual addresses into four fields.

physical device object (PDO)
A device object that's a required part of a devnode. The PDO represents the physical interface to a device. *See also* devnode.

Plug and Play (PnP) manager
A major component of the executive that determines which drivers are required to support a particular device and loads those drivers. The PnP manager retrieves the hardware resource requirements for each device during enumeration. Based on the resource requirements of each device, the PnP manager assigns the appropriate hardware resources such as I/O ports, IRQs, DMA channels, and memory locations. It is also responsible for sending proper event notification for device changes (addition or removal of a device) on the system.

port driver
A type of kernel-mode device driver that implements the processing of an I/O request specific to a type of I/O port, such as SCSI.

port object
A single executive object a local procedure call (LPC) exports to maintain the state needed for

communication.

power manager
A major component of the executive that coordinates power events and generates power management I/O notifications to device drivers. When the system is idle, the power manager can be configured to reduce power consumption by putting the CPU to sleep. Changes in power consumption by individual devices are handled by device drivers but are coordinated by the power manager.

printer driver
A driver that translates device-independent graphics requests to printer-specific commands. These commands are then typically forwarded to a kernel-mode port driver such as the parallel port driver (Parport.sys) or the USB printer port driver (Usbprint.sys).

private cache map
A structure that contains the location of the last two reads so that the cache manager can perform intelligent read-ahead.

process
The virtual address space and control information necessary for the execution of a set of thread objects.

process and thread manager
The services in the executive that export the process and thread object. These services use primitive functions in the kernel to create processes and threads.

process ID
A unique identifier for a process (internally called a client ID).

process working set
The subset of a process's virtual address space that is resident and owned by the running process. *See also* system working set.

processor affinity
The set of processors a thread is permitted to run on.

processor control register (PCR)
A data structure that along with its extension, the processor control block (PRCB), contains information about the state of each processor in the system, such as the current interrupt request level (IRQL), a pointer to the hardware interrupt dispatch table (IDT), the currently running thread, and the next thread selected to run. The kernel and the HAL use this information to perform architecture-specific and machine-specific actions. Portions of the PCR and PRCB structures are defined publicly in the Windows Device Driver Kit (DDK) header file Ntddk.h.

program
A static sequence of instructions.

protected-mode
A state during the boot process in which no virtual-to-physical translation occurs but a full 32 bits of memory becomes accessible. After the system is in protected-mode, Ntldr can access all of physical memory.

protocol driver
A driver that implements a networking protocol such as TCP/IP, NetBEUI, or IPX/SPX.

prototype page table entries (prototype PTEs)
A software structure the memory manager relies on to map potentially shared pages when a page can be shared between two processes. An array of prototype PTEs is created when a section object is first created.

quality of service (Qos)

A networking technology that ensures that critical network applications receive highest priority.

quantum
The length of time a thread is allowed to run before Windows interrupts the thread to find out whether another thread at the same priority level is waiting to run or whether the thread's priority needs to be reduced.

quantum unit
A value that represents how long a thread can run until its quantum expires. This value is an integer value, not a length of time.

queue
A method for threads to enqueue and dequeue notifications of the completion of I/O operations (called an I/O completion port in the Windows API).

queued spinlock
A special type of spinlock that is used only by the kernel and not exported for executive components or device drivers. A queued spinlock scales better on multiprocessor systems than a standard spinlock does. *See also* spinlock.

quota charges
In the Windows object manager, the record of how much the object manager subtracts from a process's allotted paged and/or nonpaged pool quota when a thread in the process opens a handle to the object.

RAID-5 volume
A fault tolerant variant of a regular stripe volume. Fault tolerance is achieved by reserving the equivalent of one disk for storing parity for each stripe. Also called stripe volume with parity.

ready summary
A bit mask (KiReadySummary) that Windows maintains to speed up the selection of which thread to run or preempt.

real-mode
An operating mode in which no virtual-to-physical translation of memory addresses occurs, which means that programs that use the memory addresses interpret them as physical addresses and that only the first 1 MB of the computer's physical memory is accessible. Simple MS-DOS programs execute in a real-mode environment.

recoverability
An advanced feature of NTFS that allows a system to recover from an unexpected system halt. If a system is halted unexpectedly, the metadata of a FAT volume can be left in an inconsistent state, leading to the corruption of large amounts of file and directory data. NTFS implements recoverability by logging changes to metadata in a transactional manner so that file system structures can be repaired to a consistent state with no loss of file or directory structure information. (File data can be lost, however.)

redo information
Information included in the NTFS checkpoint record that explains how to reapply one suboperation of a fully logged (committed) transaction to the volume if a system failure occurs before the transaction is flushed from the cache.

reference count
The object manager's record of how many object pointers it has dispensed to operating system processes. The object manager increments a reference count for an object each time it gives out a pointer to the object; when kernel-mode components finish using the pointer, they call the object manager to decrement the object's reference count.

reparse data
User-defined data about the file or directory, such as its state or location, that can be read from the reparse point by the application that created the data, a file system filter driver, or the I/O manager.

reparse point
An NTFS file or directory that has a block of data called reparse data associated with it.

Reparse tag
The reparse tag allows the component responsible for interpreting the reparse point's reparse data to recognize the reparse point without having to check the reparse data.

resident attribute
An attribute that is stored directly in the master file table. (If a file is small, all its attributes and their values [its data, for example] fit in the file record.)

resource arbitration
A process by which the Plug and Play (PnP) manager optimally assigns hardware resources so that each device meets the requirements necessary for its operation. Because hardware devices can be added to the system after boot-time resource assignment, the PnP manager must also be able to reassign resources to accommodate the needs of dynamically added devices.

restricted token
A token created from a primary or impersonation token using the CreateRestrictedToken function. The restricted token is a copy of the token it's derived from, with some possible modifications: Privileges can be removed from the token's privilege array. SIDs in the token can be marked as deny-only. SIDS in the token can be marked as restricted.

ring
A privilege level defined in the x86 and x64 processor architectures to protect system code and data from being overwritten either inadvertently or maliciously by code of lesser privilege. Windows on the x86 and x64 platforms use privilege level 0 (or ring 0) for kernel mode and privilege level 3 (or ring 3) for user mode.

safe mode
A boot configuration that consists of the minimal set of device drivers and services. By relying on only the drivers and services that are necessary for booting, Windows avoids loading third-party and other nonessential drivers that might crash.

SAM database
A database (stored in the registry under HKLM\SAM) that contains the defined users and groups, along with their passwords and other attributes.

scatter/gather I/O
A kind of high-performance I/O Windows supports, available via the Windows ReadFileScatter and WriteFileGather functions. These functions allow an application to issue a single read or write from more than one buffer in virtual memory to a contiguous area of a file on disk. To use scatter/gather I/O, the file must be opened for noncached I/O, the user buffers being used have to be page-aligned, and the I/Os must be asynchronous (overlapped).

section object
An object that represents a block of memory that two or more processes can share. A section object can be mapped to the paging file or to another file on disk. The executive uses section objects to load executable images into memory, and the cache manager uses them to access data in a cached file. In the Windows subsystem, a section object is called a file-mapping object.

section object pointers
Structure for each open file (represented by a file object) that is the key to maintaining data consistency for

all types of file access as well as to providing caching for files. The section object pointers structure points to one or two control areas. One control area is used to map the file when accessed as a data file, and one is used to map the file when it is run as an executable image.

sector

A hardware-addressable portion of a physical disk. A hard disk sector on an IBM-compatible PC is typically 512 bytes. Utilities that prepare hard disks for the definition of logical drives, including the MS-DOS Fdisk utility or the Windows Setup program, write a sector of data called a master boot record (MBR) to the first sector on a hard disk. The MBR includes a fixed amount of space that contains executable instructions (called boot code) and a partition table with four entries that define the locations of the primary partitions on the disk. *See also* boot code, partition table.

secure attention sequence (SAS)

A keystroke combination that when entered notifies Winlogon of a user logon request.

Secure logon facility

Requires that users can be uniquely identified and that they must be granted access to the computer only after they have been authenticated in some way.

Security Accounts Manager (SAM) service

A set of subroutines responsible for managing the database that contains the usernames and groups defined on the local machine or for a domain (if the system is a domain controller). The SAM runs in the context of the LSASS process.

security auditing

A way in which Windows detects and records important security-related events or any attempts to create, access, or delete system resources. Logon identifiers record the identities of all users, making it easier to trace anyone who performs an unauthorized action.

security descriptor

The data structure that specifies who can perform what actions on an object. A security descriptor consists of attributes.

security identifier (SID)

A means of uniquely identifying entities that perform actions in a system. A SID is a variable-length numeric value that consists of a SID structure revision number, a 48-bit identifier authority value, and a variable number of 32-bit subauthority or relative identifier (RID) values.

security quality of service (SQOS)

The indicator specified by a client opening a server resource that specifies the maximum level of client impersonation the server is allowed.

security reference monitor (SRM)

A component in the Windows executive (Ntoskrnl.exe) that enforces security policies on the local computer. It guards operating system resources, performing run-time object protection and auditing.

semaphore

A counter that provides a resource gate by allowing some maximum number of threads to access the resources protected by the semaphore.

server processes

User processes that are Windows services, such as the Event Log and Schedule services. Many add-on server applications, such as Microsoft SQL Server and Microsoft Exchange Server, also include components that run as Windows services.

session

Consists of the processes and other system objects (such as the window station, desktops, and windows)

that represent a single user's workstation logon session. Each session has a session-specific paged pool area used by the kernel-mode portion of the Windows subsystem (Win32k.sys) to allocate session-private GUI data structures. In addition, each session has its own copy of the Windows subsystem process (Csrss.exe) and logon process (Winlogon.exe).

session space
A component of system space used to map information specific to a user session. The session working set list describes the parts of session space that are resident and in use.

shared cache map
A structure that describes the state of a cached file, including its size and (for security reasons) its valid data length.

shared memory
Memory visible to more than one process or that is present in more than one virtual address space.

signal state
The state of a synchronization object.

simple volume
A set of objects that represent sectors from a single partition that file system drivers manage as a single unit.

single sign-on
The ability for logon information to be simultaneously authenticated on more than one system. For example, a user logging on to a Windows system might simultaneously be authenticated on a UNIX server. That user would then be able to access resources of the UNIX server from the machine running Windows without requiring additional authentication.

small memory dump
A small memory dump (the default on Windows Professional, 64 KB in size) containing the stop code and parameters, the list of loaded device drivers, the data structures that describe the current process and thread (called the EPROCESS and ETHREAD), and the kernel stack for the thread that caused the crash.

spanned volume
A single logical volume composed of a maximum of 32 free partitions on one or more disks. The Windows Disk Management Microsoft Management Console (MMC) snap-in combines the partitions into a spanned volume, which can then be formatted for any of the file systems that Windows supports.

sparse files
Files, often large, that contain only a small amount of nonzero data relative to their size.

spinlock
The locking mechanism the kernel uses to achieve multiprocessor mutual exclusion. The spinlock gets its name from the fact that the kernel (and thus, the processor) is held in limbo, "spinning," until it gets the lock. Spinlocks, like the data structures they protect, reside in global memory. *See also* queued spinlock.

stack frame
Information, representing the activation of a procedure, that is pushed onto the stack when a procedure is invoked. A stack frame can have one or more exception handlers associated with it, each of which protects a particular block of code in the source program.

stream
A sequence of bytes within a file.

striped volume
Series of up to 32 partitions, one partition per disk, that combines into a single logical volume. Striped volumes are also known as RAID level 0 (RAID-0) volumes. A partition in a striped volume need not span an entire disk; the only restriction is that the partitions on each disk be the same size.

stop code
An error code that classifies the type of error detected by a component that crashes the system.

structured exception handling
A type of exception handling that allows applications to gain control when exceptions occur. The application can then fix the condition and return to the place the exception occurred, unwind the stack (thus terminating execution of the subroutine that raised the exception), or declare back to the system that the exception isn't recognized, and continue searching for an exception handler that might process the exception.

subsection
A structure that describes the mapping information for each section of the file (read-only, read-write, copy-on-write, and so on).

subsystem dynamic-link libraries (DLLs)

DLLs that translate a documented function into the appropriate undocumented Windows system service calls. This translation might or might not involve sending a message to the environment subsystem process that is serving the user application.

symbolic link
A mechanism for referring to an object name indirectly.

symmetric encryption algorithm
An algorithm that uses the same key to encrypt and decrypt data. Symmetric encryption algorithms are typically very fast, which makes them suitable for encrypting large amounts of data, such as file data. They do have a weakness, however; their security can be bypassed if their key is obtained.

symmetric multiprocessing (SMP)
A multiprocessing operating system in which there is no master processor—the operating system as well as user threads can be scheduled to run on any processor. All the processors share just one memory space.

synchronization
A thread's ability to synchronize its execution by waiting for an object to change from one state to another. A thread can synchronize with executive process, thread, file, event, semaphore, mutex, and timer objects. Section, port, access token, object directory, symbolic-link, profile, and key objects don't support synchronization.

synchronous I/O
A model for I/O in which a device performs a data transfer and returns a status code when the I/O is complete. The program can then access the transferred data immediately. When used in their simplest form, the Windows ReadFile and WriteFile functions are executed synchronously. They complete an I/O operation before returning control to the caller.

system access-control list (SACL)
Specifies which operations by which users should be logged in the security audit log.

system code write protection
The setting of the read-only parts of the kernel as read-only pages.

system audit ACE

An access-control entry (ACE) contained by a system access-control list (SACL) that, along with the system audit-object ACE, specifies which operations performed on the object by specific users or groups should be audited. System audit-object ACEs specify a GUID indicating the types of objects or subobjects that the ACE applies to and an optional GUID that controls propagation of the ACE to particular child object types. If an SACL is null, no auditing takes place on the object.

system cache

Pages used to map files open in the system cache.

system page table entries (PTEs)

Pool of system PTEs used to map system pages such as I/O space, kernel stacks, and memory descriptor lists.

system service dispatch table

Table in which each entry contains a pointer to a system service rather than to an interrupt handling routine.

system services (or executive system services)

Native functions in the Windows operating system that are callable from user mode. For example, NtCreateProcess is the internal system service the Windows CreateProcess function calls to create a new process.

system support processes

User processes, such as the logon process and the Session Manager, that are not Windows services (that is, not started by the service controller).

system thread

A kind of thread that runs only in kernel mode. System threads always reside in the system process (always process ID 2). These threads have all the attributes and contexts of regular user-mode threads (such as a hardware context, priority, and so on) but run only in kernel-mode executing code loaded in system-space code, whether that be in Ntoskrnl.exe or in any other loaded device driver. System threads don't have a user process address space and hence must allocate any dynamic storage from operating system memory heaps, such as paged or nonpaged pool.

system worker threads

Threads created in the system process during system initialization that exist solely to perform work on behalf of other threads.

system working set

The physical memory being used by the system cache, paged pool, pageable code in Ntoskrnl.exe, and pageable code in device drivers. *See also* process working set.

thread

An entity within a process that Windows schedules for execution. A thread includes the contents of a set of volatile registers representing the state of the processor; two stacks, one for the thread to use while executing in kernel mode and one for executing in user mode; a private storage area for use by subsystems, run-time libraries, and DLLs; and a unique identifier called a thread ID (also internally called a client ID).

thread context

A thread's volatile registers, the stacks, and the private storage area. Because this information is different for each machine architecture that Windows runs on, this structure is architecture-specific. In fact, the CONTEXT structure returned by the Windows GetThreadContext function is the only public data structure in the Windows API that is machine-dependent.

timer

A mechanism that notifies a thread when a fixed period of time elapses.

transaction
An I/O operation that alters file system data or changes the volume's directory structure. The separate disk updates that make up the transaction must be executed atomically; that is, once the transaction begins to execute, all its disk updates must be completed. If a system failure interrupts the transaction, the part that has been completed must be undone, or rolled back. The rollback operation returns the database to a previously known and consistent state, as if the transaction had never occurred.

transaction table
A table that keeps track of transactions that have been started but that aren't yet committed. The suboperations of these active transactions must be removed from the disk during recovery.

transition
A kind of invalid page table entry (PTE) in which the desired page is in memory on either the standby, modified, or modified-no-write list. The page is removed from the list and added to the working set.

translation look-aside buffer (TLB)
A CPU cache of recently translated virtual page numbers.

trap
A processor's mechanism for capturing an executing thread when an exception or an interrupt occurs, switching it from user mode into kernel mode, and transferring control to a fixed location in the operating system. In Windows, the processor transfers control to the kernel's trap handler.

trap frame
A data structure in which the execution state of the interrupted thread is stored. This information allows the kernel to resume execution of the thread after handling the interrupt or the exception. The trap frame is usually a subset of a thread's complete context.

trap handler
A module in the kernel that acts as a switchboard, fielding exceptions and interrupts detected by the processor and transferring control to code that handles the condition.

trusted facility management
Requires support for separate account roles for administrative functions. For example, separate accounts are provided for administration (Administrators), user accounts charged with backing up the computer, and standard users.

trusted path functionality
Prevents Trojan horse programs from being able to intercept users' names and passwords as they try to log in. The trusted path functionality in Windows comes in the form of its Ctrl+Alt+Delete logon-attention sequence, which cannot be intercepted by nonprivileged applications. This sequence of keystrokes, which is also known as the secure attention sequence (SAS), always pops up a logon dialog box, so would-be Trojan horses can easily be recognized. A Trojan horse presenting a fake logon dialog box will be bypassed when the SAS is entered.

type object
An internal system object that contains information common to each instance of the object.

Unicode
An international character set standard that defines unique 16-bit values for most of the world's known character sets.

update records
The most common type of record NTFS writes to the log file. Each update record contains the information needed to redo an operation that updated the file system structure.

user mode
The nonprivileged processor mode that applications run in. A limited set of interfaces is available in this mode, and the access to system data is limited. *See also* kernel mode.

VACB index array
An array of pointers maintained by the cache manager to keep track of which views for a given file are mapped to the system cache.

view
The portion of the section object required by a process. A section object can refer to files that are much larger than can fit in the address space of a process. (If the paging file backs a section object, sufficient space must exist in the paging file to contain it.) To access a very large section object, a process can map only a view of the section by calling the *MapViewOfFile* function and then specifying the range to map. Mapping views permits processes to conserve address space because only the views of the section object needed at the time must be mapped into memory. *See also* section object.

virtual address control blocks (VACBs)
The data structures that the cache manager uses to manage the system address space into which it maps files.

virtual address descriptors (VADs)
Data structures the memory manager maintains that keep track of which virtual addresses have been reserved in the process's address space. VADs are structured as a self-balancing binary tree to make lookups efficient.

virtual address space
A set of virtual memory addresses that a process can use.

virtual block caching
A method the Windows cache manager uses to keep track of which parts of which files are in the cache.

virtual cluster numbers (VCNs)
VCNs number the clusters belonging to a particular file from 0 through m. VCNs aren't necessarily physically contiguous, but they can be mapped to any number of logical cluster numbers (LCNs) on a volume.

virtual device drivers (VDDs)
Drivers used to emulate 16-bit MS-DOS applications. They trap what an MS-DOS application thinks are references to I/O ports and translate them into native Windows I/O functions. Because Windows is a fully protected operating system, user-mode MS-DOS applications can't access hardware directly and thus must go through a real kernel-mode device driver.

virtual memory manager
Implements virtual memory, a memory management scheme that provides a large, private address space for each process that can exceed available physical memory.

volume
One or more logical disk partitions that are treated as a single unit.

volume file
A system file (filename $Volume) that contains the volume name, the version of NTFS for which the volume is formatted, and a bit that when set signifies that a disk corruption has occurred and must be repaired by the Chkdsk utility.

volume manager
A term used to represent both FtDisk and DMIO because both FtDisk and DMIO support the same

multipartition-volume types.

volume set
A single logical volume composed of a maximum of 32 areas of free space on one or more disks.

wait block
A data structure that represents a thread waiting on an object. Each thread that is in a wait state has a list of the wait blocks that represent the objects the thread is waiting on. Each dispatcher object has a list of the wait blocks that represent which threads are waiting on the object.

wait hint
A value indicating how long a service should wait before informing the system a shutdown is complete.

WDM drivers
Device drivers that adhere to the Windows Driver Model (WDM). WDM includes support for Windows power management, Plug and Play, and Windows Management Instrumentation (WMI). WDM is implemented on Windows 2000 and later, Windows 98, and Windows Millennium Edition, so WDM drivers are source compatible between these operating systems and in many cases are also binary compatible. There are three types of WDM drivers: bus drivers, function drivers, and filter drivers.

WDM Windows Management Instrumentation routines
Enable device drivers to publish performance and configuration information and receive commands from the user-mode WMI service. Consumers of WMI information can be on the local machine or remote across the network.

Windows API
The 32-bit interface to Windows 2000 and both the 32-bit and 64-bit programming interfaces to Windows XP and Windows Server 2003.

Windows services
A mechanism to start processes at system startup time that provide services not tied to an interactive user. Services are similar to UNIX daemon processes and often implement the server side of client/server applications.

window station
A window station contains desktops, and desktops contain windows. Only one window station can be visible on a console and receive user mouse and keyboard input. In a Terminal Services environment, one window station per session is visible, but services all run as part of the console session.

Windows drivers
Device drivers that integrate with the Windows power manager and Plug and Play manager, when required. They include drivers for mass storage devices, protocol stacks, and network adapters.

Windows Management Instrumentation (WMI) manager
A component of the executive that enables device drivers to publish performance and configuration information and receive commands from the user-mode WMI service. Consumers of WMI information can be on the local machine or remote across the network.

work item
A unit of work placed on a queue dispatcher object when a device driver or an executive component requests a system worker thread's services by calling the executive functions ExQueueWorkItem or IoQueueWorkItem. Work items include a pointer to a routine and a parameter that the thread passes to the routine when it processes the work item. The routine is implemented by the device driver or executive component that requires passive-level execution.

working set
A subset of virtual pages resident in physical memory. There are two kinds of working sets—process

working sets and the system working set.

working set manager

A routine that runs in the context of the balance set manager system thread to initiate automatic working set trimming to increase the amount of free memory available in the system. Although Windows attempts to keep memory available by writing modified pages to disk, when modified pages are being generated at a very high rate, more memory is required to meet memory demands. The working set manager is called when physical memory runs low (when MmAvailablePages is less than MmMinimumFreePages).

write-back

A caching strategy the lazy write file system uses to improve performance. In write-back, the file system writes file modifications to the cache and flushes the contents of the cache to disk in an optimized way, usually as a background activity.

write-through

An algorithm the FAT file system uses that causes disk modifications to be immediately written to the disk. Unlike the careful-write approach, the write-through technique doesn't require the file system to order its writes to prevent inconsistencies. *See also* careful write.

write throttling

Prevents system performance from degrading because of a lack of memory when a file system or network server issues a large write operation.

zero page thread

A kernel-mode system thread (thread 0 in the system process). A zero page thread zeroes out pages on the free list so that a cache of zero pages is available to satisfy future demand-zero page faults.