Microsoft®

Microsoft® Windows NT™ from a Unix® Point of View

A White Paper from the
Business Systems Technology Series

Microsoft Windows NT™ from a Unix® Point of View

A White Paper from the
Business Systems Technology Series

Abstract

This paper provides a technical overview of Windows NT for the information technology (IT) professional with a strong background in Unix. It approaches the subject from the Unix point of view and relates the concepts of Windows NT to corresponding ones found in Unix. It begins with a technical comparison of the two operating systems, moves on to cover how the two can coexist in a heterogeneous environment, and finishes with a brief section describing some of the tools available to aid developers in creating applications for both platforms.

About the Microsoft Business Systems Technology Series

The Microsoft® Business Systems Technology Series consists of a number of interrelated white papers dedicated to educating information technology (IT) professionals about Windows NT and the Microsoft BackOffice™ family of products. While current technologies used in Microsoft products are often covered, the real purpose of this series is to give the reader an idea of how major technologies are evolving, how Microsoft is using those technologies, and what this means to IT planners.

Legal Notice

The descriptions of other companies' products in this paper are provided only as a convenience to the reader. Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, this document should not be interpreted as a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

Microsoft, MS-DOS, PowerPoint, Visual Basic, Windows, and Win32 are registered trademarks, and BackOffice, SQL Server, Visual C++, and Windows NT are trademarks of Microsoft Corporation.

America Online is a registered trademark of America Online, Inc. AT&T is a registered trademark of American Telephone and Telegraph Company. AppleTalk is a registered trademark of Apple Computer, Inc. Borland is a registered trademark of Borland International, Inc. CompuServe is a registered trademark of CompuServe, Inc. CA-Unicenter is a registered trademark of Computer Associates International, Inc. Alpha AXP, DEC, DECnet, PDP-11, VAX, VMS and VT 100 are trademarks of Digital Equipment Corporation. Hewlett-Packard, HP, and HP-UX are registered trademarks, and MPE is a trademark of Hewlett-Packard Company. Pentium is a registered trademark of Intel Corporation. IBM and OS/2 are registered trademarks and CICS and RACF are trademarks of the International Business Machine Company. Kerberos and X Window System are trademarks of the Massachusetts Institute of Technology. Micro Focus is a registered trademark of Micro Focus Limited. MIPS and R4000 are registered trademarks of MIPS Computer Systems, Inc. NetWare and Novell are registered trademarks of Novell, Inc. (USL). USL is a wholly owned subsidiary of Novell, Inc. DCE and OSF are registered trademarks of the Open Software Foundation. Prodigy is a trademark of Prodigy Services Company. Open Desktop and SCO are registered trademarks of The Santa Cruz Operation, Inc. OpenGL is a trademark of Silicon Graphics, Inc. SunSoft is a trademark of Sun Microsystems, Incorporated. Unicode is a trademark of Unicode, Incorporated. UNIX is a registered trademark in the United States and or other countries licensed exclusively by X/Open, Ltd.

0895 Part No. 098-61913
Printed in the United States of America

Table of Contents

EXECUTIVE SUMMARY
INTRODUCTION 1
A Tale of Two Operating Systems *1*
HISTORICAL PERSPECTIVE 2
UNIX *2*
Windows NT *3*
TECHNICAL COMPARISON: WINDOWS NT AND UNIX 4
Operating System Basics *4*
Networking/Communication *13*
File Systems *17*
Distributed Computing - Directory Services *20*
Security *24*
System Administration *30*
User Interface and Environment *37*
Hardware Platforms *42*
UNIX-BASED ENVIRONMENT ON WINDOWS NT 45
Overview *45*
Network Access Utilities *45*
Tools *46*
Network File Systems *47*
WINDOWS-BASED ENVIRONMENT ON UNIX 49
Overview *49*
Emulation Basics *49*
Commercial Packages *49*
CROSS-PLATFORM APPLICATIONS DEVELOPMENT 51
Overview *51*
Language Support *51*
Conversion Tools *52*
Distributed Computing Environment (DCE) *54*
WISE SDK *55*
CONCLUSION 57
APPENDIX A: ASSISTANCE 58
Customer Support *58*
Windows NT Resource Kit *60*
Books *60*
Technical White Papers *61*
Training Materials *61*
APPENDIX B: CONTACTS 62
Third-Party Vendor Reference *62*
APPENDIX C: GLOSSARY 66

Executive Summary

Companies today are actively re-engineering their information technology (IT) organizations to become closely integrated with mainline business processes. To keep pace with the rapidly changing business environment, organizations must move forward while, at the same time, leveraging their investment in existing systems. For many, integration between Unix® and the Microsoft® Windows®-family of operating systems, especially Windows NT™, has become a critical success factor.

This paper is written for IT professionals with a strong background in Unix. It begins by describing the architecture of Windows NT in a technical framework that is common to Unix. This section also serves to bridge the gaps in terminology between Windows NT and Unix. The paper then moves on to describe many of the integration tools that enable the two operating systems to function in a heterogeneous environment. The final section covers tools and strategies to help with software development targeted for both Windows NT and Unix. The appendices serve as a comprehensive reference for the paper and point the reader in the direction of other pertinent material.

Introduction

A Tale of Two Operating Systems

It was the best of operating systems, it was the worst of operating systems, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of light, it was the season of darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, in short, we had Unix.

Life with Unix tends to be a love-hate relationship, with most people falling strongly on one side or the other. This paper is about another operating system, Windows NT, that, although relatively new, has roots that stretch back almost as far as those of Unix. It is an operating system that, like most, owes much to Unix. But it also strays markedly from it in both design and philosophy.

This paper is written specifically for you, the IT professional with a strong background in Unix. It approaches Windows NT from the Unix perspective. It begins with a brief look at the pedigree of both operating systems, progresses to a technical comparison of the two, moves on to show how Windows NT and Unix can peacefully coexist in a heterogeneous world, and finishes with a discussion about tools that are available to aid developers working in both environments.

This is a technical paper, and it is assumed that you have a good working knowledge of Unix and have at least seen a computer running Windows. No attempt will be made to teach fundamentals here. However, “Appendix A: Assistance” lists resources that can help you fill in the gaps.

Throughout the paper you will see reference to *Windows*, which is the Microsoft operating system. Do not confuse this with the Unix *X Window System*™, also known as *X Windows* or just *X*. Windows will be used to refer to Microsoft Windows, and X will be used to refer to the Unix X Window System.

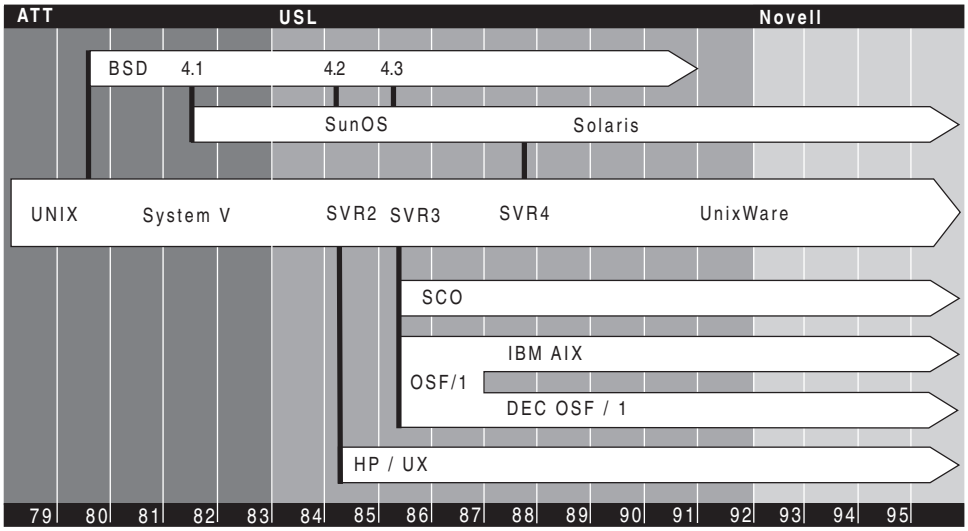
Windows NT and Unix address the same world, but from very different points of view. It is a mistake to assume that Windows NT provides a direct, feature-for-feature replacement for Unix. It does not. It represents a different perspective, a paradigm shift, and, as such, offers fresh solutions to some very old, chronic problems.

UNIX

In 1968, while working for the Computer Research Group at Bell, Ken Thompson and Dennis Ritchie were part of a joint Bell Labs/MIT team that was building a timesharing operating system for the General Electric GE645 mainframe. Although visionary for its time, the system, known as MULTICS (MULTiplexed Information and Computing System), had serious drawbacks and never found favor with AT&T management.

During the MULTICS project, Ken became interested in a program called Space Travel, whose salient feature was a spaceship that could be piloted through a simulated galaxy. Unfortunately, Space Travel did not run well under MULTICS on the GE645 mainframe. Undaunted, Ken took the logical approach to the problem. He borrowed a PDP-7 minicomputer from another group at Bell, wrote an operating system, and ported Space Travel to it. In 1970, Brian Kernighan jokingly called this new operating system UNICS (UNiplexed Information and Computing System) in reference to the much larger MULTICS. Soon after, the name was changed to Unix.

AT&T was not allowed to market Unix or Unix-based products due to a long-standing antitrust decree. This meant that Unix had no commercial viability for AT&T and was never treated as a real product. Source code was made available to other groups within AT&T and, for educational purposes, to universities. New versions of Unix proliferated, resulting in a complex and convoluted history that in many ways resembles what is going on today with the Internet. Standardization efforts have been under way for many years, and most extant versions can trace their lineage back to one of two sources: AT&T® System V or 4.xBSD (Berkeley Unix) from the University of California, Berkeley, Computer Science Department. Still, major incompatibilities exist between the different versions so that, even today, Unix is not “Unix.”



Evolution of Unix

Windows NT

In the 1970s, Unix was not the only game in town. One of the most successful general-purpose minicomputer operating systems of that time came from Digital Equipment Corporation (Digital). It was VMS™, and it supported Digital's VAX™ architecture. The person responsible for leading the VMS development effort was Dave Cutler.

In 1988, Dave joined Microsoft to lead the development effort for the new high-end operating system in the Microsoft Windows-family, Windows NT. Two primary forces shaped the Windows NT project: market requirements and sound design. The market requirements came from input from its customers around the world. The design goals came from leading-edge thinking in operating system theory and design. These same forces continue to drive the development of the Windows NT platform today.

Under the market requirements, Windows NT must provide:

- Portability across families of processors, such as the Intel x86 line.
- Portability across different processor architectures, such as CISC and RISC.
- Transparent support for single-processor and multiprocessor computers.
- Support for distributed computing.
- Standards compliance, such as POSIX.
- Certifiable security, such as C2 and F-C2, E3.

The design goals for Windows NT compliment the market requirements. They are:

- Extensibility
- Portability
- Reliability
- Robustness
- Performance
- Compatibility

Windows NT continues to blend together some of the best ideas from industry and academia on operating system theory with real-world experience in operating system design. In many ways, the result fulfills Ken Thompson's and Dennis Ritchie's vision for a portable, extensible, open operating system.

Operating System Basics

Overview and Philosophy

Windows NT was designed for client/server computing; Unix was designed for host-based terminal computing. They are two different computing paradigms. Windows NT is not a multiuser operating system in the usual sense of the word (although it can be with additional third-party products). You do not have users with limited-function dumb terminals, dumb terminal emulators, or X-terminals connecting to a Windows NT-based host. What you do have are users on single-user general-purpose workstations (clients) connecting to multiuser general-purpose servers with the processing load shared between both. The distinction between the two environments is subtle, but understanding it is key to understanding Windows NT.

Unix, X, and Server/Client

Windows NT follows what has become the industry-standard terminology for the client/server relationship, with the client being on the desktop and the server being in the back office. Unfortunately, there is a portion of the Unix world in which this relationship is reversed. The client is the server, and the server is the client.

The most common graphical user interface on Unix is X, which was developed at the Massachusetts Institute of Technology (MIT) as part of Project Athena. In X, the client software resides on the computer that performs the processing, and the server software resides on the computer that displays the output. This means that the desktop is the server, not the client. This paper will stick with today's standard client/server terminology and avoid the server/client terminology of X.

Kernel Mode vs. User Mode

In modern operating systems, applications are kept separate from the operating system itself. The operating system code runs in a privileged processor mode known as kernel-mode and has access to system data and hardware. Applications run in a nonprivileged processor mode known as user mode and have limited access to system data and hardware through a set of tightly controlled application programming interfaces (APIs).

Windows NT is a microkernel-based operating system, which shares similarities to Mach, a microkernel-based operating system developed at Carnegie Mellon University. One of the primary design goals of Windows NT was to keep the base operating system as small and as tight as possible. This was accomplished by allowing only those functions that could not reasonably be performed elsewhere to remain in the base operating system. The functionality that was pushed out of the kernel ended up in a set of nonprivileged servers known as the protected subsystems. The protected subsystems provide the traditional operating system support to applications through a feature-rich set of APIs.

This design results in a very stable base operating system. Enhancements occur at the protected subsystem level. In fact, new protected subsystems can be added without modification to either the base operating system or the other existing protected subsystems. Contrast this with traditional Unix systems where the kernel has become the dumping ground for any and all functionality.

A number of other theoretical models influenced the design of Windows NT. Three of the most important ones were: the client/server model, the object-oriented model, and the

symmetric multiprocessing model. These models provide a framework for understanding the inner workings of Windows NT.

Client/server

Windows NT is a natural fit for the world of client/server computing. Not only does it live in this world, but its internal design is based on client/server principles. For example, the applications that an user runs are clients that request services from the protected subsystems which are servers. The idea is to divide the operating system into several discrete processes, each of which implements a set of cohesive services such as process creation or memory allocation. These processes communicate with their clients, each other, and the kernel by passing well-defined messages back and forth.

The client/server approach results in a modular operating system. The servers are small and self-contained. Because each runs in its own protected, user-mode process, a server can fail without taking down the rest of the operating system with it. This self-contained nature of the operating system components also makes it possible to distribute them across multiple processors on a single computer (symmetric multiprocessing) or even multiple computers on a network (distributed computing).

Object-Based

Windows NT is not an object-oriented system in the strictest sense of the term, but it does use objects to represent internal system resources. Software objects are a combination of computer instructions and data that model the behavior of things, real or imagined, in the world. Objects are composed of three things: 1) attributes in the form of program variables that collectively define the object's state, 2) behavior in the form of code modules or methods that can modify those attributes, and 3) an identity that distinguishes one object from all others. Objects interact with each other by passing messages back and forth. The sending object is known as the client and the receiving object is known as the server. The client requests and the server responds, and often times in the course of conversation the client and server roles alternate between objects.

One of the hidden powers behind Unix is the file metaphor. In Unix, devices such as printers, tape drives, keyboards, and terminal screens all appear as ordinary files to both programmers and regular users. This simplifies many routine tasks, and is a key component in the extensibility of the system. Windows NT capitalizes on this metaphor and expands it. Windows NT uses an object metaphor that is pervasive throughout the architecture of the system. Not only are all of the things in the Unix file metaphor viewed as objects by Windows NT, but so are things such as processes and threads, shared memory segments, and access rights.

Symmetric Multiprocessing

Multitasking and multiprocessing are two terms that are closely related and easily confused. Multitasking is an operating system technique for sharing a single processor among multiple threads of execution. Multiprocessing, on the other hand, refers to computers with more than one processor. A multiprocessing computer is one that is able to execute multiple threads simultaneously, one for each processor in the computer. A multitasking operating system *appears* to execute multiple threads at the same time; a multiprocessing operating system actually does it, executing one thread on each of the computer's processors.

Multiprocessing operating systems can be either asymmetric or symmetric. The main difference is in how the processors are used. In asymmetric multiprocessing (ASMP), the operating system typically sets aside one or more processors for its exclusive use. The remainder of the processors run user applications. In symmetric multiprocessing (SMP), any processor can run any type of thread. The processors communicate with each other through shared memory.

SMP systems provide better load-balancing and fault-tolerance. Since the operating system threads can run on any and all processors, the chance of hitting a CPU bottleneck is greatly reduced over the ASMP model. A processor failure in the SMP model will only reduce the computing capacity of the system. In the ASMP model, it can easily take down the whole computer if it happens to be one of the operating system processors that fails.

SMP systems, such as Windows NT and many flavors of Unix, are inherently more complex than ASMP ones. There is a tremendous amount of coordination that must take place within the operating system to keep everything synchronized. For this reason, SMP systems are usually designed and written from the ground up. Their ASMP counterparts typically are not.

Open Systems

Finally, before delving into the inner workings of Windows NT, we need to touch on the subject of Open Systems and Industry Standards. Unfortunately, Open Systems means many different things to many different people. However, the goal of Open Systems is the same regardless of the definition of the term. Its primary aim is to give the customer cost-effective choices in a multivendor world that is in a constant state of change.

The subject of Industry Standards is every bit as perilous as that of Open Systems. There are two categories of standards: de jure and de facto. De jure standards are those that have been created by standards bodies such as the American National Standards Institute (ANSI), the Institute of Electrical and Electronic Engineers (IEEE), and the International Standards Organization (ISO). Examples of de jure standards are the ANSI American Standard Code for Information Interchange (ASCII) character encoding standard, the IEEE Portable Operating System Interface for Unix (POSIX) standard, and the ISO Open System Interconnection (OSI) reference model for computer networks. De facto standards are those that have been widely adopted by industry, but not originally endorsed by any of the standards bodies. An example of a de facto standard is the Transmission Control Protocol/Internet Protocol (TCP/IP) network communications protocol. De facto standards have arisen either to fill gaps left by the implementation specifications of the de jure standards or because no standard had yet been defined for the particular area.

It is safe to say that Open Systems based solely on de jure Industry Standards have yet to be fully realized, and it is doubtful that they ever will be. At the heart of the problem is the very different natures of the computer industry and the academic standards process. The speed with which the technology is changing is staggering, and the formal standards process is inherently unable to keep pace with it. It is sort of like what happens when an all-powerful force meets an immovable object, one or the other has to give way. In this case the all-powerful force is the market, which always has its way. So, is there no hope for Open Systems?

The utopian “de jure”-centric world of Open Systems has largely given way to the newer, reality-based recognition that real-world Open Systems cannot live on de jure standards

alone. In today's Open Systems, both de facto and de jure standards are combined to create interoperable systems. It is the strategic combining of both types of standards, this middle-of-the-road approach, that enables Open Systems to keep pace with the rapidly changing nature of technology.

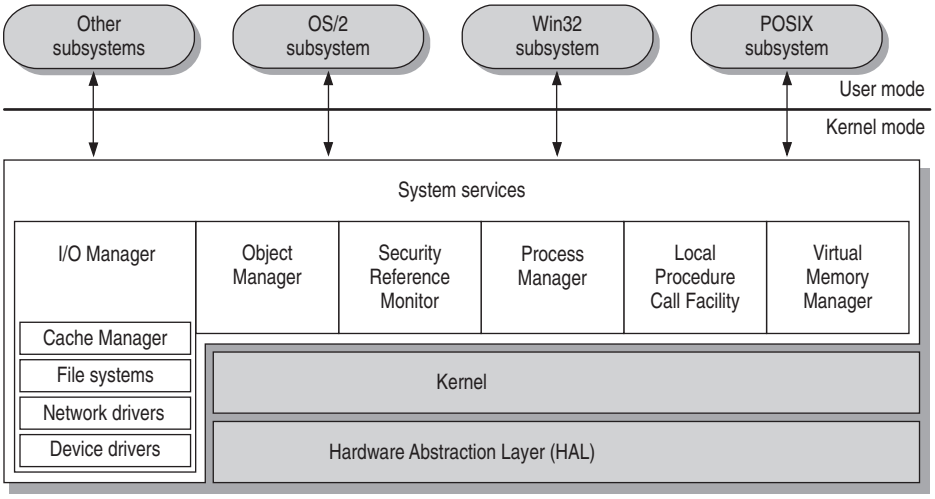
One key element of this approach is the use of strategically-placed layers of software that allow the upper and lower adjoining software layers within the operating system to be loosely coupled. These tightly controlled software layers provide a standardized and well-publicized set of APIs to both the software above and below. A good example is the Network Device Interface Specification (NDIS) that was developed jointly by Microsoft and 3Com in 1989. Another is the Desktop Management Interface (DMI) created by the Desktop Management Task Force (DMTF), an industry organization with more than 300 vendor members including Microsoft, Intel, IBM, and Novell.

The beauty of this architecture is that it allows for plug-and-play of modules above and below the isolation layer, and that means you can start out with a module that implements a de facto standard and later supplement or replace it with one that implements a de jure standard. In effect, you end up with the best of both worlds; Open Systems and Industry Standards.

A more complete discussion of Open Systems and Industry Standards can be found in a recent report from The Burton Group. [1] Specifically, the contrast between de jure and de facto standards, the concept of what it means to be "effectively open", and the enabling power of software "isolation layers" are covered in detail.

Executive

The Windows NT executive is the kernel-mode portion of Windows NT and, except for a user interface, is a complete operating system unto itself. It differs dramatically in architecture from the Unix kernel, and, unlike the Unix kernel, the Windows NT executive is never modified and recompiled by the system administrator.



Windows NT Executive and its Components

The Windows NT executive is actually a family of software components that provide basic operating system services to the protected subsystems and to each other. The

[1] "A Market-Driven Approach To Open Systems." The Burton Group, December 1994.

executive components are completely independent of one another and communicate through carefully controlled interfaces. This modular design allows existing executive components to be removed and replaced with ones that implement new technologies or features. As long as the integrity of the existing interface is maintained, the operating system runs as before.

Object Manager. The Object Manager creates, manages, and deletes Windows NT executive objects. Executive objects are abstract data types used to represent operating system resources such as shared memory segments, files and directories, and processes and threads. The Object Manager manages the global namespace for Windows NT which is modeled after the hierarchical file system, where directory names in a path are separated by a backslash (\). As with other Windows NT components, the Object Manager is extensible and modular so that new object types can be defined as the technology advances.

Process Manager. A program is a static sequence of computer instructions. A process is the dynamic invocation of a program along with the system resources needed for the program to run. A Windows NT-based process differs from a Unix-based process in that it is not an executable entity. A Windows NT-based process contains one or more executable entities known as threads, and it is these threads and not the process that the Windows NT kernel schedules for execution. Remember that Windows NT supports symmetric multiprocessing. In order for an SMP system to work, processes must be dividable into multiple threads of execution.

The process model for Windows NT works in conjunction with the security model and the Virtual Memory Manager to provide interprocess protection. The Process Manager is the Windows NT-based component that manages the creation and deletion of processes. It provides a standard set of services for creating and using threads and processes in the context of a particular protected-subsystem environment. Beyond that, the Process Manager does little to dictate rules about threads and processes. It does not impose any hierarchy or grouping rules for processes, nor does it enforce any parent/child relationships. Instead, these design decisions are left for the protected subsystems to implement. For example, the parent/child relationship that exists between Unix processes is implemented in the POSIX protected-subsystem of Windows NT.

Virtual Memory Manager. Both Windows NT and Unix implement 32-bit linear memory addressing and demand-paged virtual memory management. Under Windows NT, each process is allocated a four gigabyte (2^{32} bytes) virtual address space, with half of it reserved for program storage and half of it reserved for system storage. The Virtual Memory Manager maps virtual addresses in the process' address space to physical pages in the computer's memory. In doing so, it hides the physical organization of memory from the process's threads. This ensures that the thread can access its process' memory as needed, but not the memory of other processes.

An ordinary machine today typically has much less than four gigabytes of physical memory. When physical memory becomes full, the Virtual Memory Manager transfers, or pages, some of the memory contents to disk. Windows NT and many Unix systems share a common page size of 4K. The process that the Virtual Memory Manager uses to determine which pages to move to disk is referred to as the paging policy.

The primary goal of any paging policy is to allow as many processes or threads as possible to use the machine's memory without adversely impacting the overall

performance of the system. It is a very fine balancing act. On one side you have wasted machine resources, on the other you have a situation in which the memory manager uses up most of the CPU cycles by swapping things back and forth from memory to disk, a condition known as thrashing.

The Windows NT Virtual Memory Manager uses a paging policy known as local first in, first out (FIFO) replacement. With local FIFO replacement, the Virtual Memory Manager must keep track of the pages currently in memory for each process. This set of pages is referred to as the process's working set.

One of the most important features of this paging policy is that it enables Windows NT to do some unattended performance tuning. When physical memory runs low, the Virtual Memory Manager uses a technique called automatic working-set trimming to increase the amount of free memory in the system. Roughly speaking, this is a process whereby the Virtual Memory Manager attempts to equitably allocate memory to all of the processes currently on the machine. As it cuts the amount of memory to each process it also monitors their page-fault rates and works to strike a balance between the two.

Local Procedure Call Facility. Applications and the protected subsystems have a client/server relationship. That is, the application (client) makes calls to the protected subsystem (server) to satisfy a request for some type of system service. In general, clients and servers communicate with each other through a series of well-defined messages. When the client and server are both on the same machine, the Windows NT executive uses a message-passing mechanism known as the Local Procedure Call (LPC) facility. LPC is an optimized version of the industry-standard Remote Procedure Call (RPC) facility that is used by clients and servers communicating across networks.

I/O Manager. The I/O Manager is the part of the Windows NT executive that manages all input and output for the operating system. It is made up of a series of subcomponents such as the file systems, the network redirector and server, the system device drivers, and the cache manager. A large part of the I/O Manager's role is to manage the communications between drivers. To simplify the task, it implements a well-defined, formal interface that allows it to communicate with all drivers in the same way, without any knowledge of how the underlying devices actually work. In fact, the I/O model for Windows NT is built on a layered architecture that allows separate drivers to implement each logically distinct layer of I/O processing. The I/O Manager is the Windows NT executive component that makes the most use of the software isolation layers mentioned above in the Open Systems discussion.

In addition to the uniform driver model, the I/O Manager works with other Windows NT executive components, most notably the Virtual Memory Manager, to provide asynchronous I/O, mapped file I/O, and file caching. The latter bears special mention. File caching is controlled by a subcomponent of the I/O Manager called the Cache Manager. While most caching systems allocate a fixed number of bytes for caching files in memory, the Windows NT cache dynamically changes size depending on how much memory is available. This load-balancing feature is provided by the Cache Manager and is another example of automatic self-tuning within Windows NT.

Security Reference Monitor. The Security Reference Monitor, in conjunction with the Logon Process protected-subsystem and the Security protected-subsystem, forms the security model for Windows NT. In a multitasking operating system, applications share a variety of system resources including physical memory, I/O devices, files and directories,

as well as the system processor(s). Applications must have proper authorization before being allowed to access any of these system resources, and it is the components of the security model for Windows NT that enforce this policy.

With Unix, security of this nature usually comes in the form of an expensive add-on such as Kerberos™. Kerberos, which gets its name from the three-headed dog that guards the entrance to Hades in Greek mythology, was developed at MIT as part of Project Athena. It has become a de facto standard for network security and is expected to be incorporated into a future version of Windows NT.

The Security Reference Monitor acts as the watchdog, enforcing the access-validation and audit-generation policy defined by the local Security protected-subsystem. It provides run-time services to both kernel-mode and user-mode components for validating access to objects, checking for user privileges, and generating audit messages. As with other components in the Windows NT executive, the Security Reference runs exclusively in kernel-mode.

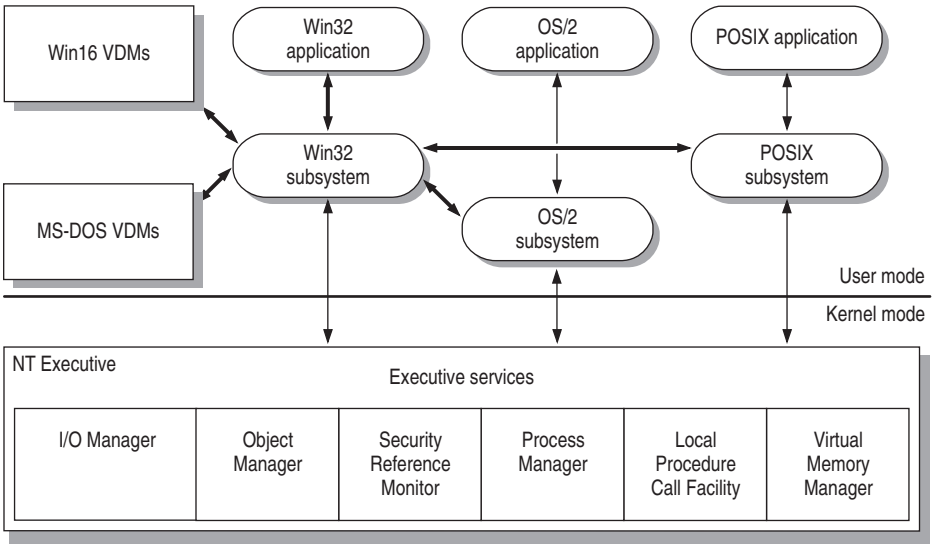
Kernel. The Kernel is at the core of the layered architecture for Windows NT and manages only the most basic of the operating system functions. The microkernel design enables this component to be small and efficient. The Kernel is responsible for thread dispatching, multiprocessor synchronization, and hardware exception handling.

Hardware Abstraction Layer (HAL). The Hardware Abstraction Layer (HAL) is an isolation layer of software provided by the hardware manufacturer that hides, or abstracts, hardware differences from higher layers of the operating system. Because of the HAL, the different types of hardware all look alike to the operating system, removing the need to specifically tailor the operating system to the hardware with which it communicates. The goal for the HAL was to provide routines that allow a single device driver to support the same device on all platforms.

HAL routines are called from both the base operating system (including the Kernel) and from the device drivers. The HAL enables device drivers to support a wide variety of I/O architectures without having to be extensively modified. The HAL is also responsible for hiding the details of symmetric multiprocessing (SMP) hardware from the rest of the operating system.

Protected (Environment) Subsystem

The protected subsystems are user-mode servers that are started when Windows NT is booted. There are two types of protected subsystems: integral and environment. An integral subsystem is a server that performs an important operating system function, such as security. An environment subsystem is a server that provides support to applications native to different operating system environments. Windows NT currently ships with three environment subsystems: the Win32® subsystem, the POSIX subsystem, and the OS/2® subsystem. The Win32 subsystem is the “native-mode” subsystem of Windows NT. It provides the most capabilities and efficiencies to its applications and, for that reason, is the subsystem of choice for new software development. The POSIX and OS/2 subsystems provide “compatibility-mode” environments for their respective applications and, by definition, are not as feature-rich as the Win32 subsystem.



Conceptual View of Windows NT Protected Subsystems

Win32. The Win32 subsystem is the most critical of the Windows NT environment subsystems. It provides the graphical user interface and controls all user input and application output. It is the server for Win32-based applications and implements the Win32 API. Not all applications are Win32-based, and the Win32 subsystem does not control the execution of non-Win32-based applications. It does, however, get involved. When the user runs an application that is foreign to the Win32 subsystem, it determines the application-type and either calls another subsystem to run the application or creates an environment for MS-DOS® or 16-bit Windows in which to run the application.

The subsystems for MS-DOS and 16-bit Windows run in user-mode in the same way the other environment subsystems do. However, unlike the Win32, POSIX, and OS/2 subsystems, they are not server processes, per se. MS-DOS-based applications run within the context of a process called a Virtual DOS Machine (VDM). A VDM is a Win32-based application that establishes a complete virtual computer running MS-DOS. The 16-bit Windows environment is a hybrid application, one that runs within the context of a VDM process, but calls the Win32 API to do most of its work. This feature of running 16-bit Windows within Win32 is known as WOW, which is short for Windows On Win32.

Creating environments for MS-DOS and 16-bit Windows as user-mode subsystems affords them the same protection that the other subsystems have. They cannot interfere with the operation of each other, the other protected-subsystems, or the Windows NT executive.

POSIX. POSIX, which stands for Portable Operating System Interface for Unix, began as an effort by the IEEE community to promote the portability of applications across Unix environments by developing a clear, consistent, and unambiguous set of standards. POSIX is not limited to the Unix environment and has been implemented on non-Unix operating systems, such as Windows NT, VMS, MPE™/iX, and CTOS.

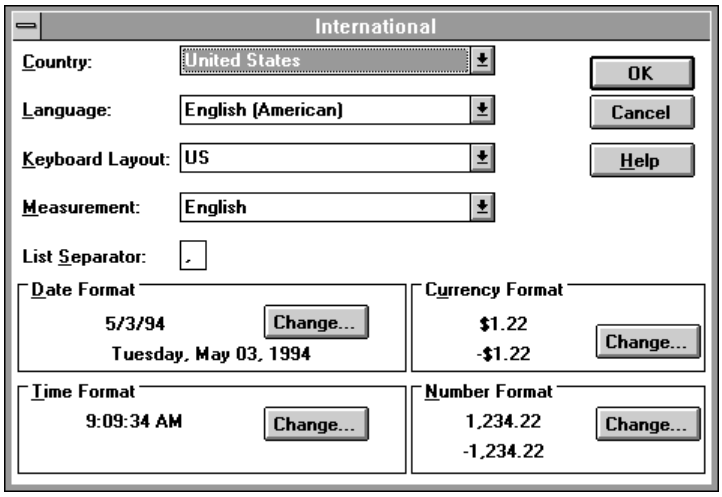
POSIX actually consists of a set of standards that range from IEEE 1003.0 to 1003.22 (also known as POSIX.0 to POSIX.22). However, most of these standards are still in the proposed state. The POSIX subsystem on Windows NT supports 1003.1, which is also

known as the international ISO/IEC IS 9945-1:1990 standard. This standard defines a C-language, source-code-level API to the operating system environment. All POSIX standards are based on specifications for which there is no binary reference implementation.

OS/2. The OS/2 subsystem supports 16-bit graphical and character based applications. It provides these applications with an execution environment that looks and acts like a native OS/2 system. Internally, the OS/2 subsystem calls the Windows NT executive to do most of the work, as the Windows NT executive services provide general-purpose mechanisms for doing most operating system tasks. However, the OS/2 subsystem implements those features that are unique to its operating environment.

Internationalization

Microsoft builds products that are *world-ready*; Windows NT is no exception. There are currently 15 international versions of Windows NT. They are Brazilian, Chinese, Danish, Dutch, Finnish, French, German, Italian, Japanese, Korean, Norwegian, Portuguese, Russian, Spanish, and Swedish. A process called localization is used to create these different versions.



Specifying International Settings

When installing Windows NT, the user selects a language to use and is assigned a default locale. The default locale gives the culturally-correct defaults for keyboard layout, sorting order, currency, and date and time formatting. Of course, these defaults can be overridden by the user.

At its most basic level, a locale consists of a language, a country, and the binary codes used to represent the characters of a particular language. The latter is referred to as the code set. The United States has traditionally adopted the ASCII standard for representing data. However, ASCII is woefully inadequate for some other countries because it lacks many of their common symbols and punctuation. For example, the British pound sign and the diacritical marks used in French, German, Dutch, and Spanish are missing.

To address these shortcomings, Windows NT employs the new Unicode™ standard for data representation. Unicode is a de jure standard for encoding international character

sets. The Unicode standard was developed by the Unicode Consortium, a consortium of vendors including Microsoft, IBM, Borland, and Lotus. Unicode separates the "essence" of a character from the font and formatting information used to display it. It employs a 16-bit character coding scheme, which means that it can represent 65,536 (2^{16}) individual characters. This is enough to include all languages in computer commerce today, several archaic or arcane languages with limited applications (such as Sanskrit, and, eventually, Egyptian hieroglyphics), all punctuation marks, mathematical symbols, and other graphical characters. With all of this, there is still plenty of room for future growth.

Unicode is the native code set of Windows NT, but the Win32 subsystem provides both ASCII and Unicode support. Character strings in the system, including object names, path names, and file and directory names, are represented with 16-bit Unicode characters. The Win32 subsystem converts any ASCII characters it receives into Unicode strings before manipulating them. It then converts them back to ASCII, if necessary, for output.

Localization is only one part of the effort that goes into ensuring that an operating system can be used effectively in a worldwide environment. A *world-ready* operating system must also provide services to support the use of international applications and to support the global market by making the application developer's job easier. For example, some of the language issues that international users and application developers face are:

- **From the user's perspective:** Some users need to include more than one language in a document. For example, they might be translating from English into Russian, or they might be writing a product instruction manual in many different languages. When using more than one language, users must deal with a series of obstacles. For example, they must repeatedly switch to another keyboard layout on-the-fly so that they can continue writing in a different language. When using a database, users face the problem of sorting the information in the correct order for a given language.
- **From the developer's perspective:** When localizing a product, developers are faced with several questions, such as the following: "What is the correct sorting order for French?", "How is a date represented in Germany?", "If a document contains text in more than one language, is there some way for the software to know which part of the document is in which language?", "Can information in a multilingual document be cut and pasted into another application?" Many developers try to address these issues in their applications and fall short, creating problems for the users, their support organization, and their own development team.

Microsoft is incorporating international language support at the operating system and API level. Built-in, international language support adds functionality that provides solutions for developing and using software and exchanging documents around the world.

Networking/Communication

Overview

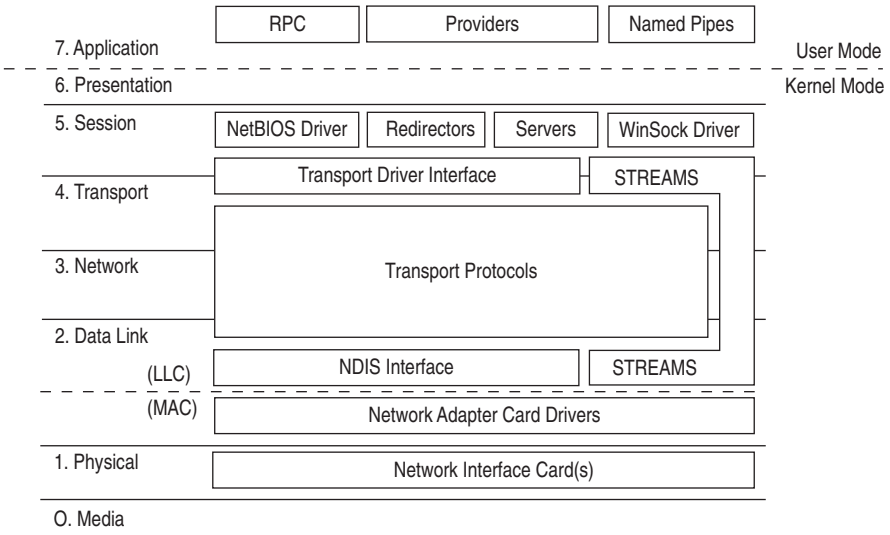
Windows NT is a complete operating system with fully integrated networking, including built-in support for multiple network protocols. These capabilities differentiate it from other operating systems such as MS-DOS and most versions of Unix. With these operating systems, network capabilities are either installed separately from the core operating system as an add-on or the range of supported network protocols is limited to only a few. With some, both limitations are true.

Windows NT offers built-in support for both peer-to-peer and client/server networking. It provides interoperability with and remote dial-in access to existing networks, support for distributed applications, file and print sharing, and the ability to easily add networking software and hardware.

ISO OSI Model

The International Standards Organization (ISO) developed a seven-layer, theoretical model called the Open Systems Interconnection (OSI) reference model. It is used to describe the flow of data between the physical connection to the network and the user application. The model is the best known and most widely used model to describe networking environments, and we will use it as the framework to discuss the networking components for Windows NT.

In the OSI model, the purpose of each of the seven layers is to provide services to the next higher layer, shielding the higher layer from the details of how the services are actually implemented. The layers are abstracted in such a way that each layer believes it is communicating with the same layer on the other computer. In reality, each layer communicates only with adjacent layers on the same machine.



Windows NT Networking Model

Layer 0, which is not officially a layer in the OSI model, is commonly used to define the underlying transmission media, such as cables or fiber, that interconnects each of the computers on the network. Layer 0 is known as the Media Layer.

Network Adapter/Network Interface Card (NIC)

The Network Adapter or Network Interface Card (NIC) connects the internal communication bus of the computer with the external network. It acts as a bridge between the Media Layer (Layer 0) and the Physical Layer (Layer 1) in the OSI model. Windows NT views the NIC as a peripheral device and controls it through a device driver.

Network Device Interface Specification (NDIS)

The IEEE 802 project further defined sublayers of the Data Link Layer (Layer 2) in the OSI model. The two sublayers are the Media Access Control (MAC) and the Logical Link Control (LLC). The MAC sublayer communicates directly with the NIC and is responsible for delivering error-free data between two computers on the network.

In 1989, Microsoft and 3Com jointly developed a specification defining an interface for communication between the MAC sublayer and protocol drivers higher in the OSI model. This standard is known as the Network Device Interface Specification (NDIS), and is a key isolation layer of software. NDIS isolates the details of the NIC from the transport protocols and vice versa.

Transport Protocols

The transport protocols reside primarily in the Network Layer (Layer 3) and the Transport Layer (Layer 4) of the OSI model and communicate with the NIC(s) through an NDIS-compliant device driver. Windows NT ships with the following transport protocols: TCP/IP; NBF, derived from NetBEUI; NWLink, an NDIS-compliant version of Novell® Internetwork Packet Exchange (IPX/SPX); Microsoft Data Link Control (DLC); and AppleTalk®.

STREAMS

STREAMS was originally developed by AT&T for Unix System V, Release 3.2. It is an isolation layer of software that wraps around STREAMS-based transport protocols. Calls to the transport protocol driver must first go through the upper layer of the STREAMS device driver to the protocol, then back through the lower layer of STREAMS to the NDIS device driver. The STREAMS environment allows the many STREAMS-based transport protocol drivers that already exist to be plugged into Windows NT with little or no modification. New transport protocol drivers, however, should be written to the newer, more versatile Transport Driver Interface.

Transport Driver Interface (TDI)

The Transport Driver Interface (TDI) is another isolation layer of software that falls at another strategic breakpoint in the OSI model, namely between the Transport Layer (Layer 4) and the Session Layer (Layer 5). The TDI is not a single piece of software but rather a protocol specification to which the upper bounds of the transport protocol device drivers are written. It enables a single version of a session-layer component, such as a network redirector or server, to use any available transport mechanism loaded on the machine, for example TCP/IP or IPX/SPX.

Windows Sockets (WinSock)

In network programming, a socket provides an endpoint to a connection; two sockets form a complete path. A socket works as a bi-directional pipe for incoming and outgoing data between networked computers.

Windows Sockets (WinSock), a session-layer interface, is a de facto standard for Windows-based network programming. Version 1.1 of Windows Sockets was developed by a group of 30 vendors, including Microsoft, and released in January 1993. This original version is compatible with the UC Berkeley (BSD) Sockets APIs, which are a de facto standard for Unix network programming. Version 1.1 provided independence from

the underlying TCP/IP protocol stack. As long as the TCP/IP stack was WinSock-compliant, an application written to the WinSock APIs would run on it.

Version 2.0 of Windows Sockets provides true transport protocol independence by extending support to additional protocols, including IPX/SPX, DECnet™, and OSI. It has also been extended to support additional network technologies, such as ATM, wireless, and telephony.

NetBIOS

The Network Basic Input/Output System (NetBIOS) is a session-layer interface similar in function to Windows Sockets. It is used by applications to communicate with NetBIOS-compliant transports such as NetBEUI Frame (NBF). The NetBIOS interface is responsible for establishing logical names on the network, establishing a connection between any two of those names, and supporting reliable data transfer between computers once the connection has been established. The network redirector is an example of a NetBIOS application.

Redirector/Server

The redirector and server are integral subsystems that are key components of the network architecture for Windows NT. The redirector is the network component responsible for sending, or redirecting, I/O requests across the network when the file or device to be accessed is not on the local machine. The server is the network component on the remote machine that entertains connection requests from the client-side redirectors and provides them with access to the desired resources. Under Windows NT, multiple redirector/server pairs can concurrently execute, enabling transparent, multi-server access.

Both of these components reside above the TDI and are implemented as file system drivers. This has several benefits. Applications can call a single API to access both local and remote files, and, from the I/O Manager's perspective, there is no difference between accessing files stored on a remote networked computer and accessing those stored on a local hard disk. This transparent resource access is, in many ways, similar to the functionality provided by remote file systems under Unix, such as the Network File System (NFS) and the Andrew File System (AFS).

Provider

For each redirector in the Session Layer (Layer 5) there is a corresponding component known as a *provider* in the Application Layer (Layer 7) of the OSI model. A provider establishes Windows NT as a client of a remote network server. When an application issues a request, a software component known as the Multiple Provider Router (MPR) determines the appropriate provider and routes the request to it. The provider then passes the request on to the corresponding redirector for transmission across the network. Typical operations performed by a provider are establishing and releasing network connections, transferring data across the connection, and printing remotely. The Windows NT File Manager is an example of an application that uses the services provided by the provider(s).

While Windows NT includes integrated networking, its open design provides for transparent access to other networks. Windows NT supplies provider/redirector pairs for its own network as well as others. For example, it includes the Client Service for NetWare® with Windows NT Workstation and the Gateway Service for NetWare with

Windows NT Server, with which a computer running Windows NT can connect as a client to a NetWare network. You can also install third-party provider/redirector pairs to expand the connectivity of Windows NT beyond what is already supported "out of the box."

Named Pipes

Though not compatible with Unix Named Pipes, Windows NT Named Pipes are conceptually similar. Named pipes provide a high-level interface for passing data between two processes, regardless of network location. Named pipes, like files, are implemented as file objects in Windows NT and operate under the same constraints and security mechanisms as other NT executive objects. The named pipe file system driver is a pseudo-file system that stores pipe data in memory and retrieves it on-demand. When processing local or remote named pipe requests, it functions like an ordinary file system.

Remote Procedure Call (RPC)

The Remote Procedure Call (RPC) facility is the backbone of true distributed computing and is rapidly becoming the InterProcess Communication (IPC) method of choice for software developers. Much of the original design work for an RPC facility was started by Sun Microsystems. It has continued with the Open Software Foundation (OSF) as a core part of their Distributed Computing Environment (DCE) standard.

The Microsoft RPC is compatible with the OSF® DCE® RPC. Being compatible is not the same as being compliant. Compliance, in this case, means starting with the OSF source code and building upon it. The key element here is not compliance; it is interoperability. If the software is interoperable, it does not need to be compliant, and the Microsoft RPC facility is completely interoperable with other DCE-based RPC systems, such as those from Hewlett-Packard and IBM.

The RPC facility is unique because it relies on other IPC mechanisms to transfer functions and data between the client and the server. In the case of Windows NT, RPC can use named pipes, NetBIOS, or Windows Sockets to communicate with remote systems and the LPC facility to communicate with systems on the local machine. This IPC-independence makes RPC the most flexible and portable of the IPC mechanisms for Windows NT.

File Systems

Overview

Windows NT automatically supports multiple file systems. The two most commonly used are:

- **The Windows NT File System (NTFS)** is a new advanced file system that supports file recovery, extremely large storage media, and long file names. NTFS provides the highest level of security and is the file system of choice on Windows NT.
- **The File Allocation Table (FAT)** is the file system used by the MS-DOS and Windows operating systems. FAT does not provide security.

A computer running Windows NT can use one or more of these file system on its disks and partitions. This is referred to as multiple active file systems. Our interest in this paper is with the NTFS file system because it most closely parallels features found in Unix file systems.

NTFS

Journaling file systems are based on the transaction processing concepts found in database theory. NTFS is a journaling file system with fast file recovery. Internally, it more resembles a relational database than a traditional file system. It is comparable in function to the Veritas file system found on some Unix implementations, and the really good news is you do not have to deal with **fsck** or **lost+found**.

NTFS was designed to provide recoverability, security, and fault tolerance through data redundancy. In addition, support was built into NTFS for large files and disks, unicode-based names, bad-cluster remapping, multiple data streams, general indexing of file attributes, and POSIX. All of these contribute to making NTFS an extremely robust file system.

Fault Tolerance

Fault tolerance is the ability of a system to continue functioning when part of the system fails. The expression *fault tolerance* is typically used to describe disk subsystems, but it can also apply to other parts of the system or the entire system.

Redundant Arrays of Inexpensive Disks (RAID)

Fault-tolerant disk systems are standardized and categorized in seven levels known as Redundant Arrays of Inexpensive Disks (RAID) level 0 through level 6. The RAID levels are somewhat loosely defined, and details on performance or disk use vary from one configuration to the next. Depending on the implementation, definitions may overlap or be combined. Each level offers various mixes of performance, reliability, and cost. Windows NT supports RAID levels 0 through 5.

The major difference between RAID and earlier, more expensive large-disk technologies (also called Single Large Expensive Disks, or SLED) is that RAID combines multiple disks with lower individual reliability ratings to reduce the total cost of storage. The lower reliability of each disk is offset by the redundancy.

RAID Level 0. This strategy is commonly known as disk striping without parity and uses a disk file system called a stripe set. Data is divided into blocks and spread in fixed order among all of the disks in the array. RAID Level 0 may enhance disk performance, but it does not provide redundancy. For that reason, it is not considered to be a true RAID level, nor does it qualify as fault tolerant.

RAID Level 1. This strategy is commonly known as disk mirroring, disk duplexing, or disk shadowing. It provides an identical twin for a selected disk; all data written to the primary disk is written to the twin or mirrored disk. This strategy provides the best performance when a member fails, but it is also the most expensive to implement due to the two-for-one disk space requirements.

RAID Level 2. This strategy is commonly known as disk striping with error correcting code (ECC). This method achieves redundancy with ECC. It employs a disk striping strategy that breaks a file into bytes and spreads it across multiple disks. At the same time, the ECC data is spread across multiple check disks. In the event of data loss, the ECC information can be used to reconstruct the lost data.

RAID Level 3. This strategy is the first in the series of levels that provide disk striping with parity. It employs the same striping method as Level 2 but replaces the ECC method

with a parity-checking scheme that requires only one disk on which to store the parity information.

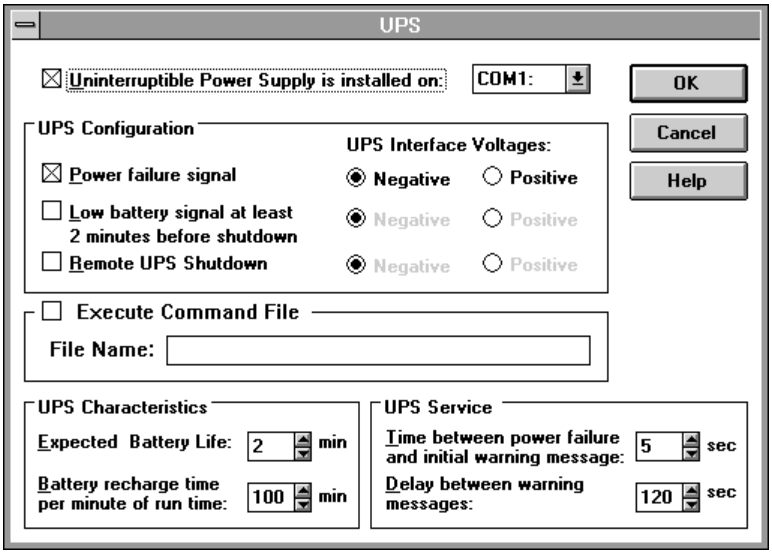
RAID Level 4. This strategy is similar to Level 3 in that it stores the parity information on a separate check disk. Where it differs is in the striping. This method stripes data in much larger chunks.

RAID Level 5. This has become the most popular strategy for recent fault-tolerance designs. Like Level 4, it stripes data in big chunks. Unlike Level 4, it does not use a separate check disk for parity information. Rather it stripes the parity across the disks as well. The data and parity information are arranged on the disk array so that the two are always on different disks.

RAID Level 6. This is essentially Level 5 with the addition of redundant hardware such as disk controllers and power supplies.

Uninterruptible Power Supplies

An uninterruptible power supply (UPS) provides power when the local power fails. It is usually rated to provide a specific amount of power for a specific period of time. This power comes from batteries that are kept charged while main power is available. The main power is converted from AC voltage to the DC voltage used to charge the battery. When needed, the DC power is converted to an AC voltage compatible with the computer power supply. Usually, all that is needed from a UPS is time to shut down the system in an orderly fashion by terminating processes and closing sessions.



Configuring UPS under Windows NT

Many UPS devices offer the ability to interface with operating systems, enabling the operating system to notify users automatically of the pending shutdown process or provide notification that the power has been restored and a shutdown is no longer necessary. Windows NT provides an interface for these types of UPS devices through a serial port connection. Communication is handled in much the same way as hardware

handshaking is handled on a normal RS-232C connection. Hardware signals are translated into power-state messages which are then interpreted by Windows NT software.

For example, during a power failure, the UPS service for Windows NT immediately pauses the Server service for Windows NT to prevent any new connections and sends a message to notify users of the power failure. The UPS service then waits a specified interval of time before notifying users to terminate their sessions. If power is restored during the interval, another message is sent to inform users that power has been restored and normal operations have resumed.

Distributed Computing - Directory Services

Overview

Let's be absolutely clear about this: the goal is distributed computing. Directory Services is one key piece of that goal, but it is only a means to an end and not the end in itself. Distributed computing can be thought of as a seamless extension of the desktop where information is readily available at your fingertips. It goes far beyond the file and print sharing services that are common today.

A complete distributed computing infrastructure will be constructed from a variety of technologies and components. One essential component is a microkernel-based operating system that will serve as the foundation. A microkernel-based operating system accommodates the heterogeneous nature of today's networks and provides the flexibility and the building block for tomorrow's networks. Another essential component is directory services. However, to make distributed computing truly seamless, directory services will need to be built on and tightly integrated with the microkernel-based operating system. It needs to be something more than adding X.500 on top of Unix. Both the microkernel-based operating system and the directory services will also need to support a widely accepted and well integrated set of APIs for application development. In the case of directory services, strategic isolation layers of software will also need to be defined and accepted.

Because directory services are an important component of a distributed computing infrastructure, it is important to understand the continuing evolution of directory services. As a recent report from The Burton Group points out, "fully functional directory services are an important element of the larger distributed computing picture, and their arrival will be the harbinger of a fundamental change in the network computing model as we have known it. Along with distributed and object-oriented file systems, security services, and object-oriented development frameworks, directory services will ultimately enable a distributed computing environment that will change the way people use networks." [2]

While the directory services provided by today's network operating systems have increased in functionality over the last several years, they are still used primarily in single-purpose administrative roles, such as electronic mail, multiuser accounting applications, and workgroup applications. From a functional point of view, these directory services are designed to make the network operating system environment more manageable.

[2] "Directory Services Strategic Overview: The Advent of Directory-Enabled Computing." The Burton Group, June, 1995.

It is important to realize, however, that the vision of distributed computing cannot be fulfilled with directory services that have such limited scope. For example, next-generation directory services will not be deployed as subsets of other applications and services; they will be implemented as an integral part of the operating systems that serve as the foundation for distributed computing. Instead of being exposed as a separate database, as current-generation directories are, next-generation directories will be integrated with the network file system. The operating system, the network file system, and the directory services will be one single, unified package rather than disparate pieces loosely glued to each other.

Next-generation directories will be capable of containing all of the information on the network—in the form of *objects*—and not just the user profiles, access control lists, and other administrative information found in today's directories. As a result, users will be able to use the query capabilities of the directory to search for more than the name of a given server or printer. They will be able to search for any and all types of information on the network and, hence, also contained in the directory. The information will be much more accessible, so that users can focus on their work, rather than on how the network works.

Next-generation directories must also provide significant levels of interoperability with the existing administrative directories deployed as part of other applications and operating systems. This seamless interoperability will allow system administrators to unify the implementation, access, and management of all network resources, including current-generation directory services, within the next-generation directory itself.

Domains and Trust Relationships

Definition. The Windows NT Domain should not be confused with a Unix Domain. In Unix, a domain refers to how a particular computer is named on a TCP/IP internetwork. In Windows NT, a domain is a group of servers running Windows NT Server that share common security policy and user account databases. Therefore, the Windows NT Domain is the basic unit of security and centralized administration for Windows NT, and the servers in the domain, in some ways, can be viewed as a single system.

One computer running Windows NT Server acts as the Primary Domain Controller (PDC), which maintains the centralized security databases for the domain. Other computers running Windows NT Server in the domain function as backup domain controllers and can authenticate logon requests. Users of a Windows NT Domain are authenticated by the PDC or by a backup domain controller. Domains can also contain computers running Windows NT Server that are not domain controllers, server computers that are not running Windows NT Server, and client computers such as those running Windows NT Workstation, Windows® for Workgroups, and MS-DOS.

The other key concept in Windows NT Domains is the Trust Relationship. A trust relationship is a link between two domains that enables a user with an account in one domain to have access to resources in another domain. When you establish a trust relationship between domains, one domain (the trusting domain) trusts the other domain (the trusted domain). Trust relationships are unidirectional. Bi-directional trusts are created with two unidirectional ones. In addition, trust relationships are not transitive. For example, if Domain A trusts Domain B, and Domain B trusts Domain C, Domain A will not trust Domain C by default. If Domain A needs to trust Domain C, a separate trust relationship must be set up between the two domains.

Domains and trust relationships are the key components in the Windows NT Directory Services. By combining domains and trusts, you are able to strike a balance between access, control, and administration for your particular network requirements. There are four common ways of combining domains and trust relationships into what are known as domain models. They are: Single Domain, Master Domain, Multiple Master Domain, and Multiple Trust.

Single Domain. In the single domain model, there is only one domain. Because there are no other domains, there are no trust relationships to administer. This model is best for organizations with fewer than 40,000 users in which establishing trust relationships among departments is not an issue. In an organization with multiple departments where there is no need to share information among them, the best configuration is often multiple single domains.

Master Domain. In an organization with fewer than 40,000 users in which establishing trust relationships among departments is an issue, the master domain model is a suitable option. In this model, one domain, the master domain, is trusted by all non-master subdomains, but does not trust any of them. The master domain contains the user accounts database and provides authentication services to the trusting subdomains. This model offers the benefits of both central administration and multiple domains.

Multiple Master Domain. In this model, there is more than one master domain. All of the master domains trust each other, and all are trusted by the non-master subdomains, but none of the master domains trusts any of the subdomains. This model works best when computer resources are grouped in some logical fashion, such as by department or by location. Each master domain can support as many as 40,000 users, so this model works well in large organizations. And because all the master domains trust each other, user accounts need only exist in one of them.

Multiple Trust. In the multiple trust model, all domains trust each other. There are no master domains. This model is sometimes referred to as the complete trust model, and is the simplest to understand. As with the multiple master domain model, the multiple trust model is scalable as the organization grows, and because each domain has full control over its own user accounts, it can work well for a company without a centralized information services (IS) department.

Name Resolution

A computer on a network usually has both a name and an address. Take TCP/IP, for example. It requires an IP address and computer name, which are unique identifiers for the specific computer on the network. Computers use the IP addresses to identify each other, but people usually find it easier to work with the computer names. Therefore, a mechanism must be available to convert computer names into their corresponding IP addresses. This mechanism is known as name resolution.

A computer running Windows NT can use one or more of the following methods to ensure accurate name resolution in TCP/IP internetworks:

- **Windows Internet Name Service (WINS)** is a NetBIOS over TCP/IP (NBT) mode of operation as defined in RFC 1001/1002 as p-node.
- **Broadcast Name Resolution** is a NetBIOS over TCP/IP (NBT) mode of operation as defined in RFC 1001/1002 as b-node.

- **Domain Name System (DNS)** is defined in RFCs 1034 and 1035.
- **A HOSTS file** is a flat file used to specify the DNS computer name and IP address mapping.
- **An LMHOSTS file** is a flat file used to specify the NetBIOS computer name and IP address mapping.

Dynamic Host Configuration Protocol (DHCP)

To fully understand the power of WINS, it is necessary to first know something about the Dynamic Host Configuration Protocol (DHCP). DHCP relieves the administrative burden associated with assigning and maintaining IP addresses. It offers dynamic configuration of IP addresses for computers. DHCP provides safe, reliable, and simple TCP/IP network configuration, ensures that address conflicts do not occur, and helps conserve the use of IP addresses through centralized management of address allocation. DHCP services for Windows NT are implemented under RFCs 1533, 1534, 1541, and 1542.

DHCP uses a client/server model and is based on leases for IP addresses. The system administrator controls how IP addresses are assigned by specifying address allocation ranges and lease durations. It is also possible to have static IP addresses, which are addresses with leases that do not expire. During system startup, a DHCP client computer sends a “discover” message that is broadcast to the local network and might be relayed to all DHCP servers on the private internetwork. Each DHCP server that receives the discover message responds with an offer message containing an IP address and valid configuration information for the client that sent the request.

The DHCP client collects the configuration offerings from the servers, chooses one of the configurations, and sends a request message to the DHCP server for the selected configuration. The selected DHCP server sends an acknowledgment message to the client. The DHCP acknowledgment message contains the IP address originally sent with the offer message, a valid lease for that address, and the appropriate TCP/IP configuration parameters for use by the client. After the client receives the acknowledgment, it enters a bound state and can now participate on the TCP/IP network and complete its system startup.

Client computers save the received address for use during subsequent system startups. By default, the client attempts to renew its lease with the DHCP server when 50% of the lease time has expired. If the current IP address lease cannot be renewed, a new IP address is assigned.

As an example of how maintenance tasks are made easy with DHCP, consider the case in which a computer is moved from one subnet to another. The IP address is released automatically for the DHCP client computer when it is removed from the first subnet, and a new address is automatically assigned to it when it is attached to the new subnet. Neither the user nor the system administrator needs to intervene to update the configuration information.

Windows Internet Name Service (WINS)

The Windows Internet Name Service provides a dynamic database for registering and querying name-to-IP address mappings in a routed network environment. When a DHCP client moves from one subnet to another, the IP address change is automatically updated

in the WINS database. As with DHCP, no intervention is required of either the user or the system administrator to update the configuration information.

WINS consists of two components: the WINS server, which handles name queries and registrations, and the client software, which queries for computer name resolution. WINS servers support multiple replication partners in order to provide increased service availability, better fault tolerance, and load balancing. Each WINS server must be configured with at least one other WINS server as its replication partner. These partners can be configured to be either pull partners or push partners depending on how replications are to be propagated. WINS can also provide name resolution service to certain non-WINS computers through proxies, which are WINS-enabled computers that act as intermediaries between the WINS server and the non-WINS clients.

Domain Name System (DNS)

The Domain Name System is a distributed database that provides a hierarchical naming system for identifying hosts on the Internet. A Unix Domain is synonymous with a DNS Domain. DNS was developed to solve the problems that arose when the number of hosts on the Internet grew dramatically in the early 1980's. Although DNS might seem similar to WINS, there is a major difference: DNS requires static configuration for computer name-to-IP address mapping, while WINS is dynamic and requires far less administration.

The DNS database is a tree structure called the domain name space, where each domain (node in the tree structure) is named and can contain subdomains. The domain name identifies the domain's position relative to its parent domain in the database. A period (.)separates each part of the name. For example, tsunami.microsoft.com could be the name of a computer owned by Microsoft. The root of the DNS database is managed by the Internet Network Information Center. The top-level domains were assigned organizationally and by country. These domain names follow the ISO 3166 standard.

DNS uses a client/server model, where the DNS servers contain information about a portion of the DNS database and make this information available to clients, called resolvers, that query the name server across the network. DNS name servers are programs that store information about parts of the domain name space called zones. The domain administrator sets up name servers that contain database files with all the resource records describing all hosts in their zones. DNS resolvers are clients that are trying to use name servers to gain information about the domain name space. Windows NT includes all the resolver functionality necessary for using DNS on the Internet.

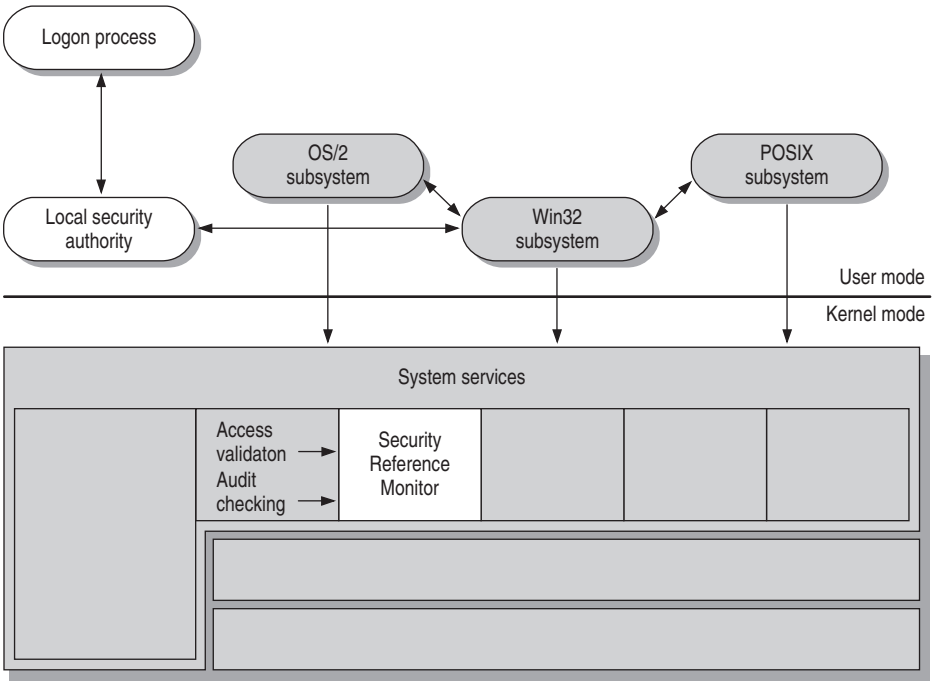
Security

Overview

The security model for Windows NT is designed to meet both national and international security criteria. In the United States it is the C2-level criteria as defined by the U.S. Department of Defense *Trusted Computer System Evaluation Criteria* document (DOD 5200.28-STD - December 1985). This document is commonly referred to as the *Orange Book*. The C2-level requires what is known as Discretionary Access Control and is one in a range of seven levels of security specified by the DOD. Some of the most important requirements for C2 are:

- **Identification and Authentication.** Each user must identify herself or himself by typing a unique logon name and password before being allowed access to the system. The system must be able to use this unique identification to track the user's activities.
- **Discretionary Access Control.** The owner of a resource (such as a file) must be able to control access to the resource.
- **Object Reuse.** The operating system must protect objects so that they are not randomly reused by other processes. For example, the system protects memory so that its contents cannot be read after it is freed by a process. In addition, when a file is deleted, users must not be able to access the file's data.
- **Audit.** System administrators must be able to audit security-related events. Access to this audit data must be limited to authorized administrators.
- **System Architecture.** The system must protect itself from external interference or tampering, such as modifications to the running system or system files stored on disk.

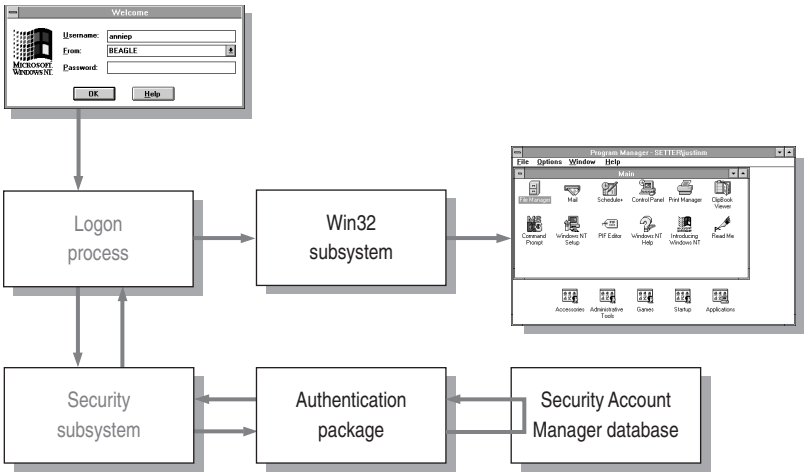
In the European community, the rough equivalent of the *Trusted Computer System Evaluation Criteria (TCSEC)* document is the *Information Technology Security Evaluation Criteria (ITSEC)* document. ITSEC is the product of the Common Criteria Editorial Board, an international standards body made up of representatives from various countries including France, Germany, the Netherlands, the United Kingdom, and the US. Because of the differences in criteria, there is no direct, one-to-one rating between the TCSEC document and the ITSEC document. A TCSEC rating of C2 translates roughly into an ITSEC rating of F-C2,E2. The higher TCSEC rating of B1 translates roughly into an ITSEC rating of F-B1,E3. Windows NT is currently being evaluated for an ITSEC rating of F-C2,E3, which is a mixture of the TCSEC B1 and C2 ratings.



Windows NT Security Model

The security model for Windows NT is made up of the following components:

- **Logon Processes,** which accept logon requests from users. These include the initial interactive logon, which displays the logon dialog box to the user, and remote logon processes, which allow access to server processes within Windows NT by remote users.
- **Local Security Authority,** which ensures that the user has permission to access the system. This component is the center of the security subsystem. It generates access tokens, manages the local security policy, and provides interactive user authentication services. The Local Security Authority also controls audit policy and logs the audit messages generated by the Security Reference Monitor.
- **Security Account Manager (SAM),** which maintains the user accounts database. This database contains information for all user and group accounts. SAM provides user validation services, which are used by the Local Security Authority.
- **Security Reference Monitor,** which checks to see if the user has permission to access an object and perform whatever action the user is attempting. This component enforces the access validation and audit generation policy defined by the Local Security Authority. It provides services to both kernel and user mode to ensure that the users and processes attempting access to an object have the necessary permissions. This component also generates audit messages when appropriate.



Windows NT Logon Security Process

Together, these components are known as the security subsystem. This protected subsystem is an integral subsystem rather than an environmental subsystem because it affects the entire Windows NT operating system.

Users and Groups

Under Windows NT, any person who needs access to resources on the network must have a valid user account on a domain that allows or is allowed that access. The Windows NT user account contains the following information:

- **Username.** The unique name the user types when logging on.
- **Password.** The user's secret password.
- **Full Name.** The user's full name.
- **Logon Hours.** The hours during which the user is allowed to gain access to the resources on the network.
- **Logon Workstations.** The computer names of the workstations that the user is allowed to work from.
- **Expiration Date.** A future date when the account automatically becomes disabled.
- **Home Directory.** A directory on the server that is private to the user; the user controls access to this directory.
- **Logon Script.** A batch or executable file that runs automatically when the user logs on.
- **Profile.** A file containing a record of the user's Desktop environment, such as program groups, network connections, and screen colors, that follows the user from one workstation to another. The system administrator can also set up standard profiles that users are not allowed to modify.
- **Account Type.** For most, if not all, user accounts, the type will be global.

Like Unix, Windows NT supports the concept of groups. With groups, you can group together users who have similar jobs and resource needs. Groups make granting rights and resource permissions easier; a single action of giving a right or permission to a group gives that right or permission to all present and future members of that group.

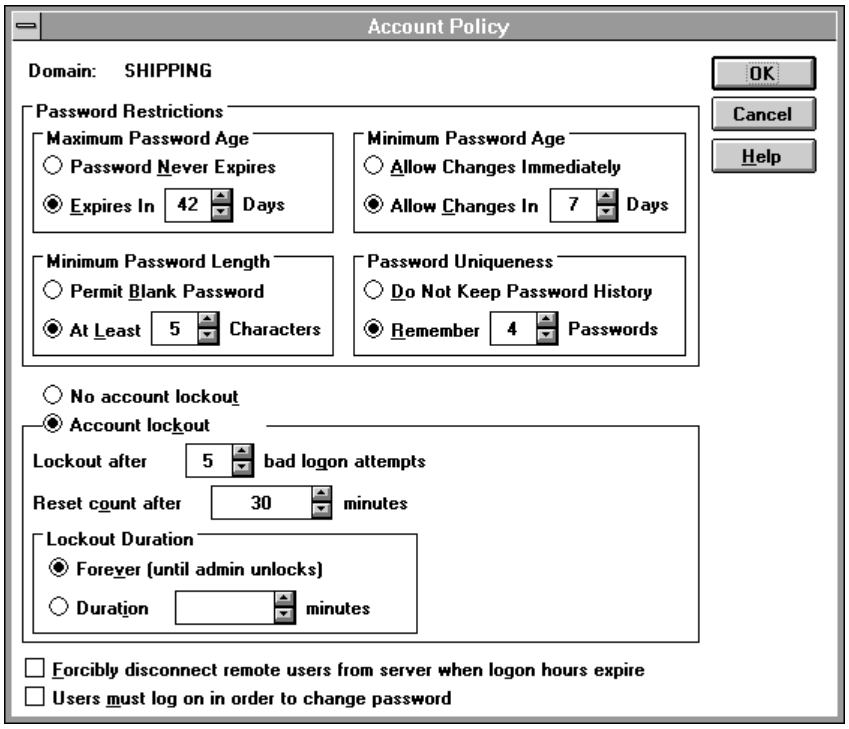
Even though they are conceptually similar, groups within Windows NT are inherently more powerful than groups within Unix. Windows NT provides a set of built-in groups that gives members rights and abilities to perform various tasks, such as backing up the computers or administering the network printers. Examples of these built-in groups are: Administrators, Backup Operators, Print Operators, Power Users, Users, and Guests. It is also possible for the system administrator to define new types of groups, although the built-in groups cover most of the standard combinations of rights and permissions that you would expect to find. User accounts can belong to more than one group at the same time and will share the combined rights and permissions of all of them.

Passwords

Windows NT provides features that enable a system administrator to create a very robust account policy. The password policy can set the following limits on user passwords:

- **Maximum Password Age.** Specifies how long a user can use a password without changing it. By default, this setting is 42 days. It can be set to any value from 1 to 999 days, or it can be set to never expire.
- **Minimum Password Age.** Specifies how long a user must wait after changing a password before the user can change it again. By default, users can change their passwords immediately. This setting can be changed to any value between 1 and 999 days.

- **Minimum Password Length.** Specifies how many characters a password must contain. By default, Windows NT permits blank passwords. Many system administrators, however, change this option. A recommended minimum is 6 characters, but it can go as high as 14 characters.
- **Password Uniqueness.** Specifies how many different passwords a user must use before being allowed to reuse one. With the default setting, a user can immediately reuse an expired password. This setting can be changed to require users to create from 1 to 24 unique passwords before reusing one.



Setting Account Policy

Windows NT has no equivalent to the Unix **/etc/passwd** file. With Windows NT, passwords are not hashed and stored in a flat file; they are instead integrated into the security model. No direct access to passwords, hashed or otherwise, is provided under Windows NT. The system administrator can only reset a user's password.

Windows NT also provides an account lockout feature. When this feature is enabled, a user account becomes locked if there are a number of incorrect attempts to log on to that account within a specified amount of time. Locked accounts cannot log on. A locked account remains locked until an administrator unlocks it, or until a specified amount of time passes, depending on how the account lockout feature has been configured. By default, account lockout is disabled.

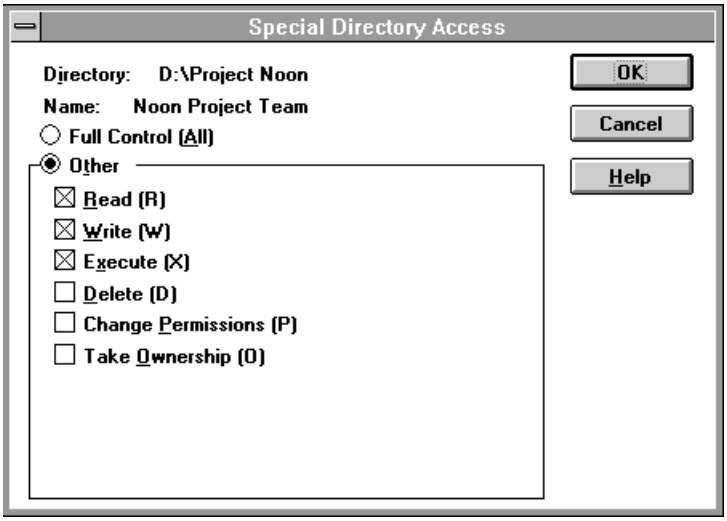
File Access (NTFS)

Windows NT supports multiple file systems. The two most commonly used are: the Windows NT file system (NTFS) and the file allocation table (FAT) file system. Permissions work differently on NTFS volumes than they do on FAT volumes. Security is

much more powerful on NTFS volumes, and, for that reason, NTFS is the file system of choice for Windows NT.

Windows NT offers a set of standard permissions for files and directories in NTFS volumes. These standard permissions offer useful combinations of specific types of access, which are called individual permissions. Individual permissions are somewhat analogous to Unix permissions, and consist of: Read (R), Write (W), Execute (X), Delete (D), Change Permissions (P), and Take Ownership (O). These permissions can be specified either directly as individual permissions or indirectly as standard permissions. Examples of standard permissions are: Read (RX), Change (RWXD), No Access (None), and Full Control (All). As with the built-in groups, it is possible for the system administrator to create new types of standard permissions, although most of the combinations that you would expect to find are covered in the existing ones.

Unix supports three sets of file and directory permissions: owner, group, and world. This is the familiar -rwxrwxrwx that shows up in the output from the Unix **ls -al** command. With Windows NT, permissions can be granted to either individual users or to groups. The big difference between Windows NT and Unix is that, with Windows NT, multiple sets of permissions can be granted to multiple combinations of groups and/or individual users. You are not limited to just three sets.



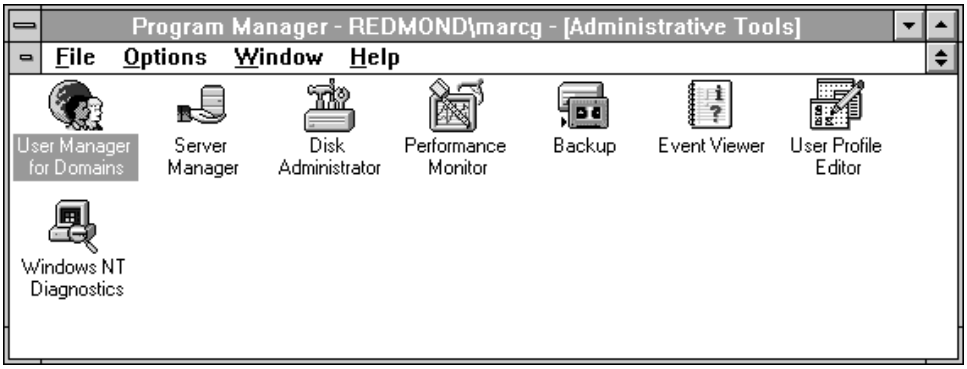
Setting Special Directory Permissions or Taking Ownership

Every file and directory on an NTFS volume has an owner. The owner controls how permissions are set on the file or directory, and can grant permissions to others. File ownership provides a way for users to keep private files private. This is another area where Windows NT differs significantly from Unix. The system administrator can take ownership of any file on the system, but the system administrator cannot then transfer the ownership to others, as can be done with Unix. Therefore, if an administrator wrongly takes ownership of someone's files, that administrator cannot subsequently transfer ownership back to the original owner, and the original owner can easily find out who the new owner is.

System Administration

Overview

System Administration differs significantly between Windows NT and Unix. Two factors contribute heavily to this difference. First, Windows NT is based on the client/server model rather than the host-based terminal model. This is inherently a more decentralized model, which dramatically alters many of the tasks usually associated with system administration in a host-based terminal environment such as Unix. Second, Windows NT is fully integrated with its Windows-based, graphical user interface (GUI). Virtually everything a system administrator does on the machine is GUI-based. This is in stark contrast to the command-line user interface (CLUI) that is common for Unix. There are X-based, system administration tools available on many of the Unix implementations. However, these GUI-based tools were added as an afterthought, are not well integrated, and, in general, are specific to a particular flavor of Unix.



Windows NT Administrative Tools

System administration is perhaps the most difficult area in which to make the paradigm shift. It is very hard to move from the "raw power" of the command-line interface into the safer, standardized realm of the graphical user interface. There is a feeling of loss of control, and it is at this point that it becomes critical to remember that client/server is a different world, requiring different tools and strategies.

Account Management

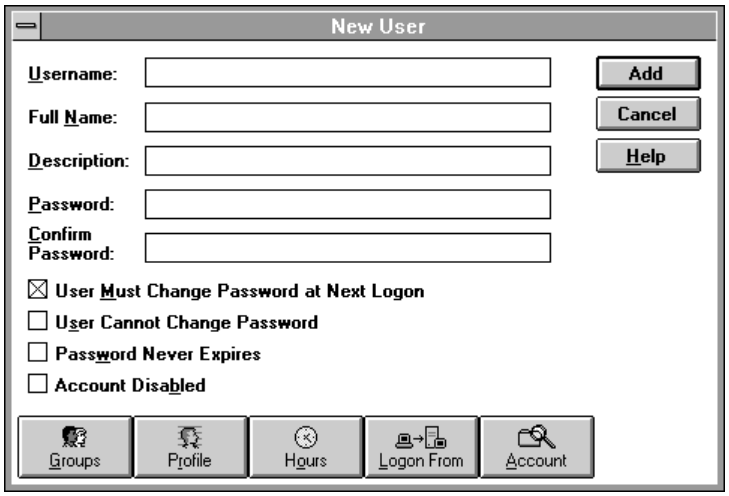
User Accounts. Through the magic of domains and trust relationships, users have a single logon to the network, which gives them access to their resources from any available workstation. This not only reduces the administrative burden on the system administrator, but it also directly benefits the users in that they have only one logon name and one associated password to remember. In addition, their desktop environments follow them as they move from workstation to workstation.

In Windows NT, user accounts are established and maintained with a Windows-based tool known as User Manager for Domains. It is a very far cry from the Unix method in which you typically use **vi** to manually edit the **/etc/passwd** and **/etc/groups** files. You can use User Manager for Domains to:

- **Maintain User Accounts.** You can add, modify, rename, disable, and delete users. You can establish and maintain password policy, as well as reset forgotten

passwords. You can set up logon hours, logon workstations, and account expiration dates.

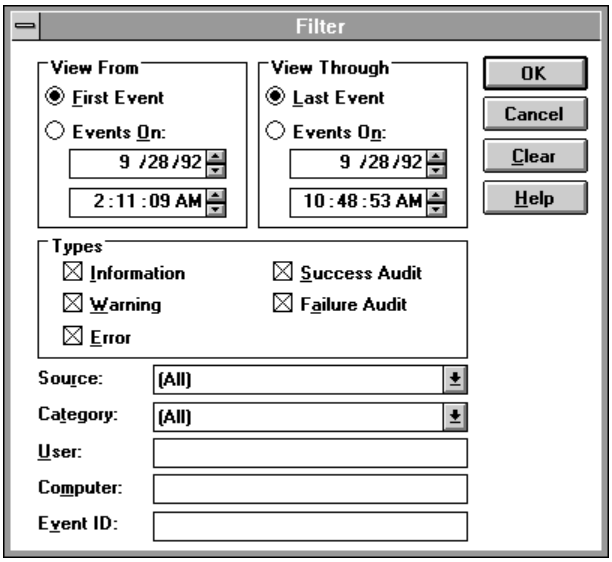
- **Maintain User Environment Profile.** You can set up or modify the path to the user's profile. You can establish a logon script for the user, and you can specify the user's home directory. This is similar to setting up **.profile** and **.kshrc** files, and establishing **$HOME** environment variables for users under Unix.
- **Maintain Groups.** You can add, modify, or delete groups, and you can add or remove users from groups.
- **Maintain the Security Policies.** You can manage the Account Policy, which controls the way passwords must be used by all user accounts and whether user accounts are locked out after so many bad logon attempts. You can manage the User Rights Policy, which controls the rights assigned to groups and user accounts, and you can manage the Audit Policy, which defines the security events that will be logged.



User Manager for Domains — New User Dialog Box

User Profile policy is established and maintained with another Windows-based tool: User Profile Editor (although it can also be accessed through User Manager for Domains). User Profile Editor is somewhat analogous to using **vi** to modify the **/etc/profile** file on Unix. However, User Profile Editor gives you complete control over all user profiles on the system, not just the default, global one.

Accounting Information. Accounting information is also less of an issue with client/server based computing than it is with host-based terminal computing. With auditing enabled, you can use the filter function of Event Viewer to see when a particular user last logged on to the domain. This is how you identify inactive user accounts. With another tool, Server Manager, you can see who is connected to a server, how long they have been connected, and what resources they have open. You can also view this information in different formats. You can see which resources are currently being used, or, if you are interested in just one resource, you can view information specific to it, such as who is connected and for how long.



Filtering with Event Viewer

Server Management

Remote Administration. The client/server model, coupled with domains, trust relationships, and single network logon, makes it possible for system administration to take place anywhere on the corporate network. Remote dial-in access is provided by a built-in feature of Windows NT known as Remote Access Services (RAS). With RAS, system administration can happen from virtually any location that has access to a telephone.

RAS for Windows NT is based on a client/server architecture, in which a remote RAS client connects to a local RAS server. TCP/IP, IPX, and NetBEUI are all supported, which means you can integrate RAS for Windows NT into existing Microsoft, Unix, or NetWare networks using the PPP remote access standard.

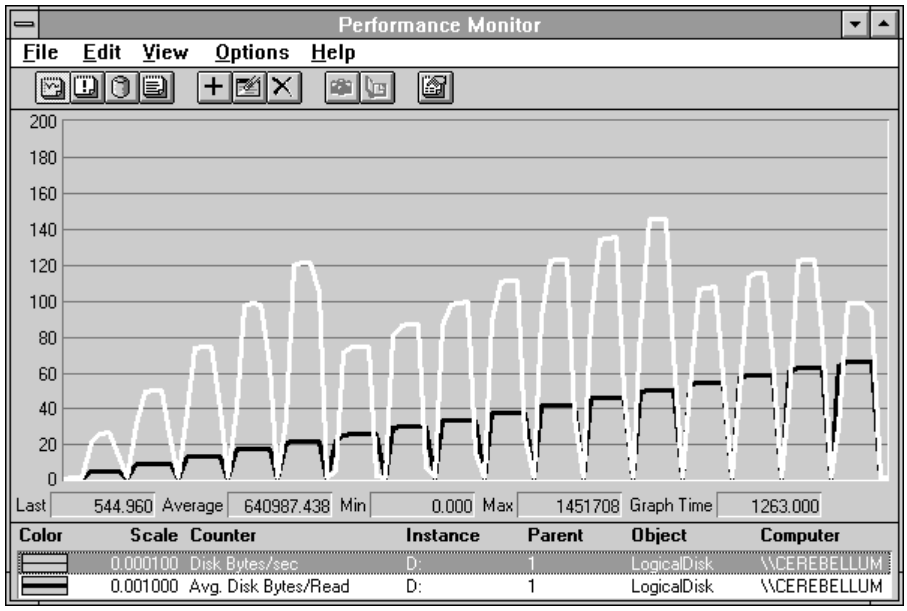
Further interoperability is possible. Non-Microsoft PPP clients using TCP/IP, IPX, or NetBEUI can also access a Windows NT-based RAS server, and Windows NT-based RAS clients can connect to existing SLIP-based remote access servers (typically Unix servers). Microsoft RAS protocol is a proprietary remote access protocol supporting the NetBIOS standard. It has been largely superseded by the newer PPP remote access standard.

Two things are needed to make a RAS server: 1) a computer running Windows NT and RAS, and 2) either a multiport adapter or modem(s). When a remote RAS client connects to the RAS server, the RAS server checks to see if the RAS client is using a preassigned IP address. If not, the RAS server automatically assigns an IP address to the RAS client from the static range of addresses owned by the RAS server.

Once the connection has been made, the RAS client becomes a full-fledged node on the corporate network, not just a dumb terminal dialing in. The remote system administrator can then use the same Windows-based tools as a local system administrator to access resources such as files, printers, electronic mail, and databases.

Performance Monitoring. A major design goal of Windows NT was to eliminate the many obtuse parameters that characterized earlier systems. Adaptive algorithms were incorporated in the design so that correct values are determined by the system as it runs. The result is that Windows NT has fundamentally changed how computers will be managed in the future. The task of optimizing the system is not the art of manually adjusting many conflicting parameters. It is a process of determining what hardware resource is experiencing the greatest demand, and then adjusting the operation to relieve that demand.

Windows NT includes a Windows-based tool for tracking computer performance called Performance Monitor, which is similar to tools on Unix, such as **top** and GlancePlus from Hewlett-Packard. Performance Monitor is based on a series of counters that track such things as the number of processes waiting for disk time, the number of network packets transmitted per second, and the percentage of processor utilization.



Performance Monitor — Chart View

Performance Monitor displays information graphically (through Charts) or as text (through Reports). Data can be displayed in real-time or collected in logs for later display as a graph or report. You can also configure Performance Monitor to generate alert logs. Alert log entries are posted every time a counter exceeds or falls below a user-specified value. They can also be used to trigger Performance Monitor to generate network messages or run programs based on the value of the log entry.

In addition to Performance Monitor, other tools are available for Windows NT. The *Windows NT Resource Kit* is a four-volume set of books with accompanying software. Volume 4, *Optimizing Windows NT*, provides over 600 pages of practical advice on performance tuning for Windows NT, along with additional software tools that compliment the features of Performance Monitor.

File Backup/Restore. Windows NT supplies a Windows-based, Backup tool, which is similar to the X-based tools found on some versions of Unix. Its salient features are:

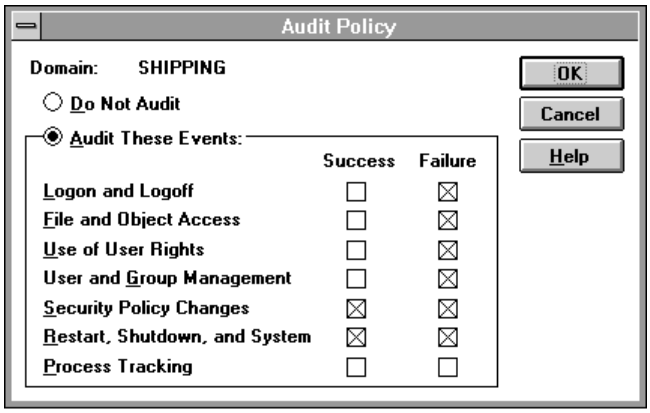
- Back up and restore both local and remote files on an NTFS or FAT file system from a local computer with an attached tape drive.
- Select files for backing up or restoring by volume, directory, or individual filename and view detailed file information, such as size or modification date.
- Select an optional verification pass to ensure reliable backups or restores.
- Perform any of the following common backup operations: Normal, Copy, Incremental, Differential, and Daily.
- Place multiple backup sets on a tape, and span multiple tapes with both backup sets and files, since there is no restriction on file size.
- Automate the backup with a batch file and the scheduler service.
- View a full catalog of backup sets, directories, and individual files.
- Control a destination drive and directory for the restore.
- Save log information on tape operations to a file.

The Backup tool provides a Backup Status dialog box that shows the active status of the tape operation. You can use another Windows-based tool, Event Viewer, to view the backup history in the system event logs.

Disk Allocation. The primary tool for day-to-day administering of files and directories is File Manager. It is a powerful, Windows-based tool that, among other things, enables you to selectively view file listings sorted by name, type, size, or date and time of last modification. Windows NT also includes a command language that is a superset of the MS-DOS batch commands. It provides a scripting capability that is similar to the Unix shell scripts. With it, you can do finer granularity sorts on files and directories and see additional information, such as date and time of creation or last access.

Currently, Windows NT does not support the concept of disk quotas per se. It is possible to do something similar with judicious use of disk partitions and shared directories, and there are third-party products, such as Quota Manager from New Technology Partners, that provide traditional disk quota features.

Auditing. Windows NT can record a range of event types, from a system-wide event such as a user logging on, to an attempt by a particular user to access a specific file. Both successful and unsuccessful attempts to perform an action can be recorded.



Setting System-Wide Audit Policy

System-wide audit policy is established and maintained with User Manager for Domains. Individual object access auditing is controlled with either File Manager, for files and directories, or Print Manager, for printers. The audit log entries are examined and manipulated with Event Viewer.

Windows NT records events in three kinds of logs:

- **The System Log** records events logged by the system components of Windows NT. For example, the failure of a driver or other system component to load during startup is recorded in the system log.
- **The Application Log** records events logged by applications. For example, a database program might record a file error in the application log.
- **The Security Log** records security events. This helps track changes to the security system and identify any possible breaches to security. For example, attempts to log on to the system can be recorded in the security log, depending on the Audit settings in User Manager for Domains.

After you select a log for display in Event Viewer, you can view, sort, filter, and search for details about events. You can also archive logs in various file formats.

The following types of events can be audited:

- **Logon and Logoff.** Entries record successful and unsuccessful logon attempts, logoff attempts, and the making or breaking of a network connection to a server.
- **File and Object Access.** Entries record attempted access of directories or files that are set for auditing in File Manager, or system printers that are set for auditing in Print Manger.
- **Use of User Rights.** Entries record successful uses of user rights and unsuccessful attempts to use rights not assigned to users.
- **User and Group Management.** Entries record changes to groups and user accounts, including changes to passwords.
- **Security Policy Changes.** Entries record changes made to the User Rights, Audit, or Trust Relationship policies.

- **Restart, Shutdown, and System.** Entries record shutdowns and restarts of the computer, the filling up of the audit log, and the discarding of audit entries if the audit log is already full and rollover has been enabled.
- **Process Tracking.** Entries record starts and stops of processes on the computer. These events provide detailed tracking information and are enabled under very rare circumstances.

Background Job Scheduling. At the time of this writing, the Microsoft corporate data center is using Windows NT to run over 1,000 batch jobs daily on 340 servers worldwide; this includes backing up more than a terabyte of data. The built-in job scheduling features of Windows NT are comparable to those that come with Unix, such as **cron** and the **at** command. Windows NT also has a built-in **at** command, and the *Windows NT Resource Kit* has a graphical utility known as the Command Scheduler. These are very basic tools, as are their Unix counterparts. For more sophisticated tools, the kind of tools you need for data center operations, there are third-party products available for Windows NT. (Some of them have even been ported from Unix). Some of these packages are listed in the following table.

Product	Vendor
Argent Queue Manager	Argent Software, Inc.
AshWin for Windows NT	Creative Interaction Technologies, Inc.
CA-Unicenter™ for Windows NT	Computer Associates International, Inc.
POLYCENTER	Digital Equipment Corporation

Network Management

Windows NT combines a sophisticated computer operating system with a fully integrated network operating system (NOS), and the synergy created by this marriage provides the system administrator with an unparalleled wealth of resources for managing the corporate network. We have already covered how DHCP and WINS work together to automate the tasks of IP address administration and name resolution on the network. Additional administrative tools include: Server Manager, Performance Monitor, Event Viewer, Control Panel, Network Client Administrator, and License Manager. All of these enable the system administrator to monitor and control various aspects of the corporate network. The *Windows NT Resource Kit* supplies even more tools to compliment the list, such as Net Viewer, Domain Monitor, Browser Monitor, and Process Viewer.

Desktop Management

Microsoft Systems Management Server. Installing and maintaining software is a major cost to corporations with distributed networks. Often, the system administrator must install, upgrade, and configure each computer manually. For a large corporation with locations across a wide geographical area, these installation and support costs may increase exponentially. In fact, most of the cost-of-ownership for a corporate computer system comes not from the initial purchase price of the software, but from the software

installation, support, and maintenance costs. You can find more information about this subject in a recent report from the Gartner Group. [3]

If a corporation already has a distributed network in place, it makes sense to take advantage of its wide-area connectivity for managing software for the entire corporation. But, before software can be installed over the network, you must know where it is going. Before software can be maintained, you must know where it is. You need to know what computers are on the network. And, after you find out what computers you have, you need to know information about the computers so you can install or maintain your software correctly. You need to know what hardware they have, what software is already installed, and how the computers are configured. In short, you need an inventory. If you have many computers, you also need a logical way to group them so you can recognize them more easily, such as by location or configuration. You need a structured way to look at the entire corporate network.

Enter Systems Management Server. Microsoft Systems Management Server provides system administrators with a method for centrally managing software and hardware for their corporate networks. It is based on client/server architecture. The server runs on a Windows NT Server-based machine; the clients need not. Systems Management Server is an easy-to-use, integrated system that:

- Maintains an inventory of hardware, software, and configuration of computers across a corporate network.
- Distributes, installs, and updates software and files.
- Manages network applications (applications run over the network from servers).
- Provides integrated support utilities, including a network monitor, that enable you to view diagnostic information for remote clients and take direct control of them if need be.

Systems Management Server maintains a database containing system information and inventory, carries out distribution and installation jobs, monitors the progress of these jobs, and alerts you to important system events. With Systems Management Server you can distribute and install software on clients and servers across your corporate network, set up network applications, automatically collect and maintain hardware and software inventory, provide direct support to users, and monitor your network.

User Interface and Environment

Overview

Windows NT will be familiar to anyone who has used Windows. The Windows NT, Windows for Workgroups, and Windows 3.1 operating systems all share a common graphical user interface. This interface has become ubiquitous in industry and is standard on more than 70 million desktops worldwide. Virtually all of the tools and facilities of Windows NT use this interface, including the online help engine and the print manager.

A major new release of Windows, Windows 95, will soon be released. Windows 95 sports a new object-oriented user interface that provides users with a work environment that closely models a traditional office. A good example of the change is word-processing.

[3] "Strategies to Control Distributed Computing's Exploding Costs." Inside Gartner Group This Week, April 12, 1995.

With the current interface, users have to select the icon representing the word processor and, once the program has started, select a file to work on. With the new interface, each separate document will be represented by its own icon. When an user selects one, the word processor runs and automatically loads the file. Users no longer need be concerned about where files are located and which programs to run to access them. This new interface will also be available on the next incremental release of Windows NT.

The transition from the Unix command line to Windows NT graphical user interface takes some getting used to. One of the keys to successful system administration on Unix is knowing which flat file to edit for any given configuration change. With Windows NT, configuration information is centrally stored in a database known as the Registry, and virtually all configuration changes are handled with Windows-based tools. The key, then, to system administration on Windows NT is knowing which tool to run, and it is a much easier task than searching the entire file system for an obscure configuration file with equally obscure configuration entries.

However, for typical users on an X-terminal or workstation, the transition to Windows NT can come as welcome relief. If they are already use Windows, Windows NT will be second-nature. For those whose only exposure to a graphical environment has been X, the transition will still be straightforward. They will find that Windows NT is far more standardized than X. The graphical user interface permeates not only the Windows NT operating system, but also the applications that run on it. In addition, it provides a true graphical environment, not just a graphical shell that overlays the command line. The typical user running Windows NT will never have reason to see or use anything on the command line. Contrast this with the typical X user and the beloved **xterm** window. Routine tasks such as printing and getting help also tend to be much easier for the user. These services are graphics-based and standardized across both the operating system and the applications.

Workstation

A workstation running Windows NT is an intelligent device. It consists of a CPU, memory, and disk, along with a monitor, keyboard, and mouse or other pointing device. On the surface, it bears a resemblance to an X-terminal, but is more akin to a Unix workstation running X. It runs its own independent operating system, typically Windows NT Workstation, and participates as a client on the network. In general, it is a smaller machine in terms of CPU, memory, and disk, than its server counterparts, but this need not be the case. When compared to a Unix workstation running X, it is also typically less expensive, both in terms of hardware and software.

Multiuser Support

Windows NT is not a multiuser operating system in the traditional sense of the word. Multiuser is a concept that comes from the one computer, many user paradigm, which is also known as host-based computing. Client/Server is a different world. Users connect to client computers, and client computers connect to server computers. The relationship between users and clients is one-to-one; the relationship between clients and servers is many-to-many. So, in a way, a server is multiuser, it is just that the users are client computers, not people.

Nonetheless, there is an operating system based on Windows NT that is multiuser in the traditional sense. Citrix Systems Inc. was granted a license by Microsoft to modify the

Windows NT source code to add traditional multiuser support. Tektronix has licensed the Citrix product and incorporated it into their Windows Distributed Desktop (WinDD) product.

WinDD is implemented as client/server software. The WinDD server provides access to personal computer applications. The WinDD client is responsible for displaying the output from these applications on Unix workstations or X-terminals. The client also manages the overall display characteristics of the X-terminal or workstation, resulting in reduced loads for both the network and the server.

WinDD appears as a complete Windows NT-based desktop inside an X window. Everything that would typically be displayed on the personal computer monitor, such as color schemes, fonts, wall paper, and application customizations, will be present in the WinDD window on the X-terminal or workstation. This means that WinDD provides the same desktop model to the X-terminals or Unix workstations as Windows NT does to the machines on which it runs.

Logon Names

The administrator account on Windows NT is the closest thing to superuser or **root** on Unix. It is the most powerful logon to the system but does not have the carte blanche powers of **root**. Windows NT also provides a **guest** account that is disabled by default.

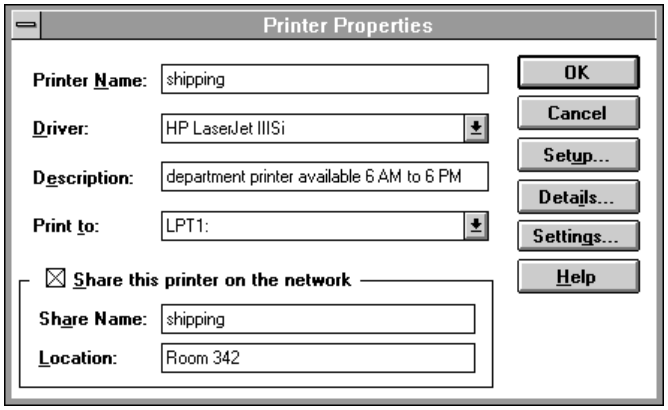
Logon names for Windows NT can be up to 20 characters in length. Uppercase and lowercase characters are permitted, but the names are not case-sensitive. For example, user names: marcg, Marcg, MarcG, and MARCG, all represent the same user. If the user is added as MarcG, then that is the way the name will appear in listings on the system. The user, however, can logon with any of the case combinations shown. Logon names for Windows NT also differ from those for Unix in that they cannot contain unprintable characters such as backspace or tab. There are other illegal characters as well such as [,], ?, >, and <.

Passwords have similar limitations. They can be up to 14 characters in length. They cannot contain unprintable characters, but can contain the other characters that are illegal for logon names. Unlike logon names, they really are case-sensitive.

Applications

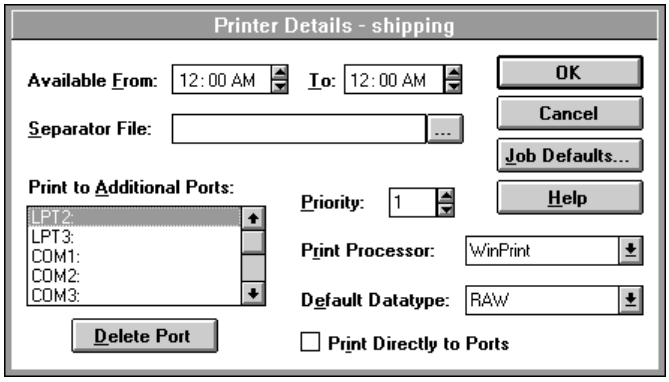
Most packages written for Windows 3.x run on Windows NT. This means that there are literally thousands (over 11,000 at last count) of shrink-wrapped, off-the-shelf packages available today, and they all share a common user interface. The same cannot be said for Unix; an accurate count is hard to derive because, again, it depends on which version of Unix you are talking about. The one thing you can be sure of is that the user interface of these applications will not be the same or even similar, in most cases.

To compliment Windows NT, there is an integrated suite of Microsoft add-on applications, known as the BackOffice, that provides enterprise functionality such as a relational database server (SQL Server™), IBM connectivity (SNA Server), enterprise mail messaging (Mail Server) and desktop workstation management (Systems Management Server).



Print Manager — Printer Properties Dialog Box

Windows NT currently ships with built-in support for over 950 printers. Printers are set up and controlled graphically on Windows NT with the Control Panel and Print Manager applications. Printers can be connected locally to a workstation or, more typically, remotely on the network. With the exception of naming convention, network printers and local printers function identically. Once the printers are configured, all applications in Windows NT have access to them, and, for the user, printing involves little more than selecting a icon with the mouse or making a menu selection.



Print Manager — Printer Details Dialog Box

System administration tasks are simplified through the Windows-based tools as well. With the Windows NT Print Manager, the system administrator can configure and control the following features:

- Set the hours a printer is available.
- Print a separator page showing who submitted the print job along with the date and time.
- Create a pool of printers.
- Set printer priority.
- Specify a custom print processor.

As a system administrator, you can manage local and remote printers from any workstation on the network. The capabilities include:

- Viewing a list of printers and their respective print jobs.
- Purging jobs waiting for a printer.
- Holding or releasing a print job.
- Restarting a print job from the beginning.
- Deleting a print job.
- Stopping a job that is currently printing.
- Pausing and continuing a printer.
- Removing a printer.

Printers can also be audited and have usage tracked. For a particular printer, you can specify which groups or users and which actions to audit. An action's success, as well as its failure, can be audited.

LPR, LPD. LPR is one of the network protocols in the TCP/IP protocol suite. It was originally developed as a standard for transmitting print jobs between computers running Berkeley Unix. The LPR standard is published as Request For Comment (RFC) 1179. Windows NT complies with this standard, as do most implementations of Berkeley Unix. However, most Unix System V implementations do not comply with this standard, so in most cases Windows NT will not be able to send print jobs to System V computers, or receive print jobs from them. Exceptions are System V computers that are configured to accept BSD jobs; these computers can accept print jobs from Windows NT.

With LPR protocol, a client application on one computer can send a print job to a print spooler service on another computer. The client application is usually named "LPR" and the service (or daemon) is usually named "LPD". Windows NT supplies a command line application, the **lpr** utility, and it supplies the LPR Port print monitor. Both act as clients sending print jobs to an LPD service running on another computer. Windows NT also supplies an LPD service, so it can receive print jobs sent by LPR clients, including computers running Unix and others running Windows NT.

Getting Help

Getting help on Windows NT is much less of an adventure game than it is on Unix. There are no **man** pages in Windows NT. Rather, there is a unified, hypertext help system that follows a single consistent model and is used by all programs throughout the system. It provides both general and context-sensitive help as well as keyword and topic search capabilities. You can also set bookmarks, annotate the text, cut and paste, and print selected topics off-line.

Microsoft also has a technical support line, forums on most of the major online services, and ftp, gopher, and Web sites on the Internet. In addition there are CD-ROM subscription services available, such as the Microsoft Developer Network (MSDN) and the Microsoft TechNet CD.

Hardware Platforms

Overview

Windows NT is a platform-independent, scalable operating system. It is designed to take full advantage of the advanced computing capabilities of either the x86 or RISC-based processors on which it runs. In addition, it supports both single processor and symmetric multiprocessor (SMP) systems. There are actually two Windows NT-based operating systems that share a common source-code base. They are: Windows NT Server and Windows NT Workstation. This is similar to Unix, which runs on both server and workstation. Typically, the workstation is a scaled-down version of the server in terms of both hardware and software, especially software for system administration. The same holds true for Windows NT.

Hardware requirements for Windows NT can be broken down into three main areas: processor, memory, and disk space. As a general rule, you will need more of each for Windows NT Server than for Windows NT Workstation. The minimum processor requirements are a 32-bit x86-based microprocessor (such as Intel® 80386/25 or higher), Intel Pentium®, PowerPC, or other supported RISC-based processor, such as the MIPS® R4000® and Digital Alpha™ AXP™. The minimum memory requirement is 16 MB, although Windows NT Workstation will run in 12 MB on x86-based systems. In general, the minimum disk space requirements for just the operating system are in the 100-MB range. It varies by processor type and whether it is Windows NT Workstation or Windows NT Server. For Windows NT Workstation it is: 75 MB for x86 and 92 MB for RISC. For Windows NT Server it is: 90 MB for x86 and 110 MB for RISC. You will need to add additional disk space for any applications that you run.

Choice

Windows NT is a portable operating system in the true sense of the word. It runs on many different hardware platforms and supports a multitude of peripheral devices. Windows NT gives you choice.

This was the dream of Unix, which, even today, is largely unrealized. There is no such thing as Unix; there are only Unixes, and each is slightly different from and generally incompatible with the others. Sometimes there are even incompatibilities within different versions of Unix from the same vendor. The application software that you buy to run on their workstation version of Unix will not run “as-is” on their server version of Unix. If you want to run it there, you need to purchase a different copy of the same application. In most cases, the same holds true with Unix itself. When you buy Unix you are very often locked in to a single-vendor hardware solution, and that vendor is usually the same one that sold you the copy of Unix.

Cost vs. Performance

In the Unix world, servers tend to be large and host multiple applications. With Windows NT, the servers are not as large and there are generally more of them. A single Unix-based server with a dozen applications might translate into two or three Windows NT-based servers with the applications split among them.

Because it is not apples-to-apples, cost versus performance comparisons become quite difficult. It very much depends on the specifics of your environment, such as number of workstations, number and type of applications, and whether or not there is an existing installed base of one or the other type of system. General data is available from industry

research firms, such as the Gartner Group, The Burton Group, the META Group, and International Data Corporation (IDC). Unfortunately, there is no one-size-fits-all right answer, and you must balance their data with knowledge of your own particular business environment.

Operating System Installation and Configuration

In comparison to Unix, installing and configuring Windows NT is a walk in the park. You do not mess with the kernel, nor do you "hand" edit a bunch of obscure files scattered hither and yon throughout the system. It is much simpler than that. You simply boot up the computer from the setup disk, and it walks you step-by-step through the process. Setup is a GUI-based program that uses standard dialog boxes to ask you for configuration information at various points throughout the process. For example, you are asked for things such as which network protocols to install and which file system to use on a specific disk partition. Once setup has finished, all configuration information is centrally stored in the Windows NT Registry database, and installation of the operating system is complete. The entire process, from start to finish, takes roughly 45 minutes for a complete install and roughly 20 minutes for an upgrade of a previous version of the operating system.

Many common peripheral device drivers are included with Windows NT. Adding a new printer to the system, for instance, is usually just a matter of running Print Manager and answering questions in a series of dialog boxes. In other situations you may need to install additional software. This involves running a setup program and answering a series of configuration questions in dialog boxes. This is the common theme for system administrators. Virtually everything you do involves the same process: run a program, and answer questions in dialog boxes. Contrast this with Unix in which every new installation of peripheral or software is an adventure unto itself.

Because Windows NT is portable, so is system administration for Windows NT. The tools and processes are the same no matter where you are. This is one of the most significant differences between Windows NT and Unix. There is virtually no standardization for system administration tools or techniques across the different versions of Unix, and this is the one area where the lack of commonality is the most glaring.

It is possible for a system administrator to make different versions of Unix appear very similar to the user, but the same cannot be said for system administration. Take, for instance, a tool such as the Windows NT Performance Monitor. There is no common, analogous tool on Unix. Such tools exist, but the names differ from version-to-version, as do the built-in features, the user interface, the command structure, and, worst of all, the type of data they present. So, if you can find it and run it, you still might not be able to make any sense out of the output, based on your previous experience with similar tools.

Application Software Installation and Configuration

Application software installation and configuration for Windows NT follows the same pattern outlined in the preceding section. At no time do you ever touch the kernel or "recompile" the operating system. You run setup and answer questions in dialog boxes. To change configurations, you typically select an *options* menu entry and answer more questions in dialog boxes. If you have ever installed a software package under Windows, you know the process. It is the same one that Windows NT uses.

Error Handling

Unix was designed not to say much. When problems occur, Unix, more often than not, simply stops doing what it was doing without so much as a wave good-bye. Nowhere is this more true than with X. If you set an invalid configuration, X simply ignores it and moves on. It will not say a word to you about the problem; it is up to you to figure it out, which is, again, an adventure.

Windows NT takes a different approach. First, you never edit flat configuration files. You make configuration changes through carefully controlled dialog boxes. This control cuts the chances of error way down. Second, when an error does occur, Windows NT informs you with a descriptive error message in a dialog box. Once the message is acknowledged by you, Windows NT attempts to recover gracefully from the error condition. If it is an application error, the application is terminated, and its resources are returned to the system.

Overview

Many companies, including Microsoft, offer software that enables you to create as much of a Unix-like environment on Windows NT as you need. These solutions range from the POSIX subsystem to a full-blown Unix development environment and just about anything in between. There are X servers, Internet access utilities, various flavors of shells, virtually every common tool known to Unix, even **vi**. This section deals mainly with software that provides user functionality. Another section, "Cross-Platform Application Development" covers the development tools and environments. There will, however, be some overlap between the two sections.

Network Access Utilities

Utilities for network access are included with Windows NT and the *Windows NT Resource Kit*. Commercial packages are also available from third-party sources.

In addition to Remote Access Server, Windows NT includes **ping**, **ftp**, and **telnet**. As you would expect, **ping** and **ftp** are non-graphical, command-line utilities. **Telnet** can be launched from either the command line or from an icon. It is a graphical version complete with VT 52/VT 100™/ANSI terminal emulation.

The *Windows NT Resource Kit* includes software for the following services:

- Gopher Server
- HTTP Server (for World Wide Web Access)
- WAIS Server and Toolkit
- Domain Name System (DNS) Server
- Time Synchronizing Service

The Gopher, HTTP, and WAIS software is provided by the European Microsoft Windows NT Academic Centre (EMWAC) and is under copyright by the University of Edinburgh. It comes "as-is" and without warranty of any kind.

There are a number of commercial packages that provide various combinations of network access utilities such as **telnet**, **ftp**, **gopher**, **archie**, and **mosaic**. Most are geared for connecting to the Internet, but they can be used for general-purpose network access tasks, such as between internal corporate machines. The following table gives examples of some of packages.

Product	Vendor
IBM® Internet Connection for Windows	International Business Machines
Internet Chameleon	NetManage, Inc.
Internet in a Box & Internet Office	SPRY, Inc.
NCSA Mosaic for Windows & Windows NT	National Center for Supercomputing Applications (Software Development Group)
NetScape Navigator	NetScape Communications Corp.
WinGopher	Ameritech Library Services (Academic Division)

Tools

Unix tools for Windows NT are available both commercially and through the public domain.

Tools included in the *Windows NT Resource Kit* are: **ar**, **cat**, **cc**, **chmod**, **chown**, **cp**, **find**, **grep**, **ld**, **ln**, **ls**, **make**, **mkdir**, **mv**, **rm**, **rmdir**, **sh**, **tape** (POSIX tape utility), **touch**, **vi**, **wc**, and **windiff** (Windows-based tool similar to **diff**). C source code is also included in the *Windows NT Resource Kit* for these tools, with the exceptions of **tape** and **windiff**. These tools also come "as-is" and are not supported by Microsoft.
There are a number of commercial packages on the market that provide varying degrees of functionality, from a single tool, such as **emacs**, to a full-up Unix environment. The following is a list of some of the products that are available.

- **MKS Toolkit for Windows NT.** The MKS Toolkit from Mortice Kern Systems, Inc., provides a complete Unix-like user environment, including **ksh**, **vi**, **awk**, and 190 additional utilities and programming tools.
- **Hamilton C shell for Windows NT.** This is a package similar to the MKS Toolkit in that it provides a complete Unix-like user environment. It recreates the original Berkeley Unix C shell and utilities, adding numerous enhancements, and provides over 130 commands including **chmod**, **cp**, **cron**, **cut**, **diff**, **grep**, **head**, **kill**, **more**, **mv**, **printf**, **rm**, **sed**, **tail**, **tar**, **touch**, **tr**, and **wc**.
- **Other Unix-like Environment Packages.** There are two additional packages with Unix-like environment features. They are NuTCRACKER from DataFocus Incorporated and Portage from Consensys Computers, Inc. However, because both of these packages provide significant features for cross-platform software development, discussion of them will be deferred until a later section.
- **Win-EMACS.** This product from Pearl Software Corp. is an example of a single Unix utility that has been ported to Windows. Win-EMACS is a version of the GNU EMACS text editor. It is based on Lucid Emacs 19.6 with windowing, font, and color support. It also includes features, such as clipboard support and drag-and-drop, that are common to Windows.

Public Domain

There are numerous places where you can find public domain software. The commercial online services, such as CompuServe®, America Online®, and Prodigy™, all have libraries of software available for downloading. The Internet provides a veritable orchard of free software, ripe for the picking. Caveat emptor is the main rule to keep in mind when shopping for public domain software; in many cases, you really do get what you pay for.

Here are a couple of places to go on the Internet to get started. They are all World Wide Web (WWW) sites:

Location	Description
http://www.berkeley.edu	University of California at Berkeley
http://www.cmu.edu	Carnegie Mellon University
http://www.microsoft.com	Microsoft
http://www.mit.edu:8001	Massachusetts Institute of Technology
http://www.ora.com	O'Reilly & Associates
http://www.research.att.com	Bell Labs Research
http://www.stanford.edu	Stanford University

X was developed at the Massachusetts Institute of Technology as part of Project Athena. With X, you can produce graphics on one networked station and display them on another. The Unix graphical user interfaces, such as Motif and OpenLook, run on top of X.

A workstation or terminal that is capable of displaying X graphical output is known as an X server. X servers can be either dedicated devices, as in the case of X-terminals, or general-purpose workstations running X server software. X server software is available for Unix and non-Unix platforms. Here are some examples for Windows NT:

Product	Vendor
eXceed for Windows NT	Hummingbird Communications Ltd.
NuTCRACKER X	DataFocus
PC-XWare 2.1	Network Computing Devices (NCD) Inc.
Reflection X	Walker Richer and Quinn (WRQ), Inc.

Network File Systems

Windows NT provides a built-in network file system through its network redirector and server components. Virtually all Unix implementations provide similar functionality. The three most common Unix network file systems are: the Network File System (NFS), the Andrew File System (AFS), and Remote File Sharing (RFS).

NFS, originally developed by Sun Microsystems, allows directories and files to be shared across a network. It is the de facto Unix standard for network file systems and has been ported to many non-Unix operating systems as well. Through NFS, users and software

can access files located on remote systems as if they were local files. It works transparently through the Unix hierarchical file system by grafting a branch from the remote file system onto a mount-point or stub of the local file system. Once attached, it appears as just another limb of the tree, and, to either a user or software, it looks like any other local file.

Just as the network file system for Windows NT has two components, the redirector and the server, so to does NFS with the NFS client and the NFS server. As you would expect, the functionality is similar. The client makes the request and the server services the request. Any given machine can be an NFS client, an NFS server, or both. By now it should come as no surprise that there is third-party software available to turn Windows NT into any combination of NFS client/server. Some of those packages are:

Product	Vendor
BW-Connect NFS and BW-Connect NFS Server for Windows NT	Beame & Whiteside Software, Inc. (subsidiary of Hummingbird Communications, Ltd.)
Chameleon32NFS	NetManage, Inc.
DiskShare for Windows NT	Intergraph Corp.

The next most popular network file system under Unix is AFS. Originally developed at Carnegie Mellon University, it is now commercially distributed by Transarc Corporation, which is owned by IBM. AFS has a somewhat different focus from NFS in that it is geared for very large, widely-dispersed Unix networks.

There is at least one package available for supporting AFS on Windows NT. It is PC-Interface (V.5.0) from Locus Computing Corp.

RFS, developed by AT&T, has been available under Unix System V for a number of years. It is not widely used, and, hence, no packages are available for Windows NT.

Overview

Several products are available that provide access to Windows-based applications from Unix. Each of these products has taken a slightly different approach to the solution. They range from emulators with somewhat limited functionality to modified versions of Windows NT with 100% functionality.

Emulation Basics

An emulator impersonates the real thing. It typically works by providing a pseudo-software library that intercepts the application program's calls and translates them into native-API calls for the machine on which the emulator runs. This library interception/translation can occur in one of two ways. The company that produces the emulator may not have access to the source code for the software library that they wish to emulate. In this situation they simply try to figure out which outputs are produced by various inputs and code for these conditions in their software. This process is known as reverse-engineering. If, on the other hand, the emulator company does have access to the library source code, it makes their job much easier, and they can be certain to handle all conditions in their software. In general, this results in a truer emulator.

Commercial Packages

Windows Interface Source Environment (WISE) is a Microsoft licensing program that enables independent software vendors (ISVs) to integrate their Windows-based applications with Unix. Microsoft has licensed the Windows source code to MainSoft Corporation, Bristol Technology Inc., Insignia Solutions Inc., and Locus Computing Corporation. The products from MainSoft and Bristol are software development kits; those from Insignia and Locus are emulators. The MainSoft and Bristol products are covered later in the "WISE SDK" section. The Insignia and Locus products are covered here.

The WISE emulators enable shrink-wrapped applications for Windows to run unmodified on a wide variety of Unix systems such as Solaris, SCO®, Open Desktop®, and HP-UX®. Because they are based on the source code for Windows, the WISE emulators provide much closer compatibility to the real thing. For example, SoftWindows, the WISE emulator from Insignia Solutions, can run virtually any application written for Windows or MS-DOS.

A WISE emulator intercepts and translates API calls from Windows-based applications into API calls for Unix. In some cases, an additional translation is required at the instruction-set level. All this translation usually results in a sacrifice of execution speed for the application.

Wabi from SunSoft™ is not a WISE product. It intercepts the output from Windows-based applications and converts it into X. Under Wabi, Windows-based applications look more like X-based applications. Additionally, each application comes up in its own separate X window, resulting in a desktop model very different from that of Windows NT.

Because Wabi is not based on the Windows source code, application compatibility is also an issue. At the time of this writing, there are only 23 Wabi-certified applications. They do, however, cover a good range of business productivity applications, such as Microsoft Word, Microsoft Excel, Microsoft PowerPoint®, and Microsoft Access.

Windows Distributed Desktop (WinDD) is not an emulator. It is, however, similar to Wabi and WISE in that it seeks to provide similar functionality. It is a modified version of Windows NT that adds traditional multiuser support. It is produced and marketed by Tektronix and consists of client and server pieces. The WinDD Server software compresses updated screen images and transmits the data over the network. The WinDD client interprets this data and displays the applications. Mouse movements and keyboard input are directed by the local client to the WinDD server. Frequently used images, such as icons, bitmaps, and buttons, are cached in the client's memory, further reducing network traffic and greatly improving the performance of the applications.

WinDD Server software can be loaded on a variety of Intel 486 or Pentium class servers. Tests with a single processor, 90 MHz Pentium server and 25 typical users running 32-bit versions of Microsoft Excel and Microsoft Word and a 16-bit version of Microsoft PowerPoint, revealed 486-DX33 class performance and very low network loading. A typical user was defined as someone who uses one or two applications up to 50% of the time.

Overview

There are a number of tools on the market that enable developers to create applications that run on both Unix and Windows NT. The differences in architecture between the two systems means that a certain amount of discipline is required on the part of the developer to make this work. For Unix developers this is nothing new; they have had to contend with the various incompatible flavors of Unix for years. These tools also enable them to migrate existing applications from Unix to Windows NT and vice versa. Some even enable them to take advantage of the best of both worlds by adding functionality from Windows NT to Unix-based applications or functionality from Unix to Windows-based applications.

Language Support

Microsoft

Microsoft offers the following three integrated development environments for developing full 32-bit software applications for Windows NT:

- Microsoft Visual C++™ for Windows NT (V.2.0)
- Microsoft Visual Basic® for Windows NT (V.3.0)
- Microsoft FORTRAN PowerStation 32 for Windows NT

They combine graphical interface design tools with industrial strength language compilers/interpreters to produce a seamless, fully integrated development environment. They are comparable to similar products for Unix, such as the SoftBench line from Hewlett-Packard.

There are third-party compilers and interpreters available for just about every known language. The Internet and the commercial online services are also good places to look for these language compilers. The following table lists some of the commercial products that are available.

Language	Product	Vendor
Ada	ADA Graduate & ADA Masters	Rocket Shareware
	Classic-Ada	Software Productivity Solutions, Inc.
	DEC™ Ada (for Alpha AXP)	Digital Equipment Corp.
	XD Ada (for Alpha AXP)	EDS-Scicon
COBOL	DEC COBOL (for Alpha AXP)	Digital Equipment Corp.
	Micro Focus® COBOL Workbench for Windows	Micro Focus, Inc.

Language	Product	Vendor
	Visual COBOL for Windows	mbp Software and Systems Technology, Inc.
Forth	LMI Forth for Windows NT	Laboratory Microsystems, Inc.
LISP	Star Sapphire Common LISP	Sapiens Software Corp.
Modula-2	Logitech Modula-2	Symantec Corp.
	Professional Modula-2	Stony Brook Software
Pascal	Borland® Pascal with Objects	Borland International, Inc.
	DEC Pascal (for Alpha AXP)	Digital Equipment Corp.
	NDP Pascal-486	Microway, Inc.
Prolog	Visual Prolog for Windows NT	Prolog Development Center
RPG	Visual RPG	Amalgamated Software of North America, Inc.

Conversion Tools

NuTCRACKER from DataFocus provides a comprehensive Unix development environment on top of Windows NT. NuTCRACKER is a software development kit (SDK) that comes in two versions: NuTCRACKER SDK and NuTCRACKER X/SDK.

The main components of the NuTCRACKER SDK are the NuTCRACKER APIs, the Unix environment, and the *Unix-to-Windows NT Porting Guide*. The NuTCRACKER APIs provides the Unix system libraries on top of the Win32 APIs. This is a full implementation of the Unix system call interface and includes things such as fork, exec, pipes, named pipes, message queues, signals, semaphores, and BSD sockets. For the Unix environment, DataFocus includes a copy of the MKS Toolkit with the NuTCRACKER SDK. The porting guide is provided both as a printed document and as an online help file for Windows.

The NuTCRACKER X/SDK adds an X11R5-based X server and the X/Motif libraries to the standard SDK product. It enables X-based Unix applications to be ported to Windows NT.

NuTCRACKER makes Windows NT look like a normal Unix development environment. Existing Unix application source code can be recompiled and run, often without modification. New Unix applications can also be developed using standard Unix tools such as **vi**, **cc**, and **make**. These applications can be completely Unix-centric, or they can take advantage of the additional functionality of the Win32 APIs. It really provides the best of both worlds and gives developers trained in Unix a comfortable place to call home on Windows NT.

Portage, from Consensys Computers Inc., provides similar functionality to that of NuTCRACKER. It differs from NuTCRACKER in that it is a full port of the actual Unix

SVR4 and SVR4.2 source code (licensed by Consensys from AT&T, USL, and Novell) to Windows NT. It remains structured exactly like native Unix and can be thought of as real Unix running on Windows NT.

The product family is broken into smaller pieces than NuTCRACKER and includes more functionality, such as NFS. The Portage packages include the following:

- Portage Kernel
- Portage Base System (kernel, 140+ utilities, graphical interface, & **man** pages)
- Portage Software Development Kit (SDK)
- Portage X Server
- Portage X/Motif Development Kit (XDK)
- Portage Networking Utilities (configurable **inet** daemon, **ftp**, **telnet**, etc.)
- Portage Network Development Kit (NDK)
- Portage Multiuser Terminal Kit (multiuser dumb terminal logon capability)
- Portage NFS Client
- Portage NFS Server

The Portage Base System is analogous in function to both the MKS Toolkit and the Hamilton C Shell and can be used to fill the same niche.

Distributed Computing Environment (DCE)

The Open Software Foundation (OSF) is a consortium of hardware and software suppliers. In 1990, OSF produced the Distributed Computing Environment (DCE) specification. The goal of DCE is to create a single set of standards and protocols through which to link diverse computers into a unified network.

In a prime example of de facto standards becoming de jure standards, OSF selected the components of DCE from among multiple vendors' technologies. The selected components were then consolidated into an Application Environment Specification (AES). AES is a reference implementation product, in the form of source code and a Validation Test Suite (VTS), that is provided to vendor participants for use in their products.

DCE is often mistakenly thought of as a single technology. The DCE specification is a collection of different technologies, each of which may in turn be used on a variety of different computer systems. Microsoft regards some of these technologies as useful building blocks for an advanced distributed computing infrastructure. Additional components of DCE are perhaps less useful as core building blocks, but still provide a basis for interoperability. The Microsoft position is explained in detail in "The Microsoft Strategy for Distributed Computing and DCE Services" white paper, but, briefly stated, the primary focus at Microsoft is on providing strong support for multi-vendor interoperability through effective use of DCE compatible services and other technologies.

For example, one of the most fundamental technologies of DCE is RPC. It provides the basis of communication and interoperability between the various DCE services. During the development of Windows NT, it was determined that a strong RPC service was required for many of the internal functions within the operating system. Rather than create a new RPC service from scratch, Microsoft used the AES as the basis for the DCE-compatible RPC services in Windows NT. This integral support for RPC allows Windows NT to integrate with DCE at the RPC level. No additional software need be purchased.

Distributed computing in general, and DCE in particular, rest on the foundation of three core services: communications, security, and naming. It is the Microsoft belief that these services need to be made available without resorting to complex low-level APIs. If organizations are to succeed in building applications that meet their changing business environments, distributed services will need to be available at a more simplistic and accessible level. One way in which Microsoft is addressing this need is detailed in the Component Object Model (COM) of OLE. Briefly, COM provides the plumbing needed for applications to use services on a distributed and cross-platform basis.

The beauty of this model is that it means that you will probably never need to write to low-level APIs to enable distributed computing applications. Instead, you will either write or use COM-enabled applications and derive support for DCE and other distributed environments in the process. These abilities are provided by the inclusion of core distributed computing support within the design of Windows NT.

There are numerous third-party packages for Windows NT, that provide either partial or full compatibility with the native DCE APIs. Microsoft has played and continues to play a key role in ensuring that these solutions exist. One such example comes from Digital Equipment Corporation (Digital).

Digital produces a product for Windows NT with full DCE functionality. It is known as *Digital DCE Services for Windows NT V1.0*, and includes the RPC service, Cell Directory Service, Distributed Time Service, DCE Security Service, and DCE Threads service. The product consists of two separate pieces: the Runtime Services and the Application Development Kit. The Runtime Services include all of the DCE client functions and administration tools. The Application Development Kit provides the Interface Definition Language (IDL) and other tools necessary for developers to create DCE-based applications.

WISE SDK

The Windows Interface Source Environment (WISE) is a licensing program from Microsoft that enables customers to integrate applications written for Windows with Unix systems. WISE Software Development Kits (SDKs) enable developers to write to the Windows APIs and use the resulting applications on various Unix systems. For each Unix system, the application source code must be recompiled. WISE SDKs provide compatibility at the source level, whereas the WISE emulators provide compatibility at the binary level.

Windows-based applications call the Windows APIs. The WISE SDK remaps the Windows APIs to X and Unix APIs. The X APIs can either be low-level window creation and manipulation functions (Xlib functions) or high-level toolkit functions (such as Motif). Motif is one of the X-based, graphical user interfaces on Unix. The Motif functions are built from the Xlib functions and provide easy-to-call functions for creating widgets. Widgets are objects, such as menus, dialog boxes, and push buttons, that are the basic building blocks of applications.

Programmers developing simultaneously for Windows NT and Unix can write to the Windows APIs and use the Windows SDK on the personal computer and the WISE SDK on the Unix system(s). Software development of this kind requires more discipline than software development for a single platform, but developers get the tremendous benefit of developing and maintaining only one code base. It is crucial that they avoid writing code

that is specific to one platform or compiler. The underlying architecture of Windows NT is different from that of Unix, and code that uses features specific to one will not be adaptable to the other.

As mentioned earlier, the two companies that market WISE SDKs are MainSoft Corporation and Bristol Technology Inc. Their respective products are MAINWin and Wind/U. Both provide support for Unix platforms from HP, IBM, Silicon Graphics, and Sun, among others.

Conclusion

There are striking similarities between Windows NT and Unix. There are also differences. They share some common ancestors, but each was born to a totally different era. In 1970, the world was made up of host-based terminal computing. In 1990, it was and still is client/server distributed computing. Making the mental shift to the new paradigm is not easy to do, but is absolutely key to understanding where Windows NT is coming from.

The neighborhoods they come from are also very different. The world of academia and research bears little resemblance to that of the corporation. Much of the genius of Unix came from its childhood environment, but so have many of its problems. It is fair to say that it has spent the past several years in therapy trying to overcome them. The architects of Windows NT sought to capitalize on the strengths of Unix while avoiding the same pitfalls. It is an operating system that incorporates much of the best thinking in the industry, and one that rests squarely on the rock-solid foundation of Windows.

The ambitions of the two have also been quite different. Unix started out with a focus toward academia and has only recently come into the corporate world, as a sort of afterthought. From the beginning, Windows NT has had a commercial, rather than academic, focus. It was born with a real-world pedigree and has always viewed the global market place as home.

The real world is not, however, a world of either-or. Windows NT and Unix can easily coexist in peaceful harmony, and that is the really good news. It means that you can mix and match the two and, where appropriate, capitalize on the strengths of each in your environment. The openness of Windows NT coupled with the cross-platform choices in both environments and development tools ultimately means that you can have your cake and eat it too. What more could you ask for?

Customer Support

Customer support is provided by the Microsoft Support Network. Some services are free; some are not, and services vary outside the United States and Canada. Inside the United States and Canada, the following support options are available:

Information Services
No-cost and low-cost electronic information services are available 24 hours a day, 365 days a year, including holidays.

Microsoft FastTips
(800) 936-4400 on a touch-tone phone. Receive automated answers to common technical problems, and access popular articles from the Microsoft Knowledge Base, all delivered by recording or fax.

Microsoft Download Service
Access, through modem, sample programs, device drivers, patches, software updates, and programming aides (1200, 2400, or 9600 baud; no parity; 8 data bits; 1 stop bit). In the United States, call (206) 936-6735. In Canada, call (905) 507-3022.

Internet
Access the Microsoft Knowledge Base (MSKB) and Software Library (MSL). The Microsoft World Wide Web (WWW) site is located at http://www.microsoft.com. The Microsoft Gopher site is located at gopher.microsoft.com, and the Microsoft FTP site is located at ftp.microsoft.com and can be accessed with the anonymous logon.

CompuServe and America Online
Access the MSKB, the MSL, and participate in MS forums. On CompuServe, type **go mskb**, **go msl**, or **go microsoft** to access these services. On America Online, type **goto microsoft** to access MSKB.

Priority Support
Priority telephone access to Microsoft support engineers is available 24 hours a day, 7 days a week, except holidays.

- In the United States, call (900) 555-2100; $150 (U.S.) per incident. Charges appear on your telephone bill.
- In the United States, call (800) 936-5900; $150 (U.S.) per incident. Charges are billed to your VISA card, MasterCard, or American Express card.
- In Canada, call (800) 668-7975 for more information.

Text Telephone
Microsoft text telephone/teletype device (TT/TDD) services are available for the deaf or hard-of-hearing. In the United States, using a TT/TDD, dial (206) 635-4948 between 6:00 A.M. and 6:00 P.M. Pacific time, Monday through Friday, excluding holidays. In Canada, using a TT/TDD modem, dial (905) 568-9641 between 8:00 A.M. and 8:00 P.M. Eastern time, Monday through Friday, excluding holidays.

Other Support Options

The Microsoft Support Network offers annual fee-based support plans. For information, in the United States, call the Microsoft Support Network Sales Group at (800) 936-3500 between 6:00 A.M. and 6:00 P.M. Pacific time, Monday through Friday, excluding holidays. In Canada, call (800) 563-9048 between 8:30 A.M. and 6:30 P.M. Eastern time, Monday through Friday, excluding holidays. Technical support is not available through these sales numbers.

Other Microsoft Services

The following services are also available from Microsoft:

Microsoft Authorized Support Centers. Microsoft Authorized Support Centers (ASCs) are a select group of strategic support providers who offer high-quality customized support services that include on-site support, integration and implementation services, help desk services, hardware support, development resources, and others. For more information, in the U.S. call (800) 936-3500 between 6:00 A.M. and 6:00 P.M. Pacific time, Monday through Friday, excluding holidays. In Canada, call (800) 563-9048 between 8:00 A.M. and 8:00 P.M. Eastern time, Monday through Friday, excluding holidays.

Microsoft Solution Providers Program. Microsoft Solution Providers are independent developers, consultants, and systems analysts that offer fee-based technical training and support, industry knowledge, objective advice, and a range of value-added services to companies of all sizes. For more information, in the U.S. call (800) 426-9400 between 6:30 A.M. and 5:30 P.M. Pacific Time, Monday through Friday, excluding holidays. In Canada, call (800) 563-9048 between 8:00 A.M. and 8:00 P.M. Eastern time, Monday through Friday, excluding holidays.

Microsoft TechNet. Microsoft TechNet is the front-line resource for fast, complete answers to technical questions on Microsoft systems and desktop products. As a TechNet subscriber you receive:

- Twelve monthly compact discs containing the Microsoft Knowledge Base, Microsoft operating systems product resource kits, customer solutions, key Microsoft conference session notes, and other valuable information.
- Twelve monthly supplemental (drivers and updated files) compact discs containing the Microsoft Software Library.
- A dedicated Microsoft TechNet forum on CompuServe (GO TECHNET).
- WinCIM, a Windows-based application for accessing CompuServe.
- A 20% discount on Microsoft Press books.

For more information, in the United States and Canada, call (800) 344-2121, between 7:00 A.M. and 7:00 P.M. Central time, Monday through Friday.

Windows NT Resource Kit

The *Windows NT Resource Kit*, published by Microsoft Press, contains a wealth of technical information about Windows NT. It is meant to supplement the standard documentation set for Windows NT, not replace it. It consists of the following four books and associated software:

- *Volume 1: Windows NT Resource Guide*
- *Volume 2: Windows NT Networking Guide*
- *Volume 3: Windows NT Messages*
- *Volume 4: Optimizing Windows NT*

Books

Microsoft Press publishes books for the entire family of Microsoft products. It is your best single-source for books about Windows NT. They range from introductory step-by-step tutorials to books about the internals of Windows NT, and cover most audiences along the way. Here are several titles to get you started:

- *Microsoft Windows NT Step by Step*
- *Running Windows NT*
- *Windows NT 3.5 Guidelines for Security, Audit, and Control*
- *Inside Windows NT*
- *Inside the Windows NT File System*
- *Advanced Windows NT*

Microsoft Press books are available wherever quality books are sold and through CompuServe's Electronic Mall (GO MSP). For ordering, in the United States, call (800) 677-7377 (1-800-MSPRESS), and, in Canada, call (800) 667-1115. Additional information about Microsoft Press titles can also be found on the Microsoft Home Page on the World Wide Web. Connect to http://www.microsoft.com, select the Microsoft Sales Information link, and then select the Microsoft Press link.

In business since 1978, O'Reilly & Associates is the leading publisher of books for Unix, X, and related topics. Their books range from general introductions to highly advanced and specialized, and pretty much everything in between. For more information, in the United States and Canada, call (800) 998-9938 or (707) 829-0515, between 6:00 A.M. and 5:00 P.M. Pacific time, Monday through Friday. You can also reach them through electronic mail at nuts@ora.com or on the Internet at http://gnn.com.

Technical White Papers

The following white papers are available from Microsoft:

- *Microsoft Directory Services Strategy*
- *Microsoft Message Queuing (MQ)*
- *Microsoft Windows NT Server 3.5 Remote Access Service (RAS)*
- *Microsoft Windows NT Server: Dynamic Host configuration Protocol (DHCP) and Windows Internet Naming Service (WINS)*
- *Moving Unix Applications To Windows NT*
- *Open Systems: Technology Leadership and Collaboration*
- *The Microsoft Object Technology Strategy*
- *The Microsoft Strategy for Distributed Computing and DCE Services*
- *Windows-family Integration with UNIX Systems*
- *Windows Interface Source Environment (WISE)*
- *Windows NT Server 3.5 Domain Planning for Your Enterprise*

The following white papers are available from The Burton Group:

- *A Market-Driven Approach To Open Systems*
- *Directory Services Strategic Overview: The Advent of Directory-Enabled Computing*

Both Microsoft and The Burton Group maintain World Wide Web (WWW) sites on the Internet. The Microsoft home page is at www.microsoft.com and many of these white papers can be found there. The Burton Group is an information services company specializing in network computing. Subscription information for The Burton Group Report can be found on their home page at www.tbg.com.

Training Materials

The *Windows NT Training* kit, from Microsoft Press, is a hands-on, self-paced interactive training program aimed at those preparing for the Windows NT Workstation and Windows NT Server Certified Professional exams. It contains information of value for anyone who has to install, configure, optimize, integrate, troubleshoot, and support Windows NT Workstation and Windows NT Server. The kit consists of two self-paced workbooks; four disks with utilities, troubleshooting files, and lesson files; and a 30-minute VHS video that explains key concepts for managing Windows NT Server.

Training for Microsoft products, including Windows NT, is available through Microsoft Solution Provider Authorized Training and Education Centers (ATECs). For information about the closest ATEC and course availability, call (800) 765-7768 (1-800-SOL-PROV) or Microsoft University at (206) 828-1507. Information is also available at the Microsoft FTP site on the Internet. Connect to ftp.microsoft.com, and then choose the MS Education and Certification folder in the Services section.

Third-Party Vendor Reference

The following table gives the names, addresses, and phone numbers for the vendors whose products are referenced in this paper.

Vendor	Address & Phone
Amalgamated Software of North America (ASNA), Inc.	611 Spruce, PO Box 1668 Big Bear Lake, CA 92315 800-321-2762; 909-866-9000
America Online, Inc.	8619 Westwood Center Dr. Vienna, VA 22182 800-827-6364; 703-448-8700
Ameritech Library Services (Academic Division)	1007 Church St. Evanston, IL 60201-3622 800-556-6847; 708-866-4944
Argent Software, Inc.	49 Main St. Torrington, CT 06790 203-489-5553
Beame & Whiteside Software, Inc. (subsidiary of Hummingbird Communications, Ltd.)	706 Hillsborough St. Raleigh, NC 27603-1655 800-463-6637; 919-831-8989
Borland International, Inc.	100 Borland Way Scotts Valley, CA 95066-3249 800-233-2444; 408-431-1000
Bristol Technology, Inc.	241 Ethan Allen Hwy. Ridgefield, CT 06877 203-438-6969
Citrix Systems, Inc.	210 University Dr., Suite 700 Coral Springs, FL 33071 800-437-7503; 305-755-0559
CompuServe Inc. (subsidiary of H&R Block, Inc.)	5000 Arlington Centre Blvd., PO Box 20212 Columbus, OH 43220 800-848-8199; 614-457-8600
Computer Associates International, Inc.	One Computer Associates Plaza Islandia, NY 11788-7000 800-225-5224; 516-342-5224
Consensys Computers, Inc.	35 Riviera Dr., Unit 9 Markham, ON, CD L3R 8N4 800-388-1896; 905-940-2900
Creative Interaction Technologies, Inc.	800 Eastowne Dr., Suite 111 Chapel Hill, NC 27514 800-545-2442; 919-419-1694
DataFocus, Inc. (subsidiary of Convergent Solutions, Inc.)	12450 Fair Lake Circle, Suite. 400 Fairfax, VA 22033-3831 800-637-8034; 703-631-6770
Digital Equipment Corp. (Digital)	146 Main St. Maynard, MA 01754-2571 800-344-4825; 508-493-5111

Vendor	Address & Phone
EDS-Scicon (subsidiary of Electronic Data Systems Corp.)	8 New England Executive Park Burlington, MA 01803 800-843-9489; 617-273-3030
Hamilton Laboratories	21 Shadow Oak Dr. Sudbury, MA 01776-3165 508-440-8307
Hewlett-Packard Co.	5301 Stevens Creek Blvd., PO Box 58059, MS 51LSJ Santa Clara, CA 95052-8059 800-637-7740; 415-857-1501
Hummingbird Communications Ltd.	1 Sparks Ave. North York, ON, CD M2H 2W1 416-496-2200
Insignia Solutions, Inc. (subsidiary of Insignia Solutions, Ltd.)	1300 Charleston Rd. Mountain View, CA 94043 800-848-7677; 415-694-7600
Intergraph Corp.	One Madison Industrial Park Huntsville, AL 35894-0014 800-345-4856; 205-730-2000
IBM (International Business Machines)	Old Orchard Rd. Armonk, NY 10504 800-426-3333; 914-765-1900
Laboratory Microsystems, Inc.	PO Box 10430 Marina del Rey, CA 90295 310-306-7412
Locus Computing Corp.	9800 La Cienega Blvd. Inglewood, CA 90301-4440 800-955-6287; 310-670-6500
MainSoft Corp.	1270 Oakmead Pkwy., Suite 310 Sunnyvale, CA 94086 800-624-6946; 408-774-3400
mbp Software and Systems Technology, Inc. (subsidiary of mbp Software & Systems)	1141 Harbor Bay Pkwy., Suite 161 Alameda, CA 94502-6576 800-231-6342; 510-769-5333
Micro Focus, Inc.	2465 E. Bayshore Rd., Suite 200 Palo Alto, CA 94303 800-872-6265; 415-856-4161
Microway, Inc.	PO Box 79, Research Park Kingston, MA 02364 508-746-4678
MKS (Mortice Kern Systems, Inc.)	185 Columbia St., W Waterloo, ON, CD N2L 5Z5 800-265-2797; 519-884-2251

Vendor	Address & Phone
National Center for Supercomputing Applications (Software Development Group)	605 E. Springfield Ave. Champaign, IL 61820-5518 217-244-3473
NetManage, Inc.	10725 N. De Anza Blvd. Cupertino, CA 95014 408-973-7171
Netscape Communications Corp.	501 E. Middlefield Rd. Mountain View, CA 94043 800-638-7483; 415-528-2600
Network Computing Devices (NCD) Inc. (PC-Xdivision)	PO Box 4900, 9590 S.W. Gemini Dr. Beaverton, OR 97005-7161 503-641-2200
New Technology Partners, Inc.	15 Constitution Drive, Suite 176 Bedford, NH 03110 603-472-4000
Pearl Software Corp.	2000 Powell St., Suite 1200 Emeryville, CA 94608 800-946-3622; 510-652-4361
Prodigy Services Co.	445 Hamilton Ave. White Plains, NY 10601 800-776-3449; 914-448-8000
Prolog Development Center	568 14th St. Atlanta, GA 30318 800-762-2710; 404-873-1366
Rocket Shareware	PO Box 39326 Edina, MN 55439 612-474-3654
Sapiens Software Corp.	PO Box 3365 Santa Cruz, CA 95063-3365 408-458-1990
Software Productivity Solutions, Inc.	122 4th Ave. Indialantic, FL 32903 800-447-7760; 407-984-3370
SPRY, Inc. (Internet Division of CompuServe)	316 Occidental Ave., S, Suite 200 Seattle, WA 98104 800-777-9638; 206-447-0300
Stony Brook Software	187 E. Wilbur Rd., Suite 4 Thousand Oaks, CA 91360 800-624-7487; 805-496-5837
Symantec Corp.	10201 Torre Ave. Cupertino, CA 95014-2132 800-441-7234; 408-253-9600
Tektronix, Inc. (Network Displays Division)	26600 Southwest Pkwy., PO Box 1000 Wilsonville, OR 97070-1000 800-547-8949; 503-682-7300

Vendor	Address & Phone
Transarc Corp. (subsidiary of IBM)	707 Grant St., Gulf Tower, 20th Fl. Pittsburgh, PA 15219 412-338-4400
Walker Richer and Quinn (WRQ), Inc.	1500 Dexter Ave. N., PO Box 31876 Seattle, WA 98103-1876 800-872-2829; 206-217-7100

A

access right The permission granted to a process to manipulate a particular object in a particular way (for example, by calling a service). Different object types support different access rights.

application programming interface (API) A set of routines that an application program uses to request and carry out lower-level services performed by the operating system.

asynchronous I/O A method many of the processes in Windows NT use to optimize their performance. When an application initiates an I/O operation, the I/O Manager accepts the request but does not block the application's execution while the I/O operation is being performed. Instead, the application is allowed to continue doing work. Most I/O devices are very slow in comparison to a computer's processor, so an application can do a lot of work while waiting for an I/O operation to complete. See also *synchronous I/O*.

audit policy Defines the type of security events that are logged for a domain or for an individual computer; determines what Windows NT will do when the security log becomes full.

auditing The ability to detect and record security-related events, particularly any attempts to create, access, or delete objects. Windows NT uses *Security IDs (SIDs)* to record which process performed the action.

authentication A security step performed by the Remote Access Server (RAS), before logon validation, to verify that the user had permission for remote access. See also *validation*.

B

batch program An ASCII file (unformatted text file) that contains one or more commands in the command language for Windows NT. A batch program's filename has a .BAT or .CMD extension. When you type the filename at the command prompt, the commands are processed sequentially.

C

character-based A mode of operation in which all information is displayed as text characters. This is the mode in which MS-DOS-based and OS/2 version 1.2 applications are displayed under Windows NT. Also called character mode, alphanumeric mode, or text mode.

client A computer that accesses shared network resources provided by another computer (called a server). For the X Window System of Unix the client/server relationship is reversed. Under the X Window System, this client definition becomes the server definition. See also *server*.

computer name A unique name of up to 15 uppercase characters that identifies a computer to the network. The name cannot be the same as any other computer or domain name in the network, and it cannot contain spaces.

D

Data Link Control (DLC) A protocol interface device driver in Windows NT, traditionally used to provide connectivity to IBM mainframes and also used to provide connectivity to local area network printers directly attached to the network.

default profile See *system default profile, user default profile*.

demand paging Refers to a method by which data is moved in pages from physical memory to a temporary paging file on disk. As the data is needed by a process, it is paged back into physical memory.

device A generic term for a computer subsystem such as a printer, serial port, or disk drive. A device frequently requires its own controlling software called a *device driver*.

device driver A software component that allows the computer to transmit and receive information to and from a specific device. For example, a printer driver translates computer data into a form understood by a particular printer. Although a device may be installed on your system, Windows NT cannot recognize the device until you have installed and configured the appropriate driver.

directory services The defining element of distributed computing, and, ultimately, a logical name space capable of including all system resources regardless of type. The goal is a blending in which the directory and the network become synonymous.

disk caching A method used by a file system to improve performance. Instead of reading and writing directly to the disk, frequently used files are temporarily stored in a cache in memory, and reads and writes to those files are performed in memory. Reading and writing to memory is much faster than reading and writing to disk.

distributed application An application that has two parts – a front-end to run on the client computer and a back-end to run on the server. In distributed computing, the goal is to divide the computing task into two sections. The front-end requires minimal resources and runs on the client's workstation. The back-end requires large amounts of data, number crunching, or specialized hardware and runs on the server. Recently, there has been much discussion in the industry about a three-tier model for distributed computing. That model separates the business logic contained in both sides of the two-tier model into a third, distinct layer. The business logic layer sits between the front-end user interface layer and the back-end database layer. It typically resides on a server platform that may or may not be the same as the one the database is on. The three-tier model arose as a solution to the limits faced by software developers trying to express complex business logic with the two-tier model.

DLC See *Data Link Control*.

DLL See *dynamic-link library*.

domain For Windows NT Server, a networked set of workstations and servers that share a Security Accounts Manager (SAM) database and that can be administered as a group. A user with an account in a particular network domain can log onto and access his or her account from any system in the domain. See also *SAM database*.

domain controller For a Windows NT Server domain, the server that authenticates domain logons and maintains the security policy and the master database for a domain.

Both servers and domain controllers are capable of validating a user's logon; however, password changes must be made by contacting the domain controller. See also *server*.

domain database See *SAM database*.

domain name The name by which a Windows NT domain is known to the network.

Domain Name System (DNS) A hierarchical name service for TCP/IP hosts (sometimes referred to as the BIND service in BSD Unix). The network administrator configures the DNS with a list of *hostnames* and IP addresses, allowing users of workstations configured to query the DNS to specify the remote systems by *hostnames* rather than IP addresses. DNS domains should not be confused with Windows NT *domains*.

dynamic-link library (DLL) An *application programming interface (API)* routine that user-mode applications access through ordinary procedure calls. The code for the API routine is not included in the user's executable image. Instead, the operating system automatically modifies the executable image to point to DLL procedures at run time.

E

environment subsystems User-mode protected servers that run and support programs from different operating systems environments. Examples of these subsystems are the Win32 subsystem, the POSIX subsystem, and the OS/2 subsystem. See also *integral subsystem*.

environment variable A string consisting of environment information, such as a drive, path, or filename, associated with a symbolic name that can be used by Windows NT. You use the System option in Control Panel or the **set** command to define environment variables.

event Any significant occurrence in the system or in an application that requires users to be notified or an entry to be added to a log.

Event Log service Records events in the system, security, and application logs.

Executive module The Kernel-mode module that provides basic operating system services to the environment subsystems. It includes several components; each manages a particular set of system services. One component, the Security Reference Monitor, works together with the protected subsystems to provide a pervasive security model for the system.

extensibility Indicates the modular design of Windows NT, which provides for the flexibility of adding future modules at several levels within the operating system.

F

FAT file system A file system based on a file allocation table maintained by the operating system to keep track of the status of various segments of disk space used for file storage.

fault tolerance The ability of a computer and an operating system to respond gracefully to catastrophic events such as power outage or hardware failure. Usually, fault tolerance implies the ability to either continue the system's operation without loss of data or to shut the system down and restart it, recovering all processing that was in progress when the fault occurred.

file sharing The ability for Windows NT Workstation or Windows NT Server to share parts (or all) of its local file system(s) with remote computers.

file system In an operating system, the overall structure in which files are named, stored, and organized.

FTP service File transfer protocol service, which offers file transfer services to remote systems supporting this protocol. FTP supports a host of commands allowing bidirectional transfer of binary and ASCII files between systems.

Fully Qualified Domain Name (FQDN) In TCP/IP, *hostnames* with their *domain names* appended to them. For example, a host with hostname *tsunami* and domain name *microsoft.com* had an FQDN of *tsunami.microsoft.com*.

G

global account For Windows NT Server, a normal user account in a user's home domain. If there are multiple domains in the network, it is best if each user in the network has only one user account, in only one domain, and each user's access to other domains is accomplished through the establishment of domain trust relationships.

group In User Manager, an account containing other accounts called members. The permissions and rights granted to a group are also provided to its members, making groups a convenient way to grant common capabilities to collections of user accounts.

H

Hardware Abstraction Layer (HAL) Virtualizes hardware interfaces, making the hardware dependencies transparent to the rest of the operating system. This allows Windows NT to be portable from one hardware platform to another.

home directory A directory that is accessible to the user and contains files and programs for that user. A home directory can be assigned to an individual user or can be shared by many users.

host table The HOSTS or LMHOSTS file that contains lists of known IP addresses.

hostname A TCP/IP command that returns the local workstation's *hostname* used for authentication by TCP/IP utilities. This value is the workstation's *computer name* by default, but it can be changed.

I

integral subsystem A subsystem such as the Security subsystem that affects the entire Windows NT operating system. See also *environment subsystems*.

interprocess communication (IPC) The exchange of data between one thread or process and another either within the same computer or across a network. Common IPC mechanisms include pipes, named pipes, semaphores, shared memory, queues, signals, mailboxes, and sockets.

K

kernel The portion of Windows NT that manages the processor.

Kernel module The core of the layered architecture of Windows NT that manages the most basic operations of Windows NT. The Kernel is responsible for thread dispatching, multiprocessor synchronization, hardware exception handling, and the implementation of low-level, hardware-dependent functions.

L

LLC Logical link control, in the Data Link layer of the networking model.

local printer A printer that is directly connected to one of the ports on your computer.

local procedure call (LPC) An optimized message-passing facility that allows one thread or process to communicate with another thread or process on the same computer. The Windows NT protected subsystems use LPC to communicate with each other and with their client processes. LPC is a variation of the remote procedure call (RPC) facility, optimized for local use. Compare with *remote procedure call*.

locale The national and cultural environment in which a system or program is running. The locale determines the language used for messages and menus, the sorting order of strings, the keyboard layout, and date and time formatting conventions.

logon authentication Refers to the validation of a user either locally or in a domain. At logon time, the user specifies his or her name, password, and the intended logon *domain*. The workstation then contacts the *domain controllers* for the domain which verify the user's logon credentials.

LPC See *local procedure call*.

M

MAC Media access control, in the Data Link layer of the networking model.

mandatory user profile For Windows NT Server, a user profile created by an administrator and assigned to one or more users. A mandatory user profile cannot be changed by the user and remains the same from one logon session to the next. See also *personal user profile, user profile*.

MS-DOS-based application An application that is designed to run with MS-DOS and which therefore may not be able to take full advantage of all of the features of Windows NT.

N

named pipe An interprocess communication mechanism that allows one process to send data to another local or remote process. Windows NT named pipes are not the same as Unix named pipes.

NBF transport protocol NetBEUI Frame protocol. A descendant of the NetBEUI protocol, which is a Transport layer protocol, not the programming interface NetBIOS.

NDIS See *Network driver interface specification*.

NetBEUI transport NetBIOS (Network Basic Input/Output System) Extended User Interface. The primary local area network transport protocol in Windows NT.

NetBIOS interface A programming interface that allows I/O requests to be sent to and received from a remote computer. It hides networking hardware for applications.

network device driver Software that coordinates communication between the network adapter card and the computer's hardware and other software, controlling the physical function of the network adapter cards.

network driver interface specification (NDIS) An interface in Windows NT for network card drivers that provides transport independence, because all transport drivers call the NDIS interface to access network cards.

NTFS (Windows NT file system) An advanced file system designed for use specifically with the Windows NT operating system. NTFS supports file system recovery and extremely large storage media, in addition to other advantages. It also supports object-oriented applications by treating all files as objects with user-defined and system-defined attributes.

O

object type Includes a system-defined data type, a list of operations that can be performed upon it (such as wait, create, or cancel), and a set of object attributes. Object Manager is the part of the Windows NT Executive that provides uniform rules for retention, naming, and security of objects.

OLE A way to transfer and share information between applications.

P

packet A unit of information transmitted as a whole from one device to another on a network.

page A fixed-size block in memory.

partition A portion of a physical disk that functions as though it were a physically separate unit.

permission A rule associated with an object (usually a directory, file, or printer) in order to regulate which users can have access to the object and in what manner. See also *right*.

personal user profile For Windows NT Server, a user profile created by an administrator and assigned to one user. A personal user profile retains changes the user makes to the per-user settings of Windows NT and reimplements the newest settings each time that the user logs on at any Windows NT Workstation. See also *mandatory user profile, user profile*.

port A connection or socket used to connect a device to a computer, such as a printer, monitor, or modem. Information is sent from the computer to the device through a cable.

portability Windows NT runs on both CISC and RISC processors. CISC includes computers running Intel 80386 or higher processors. RISC includes computers with MIPS R4000 or Digital Alpha AXP processors.

print device Refers to the actual hardware device that produces printed output.

printer In Windows NT, refers to the software interface between the application and the print device.

print processor A dynamic link library that interprets data types. It receives information from the spooler and sends the interpreted information to the graphics engine.

protocol A set of rules and conventions by which two computers pass messages across a network. Networking software usually implements multiple levels of protocols layered one on top of another.

provider The component that allows a computer running Windows NT to communicate with the network. Windows NT includes a provider for the Windows NT-based network; other providers are supplied by the alternate networks' vendors.

R

redirector Networking software that accepts I/O requests for remote files, named pipes, or mailslots and the sends (*redirects*) them to a network service on another computer. Redirectors are implemented as file system drivers in Windows NT.

remote administration Administration of one computer by an administrator located at another computer and connected to the first computer across the network.

remote procedure call (RPC) A message-passing facility that allows a distributed application to call services available on various computers in a network. Used during remote administration of computers. RPC provides a procedural view, rather than a transport-centered view, of networked operations. Compare with *local procedure call*.

resource Any part of a computer system or a network, such as a disk drive, or memory, that can be allotted to a program or a process while it is running.

right Authorizes a user to perform certain actions on the system. Rights apply to the system as a whole and are different from *permissions*, which apply to specific objects. (Sometimes called a *privilege*.)

RISC-based computer A computer based on a RISC (reduced instruction set) microprocessor, such as a Digital Alpha AXP, MIPS R4000, or IBM/Motorola PowerPC. Compare with *x86-based computer*.

router TCP/IP gateways - computers with two or more network adapters that are running some type of IP routing software: each adapter is connected to a different physical network.

RPC See *remote procedure call*.

S

SAM See *Security Accounts Manager*.

SAM database The database of security information that includes user account names and passwords and the settings of the security policies.

scalability Scalability depends on the overall architecture of the entire application server. The three critical components of a scaleable system are: operating system, application software, and hardware. No one element by itself is sufficient to guarantee scalability. High performance server hardware is designed to scale to multiple processors, providing specific functionality to ease disk and memory bottlenecks. Applications and operating systems, in turn, must be able to take advantage of multiple CPUs. All three components are equally important.

Schedule service Supports and is required for use of the **at** command, which can schedule commands and programs to run on a computer at a specified date and time.

Security Accounts Manager (SAM) A Windows NT protected subsystem that maintains the SAM database and provides an API for accessing the database.

security ID (SID) A unique name that identifies a logged-on user to the security system of Windows NT. A security ID can identify either an individual user or a group of users.

server A LAN-based computer running administrative software that controls access to all or part of the network and its resources. A computer acting as a server makes resources available to computers acting as workstations on the network. For the X Window System of Unix the client/server relationship is reversed. Under the X Window System, this server definition becomes the client definition. See also *client*.

Server service A service in Windows NT that supplies an API for managing the Windows NT-based network software. Provides RPC support, and file, print, and named pipe sharing.

service A process that performs a specific system function and often provides an API for other processes to call. Services in Windows NT are RPC-enabled, meaning that their API routines can be called from remote computers.

session A connection that two applications on different computers establish, use, and end. The Session layer performs name recognition and the functions needed to allow two applications to communicate over the network.

socket Provides an end point to a connection; two sockets form a complete path. A socket works as a bidirectional pipe for incoming and outgoing data between networked computers. The Windows Sockets API is a networking API tailored for use by Windows-based applications.

standards Windows NT provides support for many standards, some of which are: AppleTalk, Apple File Protocol, C2, Connection-oriented Transport Protocol (Class 4), Connectionless Network Protocol (CLNP), Domain Name Service (DNS), Dynamic Host Configuration Protocol (DHCP), Ethernet, Fiber Distributed Data Interface (FDDI), FIPS 151-2, Frame Relay, IEEE 802.x, IEEE 1003.1, IPX/SPX, Integrated Services Digital Network (ISDN), ISO 8073, ISO 8473, ISO 8208, ISO 8314, ISO 8802, ISO 9660, ISO 9945-1, ISO 10646, ITU FAX Standards, ITU Modem Standards, NetWare Core Protocol (NCP), OpenGL™, OSI, POSIX, Point-to-Point Protocol (PPP), Personal Computer Memory Card International (PCMCIA), Serial Line Interface Protocol (SLIP), Simple Network Management Protocol (SNMP), Token Ring, TCP/IP, Unicode, and X.25.

synchronous I/O The simplest way to perform I/O, by synchronizing the execution of applications with completion of the I/O operations that they request. When an application performs an I/O operation, the application's processing is blocked. When the I/O operation is complete, the application is allowed to continue processing. See also *asynchronous I/O*.

system default profile For Windows NT Server, the user profile that is loaded when Windows NT is running and no user is logged on. When the Welcome dialog box is visible, the system default profile is loaded. See also *user default profile, user profile*.

T

TDI See *Transport Driver Interface*.

Telnet service The service that provides basic terminal emulation to remote systems supporting the Telnet protocol over TCP/IP.

text file A file containing only letters, numbers, and symbols. A text file contains no formatting information, except possibly linefeeds and carriage returns. Text files are also known as flat files and ASCII files.

thread An executable entity that belongs to a single process, comprising a program counter, a user-mode stack, a kernel-mode stack, and a set of register values. All threads in a process have equal access to the processor's address space, object handles, and other resources. In Windows NT, threads are implemented as objects.

Transport Driver Interface (TDI) In the networking model, a common interface for network components that communicate at the Session layer.

transport protocol Defines how data should be presented to the next receiving layer in the networking model and packages the data accordingly. It passes data to the network adapter card driver through the *NDIS* Interface, and to the *redirector* through the *Transport Driver Interface*.

trust relationship Trust relationships are links between domains that enable pass-through authentication, in which a user has only one user account in one domain, yet can access the entire network. A trusting domain honors the logon authentications of a trusted domain.

U

Unicode A fixed-width, 16-bit character encoding standard capable of representing all of the world's scripts.

user account Consists of all the information that defines a user to Windows NT. This includes the username and password required for the user to log on, the groups in which the user account has membership, and the rights and permissions the user has for using the system and accessing its resources. See also *group*.

user default profile For Windows NT Server, the user profile that is loaded by a server when a user's assigned profile cannot be accessed for any reason, when a user without an assigned profile logs on to the computer for the first time, or when a user logs on the Guest account. See also *system default profile*, *user profile*.

user mode A nonprivileged processor mode in which application code runs.

user profile Configuration information retained on a user-by-user basis. The information includes all the per-user settings of Windows NT, such as the desktop arrangement, personal program groups and the program items in those groups, screen colors, screen savers, network connections, printer connections, mouse settings, window size and position, and more. When a user logs on, the user's profile is loaded, and the user's environment in Windows NT is configured according to that profile.

user right See *right*.

username A unique name identifying a user account to Windows NT. An account's username cannot be identical to any other group name or username of its own domain or workstation. See also *user account*.

V

validation Authorization check of a user's logon information. When a user logs on to an account on a Windows NT Workstation computer, the authentication is performed by that workstation. When a user logs on to an account on a Windows NT Server domain, that authentication may be performed by any server of that domain. See also *trust relationship*.

virtual DOS machine (VDM) A Windows NT protected subsystem that supplies a complete environment for MS-DOS and a console in which to run applications for MS-DOS or 16-bit Windows. A VDM is a Win32 application that establishes a complete virtual x86 (that is, 80386 or higher) computer running MS-DOS. Any number of VDMs can run simultaneously.

virtual memory Space on a hard disk that Windows NT uses as if it were actually memory. Windows NT does this through the use of paging files. The benefit of using virtual memory is that you can run more applications at one time than your system's physical memory would otherwise allow. The drawbacks are the disk space required for the virtual-memory paging file and the decreased execution speed when swapping is required.

volume A partition or collection of partitions that have been formatted for use by a file system.

W

Win32 API A 32-bit application programming interface for Windows NT. It updates earlier versions of the Windows API with sophisticated operating system capabilities, security, and API routines for displaying text-based applications in a window.

Windows on Win32 (WOW) A Windows NT protected subsystem that runs within a VDM process. It provides an environment for 16-bit Windows capable of running any number of applications for 16-bit Windows under Windows NT.

Windows Sockets An IPC mechanism based on the WinSock specification and compatible with the Berkeley Sockets IPC under Unix. The WinSock specification allows hardware and software vendors to design systems and applications that can access virtually any type of underlying network, including TCP/IP, IPX/SPX, OSI, ATM networks, wireless networks, and telephony networks.

workstation In general, a powerful computer having considerable calculating and graphics capability. For Windows NT, computers running the Windows NT Workstation operating system are called workstations, as distinguished from computers running Windows NT Server, which are called servers. See also *server, domain controller*.

Workstation service A service for Windows NT that supplies user-mode API routines to manage the Windows NT redirector. Provides network connections and communications.

WOW The subsystem for running Windows for MS-DOS under Windows NT; sometimes also called Win16 on Win32.

X

x86-based computer A computer using a microprocessor equivalent to an Intel 80386 or higher chip. Compare with a *RISC-based computer*.