Microsoft

Microsoft Windows NT Server

Enterprise Edition

The easiest operating system for your largest, and most mission critical enterprise servers

Evaluation Guide

A Guide to Evaluating Microsoft Windows NT Server, Enterprise Edition

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, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

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

Microsoft Corporation • One Microsoft Way • Redmond, WA 98052-6399 • USA

0497

Microsoft® BackOffice™

Microsoft® Windows NT® Server Enterprise Edition

Abstract

This guide provides overview information about Windows NT Server, Enterprise Edition, a new release of Windows NT Server which combines all of the integrated, comprehensive, easy capabilities of Windows NT Server with high availability, enhanced performance, and comprehensive services making it the most complete platform for today’s enterprise distributed applications.

For the latest information on Microsoft Windows NT Server, Enterprise Edition, visit **www.microsoft.com/ntserver**.

CONTENTS

MICROSOFT WINDOWS NT SERVER, ENTERPRISE EDITION VERSION 4.0

Introduction

Windows NT Server, Enterprise Edition is a new member of the Windows NT Server family designed to build on the benefits of Windows NT Server 4.0 adding enhanced performance & scalability, higher availability, and expanded services for developing enterprise applications. The product is made up of the following components:

Windows NT Server

The foundation of Windows NT Server Enterprise Edition, is Windows NT Server, an integrated, comprehensive, and easy-to-use server operating system. Windows NT Server provides fast file and print services, robust application support, standards-based communications features, and complete Internet and intranet functionality.

4 GB RAM Tuning (4GT)

With standard Windows NT Server, the per-process address limit is 2 gigabytes (GB) of random access memory (RAM). The 4GT feature of Windows NT Server, Enterprise Edition increases this limit to 3 GB without introducing new APIs. The result is higher performance for applications working with large data sets.

8-processor SMP license

Licensed for use on up to 8-processor symmetric multiprocessing (SMP) servers. (Up to 32-processor support is available from some vendors selling original-equipment-manufacturer versions of Windows NT Server, Enterprise Edition.)

Microsoft Cluster Server (MSCS)

MSCS, when run on a validated two-server cluster configuration, automatically recovers from application or server failures to keep your data and applications online. It also enables you to move workload for balancing loads and to unload servers for planned maintenance without downtime.

Microsoft Transaction Server (MTS) version 1.1

MTS supports a powerful environment that makes it easier to develop and deploy high performance, scalable, and robust enterprise, Internet, and intranet applications. MTS defines an application programming model for developing distributed component-based applications. It also provides a run-time infrastructure for deploying and managing these applications. MTS 1.1 includes MTS 1.0 plus MTS Service Pack 2A. The MTS Service Pack 2A includes a new client configuration utility, support for "Cedar" type libraries, and bug fixes.

Microsoft Message Queue Server (MSMQ)

MSMQ is a fast store-and-forward service that enables applications running at different times to communicate across heterogeneous networks and systems that may be temporarily off line. Applications send messages to MSMQ, and MSMQ uses *queues of messages* to ensure that the messages eventually reach their destinations. MSMQ provides guaranteed message delivery, efficient routing, security, and priority-based messaging.

WINDOWS NT SERVER ENTERPRISE EDITION FEATURES AT A GLANCE

The table on the next three pages provides a brief list of features in Microsoft Windows NT Server Enterprise Edition.

Performance & Scalability	
4 Gigabyte RAM Tuning	With standard Windows NT Server, the per-process address limit is 2 gigabytes (GB) of random access memory (RAM). The 4GT feature of Windows NT Server, Enterprise Edition increases this limit to 3 GB without introducing new APIs. The 4GT does this by reducing the potential RAM allocated to the Windows NT kernel from 2 GB to 1 GB
8-processor SMP License	Licensed for use on up to 8-processor symmetric multiprocessing (SMP) servers. (Up to 32-processor support is available from vendors selling original-equipment-manufacturer versions of Windows NT Server, Enterprise Edition.)
Microsoft Cluster Server	
Provides high availability	Quick, automatic recovery from server or application failures, usually in just seconds.
Complete automatic failover & restart services	Automatically detects hardware and communications failures, recovers resources from failed servers, moves the IP address to the recovery server, restarts failed applications, and re-establishes client connections.
Administrator can easily move applications to another server	Administrators can easily take a server offline for maintenance, permitting "rolling upgrades" of system & application software without interrupting services.
Runs on standard server hardware	Runs on standard server hardware from a variety of vendors for choice and competitive pricing. Cluster Server does not require exotic, expensive interconnects – it can use standard Ethernet – but is also able to exploit faster standard interconnects such as Tandem ServerNet if available.
No requirement that either server be in an idle “standby” mode	Both servers in the cluster are available to run productive work, so there is no wasted "spare" server waiting for a failure.
Easy installation, setup, and administration	Fast, easy installation because Cluster Server is based on standard hardware and familiar Windows software. Easy setup and administration using Windows-based user interface.
Requires little or no changes to existing services and applications	Automatic recovery is provided for file/print services and even for unmodified applications (using a point-and-click Wizard.) No coding is required for basic failover functions.
Easy, cost-effective development of cluster-aware applications	Clustering API is an extension to the Win32 API, the most widespread client-server programming interface used today. Benefits: easy to learn, easy to find programmers, and large market opportunities for cluster-aware development.
Microsoft Transaction Server	
Accessible three-tier application model	Developers build distributed business solutions from software components. Services separate network programming and server “plumbing” from business logic. Applications can be built as “single-user” but deployed as “multi-user.”
ActiveX language support	Transaction Server supports tools that produces ActiveX DLL’s, including Microsoft Visual C++, Microsoft Visual Basic, and Microsoft Visual J++.
Two new APIs and two new interfaces	ActiveX developers already skilled in COM have to learn only two new APIs to deploy solutions in the Transaction Server run-time environment. Extensive knowledge of COM APIs and Win32 APIs is not required. Supports **CoCreateInstance** and **Create Object**, the two common ActiveX APIs used in Visual C++ and Visual Basic, respectively.

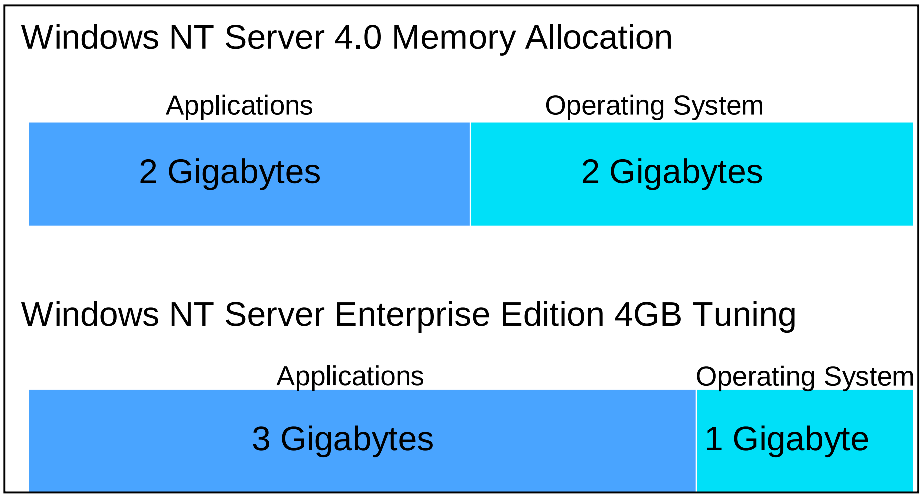
Automatic thread and process management	Developers do not have to create processes or threads as part of their application.
Object instance management	Provides just-in-time object activation, allowing server resources to be used efficiently in high-performance applications.
Component packaging	Packages provide an easy mechanism to collect components into a single application unit, with common security, management, and deployment characteristics.
Database connection pool	Manages a pool of database connections that can be shared by multiple clients.
Shared Property Manager	Multiple concurrent users can easily share global resources.
Automatic transactions through DTC	Integration with the Microsoft Distributed Transaction Coordinator provides a robust transaction management infrastructure. This transaction processing infrastructure enables high-performance server applications. Transaction support is transparent to the programmer.
Process isolation	Administrators can configure components across systems processes for enhanced integrity.
Distributed security service	Transaction Server leverages Windows NT security to control application access and use.
Integration with DCOM	Developers and administrators can deploy distributed component applications using standard technology.
Integration with Microsoft Internet Information Server	Developers can build distributed applications using HTML, HTTP, and Active Server Pages.
Support for multiple databases and resource managers	The architecture supports multiple types of resource managers, including relational databases from different software vendors, file systems, and document storage systems.
Integration with Microsoft SQL Server	Microsoft SQL Server supports the OLE Transactions two-phase commit protocol, for fast Transaction Server applications. Integration of DTC with SQL Enterprise Manager provides common management for SQL Server and Transaction Server applications.
“Fat” and “thin” client support	Transaction Server applications can be accessed by Win32 “fat” clients and HTML “thin” clients at the same time.
No client footprint	Client applications require no client libraries or client runtime.
Dynamic configuration	New clients and servers can be dynamically added and dropped from a distributed application.
Transaction Server Explorer	GUI management console Easier deployment, with the ability to: • Create packages • Install components • Install packages • Export packages across servers • Configure transaction support • Configure security Easier management, with the ability to: • Manage transactions • Monitor transactions • Resolve transactions • View transaction statistics • View trace messages • Manage components

Microsoft Message Queue Server	
ActiveX support	MSMQ provides a set of ActiveX components that implement a convenient application programming interface to all MSMQ features. ActiveX support makes it easy access MSMQ from Microsoft Transaction Server, Microsoft Internet Information Server and Active Server Pages, and a wide range of popular programming languages such as Microsoft's Visual Basic®, Visual C++®, Visual J++™, and Microfocus Cobol.
Guaranteed message delivery	MSMQ guarantees that messages will eventually be delivered even when networks go down, receiving applications are off-line, and when machines fail. This enables developers to focus on business logic and not on sophisticated error recovery programming.
Transaction support	Application developers can include MSMQ operations within transactional units of work that contain other activities such as database updates. This ensures that MSMQ operations commit or abort along with other resources in a transaction to preserve data integrity.
One time, in-order delivery	MSMQ can make sure that messages are delivered exactly one time, and that messages are delivered in the order that they were sent. If messages are delivered more than one time, or out of order, many different kinds of problems can occur within the receiving application.
Message routing services	MSMQ will deliver messages using the 'lowest cost' route that is currently available. When networks fail, MSMQ automatically uses the 'next lowest cost' route to deliver messages. Routing eliminates single points of failure and provides additional resiliency.
Dynamic configuration	The MSMQ Explorer provides a GUI-based interface for managing all machines in a MSMQ environment from a single point of control, and uses a directory service to maintain configuration information. This reduces the cost of administering an MSMQ environment because administrators in a central location can add, delete, move and manage queues dynamically.
Integration with Windows NT Security	MSMQ controls access to queues by integrating tightly with the Windows NT Access Control List mechanism, and ensures message privacy and integrity by using the Windows Crypto API for message encryption and digital signatures.
Notification services	MSMQ can notify sending applications that messages were received and processed, or that a message could not be processed correctly. This lets applications take the appropriate corrective action when failures occur. MSMQ can also journal messages automatically to provide audit trails of activity.
Asynchronous RPC Support	MSMQ enables programmers to make one-way Microsoft RPC calls over MSMQ. With this feature, MSMQ delivers RPC requests to server applications asynchronously so that client applications can move on to other tasks without waiting for responses.
Support for MAPI and Microsoft Exchange	Allows MAPI-based applications to interoperate with MSMQ applications, via the MSMQ MAPI Transport Provider and MSMQ Exchange Connector

4 GIGABYTE RAM TUNING

Applications developed for the Windows NT Server platform continue to grow both in terms of size and performance demands. For applications that are I/O intensive, such as database management systems (DBMS), the use of a larger process space can provide considerable performance benefits as time-intensive I/O access to media is reduced.   This can result in improvements of 20% or more in throughput as measured by tests such as the Transaction Processing Council TPC-C benchmark.

With the current Windows NT Server product, the per-process address limit is 2 GB. The 4GT feature increases this limit to 3 GB without introducing new APIs. This is done by reducing the potential RAM allocated to the Windows NT kernel from 2 GB to 1 GB.



This feature benefits applications that run on powerful machines with more than 2 GB of physical RAM and that can take advantage of a larger address space.

Windows NT Server, Enterprise Edition supports 4GT on Intel Architecture servers. Microsoft is developing a very large memory (VLM) technology for 64-bit processors, such as the Digital Alpha and Intel Merced, which will be included in Windows NT Server 5.0. VLM and 4GT are mutually exclusive since they employ different algorithms. Note that 4GT works only on 32-bit processors, so it does not increase the addressable memory of Windows NT Server beyond its current limit of 4 GB. In contrast, VLM will exploit the 64-bit architecture of processors like Digital Alpha to support up to 32 GB of addressable memory, a potential eight-fold improvement for memory-intensive applications.

No new APIs are required for 4GT support. However, it would be ineffective to automatically provide every application with a 3 GB address space. To provide a selective use of 4GT, executables that must see the 3 GB address space are required to have the bit IMAGE_FILE_LARGE_ADDRESS_AWARE set in their image header. This can be done using the Imagecfg tool that is included in the \Support directory on the Windows NT Server, Enterprise Edition compact disc. For example, to modify the target file DBMSApp.exe:

```
Imagecfg -l DBMSApp.exe
```

Note that the linker also has a new switch (/ LARGEADDRESSAWARE) to link executables with the IMAGE_FILE_LARGE_ADDRESS_AWARE bit. Setting this bit and then running the application on a system that does not have 4GT support should not affect the application.

8-PROCESSOR SMP LICENSE

Windows NT Server Enterprise Edition is licensed for use on up to 8-processor symmetric multiprocessing (SMP) servers. Windows NT is designed to use up to 32 processors. Up to 32-processor support is available from vendors selling original-equipment-manufacturer versions of Windows NT Server, Enterprise Edition on systems, which contain additional CPUs.

MICROSOFT CLUSTER SERVER

Microsoft Cluster Server (MSCS) is a product that provides maximum availability of applications and other resources to clients on a network. Using Cluster Server, you can ensure that when a computer running Windows NT Server on your network experiences problems or must be taken down for maintenance, another server will be there to continue client operations where the first server left off. Unlike other solutions, which require administrators to script failure recovery policies for individual applications and devices, Cluster Server permits graphical setup and automatically manages the policies of whole groups of resources at the operating-system level.

Microsoft Cluster Server Benefits

High Availability

Automatic detection of failures

- Server failure
- Application failure
- Resource failure

Automatic recovery from failures

- Reconnect, recover shareable resources
- Restart applications on same or other server
- Alert administrator with graphical console
- “Failback” to original server once available

Minimal user impact

- Typically recover in 30 seconds
- SMB and IP connections re-establish automatically
- Only “in-flight” processes & non-shareable data lost

Easy Manageability

Fast cluster setup

- Automatically detects storage & network connections
- 2nd server does automated setup from 1st server

Simple definition of “virtual servers”

- Graphically setup co-dependent applications and resources into a “virtual server”
- Simple definition of resource dependencies
- Quick visual status of applications and resources

Graphical management of server workload

- Point-and-click to Take Offline, Bring Online, Move
- Unload servers for planned downtime & maintenance without disconnecting users from data or applications
- Easily change server loads to test or balance demand

Comprehensive Application Support

“In the box” support

- Shared files
- Network-attached printers
- Internet Information Server web pages

Support for non-cluster-aware applications

- Unmodified applications linked to Microsoft Clustering for Windows NT Server with DLLs
- Simple wizard automatically sets up DLLs for applications, file shares, and various resource types

Platform for cluster-aware applications

- Standard API for cluster services on all supported platforms
- Based on Win32, so accessible to thousands of developers
- Software Developer Kit (SDK) with tools & examples

Broad Industry Support

Microsoft is working with the following partners in who are developing Cluster Server-compatible clusters, which will be available, when Cluster Server releases.

- Compaq Computer Corp.
- Digital Equipment Corp.
- Hewlett-Packard Co.
- IBM Corp.
- NCR Corp.
- Tandem Computers, Inc.

Other system vendors announcing plans to offer Cluster Server clusters in 1997:

- Amdahl Computer Corp.
- Fujitsu Personal Systems, Inc.
- Siemens-Nixdorf, Inc.
- Stratus Computer Corp.

Microsoft is committed to providing a computing platform specifically designed for today’s enterprise business applications, while anticipating the needs of the most demanding information systems of tomorrow. To meet these diverse requirements, Microsoft is has worked with leading technology vendors that provide increased availability and scalability of Information Technology (IT) services through clustering technology.

Many enterprise customers now use clustering technology to provide greater availability and scalability for their high-end mission-critical applications such as customer order entry. However, these clustering solutions are complex, difficult to configure, and use expensive proprietary hardware. Microsoft and its industry partners are working to bring the benefits of clustering technology to mainstream client/server computing.

Clustering Defined

In broad terms, a cluster is a group of independent systems working together as a single system. A client interacts with a cluster as though it is a single server. When a system in the cluster fails, the cluster software responds by dispersing the work from the failed system to the remaining systems in the cluster.

Why Clustering is Important

It is estimated that system downtime costs U.S. businesses $4.0 billion per year[1]. The average downtime event results in a $140,000 loss in the retail industry and a $450,000 loss in the securities industry[2]. Clustering promises to minimize downtime by providing an architecture that keeps systems running in the event of a system failure. Clustering also gives organizations the ability to set up separate servers as a single computing facility, thus providing flexibility for future installation growth.

Clustering Example: Data Availability in the Retail Industry

The point-of-sales system is the heartbeat of any retail operation. It provides critical ongoing access to the store's database of products, codes, names, and prices. If the point-of-sale system fails, cashiers cannot log sales and the operation loses money, customers, and its reputation for quality service.

In this case, clustering technology can deliver high system availability. The clustering solution would allow a pair of servers to access the multi-port storage devices (Disk Array) on which the database resides. In the event of a server failure[3] on Server 1, the workload is automatically moved to Server 2 and end-users are switched over to the new server—with no operator intervention and minimal downtime. Note that the disk array itself can be protected by the fault-tolerant disk technology built into Windows NT Server. The addition of clustering technology means that the overall system remains online.

[1] FIND/SVP Strategic Research Division Report, 1992

[2] *ibid.*

[3] Server failure due to hardware (CPU/Motherboard, storage adapter, network card, etc.), application failure, or operator error.

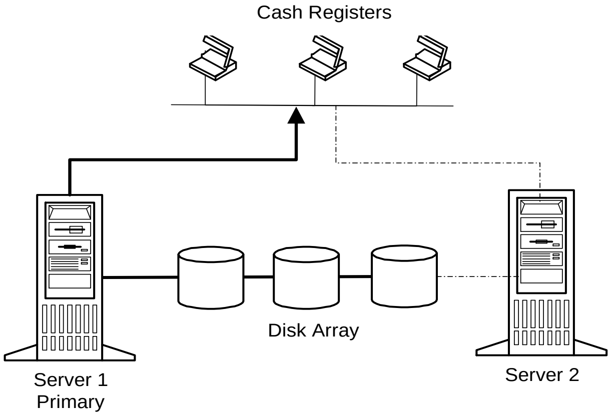


Figure 1: Retail branch before failure

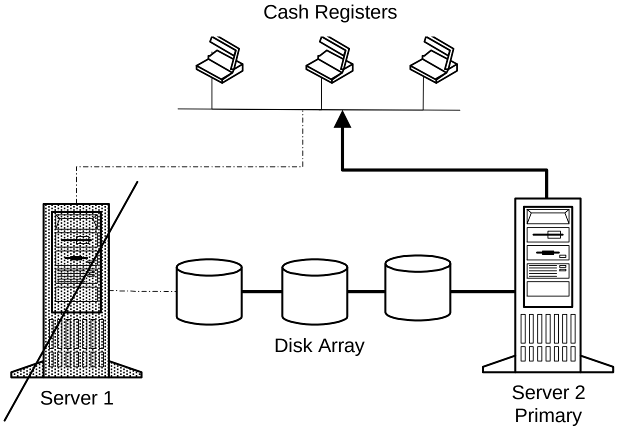


Figure 2: Retail branch during failure

Clustering Example: Scalability in the Financial Services Industry

It has been said that a chief information officer's two greatest fears are system success and system failure. If a system fails, the CIO's staff is inundated with complaints. Conversely, if a system is successful, usage demands tend to outstrip the capacity of the system as it grows.

Clustering can greatly minimize system downtime. In addition, clustering can also help Information Technology (IT) departments design systems that can grow with the demands of an organization.

For example, billions of dollars have been poured into mutual funds companies in recent years. Although this type of growth is positive in financial terms, the technological burden of managing the corresponding information systems growth can be overwhelming. As a result, chief information officers and their staffs must develop systems that not only meet current system demand, but also provide for future system growth. Formerly, the system choices were rather limited: extremely expensive mainframes and minicomputers.

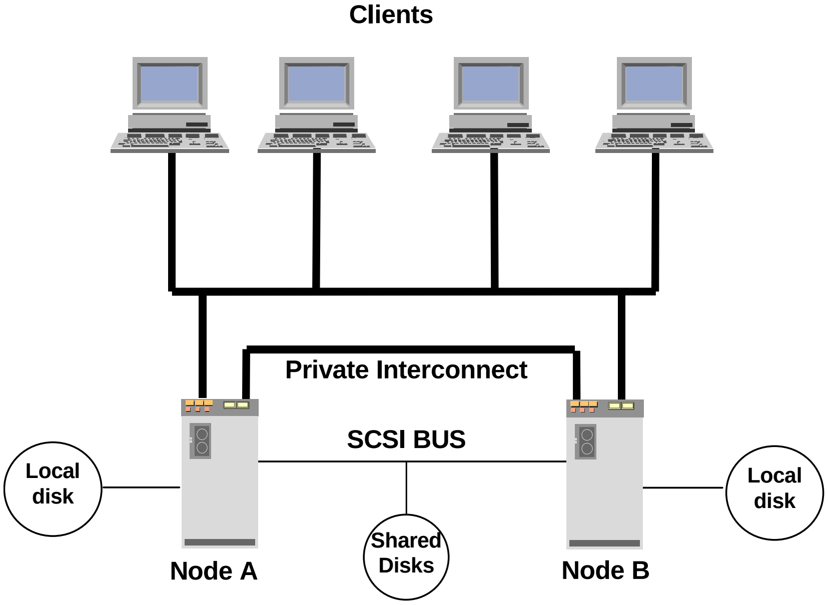
Windows NT Server clustering can provide a competitive advantage for the IT department. This technology allows faster system deployment, automatic re-tasking and easier maintenance with smaller staffs—all while using inexpensive PC components. These components are available from multiple sources, which not only ensures competitive pricing, but also ensures parts availability. Consequently, IT departments can incrementally expand their hardware without the burden of single-supplier shortages.

Clustering technology also gives IT departments greater flexibility. Multiple servers can be tied together in one system, and additional servers can be integrated into the system as usage requirements dictate. Windows NT Server clustering gives a choice to system architects that they have never enjoyed before—availability and scalability on inexpensive, mainstream platforms.

CLUSTER SERVER ARCHITECTURE

Microsoft Cluster Server is based on the concept of *clusters.* A cluster is a pair of computers, called *nodes*, both running Windows NT Server and Cluster Server. These two servers appear to the rest of the network as a single entity: a powerful, highly available “virtual” server. Clients connect to a cluster without knowing which computer is actually distributing their services.

Clusters provide high availability by detecting hardware and software malfunctions and automatically shifting data and processing to healthy nodes. Because a cluster is identified on the network as a single entity, services can move between nodes and whole nodes can even be replaced, all while clients continue to see only one server.



A typical cluster

A cluster is made up of the following components:

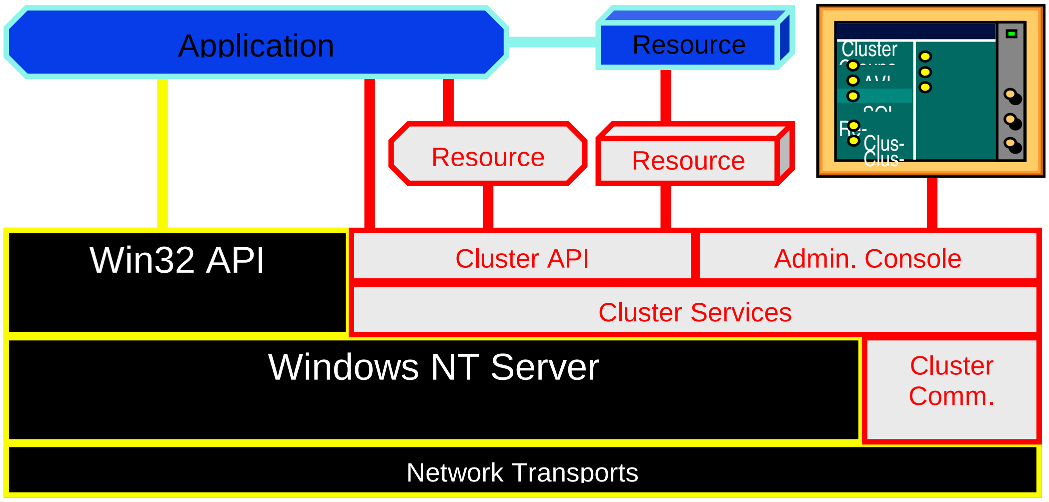
- Two Windows NT Server nodes
- One or more SCSI busses linked to both nodes
- One or more hard disks attached to the SCSI busses

The SCSI-attached disks contain the data required by the applications running on either node. Some SCSI-attached disks called *quorum disks*, also store data used to manage the cluster.

Both nodes in a cluster must connect to the following two network types:

- A Local Area Network (LAN), by which clients interact with the cluster.
- A private network cable, or *interconnect*, that is attached only to the nodes. This cable transfers crucial data as necessary.

Microsoft Cluster Server Technical Architecture

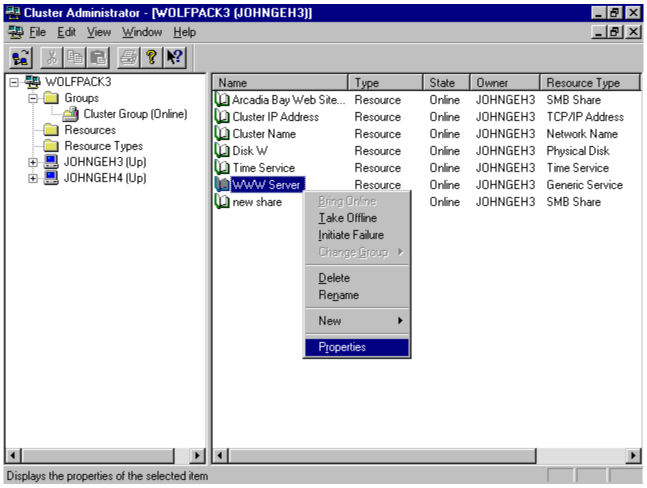


How Clusters Maintain Availability

Microsoft Cluster Server uses a process called *fail-over* to provide highly available services to all clients on your network. Fail-over is the process of transferring control of one or more client resources (applications, disks, print spoolers, and so on) from one node to another.

When a node fails, certain applications, disks, and other client resources in use by that computer "fail-over" to the other cluster node. When a node fails, clients using cluster resources either see no change in activity or else need to reconnect, depending on the resources they are using. When the failing node is replaced or returned to working condition, some or all of the same resources can "fail-back," which means that they are transferred back to the control of the original computer.

As a cluster administrator, you decide the fail-over behavior for all resources in a cluster. You control these behaviors using Cluster Administrator. Cluster Administrator is a graphical console you can use either from one of the nodes or from another computer running Windows NT Server or Workstation to perform various cluster management tasks.



Name	Type	State	Owner	Resource Type
Arcadia Bay Web Site...	Resource	Online	JOHNGEH3	SMB Share
Cluster IP Address	Resource	Online	JOHNGEH3	TCP/IP Address
Cluster Name	Resource	Online	JOHNGEH3	Network Name
Disk W	Resource	Online	JOHNGEH3	Physical Disk
Time Service	Resource	Online	JOHNGEH3	Time Service
WWW Server	Resource	Online	JOHNGEH3	Generic Service
new share		Online	JOHNGEH3	SMB Share

Displays the properties of the selected item

Resources and Groups

Microsoft Cluster Server clusters are populated by various *resources*. A resource is any physical or logical entity that is used to provide a service to clients of the cluster, such as an application or SCSI-attached disk.

Resources run on only one node at a time. Each node in a typical cluster is responsible for hosting a certain number of resources. When a node or resource fails, Cluster Server transfers some resources to the other node, and maintains their availability to clients. Depending on what operations the clients were performing, they might see minimal interruption or they might need to reconnect to the cluster.

Several basic types of resources are supported with the Cluster Server Beta 1 software:

- Windows NT services
- Applications
- SCSI-attached disks
- TCP/IP addresses
- Network names
- File shares
- Print Spoolers
- IIS Virtual Roots for WWW, FTP, and Gopher
- Time Services

In addition to these types, if you use the Cluster Server SDK you can add custom types to this list.

Cluster features will be integrated into existing Windows NT Server system management tools to enable system administrators who are already familiar with Windows NT Server systems to easily setup and configure their clusters. The initial product will include base operating system support for clusters, including components to configure, maintain, and monitor membership in the cluster, support for a cluster-wide name-space, communication, and fail-over support. Additional services will support the two primary cluster software models. As with all of the distributed services provided by Windows NT Server, ease of setup and cluster management tools will be a very high priority.

Microsoft BackOffice

The Microsoft BackOffice family of products, such as Microsoft SQL Server™ and Microsoft Exchange Server, will initially be enhanced to support the cluster's fail-over capability. Later releases of BackOffice will exploit the scalability aspects of cluster technology. SQL Server is planned to support a partitioned data model and parallel execution to take full advantage of the shared nothing environment.

Cluster Application Development

Microsoft development tools will be enhanced to support the easy creation of cluster-aware applications. Facilities will be provided for automatic fail-over of applications. It is also important to note that not all server applications will need to be cluster-aware to take advantage of cluster benefits. Applications that build on top of cluster-aware core applications, such as large commercial database packages (such as an accounting or financial database application on top of SQL Server), will benefit automatically from cluster enhancements made to the underlying application (such as SQL Server). Many server applications that take advantage of database services, client/server connection interaction, Internet/intranet web publishing, and file and print services will benefit from clustering technology without application changes.

MICROSOFT TRANSACTION SERVER

The growth of the Internet and distributed computing will increase the demand for businesses to deploy solutions on servers. The Internet has grown into a widely-used environment for publishing and sharing information online. Businesses can achieve an even greater return on their online systems by moving beyond document sharing to running actual business functions online. Instead of simply publishing sales reports and product catalogs, businesses can operate accounting systems and order-entry systems on servers, with users accessing the shared business functions from browsers and desktop systems. This new environment of online solutions will require shared business applications running on servers.

Historically, there has been a problem with this vision. Building and deploying shared applications running on servers is much harder than on desktop systems. Server applications require a sophisticated infrastructure that is costly to develop and difficult to maintain. Server applications need to be more reliable than desktop applications, because the impact of system failure or data corruption on a server can effect an entire business, not just a single user. Server applications need to be as easy to deploy and maintain as desktop applications, without requiring special training or a costly infrastructure.

To solve these problems, Microsoft has developed Transaction Server, a product that combines the flexibility and low cost of desktop applications with the mission-critical transaction processing features normally found in high-end mainframe systems. Microsoft® Transaction Server is a component-based transaction processing system for developing, deploying, and managing high performance, scalable, and robust enterprise, Internet, and intranet server applications. Transaction Server defines an application programming model for developing distributed component-based applications. It also provides a run-time infrastructure for deploying and managing these applications.

Microsoft Transaction Server provides the easiest way to run scalable, robust applications on the Windows NT® Server operating system.

- **Applications are built from ActiveX™ components, enabling easier development and greater reuse by a broad pool of existing developers.**
- **Transactions are automatically built into applications, providing mission-critical reliability for distributed applications.**
- **Transaction Server is integrated with Windows NT Server, the BackOffice™ family of products, and your existing systems, thus lowering deployment and management costs.**

Lower-Cost Server Development

Microsoft Transaction Server delivers a series of “plumbing” features to drive down the complexity and cost of building applications on the server. These features enable developers to focus on building business logic without having to build an application infrastructure. The result is shorter development time, less complex programming, and easier deployment and integration.

- **Accessible three-tiered application model:** Three-tiered applications, a design where presentation, business, and data logic are separated from each other, provide more deployment flexibility than two-tiered client/server programming, where application code is location-dependent. To date, businesses have had to purchase costly, high-end application platforms to enjoy the benefits of a three-tiered approach. Transaction Server provides an accessible three-tiered model on a commodity platform, delivering high-end distributed features at a much lower cost.

- **ActiveX language support:** Developers build Transaction Server applications as software components using tools that support ActiveX , including Microsoft Visual Basic®, Visual C++®, and Visual J++™. This leverages existing investments in training, people, and the $240 million market for prebuilt ActiveX components.
- **Simple interfaces:** Server environments like TP Monitors and Object Request Brokers require mastering hundreds of APIs before developers become productive. Transaction Server has only two main APIs, which lowers training costs while improving developer productivity.
- **Automatic thread and process management:** Transaction Server manages low-level system resources, enabling components to automatically operate in a multi-user environment without forcing developers to build this complexity into their applications.
- **Object management:** Transaction Server provides "just-in-time" instantiation, making it easier to use components to deliver high-performance applications on servers.
- **Component packaging:** Transaction Server includes a component packaging service, so developers do not have to wrestle with the complicated logistics of integrating, installing, and deploying many components as a single application.
- **Database connection pool:** Transaction Server manages a pool of ODBC connections to a database, providing high-performance database access without forcing the developer to manage complex database synchronization issues.
- **Shared property manager:** Transaction Server provides a mechanism that makes it easy to share data among multiple concurrently executing objects without forcing the developer to program complex state sharing and synchronization logic.
- **Sample applications as learning tools:** Transaction Server includes two sample applications to help developers come up to speed quickly on the product.

Easier development only solves part of the problem that businesses will face with online server applications. Applications need to run reliably and recover from failures accurately. Otherwise, records can be corrupted and business disrupted. The type of reliability and transaction processing features that have been running on mainframe applications for twenty-five years are essential for the new generation of online application servers. Transactions are to application servers what brakes are to cars.

- **Automatic support for transactions:** Transaction Server automatically provides transaction support to applications running on the server, providing a reliable failure isolation and recovery mechanism. Transaction support is transparent to the application programmer, making mainframe-class reliability available to a mass market of solution developers. This transparent support for transactions is a technology breakthrough compared to previous generations of transaction processing systems, where developers had to explicitly program low-level transaction control primitives into their applications.

- **Process isolation:** Using Transaction Server packages, developers and administrators can easily isolate components so they operate in their own system process, providing an additional level of failure isolation and data protection.
- **Automatic distributed security service:** Transaction Server provides a distributed security service that is integrated with Windows NT security, making it easy to prevent unauthorized access to business applications, even if the application includes prebuilt components purchased from third parties.

More Deployment Choices

Easier development and mainframe-class reliability have limited benefits if businesses are forced to deploy the applications in rigid, static configurations. Transaction Server is designed for flexibility, providing businesses with lots of deployment choices based on their existing systems and continuously changing requirements.

- **Location transparency:** Transaction Server application components are location transparent. The exact same components can be deployed in a standalone, workgroup, departmental, divisional, or Internet configurations. Businesses and ISVs only have to support one set of source code for multiple configurations of their solution, thereby lowering their maintenance costs while increasing their deployment options.
- **Integration with DCOM:** Transaction Server uses DCOM for component-to-component communications across a network. DCOM is a fast, native object transport for Windows NT and Windows® 95 and is being licensed as an open-industry standard to technology vendors through the Open Group.
- **Integration with Microsoft Internet Information Server (IIS):** Transaction Server applications can use HTML browsers as clients through integration with Microsoft Internet Information Server 3.0 and ActiveX server pages. This broadens the reach of Transaction Server applications to any desktop system running an Internet browser—including Netscape Navigator.
- **Database and resource manager flexibility:** Transaction Server is designed to work with any resource manager, including relational databases, file systems, and image stores, that support a transactional two-phase commit protocol. This will enable businesses to deploy high-performance applications on Transaction Server that leverage existing investments in UNIX and mainframe data stores.
- **Integration with Microsoft SQL Server™:** Transaction Server is tightly integrated with Microsoft SQL Server through support of the OLE Transactions two-phase commit protocol and through common use of the Distributed Transaction Coordinator. This tight integration provides the lowest possible deployment costs

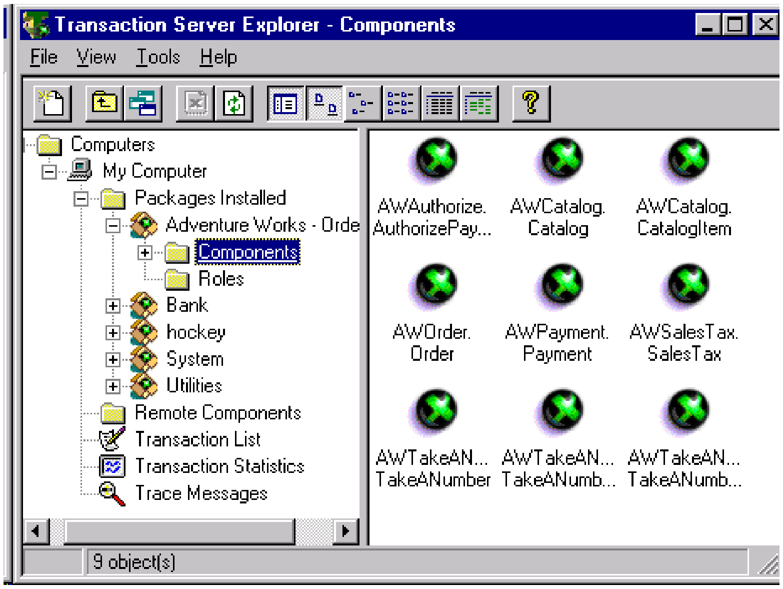
and highest performance for Transaction Server applications.

- **“Fat” and “thin” client support:** Win32® “fat” clients and HTML “thin” clients can access Transaction Server applications at the same time, providing businesses with an unlimited number of client deployment options for Transaction Server applications.
- **No client footprint:** Transaction Server does not require any specific Transaction Server code or libraries to operate on a client system that accesses Transaction Server applications, thus lowering the cost and complexity of distributed applications.
- **Application partitioning:** Administrators can easily partition an application across multiple servers by deploying an application’s components into several packages, with each package running on its own server. This improves system fault isolation, while increasing application performance and scalability.

Easier Management

Businesses have learned that the costs of deploying and managing a distributed application can far exceed the costs of developing and acquiring it. Transaction Server has been designed from the ground up to lower the costs and complexity of distributed applications by providing easier management.

- **Transaction Server Explorer:** Transaction Server includes an integrated GUI management system, the Transaction Server Explorer that provides point-and-click wizards for component, package, transaction, security, and performance configuration and administration. Compared to traditional TP Monitors and Object Request Brokers, the Transaction Server Explorer represents a significant breakthrough in terms of ease-of-installation and management for server applications.



MICROSOFT MESSAGE QUEUE SERVER

Microsoft Message Queue Server (MSMQ), also known by its code name *Falcon*, makes it easy for application programs to communicate with other application programs quickly, reliably and *asynchronously* by sending and receiving *messages*. MSMQ offers a wide range of powerful and innovative features that are tightly integrated with the Windows® 95 and Windows NT® operating system platforms. MSMQ also offers interoperability with other key platforms and applications, such as IBM's CICS and MQSeries, via products from Level 8 Systems.

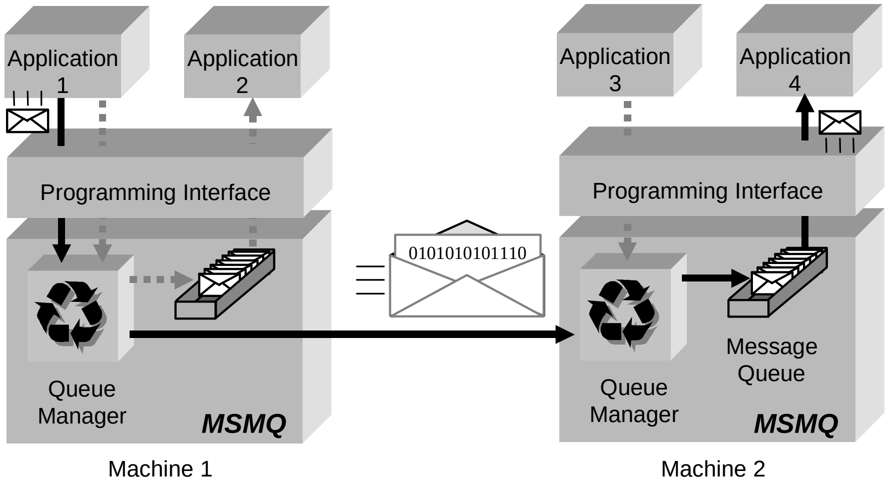
MSMQ provides an easy way for applications to communicate with other applications over a network by sending and receiving *messages.* MSMQ messages can contain data in any format that makes sense to both the sender and the receiver. When an application receives a *request message*, it processes the request by reading the contents of the message and acting accordingly. If required, the receiving application can send a *response message* back to the original requestor.

While in transit between senders and receivers, MSMQ keeps messages in holding areas called *queues* – hence the name *message queuing*. MSMQ queues protect messages from being lost in transit and provide a place for receivers to look for new messages when they are ready. Most important, applications can use MSMQ to send messages and continue processing regardless of whether the receiving application is running or reachable over the network. The receiver may be unreachable because of a problem, or naturally disconnected, as is the case with the applications used by remote or mobile users. Whenever the network becomes available or the receiving application is ready to process requests, MSMQ will deliver any waiting messages – with the *reliability* required by mission-critical applications.

MSMQ offers a wide range of other features such as ActiveX™ support, dynamic routing and configuration, multiple delivery and acknowledgment options and integration with Windows NT security facilities -- making MSMQ the message queuing product of choice for applications running on Windows 95 and Windows NT. While focused on Windows 95 and Windows NT, MSMQ is also interoperable with other important platforms and products -- via products from Level 8 Systems.

A Message Queuing Analogy

A comparison of telephone conversations and electronic mail exchanges provides a good analogy. With a telephone conversation, callers can exchange information immediately and quickly. The telephone is much less useful, however, when one of the two parties cannot be contacted. The telephone line may have become severed, the other party may be busy doing other tasks, or simply out to lunch. In any case, the calling party must continue to dial the receiver periodically in order to deliver an important message. In the case of electronic mail, the calling party could simply send an e-mail and move on with other tasks, knowing that the receiver will eventually get the message and act appropriately. Of course, what makes e-mail a viable alternative to the telephone is the knowledge that messages will be delivered reliably and receivers will eventually read their mail and perform any required actions. Message queuing is like electronic mail, except senders and receivers are *application programs* instead of *people* – and messages are *data* instead of *electronic letters*.



When to Use MSMQ

MSMQ fundamentally is a product for asynchronous communication. The types of communication scenarios where MSMQ is likely to offer value include:

- *Store-and-Forward:* When sending and receiving applications are expected to run at different times
- *Defensive:* In environments that are normally available, but the cost of message loss or interruption is high
- *Concurrent:* When sending applications cannot wait for responses before moving on to other tasks
- *Journaled:* In cases where requests for services must be logged, audited, and possibly re-processed to recover from failures
- *Connectionless:* When connection-oriented communication (i.e. that involves shared state) between a requestor and a receiver (or receivers) is not practical. For example, the ultimate receiver of a message may not be known in advance (as in publish and subscribe applications) and connections can create locking/synchronization problems in multi-user environments.

Example Uses of MSMQ

- Stock trading applications in the financial services industry can use MSMQ to deliver buy and sell orders from desktops to back office processing applications – quickly, reliably and in the order they were sent -- without requiring the desktop to wait for a response.
- Branch banking applications can use MSMQ to communicate with a corporate loan processing application. MSMQ can journal copies of each message automatically to create an audit trail of activity.
- Healthcare applications running at nursing stations can use MSMQ to report the

supplies used by a patient to inventory and billing applications that run at night. Laboratory systems can use MSMQ to report test results back to the nursing stations – even if the network that connects them is temporarily unavailable.

- Insurance salespeople can enter policy orders on their laptop computers while disconnected from the network, and trust MSMQ to forward their orders for processing when they dial in to the network at the end of the day.
- Manufacturing companies can use MSMQ for data collection applications. MSMQ guarantees that all data points will reach collection applications even if networks go down or are temporarily overloaded
- Telecommunications companies can use MSMQ to notify inventory, field installation, line allocation and customer service systems when a new service order is taken. None of these systems must actually be on-line for the order to be accepted by the customer service representative.
- Restaurant chains can use MSMQ to store sales information during the day and forward the information automatically to warehouse distribution centers and regional sales offices at night.

Key MSMQ Features and Benefits

- *ActiveX support*: MSMQ provides a set of ActiveX components that implement a convenient application programming interface to all MSMQ features. ActiveX support makes it easy access MSMQ from Microsoft Transaction Server, Microsoft Internet Information Server and Active Server Pages, and a wide range of popular programming languages such as Microsoft's Visual Basic®, Visual C++®, Visual J++™, and Microfocus Cobol.
- *Guaranteed message delivery*: MSMQ guarantees that messages will eventually be delivered even when networks go down, receiving applications are off-line, and when machines fail. This enables developers to focus on business logic and not on sophisticated error recovery programming.
- *Transaction support*: Application developers can include MSMQ operations within transactional units of work that contain other activities such as database updates. This ensures that MSMQ operations commit or abort along with other resources in a transaction to preserve data integrity.
- *One time, in-order delivery*: MSMQ can make sure that messages are delivered exactly one time, and that messages are delivered in the order that they were sent. If messages are delivered more than one time, or out of order, many different kinds of problems can occur within the receiving application.
- *Message routing services*: MSMQ will deliver messages using the 'lowest cost' route that is currently available. When networks fail, MSMQ automatically uses the 'next lowest cost' route to deliver messages. Routing eliminates single points of failure and provides additional resiliency.
- *Dynamic configuration*: The MSMQ Explorer provides a GUI-based interface for managing all machines in a MSMQ environment from a single point of control, and uses a directory service to maintain configuration information. This reduces the cost of administering an MSMQ environment because administrators in a

central location can add, delete, move and manage queues dynamically.

- *Integration with Windows NT Security*: MSMQ controls access to queues by integrating tightly with the Windows NT Access Control List mechanism, and ensures message privacy and integrity by using the Windows *Crypto API* for message encryption and digital signatures.
- *Notification services*: MSMQ can notify sending applications that messages were received and processed, or that a message could not be processed correctly. This lets applications take the appropriate corrective action when failures occur. MSMQ can also *journal* messages automatically to provide audit trails of activity.
- *Asynchronous RPC Support*: MSMQ enables programmers to make one-way Microsoft RPC calls over MSMQ. With this feature, MSMQ delivers RPC requests to server applications asynchronously so that client applications can move on to other tasks without waiting for responses.
- *Support for MAPI and Microsoft Exchange*: allows MAPI-based applications to interoperate with MSMQ applications, via the MSMQ MAPI Transport Provider and MSMQ Exchange Connector.

The version of MSMQ included with Windows NT Server 4.0, Enterprise Edition differs from the version included in Windows NT Server 4.0, Standard Edition in 3 ways. In the standard edition, a single MSMQ server will accept no more than 25 concurrent MSMQ client connections, cost-based routing will not be supported and the MSMQ Connector will not be provided. The MSMQ Connector is required by some interoperability products such as Level 8 Systems bridge to IBM's MQSeries. The standard version of MSMQ is expected to begin shipping with Windows NT Server 4.0 Standard Edition, and become available to existing registered users of Windows NT Server 4.0 – Standard Edition, later in 1997.