



Third in a series
of step-by-step
guides for
the protection of
the Microsoft
Windows Server
System

Designing High Availability for Internet Information Services

Nelson Ruest & Danielle Ruest
A Report by Resolutions Enterprises

Sponsored by  CA XOsoft





End downtime forever!

Organizations today are relying more and more on Web services for the implementation of mission-critical applications. With the advent of Service-Oriented Architectures (SOAs), which make extensive use of the core Hypertext Transfer Protocol (HTTP) and the Secure Sockets Layer (SSL), the Web server from the past has moved into a vital role in the IT infrastructure.

Web servers have always been important and have often been problematic, but with the release of Internet Information Services version 6.0 (IIS) included as part of Windows Server 2003, Microsoft has finally produced a stable, scalable Web server that is secure and reliable. To date, no security flaws have been identified on this critical server role since its release in 2003. Most impressive!

But, just because a server is secure, it doesn't mean you won't face availability issues. Like all server roles, Web services can be brought down for other reasons—patching the operating system, maintenance windows on the servers or simply because of operator error. Once again, as webmaster or Web administrator, you need to make sure that availability is not hampered by the normal operations or by any other cause. How can you do this? There are several methods, both built-in and through third-party tools. If you're interested in making sure that your Web servers or Web services are up 100 percent of the time, then read on.

Setting up Web Server Protection Mechanisms

Web servers, like their other server counterparts, provide essential services to organizations. When they become unavailable, users cry out and management becomes irate. It's in this type of situation that you must do your utmost to bring the service back online. What if it didn't have to be so? What if you knew that whenever an availability issue arises, you can take your time to repair it because another Web server automatically takes on the additional workload? Not only can you do that, but with the application of a little savvy know-how, your users and management might not even be aware that the service is down.

It is possible! You can use several different strategies to ensure your services are always up. It's an easy and straightforward three-step process. Here's how:

- Begin with Proper System Understanding
- Learn to provide Service Protection through built-in, high-availability capabilities
- Rely on Data Replication Mechanisms

Every webmaster or Web administrator should be very familiar with the first step. After all, how can you manage or administer something you don't understand? So, as you know, the very nature of IIS helps a lot at this level. The second step relies more heavily on features that are built into Windows Server 2003 itself. This allows you to build completely redundant systems by relying on the Windows Network Load Balancing service. The third approach may require

some third-party tools for both data and configuration replication, as well as automated failover when needed. Using a combination of these approaches will guarantee that you can end downtime forever for your IIS systems.

Each approach is explained in detail here. Each applies to IIS version 6.0, though in many cases IIS 5.0, running on Windows 2000 Server, can also take advantage of the same strategies.

Step 1: Know Your IIS Architecture

Today's modern Web applications rely on an n-tier architecture where different application roles are played by different servers. For example, Web services apply at the presentation layer and act as front-end devices that interact with users. A second or application layer will provide the business logic that makes the application run. This application or middleware layer may or may not run on the same servers as the front-end services. There are caveats to this pairing depending on the high-availability method you choose to rely on. At the very back-end, you'll have the data store. In Windows-based systems, this will most likely rely on Microsoft SQL Server 2000 or, preferably, 2005. This data store layer is used to persist user information in Web applications. It should also rely on high-availability strategies to make sure that the entire n-tier structure will run 24x7. If you're interested in knowing more about high-availability strategies for SQL Server, see "Designing High Availability for SQL Server," part two of this four-part series Redmondmag.com/techlibrary/resources.asp?id=270.

Webmasters and Web administrators are concerned about the first tier of the solution, the presentation layer, because this is where Web services are the most present. Other server roles are also required, but this selection of services will depend on the type of application you're running, whether it's internal-facing, external-facing or both, and whether it requires user-state persistence. Depending on these factors, these other service roles can include systems such as

Active Directory, the dynamic host configuration protocol (DHCP), the domain name system (DNS) and, of course, a series of security services such as anti-virus, anti-spyware, firewalls and so on to make sure the data is always protected (see Figure 1). Webmasters are not responsible for all of these secondary services, but they're important because they may affect the availability of your Web interfaces. For this reason, it makes sense for webmasters and administrators to at least understand the various interactions between their systems.

As you can see in Figure 1 even the simplest Web Service architecture can quickly become complex, and what complexity can be vulnerable by default. This means that the first step you should take to protect this system is to generate proper documentation on each part and parcel that makes it up. Generate is the right term because if you can produce this type of documentation automatically, why would you want to do it by hand? Of course, you can't quite produce everything automatically—someone has to record what modifications you make to default configurations when you first set up the system, but once the system is in place, you can use tools like Microsoft Visio 2003 to generate graphical images of the architecture. Better yet, when you link it to the Microsoft Baseline Security Analyzer (MBSA), you can use MBSA to

scan your network and view the results as a proper Visio diagram. To do so, you need to add the Microsoft Visio 2003 Connector for MBSA. This will help you as a webmaster to understand the dependencies of the Web sites you support. This documentation will make it easier to troubleshoot availability issues because you'll know where the failures can occur.

But beyond proper documentation, you might want to put in place a proactive monitoring system that not only warns you of issues as they arise, but can also take proactive action as soon as the issue or its potential is detected.

One of the best ways to do that is to rely on Microsoft Operations Manager 2005 (MOM). MOM provides an excellent means of monitoring and controlling Web Services, including .NET Framework and ASP.NET applications. MOM even lets you create custom consoles that are specific to the purpose at hand. This way webmasters and Web administrators can have consoles that are specifically focused on the services they need to monitor. There are several management packs—custom plug-ins that focus on specific applications—available for Web monitoring (see Resources). They include:

- The **Internet Information Services Management Pack** from Microsoft, which is designed to monitor key IIS events, as well as configurations, Web site health and more.
- The **Microsoft Web Sites and Services Management Pack**, which periodically verifies the status of Web links or entire Web pages to validate their health and performance level.
- The **AVIcode .NET Management Pack (Operations Edition) for Microsoft Operations Manager 2005** from Microsoft, which is designed to monitor both .NET and ASP.NET applications, as well as Web services for health status and potential performance bottlenecks.

Each of these can be added on to a basic MOM installation and provide both proactive and reactive monitoring functions, always keeping you abreast of the very latest status of your systems.

In addition, there's a host of information on IIS and the Microsoft Web platform on the Microsoft Web site itself. Make sure you check it out. This way, you can guarantee that you're up-to-date and know how best to configure your Web systems.

FIGURE 1

Typical n-tier Application Structure

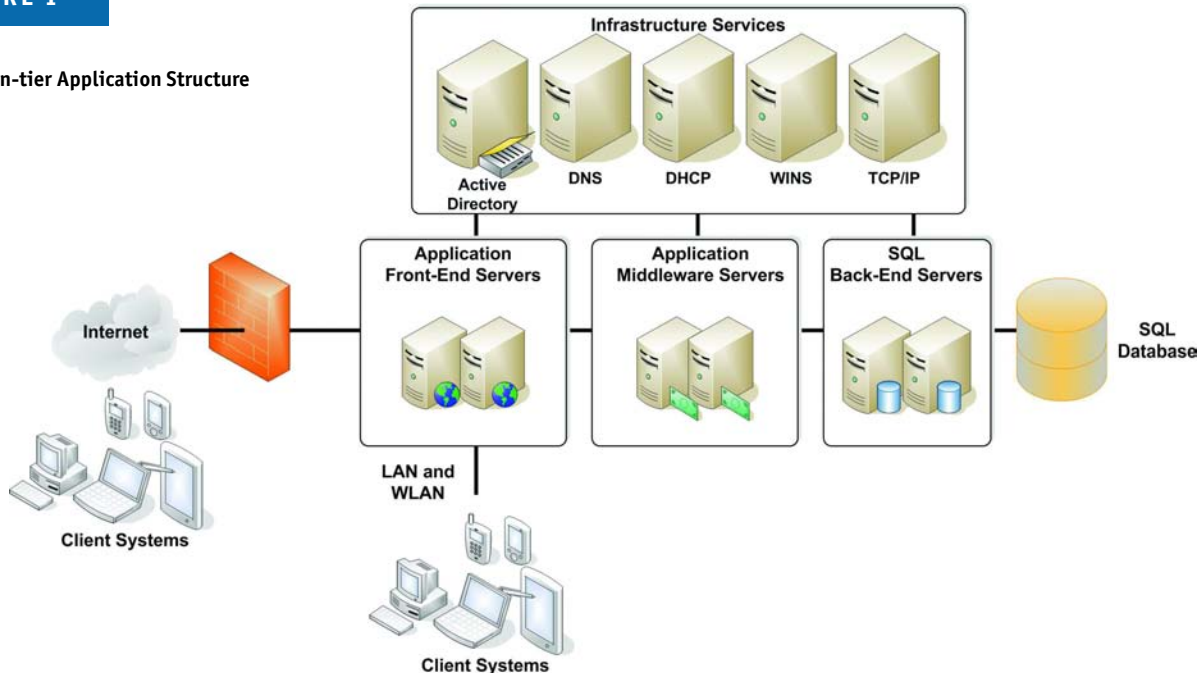
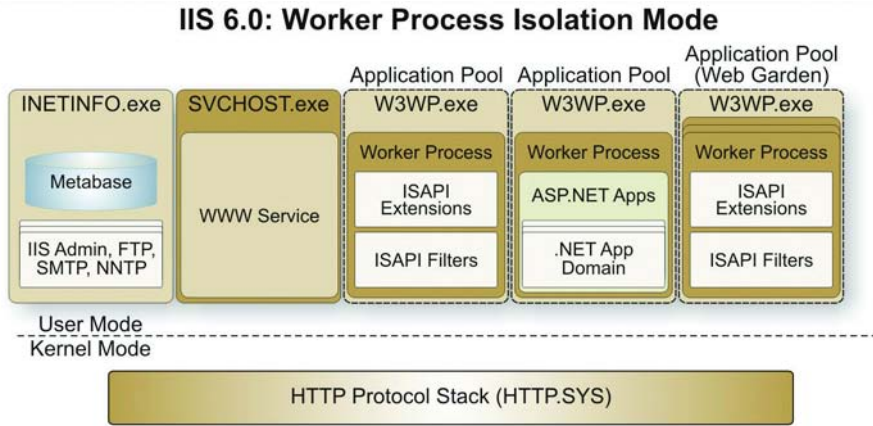


FIGURE 2

The IIS 6.0 Architecture



not work with IIS 6.0—you should upgrade them if this is the case—or they require access to the old DLLHost.exe component of IIS 5.0. Operating in this mode simulates the three application modes of IIS 5.0—In-Process, Pooled Out-of-Process and Isolated Out-of-Process—and all the corresponding stability risks. This mode is activated by default when you upgrade from either IIS 4.0 or 5.0. Bottom line: don't upgrade your servers, or at the very least, make sure you enable the default Worker Process Isolation Mode once the upgrade is complete.

Step 2: Use Built-In High Availability Measures

As you must have discovered by now, IIS 6.0 is leaner, meaner and faster than any previous version. It lets you run multiple Web sites on the same server—almost double the number of sites that IIS 5.0 in some cases—and that includes sites with static or dynamic content. You can also use IIS 6.0 to pool Web resources and expect more from your servers. This means that in terms of availability, you can rely on built-in features of IIS as well as those of Windows Server 2003 itself:

- Application Pooling will allow you to create **Web Gardens** in IIS to protect the systems it hosts.
 - Using the Microsoft Network Load Balancing Service, you can create **Web Farms** to provide highly available Web sites.
- Each of these methods leads to increased availability.

Working with Web Gardens

With the release of Windows Server 2003, Microsoft modified the structure of the IIS architecture. In Windows 2000, IIS version 5.0 was less stable than its new counterpart because of the way the architecture worked, because it included three different application execution modes: In-Process, Pooled Out-of-Process and Isolated Out-of-Process. The first, In-Process, allowed applications to run in the same process as InetInfo.exe; if one application failed, the entire Web server would fail. The second, Pooled Out-of-Process in default mode, allowed applications to execute separately from InetInfo.exe, but all applications executed in the same process. This meant that if an application failed, all other applications failed. The third, Isolated Out-of-Process, provided complete application isolation, but it was rarely used because it required a custom-application configuration.

Microsoft did away with the IIS 5.0 architecture in IIS 6.0. Applications now work in one of two modes: Worker Process Isolation Mode, which is the default for new installations of Windows Server 2003 (see Figure 2) or IIS 5.0 Isolation Mode. The latter is included for backward compatibility purposes. Use this only if your applications do

That's because when you use the new Worker Process Isolation Mode, you can gain much greater reliability and stability for your applications. Each application is automatically isolated by having its own pool of resources—a pool that is completely independent of all of the others on the same server. What's more, IIS 6.0 can automatically recycle each of the applications on a regular basis. Recycling can be performed based on duration of operation, number of requests, scheduled time or even memory corruption (see Figure 3).

This is great because applications can be recycled or restarted without having to restart the Web server itself. When applications run in this mode, you can monitor their health and control their behavior when untoward events occur (see Figure 4, next page). You can also

FIGURE 3

Recycling Applications in IIS 6.0

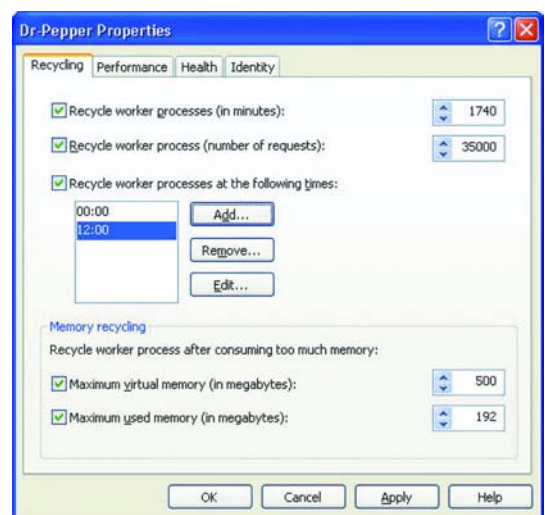


FIGURE 4

Ensuring Application Health in IIS 6.0

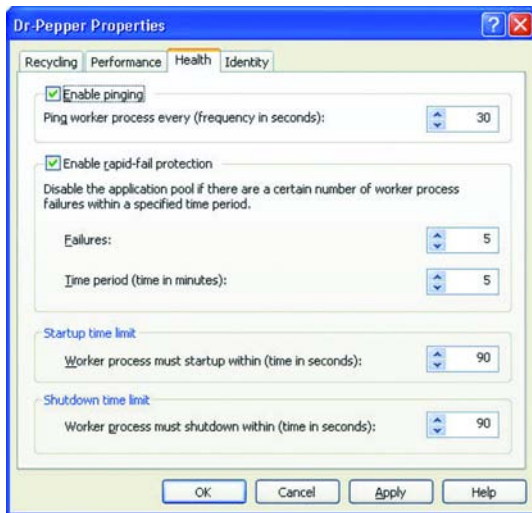
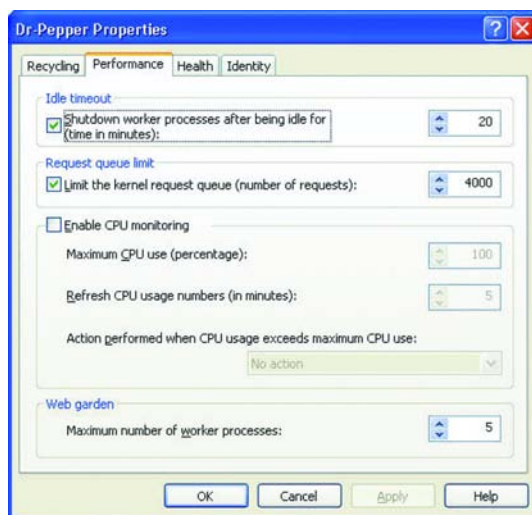


FIGURE 5

Creating a Web Garden



create special application pools, running several applications in a single pool or several instances of the same application in the same pool.

Worker processes support the concept of a Web Garden. Web administrators will be familiar with the concept of a Web Farm—a series of identical servers running the same application and communicating through load-balancing services—but may not know of Web Gardens. In IIS 6.0, a Web Garden is a special application pool that is configured to run multiple worker processes. You do this through the Performance tab of the application pool's properties (see Figure 5). Using a Web Garden lets multiple instances of the application respond

to requests. HTTP.sys automatically redirects requests to the different worker processes in the pool, providing added response for your application. Further stability and performance improvements can be obtained by assigning processor affinity for the pool, forcing it to use specific processors on the server and making sure these processors are reserved for the applications running in the Web Garden.

Working with Web Farms

Web servers often act as front-ends to other applications. When this is the case, the Web server only includes static or stateless information. This means that the Web server operates in more or less a read-only mode because data is being persisted with back-end servers. The advantage of this operation mode is that it becomes very easy to improve availability by building Web Farms—groups of identical IIS servers that perform the same function and respond to the same address. This is done with the Microsoft Network Load Balancing (NLB) service.

NLB basically remaps the media access control (MAC) address of each server's network interface card (NIC) so that they can respond with the same address, creating a pool of servers that can include up to 32 servers (see Figure 6, next page). NLB is easy to set up. Your server's physical configurations do not need to be the same, but the software configurations should because you want the Web servers to respond with identical results. Each server should have at least two NICs, one for the NLB service and one for server management.

NLB can be configured in either unicast or multicast mode. Unicast is often the preferred mode because it's simple and easy to implement. But multicast offers special features in an NLB cluster because it allows the NLB NIC to work with two MAC addresses—the MAC that's assigned to the Cluster IP address and the original MAC that's located on the NIC, which is integrated with the cluster. In unicast mode, the original NIC MAC address is disabled, disabling direct communications with the NIC, thus disabling the modification of port rules—rules that determine how the cluster will react when responding to user requests. With multicast, you can also use the Internet Group Management Protocol (IGMP) multicast setting where all multicast communications are limited to the ports that specifically deal with internal NLB communications, limiting switch flooding and increasing performance.

There are some caveats, however. Routers and switches must be compatible to this mode of operation. In fact, routers must support two specific features. First, they must accept an address resolution protocol (ARP) reply that has one MAC address in the ARP message but appears to arrive from a station with another MAC address, which is in the Ethernet header. Second, in multicast mode, it must accept an ARP reply that has a multicast MAC address in the ARP message.

Note: Cisco routers do not support the resolution of unicast addresses to multicast addresses. Therefore, they require static entries to support this cluster operation mode.

There are a number of different parameters available to configure NLB clusters. Choose the ones that best suit your environment. Once it's set up, the NLB cluster will let you stop or drainstop servers—stopping the server after it has finished serving all current requests—one at a time while maintaining service levels. This provides excellent high availability for a low cost.

FIGURE 6

An NLB Cluster can have up to 32 members



Step 3: Protect Your Web Servers with Replication Technology

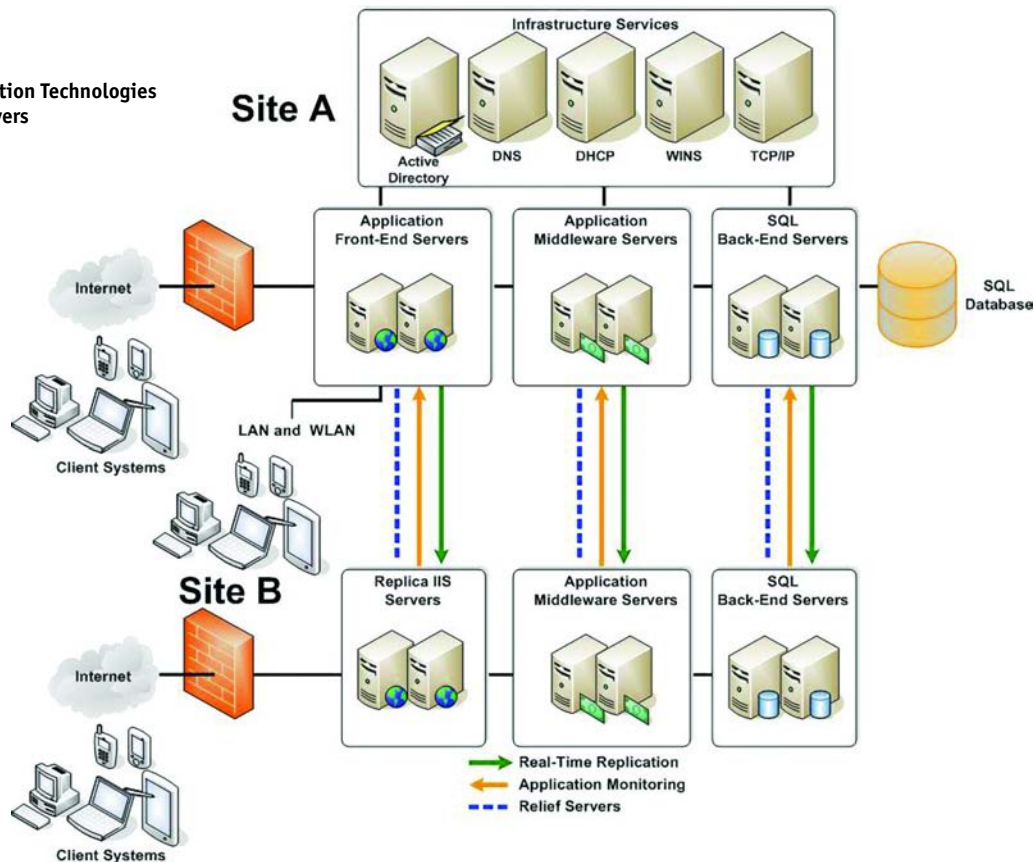
Traditional mechanisms will provide some support for high availability, but they may not be the complete answer. That's because these traditional techniques tend to work only within a single site. If the site is unavailable, then the entire service is unavailable. It is possible to put site-spanning NLB clusters in place, but they require custom site-spanning virtual LANs, which are difficult to implement and maintain.

If you want simpler site-spanning solutions, then you should examine third-party solutions—solutions from CA XOssoft, Symantec, EMC, Double-Take Software and others. These solutions rely on replication technology to create duplicate copies of your Web servers, either in the same or in other sites. These solutions also offer application monitoring, automatic pushbutton failover and automatic failback for complete system protection. Distance is no issue because these tools include bandwidth control capabilities, ensuring your systems are up-to-date without hogging all of the WAN traffic. They also provide automated failover, redirecting Domain Name System (DNS) entries to the failover server, making failover completely transparent and avoiding the need for special clients. They also offer real-time testing so that you know the solution always works.

In addition, the source of the data doesn't matter—your IIS servers can be stand-alone or in an NLB cluster—a replication solution won't care. With these tools, you can set up one of many different high-availability scenarios (see Figure 7).

FIGURE 7

Relying on Replication Technologies to protect SQL Servers



Try it for yourself. Log onto an online demo, or better yet, download an evaluation copy of their software and test it in your own environment. Use this step-by-step process:

- 1 Set up a testing lab using virtual machines. Both Microsoft and VMware offer free copies of their virtualization tools (see Resources). That means you can even do it on a workstation if you have enough RAM. Begin by preparing a copy of the original servers you want to protect, and then create duplicate servers for each. Don't forget to include services such as AD and DNS to support the failover test. To make it easier to capture your existing IIS servers, use the **Microsoft Virtual Server Migration Tool** (see Resources) to create virtual machine copies of the existing physical servers.
- 2 Next, download the trial version of the software from the manufacturer you selected (see Resources).
- 3 Run the tool's installer to install the necessary components on your workstation.
- 4 You might also have to install the replication engine on the Master and Replica servers.
- 5 Now, create a replication and failover scenario. Make sure you use automatic failover. This might imply relying on a DNS redirection.
- 6 Next, test the failover, either automatically or manually. When you fail over, the replica server becomes the holder of live data. Check the tool for a "backward" scenario to ensure that the data changes are replicated back to the original production server. Repeat often until you're satisfied with the results.
- 7 Finally, when you're satisfied that everything works as advertised, acquire a license for the product and install it in production.

That's it. You'll see that replication technology is often easier and faster to set up than anything else. You don't need custom hardware and you don't need complications. It just works.

What's Next ...

If you do take the time to perform this test, you'll be amazed at the results. You'll soon see that this may be the only way to get irate users off your back. So, your next step should be deciding which tool to use. Rely on the following guidelines:

- 1 The engine must be specifically designed to replicate IIS servers, including configuration data.
- 2 The engine must provide Web-server monitoring to identify if or when a failover occurs.
- 3 The engine must make sure the data it copies is valid data and is not corrupted.
- 4 Finally, the engine must support real-time testing to help keep you abreast of the working solution.

End downtime forever. That's what these tools are designed to do. Don't you want to take advantage of that? Just think about the impact it will have on your professional life, you'll soon be everybody's hero.

About the Authors Danielle Ruest and Nelson Ruest, MCSE, MCT, Microsoft MVP, are IT professionals specializing in systems administration, migration planning, software management and architecture design. They are authors of multiple books, notably two books published by McGraw-Hill Osborne, "Windows Server 2003: Best Practices for Enterprise Deployments", ISBN 0-07-222343-X and "Windows Server 2003 Pocket Administrator", ISBN 0-07-222977-2 as well as "Preparing for .NET Enterprise Technologies", published by Addison Wesley, ISBN 0-201-73487-7. They have extensive experience in high availability and systems recovery.



Resources

[Microsoft Web Platforms Home Page](http://www.microsoft.com/windowsserver2003/technologies/webplat/default.aspx): www.microsoft.com/windowsserver2003/technologies/webplat/default.aspx

[Microsoft Internet Information Services Home Page](http://www.microsoft.com/WindowsServer2003/iis/default.aspx): www.microsoft.com/WindowsServer2003/iis/default.aspx

[Microsoft Visio 2003 Connector for Microsoft Baseline Security Analyzer](http://www.microsoft.com/technet/security/tools/mbsavisio.aspx): www.microsoft.com/technet/security/tools/mbsavisio.aspx

[Microsoft Operations Manager 2005](http://www.microsoft.com/mom/default.aspx): www.microsoft.com/mom/default.aspx

[Internet Information Server Management Pack for MOM 2005](http://www.microsoft.com/downloads/details.aspx?familyid=9451088D-87DE-42E9-8FD3-F005A184DD65&displaylang=en):

www.microsoft.com/downloads/details.aspx?familyid=9451088D-87DE-42E9-8FD3-F005A184DD65&displaylang=en

[Microsoft Web Sites and Services Management Pack](http://www.microsoft.com/downloads/details.aspx?FamilyID=53bc39b6-756b-4f01-b0d2-a8ca9751011f&DisplayLang=en):

www.microsoft.com/downloads/details.aspx?FamilyID=53bc39b6-756b-4f01-b0d2-a8ca9751011f&DisplayLang=en

[AVIcode .NET Management Pack \(Operations Edition\) for Microsoft Operations Manager 2005](http://www.microsoft.com/downloads/details.aspx?FamilyID=773CD14A-5763-4739-B775-81416C831846&displaylang=en):

www.microsoft.com/downloads/details.aspx?FamilyID=773CD14A-5763-4739-B775-81416C831846&displaylang=en

[Working with NLB Clusters](http://support.microsoft.com/?id=323437): <http://support.microsoft.com/?id=323437>

[Microsoft Virtual Server 2005](http://www.microsoft.com/windowsserversystem/virtualserver/default.aspx): www.microsoft.com/windowsserversystem/virtualserver/default.aspx

[VMware Server](http://www.vmware.com/products/server/): www.vmware.com/products/server/

[Microsoft Virtual Server Migration Toolkit](http://www.microsoft.com/windowsserversystem/virtualserver/evaluation/vsmt.aspx): www.microsoft.com/windowsserversystem/virtualserver/evaluation/vsmt.aspx

[CA X0soft Solutions site](http://www.XOsoft.com/products/index.shtml): www.XOsoft.com/products/index.shtml

[CA X0soft Download site](http://www.XOsoft.com/download/index.shtml): www.XOsoft.com/download/index.shtml

[EMC RepliStor Web site](http://software.emc.com/products/software_az/replistor.htm): http://software.emc.com/products/software_az/replistor.htm

[Symantec Veritas Replication Exec Web Site](http://www.symantec.com/Products/enterprise?c=prodinfo&refId=50): www.symantec.com/Products/enterprise?c=prodinfo&refId=50

[Double-Take Software Web Site](http://www.nsisoftware.com/default.aspx): www.nsisoftware.com/default.aspx

Build Your Own Web Server Test Bed

The first online segment of this series, “Build your Own SQL Server Test Bed,” provided information on how to create a test bed with virtual machine (VM) technology. If you followed its guidance, you should have a running VM test bed that includes most of the core roles required to test high-availability technologies for your servers. If you haven’t, here’s the link: Redmondmag.com/techlibrary/resources.asp?id=270.

You can continue to add to this server test bed to test other high-availability solutions. In the case of Web servers, you can use this test bed to try out technologies such as Network Load Balancing. Here’s how.

Note: Because several licenses of Windows Server and Windows XP are required in this test bed, it is best to obtain a subscription to either Microsoft TechNet (<http://technet.microsoft.com/en-us/subscriptions/default.aspx>) or Microsoft MSDN (<http://msdn.microsoft.com/subscriptions/>) before proceeding. Each gives you access to 10 licenses of the operating systems as well as licenses of other Windows Server System technologies.

Your default test bed should already be simulating the following roles:

- **Domain Controller** to support integrated authentication scenarios.
- **Primary SQL Server** to support backend data persistence.
- A **Workstation** to act as a management machine as well as a testing machine to test access to the server services.

To test NLB, you will need to add at least two new server roles:

- **Primary Web Server**—a machine running Windows Server 2003 Standard or Web Edition. This role also includes the .NET Framework—installed by default on Windows Server 2003—and Internet Information Services.
- **Secondary Web Server**—a second machine running Windows Server 2003 Standard or Web Edition using the same configuration as the primary Web server.

Use the process identified in the previous online segment to create your Web servers, then, once they’re created, proceed to the implementation of Network Load Balancing.

1. Make sure that both of your Web servers include two network interface cards (NIC). One will be used for the NLB service and one for server management.

2. Before you configure the NLB cluster, you need to make sure the IIS servers are configured in the same manner. The easiest way to do this is to configure the first server, then export its Metabase configura-

tion file and import it on the second server. Use the Backup/Restore Configuration command from the Web server context menu to do so.

3. Next, consider the requirements in **Appendix A — Configuration Worksheet for NLB**. Fill in the requirements as much as you can. The most important aspects of this worksheet is **Part III — Physical Cluster Configuration** because it deals with the actual NLB configuration.

4. The NLB service is installed by default. Activating it and creating your first NLB cluster is very simple. Begin by launching NLB Manager from **Start Menu | Administrative Tools**.

5. IN NLB Manager, right-click on the top node item in the left tree pane and select **New Cluster**. Assign the following values to create your cluster. **Note:** These values will need to be adjusted to work in your environment. You should check with network administrators before doing this or use the Internal or Host Only network settings for your VM.

- a. **Cluster IP Address:** 192.168.100.100
- b. **Subnet Mask:** 255.255.255.0
- c. **Full Internet Name:** WebCluster.domain.com
- d. **Operation Mode:** Multicast
- e. **Operation Mode Option:** IGMP Multicast
- f. **Remote Control:** Unchecked and not allowed

6. Click **Next**, then **Next** again unless you want to add a second IP address to your cluster.

7. Use the default port rule proposed by the wizard. You can experiment with port rules later. For now, you just want to get your cluster operational. Click **Next**.

8. Add your first host. You can do so by typing in the **IP address** or the **host name**. Click on **Connect**. Next, select the **NLB NIC** and click **Next**.

9. Assign the **Priority** of **1** because this is the first node, set the default state as **Started** and click **Finish**. This will create the cluster and add the first node.

10. Right-click on the cluster name to add a second host using similar options as the first. Your cluster is ready to operate.

Before you perform any test, copy the entire test bed. This will allow you to keep a pristine copy and also allow you to destroy any failed tests without any worries. You can also experiment with NLB to see how hosts behave. Right-click on the host name to control its operation. Remember to use the “drainstop” command instead of just stop to make sure user connections are not lost when the host is shut down.

Appendix A – Configuration Worksheet for NLB

Part I — Application or Software

- Application or service to be supported by NLB (Web Server, FTP, Media or VPN server)
- Protocol support required: TCP UDP Both
- Inbound port usage (typically Port 80): _____
- Outbound port usage (numbers can range from 1024-65535): _____
- Is the use of NLB required for: Performance Scaling Fault Tolerance

Part II — Physical Network Constraints

- Assess network and infrastructure risks (redundant routed paths, DNS): Complete Incomplete
- Aggregate throughput requirements for the cluster: Mbps _____
- Select multiple clusters if the aggregate throughput exceeds the segment throughput: Mbps _____
- Determine the throughput for each server: Mbps _____
- Number of hosts required in the cluster (1-32): _____
- Is there a need for public (Internet) and private (Intranet) connections? Yes No
- Are staging servers and interhost communication required for the cluster? Yes No
- Number of network interface adapters required (minimum of 2): _____
- Are there special network considerations? (switched network, proxy servers): Yes No

Part III — Physical Cluster Configuration

- Select multicast or unicast (multicast recommended): Multicast Unicast
- Select the cluster's full Internet name: _____
- Select the cluster's virtual IP address: _____
- Select the subnet mask for the virtual IP of the cluster: _____
- Select the dedicated IP address for each host in the cluster: _____
- Select the subnet mask for the dedicated IP for each host in the cluster: _____
- Select the priority for each host in the cluster: _____

Part IV — Cluster Traffic Handling Configurations

- For each port rule, select the following:
- Select the required port range (Minimum, Maximum): _____
 - Select TCP, UDP, or both for the supported protocols: TCP UDP Both
 - Select the filtering mode for inbound traffic: Multiple Single Disabled
 - Select client affinity: None Single Class C
 - Select load weight for this host when filtering is Multiple (percent): _____
 - Select handling priority for this rule when filtering is single (1-32): _____

Notes
