Building the Foundations for a Highly Available Architecture

Contributed by David Noel-Davies

Cliustering in Windows Server 2003 as these series of articles describes the process of building the forest and forest root domain; the process of providing a resource for OS installations, tools, utilities, and patches; and how to establish the cluster virtual server to begin hosting resources.
INTRODUCTION

Alot of theory has been covered in the previous chapters. Now it’s time to start implementing. This chapter and the ones to follow take what we have discussed up to now and roll it into an implementation plan for a data center that tens of thousands of users will rely upon.

First, we introduce clustering as it is accomplished on the Windows Server 2003 platform. We also discuss cluster concepts, models, and architecture. Then we implement the Active Directory architecture and network architecture as discussed in Chapter 5, "Preparing the Platform for a High-Performance Network," and lay the foundations for a highly available and reliable Web, database, and email server architecture, a network that will eventually comprise NLB IIS servers, NLB application servers, SQL Server clusters, Exchange clusters, and file and print clusters.

You can look at this chapter as the foundation implementation plan. It is what you need to follow if tasked with constructing and deploying a highly available solution. In the practical part, this chapter first outlines the process of building the forest and forest root domain, on either your lab or production network. It also covers the process of providing a resource for OS installations, tools, utilities, and patches. Then we prepare the cluster virtual server to begin hosting resources. In this chapter, you implement Active Directory. At first glance, it seems that you are doing nothing more than setting up the usual AD network. But as you install the various cluster servers and services, you see that what is laid down in this chapter provides the solid foundation for the future systems. Then we deal with the actual process of clustering the servers, setting up cluster resources, and getting ready to activate the fail-over resources in Part II, "Building High Availability Windows Server 2003 Solutions." This is something you cannot do unless AD is well implemented beforehand.

WINDOWS CLUSTERING 101

There was a time in the not-too-distant past when the thought of clustering Windows servers sent a chill down the spines of network engineers and caused them to go take out long-term care insurance. Those days are gone with the clustering services that are now built into the Windows Server 2003 operating system. Only Windows Server 2003, Enterprise Edition and Windows Server 2003, Datacenter Edition can create clusters. Windows Storage Server 2003 is a version of Enterprise Edition for clustering file share resources.

There are two parts to clustering a high-availability service or application, such as Exchange 2003 or SQL Server 2000 or SQL Server 2005. The first part entails setting up the base cluster service and getting a virtual server going. The second part entails creating the resources that failover on that virtual server. Most of the second part of clustering is dealt with in Part II of this book.

By the time you are ready to cluster Exchange or SQL Server, you will be able to failover the virtual server resources from one node to the other and keep services, like drives and network interface cards, under the control of the cluster.

The Cluster Model

With Windows Server 2003, you have three models from which to choose; they are built into the operating system and are, thus, supported by Microsoft. The third option may require third-party software. Table 6.1 discusses the models in order of increasing complexity.

The most common cluster model (and inherited from Windows 2000 and Windows NT) is the single quorum cluster model in which multiple nodes of a cluster share a single quorum resource. In this model, all nodes communicate with each other across a local interconnect, and all nodes share a common disk array (in a SAN or a SCSI enclosure).

Windows Server 2003 also introduces the concept of a single node cluster, which is a cluster that is comprised of a single node or server. For obvious reasons, a single node cluster runs host cluster resources, but the cluster resources cannot fail-over to anything.

Then there is the geographic cluster or so-called "geo-cluster" in which the nodes that comprise the cluster are separated over a geographic divide. A wide area network usually separates the nodes and the geo-cluster nodes can be in different buildings or even across the country. They don’t share storage or a quorum.

The central repository of data in a cluster is the so-called quorum resource. You can think of the quorum as the brain

center of the cluster. The idea of a cluster is to provide system or server redundancy. In other words, when a server in the cluster fails, the cluster service is able to transfer operations to a healthy node. This is called failover. The quorum resource data is persistent and the quorum must survive node failure in the cluster or the resources cannot fail to the healthy node and start up.

This is why in a traditional, single quorum resource cluster, the quorum cannot be mounted into any single device on the node of the cluster unless the cluster can gain exclusive access to the device (and unless it can be moved or transferred upon node failure, which is technically possible even on a local disk resource as we will soon see). There are two exceptions to this rule: the single node cluster and the so-called geocluster, a concept in clustering now possible with Windows Server 2003.

Each of the cluster models discussed employs a different quorum resource type. Table 6.1 discusses the models.

Single Node
Of particular interest is the Single Node cluster model in which the quorum resource can be maintained on a storage device on the local node. The idea behind the single node model is novel. With previous versions of the operating system, it was impossible to establish a virtual server, what users attach to, on a cluster comprising only one node. The single node cluster enables this. The Single Node cluster model is illustrated in Figure 6.1. NOTE: This article covers the creation of a single quorum cluster.
You can use the single node cluster for lab testing of applications that have been engineered for clustering. You can also use it to test access to storage devices, quorum resources, and so on. The lab or development work is, thus, used to migrate the cluster-aware application into production as a standard single quorum cluster. It is also possible to simply cluster the single node with other nodes at a later time. The resource groups are in place and all you need to do is configure fail-over policies for the groups.

A single node cluster can also be used to simply provide a virtual server that users connect to. The virtual server service and name, thus, survives hardware failure. Both administrators and clients can see the virtual servers on the network and they do not have to browse a list of actual servers to find file shares.

What happens when the server hosting the single node cluster and the virtual server fail? The Cluster service automatically restarts the various application and dependent resources when the node is repaired. You can also use this service to automatically restart applications that would not otherwise be able to restart themselves.

For example, you can use this model to locate all the file and print resources in your organization on a single computer, establishing separate groups for each department. When clients from one department need to connect to the appropriate file or print share, they can find the share as easily as they would find an actual computer.

You can move the virtual server to a new node and end users never know the physical server behind the virtual server name has been changed. The real NetBIOS name of the server is never used. The downside of this idea is downtime. Moving the virtual server name to a new server requires downtime. Therefore, this is not suitable for a highavailability solution.

Single Quorum Cluster
This cluster model prescribes the quorum resource maintains all cluster configuration data on a single cluster storage device that all nodes have the potential to control. As mentioned earlier, this is the cluster model available in previous versions of Windows. The Single Quorum cluster model is illustrated in Figure 6.2.

Microsoft discounts the perception that the cluster storage device can be a single point of failure and promotes the idea that a Storage Area Network (SAN) where there are often multiple, redundant paths from the cluster nodes to the storage device mitigates in favor of this solution. While not discounting this model, if you study how a SAN is built, you discover there is some truth that a SAN is a single point of failure.

You can indeed have multple paths to the storage device (the "heart" of the SAN) as discussed in Chapters 3, "Storage for Highly Available Systems," and 4, "Highly Available Networks." However, the SAN controller is really nothing more than a server with an operating system that is dedicated to hosting the drive arrays in its enclosures. Unless you have redundant controllers, your SAN will fail if a component in the SAN controller fails. SAN memory can fail, its operating system can hang, the processors can be fried, and so on. Thus, to really elimimate every single point of failure in this model, you really need to have two SANs on the back end. This idea really opens a can of worms. After all, most IT shops do not budget for two SANs for every cluster. The SANs of today have many redundant components within their single footprint (usually a very large footprint) in the data center. To deploy two-mirrored SANs on a cluster is not only a very expensive proposition, but it is technically very difficult to install and manage.

Majority Node Set
As mentioned, geo-cluster nodes can reside on opposite sides of the planet because each node maintains its own copy of the cluster configuration data. The quorum resource in the geo-cluster is called the Majority Node Set resource. Its job

is to ensure the cluster configuration data is kept consistent across the different nodes; it is essentially a mirroring mechanism. The Majority Node Set cluster model is illustrated in Figure 6.3.

The quorum data is transmitted unencrypted over Server Message Block (SMB) file shares from one node to the other. Naturally, the cluster nodes cannot be connected to a common cluster disk array, which is the main idea behind this model.

You can use a majority node set cluster in special situations, and it will likely require special third-party software and hardware offered by your m>Original Equipment Manufacturer (OEM), Independent Software Vendor (ISV), or Independent Hardware Vendor (IHV).

Let’s look at an example. Let’s say we create an 8-node geo-cluster. We could, for example, locate four nodes in one data center, say in Atlanta, and the other 4 nodes in another data center in Phoenix. This can be achieved, and you can still present a single point of access to your clients. At any time a node in the geo-cluster can be taken offline, either intentionally or as a result of failure, and the cluster still remains available.

You can create these clusters without cluster disks. In other words, you can host applications that can failover, but the data the application needs are replicated or mirrored to the quorum data repositories on the other nodes on the cluster. For example, we can use this model with SQL Server to keep a database state up-to-date with log shipping. In Chapter 10, we investigate the particular solutions offered by NSI Software: Double-Take and GeoCluster.

The majority node set is enticing, but there are disadvantages. For starters, if more than half the nodes fail at any one time, then the entire cluster itself fails. When this happens, we say the cluster has lost quorum. This fail-over limitation is in contrast to the Single Quorum cluster model discussed ealier which will not fail until the last node in the cluster fails.

The Quorum Resource

Every cluster requires a resource which is designated as the quorum resource. The idea of the quorum is to provide a place to store configuration data for the cluster. Thus, when a cluster node fails, the quorum lives to service the new active node (or nodes) in the cluster. The quorum essentially maintains the configuration data the cluster needs to recover.

This data in the quorum is saved in the form of recovery logs. These logs store the changes that have been saved in the cluster database. Each node in the cluster depends on the data in the cluster database for con- figuration and state.

A cluster cannot exist without the cluster database. For example, a cluster is created when each node that joins the cluster updates its private copy of the cluster database. When you add a node to the existing cluster, the Cluster service retrieves data from the other active nodes and uses it to expand the cluster. When you create the first node in a cluster, the creation process updates the cluster database with details about the new node. This is discussed in more detail in the section "Clustering" later in this chapter.

The quorum resource is also used by the cluster service to ensure the cluster is composed of an active collection of communicating nodes. If the nodes in the cluster can communicate normally with each other (across the cluster interconnect), then you have a cluster. Like all service databases on the Windows platform, the cluster database and the quorum resource logs can become corrupt. There are procedures to fix these resources and we cover this a little later in this chapter.

When you attempt to create a cluster, the first node in the cluster needs to gain control of the quorum resource. If it cannot see the resource (this quorum), then the cluster installation fails. We show you this later. In addition, a new node is allowed to join a cluster or remain in the cluster only if it can communicate with the node that controls the quorum resource.

Let’s now look at how the quorum resource is used in a two-node cluster, which is the type of cluster we will in the coming chapters. When the first node in the cluster fails, the second node continues to write changes to the cluster database that it has taken control of. When the first node recovers and a fail-back is initiated, then ownership on the cluster database and quorum resource is returned in the fail-back mechanism.

But what if the second node fails before the first is recovered? In such a case, the first node must first update its copy of the cluster database with the changes made by the second node before it failed. It does this using the quorum resource recovery logs.

If the event the interconnect between the nodes fails, then each node automatically assumes the other node has failed. Typically, both nodes then attempt to continue operating as the cluster, and what you now have is a state called split brain sydrome. Imagine both servers succeeded in operating the cluster, you would then have two separate clusters claiming the same virtual server name and competing for the same disk resources. This is not a good condition for a

system to find itself in.

The operating system prevents this scenario with quorum resource ownership. The node that succeeds in gaining control of the quorum resource wins and continues to present the cluster. In other words, whoever controls the "brain" wins. The other node submits, the fail-over completes, and the resources on the failed node are deactivated.

What constitutes a valid quorum resource? The quorum can be any resource that meets the following attributes:
- It can be accessed by a single node that must be able to gain physical control of it and defend the control.
- It must reside on physical storage that can be accessed by any node in the cluster.
- It must be established on the NTFS file system. It is possible to create custom resource types as long as developers meet the arbitration and storage requirements specified in the API exposed by the Microsoft Software Development Kit. Let’s now look at some deployment scenarios.

Deployment Scenarios

Let’s discuss some example deployment schemes, namely the n-node fail-over scheme, the fail-over ring scheme, and the hot-standby server scheme.

In the n-node fail-over scheme you deploy applications that are setup to be moved to a passive node when the primary node on a 2-node cluster fails. In this configuration, you limit the possible owners for each resource group. You will see how we do this in Part II of this book.

Let’s consider the so-called N+I hot-standby server scheme. Here you reduce the overhead of the 2-node failover by adding a "spare" node (one for each cluster pair) to the cluster. This provides a so-called "hotstandby" server that is part and parcel of the cluster and equally capable of running the applications from each node pair in the event of a failure of both of the other nodes. Both of these solutions are called active/passive clusters—n-node and n-node+1 (or N+1).

As you create the N+1 mode cluster, you will discover it is a simple matter to configure as the spare node. How you use a combination of the preferred owners lists and the possible owners list depends on your application. You typically set the preferred node to the node that the application runs on by default; and you set the possible owners for a given resource group to the preferred node and the spare node.

Then there is the concept of a Failover Ring. Here you set up each node in the cluster to run an application instance. Let’s assume we have an instance of SQL Server on each node of the cluster. In the event of a failure, the SQL Server on the failed node is moved to the next node in sequence. Actually, an instance of SQL Server is installed on every server. Fail-over simply activates the SQL Server instance, and it takes control of the databases stored on the SAN or SCSI array. We call this the Active-Active cluster.

You can also allow the server cluster to choose the failover node at random. You can do this with large clusters and you’ll just not define a preferred owners list for the resource groups. In other words, each resource group that has an empty preferred owners list is failed over to any node in random fashion in the event that the node currently hosting that group fails.

We will leave the clustering subject now and return to the creation of the infrastructure to support our clusters.

FOREST CREATION PROCESS

Assuming we are starting from scratch, a so-called green fields site, we must first create a forest into which your systems will be integrated. This is called the forest creation process. This is the process that starts with the provisioning of an installation server through the creation of the forest.

Installation of Support Server

The first server installed in your network, you may be surprised to know, is not a domain controller. It is not even a new server. It should be a nonservice server installed with either Windows 2000 or Windows Server 2003 in its own workgroup. This server is placed on the lab or future production subnet, initially as a workgroup server, and exposes a number of shares used for accessing operating systems, tools, software, utilities, and patches. The idea is to provide a secure, closed network that does not have access to the outside network that might likely contaminate your implementation. The support server is used for patches, access to tools, resource kits, and so on.

It is critical at this stage that none of your new servers "touch" the Internet or are exposed to the outside. It is very easy to "catch" a virus and not notice it until the entire forest is created and all your servers start croaking.

This server is eventually joined to the network as a temporary Windows Update Server (WUS). The server may also function as a temporary DHCP server. To configure the support server, do as follows:
- Log on to support server as Administrator while this server is still in the lab.

- Create a folder named C:\ADSTUFF and share as ADSTUFF (actually any name will do).

- Create a folder named C:\ADSTUFF\Adminpak\.

- Create a folder named C:\ADSTUFF\Support\.

- Create a folder named C:\ADSTUFF\Exchange Tools\.

- Create a folder named C:\ADSTUFF\SQL Server Tools\.

- Create a folder named C:\ADSTUFF\QA documents\.

- Crate a folder named C:\ADSTUFF\Scripts\.

- Create a folder named C:\ADSTUFF\RKTools\.

- Copy needed tools, MSI files, scripts, data, packages, and so on to these folders.

- Install anti-virus services and make sure the support server has the latest anti-virus DAT files and is performing the correct scans of its file system.

- Install Software Update Services Software Update Services on the support server.

- If needed, create distribution folders for operating system images. You can call the shares STDINST for the Windows Server 2003 Standard Edition or ENTINST for the Windows Server 2003 Enterprise Edition operating system.

- If needed, create the distribution folders named C:\WEBINST and share as WEBINST for the Windows Server 2003 Web Edition operating system.

- If needed, create the distribution folders named C:\XPINST and share as XPINST for the XP workstation images.

- Create distribution shares (for example, C:\..\I386) and copy installation sub-folders and files to the distribution shares (see Table 6.2). This process can be done automatically using the Setup manager utility (setupmgr.exe) on the operating system CD’s Support, Tools folder. Setupmgr is found in the deploy.cab file.

- Configure Software Update Services on the installation.

- Validate this server (including last scan for anti-virus).With the support server in place on your isolated network, you can begin working on the creation of the forest and the domains, accessing your server for support materials as if it were your own mini Microsoft.com Web site.

INSTALLATION

Upon installation of the support server to the isolated network, proceed to the installation procedures.
- Rack and stack your servers in the production racks or on the data center floor with access to the isolated network.

- Power up the support/installation server.

- Log on to the installation server as Administrator on the isolated subnet.

- Reset the Administrator password.

- Change the IP configuration to statically assigned addressing.

- Assign the IP address of 10.10.20.23 (on a /22 subnet where the 10.10.20.0 space is reserved for the data center servers).

- Assign the same IP address as the gateway and DNS.

- Install DHCP and configure the new scope for the subnet (see Table 6.3).At this point, the installation and provisioning of the support server is complete.

After the DHCP server has been installed on the support server, reserve IP addresses for the root domain controllers. This ensures the root DCs obtain the correct IP addresses as soon as their IP configuration is changed from static addressing to DHCP-assigned.

One note to consider before we move on: The subnet we have used here will provide sufficient addresses to meet the needs of a high availabilty network. Don’t short change yourself with IP addresses. You should be good to go with a subnet that provides more than a thousand IP addresses. High availability systems use a lot of IP addresses. A typical two-node cluster should be allocated a block of about 24 addresses for future expansion.

INSTALLATION OF ROOT DOMAIN

This section covers the promotion of the root domain controllers. By promoting root domain controllers, we are, in fact, creating the forest in which all future high-availability systems will be installed (see Chapter 5 for the discussion of the dual domain [root-and-child] model). The prerequisite to this process is installation of the operating system (Windows Server 2003, Standard Edition) to the domain controller computers on a RAID-1 array. See Chapter 4 for instructions on the configuration of RAID-1 on this server. The servers should be configured for second and third RAID-5 arrays as required.

It is critical this process completes and proceeds as described herein. Deviation from the process or shortcuts may render the root domain useless and it will have to be rebuilt. The updating of the domain controller servers with the required software updates and security patches can take place after promotion, QA, and validation. (See Chapter 5 for the overall architecture this implementation supports.)

Process

Name the Root Domain DCs. Upon completion of the server installations, the root domain controllers will be given miscellaneous names, and they will be a member of the workgroup setup on the support server. Change the names of the root domain controllers to the names provided in your Active Directory Architecture (discussed in Chapter 5). For the corporate hub, the server names we use here are HQRDC01 and HQRDC02 (for later implementation).

It is important to remember to rename the servers to their DC names prior to running DC promo. The names cannot be changed after promotion of these servers to domain controllers, and they have to be destroyed if the names are incorrect. Do not change the workgroup when changing the names.

Configure TCP/IP on HQRDC01. Log on as Administrator to the server designated to become the root DC (HQRDC01). Open the TCP/IP properties of the network interface card (NIC), and enter the parameters listed in Table 6.4.

Configure TCP/IP on HQRDC02. Log on as Administrator to the server designated to become the root DC (RDC02).

Open the TCP/IP properties of the NIC, and enter the parameters listed in Table 6.5.

To install DNS, do as follows:
- Log on as Administrator to the server designated to become the root DC (HQRDC01) and install DNS on this server. This is achieved by opening Control Panel, Add or Remove Programs, and Add/Remove Windows Components. This launches the Windows Components Wizard.

- Select Networking Services in the wizard and click the Details button. In the Networking Services dialog box, check the option to install Domain Name System (DNS).

- Complete the procedures and, when prompted by the installation procedure for the Windows Server operating system CD, provide a CD or browse to the I386 folder under the STDINST share (the source for OS installation files) on the installation or support server.

- Complete the process to install DNS on the server. Repeat the process for all hub root domain controllers.Now you can create the Forest Root Zone on HQRDC01. To create the forest root zone, perform the following steps (note: this process is not repeated on HQRDC02 or any other root server destined to become a DC):
- Start DNS and right-click on the HQRDC01 icon.

- Select New Zone. The New Zone Wizard launches. Click Next.

- Select the option to create a Primary zone and click Next.

- Select Forward Lookup zone and click Next.

- Enter the domain name (such as MCITY.CTY) as the name of the zone and click Next.

- Keep the default DNS file name (it should be MCITY.CTY. dns) for the zone file name and click Next.

- If prompted for Dynamic Update configuration, choose the option to allow Dynamic Updates. Click Next.

- Complete the process by selecting Finish.Create the Reverse Lookup Zone on HQRDC01. To create the reverse lookup zone for the forest, perform the following steps:
- Open the DNS console and expand the HQRDC01 server icon.

- Select Reverse Lookup Zones and click on New Zone. The New Zone Wizard launches.

- Select options for a Primary non-integrated zone and click Next.

- Enter the IP address range for the zone; this is the 10.10.20.X network.

- Click Next and select the options to enable dynamic update.

- Complete the process by selecting Finish.Create the Forest Root Domain Controller on HQRDC01. To create the forest root domain, perform the following steps:
- Click Start, Run, and type DCPROMO on HQRDC01.

- Choose the options for creating a root domain controller in a new forest.

- Choose the root domain name as the full DNS name for the new domain (MCITY.CTY).

- Accept the default NetBIOS name for the domain.

- Choose the default path for the SYSVOL folder on the RAID-5 array. However, the drive letter should point to the RAID-5 array on (D, E, or F) and not C:\ (for example E:\Windows\…). Choose the path options provided for the NTDS Active Directory database and its log files, changing only the drive letters to point to the RAID 5.

- Accept permissions compatible with Windows 2000 and Windows Server 2003.

- Enter the Directory Services Restore Mode Administrator password (this should be a complex password, choose something like 4NTDS@mcity), ignoring the quotes. (Remember the server’s local Administrator password becomes the password required to log on to the DC after promotion.)Review the settings, and click Finish to begin the process. Restart the server when prompted.

Enable Active Directory Integration of the Forest Root Zone and the Reverse Lookup Zone. To enable AD integration for the root zone, do as follows:

- Open the DNS console and expand the root server HQRDC01 icon.

- Expand the Forward Lookup Zones folder and select the MCITY.CTY zone. Right-click this zone and select Properties.

- The Properties dialog box for MCITY opens. On the General tab, select the Change button on the Type option. The Change Zone Type dialog box launches.

- Select the option to change the zone to Active Directory Integrated and click OK.Perform the same procedure on the Reverse Lookup Zone folder. Verify HQRDC01 Name Registration. To verify name registration, perform the following actions:

- Open the DNS console and expand the root server HQRDC01 icon.

- Expand the Forward Lookup Zones folder and select the MCITY.CTY zone.

- Verify whether _msdcs, _sites, _tcp, and _udp sub-domains are registered under MCITY.CTY.

- If these sub-domains are not registered, then start a command prompt and type NET STOP NETLOGON. Wait for the service to stop and then type NET START NETLOGON.

- Repeat steps 1 through 3 to verify the registration.

- Verify the Reverse Lookup Zone has replicated.Verify DNS name resolution on HQRDC02. Before HQRDC02 can be promoted as a root DC, DNS first must be verified. This can be achieved as follows: 1. Log on to HQRDC02 as the Administrator.

2. Open the command prompt and type NSLOOKUP MCITY. CTY and press Enter. You should see the following result:
C:\>nslookup MCITY.CTY
Server: HQRDC01.MCITY.CTY
Address: 10.10.20.21
Name: MCITY.CTY
Address: 10.10.20.21If you do not see this, check to see whether the IP settings on HQRDC02 are correct. It should have HQRDC01 (10.10.20.21) as its preferred DNS server. Do not proceed with DCPROMO of HQRDC02 until DNS is working properly.

Perform DCPROMO on the server HQRDC02. To create the second domain controller, perform the following steps:

- Click Start, Run, and type DCPROMO on HQRDC02.

- Choose the options for creating an additional domain controller for an existing domain and click Next.

- You are prompted for access to the root domain. Choose the Administrator account because this account has Enterprise Administrator credentials. See the previous steps for account and password information.

- Choose the default path for the SYSVOL folder on the RAID-5 array. However, the drive letter should point to the RAID-5 array on (D, E, or F) and not C:\. Choose the path options provided for the NTDS Active Directory database and its log files, changing only the drive letters to point to the RAID 5 volume as previously mentioned (see Chapter 4).

- Enter the Directory Services Restore Mode Administrator password for this server (this should be a complex password; choose 4NTDS@MCITY). DCs can and should have the same Directory Services Restore Mode Administrator password to simplify administration.Review the settings and then click Finish to begin the process. Restart the server when prompted. Verify HQRDC02 Name Registration. To verify name registration, perform the following actions:
- Open the DNS console and expand the root server HQRDC02 icon.

- Expand the Forward Lookup Zones folder and select the MCITY.CTY zone.

- Verify whether _msdcs, _sites, _tcp, and _udp sub-domains are registered under MCITY.CTY.

- If the sub-domains are not registered, then start a command prompt and type NET STOP NETLOGON. Wait for the service to stop and then type NET START NETLOGON.

- Repeat steps 1 through 3 to verify the registration.Verify the Reverse Lookup Zone has replicated. Update the Preferred DNS Parameters on HQRDC01. Log on to HQRDC01 and open the TCP/IP properties for the NIC. Change the preferred DNS server from 10.10.20.21 to 10.10.20.24.

Create Automated System Recovery (ASR) media for the domain controllers. The creation of the root domain and promotion of the first domain controllers is now complete. System recovery using ASR media now must be performed on the domain controllers. After the ASR disks have been created, you can start the QA discussed in the next section.

QUALITY ASSURANCE

QA and validation must be performed before continuing further. QA can be achieved by following these steps:
- Join a clean Windows XP SP1, SP2, or higher workstation to the root domain. Remember to follow the naming convention for the workstation according to Active Directory Architecture.

- Install the WSO3 support tools on the workstation. The tools can be accessed from the ADSTUFF\SHQPORT\TOOLS share on the installation server. Install the tools to the default path on the C: drive.

- Install the ADMINPAK on the workstation. This installs management tools, such as DSA.MSC, to the workstation. The tools can be accessed from the ADSTUFF\ADMINPAK share on the installation server. Install the tools to the default path on the C: drive.

- Install the Resource Kit tools to the workstation. This installs tools, such as DNSDIAG, DCDIAG, and DSQUERY to the workstation. The tools can be accessed from the ADSTUFF\ RESKIT share on the installation server. Install the tools to the default path on the C: drive.

- Open a command console and run DCDIAG /s: /a /f /ferr. Perform the DCDIAG against both HQRDC01 and HQRDC02. The data generated by DCDIAG is piped to the default log file location on the workstation.

- Perform DCDIAG several times a day during the installation.

- Open the replication monitor and check that replication is occurring without errors between the domain controllers.Finally, you can run DSQUERY against the domain controllers to see that all FSMO roles are intact (the roles are moved later on in the implementation). Much of this manual diagnostics and QA can be left to Microsoft Operations Manager (MOM) to handle. Without MOM, QA can become something of an endurance during the life of a long project to stand-up a high availability infrastructure.