

Clusters help you run a highly available SQL Server, but setup requires a whole bunch of steps. We walk you through the process, and help you make the right configuration decisions *before* you click on Install.

How to build a SQL Server cluster *(and keep your hair, too)*



INSIDE

- 03 Cluster basics
- 04 Networking concerns
- 07 Configuring storage
- 09 Installing the cluster
- 13 The final setup
- 14 Live talk with Geoff Hiten

How to build a SQL Server cluster

By Geoff Hiten

WHEN A COMPANY realizes that its databases can be single points of failure for a major system or business process, it may look to clustering to increase availability of the database's host computers. As a result, a DBA could find himself as the first person in the organization given the responsibility for a server cluster. Who, *me*?

Building a SQL Server cluster can be an intimidating experience the first time you tackle the task. This guide walks you through the necessary steps for creating a working database cluster.

As with any high-availability component, the real work in clustering happens long before an install disk is inserted into a server. Careful planning and preparation makes the actual install process no more complex than a standalone SQL Server build. However, you should be comfortable and familiar with the standalone SQL installation process before you attempt to build a cluster.

Let's begin by examining a cluster and learning how it is different from a standalone SQL Server.

“A DBA could find himself as the first person in the organization given the responsibility for a server cluster. Who, *me*?”



A Windows Cluster consists of a minimum of three pieces...

CLUSTER BASICS

A Windows cluster consists of a minimum of three pieces: two or more host nodes (servers) and some type of multi-connected storage system. It is possible to build a single-node cluster, but typically a single-node cluster is a transition system to a multi-node cluster.

Make sure you're working with the right level of the operating system. Windows Server 2003 supports up to eight nodes on a cluster. SQL 2000 Enterprise Edition supports up to four nodes in a cluster, while SQL 2005 offers two options. SQL 2005 Standard Edition supports up to two nodes, while SQL 2005 Enterprise Edition can support the operating sys-

tem node limit. SQL Server can install one or more instances on a single cluster in a variety of configurations.

Give a lot of thought to hardware; hardware selection is essential. Since clustering adds an extra layer of complexity, it is imperative that the hardware meets very specific standards. To build a working cluster, choose hardware that is identified in the Windows Hardware Catalog as cluster-certified.

Don't skimp or cut corners. The hardware must be purchased as a cluster from the vendor. You cannot put it together from existing non-clustered hardware. Again, this is to ensure that the system operates as a stable database platform. While it is perfectly acceptable to build a test and training cluster from left-over components, it is a very bad idea to run any critical business process on such a system.

The core of a cluster is the multi-connected storage system. This component ranges from a simple hot-swap drive enclosure, costing a few thousand dollars, up to multi-million dollar storage area network (SAN) systems. What makes this type of hardware essential to clustering is that the enclosure is separate and independent from any individual host node. That allows the cluster to function with one or more nodes offline.

Windows clustering supports a maximum of two nodes on a SCSI storage system. This is because of the inherent limitations of a SCSI bus, not because of any Windows clustering limitation. Most commercial controllers only allow two address selections, thus limiting the system to a maximum of two host controllers. There are also specific startup processes to follow when using SCSI stor-

“Give a lot of thought to hardware; hardware selection is essential.”

age systems beyond the normal cluster startup procedure.

NETWORKING CONCERNS

Once you select an appropriate hardware platform for your cluster, you can start to plan the actual installation. First, you need to choose names and IP addresses for the systems. You need one name for each computer in the cluster, one



Always set all NICs in your cluster to the maximum speed supported by the network hardware. Do not allow the NICs to auto-detect speed. The auto-detect process can introduce additional latency into the communications process, introducing system instability and possibly causing a false cluster failure.

IP ADDRESS SELECTIONS FOR A SAMPLE CLUSTER

NAME	IP ADDRESS	ROLE	CLASS	NETWORK	SUBNET	GATEWAY
East	10.10.1.20	Host Node	Physical	Public	255.255.255.0	10.10.1.1
East	10.200.1.20	Host Node	Physical	Private	255.255.255.0	None
West	10.10.1.30	Host Node	Physical	Public	255.255.255.0	10.10.1.1
West	10.200.1.30	Host Node	Physical	Private	255.255.255.0	None
TheCluster	10.10.1.40	Cluster	Virtual	Public	255.255.255.0	10.10.1.1
TheSQL	10.10.1.50	SQL	Virtual	Public	255.255.255.0	10.10.1.1

name for the cluster itself and a different name for the SQL Server. Each of these names appears on the network as an independent computer system. Nodes are actual physical machines, while the cluster and SQL names are referred to as “virtual servers.”

Each node needs two physical network cards, each with a unique IP address. One card is used for public connections and the other is for private, cluster-only communications. In this context, “public” means any computer connecting from

outside the cluster.

In the table example, East and West are the node names: TheCluster is the Cluster name, and TheSQL is the SQL Server name. The table shows the IP address selections for a sample cluster.

Notice that the private network has no gateway. Disable NETBIOS communication on the private NIC. Under network properties, make sure TCP/IP is the only selected protocol on the private NIC. For a two-node cluster, a simple crossover cable is the best solution.

A System of Excellence

The Kenton County School District



An IT infrastructure based on Dell platforms keeps the Kenton County School District at the head of the class

The Kenton County School District is a big district with an even bigger reputation for excellence. Located in northern Kentucky, it serves over 13,000 students throughout 21 schools and has reaped many honors, including being selected as a spotlight school district by the U.S. Department of Education in 1999 and 2000, and receiving the national “What Parents Want” award for 12 years in a row.

Maintaining such a high level of excellence in academics requires excellent IT. In order to support classroom activities, the district’s administrative functions, transportation, food service, and finance must run reliably and efficiently. Also, critical information such as student records, class schedules, and other data must be collected, maintained, and made readily available to teachers and other staff who require it. “We need an IT infrastructure that can support all of our departments,” says Vicki Fields, district technology coordinator for the Kenton County School District. “Plus, we want our teachers to have instant access to student information so they can spend their time teaching instead of chasing down student records.”

“Without question, Dell keeps our applications up and running better than any other vendor with whom I have worked in the past.”

— **Vicki Fields**, District Technology Coordinator
Kenton County School District

A System of Excellence

The Kenton County School District



Kenton County looks for a new platform

A few years ago, the district wanted to improve teacher and staff access to student information and other types of data. “We have information distributed across a variety of delivery systems,” says Fields. “We wanted to provide comprehensive access to that information to better facilitate teaching, administrative functions, and decision making throughout the district.”

Also, Fields wanted to standardize on a single hardware vendor. “We have many different applications throughout the district, and we wanted to find one vendor that could support them all,” says Fields. “In addition, we wanted a vendor with excellent support and services that would be with us over the long haul.”

Kenton County decides on Dell

After evaluating several vendors, Fields was impressed with the flexibility of Dell offerings. “Other off-the-shelf types of technologies didn’t have the flexibility that we needed,” says Fields. “When I approached Dell, their willingness to comply with our requirements was a key selling point. Dell told us we could set up our system however we wanted, and that was just what we needed to hear.”

To view the entire story, go to www.dell.com/sql

Some clusters use additional network adapters for general public or limited public access. These may be teamed to a single IP address or segmented using a connection-specific domain suffix and a different IP address and subnet. *Teaming* is often used to increase bandwidth and availability, while *segmentation* dedicates a

portion of network bandwidth for a specific set of clients or to isolate bandwidth-intensive operations, such as replication or backup.

Do not set up additional NICs for teaming during the initial installation. Segmented NICs should be selected and entered during the cluster installation process. With multiple NICs in a system, I find it helpful to rename the network connections on all the host nodes to match their function, such as “Public,” “Heartbeat,” “Backup” or “Private.” That helps in both installation and troubleshooting.

CONFIGURING STORAGE

During the build process, the multi-connected storage unit should not be visible to the host computers. For small SCSI storage units, powering them down is adequate. With a SAN, you can create the logical unit numbers (LUNs), but do not

grant access for them to any host node. If you are booting from the SAN (which I do not recommend), present only the boot partition LUNs to the host nodes while you are configuring the servers.

Whew. That takes care of hardware. We can finally begin to address the software.

The first actual installation for the cluster should be the operating system for each host node. Install Windows 2003 Enterprise Edition exactly as you would for a standalone SQL Server. Apply all current service packs and patches as you would with any other database server.

Once all nodes are completely installed and operational, including all network settings on the physical connections, it is time to configure the storage system. For this portion, you’ll need to have only one node controlling the storage unit.

“Teaming is often used to increase bandwidth and availability, while segmentation dedicates a portion of network bandwidth for a specific set of clients or to isolate bandwidth-intensive operations, such as replication or backup.”

The easiest method is to power down all nodes, except the one on which you are working.

Note that you cannot divide a disk at the operating system level for clustering, only at the physical disk. A disk with multiple partitions is treated as a single disk resource for clustering.

For the example cluster, we need three sharable drives on the shared storage array. For a SAN, these are individual LUNs; for a SCSI array, typically they are RAID containers. We also need two clustered disk re-

sources: one for the cluster and one for The SQL Cluster itself.

Best practices for SQL recommend placing data and transaction logs on different physical media. You can create additional disk resources for your cluster and have SQL install them all together. Microsoft Distributed Transaction Coordinator can reside on the cluster disk in most cases, but in high-volume environments, it should be on its own disk resource.

Follow the instructions for your specific storage system to create LUNs or RAID containers to present to the host nodes. Some SANs may come with LUNs created according to recommendations from a storage architect. They may also have specific partitioning procedures to maximize performance and availability. Follow the vendor's recommendations exactly for a clustered installation; you'll be sorry if you don't.

Once the disks are created and visible to the operating system, use the regular Windows Disk Manager to partition and format the disks. Assign the disk letters you intend to use for the clustered system to the partitions at this time. It is common to use the letter "Q" for the Cluster disk (often referred to as the *Quorum disk*).

At this time, power up the other host nodes to verify disk visibility. Since there is no cluster to arbitrate ownership, you can write to each disk from any host node. Windows uses an automatic disk cache that assumes exclusive ownership of local disk resources. This will lead to disk corruption if you try to use a multi-connected storage system outside the control of a cluster.

You only have a couple more decisions to make before starting the actual cluster installation. You need two service

“Follow the storage vendor’s recommendations exactly for a clustered installation; you’ll be sorry if you don’t.”

accounts: one for the cluster and one for the SQL service. All accounts must be domain-level accounts. A domain (active directory or NT4.0) is necessary to provide a common security context. The cluster service accounts on the nodes can work together as a single logical entity in a domain.

I don't advise you to use the same account for the cluster service and the SQL Server. While it is possible to use separate accounts for the SQL Service and the SQL Agent, most installations use a single account for both. After you create the accounts in the domain, add



Note that the cluster shown here is for a single node only. Once the cluster is working, you can add more nodes.

the cluster service account to the local administrator's group on each node. You can do this to the SQL Service account as well.

INSTALLING THE CLUSTER

Now that we have all the building blocks in place, you can start the actual cluster installation. You may need the OS disk for additional files, so find it before you settle down.

Start the cluster installation from the cluster administrator tool. You'll find this tool installed under Administrative Tools on all Windows Server 2003 Enterprise Edition systems. **FILE | NEW | CLUSTER** invokes the cluster setup wizard from the Cluster Administrator tool.

The Create New Cluster wizard asks for the cluster name and the name of the first node in the cluster. Fill in those fields; the wizard will validate your

cluster configuration. The validation step generates a detailed log that gives excellent guidance for fixing any problem you encounter. (A Re-analyze button reruns the test after you fix it.) When the completion bar is solid green, you can click Next to continue installing the cluster.

You then enter the Cluster Public IP address and the cluster service account information, which you chose in our earlier discussion. The cluster wizard displays a summary page before it begins.

The most important element to check here is to verify that the Quorum drive is the correct disk. If a cluster-eligible disk has drive letter Q:, the cluster wizard will use it for a quorum.

Clicking "Next" launches the cluster build process. It re-runs the cluster verification test and then performs the cluster build. You may be surprised at how

rapidly the cluster build process runs. In a very few moments, you will have a working single-node cluster.

After the first node of the cluster is configured, the Cluster Administrator tool automatically connects to the cluster and displays its components. Take a few moments to explore your cluster's groups and resources.

NEXT, WE ADD additional nodes to the cluster. Right-click on the cluster name and select **NEW | NODE**. You can add multiple nodes at this time. If you used LUN masking from a SAN, be sure all your nodes have been granted access to all cluster-wide disks.

Enter the node name(s) and click Next. Once again, the cluster verification wizard does its thing, making sure everything matches up with the existing node and any new nodes. In the

next screen, the wizard asks for the cluster service password and begins adding the extra node. This will likely take longer than did the initial cluster build process. Once again, a green status bar indicates complete success.

This is an excellent time to make sure all the groups and resources can be moved to any cluster node. The right-click menu option "Move Group" shifts resource groups between host nodes without incrementing the group failure count.

Many clusters that do not have much distributed transac-



You may get a warning if you uninstalled a cluster from any of the new host nodes without reinstalling the OS. That warning is not critical and does not affect the new cluster.

tion activity have the MSDTC (Microsoft Distributed Transaction Coordinator) installed in the cluster resource group. You can have a cluster without a MSDTC resource, but there are often undesirable side effects, including legacy client access problems. Use the cluster tool to create a new resource of type "Distributed Transaction Coordinator," add it to the quorum group and make it dependent on the Quorum disk. Create a separate group with a disk resource, network name and IP address if you expect a high volume of distributed transactions.

If you are teaming using NIC teaming, now is the time to switch from standalone to teamed NICs.

As I noted early in this article, most of the work in creating a SQL cluster happens long before starting the install application. So far, the entire exercise

created a stable platform for hosting a SQL cluster. Now we can leverage that platform to run a highly available SQL Server. After one more final configuration tweak, we will be ready to install the SQL Server itself.

THE LAST STEP in setting up the cluster is to clean up any installation leftovers. Make *absolutely* sure there are no offline

resources in your cluster. If you have an offline resource, the SQL installer will crash before doing any meaningful work.

In our example, we use only one disk for our SQL resource, but if you are building a cluster with multiple disks, place them all in the same resource group. Delete empty resource groups that may be left over from earlier steps.

Start by verifying that the Windows Task Scheduler is enabled and running on all cluster nodes. Log out of all cluster nodes, including remote consoles, except for the node that currently owns the resource group with the soon-to-be SQL Disk Resources. Log into the physical console of that node or use the `/console` option if you are using RDP.

Use the SQL Server 2005 install disk appropriate for your hardware platform and operating system. The install process

for a two-node cluster is exactly the same as it is for an eight-node cluster. It is also similar to the standalone install process: The installation begins with a click-through license and the pre-install component installation, followed by the standard pre-installation system check. But that's where the installation diverges from the standalone SQL installer sequence: The SQL installer checks the local computer and checks all cluster nodes.

After filling out the standard registration dialog box, you are shown the most critical dialog of the SQL Cluster installation sequence. Here is where you choose the actual components of the SQL Server to install. For this example, we install SQL Server Database Services and the Workstation tools. If you select SQL Server Database Services while on an operating cluster node, you can select the

“After filling out the standard registration dialog box, you are shown the most critical dialog of the SQL Cluster installation sequence. Here is where you choose the actual components of the SQL Server to install.”

Create a SQL Server Failover Cluster option. Notice that the workstation tools installation is a separate option and is not clustered.

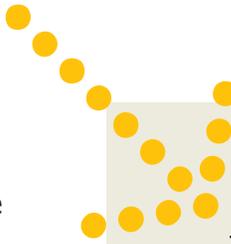
The installer only creates the SQL tools on the local computer. You must install the tool set separately on each cluster node. Use the “Advanced” button to fine-tune which sub-components to install, but I do not recommend altering the SQL default installation on a cluster. You can also create an Analysis Services cluster instance, but that is beyond the scope of this article.

You then choose whether to install a default instance or a named SQL instance. I prefer to install clustered SQL Servers as Named Instances; doing so avoids naming confusions if you have more than one instance on a cluster.

You register and connect to the SQL Server as Virtu-

alServerName\InstanceName when you use a default instance. While the two names can be different, I prefer to follow the SQL 2000 naming convention and keep them identical. For our example, we have the Named Instance: TheSQL\TheSQL.

THE NEXT PIECE of information to provide is the network configuration. Select the network connection based on its name, enter the corresponding IP address for your Virtual Server and add it to the list. You do not need to add a virtual IP to the heartbeat network — nor do you want to do so. Note that you can only add one address per network connection. The subnet mask and gateway are inherited from the underlying physical network connection and you cannot change it here. Again, preparation is everything in a cluster!



You do not need to add a virtual IP to the heartbeat network — nor do you want to do so.

The next dialog box allows you to select which nodes can run this Virtual SQL Server. After that, you enter the installation account password, which must be a member of the local administrator’s group on each server. Then, you specify the service accounts and passwords for the SQL Server services.

The next dialog box is, again, unique to a clustered installation; it’s a best-practice recommendation that is now enforced by the SQL Installer. Under previous versions of SQL Server,

the SQL Server service account needed to be a local administrator on all nodes. However, since that violates the “least privileges” security guideline, permissions are now granted to a domain-level group that has the appropriate service account as its only member. You can use the same group for all three services or set them up independently. This is helpful if you ever need to change the SQL Service account; simply add the new account to the group to set all the relevant permissions.

In the next two dialog boxes, we are back to the normal install sequence: Security mode (with sa password) and collation selection work in exactly the same manner as in a stand-

alone SQL install. Then you can choose to send error or feature usage information to Microsoft for analysis and for future SQL improvement.

THE FINAL SETUP:

The “Ready to Install” summary box repeats the top-level components you chose back at the beginning. It isn’t a detailed description of your cluster configuration choices. Think of it as more of an “are you sure?” page. Notice that there is no “Next” button; we are ready to “Install.”

This is a good time to get a cup of coffee and watch the blinky lights. The SQL Installer runs its list of components and shows the progress of each item. All SQL 2005 installations take some time, and clusters are no exception. If anything, they take a bit longer than a standalone installation. Do not be alarmed if you have to wait

as long as 30 to 60 minutes, especially if you included additional SQL components, such as Integration Services, or if you are using older, slower hardware to build a test platform.

GENTLEMEN, START YOUR CLUSTERS

Congratulations! You are now the proud owner, keeper, feeder, and first-person-on-call for a SQL Server cluster. From here on out, except for actual cluster repair or troubleshooting tasks, you can treat this as you would any standalone server, at least from an application perspective. You simply have a second server constantly monitoring your primary system, ready to take over nearly instantly. Instead of the dreaded “Server Down” call, you will get a “Cluster Failed Over” notification. If you did everything else right, the failure may not be noticed outside your team.

“This is a good time to get a cup of coffee and watch the blinky lights.”

ABOUT THE AUTHOR

Microsoft MVP Geoff Hiten has been working with SQL Server since 1993. In this podcast, he talks with SearchSQLServer.com's Heidi Sweeney about the benefits of clustering, how to get started and key issues to consider before you click "Install." Hiten has a bachelor of science in Computer Science from the engineering department of the University of Alabama in Tuscaloosa. He specializes in high availability and high throughput transactional solutions.

**Live talk with Geoff Hiten**

Starting points for building a SQL Server cluster

http://media.techtarget.com/audioCast/WINDOWS/SearchSQLServer_Podcast_Clustering_2006-11-27.mp3

The real work in clustering happens long before an install disk is inserted into a server.

Additional Resources from Dell

- **System of Excellence: IT Infrastructure Keeps the Kenton School System at the Head of the Class**
www.dell.com/sql
- **SQL Server 2005: Preparing for a Smooth Upgrade**
<http://www.dell.com/downloads/global/power/ps1q06-20060126-Microsoft.pdf>
- **Maximizing SQL Server Performance**
<http://www.dell.com/downloads/global/power/ps4q05-20050272-Symantec.pdf>
- **The Scalable Enterprise Technology Center**
<http://www.dell.com/content/topics/global.aspx/power/en/setc?c=us&cs=555&l=en&s=biz>
- **Microsoft SQL Server 2005 Virtualization**
<http://www.dell.com/downloads/global/power/ps4q06-20060405-Muirhead.pdf>
- **The Definitive Guide to Scaling Out SQL Server 2005**
http://www.dell.com/content/topics/global.aspx/alliances/en/ebook_landing?c=us&cs=555&l=en&s=biz