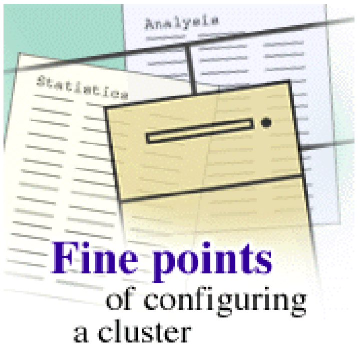
Notes.net

Iris Today



Fine points of configuring a cluster

by
Michael
Kistler

Level: Advanced
Works with: Domino 4.6
Updated: 06/01/1998

There have been a number of *Iris Today* articles touting the high availability and scalability features of Domino clusters (these include "Workload balancing with Domino clusters" and "Lotus Domino Advanced Services: High Availability Powered by Notes"), so by now you must be just itching to set up a cluster and put it to work. There are a number of sources of good information on the basic procedures for setting up a cluster. In addition to the Domino Administration Help, you can find good advice on configuring Domino clusters in the IBM Redbooks. There are two new IBM Redbooks in the works, which will be available in the next few months, that are dedicated to Domino clusters. You can order all of these Redbooks from IBM, or you can view them online. You should also check out the Domino IT Central Cluster Zone for more information on Domino clusters, and the Lotus Technical Learning Center tutorial on Advanced Services (clusters and partitioned servers).

Given the many good sources of information on the basics of cluster configuration, this article covers some of the finer points about how to set up your Domino cluster. You'll learn how to:

- Configure your network for clusters
- Determine how many servers to put in a cluster
- Set up replication for the cluster (both cluster replication and standard replication)
- Set up mail for a cluster (mail routing, directory assistance, and shared mail)
- Set up database access control lists (ACLs) for the cluster

Configuring your network

One of the first things to consider when setting up a cluster is the network topology of the servers. Specifically this includes:

- General network configuration issues
- Using a WAN to connect the servers in your cluster
- The network protocols you can use for communications between servers in a cluster
- How to configure your named networks
- The benefits of using a private LAN for intra-cluster communication

General Network Configuration Issues: Lotus recommends that you connect all cluster members using a high-speed local area network (LAN). This recommendation applies to the network used for communication between the cluster members. Clients are not restricted to a LAN when accessing a cluster. Don't be intimidated by the words "high-speed" -- 10 Megabit Ethernet is sufficient in most cases. This recommendation is meant to discourage you from connecting cluster members on a wide-area network (WAN).

WAN vs LAN: Lotus also states that the servers in a cluster should all be connected on a LAN. Even so, a number of customers do use WAN connections in between servers in a cluster, and are quite satisfied with the reliability and performance of this configuration. This is most often done because one or more of the cluster members are located in a different city for disaster tolerance. The concern with WAN connections is that WANs typically have higher latency, lower bandwidth, and lower reliability than LAN connections. Domino clusters depend on fast and reliable communications between servers to determine the availability and load on the cluster members and for cluster replication (discussed in more detail below), which keeps databases in tight synchronization across the cluster. Depending on the speed of the WAN, number of cluster members, and the amount of cluster replication traffic between cluster members, it is possible to use WAN connections for one or two cluster members and still have a well behaved cluster. If you decide to try this, be sure you test thoroughly -- your mileage may vary. You should test that the servers can continue to determine the availability and load on the other servers in their cluster, and that cluster replication keeps databases in tight synchronization across all servers in the cluster. You should test both of these during periods of heavy load, and when one or more servers are unavailable.

Network protocols: In addition, Lotus states that you should use TCP/IP for cluster server configurations. I recommend this, particularly for communications between the servers in the cluster, to ensure that you get proper support from Lotus in case you encounter any problems. However in truth, the failover and workload balancing logic was designed and implemented to be independent of the actual network protocol. If you use a protocol other than TCP/IP for client access, failover may not be as smooth or as timely, but it still works properly in most environments. However, remember that Lotus only fully certifies Domino clusters running with TCP/IP, so you may run into problems using other protocols.

If you use multiple protocols, you should configure all servers in the cluster to support the same set of network protocols. This ensures that clients can connect to any server in the cluster in the event of failover or workload balancing. You should also specify the TCP/IP port of the server on the Server_Cluster_Default_Port setting in the NOTES.INI file of all servers that belong to a cluster. For example:

```
Server_Cluster_Default_Port =TCPIP
```

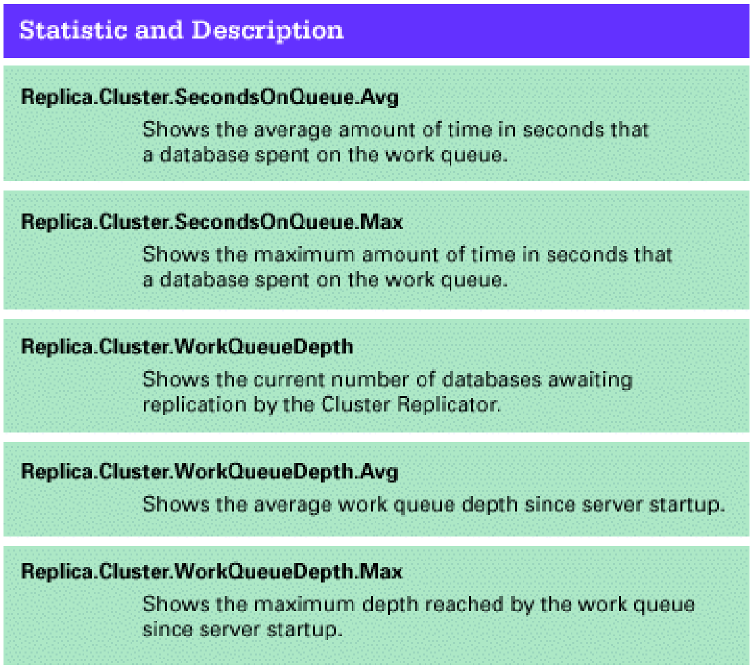
This directs cluster communications between the servers through TCP/IP. (The release notes for some versions of Domino also instruct you to set the Server_Cluster_Probe_Port NOTES.INI variable. This is an error in the documentation ... Domino does not use the Server_Cluster_Probe_Port setting.)

Notes Named Network configuration: A related point is how to configure your named networks. A Notes Named Network (NNN) is a group of Domino servers that share a common LAN protocol and have the same Notes Network name in their Server documents in the Public Address Book. Since we just said that all the cluster members are configured with the same network protocols and they are connected with a LAN, it seems like a given that the servers are all on the same NNN. But you still need to set the Notes Network name fields consistently, or Domino considers the servers to be in different NNNs. If you inadvertently or intentionally put servers in the same cluster in different NNNs, you may not get the behavior you want. For example, failover of mail delivery (discussed in more detail below) may not work unless you set up Connection documents to explicitly route mail within the cluster.

You can verify that you have configured your Notes Named Networks correctly by checking the Server/Networks view in the Public Address Book. This view shows each configured network for all servers in the domain, grouped by network. From this view, you can check each network to ensure that either all cluster members are shown or that none are shown.

A private LAN for intra-cluster communication: One more thing to consider when designing your cluster network topology is whether to use a private LAN for intra-cluster communication. The main reason for doing this is to offload the cluster probe and cluster replication network traffic from the LAN, leaving more bandwidth for client communication with the cluster servers. If you anticipate a large volume of cluster replication traffic, you should definitely configure an intra-cluster private LAN. Note that all cluster members must be connected by the intra-cluster private LAN as well as the LAN for client access. If you already have a cluster, and you want to know if you'd benefit from adding an intra-cluster private LAN, use a network sniffer to monitor the network utilization of the LAN during periods of peak server activity. A common rule of thumb is to add extra capacity if network utilization exceeds 40%. If you don't have a network sniffer, or if you just want a rough indication of whether bandwidth between cluster servers is a concern, check the following statistics:

Note: These statistics do not appear in a cluster statistics report, but you can view them using the Show Stat command or by adding them to the cluster statistics report form.



Statistic and Description
Replica.Cluster.SecondsOnQueue.Avg Shows the average amount of time in seconds that a database spent on the work queue.
Replica.Cluster.SecondsOnQueue.Max Shows the maximum amount of time in seconds that a database spent on the work queue.
Replica.Cluster.WorkQueueDepth Shows the current number of databases awaiting replication by the Cluster Replicator.
Replica.Cluster.WorkQueueDepth.Avg Shows the average work queue depth since server startup.
Replica.Cluster.WorkQueueDepth.Max Shows the maximum depth reached by the work queue since server startup.

The SecondsOnQueue.Avg statistic indicates the average amount of time needed to push changes made on this server to other servers in the cluster. A value of 60 or less means cluster replication is updating the other servers within one minute of the change on this server. Check the SecondsOnQueue.Max statistic to see the maximum amount of time it takes for changes on this server to get to the other servers in the cluster. The values of the WorkQueueDepth statistics depend on the frequency of database updates and number of different databases updated on the server. Typically, the current and average depth values will be in the single digits and the maximum will be in the low double digits. Values outside this range could be an indicator that network capacity is affecting the performance of cluster replication.

Another very good reason for providing a private LAN for intra-cluster communication is to eliminate the network as a single point of failure in your cluster. If you have a private LAN for intra-cluster communication, this means that all the servers in the cluster are connected over at least two distinct LAN segments. So if a network interface card (NIC), network cable, or some other component on one LAN segment fails, Domino still has network connectivity between all the servers in the cluster. This is important, because it ensures that servers remain aware of the availability and load of the other servers in the cluster, and that cluster replication will continue to keep databases in tight synchronization. To learn more about how to configure your private LAN for intra-cluster communication, see the sidebar "**Configuring a LAN for intra-cluster communication**."

Determining how many servers to put in a cluster

Another basic decision when setting up a cluster is how many servers to put in the cluster. If you are just getting started with Domino clustering, you'll probably want to start with just two or three. But, as your confidence grows, you may want to add more servers. Lotus states that a cluster can have as many as six servers. While it's true that the overhead of cluster probes and cluster replication could begin to degrade performance when there are more than six servers in the cluster, there is no hard limit imposed by Domino. Some customers (and even Notes.net and Iris itself) routinely run clusters with eight or nine servers. I'm not trying to encourage you to build clusters this large, but if you have valid reasons for having more than six servers in a cluster (for example, to migrate a server that is already in a cluster of six), you'll probably be happy to know that Domino does not actually prevent you from doing this. You also need to be aware that clusters of more than six servers are still officially "unsupported," so it would be best to use this capability sparingly, if at all.

If you have a lot of servers and you want all of them clustered, should you build a small number of large clusters (five or six servers per cluster) or a large number of smaller clusters (two or three servers per cluster)? That's a hard question to answer in general, but consider that when one server in a cluster fails, the workload of that server must be absorbed by the other members of the cluster. This means that these cluster members should have enough spare capacity to service the users of the failed server. For a cluster of two, that's a pretty steep requirement. It means that normally both servers would be running at about 50% capacity. But for a cluster of six, the story is very different. Since, on average, each of the remaining servers in the cluster would only have to absorb one-fifth of the failed server's workload, you could run these servers at 80% capacity in non-failure cases and still have the spare capacity needed for a failure situation. A similar argument holds for disk capacity, since each database needs at least one other replica in the cluster to make it highly available. This makes larger clusters seem somewhat more attractive. But, if you've suddenly gotten the urge to create a 16-node cluster, please re-read the preceding paragraph.

Setting up Cluster Replication

Another unique aspect of Domino clusters is the event-driven (cluster) replication that is used to keep databases in sync across all the servers in a cluster. The Cluster Replicator is the component that performs event-driven replication. It is integrated with the server to receive notification of all database modification events. These events trigger replication of the changes to the other replicas within the cluster. In this way, replication is "event driven," rather than schedule driven.

Multiple Cluster Replicators: When a server is added to a cluster, Domino loads the Cluster Replicator (clrepl) and adds it to the ServerTasks= setting of NOTES.INI so that it starts automatically if the server is shutdown and restarted. This is to ensure that at least one Cluster Replicator runs on each server in the cluster. This should be sufficient for most cluster configurations. However, in some cases, a single Cluster Replicator may not be able to keep up with the replication workload. In this situation, you may want to run multiple Cluster Replicators, which split up the replication workload and process it in parallel. This capability is similar to the support in the (standard) Domino replicator for running multiple replicator tasks. You can load additional Cluster Replicators from the Domino console with the Load Clrepl console command. You can also specify multiple instances of Clrepl on the ServerTasks= line, which will cause the specified number of Cluster

Replicators to load at server startup. After you start multiple Cluster Replicators, you can use the Tell Server command to stop all Cluster Replicators; however, you cannot use the Tell Server command to stop a specific Cluster Replicator.

As a guideline, the Domino Administration Help recommends that you run as many Cluster Replicators as there are cluster members, minus one. To hone the number even further, you should monitor the Replica.Cluster.WorkQueueDepth statistic (in the Replica Statistics report). This statistic shows the current number of modified databases awaiting cluster replication. If this value is consistently greater than zero, you may need to enable more Cluster Replicators. However, it's possible that network bandwidth, and not the number of Cluster Replicators, is the real cause of the backlog. For more information, see the section "A Private LAN for intra-cluster communication," earlier in this article. Keep in mind that if you add more CPUs or memory to your server, you will see better Cluster Replication performance.

Setting up standard replication

Even though cluster replication is on the job, you still need to configure regular scheduled replication among the servers in the cluster. Some of the reasons for this are:

- It is possible to disable cluster replication for certain databases (see the Domino Administration Help for more information on this). In most cases, you will still want these databases to replicate, just not every time they are updated. Scheduled replication handles this.
- Since cluster replication events are kept in memory only (for performance reasons), there are some situations where these events could be lost when multiple servers in the cluster fail. No data is lost in these situations -- but cluster replications that were pending when the server failed won't be performed until the next scheduled replication.
- The Cluster Replicator defers processing of selective replication formulas to the standard replicator. Evaluating selective replication formulas can be expensive in terms of performance, since these formulas can be arbitrarily complex, so this processing is deferred to minimize the overhead of cluster replication.
- The Cluster Replicator pushes changes to other servers that have a replica of the changed database, but it does not update other replicas on its own server. You may ask ... why would there be more than one replica of a database on a server? Typically, there isn't, but Domino does not prevent this. And it may seem perfectly reasonable to have two replicas on a server if one is a selective replica. Perhaps the best advice to give here is don't put multiple replicas of a database on a server in a cluster. But if you have a pressing need to create a second replica of a database on a server in a cluster, you need scheduled replication to bring everything back in sync.

  Note: When there are multiple replicas of a database on a server, the Cluster Manager uses "failover by pathname" to select the replica for a user to open in a failover situation. So if you do elect to put multiple replicas of a database on a server, make sure you use pathnames (directory path and file name) that are consistent with the pathnames of the other replicas in the cluster.
- When one of the servers in a cluster fails, the Cluster Replicators on the other servers queue replications that must be pushed to that server, and periodically check for the server to become active again. The time interval between attempts to reach the failed server starts at one minute, and doubles on each unsuccessful attempt, until it reaches the maximum delay of one hour. This means that if one of your clustered servers has been down for more than 63 minutes, it

might take as much as an hour for the Cluster Replicators on the other servers to push the pending changes over to it and bring all its databases up to date. (In Domino R5, the Cluster Replicator detects servers rejoining the cluster much more rapidly -- within a minute or two. This largely eliminates this particular issue.)

Regular and frequent scheduled replication is the solution to all of the above issues. I recommend enabling scheduled replication 24 hours a day, seven days a week, and setting the time interval to 60 minutes. Each time a server starts, it performs any due or overdue replications. With replication set for every 60 minutes and always enabled, if a cluster server has been down for more than 60 minutes, when it restarts, it immediately replicates with the other servers in the cluster.

Replication with servers outside the cluster: One of the benefits of Domino clusters is that they can help simplify your replication topology, and at the same time, improve its reliability and performance. This is because Domino allows you to configure a server outside the cluster to effectively replicate with all servers in the cluster. This is done with a single Server Connection document that specifies the server outside the cluster as the source and the cluster name as the target of the replication. When a server replicates with a cluster, any databases on the server that also have a replica on any server in the cluster will replicate. Databases do not have to be on the same server in the cluster. If more than one replica of a database exists in the cluster, Domino replicates with one replica, and cluster replication propagates the changes to the other replicas in the cluster. The server that initiates the replication does not require an Advanced Services license, but must be running Release 4.5.

Replication with a cluster is also more reliable, since Domino replicates with any server in the cluster that has a replica of the database processing. So if one server in the cluster is unavailable, replication can continue as normal as long as all databases on the unavailable server also reside on another server in the cluster. Furthermore, when replicating with a cluster, Domino uses its workload balancing mechanisms to choose the target servers so that replication performs as quickly as possible.

Setting up mail for a cluster

When you set up mail for your cluster, you need to set up failover for mail routing and mail delivery, directory assistance, and shared mail. The following section provides you with tips for setting up each of these areas:

Failover for mail routing and mail delivery: To provide high availability for mail routing, Domino allows administrators to define alternate paths for mail routing. Domino uses a "least-cost" algorithm to select a route to the target server. This algorithm uses a failure history to weigh the routing choices, so that it can re-route messages around network failures. Alternate routes and least-cost mail routing are standard features of Domino, and operate independently of clusters.

However, clusters do have a role to play in mail delivery. Mail delivery is the process of depositing a message into the actual mail file of a user. Normally, mail is routed to a user's home server, and this server then delivers the message into the user's mail file. In R4.0, the mail router was enhanced to support delivery of mail to mail files in a cluster. When a message is about to make its final hop (to the user's home server) and the destination server is unavailable, the router checks if the destination server is a member of a cluster. If it is, the router then checks to see if its server is also a member of the cluster and contains a local replica of the user's mail file. If so, the router delivers the message to the local replica. Otherwise, the router looks for another server in the cluster with a replica of the user's mail file. If it finds one and that server is available, it routes the message to that server. When the router on the cluster member processes the message, it again attempts to deliver the message to the user's home server. If the user's home server is still unavailable, the router then delivers the message into the cluster replica of the user's mail file.

The combination of alternate routes, least-cost mail routing, and mail delivery failover provides a complete, end-to-end solution for high availability of mail routing and delivery to end users.

Unfortunately, this processing is not enabled by default. You must enable this processing by specifying the NOTES.INI setting MailClusterFailover=1 on all servers in the domain that could route mail to a cluster. To simplify administration, the best practice is to set MailClusterFailover=1 on all servers in the domain using a Domain Configuration document. A Domain Configuration document is similar to a Server Configuration document, described earlier, in that it allows you to specify NOTES.INI settings for a group of servers. In the case of the Domain Configuration document, the settings are applied to all servers in the domain. (Prior to Domino 4.6, you created a Domain Configuration document by specifying "*" in the Server Name field of a Server Configuration document.)

Directory assistance: Directory assistance is a feature that searches multiple Public Address Books to resolve names of mail recipients. This allows users to easily send mail to users in other domains. You enable directory assistance by creating a Master Address Book (MAB), which specifies the set of Public Address Books to be searched. Then you specify the name of the MAB in the Server document. The MAB must reside on the server, but it can point to Public Address Books that reside on another server.

There are two important aspects of Directory assistance that relate to clusters. The first is that Directory assistance will failover if a server pointed to by the MAB is unavailable. This means you can make the Directory assistance feature highly available if you use clustering to ensure that the Public Address Books pointed to by the MAB are highly available. Second, you should make sure that all servers in a cluster specify an equivalent Master Address book (the best practice would be to create a single MAB with replicas on each cluster member, to ensure they are kept in sync). This is important because the NAMELookup function, the fundamental service in Domino for name resolution, will failover if it attempts a lookup on a server in a cluster and that server is unavailable. In order for name resolution to work consistently for servers in a cluster, it is important that you configure all the cluster members with an equivalent (replica) MAB.

Using shared mail in a cluster: Shared mail, also known as "single copy object store," or "SCOS," is a space-saving feature that stores single copies of mail messages in a central database on a server. A pointer to this message, rather than the entire document, is stored in the mail file of each message recipient on the server. Users access and manipulate this message just as they would a message in their own mail database.

In the short time that I have been involved in Domino, I have found that shared mail is almost a personal issue with Domino Administrators. Some love it, and some hate it. For those of you that hate it, feel free to skip to the next section. If you love shared mail, I am happy to tell you that there is no need to give it up when going to a cluster configuration.

But there is some extra work needed to make shared mail works properly in a cluster. In particular, you need to inform the replicator and Cluster Replicator that you use shared mail on the server. You see, the logic to implement shared mail is largely contained in the mail router. The router handles the task of splitting the message into its unique part, the header information, which is stored in the user's mail database, and its common part, which is stored in the shared mail database. But if the message is subsequently replicated, it is reassembled into a complete message before it is sent to the target server. When the replicator stores the message on the target server, it needs to be told to split it apart again. To tell the replicator (and Cluster Replicator) to do this, you must issue the following command on the server console of each server:

Load Object Set -Always *MAIL SHARED.NSF*

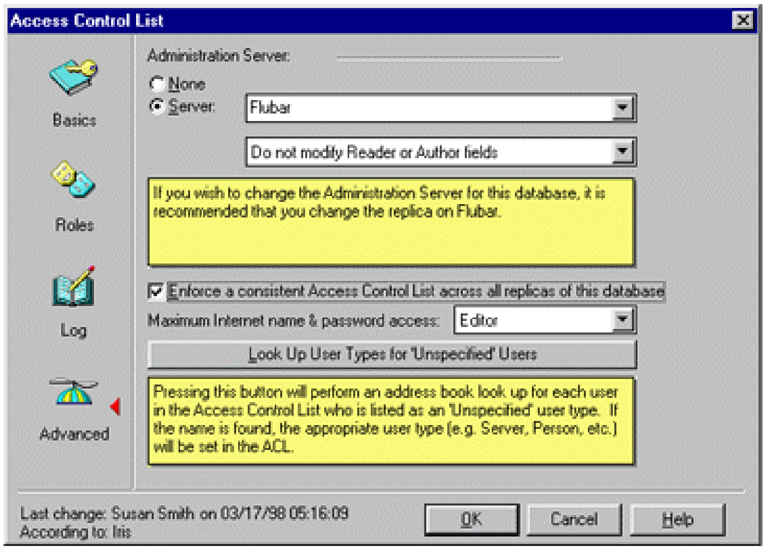
where *MAIL* is the name of a replica mail file or a directory of replica mail files and *SHARED.NSF* is the name of the shared mail database on the server. Actually, this configuration step is not unique to clusters -- it is necessary whenever you have replicated mail databases on a server that uses shared mail (including replicas on a user's workstation). See the Domino Administration Help for more information about using shared mail on replicated mail files.

Setting up database Access Control Lists

Another fine point of cluster configuration is setting up database Access Control Lists (ACLs). Management of database access is always an important issue for Domino administrators, but there are two special concerns to ACL management when a database resides on a server in a cluster.

Consistent User Access: The first concern is consistent access for users. Database ACLs should be set up so that a user has the same access rights on all replicas of the database on all servers in the cluster. This ensures that failover and workload balancing work correctly. For example, if a user attempts to open a database on Server A and fails over to the replica copy of the database on Server B, it is important that the user has the same access rights to the replica copy that they had to the original database. (For more information on ACLs, see the article "**The ABC's of using the ACL**."

One way to ensure that database ACLs are kept in sync across all replicas is to use the "Enforce Consistent ACLs" feature. You do this by selecting Advanced in the Access Control List dialog box and selecting the "Enforce a consistent Access Control List across all replicas of this database" option, as follows:



Administration Server:

○ None

● Server: Flubar

Do not modify Reader or Author fields

If you wish to change the Administration Server for this database, it is recommended that you change the replica on Flubar.

☑ Enforce a consistent Access Control List across all replicas of this database

Maximum Internet name & password access: Editor

Look Up User Types for 'Unspecified' Users

Pressing this button will perform an address book look up for each user in the Access Control List who is listed as an 'Unspecified' user type. If the name is found, the appropriate user type (e.g. Server, Person, etc.) will be set in the ACL.

Last change: Susan Smith on 03/17/98 05:16:09
According to: Iris

Selecting this option not only ensures that the ACL remains consistent across server replicas, but also that the ACL is enforced when replicas are accessed locally on either a server or client.

Another approach to keeping database ACLs consistent is to give all servers in the cluster Manager access to the database. This ensures that when the ACL changes on any of the servers in the cluster, this change replicates to all the other servers. The ACL is a special document that never suffers a replication conflict. Even if the ACL changes on separate database replicas between replications, when replication occurs, the most recent change prevails (assuming that both servers have Manager access). To give all servers in the cluster Manager access to a database, you can create a group in the Public Address Book that names all the servers in the cluster, and then add this group to the ACL with Manager access. Or you can use the LocalDomainServers group. By default, Domino creates a group in the Public Address Book named LocalDomainServers. This group lists all servers in the current domain. The names of servers you register in the current domain are automatically added to the LocalDomainServers group. The default ACL of most databases created with Domino templates has an entry for LocalDomainServers, so all you need to do is make sure this entry has Manager access, and you're all set. (Since the domain probably includes servers other than those in the cluster, using LocalDomainServers goes a bit farther than necessary, but it is such a convenient mechanism, many administrators choose to use it).

In order to provide consistent access to databases, you need to understand that there are a number of other mechanisms for restricting user access. It is equally important that these be consistent across the servers in the cluster. These mechanisms include:

- Server restrictions (for example, deny access lists)
- Access lists within the database and directory links
- Reader lists attached to documents, views, folders, and so on

Good general advice on these items is to use group names instead of user names wherever possible. This gives you a single point of control for managing these forms of access.

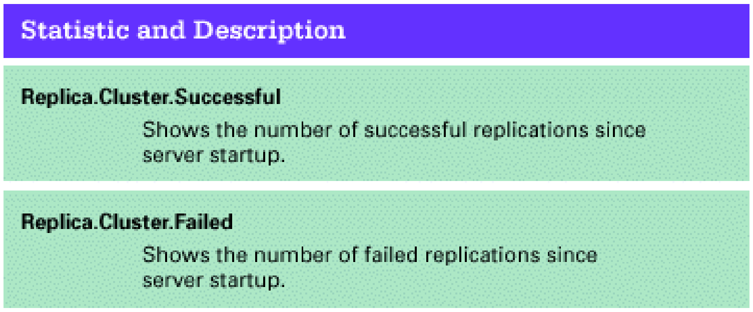
The second special concern in ACL management for a cluster is ensuring that the cluster servers have sufficient access to maintain the replicas within the cluster. This is important because if access restrictions prevent some data from replicating to other replicas in the cluster, users will see this data as "missing" when failover or workload balancing causes them to open a different instance of the database. In addition to controlling user access, the ACL also tells the server what content they can access and update. Since the replicator and Cluster Replicator run under the ID of the server, their ability to update the contents of a database is limited by the access granted to the server in the ACL. This is another good reason to give all servers in the cluster Manager access to all databases. In fact, ensuring that the ACL itself replicates, as described above, is really just one instance of the general issue of ensuring all the contents of the database replicate properly.

In addition to giving all servers in the cluster Manager access, there are a few additional caveats to be aware of. First, private folders are folders created by a user that can only be accessed by the creator of the folder. A good example of a private folder is the "My Favorites" folder that now appears in many of the standard templates shipped with Domino. Each user of a database can create a "My Favorites" folder, and this folder will only be visible to the user that originally created the folder. Normally, private folders and their content do not replicate during database replication between servers. Within clusters, however, private folders are replicated to other replicas within the cluster. This ensures that users can access their private folders even if they fail over. Both cluster replication and standard replication support replicating private folders and their content within a cluster. However, to ensure that this feature is only used between servers, it is only enabled if the user type of the ACL entry for the server is set to "Server" or "Server group." Servers that do not have an user type of "Server" or "Server group" in their ACL entry cannot replicate private folders, even if they have Manager access to the database.

Second, as mentioned above, there are other access control mechanisms besides the database ACL. Of particular importance for replication are access rights on database and directory links. Even if you have given a server Manager access to a database, the server could be blocked from accessing the database if the database is also protected with a database or directory link. If there is no access list specified in a directory link, Domino allows any user (or server) to access the directory. However, if an access list is specified, then any users or servers not on this list will not be allowed to access the directory. So, if you use directory links with access lists on the servers in the cluster, make sure to include the cluster servers (or LocalDomainServers) on the list. Otherwise, other servers in the cluster will be unable to replicate with databases within these directories.

Third, Domino also allows database designers to control access to individual documents through the use of Readers fields. A Readers field explicitly lists the users who can read a document (provided they also have Reader access to the database). If a document includes a Readers field, even users with Editor access or above to the database can't read the document if they aren't included in the Readers field. Reader lists can also be defined for folders and views. You want the servers in the cluster to have access to all these objects so that they are replicated consistently throughout the cluster. However, access at this level is often controlled by database designers and/or document authors and editors, not by database managers or Domino administrators. If the designer or author doesn't specify the servers in the cluster on the Reader list, that object won't replicate to the other servers in the cluster.

You can check the Replication Events view in the server's log file for replication problems with other servers in the cluster. You can also check the following statistics to determine the number and frequency of replication failures encountered by the Cluster Replicator:



Statistic and Description	
Replica.Cluster.Successful	
	Shows the number of successful replications since server startup.
Replica.Cluster.Failed	
	Shows the number of failed replications since server startup.

Conclusion

For the most part, cluster configuration is a straightforward process that is well covered in the Domino Administration Help and other sources. However, I've found that some customers would like just a little more information about some configuration issues than what is provided in the published materials. I have tried to cover a number of these issues in this article. Hopefully, you have found one or two nuggets that you can employ in your own Domino cluster environments. Lotus plans to enhance the Domino Release 5 documentation, to help you to better understand clusters. If you have any feedback about clusters, feel free to direct it to the **Notes/Domino Gold Release Forum** here on Notes.net, and be sure to categorize it under Domino Server - Clustering.

ABOUT THE AUTHOR

Michael Kistler is a Senior Software Engineer in IBM's Software Solutions Division. He is currently on assignment at Iris, working on significant extensions to cluster functions of Domino. Prior to his assignment at Iris, Mike worked in an AdTech group that was exploring new technologies for high availablity and scalability. Prior to joining the Software Solutions Division, Mike was a software architect in IBM's Large Systems Computing Division, working on a number of enhancements to IBM's Multiple Virtual Storage (MVS) operating system. Mike holds an MS degree in Computer Science from Syracuse University, and an MBA from New York University.