White Paper: UNC Paths for Data Storage and Scalability

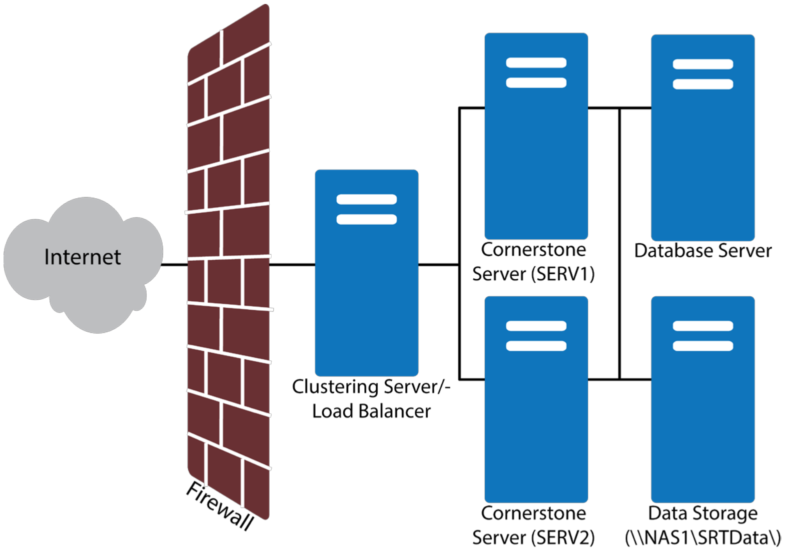
Introduction

Scalability is a hot topic in server technology. Large companies want to use their hardware to maximum efficiency. They want servers that handle both everyday file exchange and spikes of high traffic.

At the most basic level, a server must have access to the data a user needs and the capacity to deliver it. This becomes more difficult with increased demand from large numbers of users. Servers get bogged down with increased activity, creating headaches for users and IT departments equally. To ease the traffic load, more servers can be introduced in a network, to form a clustered environment with a load balancer to portion out the server requests to multiple servers.

But where do the servers get the data to answer client requests? Storing duplicate information on each server is impractical—it wastes disk space and invites unsynchronized content, which leads to mistakes and lost time.

To solve this scalability issue, many companies store all data on one machine—either one of the clustered servers, or a separate data storage unit—and use a UNC to allow all of the clustered servers to access the same bank of data.



A UNC, or **Universal Naming Convention**, allows a user to store and access data on any server in a network. The UNC specifies a common syntax to describe location information for a resource. This could be a file or folder or even a printer, any asset that can be used by a network. The UNC path is a fixed point on the network the server can locate in order to access data.

UNC syntax includes the computer name, share name, and optional subdirectory where the resource is stored. The UNC syntax for Windows systems is **\\computername\sharedfolder\resource**.

Another example: Say you have a computer named SERV1 which has a shared folder called SrtData with a subdirectory called Cluster Test. The UNC referencing the location would be **\\SERV1\SrtData\Cluster Test**.

Deploying UNC Paths in a Scaled Environment

If a server is deployed in a scalable environment, one or more servers run in parallel and can access the same backend data storage to serve different clients.

In order to scale your server to multiple boxes, you will need to configure the server to access all data via a UNC share rather than a local drive or a mapped network drive letter.

For example:

Say you have multiple servers networked into a clustered environment.

SERV1 is the primary, or first, server installed. SERV2 will be a backup server to be added at a later date. With a multi-server environment, there are two scenarios, both of which require the same configuration:

- User data will be stored on a local fixed disk on the SERV1 primary server box. Both SERV1 and SERV2 will need access to this data.
- User data will be stored on a third hard drive on the network, a box that is neither SERV1 nor SERV2. In this case, we'll call this NAS1.

For either scenario, the configuration setup is basically the same.

First, you will need to configure the UNC to allow your servers to access it. This requires a UNC share and NTFS permissions adjustments to the directory where the data is stored.

Second, update the permissions on the share to allow the servers to access data on the share. This is very important; incorrect permissions will prevent the server from accessing the data.

Typically servers run under the context of one of the special built-in Windows system accounts such as Local System or Local Service. These built-in accounts do not have proper NTFS rights to access files stored on remote UNCs.

How do you fix this?

You can either grant full NTFS rights to all users (a very unsecure method) which would include the server service, or you can create a special NT User Account for the server and add it to the Access Control List (ACL) for both the share and the underlying NTFS file system.

UNC Share Permissions vs NTFS File System Permissions

When leveraging UNC shares for data access, keep in mind that the NTFS file system has its own set of permissions for users who access the files, separate from the permissions on the UNC share you create. Unexpected problems can result if the two sets of permissions conflict, particularly with deleting data from the server.

If your NTFS permissions permit you to delete data, but your UNC share permissions do not, the data can't be deleted because the restriction translates through the UNC share. The same is true if the permissions are switched (delete is permitted through the UNC Share but not permitted on the NTFS permissions).

It's best to ensure that the UNC Share permissions align with the NTFS permissions for the special NT User Account created to be used by the Cornerstone MFT Server.

Cornerstone NTFS Integration

The Cornerstone MFT Service deals with the NFTS rights problem through built-in features, which removes complicated setup procedures for the administrator. For information on using UNC with Cornerstone servers, see our QuickStart.