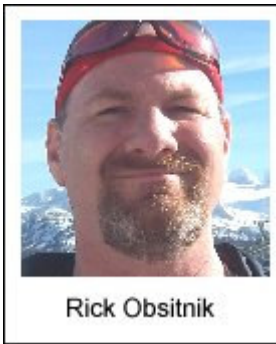


Data Shepherds! Protect Your Flock!

By Rick Obsitnik



About the Author

Rick, a specialist in database application development, who recently made the change to database administration, has worked in IT for 25 years in nearly all areas excluding sales! Quite an achievement! He currently works in a SQL Server 2005, ESRI ArcGIS SDE 9.3 environment with Visual Studio 2005. His appetite for knowledge in all things IT grows daily, although his first love is databases. His interest in humor and the absurd led to the creation of the [Sniglets](#) section on the SQL Server Club web site.

You can read more articles and blogs by Rick on his [Critical Status blog](#), which was designed to keep track of and access his database and development related thoughts, ideas and issues.

It's no secret that the economy is crumbling. Is there a light at the end of the tunnel? Maybe. Maybe not. One thing that is for sure, those organizations surviving in these dark times really need to ensure their information is secure and is getting maximum and optimal usage.

Database Administrators are guardians of organizational information. Data Shepherds if you will. This is a good time to strengthen and expand your skills in ensuring the security and availability of the stored data you were hired to protect. Below are just some thoughts and my opinions on areas to implement a plan, if you don't currently do it or tighten if you do. I welcome all comments on this as there is always improvement in everything including database administration.

Security and availability go hand in hand. If your SQL Server is not secure, then the risk of down time for various reasons make the system unavailable. There are some simple steps that can quickly increase the security of your server.

The SA password should be kept incredibly safe and should have a bare minimum need-to-know what it is. It should not be used except for the most secure information. Giving away this password is giving away not just the barn but the whole farm.

In a similar scenario, be very frugal in assigning sysadmin rights to any login. This is something for all DBAs to fight for. Yes, management can throw up their hands in a pissing contest over who demands rights to what just to end the discussion, YOU should be the defender of the faith! The old adage is true in that too many cooks in the kitchen spoil the broth. Too many sysadmins with different agenda can confound a system and bring it down. Let the finger pointing begin! We've all been there.

Limit the amount of users that can access the physical box especially with admin rights. This may need to be worked out with Sys Admins but they will be of the same mindset. Hosing the box, by definition, hoses your SQL.

Use instance and database roles. Assign rights to roles, then associate users to roles. Special rights superseding the roles can be done but are generally the exception. A

similar situation is true with schemas. One thing you can do with schemas is have redundant table, views, etc for users or groups. Generally, this is not done but there are situations where this can be beneficial. Remember, you want to simplify while locking down.

Managing users and other objects in a database can be complex and cumbersome. Group as much together as possible. Work with your development teams, project managers, business analysts and whoever to sort out permissions groups as soon as a project begins to take shape.

In this same vein, Active Directory Security Groups (ADSG) is an outstanding tool. By creating ADSGs based on permissions, such as "Internal_Planning_Viewers" and "Internal_Planning_Editors" and "Internal_Planning_Admins," you can add these groups as a login in your SQLs – yes multiple instances.

A point of contact for an organizational functional group can be tasked to add members to each respective group. These groups can be used in multiple databases and assigned to similar sounding roles, such as "MasterPlan_Viewers" and "MasterPlan_Editors" and "MasterPlan_Admins" in a MasterPlan database. They may have the similar roles with similar rights in the "Design" database. You only need to create six roles in two databases but may have 150 users on both systems. It's far easier to maintain the roles. Do the math...

This alone will keep your systems rather secure and easy to manage. Two birds, one stone! The risks of damage due to unauthorized access are now much minimized. You're already winning the battle of availability. There are several things a DBA can do to maintain high available of any sized system. Let's take a look...

At the top of the list is having a sound recovery plan. Backing up and restoring databases and servers effectively and efficiently is the best way to reduce down time. Let's face it, shit happens. Lightning strikes, UPSs fail, servers go down. Your phone rings endlessly when power is restored! To stop your phone from ringing, you must have your systems online as quickly as possible and functioning as flawlessly as possible. There should be daily backups and the backups stored in redundant places and with a mix of media to ensure that something can be put in place as quick as possible.

I recommend that most systems have a weekly full backup with daily incremental. This reduces the load of your backup media and can be fairly quickly restore. Got space? Do nightly full backups! Do it more often, if necessary and if it's feasible.

Make sure you use multiple media per backup. Backing up to tape and placing the tape off-site is a smart thing to do. If the facility is damaged, the probability of the remote facility where the tape is store is likely not. I also like to store a copy of backups on multiple servers, compressed for quick restores. Again, if you have the space, redundancy is a good thing. Make sure that you have a source that's close at hand for quick recoveries. I guarantee that you will impress your users, your boss and even executive management with fast response times. You'll most likely notice it at review time even in this time when your neighbors are standing in line at the unemployment office!

Clustered servers are very common place, as are redundant array of independent disks (RAID) and Storage Area Networks (SANs) as servers/CPU's, storage capacity

and memory are rather affordable to most budgets. Such technologies use redundant technologies to allow availability when a system fails. It allows for systems to seamlessly failover to running systems, which allow systems teams to repair the downed system while users notice no change.

Physical systems cost money BUT downed systems cost more in productivity. People sitting around waiting to work yet still being paid. The bean-counters do some serious frowning. Keeping them smiling with available systems, it keeps you in their good graces. You'll be in their minds if there is some cash left in the kitty at the end of the year. MORE important is job security in a vastly unstable time. Also, there are things that you can do that are only charged by your hours which you are already being paid for. Now is a great time to save your organization money in a time of dwindling funds.

I chose to move from database applications development to database administration due to its stability. Systems need to keep running when projects are over. The pay is generally comparable and usually higher. I get to keep programming with T-SQL and VBScript and hopefully soon some VB.NET user defined functions. I keep our databases available by whatever means necessary. Not a single one in production has been down for over a year outside of power outages. Because of this, I have job security. We all need to think about this in these dangerous times. Stay nimble and stay on the job!