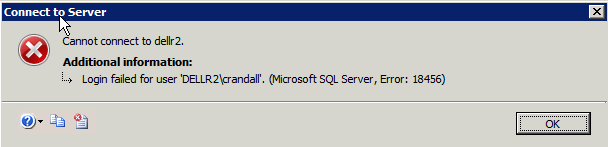
SQL Server 2008: Forgot to add an administrator account?

Thursday, December 10, 2009 at 09:50PM  
Chris Randall

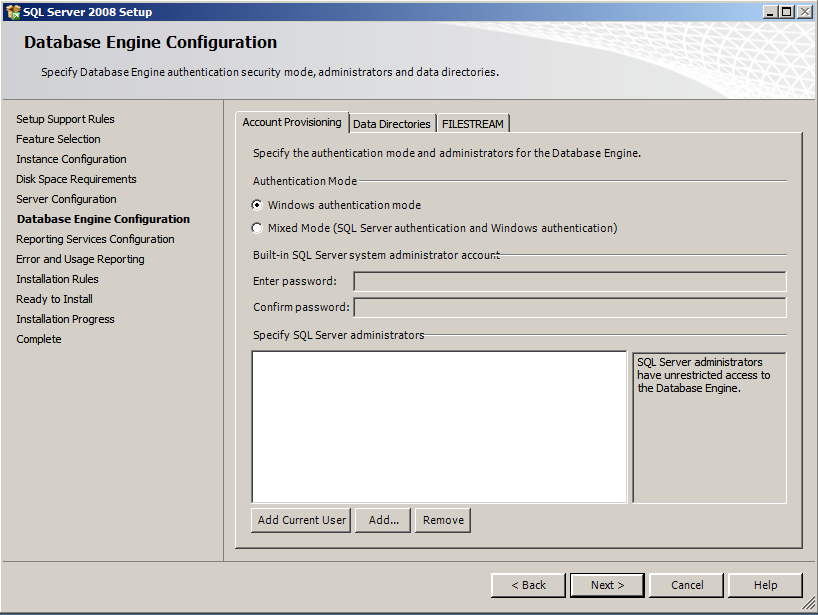


I recently built a test box with an instance of SQL Server 2008 Developer Edition. I've done this so many times that I apparently got sloppy and stopped paying attention during setup. This time it caught up with me. On first launching Management Studio to do some post-install configuration, I realized that I'd "locked myself out" - I skipped the step to add my account to the sysadmin role on one of the instances. Setup requires that some account be added as sysadmin, but I didn't remember what I'd chosen, and there were no other accounts on the box that stood out as likely candidates.

I'd also opted to stick with the default Windows authentication mode, meaning the built-in **sa** account was not available.

For those who haven't made the move to SQL 2008, you'll find the following step in the Setup program. Unlike in earlier versions, local Windows admins aren't automatically granted access to SQL Server 2008. You'll want/need to designate one or more account to be added to the sysadmin

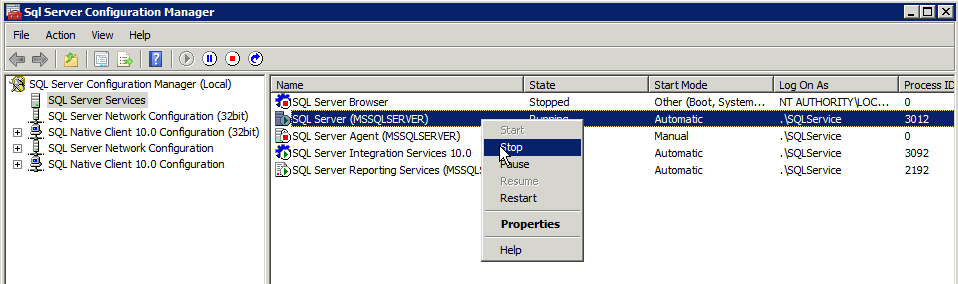
server role. Skipping that step somehow got me in this mess.



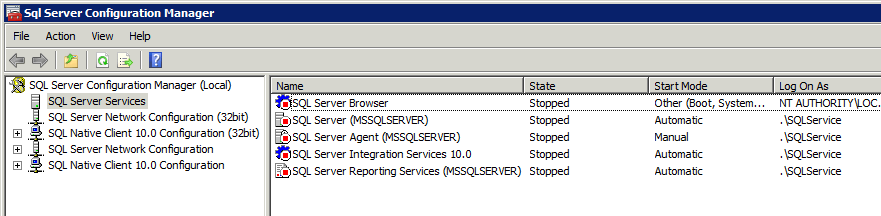
I really wanted to avoid the time it would take to uninstall and reinstall the instance. Happily, I remembered another way: as a member of the local Windows Administrators group, I could start the SQL instance from a command line with the -m parameter for single user mode, connect to the service, and let myself in the "back door". ("m" for single user? Sure, that makes perfect sense. I remember this as "mono" for one at a time.)

One of the side effects of starting in single user mode is that members of the local Windows Administrator group can connect to SQL Server with its one connection. (Whether this is a good idea from a security standpoint remains open for debate, but it sure saved my bacon this time.) This would give me the ability to log in, map my Windows account into a SQL Server login, and add my new login to the sysadmin role. Here are the steps I followed.

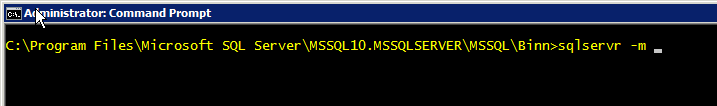
1. Using the SQL Configuration Manager, I stopped the instance I needed to work with.



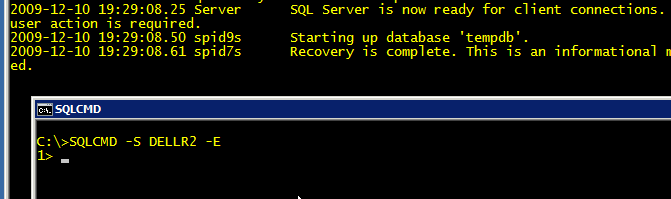
1. I also stopped any other SQL-related services which might try to connect to the instance, using up my one connection. This included Agent, and might also include Analysis Services or Reporting Services, depending on your installed options.



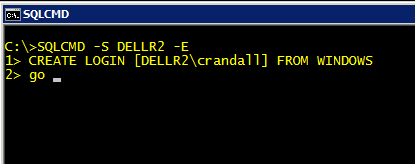
1. From a command prompt, I navigated to the SQL Server folder where the program executable lived.
2. I started SQL Server's executable with the -m parameter and waited for it to complete startup and recovery of any databases. (Note that the error log information echoes to the command prompt window but will simply pause when it's done startup - leave this window open to keep SQL Server running in this single-user mode. You can confirm if you successfully launched in single-user mode by looking for a line in the output that reads "**SQL Server started in single-user mode. This an informational message only.**"



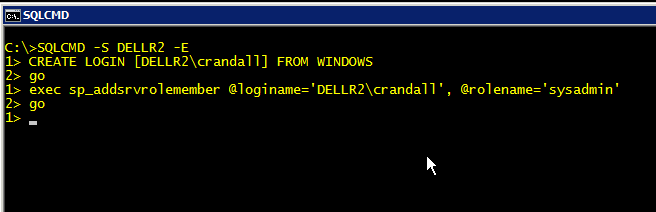
1. From a query tool, I connected to SQL Server using my Windows account. While I could have used a new query window in SQL Server Management Studio, I didn't want to wait for it to load, so I used SQLCMD from another command prompt session. (You can't use the Object Explorer pane in SSMS to connect when in single-user mode: clickers beware! This is one of those times where knowing the code (or having a saved script) is essential) If you're unfamiliar with SQLCMD, the -S switch specifies the instance name, the -E specifies that you're using Windows Authentication. These switches are case-sensitive.



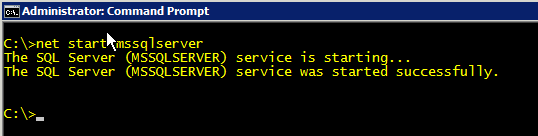
1. Once connected, I created a login for my Windows account:



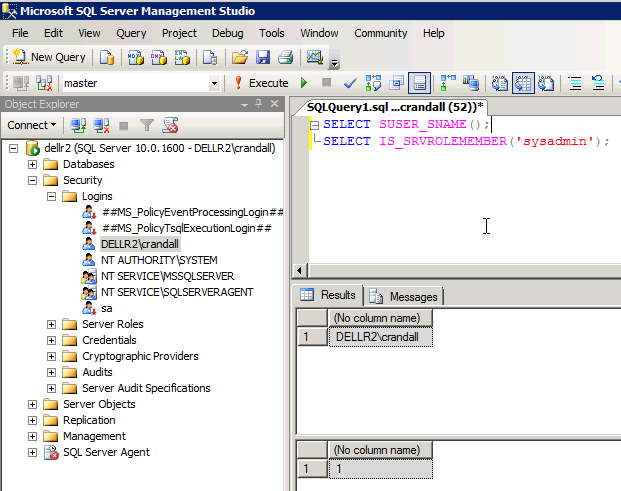
1. Then I used a system stored procedure to add my account to the sysadmin fixed server role.



1. Now that I've let myself back in to SQL Server, I stopped the service with the SHUTDOWN command. I could have closed the other command prompt window as well.
2. Finally, I restarted the service. Since I already had a command prompt window open, I opted for Windows' NET START command. The SQL Configuration Manager, the Windows Services Control Panel, or the Registered Servers pane in SSMS would also have worked.



And here's my visual confirmation (I could have simply tried to reconnect using SQLCMD, but I think we're all tired of command prompt screenshots by this point). Note that I used the system function **IS\_SRVROLEMEMBER** to check my status. Looking at the properties of my login, or of the sysadmin role in Object Explorer would have shown this as well.



I hope this helps someone else in the same predicament!

Article originally appeared on AmeriTeach Instructor Blogs (http://blogs.ameriteach.com/).

See website for complete article licensing information.