

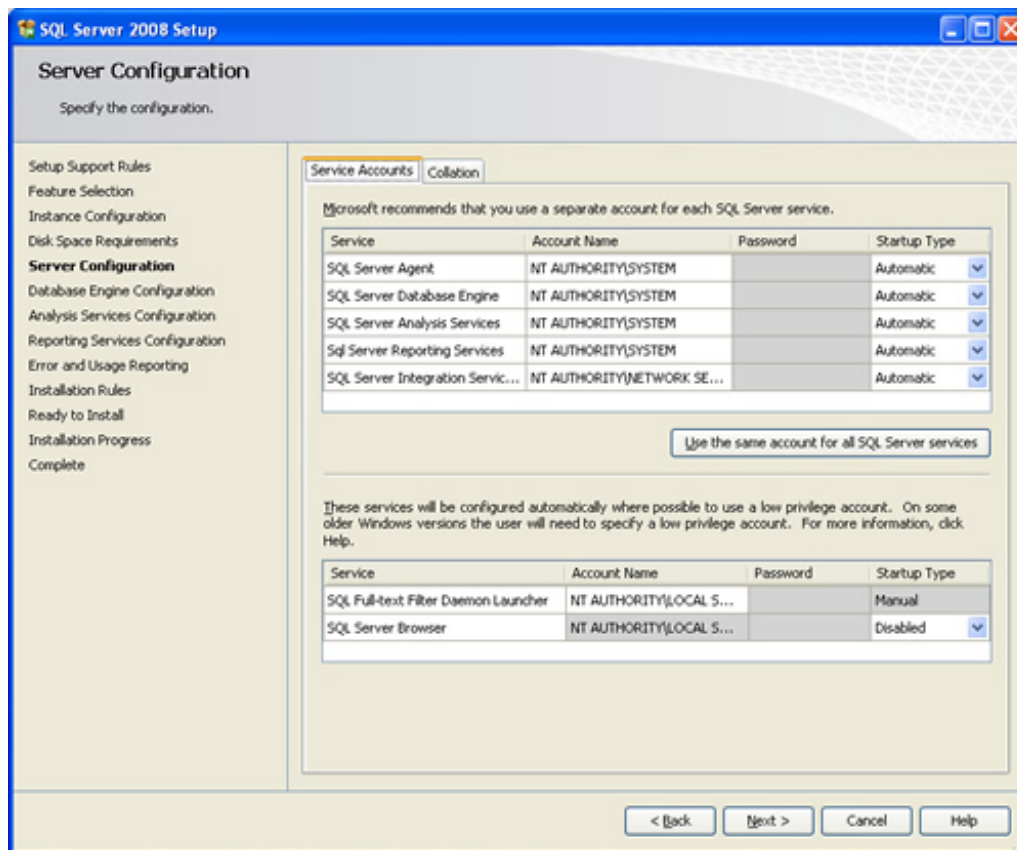
Fun With SQL Server 2008 Login Properties

Posted At : January 24, 2011 6:03 PM | Posted By : Mark Kruger

Related Categories: MS SQL Server

Most of you probably know you can run an instance of SQL server as the "local system" account and it works fine. You can also run an instance of SQL server as a domain *or* a local user account and (if the permissions are set correctly) that is also fine. But if you have never actually *installed* MSSQL Server 2008 you may not know of a change in how the installation routine "suggests" you run SQL server. The 2008 install really *wants* an account to run under. It no longer uses "local system" as the default account. Instead the account area is blank. You can, of course, specify the local system account but it's no longer obvious. So what sometimes happens is that the install user scratches his head and then uses what he knows.

For example, let's say you are managing a colo server and your hosting company has issued you the username "developer345". If you install SQL server you will eventually get to the screen where you have to make this decision:



You might just enter developer345 (the username the host has given you) into the field and choose "use the same account for all SQL services". The user would be granted "log on as a service" rights automatically and the server would function as you expect it to. However, you may have just laid a trap for yourself without realizing it.

The Gotcha

Remember, "developer345" is a username issued by the host. Besides the possible security implications, the host is going to want to periodically enforce a password change policy. In other words your hosting company is going to dictate that you change

your password... say every 6 weeks. If the host does not do this, then frankly *you* should enforce this policy (the Muse said somewhat hypocritically). So when you change the password you can *easily forget* that SQL is also running under those credentials with the *old* password. Why wouldn't you notice right away? Because SQL server is quite a happy camper and runs forever without the need to restart. Until it's restarted it is going to tool along happily servicing your requests with the old credentials and none the wiser.

When MS SQL is finally restarted your system is going to fail. The engine will fail to restart and you will end up scratching your head. It was running for weeks after all. You may not recall that the password has been changed.

Advice

For my part, I like to set up a completely separate account on the local SAM for the SQL services. In fact I often call it "SQLservices". When it comes time to change the password, the account name pokes my brain and reminds me that it is likely used for (guess what) SQL services. That little reminder is what I need in order to remember to change the password both in the SAM *and* in the login properties.