Login triggers, data file defrags, and more

Edited by Nancy Michell

Setting the service account

Q In SQL Server 2000, I used to set the service account for the SQL Server Engine and Agent using the Services applet in Administrative Tools. Now I hear that in SQL Server 2005 I'm supposed to use the Configuration Manager tool. Why can't I just keep using the Windows tools?

A SQL Server 2005 is built to be more secure than previous versions. In many shops, users would just set internal accounts, such as LocalSystem, to run SQL Server. But these accounts often either have more or fewer rights and permissions under Windows than they need. You should create a Windows account with no elevated privileges to run the SQL Server 2005 Engine and Agent services. If you select these ac-

Tip: more secure passwords

The SQL Server 2000 engine maintains two copies of each SQL Server login password. One version is the actual password supplied by the user; the second is the password in all uppercase letters.

This practice helps in case-insensitive validation of passwords, because a user can log in using either mixed-case or uppercase and be granted access to the server. However, this convenience comes with a catch. Saving passwords in all uppercase makes brute-force password-guessing attacks easier by reducing the number of possible passwords.

SQL Server 2005 stores only the original copy of the password. A password entered by a user must match the password stored on the server. If it doesn't match, the login fails and the user is denied access. If the precise case of the password characters is forgotten, the password must be reset.

Assuming a user's login name is SQLCOMMUNITY, you can reset his SQL Server password using the following command:

```
Use Master;
ALTER LOGIN SQLCOMMUNITY WITH PASSWORD = 'k3t9h4s8wJF7t';
```

This command would reset the password for SQLCOMMUNITY SQL Server login to "k3t9h4s8wJF7t".

counts with the Configuration Manager, they will automatically be granted the proper rights and permissions in both SQL Server and the operating system. If you use the Windows tools to manage the SQL Server services, you might not grant the proper rights, or you may grant too many.

For more details, see the tip Changing the Service Account.

Who's logging onto my server?

Q I want to know who is logging onto my server and when. I also want to restrict some users to certain time periods and would like to know how to fire a trace to track user activity. Is any of this possible?

A Yes, you can do all of these things with SQL Server 2005 if you have Service Pack 2 installed.

SQL Server 2005 allows you to create login triggers that can fire a T-SQL or stored procedure in response to a LOGON event. You can use a login trigger to audit and control users by tracking login activity, restricting logins to SQL Server, or limiting the number of sessions for specific logins. Note that the event is fired only after a login is successfully authenticated, but just before the user session is actually established. Therefore, all messages originating from inside the trigger (such as messages or errors) from the PRINT statement are sent to the SQL Server error log. If the authentication happens to fail for a login, then the Logon triggers are not fired.

The following example shows how you can create a login trigger and send a message to the SQL Server error log as soon as any user logs in:

```
ALTER TRIGGER Ops_Login
ON ALL SERVER
AFTER LOGIN
AS
PRINT SUSER_SNAME() + ' has just logged in
to ' + LTRIM(@@ServerName) + ' SQL Server at
'+LTRIM(getdate())
GO
```

To view all the triggers set at the server level, use the following query:

```
SELECT * FROM sys.server_triggers;
```

Best practices for defragging

Q What's the best way to fix data file fragmentation in SQL Server? If we use the defragmentation tools in Windows, they treat the SQL data file as a whole and will not defragment it granularly.

A You could back up the database and then restore it. If the space exists for a contiguous file, the database should then be written contiguously. That said, it is normally not worth the downtime to try to defrag the physical files. Typically there isn't much external fragmentation anyway. It's more helpful to regularly reindex your data to reduce the internal fragmentation as much as possible. This will maximise the effectiveness of the read-aheads and the amount of data that can be buffered.

The most important factors for efficient disk I/O are making sure the disk alignment and RAID configuration is correct, scaling your disk arrays to properly handle the I/O load and maintaining proper layout of the Log, Data, TempD and backup files. If you avoid using auto-grow and auto-shrink as your primary method for sizing data files, you'll reduce the number of volume-level file fragments created. For instance, performing 10 auto-grows of 500MB each would probably add 10 new physical file fragments. In contrast, a single manual grow of 5GB will add only one. ■

Thanks to the following Microsoft IT pros for their technical expertise:
Cary Gottesman, Saleem Hakani, Trayce Jordan, Peter Kalbach, Al Noel, Uttam Parui, Amber Sitko and Buck Woody.

Tip: changing the service account

When the SQL Server service login account is configured with a Windows NT account, SQL Server sets Windows user rights and permissions on several files, folders and registry keys. You can also set the SQL Server service account from the Services console within Administrative Tools. However, when you do this through Services, the rights and permissions are not set and you may run into serious issues owing to lack of proper security settings on the aforementioned SQL Server and Windows items.

Therefore, it's strongly recommended that you use SQL Server Configuration Manager and not the Services console when changing SQL Server or the SQL Server Agent service account. However, if you've already made changes to the account using the Services console, you can still fix this problem.

Step 1: Apply full permissions on the following registry keys and its subkeys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
Microsoft SQL Server\90
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
Microsoft SQL Server\<MSSQL.x>
```

Step 2: Set full control for the startup account for the MSSQLServer service and the SQLServerAgent service (either a local Windows NT account or a domain Windows NT account) on this NTFS folder:

```
Drive:\Program Files\Microsoft SQL
Server\<MSSQL.1>\MSSQL
```

Instead of doing this manually, however, it's recommended that you use SQL Server Configuration Manager for making changes to SQL Server/Agent service accounts.