Moving Default Directories in SQL Server

Posted on April 27, 2011 by David Masciangelo

6

Installing SQL Server to default directories can be difficult to maintan primarily because Microsoft changes the default path of the data directory at each major release. For example, the default paths to database files for the default instance in SQL 2005-2008 R2 are:

- *SQL 2005 – %Program Files%\Microsoft SQL Server\MSSQL.1\MSSQL\Data*
- *SQL 2008 – %Program Files%\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Data*
- *SQL 2008 R2 – %Program Files%\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA*

Performing a database restore from one server to another of different versions can add complexity to the restore command since the destination path is not consistent. Also, after an in-place upgrade of SQL Server the database files are not moved from the previous default path to the new so the server may have mdf and ldf files in a directory with a confusingly named path.

Using a Standard Directory

I've standardized my installations to use the same path for all mdf and ldf file locations. This simplifies scripting restores and reduces complexity during migrations since all data files across all servers are in the same path. I always keep my log files on a separate drive from the data files so they are in the same paths on different drives, for example:

- *Logs – y:\mssql\%instanceName%\data*
- *Data – z:\mssql\%instanceName%\data*

SQL Server 2008 and 2008 R2 make configuring the new paths simple at install. On the Database Engine Configuration – Data Directories page of the install wizard you can configure your paths as you see fit.

A Little Housekeeping

Once you've installed and all your mdf and ldf files are located in the new and improved standard locations I found three additional directories I wanted to standardize and move to my default path:

1. database engine default backup directory
   - %SQL Install Directory%\MSSQL10_50.MSSQLSERVER\MSSQL\BACKUP
2. sql agent working directory
   - %SQL Install Directory%\MSSQL10_50.MSSQLSERVER\MSSQL\JOBS
3. sql agent log directory
   - %SQL Install Directory%\MSSQL10_50.MSSQLSERVER\MSSQL\LOG

The configuration for these is not available via the SSMS GUI or during install. The only way to change them are directly in the registry or via a t-sql script that changes the registry.

Here's a script that will change them to whatever path you choose and is instance aware so the correct registry key is updated for any named instance. Note – this script uses the xp_instance _regwrite undocumented extended procedure so it may not be supported in the future.

```
USE [msdb]
GO

--      change the @instancename to whatever named instance this is for, the rest will sort itself out  -----------------
DECLARE @instancename nvarchar(255) = N'mssqlserver'    -- use 'mssqlserver' for the default instance

DECLARE @mssqlpath nvarchar(255) = N'z:\mssql\' + @instancename
DECLARE @dbengine_defaultbackup_path nvarchar(255) = @mssqlpath + N'\backup'
DECLARE @sqlagent_errorlogfile nvarchar(255) = @mssqlpath + N'\log\SQLAGENT.OUT'
DECLARE @sqlagent_workingdirectory_path nvarchar(255) = @mssqlpath + N'\jobs'

--SELECT @dbengine_defaultbackup_path, @sqlagent_workingdirectory_path, @sqlagent_errorlogfile

--      update the database engine's default backup directory path      -------------------------------------------------
EXECUTE [master].dbo.xp_instance_regwrite
          N'HKEY_LOCAL_MACHINE'
        , N'SOFTWARE\Microsoft\MSSQLServer\MSSQLServer'
        , N'BackupDirectory'
        , N'REG_SZ'
        , @dbengine_defaultbackup_path

--      update the SQL Agent working directory  -------------------------------------------------------------------------
EXECUTE [master].dbo.xp_instance_regwrite
          N'HKEY_LOCAL_MACHINE'
        , N'SOFTWARE\Microsoft\MSSQLServer\SQLServerAgent'
        , N'WorkingDirectory'
        , N'REG_SZ'
        , @sqlagent_workingdirectory_path

--      update the SQL Agent log file location  -------------------------------------------------------------------------
EXECUTE msdb.dbo.sp_set_sqlagent_properties @errorlog_file = @sqlagent_errorlogfile

--      update the setup's SQL Data Root path   -------------------------------------------------------------------------
EXECUTE [master].dbo.xp_instance_regwrite
          N'HKEY_LOCAL_MACHINE'
        , N'SOFTWARE\Microsoft\MSSQLServer\Setup'
        , N'SQLDataRoot'
        , N'REG_SZ'
        , @mssqlpath
```

Modify the script to use whatever path is your default. SQL Server and SQL Agent services will need to be restarted to begin using the new file locations. Once the new locations are in use you can delete the old directories.

Update

After attempting to upgrade one of my servers to SQL Server 2012 I found that the installer would error on installing the database engine. The messge was:

> *"The Database Engine system data directory in the registry is not valid."*

I found the installer looks in HKLM\software\microsoft\microsoft sql server\%instance_name%\setup\SQLDataRoot to discover the path to the system databases. This key was created during the 2008 R2 installation and the files were subsequently moved so what was created at install time is no longer correct.

If you modify this key to the path configured in the @mssqlpath from the script above the 2012 installer works just fine. I've also modified the script, the last section makes the appropriate change to prevent this exception in the future.

2 Responses to *Moving Default Directories in SQL Server*

Victor Barajas *says:*
January 23, 2014 at 5:06 pm

Thanks for this article. I've found almost nothing about this topic on the web and the ones I've found are not as accurate as yours. Thanks!

Sean *says:*
June 11, 2015 at 7:58 am

Thanks for your article, as Victor has stated this is the best one. I might have to use your script, today. As SQL is installed on the SAN that's being decommissioned.