SQL Server Connection Strings

Written by **Bill Graziano** on **14 November 2007** | **7 Comments**
Tagged with **Security**, **.NET**

Some common and not so common connection strings for the .NET SqlConnection object. The article includes .NET sample code and some tricks to increase the supportability of your application.

Trusted Authentication

```
Data Source=ServerName; Initial Catalog=DatabaseName; Integrated
Security=SSPI;
```

Trusted authentication uses the security credentials of the current user to make the connection to SQL Server. SQL Server uses Windows (or Active Directory) to validate the current user. *ServerName* can be the name of a server or the name of a SQL Server instance such as Server1\Instance2. *ServerName* can also be expressed as an IP address. SSPI stands for Security Support Provider Interface (in you were curious).

SQL Server Security Authentication

```
Data Source=ServerName; Initial Catalog=DatabaseName; User
Id=UserName; Password=UserPassword;
```

In SQL Server authentication SQL Server stores the username and password. *ServerName* can be the name of a server or the name of a SQL Server instance such as Server1\Instance2. *ServerName* can also be expressed as an IP address.

Setting the Application Name

```
Data Source=ServerName; Initial Catalog=DatabaseName; Integrated
Security=SSPI; Application Name=MyAppName;
```

I often set the Application Name when I construct connections strings. Whatever text you assign to Application Name will appear in a couple of different places:

- It will be displayed in Profiler under the Application Name column.
- It will be shown in the output of **sp_who2** in the Program Name column.
- It will be shown in the Activity Monitor in the Application column. You can get to the Activity Monitor in SQL Server Management Studio by Management -> Activity Monitor.

- It will appear in the program_name column if you select from master.dbo.sysprocesses (for SQL Server 2000)
- It will appear int he program_name column if you select from sys.dm_exec_sessions (for SQL Server 2005 and later).

Setting the application name makes it very easy to find out what applications are issuing particular SQL statements against my database. Setting the application name can also lead to an increase in the number of connections to your SQL Server. Each client that uses connection pooling will create one pool inside each application per unique connection string. If you use multiple application names you have the possibility to increase the number of pools and thus the number of connections to SQL Server. I've always found it more beneficial to have the application name than to have a few less connections to my database servers.

Using MARS (Multiple Active Result Sets)

```
Data Source=ServerName; Initial Catalog=DatabaseName; Integrated
Security=SSPI; MultipleActiveResultSets=True;
```

If you want to use **MARS** you'll need to enable it in the connection string.

Sample .NET code

There are two common ways to create a connection string in .NET. The first is to use an explicit connection string.

```
SqlConnection conn = new SqlConnection();
conn.ConnectionString = "Data Source=L40; Initial Catalog=master;
Integrated Security=SSPI;";
```

The second is to use the Connection String Builder object in .NET to construct a connection string.

```
SqlConnectionStringBuilder csb = new SqlConnectionStringBuilder();
csb.DataSource = "L40";
csb.InitialCatalog = "master";
csb.IntegratedSecurity = true;

SqlConnection conn = new SqlConnection();
conn.ConnectionString = csb.ToString();
```