



SQL Server 2008 connection strings

| .NET libraries | OLEDB providers | ODBC drivers | Wrappers and others |
|---|--|---|--|
| <ul style="list-style-type: none"> .NET Framework Data Provider for SQL Server Context Connection | <ul style="list-style-type: none"> SQL Server Native Client 10.0 OLE DB Provider SQLXML 4.0 OLEDB Provider | <ul style="list-style-type: none"> SQL Server Native Client 10.0 ODBC Driver | <ul style="list-style-type: none"> .NET Framework Data Provider for OLE DB .NET Framework Data Provider for ODBC |



.NET Framework Data Provider for SQL Server

Standard Security

```
Server=myServerAddress; Database=myDataBase; User Id=myUsername; Password=myPassword;
```

[SQL Server 7.0](#) [SQL Server 2012](#) [SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)

Trusted Connection

```
Server=myServerAddress; Database=myDataBase; Trusted_Connection=True;
```

[SQL Server 7.0](#) [SQL Server 2012](#) [SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)

Connection to a SQL Server instance

The *server/instance* name syntax used in the *server* option is the same for all SQL Server connection strings.

```
Server=myServerName\myInstanceName; Database=myDataBase; User Id=myUsername; Password=myPassword;
```

[SQL Server 7.0](#) [SQL Server 2012](#) [SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)



Trusted Connection from a CE device

A Windows CE device is most often not authenticated and logged in to a domain but it is possible to use SSPI or trusted connection and authentication from a CE device using this connection string.

```
Data Source=myServerAddress; Initial Catalog=myDataBase; Integrated Security=SSPI; User ID=myDomain\myUsername; Password=myPassword;
```

Note that this will *only* work on a CE device.

[SQL Server 7.0](#) [SQL Server 2012](#) [SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)

Connect

- SQL Server ×89
- SQL Server 2012 ×38
- SQL Server 2008 ×33**
- SQL Server 2005 ×51
- SQL Server 2000 ×39
- SQL Server 7.0 ×39

Q&A [ask question](#)

[FailoverPartner not connecting with primary is down](#)

[Unable to make any .NET connection to MSSQL Server 2008 Express](#)

[Error: A native exception has occurred in SQLCompact 3.5 under Windows Embedded Compact 7](#)

[Connect from Excel to SQL not working with id and password](#)

[SQL 2008 R2 on Server 2012 R2 connection string issues](#)

Articles [read all](#)

[All SQL Server SqlConnection Properties](#)

[Application Name for SQL Server Connections](#)

[SQL Server Data Types Reference](#)

[When to use the SQL Native Client](#)

[Download SQL Server Native Client](#)



Connect via an IP address

```
Data Source=190.190.200.100,1433; Network Library=DBMSSOCN;  
Initial Catalog=myDataBase; User ID=myUsername; Password=myPassword;
```

DBMSSOCN=TCP/IP is how to use TCP/IP instead of Named Pipes.
At the end of the Data Source is the port to use. 1433 is the default
port for SQL Server. Read more [here](#).

SQL Server 2008 SQL Server 2005 SQL Server 2000
SQL Server 7.0 SQL Server 2012

[Network Protocol for SQL Server Connection](#)

[SQL Server 2008 Data Types Reference](#)

[SQL Server 2000 Data Types Reference](#)

[SQL Server 2005 Data Types Reference](#)

[SQL Server 2012 Data Types Reference](#)

Enable MARS

```
Server=myServerAddress; Database=myDataBase; Trusted_Connection=True;  
MultipleActiveResultSets=true;
```

SQL Server 2012 SQL Server 2008 SQL Server 2005

Attach a database file on connect to a local SQL Server Express instance

```
Server=.\SQLEXPRESS; AttachDbFilename=C:\MyFolder\MyDataFile.mdf; Database=dbname;  
Trusted_Connection=Yes;
```

Why is the Database parameter needed? If the named database have already
been attached, SQL Server does not reattach it. It uses the attached database
as the default for the connection.

SQL Server 2008 SQL Server 2005
SQL Server 2012

Attach a database file, located in the data directory, on connect to a local SQL Server Express instance

```
Server=.\SQLEXPRESS; AttachDbFilename=|DataDirectory|mydbfile.mdf; Database=dbname;  
Trusted_Connection=Yes;
```

Why is the Database parameter needed? If the named database have already
been attached, SQL Server does not reattach it. It uses the attached database
as the default for the connection.

SQL Server 2008 SQL Server 2005
SQL Server 2012

Database mirroring

If you connect with ADO.NET or the SQL Native Client to a database that is being mirrored, your
application can take advantage of the drivers ability to automatically redirect connections when a
database mirroring failover occurs. You must specify the initial principal server and database in the
connection string and the failover partner server.

```
Data Source=myServerAddress; Failover Partner=myMirrorServerAddress;  
Initial Catalog=myDataBase; Integrated Security=True;
```

There is ofcourse many other ways to write the connection string using database
mirroring, this is just one example pointing out the failover functionality. You can
combine this with the other connection strings options available.

SQL Server 2008 SQL Server 2005
SQL Server 2012

Asynchronous processing

A connection to SQL Server that allows for the issuing of async requests through ADO.NET objects.

```
Server=myServerAddress; Database=myDataBase; Integrated Security=True;  
Asynchronous Processing=True;
```

See also the [List of all SqlConnection connection string properties](#).

SQL Server 2012 SQL Server 2008 SQL Server 2005

Using an User Instance on a local SQL Server Express instance

The User Instance functionality creates a new SQL Server instance on the fly during connect. This
works only on a local SQL Server instance and only when connecting using windows authentication
over local named pipes. The purpose is to be able to create a full rights SQL Server instance to a
user with limited administrative rights on the computer.

```
Data Source=.\SQLEXPRESS; Integrated Security=true;  
AttachDbFilename=C:\MyFolder\MyDataFile.mdf; User Instance=true;
```

To use the User Instance functionality you need to enable it on the SQL Server. This is
done by executing the following command: `sp_configure 'user instances enabled', '1'`. To
disable the functionality execute `sp_configure 'user instances enabled', '0'`.

SQL Server 2005
SQL Server 2008

↳ **Problems connecting?** Get answer in the [SQL Server 2008 Q & A forum](#) →

⚙️ SQL Server Native Client 10.0 OLE DB Provider

Standard security

```
Provider=SQLNCLI10; Server=myServerAddress; Database=myDataBase; Uid=myUsername;  
Pwd=myPassword;
```

Are you using SQL Server 2008 Express? Don't miss the server name syntax `Servername\SQLEXPRESS` where you substitute `Servername` with the name of the computer where the SQL Server 2008 Express installation resides.

[SQL Server 2005](#) [SQL Server 2000](#)
[SQL Server 7.0](#) [SQL Server 2008](#)

[When to use SQL Native Client?](#) 

Trusted connection

```
Provider=SQLNCLI10; Server=myServerAddress; Database=myDataBase;  
Trusted_Connection=yes;
```

Equivalent key-value pair: "Integrated Security=SSPI" equals
"Trusted_Connection=yes"

[SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)
[SQL Server 7.0](#)

Connecting to an SQL Server instance

The syntax of specifying the server instance in the value of the server key is the same for all connection strings for SQL Server.

```
Provider=SQLNCLI10; Server=myServerName\theInstanceName; Database=myDataBase;  
Trusted_Connection=yes;
```

[SQL Server 7.0](#) [SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)

Prompt for username and password

This one is a bit tricky. First you need to set the connection object's `Prompt` property to `adPromptAlways`. Then use the connection string to connect to the database.

```
oConn.Properties("Prompt") = adPromptAlways
```

```
oConn.Open "Provider=SQLNCLI10;Server=myServerAddress;DataBase=myDataBase;"
```

[SQL Server 7.0](#) [SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)

Enable MARS

```
Provider=SQLNCLI10; Server=myServerAddress; Database=myDataBase;  
Trusted_Connection=yes; MARS_Connection=True;
```

[SQL Server 2008](#) [SQL Server 2005](#)

Encrypt data sent over network

```
Provider=SQLNCLI10; Server=myServerAddress; Database=myDataBase;  
Trusted_Connection=yes; Encrypt=yes;
```

[SQL Server 7.0](#) [SQL Server 2008](#) [SQL Server 2005](#) [SQL Server 2000](#)

Attach a database file on connect to a local SQL Server Express instance

```
Provider=SQLNCLI10; Server=.\SQLEXPRESS; AttachDbFilename=c:\asd\qwe\mydbfile.mdf;  
Database=dbname; Trusted_Connection=Yes;
```

Why is the Database parameter needed? If the named database have already been attached, SQL Server does not reattach it. It uses the attached database as the default for the connection.

[SQL Server 2005](#)
[SQL Server 2008](#)

Attach a database file, located in the data directory, on connect to a local SQL Server Express instance

```
Provider=SQLNCLI10; Server=.\SQLEXPRESS;  
AttachDbFilename=|DataDirectory|mydbfile.mdf; Database=dbname;  
Trusted_Connection=Yes;
```

Why is the Database parameter needed? If the named database have already been attached, SQL Server does not reattach it. It uses the attached database as the default for the connection.

SQL Server 2005

SQL Server 2008

Database mirroring

If you connect with ADO.NET or the SQL Native Client to a database that is being mirrored, your application can take advantage of the drivers ability to automatically redirect connections when a database mirroring failover occurs. You must specify the initial principal server and database in the connection string and the failover partner server.

```
Provider=SQLNCLI10; Data Source=myServerAddress;  
Failover Partner=myMirrorServerAddress; Initial Catalog=myDataBase;  
Integrated Security=True;
```

There is ofcourse many other ways to write the connection string using database mirroring, this is just one example pointing out the failover functionality. You can combine this with the other connection strings options available.

SQL Server 2005

SQL Server 2008

SQLXML 4.0 OLEDB Provider

Using SQL Server Native Client provider (SQLNCLI10)

```
Provider=SQLXMLOLEDB.4.0; Data Provider=SQLNCLI10; Data Source=myServerAddress;  
Initial Catalog=myDataBase; User Id=myUsername; Password=myPassword;
```

SQL Server 2008

.NET Framework Data Provider for OLE DB

Use an OLE DB provider from .NET

```
Provider=any oledb provider's name; OleDbKey1=someValue; OleDbKey2=someValue;
```

See the respective OLEDB provider's connection strings options. The .net OleDbConnection will just pass on the connection string to the specified OLEDB provider. Read more [here](#).

SQL Server Native Client 10.0 ODBC Driver

Standard security

```
Driver={SQL Server Native Client 10.0}; Server=myServerAddress;  
Database=myDataBase; Uid=myUsername; Pwd=myPassword;
```

Are you using SQL Server 2008 Express? Don't miss the server name syntax Servername\SQLEXPRESS where you substitute Servername with the name of the computer where the SQL Server 2008 Express installation resides.

SQL Server 2005 SQL Server 2008

SQL Server 7.0 SQL Server 2008

[When to use SQL Native Client?](#)

Trusted Connection

```
Driver={SQL Server Native Client 10.0}; Server=myServerAddress;  
Database=myDataBase; Trusted_Connection=yes;
```

Equivalent key-value pair: "Integrated Security=SSPI" equals "Trusted_Connection=yes"

SQL Server 2008 SQL Server 2005 SQL Server 2008

SQL Server 7.0

Connecting to an SQL Server instance

The syntax of specifying the server instance in the value of the server key is the same for all connection strings for SQL Server.

```
Driver={SQL Server Native Client 10.0}; Server=myServerName\theInstanceName;  
Database=myDataBase; Trusted_Connection=yes;
```

SQL Server 7.0 SQL Server 2008 SQL Server 2005 SQL Server 2000

Prompt for username and password

This one is a bit tricky. First you need to set the connection object's Prompt property to adPromptAlways. Then use the connection string to connect to the database.

```
oConn.Properties("Prompt") = adPromptAlways  
  
oConn.Open "Driver={SQL Server Native Client  
10.0};Server=myServerAddress;Database=myDataBase;"
```

SQL Server 7.0 SQL Server 2008 SQL Server 2005 SQL Server 2000

Enable MARS

```
Driver={SQL Server Native Client 10.0}; Server=myServerAddress;  
Database=myDataBase; Trusted_Connection=yes; MARS_Connection=yes;
```

SQL Server 2008 SQL Server 2005

Encrypt data sent over network

```
Driver={SQL Server Native Client 10.0}; Server=myServerAddress;  
Database=myDataBase; Trusted_Connection=yes; Encrypt=yes;
```

SQL Server 7.0 SQL Server 2008 SQL Server 2005 SQL Server 2000

Attach a database file on connect to a local SQL Server Express instance

```
Driver={SQL Server Native Client 10.0}; Server=.\SQLEXPRESS;  
AttachDbFilename=c:\asd\qwe\mydbfile.mdf; Database=dbname; Trusted_Connection=Yes;
```

Why is the Database parameter needed? If the named database have already been attached, SQL Server does not reattach it. It uses the attached database as the default for the connection.

SQL Server 2005
SQL Server 2008

Attach a database file, located in the data directory, on connect to a local SQL Server Express instance

```
Driver={SQL Server Native Client 10.0}; Server=.\SQLEXPRESS;  
AttachDbFilename=|DataDirectory|mydbfile.mdf; Database=dbname;  
Trusted_Connection=Yes;
```

Why is the Database parameter needed? If the named database have already been attached, SQL Server does not reattach it. It uses the attached database as the default for the connection.

SQL Server 2005
SQL Server 2008

Database mirroring

If you connect with ADO.NET or the SQL Native Client to a database that is being mirrored, your application can take advantage of the drivers ability to automatically redirect connections when a database mirroring failover occurs. You must specify the initial principal server and database in the connection string and the failover partner server.

```
Driver={SQL Server Native Client 10.0}; Server=myServerAddress;  
Failover_Partner=myMirrorServerAddress; Database=myDataBase;  
Trusted_Connection=yes;
```


There is ofcourse many other ways to write the connection string using database mirroring, this is just one example pointing out the failover functionality. You can combine this with the other connection strings options available.

SQL Server 2005
SQL Server 2008

Please note if you are using TCP/IP (using the network library parameter) and database mirroring, including port number in the address (formed as servername,portnumber) for both the main server and the failover partner can solve some reported issues.

Use an ODBC driver from .NET

```
Driver={any odbc driver's name}; OdbcKey1=someValue; OdbcKey2=someValue;
```

See the respective ODBC driver's connection strings options. The .net OdbcConnection will just pass on the connection string to the specified ODBC driver. Read more [here](#) .

Context Connection

Context Connection

Connecting to "self" from within your CLR stored procedure/function. The context connection lets you execute Transact-SQL statements in the same context (connection) that your code was invoked in the first place.

```
C#
using(SqlConnection connection = new SqlConnection("context connection=true"))
{
    connection.Open();
    // Use the connection
}
```

```
VB.Net
Using connection as new SqlConnection("context connection=true")
    connection.Open()
    ' Use the connection
End Using
```

[SQL Server 2012](#) [SQL Server 2008](#) [SQL Server 2005](#)

Connect

| | |
|---------------------------------|---------------------------------|
| SQL Server | SQL Server 2012 |
| SQL Server 2008 | SQL Server 2005 |
| SQL Server 2000 | SQL Server 7.0 |

Articles [read all](#)

| | |
|---|---|
| All SQL Server SqlConnection Properties | Application Name for SQL Server Connections |
| SQL Server Data Types Reference | When to use the SQL Native Client |
| Download SQL Server Native Client | Network Protocol for SQL Server Connection |
| SQL Server 2008 Data Types Reference | SQL Server 2000 Data Types Reference |
| SQL Server 2005 Data Types Reference | SQL Server 2012 Data Types Reference |

Didn't find your connection string?

Start over from the [connection string reference index](#) - or try a [search](#)!

In the [Q&A forums](#) you can [ask your own question](#) and let somebody help you.

The [knowledge articles](#) contains solutions and guides.

