Controlling ODBC data sources with the connection string in ColdFusion 5

Michael Stillman
Senior Technical Writer

Macromedia Instructional
Media Development Group

The **connection string** was added to ColdFusion 5 in response to user requests for greater control when connecting to ODBC data sources. You can use the connection string to do tasks such as:

- Specifying connection attributes that cannot be defined in the odbc.ini settings
- Limiting the number of data source names (DSNs)
- Making database connection properties easier to tune
- Making ODBC connections dynamically when there is no data source defined in the odbc.ini settings

Background Information

In ColdFusion 4.x, you could specify three Open Database Connectivity (ODBC) standard attributes: data source, user name, and password. This is because of the ColdFusion Application Server ODBC client interface, which used the SQLConnect API, which accepts only these parameters. However, some ODBC data sources permit additional database-specific connection attributes.

About the Connection String

ColdFusion 5 allows you to specify a connection string for ODBC data sources so that you can pass these attributes as name-value pairs (using the ODBC SQLDriverConnect API behind the scenes). For example, many applications that connect to SQL Server pass the APP="appname" and WSID="workstation_id." By including these parameters, a database administrator (DBA) can identify which applications are connected to the database server and which computers are running those applications.

Using the connection string in ColdFusion 5, you can also connect to ODBC databases that do not have a defined data source in the odbc.ini settings. This is called a **dynamic connection**.

You cannot readily specify in the connection string ColdFusion-specific settings, such as Maintain Connection, Buffer Size, and Limit Connections; these parameters are managed by the ColdFusion Data Access Layer and are independent of the connection string. The attributes that you specify in the connection string are database-specific. Consult your database's documentation for details. For example, to learn more about supported SQL Server settings, consult the SQLDriverConnect topic in Books Online; for Merant driver settings, see the **odbcref.pdf** file that ships with ColdFusion 5.

Using the Connection String

You can use connection string functionality in the ColdFusion Administrator and in the following CFML tags that interact with data sources:

- `cfquery`
- `cfinsert`
- `cfupdate`
- `cfstoredproc`
- `cfgridupdate`

In the ColdFusion Administrator, enter the keyword name-value pairs separated by semicolons. In CFML, use the `connectstring` attribute in the following format:

```
connectstring="keyword1=value1;keyword2=value2;…keywordn=valuen"
```

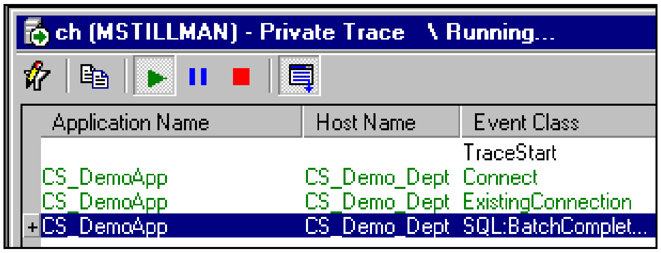
The `connectstring` attribute overrides any connection string values defined in the ColdFusion Administrator settings for a data source. To enable a DBA to identify which applications and workstations are connected to the database server, add the following text to the connection string option in the ColdFusion Administrator:

```
APP=appname;WSID=workstation_ID
```

For example, using a SQL Server data source named **2pubs** that connects to the **pubs** database, you can specify default application and workstation ID values by adding the following text:

```
APP=CS_DemoApp;WSID=CS_Demo_Dept
```

The following figure illustrates how this information appears in a SQL Server trace (the WSID value appears in SQL Server 7 as the Host Name):

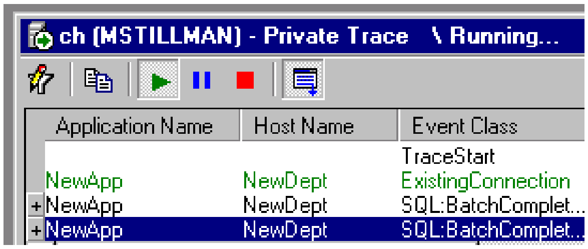


Application Name	Host Name	Event Class
		TraceStart
CS_DemoApp	CS_Demo_Dept	Connect
CS_DemoApp	CS_Demo_Dept	ExistingConnection
+CS_DemoApp	CS_Demo_Dept	SQL:BatchComplet...

You can use the `connectstring` attribute to override connection properties for a data source. For example, you can add different values to the `connectstring` attribute in a simple query:

```
APP=NewApp;WSID=NewDept">
        select      au_fname, au_lname
        from        authors
```

The new values now display in a SQL Server trace:



Application Name	Host Name	Event Class
		TraceStart
NewApp	NewDept	ExistingConnection
+NewApp	NewDept	SQL:BatchComplet...
+NewApp	NewDept	SQL:BatchComplet...

Changing the Target Database

The ability of the connection string to pass information becomes more useful when you modify the database value. By doing so, you can connect to a database other than the one specified in the data source. One advantage is that you no longer need one data source definition per database—you can have one data source *per server*. To connect to multiple databases on the same server, configure one data source and override its database setting in the `connectstring` attribute of one of the five CFML tags mentioned previously.

In the example, the 2pubs data source connects to the SQL Server pubs database. To modify the query to connect to another database, include the **database** keyword and the name of the target database in the `connectstring` attribute:

```
database=northwind">
        select      categoryname, description
        from        categories
```

By changing the value for the database keyword, you can use one data source to connect to many databases, provided that they're located on the same server. You can have multiple connections within a template, each connecting to a different database.

The connection string offers flexibility in managing the number of simultaneous connections per server. In ColdFusion 4.x, to set a limit of five simultaneous connections to a server, you would have limited that number in the ColdFusion Administrator. For example, if you had five databases on the server, you would limit each of the five data sources to one connection (because you had one data source per database). In ColdFusion 5, you can configure one data source that connects to the server and overrides the default database in the `connectstring` attribute of your template. This ensures that you get the maximum number of connections (five in this example), but you need not limit each database to one connection. At any one time, you might want three connections to Database A and two communicating with Database B.

The connection string functionality also facilitates adding new data sources. For example, in ColdFusion 4.x, if your application ran on ten servers and you added a new data source, you had to add that new data source to all ten servers for a ColdFusion template to be able to use the application. In ColdFusion 5, you can simply override the name of the database in the `connectstring` attribute.

The `connectstring` attribute overrides values in the ColdFusion Administrator in an all-or-none fashion. You cannot merely specify a keyword-value pair in your CFML tag.

For example, compare the two connectstring attributes mentioned previously with the values you configured in the ColdFusion Administrator. In the ColdFusion Administrator, you set the connection string to:

```
APP=CS_DemoApp;WSID=CS_Demo_Dept
```

In a ColdFusion template, you modified a cfquery by adding the connectstring attribute that passes two values:

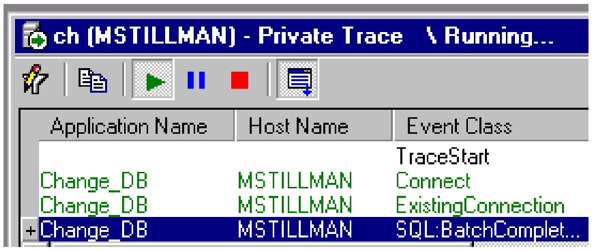
```
connectstring="APP=NewApp;WSID=NewDept"
```

You further modified it to connect to a different database:

```
connectstring="APP=Change_DB;database=northwind"
```

Notice that this `connectstring` attribute does not define any workstation ID value. The application name and database values are passed, but the workstation ID value appears as the default username (which appears in SQL Server 7 as the Host

Name):


ch (MSTILLMAN) - Private Trace \ Running...

Application Name	Host Name	Event Class
		TraceStart
Change_DB	MSTILLMAN	Connect
Change_DB	MSTILLMAN	ExistingConnection
Change_DB	MSTILLMAN	SQL:BatchComplet...

This example illustrates the importance of including all necessary information in your connectstring attribute. For example, if you specify ten values in the ColdFusion Administrator connection string and only two values in a cfquery `connectstring` attribute, you may unintentionally omit eight values.

Using Dynamic Connections

The previous examples demonstrate how you can change databases by using the connectstring attribute. Notice that such a change requires a data source. In ColdFusion 5, you can connect to an ODBC data source that is not defined in the ColdFusion Administrator (and therefore, is not in the Windows Registry). You can create this dynamic connection by entering **`dbtype="dynamic"`** in CFML:

```
dbtype="dynamic"
        connectstring="DRIVER={SQL SERVER}; SERVER=(local);
                UID=sa;PWD=;DATABASE=pubs;APP=NewApp;WSID=NewDept">
        SELECT * FROM authors
```

You must specify all required ODBC connection information using the `connectstring` attribute. You do not need to specify a data source name (DSN) for a dynamic connection; ColdFusion creates a virtual one by default named "__dynamic__" (the word "dynamic" followed and preceded by two underscores). This DSN uses the following default ColdFusion settings:

- Do not maintain database connections
- Allow an unlimited number of connections
- Do not restrict SQL operations
- Do not enable long text retrieval

To change the default ColdFusion settings, use the ColdFusion Administrator to add a DSN named __dynamic__ and adjust the settings. When you define a dynamic DSN, you can use any ODBC driver type because ColdFusion does not pass this DSN name to the ODBC Driver.

Pooling Dynamic Connections

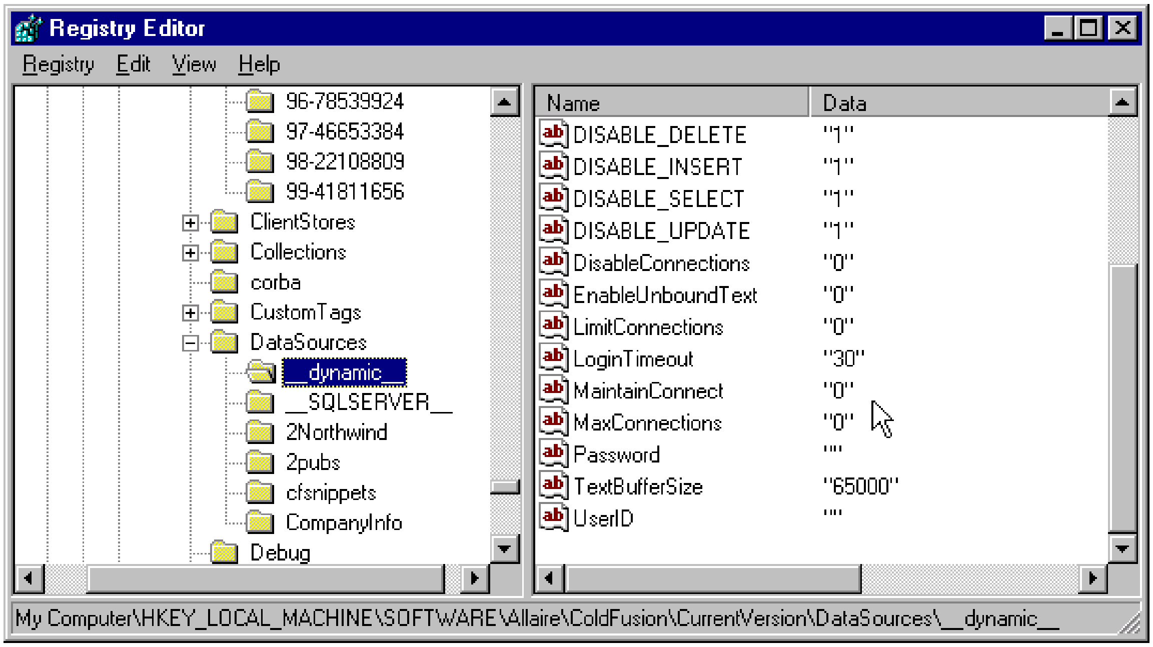
You can specify a DSN for dynamic connections to associate connection attributes and thereby create pools for different types of dynamic connections. For example, you could have a "__DB2__" DSN to dynamically connect to your DB2 data sources and an "Oracle_Dynamic" to dynamically connect to your Oracle data sources. The leading and trailing underscores are not required when you name a dynamic DSN (unless you are manually creating the __dynamic__ DSN). You can use underscores to readily distinguish dynamic connection DSNs from standard DSNs in a list of ODBC data sources, such as the Connection Summary in the ColdFusion Administrator.

For an example of why you might pool dynamic connections, consider a scenario in which you want to maintain all your SQL Server dynamic connections with a maximum of ten simultaneous connections, but you do not want to maintain your other dynamic connections. You must manually create a __dynamic__ DSN that does not maintain connections. Think of this DSN as a "virtual data source" that exists only to provide nondefault values for ColdFusion settings. You must also create a DSN for SQL Server that maintains connections (the default setting), and limits the number of connections to ten.

To manually create the __dynamic__ DSN, use the ColdFusion Administrator.

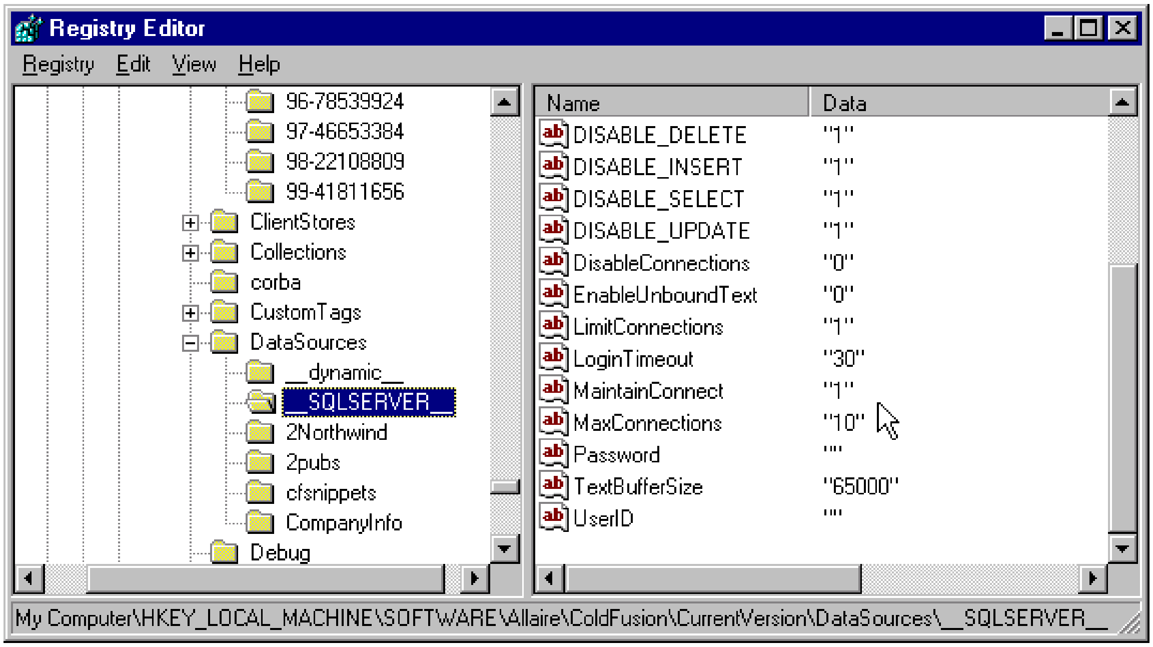
1. Under ColdFusion Settings, clear the Maintain Connections check box
2. Create a DSN named __SQLSERVER__ and specify the server name.
3. Under CF Settings, leave the Maintain Connections check box enabled and limit the number of simultaneous connections to ten

You can note these settings in the Windows Registry under \HKEY_LOCAL_MACHINE\SOFTWARE\Allaire\ColdFusion\CurrentVersion\DataSources\, as the following figures demonstrate, respectively, for __dynamic__ and __SQLSERVER__:



Name	Data
DISABLE_DELETE	"1"
DISABLE_INSERT	"1"
DISABLE_SELECT	"1"
DISABLE_UPDATE	"1"
DisableConnections	"0"
EnableUnboundText	"0"
LimitConnections	"0"
LoginTimeout	"30"
MaintainConnect	"0"
MaxConnections	"0"
Password	""
TextBufferSize	"65000"
UserID	""

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Allaire\ColdFusion\CurrentVersion\DataSources\__dynamic__



Name	Data
DISABLE_DELETE	"1"
DISABLE_INSERT	"1"
DISABLE_SELECT	"1"
DISABLE_UPDATE	"1"
DisableConnections	"0"
EnableUnboundText	"0"
LimitConnections	"1"
LoginTimeout	"30"
MaintainConnect	"1"
MaxConnections	"10"
Password	""
TextBufferSize	"65000"
UserID	""

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Allaire\ColdFusion\CurrentVersion\DataSources\__SQLSERVER__

When you make a dynamic connection and you do not specify a data source (or you specify `datasource="__dynamic__"`), ColdFusion obtains the settings from the __dynamic__ DSN and does not maintain connections. Similarly, when your dynamic connection specifies `datasource="__SQLSERVER__"`, ColdFusion maintains connections and limits the number of simultaneous connections to ten. If you have a large multipage application with several connections, the 11th and subsequent connections would queue if using the __SQLSERVER__ data source. However, if you remove the `datasource="__SQLSERVER__"` code, ColdFusion uses the __dynamic__ settings (unlimited number of connections).

Because you make a connection for every unique DSN, do not use different DSNs for each dynamic query. If you do use different DSNs, ColdFusion will create a connection pool for each unique DSN, which will incur more overhead and could affect performance.

Increasing Security of ColdFusion Applications

The `connectstring` and `dbtype=dynamic` functionality add extra measures of security to ColdFusion 5 applications. In a deployed ColdFusion application using dynamic connections, data sources are not exposed to development tools outside of ColdFusion. By using a dynamic connection, a ColdFusion template can run on multiple servers without requiring data source information on all servers. Thus, only a developer who knows what options the target database supports could use the feature. Also, if your application does not require a defined data source to connect to, it is easier to distribute and install.

If you do not want to allow ColdFusion authors to use `dbtype=dynamic` and/or `connectstring` tag attributes, you can disable these in the Basic Security section of the ColdFusion Administrator.

More Resources

Read more about connection strings in the following resources:

- ColdFusion 5: Dynamic ODBC Connection with Oracle on WinNT and Win2000 (Article 21582)

About the Author

Michael Stillman is a Senior Technical Writer in the Instructional Media Development group of Macromedia. Mike's academic degrees are in biology and psychology, which helped him get adjunct faculty gigs while becoming a full-time writer. He was a neuroscientist with the U.S. Army and has several pharmacology publications that few non-scientists will ever read. To balance his technical side, Mike enjoys softball, volleyball, and playing bass and keyboards in a garage band.