SQL Server и Oracle

Доставчиците на данни .NET за SQL Server и Oracle обезпечава ефективен пулинг на връзките с поддръжка на транзакциите. Пуловете се създават за всеки процес и не се разрушават докато процеса не завърши. Пулинга на връзките е включен по подразбиране.

OLE DB

Доставчика на данни .NET за OLE DB осъществява пулинг на връзки чрез използване на пулинг на ресурсите, предоставяни от ядрото на OLE DB.

Услугите на OLE DB са включени по подразбиране за доставчика. Те се определят от стойността на ключа HKEY_CLASSES_ROOTCLSID\OLE_DBSERVICES от системния регистър на Windows, имащ тип DWORD.

Стойности на ключа OLE_DBSERVICES

0xffffffff	всички услуги (по подразбиране)
0xfffffffe	всички услуги, освен пулинга и автоматичните връзки
0xfffffffb	всички услуги, освен клиентския курсор
0xfffffff0	всички услуги, освен пулинга, автом. връзки и кл. курсор

0x00000000 няма услуги

ако няма стойност няма ограничения. Всички услуги са отключени

Може да се преопределят зададените по премълчаване конфигурационни услуги на доставчика OLE DB, чрез използване на атрибута OLE DB Services в низа за връзка.

Стойности на ключа OLE_DBSERVICES

–1 всички услуги (по подразбиране)

–2 всички услуги, освен пулинга и автоматичните връзки

–5 всички услуги, освен клиентския курсор

–7 всички услуги, освен пулинга, автоматичните връзки и клиентския курсор

0 няма услуги

Параметрите на пулинга на връзките OLE DB управляват и следните атрибути за настройка:

- SPTimeout – дължина на интервала време в секунди, в продължение на който неизползваната връзка остава в пула, преди тя да бъде освободена. Тази продължителност може да се настройва за всеки доставчик и по подразбиране

продължава 60 секунди.

- RetryWait – продължителността на интервала време в секунди до повторен опит да се получи връзка в случай, че сървера не отговаря. Тази стойност е глобална за всички доставчици и по подразбиране е равна на 64 секунди.
- ExpBackOff – броя пъти, които трябва да се увеличи времето на чакане до повторен опит за връзка, ако текущия опит не успее. Този множител е глобален за всички доставчици и по подразбиране е равен на 2.

По подразбиране пулинга на връзките в OLE DB е включен. Той може да се управлява по три различни начина:

- даване на стойност на атрибута OLE DB Services в низа за връзка;
- редактиране на системното регистри за включване или отключване на пулинга на връзките на конкретен доставчик на данни или в глобален мащаб;
- използвайки интерфейса на приложните програми (API) OLE DB в приложенията за програмно включване и изключване на пулинга на връзките. Програмната промяна на параметрите SPTimeout и RetryWait е възможна само чрез манипулация с ключовете на системното регистри

Доставчика на данни .NET за ODBC осъществява пулинг на връзките с помощта на диспечера на драйвера ODBC (ODBC Driver Manager). Параметрите на пулинга на драйвера ODBC действат на всички приложения, използващи този драйвер, ако тези параметри не се променят в собствени приложения на ODBC.

Параметрите на пулинга на връзките ODBC управлява следните два настройващи се атрибута:

- CPTimeout – дължина на интервала време в секунди, в продължение на който неизползваната връзка остава в пула, преди тя да бъде освободена.
- RetryWait – продължителност на интервала време в секунди до повторен опит да се получи връзка в случай, че сървера не отговаря.

По подразбиране пулинга на връзките в OLE DB е включен. Включването, изключването и настройката на параметрите му може да се извърши по три различни начина:

използвайки ODBC Data Source Administrator, който за първи път се появява в ODBC 3.5 (MDAC 1.5), за включване и отключване на пулинга на целия драйвер и настройка на параметрите CPTimeout и RetryWait;

редактирайки системните регистри;

използвайки интерфейса на приложните програми (API) ODBC в ODBC приложения за ограничаване на рамките на пулинга само за обработчика на обкръжението или драйвера, а също за настройка на другите параметри на пулинга.

Настройка на параметрите на пулинга на връзките

Проблем. Необходимо е да се знаят различните параметри на пулинга на връзките и начините за управлението им.

Решение. Параметрите на пулинга на връзките на източниците на данни .NET за SQL Server, OLE DS, Oracle и ODBC могат да се управляват с помощта на атрибути в свързващия низ.

Кода на примера съдържа един метод и четири обработчика на събития:

- Form.Load – създава връзка и задава обработчик на събитието StateChange на обекта Connection и установява в изходно състояние елементите на екранната форма, управляващи параметрите на връзката. Извиква метода UpdateConnectionString(), който динамично създава низ за връзка със зададени стойности на параметрите.
- UpdateConnectionString() – този метод динамично създава низ за връзка, използвайки стойности на свойствата, въведени от потребителя в текстовото поле на екранната форма. Този метод служи за обновяване на низа за връзка след като потребителя измени състоянието на някакви елементи на управлението, определящи свойствата на низа за връзка.
- Button.Click – (бутон Отвори връзка) – отваря връзка, използвайки низа за връзка, създадена от метода UpdateConnectionString().
- Button.Click (бутон Закриване на връзката).

- Connection.StateChange – отразява информация за изходното и текущото състояние на връзките, когато състоянието се промени.

```
using System;

using System.Configuration;

using System.Windows.Forms;

using System.Data;

using System.Data.SqlClient;

//...

private void ConnectionPoolingOptionsForm_Load(object sender, System.EventArgs e)

{

conn = new SqlConnection();

conn.StateChange += new StateChangeEventHandler(conn_StateChange);

connectionStringTextBox.Text = ConfigurationSettings.AppSettings["Sql_ConnectString"];
```

```
connectTimeoutTextBox.Text = "15";

connectLifetimeTextBox.Text = "0";

minPoolSizeTextBox.Text = "0";

maxPoolSizeTextBox.Text = "100";

poolCheckBox.Checked = true;

UpdateConnectionString();

}

private void UpdateConnectionString()

{

connectionStringTextBox.Text = ConfigurationSettings.AppSettings["Sql_ConnectString"] +

"Connection Timeout = " + connectTimeoutTextBox.Text + ";" +
```

```
"Connection Lifetime = " + connectLifetimeTextBox.Text + ";" +

"Min Pool Size = " + minPoolSizeTextBox.Text + ";" +

"Max Pool Size = " + maxPoolSizeTextBox.Text + ";" +

"Pooling = " + poolCheckBox.Checked.ToString();

}

private void openButton_Click(object sender, System.EventArgs e)

{

try

{

conn.ConnectionString = connectionStringTextBox.Text;

conn.Open();

}
```

```
catch(SqlException ex)

{

MessageBox.Show("ERROR: " + ex.ToString(), "Open Connection",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

catch(InvalidOperationException ex)

{

MessageBox.Show("ERROR: " + ex.ToString(), "Open Connection",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}
```

```
private void closeButton_Click(object sender, System.EventArgs e)

{

conn.Close();

}

private void conn_StateChange(object sender, StateChangeEventArgs e)

{

connectionStateTextBox.Text =

"Connection.StateChange event occurred" + Environment.NewLine +

"OriginalState = " + e.OriginalState.ToString() + Environment.NewLine +

"CurrentState = " + e.CurrentState.ToString();

}
```

SQL Server.

Connection Lifetime – Продължителност на интервала време в секунди, при преминаването, на което връзката се унищожава. По подразбиране този атрибут има стойност 0, т.е. Връзката има максимално време на изчакване.

Connection Reset – Указва, трябва ли да се премахва връзката при отделянето му от пула. По подразбиране е true.

Enlist – Указва, следва ли автоматично да се свързва връзката с текущото съдържание на транзакцията, създаваща връзка с потока, ако този контекст на транзакцията съществува. По подразбиране е true.

Max Pool Size – Максимално допустимият брой връзки в пула. По подразбиране е равно на 100

Min Pool Size – Минималният брой връзки, поддържани в пула. По подразбиране в равен на 0.

Pooling – Указва трябва ли да се вземе връзка от пула или при необходимост го създава и добавя в пула. По подразбиране е true.

Oracle.

Connection Lifetime – Продължителност на интервала време в секунди, при преминаването, на което връзката се унищожава. По подразбиране този атрибут има стойност 0, т.е. Връзката има максимално време на изчакване.

Connection Reset – Указва, трябва ли да се премахва връзката при отделянето му от пула. По подразбиране е true.

Enlist – Указва, следва ли автоматично да се свързва връзката с текущото съдържание на транзакцията, създаваща връзка с потока, ако този контекст на транзакцията съществува. По подразбиране е true.

Max Pool Size – Максимално допустимият брой връзки в пула. По подразбиране е равно на 100

Min Pool Size – Минималният брой връзки, поддържани в пула. По подразбиране в равен на 0.

Pooling – Указва трябва ли да се вземе връзка от пула или при необходимост го създава и добавя в пула. По подразбиране е true.

OLE DB.

Доставчика .NET за OLE DB осъществява пулинг на връзките чрез използване на пулинг на ресурсите, предоставяни от компонента OLE DB Service. Може да се предефинира зададената по подразбиране конфигурация на услуги на доставчика OLE DB, указвайки стойността на атрибута OLE DB Services в низа за връзка.

Параметрите на пулинга на ресурсите OLE DB може да се управлява чрез системното регистри. Няма потребителския интерфейс за настройка на тези параметри – необходимо е директно да се редактира регистрито. Ключовете в регистрито се идентифицират по доставчика CLSID:

SQLOLEDB (SQLServer):

HKEY_CLASSES_ROOTCLSID{0C7FFl6C-38E3-11d0-97AB-00C04FC2AD98}

Microsoft.Jet.OLEDB.4.0 (Jet):

HKEY_CLASSES_ROOTCLSID{dee35070-506b-11cf-blaa-00aa00b8de95}

MSDAORA(Oracle):

HKEY_CLASSES_ROOTCLSID{e8cc4cbe-fdff-11d0-b865-00a0c9081cld}

MSDASQL (доставчик OLE DB за ODBC):

HKEY_CLASSES_ROOTCLSID{c8b522cb-5cf3-11ce-ade5-00aa0044773d}

Един от параметрите на доставчика OLE DB, намиращи се в регистрито е параметъра HKEY_CLASSES_ROOTCLSID\SPTimeout. Това е времето на очакване на сеанса на пулинга – продължение на интервала време в секунди, в продължение на който неизползвания сеанс остава в пула, преди да доведе до изтичане на времето и той да бъде закрит. Той има тип DWORD и по подразбиране е равен на 60, ако не е зададена стойност в регистрито.

Следните ключове от регистрито са глобални за всички доставчици:
HKEY_LOCAL_MACHINESOFTWAREMicrosoftDataAccessSessionPoolingRetryWait

Времето, през което обслужващите компоненти ще изчакат преди да се предприеме повторно обръщение към сървера в случай на неуспешен опит за връзка. Тази стойност има тип DWORD и по подразбиране е равно на 64 секунди, ако не е зададена стойност в регистрито.

HKEY_LOCAL_MACHINESOFTWAREMicrosoftDataAccessSession PoolingExpBackOff

Броя пъти, в които обслужващите компоненти увеличават времето за изчакване до повторен опит за връзка, ако текущия опит не завърши успешно. Тази стойност има тип DWORD и по подразбиране е равно на 64 секунди, ако не е зададена стойност в регистрито.

HKEY_CLASSES_R00TCLSID{2206CDB0-19C1-11D1-89E0-00C04FD7A829}

Стойност от тип DWORD, задаващ максималното време на живот на връзката в пула в секунди. По подразбиране тази стойност е равна на 600. На дадения CLSID принадлежи компонента MSDAINITIALIZE – диспечер на обслужващите компоненти на OLE DB, който се използва за синтактичен анализ на низовете за връзка OLE DB и инициализация на съответния доставчик.

ODBC

Доставчика на данни .NET за ODBC осъществява пулинг на връзките с помощта на диспечера на драйвера ODBC (ODBC Driver Manager). Пулинга на връзките се поддържа от диспечера на драйвера, започвайки от версия 3.0; версията на драйвера ODBC няма значение.

Пулинга на връзките в ODBC управлява следните два параметъра на системното регистри:

Wait Retry — време в секунди, за което се блокира пула в случая, ако сървера не отговаря. Този параметър влияе на всички приложения, използващи ODBC драйвера. Ключът в регистрито задава стойност на типа REG_SZ:

HKEY_LOCAL_MACHINESOfTWAREODBCODBCINST.INI\Wait Retry

CPTimeout – време в секунди, в продължение на което неизползваните връзки остават в пула. Този параметър влияе на всички ODBC драйвери в системата. Ключът в регистрито задава стойност на типа REG_SZ:

HKEY_LOCAL_MACHINESOFTWAREODBCODBCINST.INICPTimeout

Пулинга на връзките в ODBC може да се управлява по три начина:

прилагайки ODBC Data Source Administrator за включване и отключване на пулинга на целия драйвер и настройка на параметрите CPTimeout и Wait Retry;

чрез редактиране на съответните ключове от системното регистри;

Използвайки интерфейса на приложните програми (API) ODBC за настройка на параметрите н пулинга в ODBC-приложения.

Използване на транзакциите със свързване към пула

Проблем. Необходимо е да се използва пулинга на връзките в .NET-приложения с транзакции за максимална производителност.

Обсъждане. Връзките, участващи в транзакцията се извличат от пула и се отделят от потока въз основа на точно съответствие на контекста на транзакцията на изискващия поток и низа за връзка.

Във всеки пул на връзки се отделя раздел за връзки, нямащи съдържание на

транзакциите, и нула или повече раздели за връзки, свързани с определено съдържание на транзакциите. Във всеки от тези раздели, независимо от това дали съответства на някакъв контекст на транзакцията, пулинга на връзките се осъществява въз основа на точното съответствие със свързващия низ.

Когато потока, свързан с определено съдържание на транзакцията, даде заявка към връзка, тя автоматично го отделя за съответния пул, свързан с тази транзакция.

Когато връзката се затвори, тя се връща в съответния раздел на пула за връзки съгласно контекста на транзакцията. Това позволява да се затворят съобщения, като при това не се генерира грешка, даже ако разпределената транзакция все още не се изпълнява. Транзакцията може да бъде съхранена или възстановена по-късно.

Превключване на отворена връзка към друга база данни

Проблем. Необходимо е да се измени БД, която използва връзка, като не я създава отново.

Решение. За промяна на отворената връзка към БД може да се използва метода ChangeDatabase().

В кода на примера се създава връзка към база данни с помощта на доставчика на данни .NET за SQL Server. След това на дадената връзка се назначава друга БД. След това връзката се затваря. При изпълнението на примера на екрана се извежда стойност на свойството DataBase на обекта SqlConnection при различни състояния на връзките.

```
using System;

using System.Configuration;
```

```
using System.Text;

using System.Data;

using System.Data.SqlClient;

//...

private void ChangeDatabaseForm_Load(object sender, System.EventArgs e)

{

StringBuilder result = new StringBuilder();

// създаване на връзка към БД Northwind

SqlConnection conn = new
SqlConnection(ConfigurationSettings.AppSettings["Sql_ConnectString"]);

result.Append("Connection String:" + Environment.NewLine);

result.Append(conn.ConnectionString + Environment.NewLine + Environment.NewLine);
```

```
// отваряне на връзката

conn.Open();

result.Append("Connection.State: " + conn.State + Environment.NewLine);

result.Append("Database: " + conn.Database + Environment.NewLine);

// промяна на връзката да е към БД pubs

conn.ChangeDatabase("pubs");

result.Append("Database: " + conn.Database + Environment.NewLine);

// затваряне на връзката

conn.Close();

result.Append("Connection.State: " + conn.State + Environment.NewLine);

result.Append("Database: " + conn.Database);

resultTextBox.Text = result.ToString();
```

}

Обсъждане. Метода ChangeDatabase(), който е определен в интерфейса IDbConnection – връзка към източника на данни и е реализиран в доставчика .NET за релационни БД, включвайки SQL Server, Oracle и OLE DB. Този метод променя текущата БД за отворена връзка. Той приема единствен параметър – име на БД, която трябва да се използва вместо текущата. Името на БД трябва да бъде допустимо, иначе ще се генерира изключение ArgumentException. Ако в момента на извикване на метода връзка не е отворена, то се генерира изключение, за висещо от доставчика на данни.

Свойството DataBase на обекта Connection динамично се обновява и връща името на текущата БД за отворена връзка или БД, която ще се използва за затваряне на връзката, когато тя бъде преназначена.

Когато връзката за затваря след извикване на метода ChangeDatabase() за него се избира тази БД, която е била първоначално зададена в низа за връзка.

Свързване с текстов файл

Проблем. Изисква се с помощта на ADO.NET да се получи достъп до данни, съхраняващи се в текстов файл.

Решение. За достъпа до данни от текстов файл се използва доставчика OLE DB Jet.

В кода на примера се създава адаптер на данните OleDbDataAdapter, който с помощта на доставчика OLE DB Jet зарежда съдържанието на текстовия файл Categories.txt в локална таблица и отразява нейното съдържание в табличен елемент в екранна форма

Съдържание на файла Categories.txt

"CategoryID","CategoryName","Description"

1,"Beverages","Soft drinks, coffees, teas, beers, and ales"

2,"Condiments","Sweet and savory sauces, relishes, spreads, and seasonings"

3,"Confections","Desserts, candies, and sweet breads"

4,"Dairy Products","Cheeses"

5,"Grains/Cereals","Breads, crackers, pasta, and cereal"

6,"Meat/Poultry","Prepared meats"

7,"Produce","Dried fruit and bean curd"

8,"Seafood","Seaweed and fish"

```
using System;

using System.Configuration;
```

```
using System.Windows.Forms;

using System.Data;

using System.Data.OleDb;

//...

private void goButton_Click(object sender, System.EventArgs e)

{

// създаване на адаптер за данните за прочитане на всички редове от текстовия файл

OleDbDataAdapter da = new OleDbDataAdapter("SELECT * FROM [Categories.txt]",

ConfigurationSettings.AppSettings["TextFile_0119_ConnectString"]);

// създаване на локална таблица и попълването й с данни

DataTable dt = new DataTable("Categories");
```

```
da.Fill(dt);

// свързване на представянето на таблицата по подразбиране към табличен елемент

categoriesDataGrid.DataSource = dt.DefaultView;

}
```

Обсъждане.

Доставчика OLE DB Jet може да чете и записва редове в текстов файл. Процесора на БД Jet може да работи с други формати на файловете от БД чрез ISAM – драйвери, указани в атрибута Extended Properties на низа за връзка. Текстовия файл съответства на типа на изходната БД text, както демонстрира следния пример:

```
Provider=Microsoft.Jet.OLEDB.4.0:Data Source=c:MyTextFileDirectory; Extended Properties="text;HDR=yes;FMT=delimited";
```

Атрибута Extended Properties може да включва свойството HDR, определящо трябва ли да се включва в таблицата заглавния ред като наименования на полетата в първия ред на диапазона.

Не е възможно да се определят всички характеристики на текстовия файл чрез низа за връзка. За достъп до файла, използващ нестандартни разделители между полетата, и файлове с фиксирана ширина на текста може да се създаде файл schema.ini в същия каталог, където се намира текстовия файл. Пример – възможно съдържание на файла schema.ini за файла Categories.txt:

[Categories.txt]

Format=CSVDe1imited

ColNameHeader=True

MaxScanRows=0

Character=OEM

Coll=CategoryID Long Width 4

Col2=CategoryName Text Width 15

Col3=Description Text Width 100

Файла schema.ini представлява следната информация за данните от текстовия файл:

- име на файла;
- формат на файла;
- имена на полетата, тяхната дължина и тип на данните;
- кодировка на символите;
- специални преобразования на типовете данни.

Първия запис във файла schema.ini – това е името на текстовия файл, затворен в квадратни скоби, например [Categories.txt].

Параметърът Format задава формата на текстовия файл.

Полето в текстовия файл може да се задава по два начина:

- да се укажат имената на полетата в първия ред на текстовия файл и да се установи параметърът ColumnNameHeader да има стойност True;
- да се идентифицират стълбовете, използвайки формата Col*N* (където *N* е едноцифрен номер на стълба), и указва за всеки стълб име, ширина на стълба и типа данни.

Типове формати във файла schema.ini

Разделител – запетая -> полетата се разделят със запетаи: Format=CSDelimited

Нестандартен разделител -> полетата се разделят с помощта на определен от потребителя символ. В качеството на разделител може да се използва всеки единичен символ, с изключение на двойните кавички (“): Format=Delimited(customCharacter)

Фиксирана дължина на полето -> полетата имат фиксирана дължина: Format=FixedLength. Ако параметъра ColumnNameHeader е равен на True, то в първия ред, съдържащ наименование на стълба, в качеството на разделител трябва да се използва запетая

Разделител – табулация -> полетата се разделят със символа за табулация: Format=TabDelimited

Параметъра MaxScanRows указва колко реда трябва да се прегледат за автоматично определяне на типа на стълба. Стойността 0 означава, че трябва да се прегледат всички редове.

Записът Col*N* задава име, дължина и тип данни за всеки стълб. Тези записи са задължителни при използване на формат с фиксирана дължина и могат да се пропускат при използване на формат с разделител символ. Записът Col *N* има следния синтаксис:

ColN= [Width n]

Параметрите й са такива:

- - име на стълба. Ако в името има интервали, то трябва да се затвори в двойни кавички;
- - тип данни в стълба. Възможни стойности са: Bit, Byte, Currency, DataTime, Double, Long, Memo, Short, Single и Text.

Стойностите от типа DataTime трябва да бъдат в един от следните формати: дд-ммм-гг, мм-дд-гг, ммм-дд-гг, гггг-мм-дд или гггг-ммм-дд, където мм е номер на месеца, а ммм е трибуквено съкращение на името на месеца.

- Width n – думата Width, след която се задава цяло число n, задаващо ширина на стълба.

Параметър Character задава кодировка на символите, той може да има стойност ANSI или OEM.