

به نام خدا

عنوان مقاله:

بخش اول: ذخیره و بازیابی اطلاعات توسط ADO.net

بخش دوم: ارتباط با منابع داده در ADO.net

ارائه دهنده: مهری خیراله پور

جناب استاد: محمد سلیمی

ذخیره و بازیابی داده توسط ADO.NET (بخش اول)

با استفاده از ADO.NET ، امکان اتصال به منابع داده متفاوت ، بازیابی ، پردازش و بهنگام سازی داده ، فراهم می گردد . ADO.NET از XML به منظور انتقال داده بین برنامه ها و منابع داده استفاده می نماید . ویژوال استودیو دات نت دارای امکانات متعددی به منظور دستیابی به بانک های اطلاعاتی (برخاسته از ADO.NET) ، می باشد. پس از اتصال به یک بانک اطلاعاتی می توان با استفاده از مجموعه ای از اشیاء ، خصلت ها و متدها ، صرفنظر از نوع بانک اطلاعاتی ، عملیات مورد نظر خود در ارتباط با یک بانک اطلاعاتی را انجام داد. در این مقاله به بررسی امکانات ویژوال استودیو دات نت به منظور اتصال به یک بانک اطلاعاتی و انجام عملیات متفاوتی همچون خواندن ، تغییر و حذف رکوردهای موجود در یک بانک اطلاعاتی ، خواهیم پرداخت .

مبانی اولیه ADO.NET

به منظور دستیابی به داده در ADO.NET ، سه لایه وجود دارد :

- محل فیزیکی ذخیره سازی داده : لایه فوق می تواند یک بانک اطلاعاتی OLE ، یک بانک اطلاعاتی SQL و یا یک فایل XML باشد .
- ارائه دهنده داده : لایه فوق ، شامل شی Connection و اشیاء Command بوده و تصویری از داده ها را در حافظه ایجاد می نماید.
- Data set : لایه فوق ، رکوردهای بازیابی شده از یک منبع داده را در خود نگهداری می نماید. یک Data Set شامل رکوردهائی از یک و یا چندین جدول می باشد .



لایه ارائه دهنده داده ، واسطه بین محل ذخیره سازی داده و Data Set ، می باشد . Data Set مستقل از منبع داده است .

در ADO.NET از دو نوع Connection به منظور اتصال به یک بانک اطلاعاتی (با توجه به نوع بانک اطلاعاتی) استفاده می گردد :

- شی OleDbConnection که از آن به منظور ارتباط به یک بانک اطلاعاتی محلی استفاده می گردد. این نوع ارتباطات از شی OleDbDataAdapter به منظور اجرای دستورات و برگرداندن داده استفاده می نمایند .

- شی `SqlConnection` که از آن به منظور ارتباط به یک بانک اطلاعاتی سرویس دهنده ، استفاده می گردد. این نوع ارتباطات از شی `SqlDataAdapter` به منظور اجرای دستورات و برگرداندن داده استفاده می نمایند .

علاوه بر موارد فوق ، امکان دستیابی مستقیم به فایل های XML با استفاده از متدهای `ReadXML` و `WriteXML` مربوط به شی `DataSet` ، نیز وجود دارد .
 اشیاء ، خصلت ها و متدهای حمایت شده توسط ADO.NET توسط سه namespace ارائه می گردد :

- `System.Data` : در این namespace کلاس ها ، نوع ها و سرویس های لازم به منظور ایجاد و دستیابی به `data set` و اشیاء زیر مجموعه آن وجود دارد .
- `System.Data.SqlClient` : در این namespace کلاس ها و نوع های لازم به منظور دستیابی به بانک های اطلاعاتی SQL Server ، وجود دارد .
- `System.Data.OleDb` : در این namespace کلاس ها و نوع های لازم به منظور دستیابی به بانک های اطلاعاتی OLE ، وجود دارد .

در زمان دستیابی و استفاده از بانک های اطلاعاتی (از طریق کد نوشته شده) ، می بایست از عبارت `Imports` در ابتدای ماژول استفاده گردد .

VB.NET

`Imports System.Data`

`Imports System.Data.SqlClient 'For SQL database Connection`

`Imports System.Data.OleDb 'For OLE DB database Connection`

به منظور دستیابی داده از طریق ADO.NET ، مراحل زیر را دنبال می نمایم :

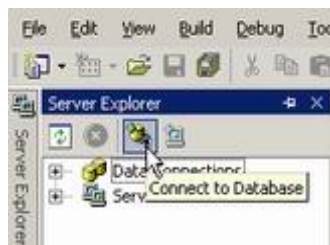
- ایجاد ارتباط با بانک اطلاعاتی توسط یک شی `Connection`
- فراخوانی یک `command` به منظور ایجاد یک `DataSet` با استفاده از یک شی `adapter`
- استفاده از شی `DataSet` (در کد نوشته شده) ، به منظور نمایش داده و یا تغییر آیتم های موجود در بانک اطلاعاتی
- فراخوانی یک `Command` به منظور بهنگام سازی بانک اطلاعاتی از طریق `DataSet` با استفاده از یک شی `adapter`
- غیر فعال نمودن (`Close`) ارتباط ایجاد شده با بانک اطلاعاتی در صورتیکه ارتباط با بانک اطلاعاتی توسط متد `Open` فعال شده باشد .

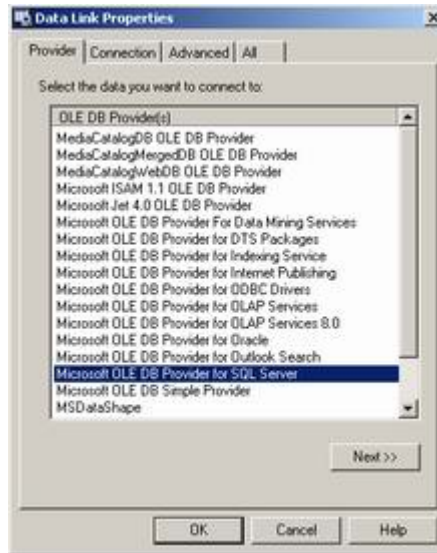
در ادامه به تشریح هر یک از مراحل فوق خواهیم پرداخت .

ارتباط با بانک اطلاعاتی

با استفاده از `Server Explorer` در ویژوال استودیو دات نت ، امکان ایجاد یک ارتباط با بانک اطلاعاتی فراهم می گردد. در این رابطه مراحل زیر را دنبال می نمایم (در حالت `Design`) :

- فعال نمودن `Server Explorer` (از طریق `View|Server Explorer`)
- در `Server Explorer` ، گزینه `Connect To Database` را انتخاب می نمایم .در ادامه جعبه محاوره ای `Properties DataLink` نمایش داده می شود.





- از طریق Provider Tab ، نوع بانک اطلاعاتی را مشخص می نمایم . به صورت پیش فرض ، OLE DB Provider for SQL Server انتخاب شده است . در صورتیکه بانک اطلاعاتی تحت SQL Server اجرا می گردد ، انتخاب پیش فرض درست می باشد . به منظور دستیابی به یک نوع دیگر بانک اطلاعاتی ، می بایست Provider مربوطه را انتخاب نمود. مثلاً" برای دستیابی به یک بانک اطلاعاتی اکسس ، Microsoft Jet 4.0 OLE DB انتخاب می گردد .
- از طریق Connection Tab ، بانک اطلاعاتی مورد نظری که قصد برقراری ارتباط با آن وجود دارد را انتخاب می نمایم
- پس از اعمال تنظیمات لازم با کلیک نمودن بر روی دکمه Test Connection ، می توان از صحت تنظیمات انجام شده ، اطمینان حاصل نمود . پس از تست موفقیت آمیز ارتباط ایجاد شده با بانک اطلاعاتی ، ویژوال استودیو دات نت ، ارتباط ایجاد شده را به Server Explorer اضافه می نماید .
- با کلیک نمودن بر روی علامت "+" ، آیتم های مربوط به بانک اطلاعاتی نمایش داده خواهند شد . برای مشاهده جداول ، می توان آیتم های مربوطه را تحت Data Connection فعال و در ادامه از طریق Table و فعال نمودن آن ، جداول مربوط به بانک اطلاعاتی را مشاهده نمود.



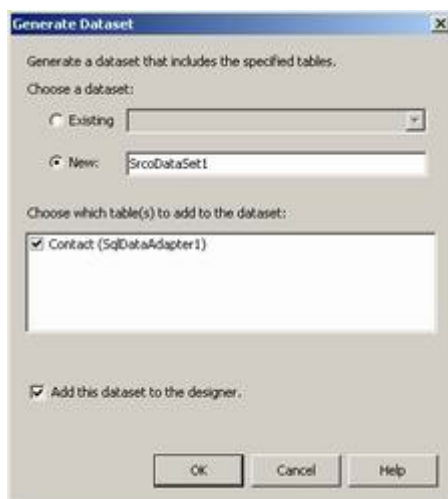
- برای افزودن یک آیتم به برنامه ، از طریق Sever Explorer آن را انتخاب و بر روی فرم وب قرار می دهیم . در مواردی که یک جدول بر روی فرم وب مستقر می گردد ، ویژوال استودیو دات نت ، اشیاء Connection و adapter را به همراه تنظیمات مناسب ، ایجاد می نماید.



ایجاد یک Data Set

با استفاده از اشیاء Connection و Adapter (ایجاد شده در بخش قبل)، می توان یک Data set را ایجاد نمود. برای ایجاد یک Data set پس از استقرار در حالت Design ، مراحل زیر را دنبال می نمایم .

- بر روی شی adapter کلیک سمت راست نموده و گزینه Generate Dataset را انتخاب می نمایم . ویژوال استودیو در ادامه ، جعبه محاوره ای Generate Dataset را نمایش خواهد داد .



- جداول مورد نظر را برای اضافه نمودن به Data set انتخاب کرده و در ادامه بر روی دکمه Ok کلیک می نمائیم .
- ویژوال استودیو یک Data set جدید را ایجاد و آن را به فرم وب اضافه می نماید .
- برای مشاهده داده موجود در Data set ، (در حالت Design) بر روی شی DataSet کلیک سمت راست نموده و گزینه View Schema را انتخاب می نمائیم . در ادامه Set Data در پنجره XML Designer نمایش داده می شود .

E Contact (Contact)	
E Fname	string
E Lname	string
E Email	string

نمایش یک Data set

برای نمایش یک Data Set بر روی یک فرم وب و در زمان اجراء ، مراحل زیر را دنبال می نمائیم :

- افزودن یک کنترل بر روی فرم وب به منظور نمایش داده . مثلاً می توان یک کنترل DataGrid را به فرم وب اضافه نمود .
- انتخاب Data set به عنوان منبع داده برای کنترل . مثلاً برای کنترل DataGrid ، گزینه Property Builder را از طریق پنجره Properties انتخاب و DataSource آن را به شی DataSet نسبت داده و خصلت DataMember را به یک جدول در Data Set نسبت می دهیم .



- ستون های مورد نظر برای نمایش در کنترل را مشخص می نمائیم . برای کنترل DataGrid ، بر روی آیتم Columns کلیک نموده (از طریق جعبه محاوره ای Properties) و گزینه Automatically At Run Create Columns را Available Time را غیر فعال نموده (Deselect) و در ادامه ستون های مورد نظر برای نمایش را از طریق Columns list ، اضافه می نمائیم . (در این مثال ، ستون های نام و آدرس پست الکترونیکی اضافه شده اند) .

نام	آدرس پست الکترونیکی
abc	abc
abc	abc
abc	abc
abc	abc
abc	abc

- اضافه نمودن کد لازم در روتین مربوط به رویداد Page_Load به منظور پر نمودن Data set از طریق Data Adapter و نسبت دهی داده از طریق شی DataSet برای کنترل . مثلاً " کد زیر باعث نمایش داده در کنترل DataGrid (ایجاد شده در مرحله قبل) می نماید .

VB.NET

```
Private Sub Page_Load (ByVal sender As System.Object , ByVal e As System.EventArgs ) Handles MyBase.Load
    SqlDataAdapter1.Fill ( SrcDataSet1)
    DataGrid1.DataBind ( )
End Sub
```

- پس از اجرای برنامه و لود شدن صفحه ، Dataset نمایش داده می شود .



- با توجه به این که پر نمودن Data set و نسبت دهی داده به کنترل DataGrid از طریق رویداد Page_Load انجام می شود ، ضرورتی به نگهداری اطلاعات وضعیت برای کنترل DataGrid وجود نخواهد داشت . با مقداری خصلت EnableViewState به False ، کارائی برنامه بهبود می یابد .

تغییر رکوردها در یک بانک اطلاعاتی

DataSet یک شی اصلی و مهم در ADO.NET بوده و هرگونه تغییرات شامل افزودن ، حذف و یا تغییر رکوردهای موجود در یک بانک اطلاعاتی عموماً از طریق شی فوق ، انجام می شود . برای تغییر رکوردها از طریق یک DataSet مراحل زیر را دنبال می نمائیم .

- دستیابی به شی DataSet (مشابه روشی که به آن اشاره گردید)
- تغییر DataSet
- بهنگام سازی بانک اطلاعاتی از طریق DataSet توسط فراخوانی متد Update مربوط به شی Adapter

از مجموعه های Tables, Rows و Columns برای دستیابی به داده موجود در یک DataSet استفاده می گردد . در بخش دوم این مقاله با نحوه استفاده از مجموعه های فوق به منظور ویرایش یک بانک اطلاعاتی ، آشنا خواهیم شد .

ارتباط به منابع داده در ADO.NET (بخش دوم)

در بخش اول این مقاله به تشریح انواع Data Provider پرداخته و با نحوه انتخاب آنان بمنظور استفاده در برنامه ها ، آشنا شدیم . در این بخش به بررسی نحوه ایجاد یک Connection خواهیم پرداخت . Connection String ، بخشی ضروری بمنظور ارتباط با یک منبع داده است . خصلت ConnectionString مربوط به شی Connection ، اطلاعات لازم را در اختیار شی Connection قرار می دهد .

امنیت بانک اطلاعاتی

در زمان دستیابی به داده از طریق ADO.NET ، می بایست با استفاده از یک روش ایمن به بانک اطلاعاتی مرتبط گردید . بمنظور نیل به هدف فوق ، اطلاعات امنیتی نظیر نام کاربر و رمز عبور می بایست قبل از ایجاد هر گونه ارتباط با بانک اطلاعاتی در اختیار وی گذاشته شوند (ارسال اطلاعات اولیه بمنظور اطمینان از معتبر بودن کاربر) . امنیت بانک اطلاعاتی ، بستگی به نوع بانک اطلاعاتی مربوطه خواهد داشت .

استفاده از امنیت سرویس دهنده SQL

سرویس دهنده SQL ، با استفاده از دو روش متفاوت اقدام به تأیید کاربر می نماید : تأیید کاربران بر اساس سیستم امنیتی ویندوز (Authentication Microsoft Windows) و حالت ترکیبی (Authentication Mixed Mode) .

روش Windows Authentication

در مدل فوق ، این امکان برای یک کاربر فراهم خواهد شد که با استفاده از یک Account ویندوز به بانک اطلاعاتی مرتبط گردد . خصلت امنیتی شبکه برای کاربر در زمان ورود به شبکه ایجاد می گردد . زمانیکه یک کاربر شبکه سعی در ارتباط با بانک اطلاعاتی سرویس دهنده SQL می نماید ، عملیات تأیید وی آغاز و در ادامه بر اساس سیاست های امنیتی ورود به شبکه که قبلاً برای وی تعریف و تنظیم شده است ، امکان دستیابی وی به بانک اطلاعاتی محقق و یا با درخواست وی بدلائل امنیتی مخالفت خواهد شد . در چنین حالتی ضرورتی به یک Login مجدد (ورود نام و رمز عبور توسط کاربر) وجود نخواهد داشت . استفاده از روش فوق ، بمنظور ارتباط با بانک اطلاعاتی دارای مزایای زیر است :

- متعیرسازی ایمن به همراه رمزنگاری رمزهای عبور
- Auditing
- تاریخ سررسید رمز عبور
- تعریف حداقل طول رمز عبور
- غیر فعال شدن (قفل نمودن) رمز عبور پس از انجام چندین تلاش نافرمان جهت ورود به شبکه

نکته : با توجه به اینکه کاربران و گروههای ویندوز ، صرفاً توسط ویندوز نگهداری و پشتیبانی می گردند ، سرویس دهنده SQL اطلاعات مربوط به یک گروه کاربران را در زمان اتصال به بانک اطلاعاتی می خواند . در صورتیکه تغییری در رابطه با مجوزهای دستیابی برای یک کاربر متصل شده ایجاد گردد ، تغییرات ایجاد شده در زمان آتی که کاربر به سرویس دهنده SQL مرتبط می گردد ؛ در نظر گرفته شده و متناسب با تغییرات ایجاد شده با کاربر برخورد خواهد شد .

روش Mixed Mode Authentication

در روش فوق ، این امکان برای کاربران فراهم می گردد که به یک سرویس دهنده SQL بر اساس Windows Authentication و با SQL Server Authentication متصل گردند . کاربرانی که از طریق ویندوز NT 4.0 و یا ویندوز 2000 به بانک اطلاعاتی مرتبط می گردند ، قادر به استفاده از ارتباطات trusted در Windows Authentication و یا Mixed Mode Authentication می باشند . زمانیکه کاربری با استفاده از یک نام و رمز عبور خاص و بکمک یک ارتباط non-trusted به بانک اطلاعاتی متصل می گردد ، سرویس دهنده SQL خود عملیات تأیید کاربر را بررسی می نماید (آیا یک account سرویس دهنده SQL پیکربندی و تنظیم شده است ؟) . در صورتیکه ماحصل بررسی فوق منفی باشد ، کاربر تأیید نشده و وی با یک پیام خطا مواجه خواهد شد .

نکته : در صورتیکه کاربری سعی در ارتباط با یک سرویس دهنده SQL 7.0 ، (مورد نظر سرویس دهنده SQL 2000 نمی باشد) را داشته و یک نام خالی (blank) را وارد نماید ، سرویس دهنده SQL 7.0 ، از Windows Authentication استفاده می نماید . علاوه بر این ، در صورتیکه کاربری سعی در ارتباط با یک سرویس دهنده SQL 7.0 نماید که برای Windows Authentication پیکربندی شده باشد (استفاده از یک Login خاص) ، login مورد نظر نادیده گرفته شده و از روش Windows Authentication استفاده خواهد شد .

روش SQL Server Authentication بمنظور سازگاری با نسخه های قبلی ارائه شده است . زیرا بر نامه های نوشته شده برای سرویس دهنده SQL 7.0 ، و یا قبل از آن می بایست از نام و رمز عبور سرویس دهنده SQL استفاده نمایند . علاوه بر این ، زمانیکه یک نمونه از سرویس دهنده SQL تحت ویندوز 98 اجرا می گردد ، چون روش Windows Authentication در نسخه ویندوز 98 حمایت نمی گردد ، مجدداً از روش SQL Server Authentication استفاده خواهد شد . بنابراین سرویس دهنده SQL از Mixed Mode در زمان اجراء ویندوز 98 استفاده می نماید (ولی صرفاً SQL Server Authentication حمایت می گردد) .

Connection String چیست ؟

بمنظور انتقال اطلاعات بین یک منبع داده و برنامه ، می بایست در ابتدا دارای یک ارتباط (اتصال) با منبع داده باشیم . خصلت ConnectionString ، اطلاعات ضروری و مورد نیاز در ارتباط با تعریف یک اتصال به منبع داده را از طریق یک رشته مشخص بر مجموعه ای از پارامترها ، ارائه می نماید . جدول زیر چندین پارامتر متداول در رابطه با Connection string را نشان می دهد .

پارامتر	توضیحات
Provider	از خصالت فوق بمنظور تنظیم و یا برگرداندن نام Provider مرتبط با
Connection TimeOut or Connect TimeOut	مدت زمان انتظار برحسب ثانیه بمنظور ارتباط با بانک اطلاعاتی را
Initial Catalog	نام بانک اطلاعاتی
Data Source	نام سرویس دهنده SQL استفاده شده در زمانیکه یک Connection فعال
Password	رمز عبور برای SQL Server account
User ID	account SQL Server Login
Integrated Security or Trusted Connection	ایمن بودن ارتباط را مشخص می نماید. مقادیر مورد نظر می تواند
Perisist Security Info	زمانیکه مقدار آن False باشد ، اطلاعات حساس امنیتی نظیر رمز عبور

نحوه تنظیم یک Connection String

بمنظور ایجاد و مدیریت یک Connection می بایست یکی از اشیاء Connection مربوط به ADO.NET را استفاده نمود : شیء SqlConnection و یا شیء OleDbConnection . تنظیم خصالت ConnectionString صرفاً زمانی که Connection بسته می گردد، میسر خواهد بود . بمنظور Reset نمودن یک Connection String می بایست Connection مورد نظر را بست و مجدداً آن را فعال نمود.

مثال

در این بخش به بررسی چندین مثال بمنظور آشنائی با نحوه استفاده از Connection String به همراه پارامترهای مربوطه ، خواهیم پرداخت . دقت داشته باشید که تمامی Connection String ها ، دارای پارامترهای یکسان نمی باشند .
مثال 1 - نحوه اتصال به یک بانک اطلاعاتی SQL Server 2000 با استفاده از شیء SqlConnection و VB.NET

پارامتر	مقدار
Product	SQL Server 2000
Server name	Tehran
DatabaseName	Northwind
Security	Mixed mode
Username	sa
Password	110
Timeout	1 minute
نحوه اتصال به بانک اطلاعاتی	
<pre>Dim cnNorthwind as New _ system.Data.SqlClient.SqlConnection() cnNorthwind.ConnectionString = _ "User ID = sa;" & _ >Password=" 110;" & _ "Initial Catalog=Northwind;"& _ >Data Source =Tehran;" & _ "Connection TimeOut= 60;"</pre>	

مثال 2 - نحوه اتصال به یک بانک اطلاعاتی Access با استفاده از یک OleDbConnection و ویژوال بیسیک

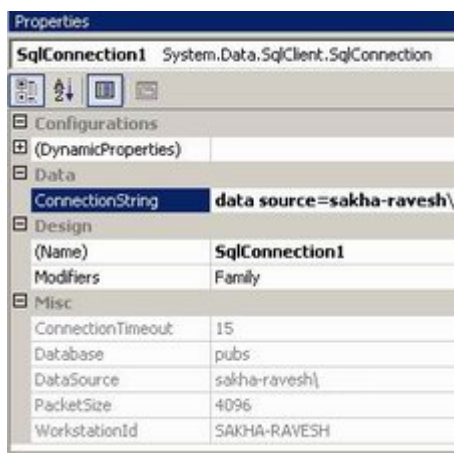
مقدار	پارامتر
Microsoft Access 2000	Product
\\Samples\Northwind.mdb	Database location
نحوه اتصال به بانک اطلاعاتی	
<pre>Dim cnNorthwind as New _ system.Data.OleDb.OleDbConnection() cnNorthwind.ConnectionString = _ "Provider = Microsoft.Jet.OLED.4.0" & _ "Data Source= \Samples\Northwind.mdb;"</pre>	

مثال 3 - نحوه اتصال به یک بانک اطلاعاتی SQL Server 6.5 با استفاده از شی OleDbConnection و C#

مقدار	پارامتر
SQL Server 6.5	Product
Myserver	Server name
Pubs	DatabaseName
Windows authentication	Security
نحوه اتصال به بانک اطلاعاتی	
<pre>System.Data.OleDb.OleDbConnection cnNorthwind = new System.Data.OleDb.OleDbConnection (); cnNorthwind.ConnectionString = "Provider = SQLOLEDB;" + "Data Source =Myserver;" + "Initial Catalog= Pubs;" + "Integrated Security=SSPI;" ;</pre>	

ساده ترین روش تنظیم یک Connection string استفاده از ویژوال استودیو دات نت است :

- برنامه ویژوال استودیو را فعال نمایید.
- یک Application ASP.NET و یا Windows Application را با استفاده از VB ، ایجاد نمایید.
- کنترل SqlConnection را بر روی فرم مورد نظر قرار دهید (Drop&Darg) .
- در پنجره Properties ، خصلت SqlConnection را تنظیم نمایید.



- پنجره ویرایشگر کد را فعال و کدی را که بصورت اتوماتیک ایجاد شده است ، مشاهده نمایید.

ارتباط به منابع داده در ADO.NET (بخش سوم)

در بخش اول این مقاله به تشریح انواع Data Provider و در بخش دوم با نحوه ایجاد یک Connection آشنا شدیم. در این بخش به بررسی نحوه مدیریت یک Connection خواهیم پرداخت.

پس از تعریف خصلت ConnectionString مربوط به شی Connection، می توان با استفاده از متدهای Open و Close، مدیریت وضعیت Connection جاری را انجام داد. در ادامه با نحوه استفاده از متدهای فوق و پاسخگویی به رویدادهای Connection آشنا خواهیم شد.

فعال نمودن و غیر فعال نمودن یک Connection

بمنظور فعال نمودن و یا غیر فعال نمودن یک Connection می توان بصورت تلویحی عملیات فوق را با فراخوانی متدهای یک شی که از Connection استفاده می نمایند انجام و یا با صراحت و بکمک متدهای Open و Close اقدام به فعال نمودن و یا غیر فعال نمودن Connection نمود. متدهای Open و Close دو متد اولیه برای هر Connection می باشند.

- متد Open با استفاده از اطلاعات موجود در خصلت ConnectionString، با یک منبع داده ارتباط و یک ارتباط فعال شده را ایجاد خواهد کرد.
- متد Close باعث غیر فعال شدن Connection می گردد.

لازم است به این نکته اشاره گردد که بستن (غیر فعال نمودن) یک Connection امری لازم و ضروری است، چراکه اکثر منابع داده صرفاً تعداد محدودی از ارتباطات فعال را حمایت نموده و در ضمن ارتباطات فعال، منابع با ارزش سیستم را اشغال و حجم عملیاتی بر روی سیستم را افزایش خواهند داد.

فعال نمودن و غیر فعال نمودن صریح یک Connection

استفاده صریح از متدهای Open و Close بمنظور فعال نمودن و یا غیر فعال نمودن Connection توصیه و دارای مزایای زیر است:

- کد نوشته شده دارای خوانایی به مراتب بیشتری خواهد بود.
- اشکال زدائی برنامه ها با سرعت مطلوبی انجام خواهد شد.
- موثرتر می باشد.

پس از استفاده Connection، لازم است که ارتباط ایجاد شده غیر فعال گردد. بدین منظور، می توان از متدهای Close و یا Dispose مربوط به شی Connection استفاده نمود. متد Close، تراکنش های بلا تکلیف را تعیین تکلیف نموده و در ادامه ارتباط را غیر فعال می نماید. در مواردیکه Pooling فعال شده باشد، متد فوق باعث آزادسازی ارتباط ایجاد شده از Connection pool می گردد. فراخوانی متد Close توسط یک برنامه می تواند چندین مرتبه صورت پذیرد.

فعال نمودن و غیر فعال نمودن تلویحی یک Connection

در صورت استفاده از DataAdapters، ضرورتی به فعال نمودن و یا غیر فعال نمودن یک Connection وجود نخواهد داشت. زمانیکه یکی از متدهای مربوط به اشیاء فوق، فراخوانده می گردد (مثلاً متد Update و یا Fill مربوط به شی SqlDataAdapter)، متد مورد نظر بررسی لازم در خصوص فعال بودن ارتباط را انجام و در صورتیکه ارتباط فعال نشده باشد، SqlDataAdapter ارتباط را فعال و پس از انجام عملیات مورد نظر، ارتباط را غیر فعال خواهد نمود.

مثال: در صورتیکه یک DataSet داده خود را از چندین جدول متفاوت که در بانک اطلاعاتی مشابه قرار دارند، دریافت می نماید، صرفاً یک Connection وجود داشته و دارای چندین DataAdapters خواهد بود (هر DataAdapters مرتبط با یک جدول). در صورتیکه Connection بصورت تلویحی (عدم استفاده صریح از متدهای Open و Close) فعال و یا غیر فعال گردد، در زمان پر نمودن اطلاعات در Dataset، ارتباط با منبع داده چندین مرتبه فعال و یا غیر فعال خواهد شد. در چنین مواردی مناسب است که در ابتدا Connection با صراحت فعال و پس از فراخوانی متدهای Filling مربوط به چندین DataAdapters، ارتباط ایجاد شده با منبع داده را با بکارگیری متد Close و باصراحت غیر فعال نمود.

استفاده از متد Dispose

زمانیکه یک Connection غیر فعال می گردد، جریان اطلاعات به و یا از منبع داده غیر فعال می گردد. در چنین حالتی، منابع غیر مدیریت یافته استفاده شده توسط شی Connection، آزاد نخواهند گردید. در صورتیکه Connection pooling فعال شده باشد، عملیات آزاد سازی Connection از Pool نیز انجام خواهد شد. اشیاء SqlConnection و OleDbConnection دارای یک متد Dispose بوده که از آن بمنظور آزادسازی منابع غیرمدیریت یافته استفاده می گردد. با فراخوانی متد Dispose، عملیات حذف Connection pool از Connection نیز انجام خواهد شد.

مثال : استفاده از متد Dispose

در مثال زیر، یک شیء SqlConnection ایجاد و در ادامه با استفاده از متد Open فعال و سپس بمنظور غیرفعال نمودن Connection و آزاد سازی منابع استفاده شده توسط Connection از متد Dispose استفاده و در نهایت شیء مورد نظر به Nothing مقداردهی شده است .

```
نحوه استفاده از متد Dispose
تعریف و نمونه سازی یک شیء جدید SqlConnection
Dim cnNorthwind AS New System.Data.SqlClient.SqlConnection ()
مقدار دهی مناسب خصیلت ConnectionString
cnNorthwind.ConnectionString = "Data Source=(Local); " & _
    "Initial Catalog = Northwind;" & _
    "Integrated Security=SSPI;"
فعال نمودن Connection
cnNorthwind.open()
انجام عملیات دلخواه و مرتبط با بانک اطلاعاتی
cnNorthwind.Close()
Dispose نمودن ارتباط که باعث حذف connection از pool connection بر روی
cnNorthwind.Dispose()
آزاد سازی حافظه اشغال شده توسط شیء SqlConnection
cnNorthwind = Nothing
```

برخورد با رویدادهای Connection

رویداد StateChange زمانی که وضعیت Connection تغییر نماید ، محقق می گردد (وضعیت Connection از فعال به غیر فعال و یا از غیر فعال به فعال تبدیل گردد) . بمنظور برخورد مناسب با هر یک از رویدادها ، از یک Event handler استفاده می گردد . هر رویداد دارای Event handler مختص خود می باشد . Event handler مربوط به رویداد StateChange ، متدی است که دارای آرگومانی از نوع StateChangeEventArgs می باشد . آرگومان فوق ، شامل داده مرتبط با رویداد است .

رویداد StateChange

نوع NET data provider ، ، که از آن استفاده می گردد ، آرگومان های خاص مربوط به رویداد StateChange را مشخص می نماید :

- رویداد SqlConnection.StateChange برای یک شیء SqlConnection
- رویداد OleDbConnection.StateChange برای یک شیء OleDbConnection

لازم است بدین نکته اشاره گردد که تمامی رویدادها در دات نت دارای دو پارامتر زیر می باشند :

- ارسال کننده (از نوع شیء)
- e از نوع XxxEventArgs

برای رویداد StateChange، پارامتر e از نوع StateChangeEventArgs می باشد. handler event مربوط به دو رویداد فوق، نوع مشابهی از آرگومان ها را دریافت می نماید: StateChangeEventArgs که شامل داده مرتبط با این رویداد است. جدول زیر خصالت های مربوط به کلاس StateChangeEventArgs را نشان می دهد:

شرح	خصالت
خصالت فوق، وضعیت جدید Connection را مشخص	CurrentSatate
وضعیت اولیه Connection را برمی گرداند.	OriginalSate

نمونه ای از یک Event Handler مربوط به StateChange

مثال زیر نحوه ایجاد StateChangeEventHandler را با VB.NET، نشان می دهد:

نحوه استفاده از یک Event Handler با استفاده از VB.NET
<pre>Private Sub cnNorthwind_State (ByVal Sender As Object , ByVal e As System.Data.StateChangeEventArgs) Handles cnNorthwind.StateChange ' نمایش وضعیت اولیه و جاری State در یک messagebox MessageBox.Show("CurrentState: "& e.Currentstate.ToString() & vbCrLf & _ "OriginalState : "& e.OriginalState.ToString() , _ "cn.Northwind.StateChange" , _ MessageBoxButtons.OK, _ MessageBoxIcon.Information) End Sub</pre>