

برنامه نویسی بانکهای اطلاعاتی

به روش



قسمت اول:

بطور کلی ADO روشی است برای اتصال و تغییر بانکهای اطلاعاتی و مخفف کلمات ActiveX Data Objects است. در ویژوال بیسیک شش با استفاده از کنترل Data می توان به روش ADO برنامه نویسی بانکهای اطلاعاتی را انجام داد. برنامه نویسی به روش ADO به دو صورت ADODB و ADODC انجام می شود. که البته تفاوت چندانی ندارند. در روش اول با استفاده از یک کنترل همل برنامه نویس انجام می شود ولی در روش دوم با استفاده توابع و اشیا تعریفی این کار انجام می شود. روش اول در ویژوال بیسیک معمول تر است ولی روش دوم علاوه بر استفاده در خود ویژوال بیسیک در ASP نیز کاربرد دارد. بهر حال تفاوت چندانی بین این دو روش وجود ندارد.

فعلا به طور خیلی مقدماتی برنامه نویسی به روش ADODC را شروع می کنیم.

ابتدا باید بانک اطلاعاتی خود را درست کنید برای این منظور در ویژوال بیسیک از منوی Add-Ins به برنامه Visual Data Manager بروید و از منوی New یک بانک اطلاعاتی جدید ایجاد کنید. بروی پنل سمت چپ راست کلیک کنید و گزینه New Table را انتخاب کنید. در پنجره جدید نام جدول خود را بنویسید و در سمت چپ با استفاده از کلیدهای Add Field و Remove Field ستون‌های مورد نظر خود را ایجاد کنید. توجه داشته باشید که شما می‌توانید نوع فیلدهای خود را نیز مشخص کنید. پس از پایان کار کلید Build Table را کلیک کنید و از برنامه Visual Data Manager خارج شوید.

یک پروژه جدید در ویژوال بیسیک ایجاد کنید و یک کنترل Data روی فرم قرار دهید و خصوصیت Visible آن را False کنید. خصوصیت Connect کنترل را به توجه به نوع بانکی که ایجاد کرده‌اید تنظیم کنید بطور مثال اگر بانک شما از نوع Access می‌باشد □ خصوصیت Connect نیز باید از نوع Access باشد. سپس خصوصیت DatabaseName را با استفاده از کلید کنار آن به دایرکتوری بانک خود تغییر دهید. خصوصیت RecordSource را انتخاب و از لیست مربوطه نام جدول بانک اطلاعاتی را انتخاب کنید.

حال به رکوردهای جدول سر و کار داریم برای این کار باید با استفاده از خصوصیت زیر به توابع و خصوصیات رکوردها دسترسی پیدا کنیم:

Data1.RecordSet

دستورا زیر برای حرکت در بین رکوردها استفاده می‌شود. دقت داشته باشید در روش ADO در یک لحظه از بین تمامی رکوردها □ RecordSet تنها می‌تواند روی یکی قرار داشته باشد. یک رکورد در واقع مانند یک ردیف در جدول می‌ماند.

برای حرکت به یک رکورد دلخواه بر اساس شماره ستون:

Data1.RecordSet.Move (Rows As long, [StartBookmark])

برای حرکت به اولین رکورد:

Data1.RecordSet.MoveFirst

برای حرکت به آخرین رکورد:

Data1.RecordSet.MoveLast

برای حرکت به رکورد بعدی:

Data1.RecordSet.MoveNext

برای حرکت به رکورد قبلی:

Data1.RecordSet.MovePrevious

خصوصیات و توابع زیر هم کاربردهای فراوانی دارند.

برای تایید تغییرات انجام شده در جدول:

Data1.RecordSet.Update

تعداد رکوردهای جدول را در خود نگه می‌دارد:

Data1.RecordSet.RecordCount

برای تصحیح جدول:

Data1.RecordSet.Edit

یک خصوصیت از نوع Boolean که قابلیت تصحیح شدن یا نشدن جدول را مشخص می‌کند:

Data1.RecordSet.LockEdit

یک خصوصیت از نوع Boolean که مشخص می‌کند آیا RecordSet بر روی رکورد آخر قرار دارد یا نه:

Data1.RecordSet.EOF

یک خصوصیت از نوع Boolean که مشخص می‌کند آیا RecordSet بر روی رکورد اول قرار دارد یا نه:

Data1.RecordSet.BOF

شماره رکوردی را که RecordSet بر روی آن قرار دارد را بر می‌گرداند:

Data1.RecordSet.GetRows

برای جستجو در جدول بکار می‌روند:

Data1.RecordSet.FindFirst

Data1.RecordSet.FindLast

Data1.RecordSet.FindNext

Data1.RecordSet.Previous

اضافه نمودن یک رکورد جدید:

Data1.Recordset.AddNew

```
Data1.Recordset("FieldName") = "Value"  
Data1.Recordset("FieldName") = "Value"  
Data1.Recordset("FieldName") = "Value"  
.....  
...  
.
```

Data1.Recordset.Update

حالت کلی ایل دستور بدین صورت است که ابتدا یک رکورد جدید اضافه می‌کنیم و سپس فیلدهای موجود را با مقادیر دلخواه پر می‌کنیم و در آخر نیز جدول را Update می‌کنیم.

حذف یک رکورد:

دستور زیر رکوردی را که RecordSet بر روی آن قرار دارد را از بین می‌برد.

Data1.Recordset.Delete

در پایان لازم به ذکر است که روش ADODB روش حرفه‌های تر پیشرفته تری است و براحتی می‌توان با استفاده از آن در هنگام اجرای برنامه به بانک‌های جدیدی متصل شد و کارهای پیشرفته تری انجام داد.

قسمت دوم:

خوب تو این مقاله سعی داریم تا با مقدمات برنامه نویسی به روش ADO و با سبک ADODB آشنا شویم. ADODB حرفه‌ای ترین روش برنامه نویسی تکنولوژی ADO است که در تمامی محیط‌های برنامه نویسی به یک صورت کاربرد دارد. در زیر با استفاده از گرامر زبان VB اقدام به پیاده سازی این روش می‌کنیم. اگر شما این نحوه برنامه نویسی را بخوبی بفهمید قادر

خواهید بود تا براحتهی تحت نرم افزارهایی مثل اکسس یا اکسل یا تحت تکنولوژی های ASP و ASP.net از این روش ADODB استفاده نمایید.

از مزایا استفاده از این روش امکان استفاده از بانک هایی است که با تکنولوژی Jet 4.0 نوشته شده اند. اگر بخاطر داشته باشید در مقاله بانک های خود را در نرم افزار Visual Data Manager تهیه کردیم. این بانک ها از Jet 2.0 پیروی می کنند. و اگر اقدام به باز کردن این گونه بانک ها با اکسس ۹۷ کرده باشید. متوجه خواهید شد که این بانک ها باید تبدیل شوند. اما در روش ADODB می توانید با استفاده Jet 4.0 با بانک های اطلاعاتی ارتباط برقرار کنید. برای بوجود آوردن یک بانک جدید از این نوع براحتهی می توانید تحت نرم افزار Access 2000 یا Access XP اقدام به ایجاد این گونه بانک ها کنید. دقت کنید که تکنولوژی Jet یک استاندارد برای کار با بانک های اطلاعاتی است که توسط مایکروسافت ایجاد شده است. در پایان این مقدمه نیز به شما توصیه می کنیم که هرگاه خواستید در برنامه با یک بانک اطلاعاتی ارتباط برقرار کنید (حتی در VB6) از روش ADODB استفاده نمایید.

ADO شامل سه Object اصلی زیر است:

۱. Connection

۲. Recordset

۳. Command

شی اول برای اتصال به بانک اطلاعاتی و جداول آن بکار می رود. شی دوم دقیقاً عمل رکوردست Data Object مقاله قبلی را بر عهده دارد و برای کار با رکوردهای یک جدول بکار می رود. Command هم یک شی پیشرفته است که زیاد با آن سر و کار نداریم و بیشتر برای کار با پارامترهای مربوط به جداول بکار می رود. برای فعال کردن این سه Object در ویژوال بیسیک ۶ از منوی Project به قسمت Reference بروید و از لیست ظاهر شده گزینه Microsoft ActiveX Data Objects را فعال کنید و دکمه OK را کلیک کنید.

حال به سراغ کد نویسی شی Connection می رویم. ابتدا باید این شی را برای برنامه خودمان تعریف کنیم یا به زبان ساده تر یک نمونه از آن را برای خودمان بوجود آوریم. این کار به روش زیر انجام می شود.

Dim [Name] As New ADODB.Connection

نکته که در اینجا باید ذکر کنیم انعطاف پذیری ADO است لذا روشی که در اینجا برای اتصال به بانک بکار می رود لزوماً ترتیب الزامی نیست یعنی شما می توانید به طرق دیگر نیز به نتیجه پایانی کار دست پیدا کنید. در زیر نحوه اتصال به یک بانک اطلاعاتی توسط متد Open و پارامترهای اختیاری UserID و Password آورده ام.

[ConnectionString].Open ConnectionString, UserID, Password

حالت کلی آرگومان اول بصورت زیر است.

Provider=[Provider Name] ;Data Source=[Source]

بطور نمونه در زیر به بانکی با تکنولوژی Jet و نسخه ۴,۰ آن متصل شده‌ایم که براحتی می‌توانید نسخه و حتی تکنولوژی Provider را نیز تغییر دهید.

[ConnectionName].Open "Provider=Microsoft.jet.oledb.3.51;Data Source=C:\Example.mdb"

دو تا آرگومان آخر هم برای مواقعی است که بانک با پسورد محافظت شده باشد.

خوب بریم سراغ شی RecordSet این شی به صورت زیر تعریف می‌شود.

Dim [Name] As ADODB.Recordset

حالت کلی آن نیز برای اتصال به یک جدول بصورت زیر است.

Open (Source, ActiveConnection, CursorType, LockType, Options)

آرگومان اول نام جدولی است که می‌خواهیم به آن متصل شویم. آرگومان دوم نام شی Connection ماست (این نام دقیقاً نام شی‌ای است که در مرحله قبل تعریف کردیم و با استفاده از آن به بانک متصل شدیم).

در مورد آرگومان سوم (CursorType) مقادیر زیر قابل انتساب است.

adOpenForwardOnly: رکوردست بصورت یک طرفه بر روی رکوردها حرکت می‌کند و جلو می‌رود.

adOpenKeySet: یک رکوردست که کاربران هم می‌توانند با استفاده از آن جدول را تغییر دهند. توجه کنید که رکوردهای ساخته شده توسط سایر کاربران قابل دسترسی برای شما نیست.

adOpenDynamic: یک رکوردست مانند گزینه قبلی که کاربران هم می‌توانند با استفاده از آن جدول را تغییر دهند. توجه کنید که برخلاف گزینه قبل رکوردهای ساخته شده توسط سایر کاربران قابل دسترسی برای شما نیز هست.

adOpenStatic: این رکوردست یک رکوردست ایستا خواهد بود. یعنی تغییرات بر آن اثری ندارد.

در مورد آرگومان چهارم (LockType) مقادیر زیر قابل انتساب هستند.

adLockReadOnly: امکان تغییر رکورد وجود ندارد.

adLockOptimistic: رکوردها در هنگام Update رکوردست قفل می‌شوند.

adLockPessimistic: رکوردها در هنگام آغاز ویرایش قفل می‌شوند.

adLockBatchOptimistic: هنگامی که متد UpdateBatch روی رکوردست اعمال می‌شود رکوردها قفل می‌شوند.

برای آرگومان آخر (Options) نیز مقادیر زیر قابل انتساب هستند.

adCmdText: مقدار آرگومان Source متن دستور مورد نظر می‌باشد. بطور مثال یک دستور SQL

adCmdTable: مقدار آرگومان Source نام جدولی است که تمامی فیلدهای آن برگردانده خواهند شد.

ساده‌ترین حالت استفاده از متد Open به شکل زیر است که بدون درد سرترین حال آن است و برای استفاده‌های معمولی استفاده می‌شود.

`[Name].Open "[TableName]", [ConnectionString], adOpenKeyset, adLockOptimistic, adCmdTable`

دقت کنید که آرگومان دوم از نوع String نیست و نیازی به علامت [""] ندارد.

در ادامه چندتا از متدهای پر کاربرد شی RecordSet را شرح می‌دهیم. خصوصیت زیر یک Property از نوع Boolean است که مشخص می‌کند که آیا رکوردست بر روی رکورد آخر قرار دارد یا خیر.

`[RecordsetName].EOF`

خصوصیت زیر دقیقاً مانند Property قبلی است با این تفاوت که مشخص می‌کند که آیا رکوردست بر روی رکورد اول قرار دارد یا خیر.

`[RecordsetName].BOF`

متد زیر رکوردست را به رکورد بعدی منتقل می‌کند.

`[RecordsetName].MoveNext`

متد زیر رکوردست را به رکورد قبلی منتقل می‌کند.

`[RecordsetName].MovePrevious`

متد زیر رکوردست را به اولین رکورد انتقال می‌دهد.

[RecordsetName].MoveFirst

متد زیر رکوردست را به آخرین رکورد انتقال می‌دهد.

[RecordsetName].MoveLast

متد زیر رکوردست را به رکوردی دلخواه که توسط شما مشخص می‌شود می‌برد.

[RecordsetName].Move (NumRecords As ADO_LONGPTR, [Start])

متد زیر یک رکورد جدید به جدول اضافه می‌کند.

[RecordsetName].AddNew

متد زیر تغییرات انجام شده در رابطه با جدول و مقادیر فیلدها را اعمال می‌کند.

[RecordsetName].Update

متد زیر برای جستجو در جدول بکار می‌رود.

[RecordsetName].Find

متد زیر رکوردی را که رکوردست بر روی آن قرار دارد را حذف می‌کند.

[RecordsetName].Delete

Property زیر تعداد رکوردهای جدول را در خود نگه می‌دارد.

[RecordsetName].RecordCount

متد زیر نیز برای هر دو شی Connection و Recordset کاربرد دارد و اتصال آنها را به ترتیب با بانک اطلاعاتی و جدول قطع می‌کند.

[ConnectionName].Close

[RecordsetName].Close

توجه داشته باشید که پس از بستن اشیا با استفاده از تکنیک زیر آنها را از حافظه خارج کنید.

Set [YourVar] = Nothing

متدهای بالا پر کاربرد ترین متدهای لازم می‌باشد. برای آگاهی از نحوه کارکرد سایر متدها می‌توانید توضیح مربوطه آن متد را مطالعه کنید و برای اطلاعات بیشتر نیز به قسمت ADO یا ADO.net در MSDN بروید.



نگارنده: میلاد ایران نژاد

Mail: milad_irannejad@yahoo.com

URL: <http://www.MVB.persianblog.com>