

ذخیره کردن تصاویر در بانک اطلاعاتی

مقدمه:

یک راه ساده برای ذخیره و بازیابی تصاویر در برنامه های شما استفاده از VB و بانک اطلاعاتی اکسس مایکروسافت ارائه شده بوسیله اندی رز بروک و استن شولتز چه بسا نقشه شما برای گرافیکهای موجود در برنامه تان، ساخت فرمهای ورود داده یا طراحی بانک اطلاعاتی برای سایت وب شما، یک روش ساده برای تغییر و اضافه کردن تصاویر در بانک اطلاعاتی، اجازه می دهد برنامه های شما انعطاف پذیر تر و قابل حمل تر باشند. در این بخش ما روشهای متفاوتی را برای ویرایش اشیا تصاویر یا اشاره گرهای فایل در داخل بانک اطلاعاتی برنامه شما متمرکز خواهیم شد.

مشتریان اغلب درباره نحوه نمایش لوگو و تصاویر متعددشان بصورت محلی در برنامه سوال می کنند. پروژه های وب اکثرا نیاز دارند که هر رکورد، یک تصویر مرتبط با داده ها داشته باشد. فراهم کردن تصاویر یک راه ساده برای ارتباط بصری یک آیتم به رکورد داده است. مشکل این است که این تصاویر بصورت ثابت تغییر می کنند و مشتریانی بطور دائمی اینها را اضافه می کنند. برنامه شما باید به مشتریان اجازه دهد که انعطاف پذیری مورد نیازشان را داشته باشند.

ویژوال بیسیک چندین روش مختلف را برای ذخیره تصاویر پیشنهاد می کند، مانند استفاده از کنترل ImageList، اما این روشها نیاز به کمپایل مجدد برنامه دارند و عموماً محدود به تعداد مشخصی از تصاویر می باشند. بهترین راه، ذخیره تصاویر در یک بانک اطلاعاتی است. شما می توانید این کار را به دو طریق انجام دهید: بوسیله ذخیره تصاویر بصورت یک شی بزرگ باینری (BLOB) در یک فیلد بانک اطلاعاتی، یا بوسیله ذخیره ساده یک اشاره گر به محل فایل روی دیسک. هر روش معایب و مزایایی دارد. ذخیره یک تصویر بصورت یک BLOB می تواند اندازه بانک اطلاعاتی شما را حجیم کند، البته همه تصاویر شما در یک مکان مرکزی خواهند بود. ذخیره تصاویر بصورت اشاره گر فایل اجازه می دهد بانک اطلاعاتی شما بصورت قابل ملاحظه ای کوچک باشد، اما نبودن و یا خراب بودن فایلها می تواند موجب مشکلات بعدی شود.

بنا به سلیقه شما، باید توازن اندازه و تاثیر را برای روشی که بعنوان بهترین روش در برنامه های مجزا است، انتخاب کنید. ما بطور اختصار هر دو روش را با استفاده از ADO برای ذخیره و بازیابی تصاویر در یک بانک اطلاعاتی 97 Access شرح می دهیم. اگر بخواهید از یک بانک اطلاعاتی 2000 Access استفاده کنید، این کد به همین صورت می باشد اما شما باید آخرین نسخه از بسته مولفه های دسترسی به داده مایکروسافت (MDAC) را دانلود و نصب کنید.

ساختن بانک اطلاعاتی:

در اینجا یک برنامه ساده برای ذخیره یک کتابخانه تصویر اصلی خواهید ساخت. قبل از اینکه با یک پروژه VB کار را شروع کنیم، شما باید بانک اطلاعاتی که برای ذخیره داده ها می خواهید استفاده کنید را بسازید. Access را اجرا کرده و یک جدول جدید با نام ImageLibrary بسازید. فیلدهای جدیدی را که نیاز دارید به همراه مشخصات آنها اضافه کنید (جدول ۱ را ببینید).

بعد از ساختن بانک اطلاعاتی، VB را اجرا کنید، منوی Project\References را انتخاب کرده و گزینه Microsoft ActiveX Data Object 2.1 را انتخاب کنید (یا شما می توانید ADO 2.5 Library را اگر نصب شده باشد استفاده کنید). نام پیش فرض فرم را به frmMain تغییر دهید. فرمتان را بنا به نیازتان برای ذخیره تصاویر طراحی کنید. پروژه ImageLibrary طوری تنظیم شده است که به شما اجازه می دهد چگونگی ذخیره تصاویر در بانک اطلاعاتی را انتخاب کنید. (شکل ۱ را ببینید و پروژه نمونه را از آدرس زیر دانلود کنید).

شما باید متغیرهایی را برای ارجاع به Recordset و Connection برای بانک اطلاعاتی تعریف کنید. از این متغیرها برای دسترسی و مرور بانک اطلاعاتی استفاده خواهید کرد و همینطور برای اضافه کردن و یا ویرایش یک رکورد. این متغیرها را به قسمت اعلان فرم frmMain اضافه کنید:

```
Private Conn As ADODB.Connection
Private rs As ADODB.Recordset
```

سپس شما باید با بانک اطلاعاتی و رکوردست ارتباط برقرار کنید. مطمئن شوید که فایل ImageLib.mdb در شاخه برنامه شما قرار دارد و این کد را در رویداد Foem_Load فرم frmMain قرار دهید:

```
'Establish the connection.
Set Conn = New ADODB.Connection
Conn.ConnectionString = _
    "Provider=Microsoft.Jet.OLEDB" & _
    ".3.51;Data Source=" & App.Path & _
    "\ImageLib.mdb"
Conn.Open
```

```
'Open the recordset
Set rs = New ADODB.Recordset
rs.Open "ImageLibrary", Conn, _
    adOpenKeyset, adLockPessimistic, adCmdTable
```

حالا که شما رکوردست را باز کرده اید، می توانید دیدن و ویرایش داده های رکورد را نیز آغاز کنید. ما با نگاه کردن در چگونگی ذخیره و بازیابی تصویر واقعی در بانک اطلاعاتی با استفاده از متدهای GetChunk و AppendChunk شروع می کنیم. این دو متد را برای نمایش یک فیلد BLOB ذخیره شده در بانک اطلاعاتی استفاده می کنند. در ابتدا بسادگی خواص DataSource و DataField کنترل Image را به رکوردست و فیلد BLOB مقداردهی می کنیم:

```
Set imgDBImage.DataSource = rs
imgDBImage.DataField = "ImageBLOB"
```

روشی را که مشاهده می کنید عموماً ساده ترین راه برای شروع نمایش تصاویر ذخیره شده در بانک اطلاعاتی است. به هر صورت شروع فایل‌های بزرگتر ممکن است به خاطر حافظه سیستم شما در تلاش کنترل Image برای بارگذاری از بانک اطلاعاتی طاقت فرسا باشد. اگر مقدار حافظه مصرفی برای برنامه شما بدلیل فایل‌های بزرگ مسئله مهمی است، باید از روش دوم (متد GetChunk در ADO) برای بازیابی اشیای باینری از بانک اطلاعاتی استفاده کنید. متد GetChunk به شما اجازه می دهد تعداد بایت (Chunk) را از فیلد BLOB بازیابی کنید. شما می توانید اشیای ذخیره شده را در این Chunk های کوچک (برای تصاویر بزرگ) یا یک Chunk تکی بارگذاری کنید.

```
strData = rs("ImageBLOB"). _
    GetChunk(rs("ImageBLOB").ActualSize)
```

متد GetChunk یک زبان دارد: از آنجا که خاصیت Picture کنترل Image نمی تواند آرایه بایتی بارگذاری شده از بانک اطلاعاتی را بخواند، شما باید ابتدا یک فایل موقت از فیلد داده را ذخیره کنید و سپس آن فایل را با متد LoadPicture بخوانید. علیرغم این نقص، متد GetChunk واقعا سریعتر از مقداردهی خواص داده کنترل Image است.

تشخیص بهترین روش:

ما یک آزمایش کوچک برای تشخیص سریعترین روش بوسیله ساختن یک جدول با ۱۰۰ رکورد انجام دادیم. هر رکورد شامل داده های باینری حدوداً به اندازه ۵۰ کیلوبایت بود. ما در میان همه رکوردها در جدول تا وقتی که داده های ذخیره شده در یک کنترل Image نمایش داده شوند گشت و گذار کردیم. روش DataField در ۵۸ ثانیه کامل شد در حالی که روش Chunk در ۲۷ ثانیه (با یک Cunk به اندازه ۱۰۰ بایت). چون این تصاویر مجزا خیلی بزرگ نبودند، ما محتویات شی را در یک Chunk تکی بازیابی کردیم. این روش در سریعترین زمتن در ۱۸ ثانیه نتیجه داد. شما باید برای استفاده در برنامه خودتان بهترین روش را تشخیص دهید. در نهایت برای این مقاله استفاده از روش GetChunk با یک Chunk به اندازه ۱۰۰ بایت در نظر گرفته شد. (لیست ۱ را نگاه کنید).

با بازیابی تصاویر ذخیره شده، ذخیره یک تصویر در بانک اطلاعاتی نیاز به این دارد که شما در ابتدا آن را از تصویر به یک آرایه بیتی تبدیل کنید. رویهم رفته شما می توانید بایتهایی را در فیلدهای بانک اطلاعاتی با استفاده از متد AppendChunk بنویسید. در این کد شما می توانید این کار را بوسیله نوشتن داده ها در فیلد در یک مرحله با استفاده از تنظیم اندازه Chunk برابر اندازه کل فایل تصویر، انجام دهید. (لیست ۲ را نگاه کنید).

در پروژه نمونه، اگر شما گزینه ذخیره تصویر بصورت اشاره گر فایل را انتخاب کرده باشید، فقط مسیر فایل در بانک اطلاعاتی ذخیره می شود و این اندازه بانک اطلاعاتی را کاهش و سرعت برنامه را افزایش می دهد. روال FillField در لیست ۱ ابتدا تشخیص می دهد که فیلد ImagePath در بانک اطلاعاتی دارای مقدار است یا خیر. اگر شما یک مسیر را بصورت یک رکورد ذخیره کرده باشید پس تصویر بصورت اشاره گر فایل ذخیره شده است. اگر مسیر ذخیره نشده باشد، پس تصویر بصورت BLOB ذخیره شده است (شرح متد GetChunk را نگاه کنید). شما می توانید یک اشاره گر را بسادگی با استفاده از متد LoadPicture کنترل Image نمایش دهید.

فایده آشکار ذخیره تصاویر بصورت اشاره گر فایل این است که فقط مسیر فایل ذخیره می شود. در نتیجه بانک اطلاعاتی شما بصورت ناراحت کننده ای رشد نمی کند مانند آنچه که در ذخیره تصاویر بصورت فیلد BLOB وجود دارد. در مثالی که تشریح شد، با ۱۰۰ رکورد از تصاویر ۵۰ کیلو بیتی ذخیره شده در فیلدهای BLOB، بانک اطلاعاتی شما در حدود ۴ مگابایت بزرگ می شود. همان بانک اطلاعاتی در صورت استفاده از اشاره گر فایل به جای آن، زیر ۱۰۰ کیلوبایت بزرگ می شود. در مقایسه سرعت، روش اشاره گر فایل برنده است، در این زمینه آزمایش ما در ۵ ثانیه کامل شد. این برتری عموماً روش اشاره گر به فایل را روش برتر در ذخیره تصاویر می سازد.

بزرگترین عیب استفاده از اشاره گرهای فایل این است که اگر محل فایل تغییر کرد، برنامه شما نمی تواند تصویر را بارگذاری کند. عیب دیگر آن عدم قابلیت حمل برنامه می باشد. اگر شما برنامه را به جای دیگری ببرید، باید بانک اطلاعاتی را به همراه همه تصاویر به آن محل منتقل کنید. یکی از راههای آسان کردن این مشکل این است که به جای محل واقعی روی دیسک از تبدیل نام جهانی فایل (UNC) استفاده کنیم (مانند landyr/images/test.jpg به جای c:\images\test.jpg). البته اگر برنامه برای جای دیگری غیر از مشتری فرستاده شود، احتمالاً آنها به همان منبع شما دسترسی ندارند (مگر اینکه شما اشاره گر فایل را بصورت تصویر محلی در اینترنت مانند www.yourSite.com/Images/test.jpg ذخیره کرده باشید). این احتمالاً تنها جایی است که توصیه شده است تصاویرتان را بصورت اشیای باینری در بانک اطلاعاتی ذخیره کنید. توازن بین سرعت و اندازه بانک اطلاعاتی آنقدر بارز نیست مگر اینکه پیش بینی کنید که تغییر محل فایل، برنامه شما را متوقف خواهد کرد.

در پایان ممکن است ندانید که این روشها را در SQL Server چگونه استفاده کنید. متدهای AppendChunk و GetChunk اغلب برای بانکهای اطلاعاتی SQL Server کالر می کنند، مایکروسافت در این زمینه دلسردتان می کند زیرا فیلدهای BLOB در جداول SQL باعث تنزل اجرا در سرور می شوند. مایکروسافت روشی را که برای استفاده از جداول SQL پیشنهاد کرده است، ذخیره اشاره گر به محل فایل است.

ذخیره همه تصاویر در بانک اطلاعاتی ممکن است در ابتدا زمان زیادی را بگیرد، اما ظاهر سریع را به شما اعطا خواهد کرد. شما حالا می توانید یک فرم ورود داده سریع را برای بروز کردن تصاویر سایت وبتان، طراحی کنید. شما دیگر احتیاج به کمپایل مجدد در صورت تغییر تصاویر برنامه هایتان ندارید. سعی کنید در پروژه های بعدیتان از تصاویر ذخیره شده در بانک اطلاعاتی استفاده کنید و تفاوت آن را در انعطاف پذیری و قابل حمل بودن آن، برای کاربرانان ببینید.

اندی رزبروک: اندی مالک شرکت Axe است (<http://www.axeinc.com>) . یک شرکت برنامه نویسی و توسعه وب که با VB و دیگر سکوه‌های توسعه کار می کند. با اندی با آدرس andy@axeinc.com می توانید ارتباط داشته باشید.

استن شولتز: یک مدیر پروژه و توسعه دهنده حرفه ای وب و VB در ساراسوتا است. استن مدرک MCP در VB دارد و در کنفرانس DevDays از شرکت میکروسافت درباره توسعه VB سخنرانی کرد. او یک ویرایشگر برای VBPG است و برای این مجله مقاله می نویسد. با استن با آدرس stan@vbexpert.com می توانید مکاتبه داشته باشید.

منبع: سایت شرکت میکروسافت

مترجم: **وبلاگ بادوی ویزوال بیسیکا** <http://vblog.persianblog.com>