

# Zpravodaj moderní Programování 7/2013: Binární data v databázi

*Obtížnost: pokročilí*

Databáze jsou primárně navrženy pro ukládání strukturovaných dat členěných až na dílčí, nedělitelné údaje, jako jsou texty, čísla nebo data. Někdy je však za pomoci databáze zapotřebí sledovat také větší bloky binárních dat - obrázky, dokumenty (DOC, PDF) ap. Pro takové případy jsou k dispozici v zásadě dva přístupy:

1. Příslušný soubor s obrázkem nebo dokumentem uložit někde bokem ve vyhrazené složce a do databáze zapsat pouze cestu k souboru jako obyčejný textový řetězec;
2. Zapsat příslušná binární data přímo do databáze.

Každý z přístupů má své výhody. První přístup asi přivítáte, když máte databázi omezenou kapacitně, např. na webhostingu, a nechcete si ji „zaplácat“ obvykle velikými binárními daty.

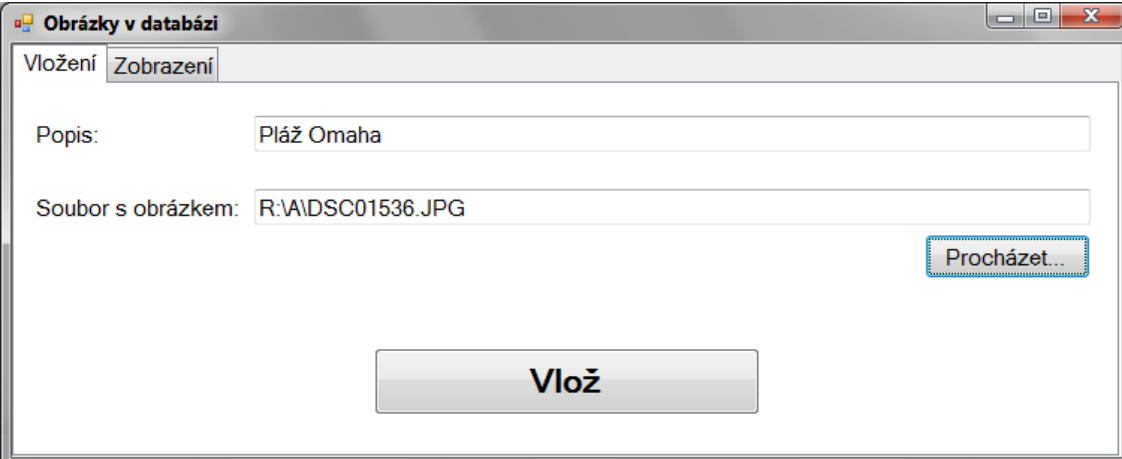
Druhý přístup vám naopak zjednoduší správu:

- Když smažete záznam z databáze, automaticky zmizí i binární data;
- Když zálohujete databázi, máte v záloze už i ta binární data a podobně.

Pokud jde o programování, tak u prvního přístupu jde pouze o práci se soubory, což asi znáte. V tomto Zpravodaji se tedy zaměříme na druhý přístup. Vytvoříme si aplikaci, která bude do databáze zapisovat obrázky.

## Aplikace

Uživatelské rozhraní si rozčleníme pomocí záložek na dvě části. První bude sloužit ke vkládání nových obrázků:



Obrázky v databázi

Vložení **Zobrazení**

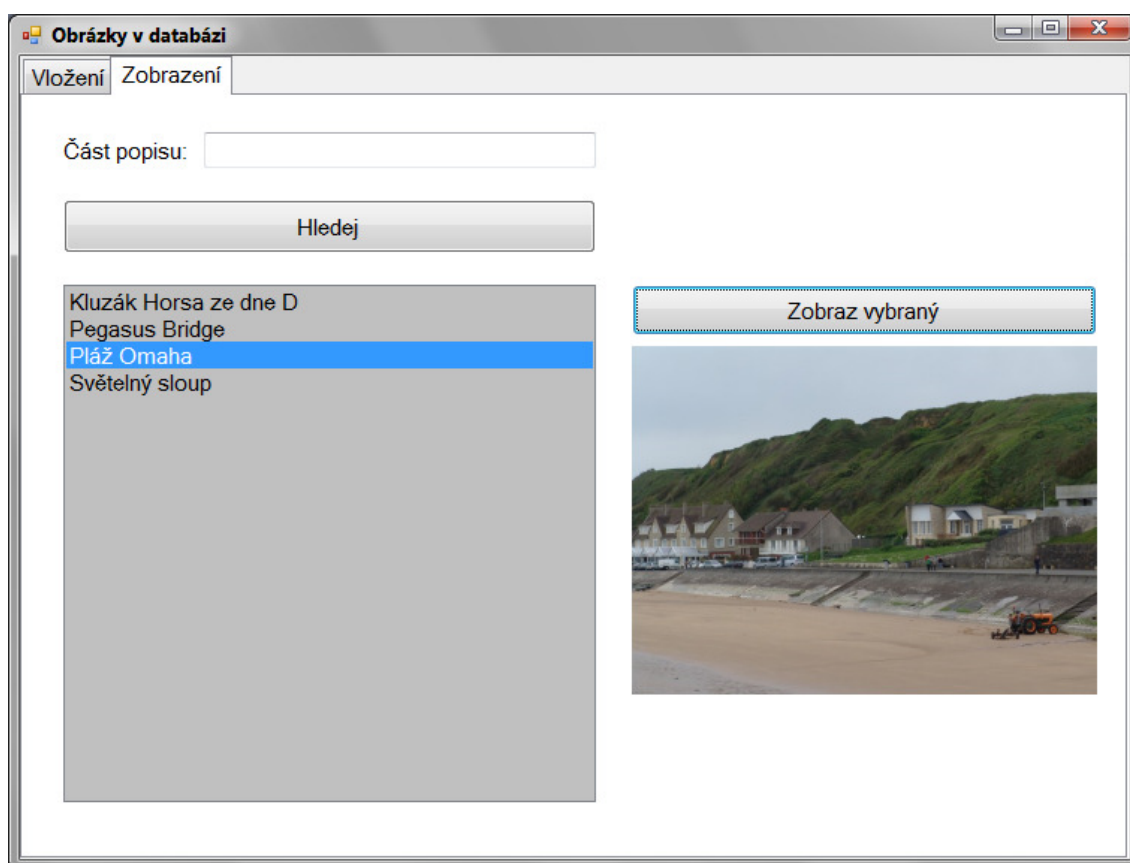
Popis: Pláž Omaha

Soubor s obrázkem: R:\DSC01536.JPG

Procházet...

Vlož

Druhá záložka nechá uživatele obrázky hledat a prohlížet:



Textové pole „Část popisu“ umožní ve velkém množství obrázků v databázi vyhledávat dle jeho popisu. Ponechá-li jej uživatel prázdné, vypíše se seznam všech obrázků.

Pro záložky využijeme ovládacího prvku **TabControl** z kategorie „Containers“. Jednotlivé záložky se definují pomocí vlastnosti `TabPage`.

## Databáze

Jako SŘBD použijeme Microsoft SQL Server Express Edition nebo LocalDB. Pomocí Průzkumníka databáze ve Visual Studiu vytvořte novou databázi a v ní jedinou tabulku `Obrázky` s následující strukturou:

	Column Name	Data Type	Allow Nulls
🔑	Klíč	int	<input type="checkbox"/>
	Popis	nvarchar(200)	<input type="checkbox"/>
	Obrázek	varbinary(MAX)	<input type="checkbox"/>

Primární klíč nastavte jako automaticky generovaný (*Identity Specification* > *Is Identity* na „Yes“ ve vlastnostech sloupce `Klíč`). Pro obrázek zadejte datový typ **varbinary(MAX)** - binární data proměnlivé, neomezené délky. Nepoužívejte typ `image`! Ten je již zastaralý a Microsoft nadále nedoporučuje jeho používání v nových aplikacích.

Pro přístup z aplikace použijeme připojený režim. Alternativou by bylo LINQ.

## Princip řešení

To hlavní na celé aplikaci je způsob, jak zapisovat/číst binární data. Celý princip spočívá v tom, že **s těmito daty budeme pracovat jako s polem bytů** - `byte[]`.

## Vkládání obrázků

Pustíme se do jednotlivých technických detailů, nejprve v části vkládání obrázků. Na začátku vyřešíme rutinní úkony.

**Uživatelský výběr souboru s obrázkem:** Do okna programu v Designeru jako obvykle z Toolboxu přetáhnete „OpenFileDialog“ z kategorie „Dialogs“. Obsluha tlačítka „Procházet...“ pak vypadá následovně:

```
private void tlačítkoProcházet_Click(object sender, EventArgs e)
{
    if (oknoVýběruSouboru.ShowDialog() == System.Windows.Forms.DialogResult.OK)
        poleSoubor.Text = oknoVýběruSouboru.FileName;
}
```

**Usingy:** Jelikož budeme pracovat se soubory a s připojeným režimem pro MS SQL Server, je třeba na začátek zdrojového kódu `Form1.cs` přidat tyto řádky:

```
using System.Data.SqlClient;
using System.IO;
```

**Tlačítko „Vlož“:** Teď to již začne být zajímavé. Tlačítko „Vlož“ pracuje dle následujícího výpisu:

```
private void tlačítkoVlož_Click(object sender, EventArgs e)
{
    // Kontrola vstupů
    string popis = polePopis.Text.Trim();
    string jménoSouboru = poleSoubor.Text;
    if (popis == "" || jménoSouboru == "")
    {
        MessageBox.Show("Nutno zadat popis obrázku a vybrat soubor");
        return;
    }

    // Načtení obrázku
    byte[] obrázek;
    try
    {
        obrázek = File.ReadAllBytes(jménoSouboru);
    }
    catch
    {
        MessageBox.Show("Obrázek nelze načíst");
        return;
    }
}
```

```
// Vložení do DB
string připojovacíŘetězec =
    Properties.Settings.Default.PřipojovacíŘetězec;
SqlConnection připojení = new SqlConnection(připojovacíŘetězec);

string textPříkazu =
    "INSERT INTO Obrázky " +
    "(Popis, Obrázek) " +
    "VALUES " +
    "(@Popis, @Obrázek)";
SqlCommand příkaz = new SqlCommand(textPříkazu, připojení);
příkaz.Parameters.AddWithValue("@Popis", popis);
příkaz.Parameters.AddWithValue("@Obrázek", obrázek);

připojení.Open();
příkaz.ExecuteNonQuery();
připojení.Close();

MessageBox.Show("Záznam vložen");
polePopis.Text = poleSoubor.Text = "";
}
```

### Vysvětlení kódu:

- Pro data vkládaného obrázku je vytvořena proměnná `obrázek` v již zmíněném typu `byte[]`. Tato proměnná se naplní čtením ze souboru vybraného uživatelem. K tomu slouží volání `File.ReadAllBytes`;
- Záznam do databázové tabulky vkládáme parametrizovaným SQL příkazem `Insert`. Jako hodnotu parametru `@Obrázek` jednoduše předáme proměnnou `obrázek` s načtenými daty. `Insert` se jako obvykle odesílá voláním `ExecuteNonQuery`;
- V SQL příkazu je třeba si hlídat mezery na koncích řádků, aby se jednotlivá slova SQL příkazu „nescukla“ dohromady;
- Abychom na mnoha místech programu nemuseli opakovat připojovací řetězec do databáze (opakování nikdy není dobré), uložíme jej do konfiguračního souboru aplikace, viz dále.

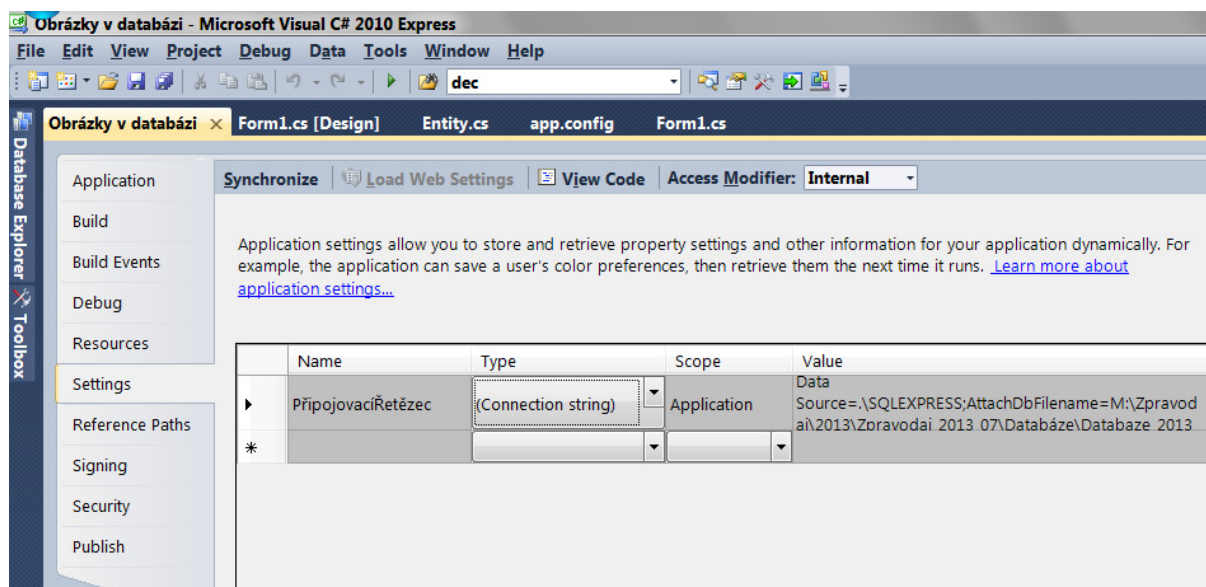
## Konfigurační soubor aplikace

Konfigurační soubor desktopové .NET aplikace je soubor šířený spolu s .EXE souborem aplikace, který má stejné jméno jako .EXE soubor a navíc přidanou příponu `.config`. Data jsou v něm zapsána v lidsky čitelném formátu XML, což znamená, že jej uživatel může snadno editovat a sám si aplikaci „překonfigurovat“. **Mezi nastavení, která se v něm mohou nacházet, patří mimo jiné připojovací řetězce k databázi.**

Konfigurační soubor lze do projektu vložit přes *Project > Add New Item > šablona Application Configuration File*. V projektu se obvykle jmenuje `app.config`, při sestavování programu se zkopíruje do podsložky s .EXE souborem a podle něj se také přejmenuje.

Není třeba jej však tímto způsobem vkládat a ručně editovat. Příjemnější je použití nástroje Visual Studio:

1. Z nabídky *Project* vyberte předposlední položku (*název\_projektu Properties*);
2. Přepněte se na kartu *Settings* (nastavení);
3. U nového nastavení pak zvolte název (např. *PřipojovacíŘetězec*), typ (buď *string* nebo (*Connection string*)), může být obojí) a jako hodnotu zkopírujte připojovací řetězec k vaší databázi.



Toto nastavení pak může uživatel v konfiguračním souboru sám změnit a váš program si jej snadno načte takto:

```
string připojovacíŘetězec =
    Properties.Settings.Default.PřipojovacíŘetězec;
```

## Závěr

Polovina programu je hotova. Do databáze je již možno vkládat obrázky. Pomocí Průzkumníka databáze se můžete přesvědčit, že se tabulka plní. Na prohlížení obrázků si ale musíte počkat do příštího Zpravodaje. V něm program dokončíme.

*Radek Vystavěl, 13. srpna 2013*

*Pokud Vám Zpravodaje moderníProgramování připadají užitečné, doporučte jejich odběr svým známým. Mohou se přihlásit na webu [www.moderniProgramovani.cz](http://www.moderniProgramovani.cz).*