

Jak vznikala webová aplikace kolonizace.asp2.cz

Cílem tohoto textu je poskytnout začínajícím vývojářům rady a zejména návod jak krok za krokem vytvořit jednoduchou statickou webovou aplikaci s použitím co nejmenšího programování, ale zároveň využívající některé šikovné ovládací prvky ASP.NET 2.0.

Motivací pro vznik aplikace a tohoto textu byla autorova účast v soutěži www.lcdzaweb.cz

V tomto článku se čtenář setká s vývojovým prostředím Microsoft Visual Studio 2005, ASP.NET 2.0, jazykem C# a v neposlední řadě s jazykem HTML a CSS. Alespoň drobná znalost těchto technologií není na škodu, ale není nezbytnou podmínkou pro porozumění textu.

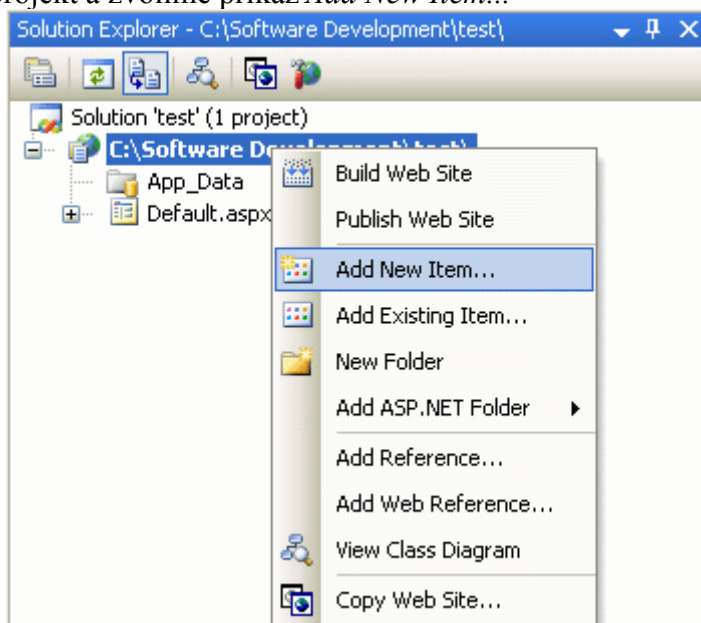
Začínáme...

Ve Visual Studiu 2005 si otevřeme novou webovou aplikaci (File > New > Web Site). V dialogu vybereme *ASP.NET WebSite* a nastavíme umístění aplikace (*Location*) a jazyk (C#, ale možno i VB.NET).

Pro aplikaci použijeme jednoduchý design složený ze dvou vodorovných pruhů, z nichž horní bude tvořit záhlaví stránky a spodní se bude dále dělit na dva svislé sloupce – nabídkový a obsahový. Před příchodem ASP.NET 2.0 bylo třeba definovat kód pro společné části stránky v každém aspx souboru. Pozdější změny byly proto obtížné. ASP.NET 2.0 přichází s novým konceptem návrhu stránek, s tzv. Master Pages.

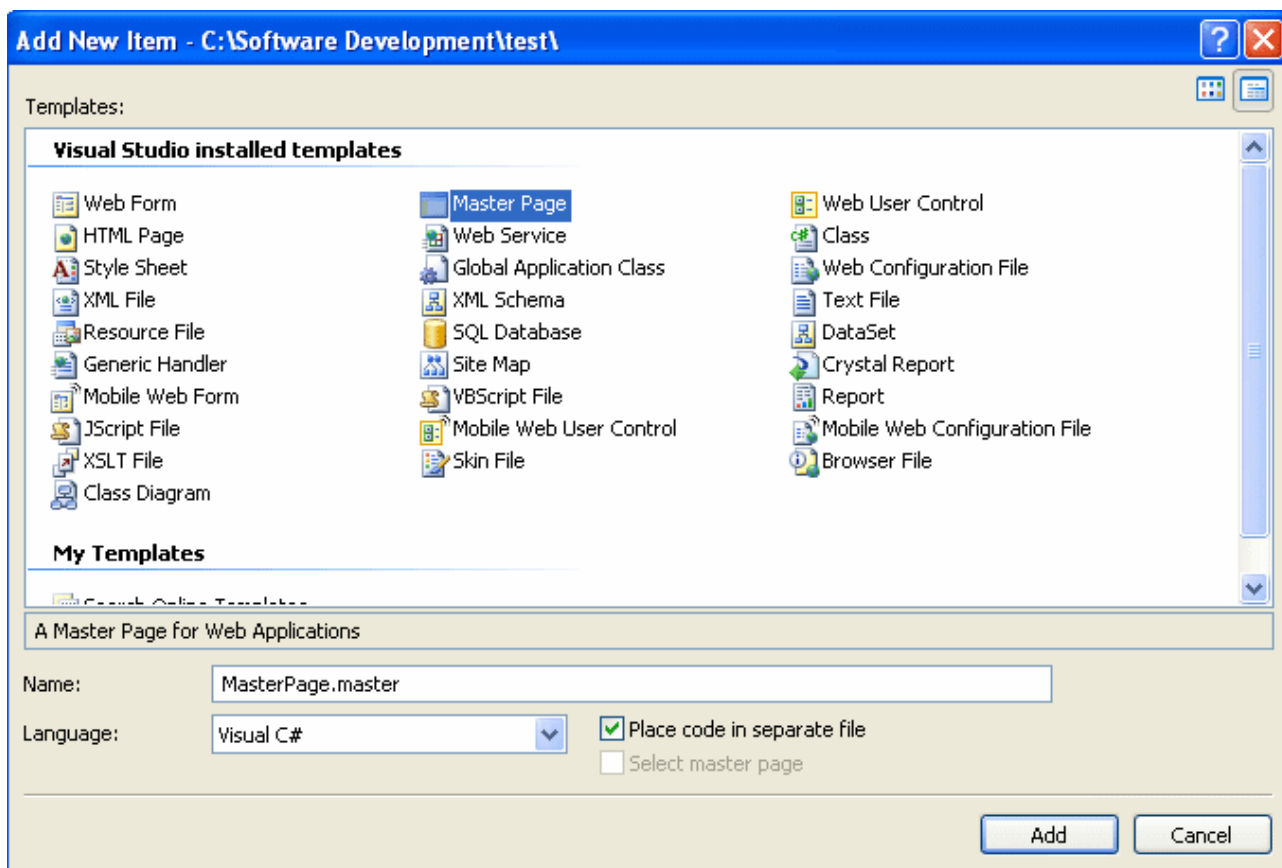
Master page tvoří obálku pro další webové stránky a obsahuje kód společný všem stránkám. Běžné aspx stránky, které mají deklarovanou příslušnost k nějaké master stránce ve svém kódu deklarují pouze svůj obsah. V okamžiku kdy zavoláme aspx stránku, ASP.NET automaticky zařídí svázání se správnou master stránkou. V naší webové aplikaci bude master stránka obsahovat kód pro záhlaví stránky a navigační lištu.

Nyní přidáme master stránku do projektu. V okně *Solution Explorer* (viz. Ilustrace 1) klikneme pravým tlačítkem na projekt a zvolíme příkaz *Add New Item...*



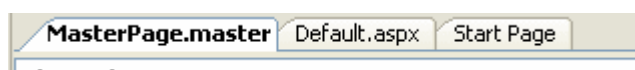
Ilustrace 1: Přidání nového objektu

Objeví se dialogové okno (viz. Ilustrace 2), v kterém zvolíme Master page. S tímto dialogovým oknem se setkáme během práce ještě vícekrát, tak si ho dobře zapamatujeme.

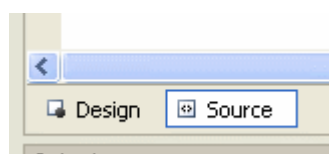


Ilustrace 2: Dialog - Přidání nového objektu do projektu

V okně pro editaci textu zvolíme záložku se souborem *MasterPage.master* (viz. Ilustrace 3) a přepneme se do zobrazení *Source* (viz. Ilustrace 4).



Ilustrace 3: Záložky se soubory



Ilustrace 4: Tlačítka Design/Source

Zdrojový HTML kód souboru *MasterPage.master* nahradíme následujícím kódem, čímž si vytvoříme šablonu pro budoucí stránku. Jak bylo popsáno výše, stránka se bude skládat ze tří bloků – záhlaví, nabídkové lišty a vlastního obsahu. Pro umístění bloků na stránce použijeme CSS, což nám v budoucnosti umožní snadno změnit design bez nutnosti přepisovat HTML soubory.

```

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
Inherits="MasterPage" %>

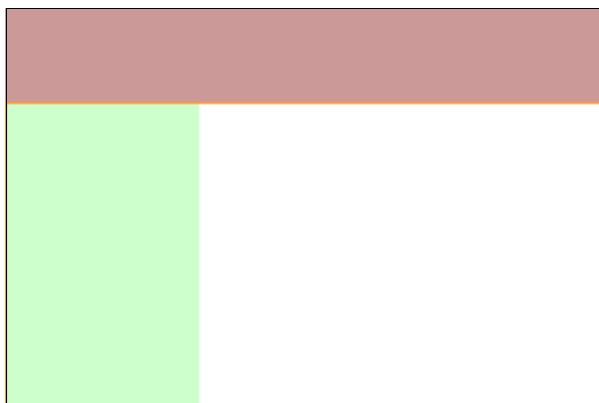
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
    <title>Ukázková aplikace</title>
    <link href="StyleSheet.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <form id="form1" runat="server">
        <div id="page_container">
            <div id="page_header">
            </div>
            <div id="page_body">
                <div id="page_menu">
                </div>
                <div id="page_content">
                    <asp:contentplaceholder id="ContentPlaceholder1"
runat="server">
                    </asp:contentplaceholder>
                </div>
            </div>
        </div>
    </form>
</body>
</html>

```

Rozvržení stránky

Bloky jsou tvořeny HTML tagy `<div>` s jednoznačnými *id* v rámci celého dokumentu. Celá stránka je zabalena do bloku *page_container* (černý rámeček). Horní panel je tvořen blokem *page_header* (fialový), dva sloupce pak bloky *page_menu* (zelený) a *page_content* (bílý), které jsou uzavřeny do bloku *page_body* (oranžový rámeček). Horní panel bude obsahovat logo stránek, levý panel je určen pro nabídku a pravý pro vlastní obsah webu. Obsah souboru *StyleSheet.css*, který deklaruje vlastní CSS styly je popsán a vysvětlen v Příloze 1.

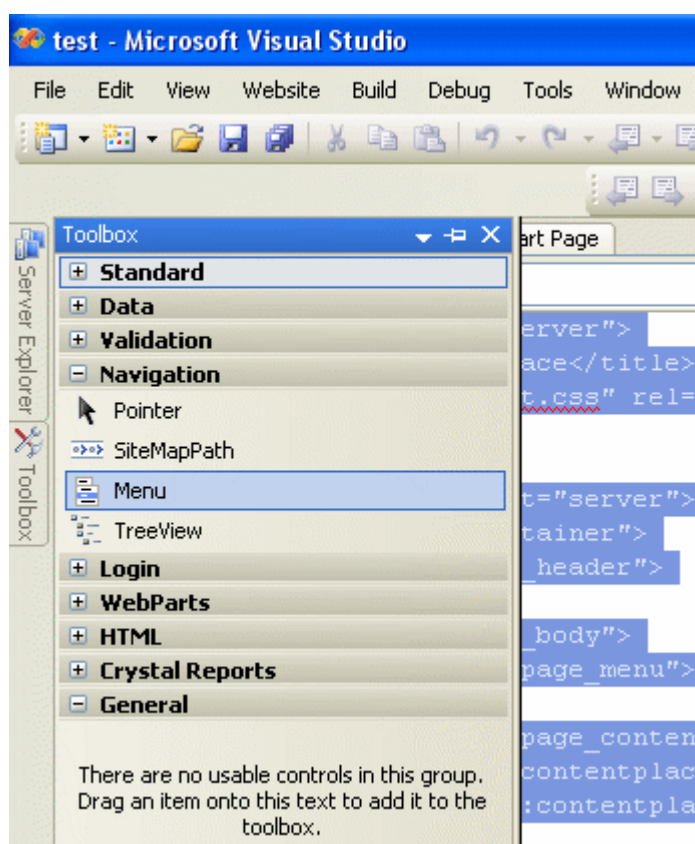


Všimněme si, že blok *page_content* obsahuje ASP.NET ovládací prvek *ContentPlaceHolder*. V tomto prvku se budou zobrazovat vnořené stránky.

Nyní přidáme do projektu soubor s CSS styly postupem popsaným v Příloze 1.

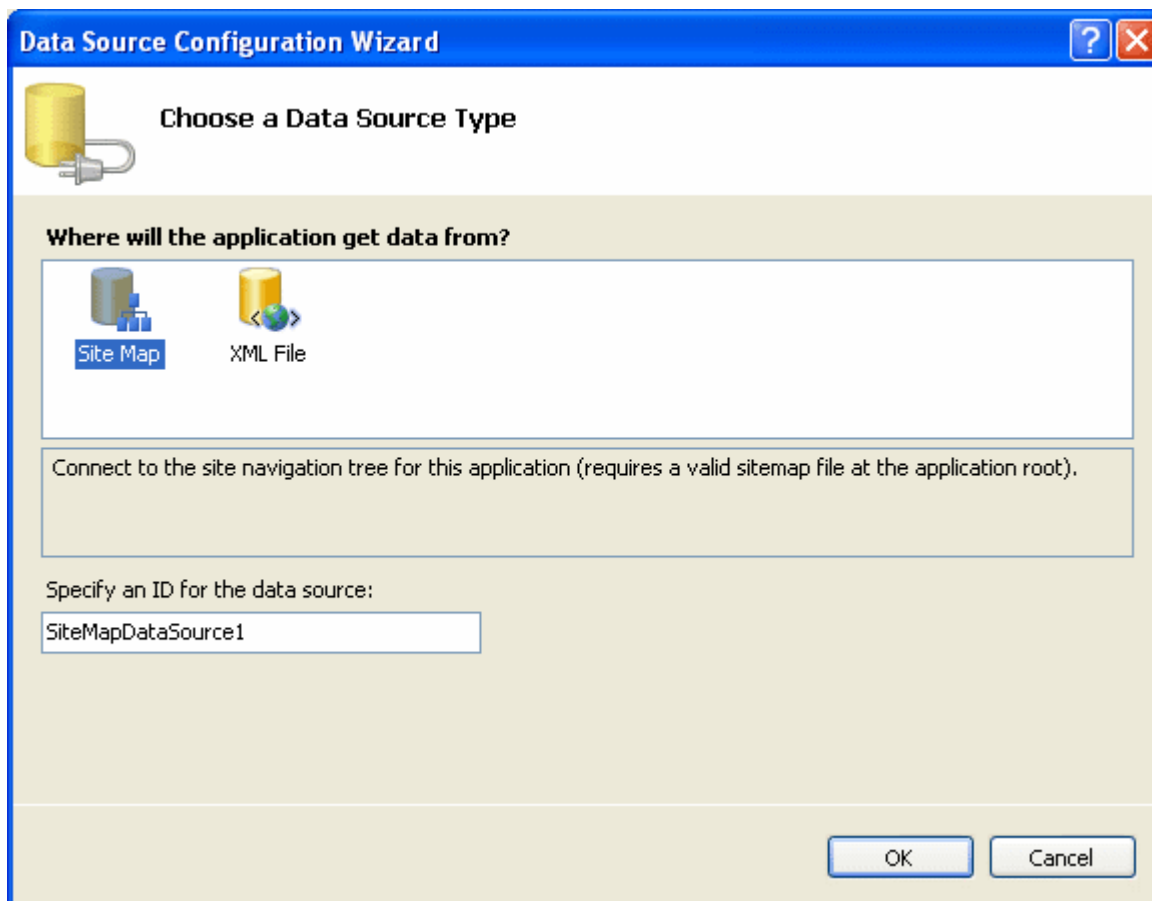
Zvolíme zpátky soubor *MasterPage.master* a z Toolboxu vybereme ovládací prvek *Menu*. Nalezneme jej v sekci *Navigation* (viz. Ilustrace 5). Tažením myši (Drag&Drop) umístíme *Menu* do HTML kódu do bloku *page_menu*, příslušný HTML kód bude nyní vypadat následovně:

```
<div id="page_menu">
  <asp:Menu ID="Menu1" runat="server">
  </asp:Menu>
</div>
```



Ilustrace 5: Výběr prvku *Menu* z Toolboxu

V dalším kroku nadefinuje datový zdroj, z kterého bude *Menu* načítat data. Přepneme se do Design módu (viz. Ilustrace 4) a klikneme na malou šipku v pravém horním rohu grafické podoby *Menu*. Objeví se malé pop-up dialogové okno s nabídkou, v němž je položka *Choose data source*. My vytvoříme nový datový zdroj (<New data source...>), z dialogu (viz. Ilustrace 6) vybereme *SiteMap* a klikneme na *OK*.



Ilustrace 6: Dialog: volba datového zdroje

Na stránce se objevil nový prvek *SiteMapDataSource*. Tento prvek umí načíst strukturu webu ze souboru *Web.sitemap* a předat ji do *Menu*, které poté vytvoří grafickou podobu s položkami odpovídajícími hierarchické struktuře webu.

Soubor *Web.sitemap*

Web.sitemap je XML soubor v kterém snadno popíšeme strukturu webu čili vztahy mezi stránkami. Naše webová aplikace bude pro jednoduchost obsahovat pouze tři stránky. Reálně je vytvoříme za okamžik, ale již nyní můžeme napsat obsah souboru *Web.sitemap*.

Přidáme do projektu novou položku *Site Map* (viz. Ilustrace 2). Otevřeme soubor a jeho obsah upravíme dle následující ukázky:

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="Default.aspx" title="Home" description="">
    <siteMapNode url="Products.aspx" title="Produkty" description="" />
    <siteMapNode url="Contact.aspx" title="Kontakt" description="" />
  </siteMapNode>
</siteMap>
```

Mapa webu vždy musí obsahovat kořenový uzel, v našem případě stránku *Default.aspx*, kterou označíme jako domácí. Hloubka vnořených uzlů není omezena. V naší aplikaci budeme mít dvě

podřízené stránky *Products.aspx* a *Contact.aspx*.

Soubor *Web.sitemap* uložíme a přepneme se zpátky na stránku *MasterPage.master*. Na prvku *Menu* opět klikneme na malou šipku a z pop-up nabídky tentokrát vybereme *Refresh Schema*. Vidíme, že se *Menu* upravilo dle právě vytvořené struktury webu ve *Web.sitemap*.

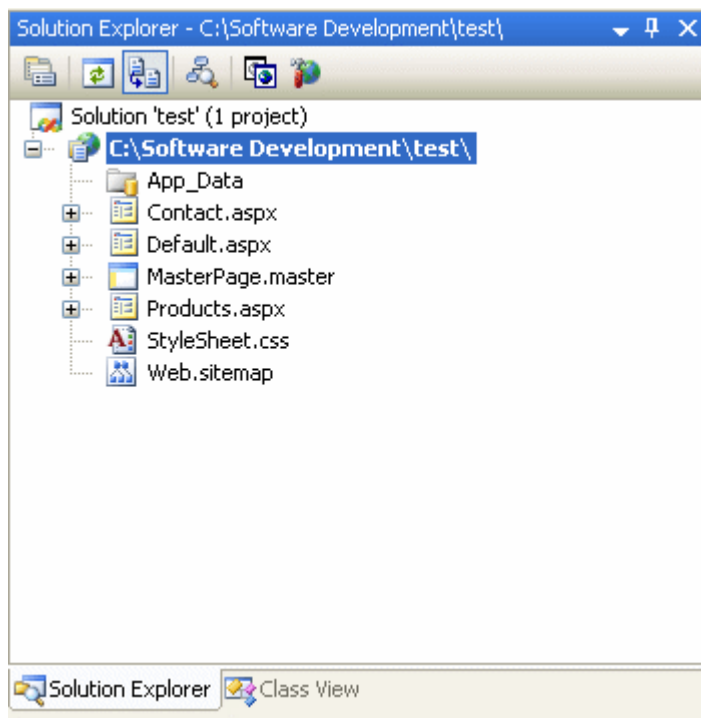
Ovládací prvek *Menu* obsahuje řadu možností jak jej nastavit, zobrazená nabídka se může rozbalovat dynamicky nebo být statická, možné jsou i kombinace obou přístupů. V našem případě budeme chtít menu statické, proto na prvku *Menu* nastavíme vlastnost (property) *StaticDisplayLevels* na hodnotu 2. Jak z názvu vyplývá, tato vlastnost ovlivňuje počet položek, které se vygenerují staticky. Pro *Menu*, které mají větší hloubku (což není náš případ) se zobrazují zbylé položky dynamicky (tj. na kliknutí).

Vytvoření vnořených stránek

Nabídku jsme úspěšně vytvořili, nyní přistoupíme k vytvoření stránek s obsahem. Stránka *Default.aspx* byla vygenerována automaticky s projektem a není propojena s naší master stránkou. Proto ji z projektu odstraníme – v *Solution Exploreru* klikneme na soubor *Default.aspx* pravým tlačítkem myši a zvolíme příkaz *Delete*.

Následně do projektu přidáme tři nové objekty typu *Web Page* (viz. Ilustrace 2) – *Default.aspx*, *Products.aspx* a *Contact.aspx*. V dialogu nesmíme opomenout zaškrtnout check box *Select master page*, k čemuž budeme vyzváni hned vzápětí v dalším dialogu. Náš projekt obsahuje pouze jednu master stránku, takže se nemůžeme výběrem splést.

Ve výsledku by měl náš projekt vypadat takto



Přepneme se na stránku Default.aspx a zvolíme pohled do kódu (Source). První, čeho si pravděpodobně všimneme je absence deklarace HTML hlavičky a těla. Jak bylo popsáno výše, vnořené stránky přejímají deklarace od nadřazené Master page a sami obsahují pouze kontejner pro svůj obsah (*ContentPlaceHolder*). Tento kontejner naplníme textem a soubor uložíme. Kód stránky by nyní měl vypadat obdobně následující ukázce:

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    Vítejte na stránkách firmy <b>A je to</b>.
</asp:Content>
```

Obdobně vytvoříme obsah stránek Products.aspx a Contacts.aspx.

SiteMapPath navigátor

Vnořené stránky obohatíme o zobrazení aktuální cesty ve stromu stránek, prvek zvaný anglicky breadcrumb. Z *Toolboxu* vybereme prvek *SiteMapPath* a umístíme ho do kódu stránky Products.aspx a Contact.aspx hned na začátek kontejneru. Kód bude vypadat následovně:

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="Products.aspx.cs" Inherits="_Products"
Title="Produkty firmy A je to" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
    <asp:SiteMapPath ID="SiteMapPath1" runat="server">
</asp:SiteMapPath>
    Firma <b>A je to</b> pro vás vytvoří webovou prezentaci snů.
</asp:Content>
```

Nemusíme nic nastavovat, prvek sám pozná na které je stránce a cestu vytvoří na základě souboru *Web.sitemap*.

Hlavička webu

Nakonec nesmíme zapomenout na záhlaví stránek. Opět se přepneme do souboru MasterPage.master a do bloku *page_header* umístíme obrázek nebo text.

Kompilace a publikace webu

Aplikace je hotová, nyní můžeme přistoupit k první zkoušce a spustit ji. V nabídce Visual Studia zvolíme *Debug > Start Debugging* nebo stiskneme klávesu F5. Visual Studio se nás možná zeptá, zda chceme vytvořit soubor *Web.config* a povolit v něm debugování aplikace. Zvolíme Ano.

Po malé chvíli, kdy probíhá kompilace aplikace se otevře okno prohlížeče a můžeme aplikaci otestovat.

Publikování webu

Pokud vše funguje jak má, můžeme přistoupit k poslednímu kroku a publikovat aplikaci na webovém serveru. Visual Studio 2005 (kromě verze Web Developer) obsahuje šikovnou funkci *Publish Web*, ke které se dostaneme kliknutím pravé myši na projekt v Solution Exploretu.

Do zvoleného adresáře, který by měl být prázdný, vytvoří Visual Studio kopii našeho projektu vhodnou k publikaci na webu. Odstraní se všechny soubory s C# kódem a další soubory, které nejsou pro běh aplikace potřeba. Aplikace se zároveň předkompiluje.

Příloha 1. CSS soubor

Podrobný popis vlastností stylů CSS je nad rámec tohoto článku a proto zde uvádíme pouze obsah souboru StyleSheet.css, tak jak jej máme přidat do projektu

... obsah souboru

Další typy a triky

Menu bez kořenového uzlu

Navigace webových aplikací obvykle neobsahuje kořenovou položku od které by se odvíjely podřízené kategorie, ale přímo několik rovnocenných kategorií. ASP.NET takovou konstrukci přímo nepodporuje, ale lze ji docílit několika triky.

V souboru Web.sitemap ponecháme kořenový uzel nevyplněný a uzel Home přidáme do druhé úrovně.

```
<?xml version="1.0" encoding="utf-8" ?>
<sitemap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="" description="">
    <siteMapNode url="Home.aspx" title="Home" description="" />
    <siteMapNode url="Products.aspx" title="Produkty" description="" />
    <siteMapNode url="Contact.aspx" title="Kontakt" description="" />
  </siteMapNode>
</sitemap>
```

* na stránce s Menu pozměnit properties tak, aby se Menu zobrazovalo bez kořene