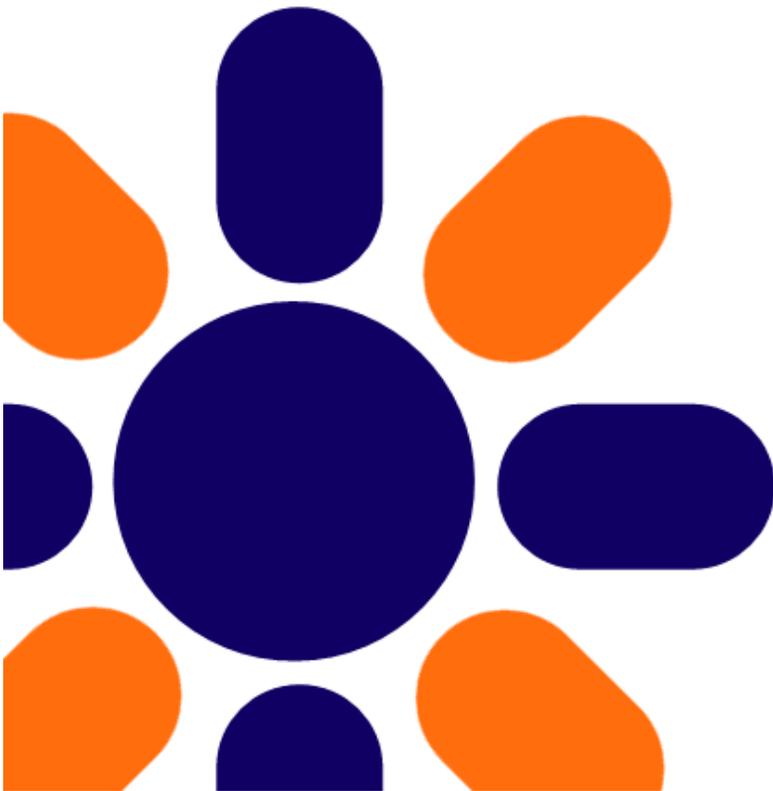


# Kentico CMS Tutorial ASPX 4.1



# Table of Contents

<b>Part I Introduction</b>	<b>5</b>
1 Kentico CMS Overview.....	5
<b>Part II Installation</b>	<b>7</b>
1 Prerequisites.....	7
2 Setup installation.....	8
3 Web application installation.....	9
4 Database setup and Corporate Site.....	14
<b>Part III Managing content</b>	<b>20</b>
1 User interface overview.....	20
2 Editing home page content.....	23
3 Creating a new page.....	24
4 Inserting an image.....	27
5 Creating a link.....	29
6 Creating a news item.....	32
<b>Part IV Site Development Overview</b>	<b>36</b>
1 Site Development Overview.....	36
<b>Part V Creating pages using ASPX templates</b>	<b>41</b>
1 ASPX page templates.....	41
2 Creating a simple ASPX page template.....	44
3 Using the master pages.....	49
<b>Part VI Managing styles and design</b>	<b>53</b>
1 CSS styles .....	53
2 App themes.....	55
3 Menu design.....	56
<b>Part VII Creating a new site using ASPX templates</b>	<b>60</b>
1 Overview .....	60
2 Creating a new web site using wizard.....	60
3 Creating the CSS stylesheet.....	67
4 Opening and configuring the web project.....	69
5 Master page.....	70
6 Main menu .....	74

---

7	Home page .....	75
8	News page .....	80
9	Services page.....	86
10	Products page.....	91
	Overview .....	91
	New document type .....	91
	Transformations .....	97
	Page template .....	100
11	Search page.....	104
12	Secured section for partners.....	108
<b>Part VIII Further steps</b>		<b>114</b>
1	Further steps.....	114

**Part**



# 1 Introduction

## 1.1 Kentico CMS Overview

Kentico CMS for ASP.NET helps you create powerful dynamic web sites with minimum effort. This document will guide you through the most important features of the system step-by-step, so that you can start creating your own web sites.

This document was written for evaluators and new users. It's intended for developers who create the web sites. It's not intended for end-users without programming knowledge.

If you need a more detailed documentation of some features, please see one of the following documents:

- Developer's Guide
- Web parts and Controls Reference
- API Reference
- Database Reference



### **Kentico CMS Support**

You get free technical support during your evaluation period. If you need any help, please visit <http://www.kentico.com/Support.aspx>

**Part**



---

## 2 Installation

### 2.1 Prerequisites

Before you start the installation, please make sure you have the following software installed:

- **Operating system:** Microsoft Windows 2000, XP, or Vista, Microsoft Windows Server 2000, 2003 or 2008
- **Web server:** Internet Information Services (IIS) or Visual Studio 2005 or 2008 or Visual Web Developer 2005 or 2008.
- **Database server:** Microsoft SQL Server 2000, Microsoft SQL Server 2005 or Microsoft SQL Server 2005 Express Edition, Microsoft SQL Server 2008. The database server can be installed on a remote server.
- **Development tools:** Visual Studio 2005 or 2008 or Visual Web Developer 2005 or 2008.

#### Required experience

Although Kentico CMS allows you to create dynamic web sites without programming, you may need to create custom web parts or to add custom code when developing a more complex web site. You should be able to create a simple application in ASP.NET 2.0 using Visual Studio 2005 and have some experience with relational databases and SQL, so that you can leverage the flexibility of Kentico CMS.

## 2.2 Setup installation



### Troubleshooting installation issues

If you encounter any problems during the installation, please see **Kentico CMS Developer's Guide -> Installation and deployment -> Troubleshooting installation issues** or contact our support at <http://www.kentico.com/Support.aspx>

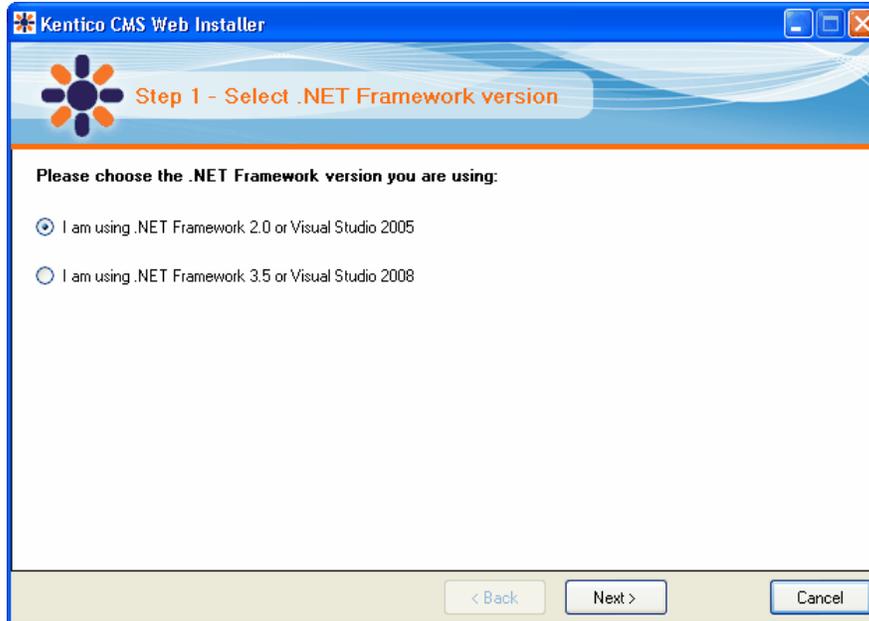
Run KenticoCMS.exe and follow the installation wizard:



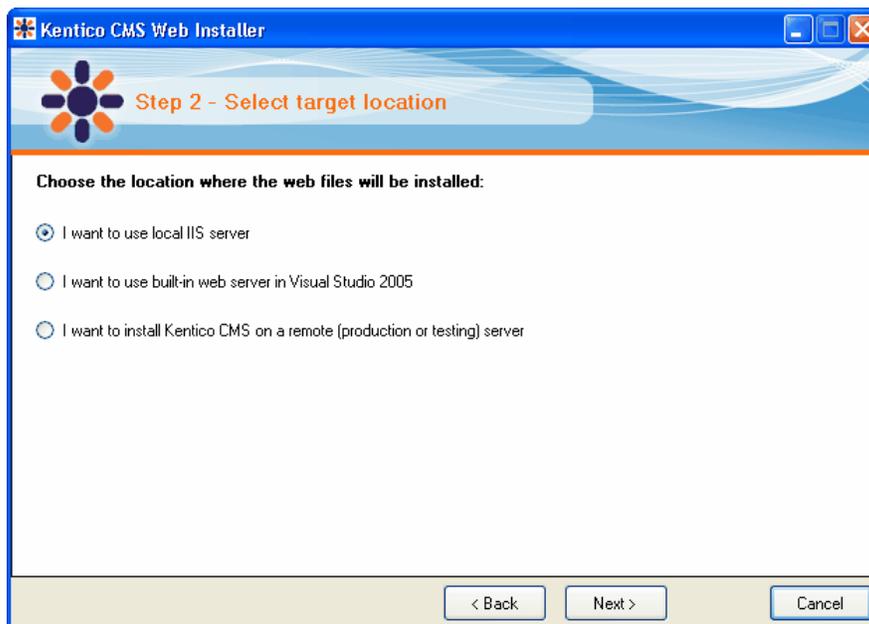
Read and accept the license agreement and click **Next**. Choose the installation location of the binary files and documentation on your disk. After the setup completes the installation, choose to **Launch Kentico CMS Web Installer** and click **Finish**.

## 2.3 Web application installation

Now you should see the Kentico CMS Web Installer. If you cannot see it, you can always run it from **Start menu -> All Programs -> Kentico CMS 3.x -> Kentico CMS Web Installer**.



First, you need to choose the version of .NET Framework and Visual Studio that you wish to use. Click **Next**.

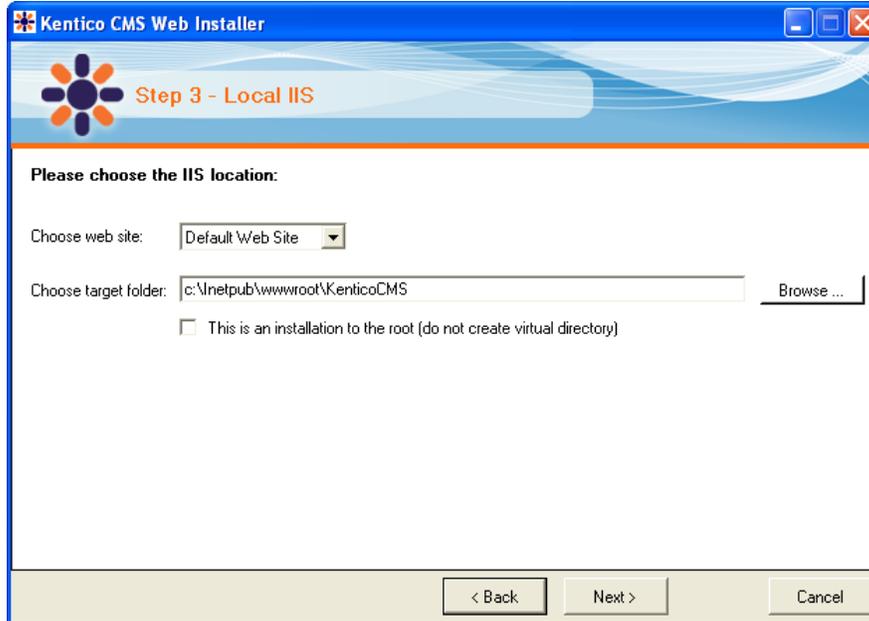


Choose to use either IIS server or Visual Studio 2005 built-in web server (if you do not have IIS installed). Click **Next**.

## IIS installation

If you choose the IIS server, you can choose the web site where the virtual directory will be created and the folder on your local disk where project files will be deployed. The installer will create a new virtual directory on your server and configure it for ASP.NET. Click **Next**.

**Please note:** if you're installing Kentico CMS into the root of your web site (such as <http://www.domain.com>) and do not wish to create a virtual directory (such as <http://www.domain.com/cms>), please check the **This is an installation to the root (do not create virtual directory)** check-box.



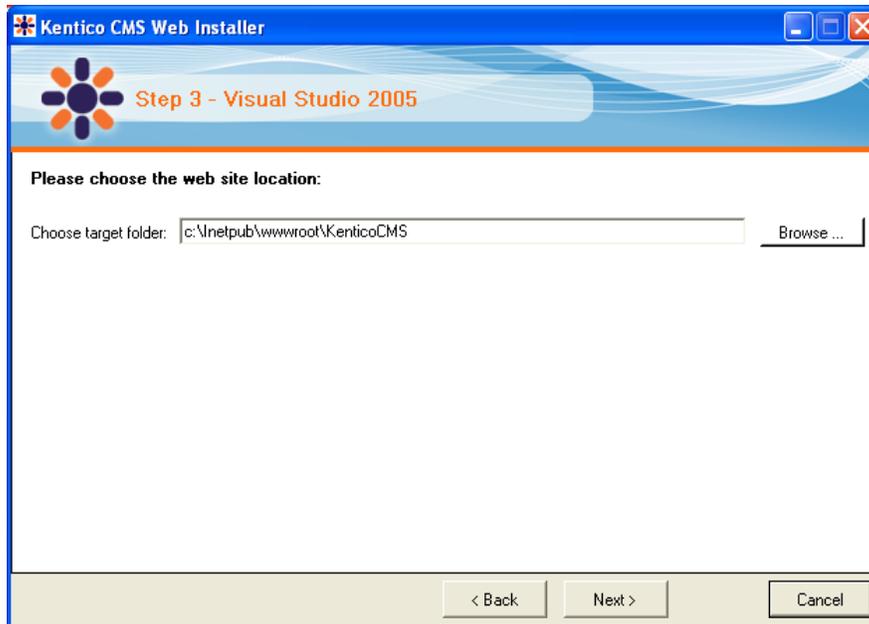
The screenshot shows the 'Kentico CMS Web Installer' window at 'Step 3 - Local IIS'. The window title bar includes the Kentico logo and standard window controls. The main content area is titled 'Please choose the IIS location:' and contains the following elements:

- 'Choose web site:' with a dropdown menu currently set to 'Default Web Site'.
- 'Choose target folder:' with a text input field containing 'c:\inetpub\wwwroot\KenticoCMS' and a 'Browse ...' button to its right.
- A checkbox labeled 'This is an installation to the root (do not create virtual directory)' which is currently unchecked.

At the bottom of the window, there are three buttons: '< Back', 'Next >', and 'Cancel'.

## VS2005 installation

If you chose to use the built-in server in VS2005, you only need to specify the local disk where the project files will be deployed. Click **Finish**.

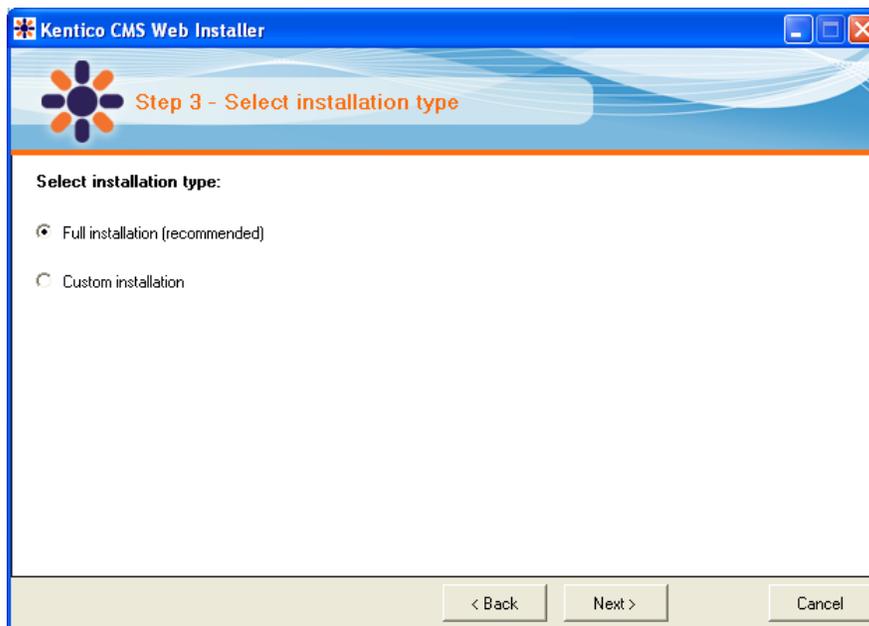


## Type of installation

No matter if you chose the IIS or VS installation, the next step after specifying the target folder is the selection of installation type. The following two types of installation are available:

- **Full installation** - this is the recommended option for the purposes of this tutorial; in this type of installation, all components of the CMS will be installed
- **Custom installation** - in this type of installation, one extra step will be displayed, letting you choose which components you want to include in the installation

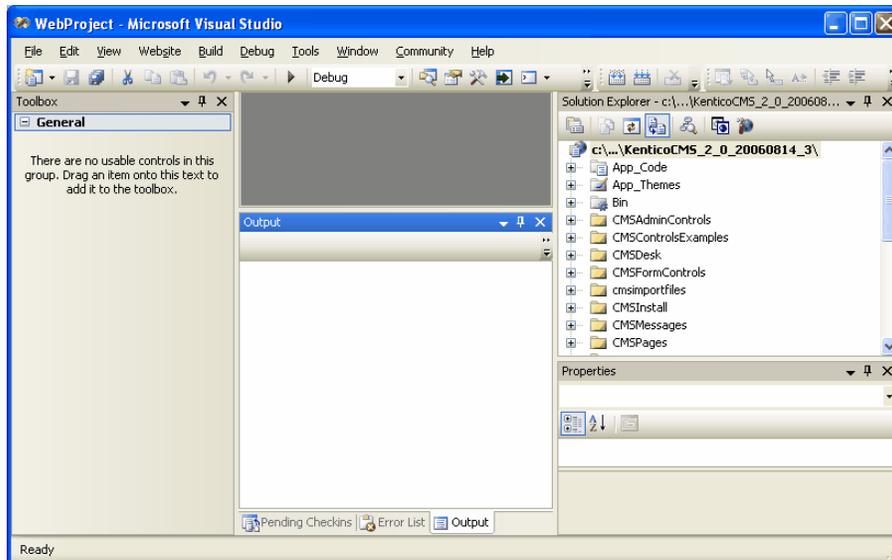
Click **Next** to start the installation process.



After the setup copies all files, you will see the link for opening the web application in browser or the web project in Visual Studio. **Click the link.**

## Opening the web site in VS2005

If you chose VS2005 installation, the project is opened in VS2005:



Choose **Debug -> Start without debugging** from the main menu. The site will be displayed in new browser window using the built-in web server.

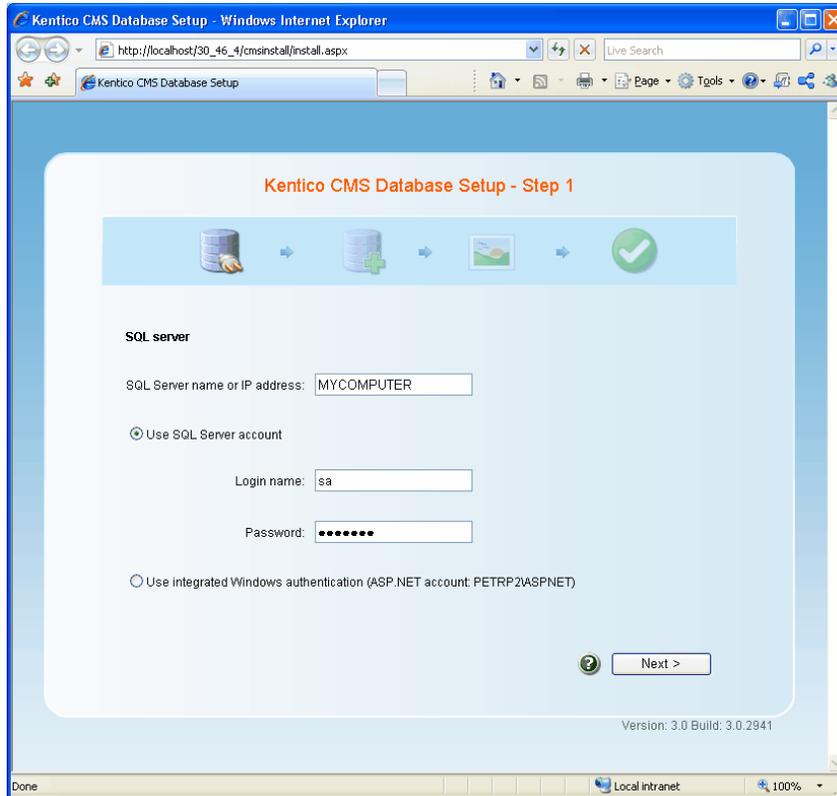


### When you cannot open the web site in Visual Studio 2005

If the link for opening the project in Visual Studio doesn't work, you may need to start Visual Studio manually and choose **File -> Open -> Web Site...** and locate the project folder on your disk manually.

## 2.4 Database setup and Corporate Site

Now you should see the **Database setup** in your web browser.



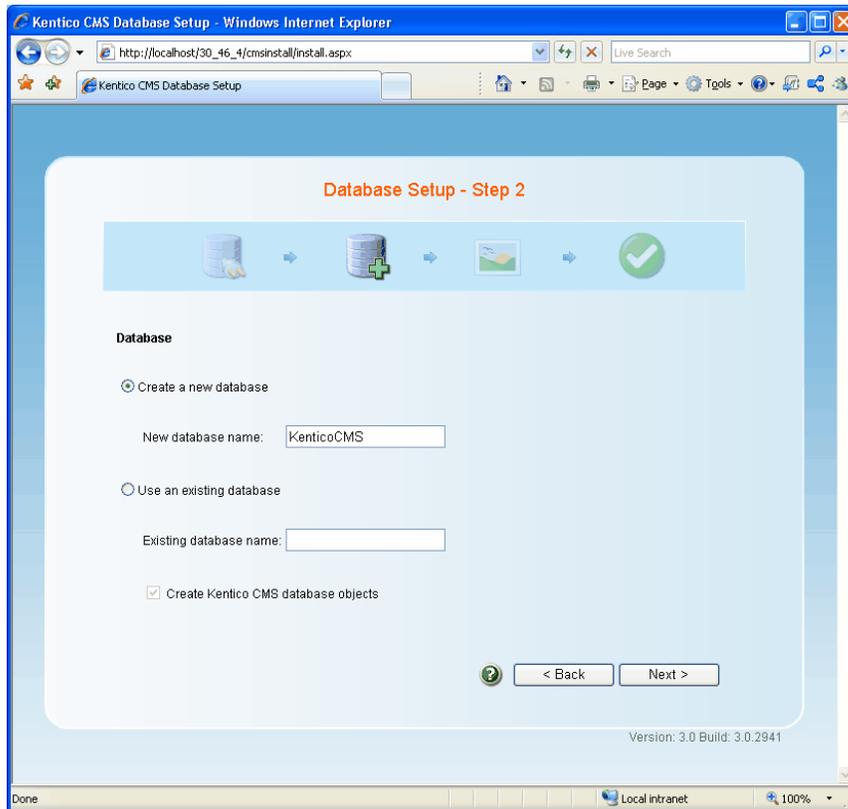
Choose the SQL Server name or IP address. If you're using SQL Server 2005 Express Edition, the default server name is `.\SQLEXPRESS` or `(local)\SQLEXPRESS`.

You can use either SQL Server authentication (recommended) or integrated Windows authentication.

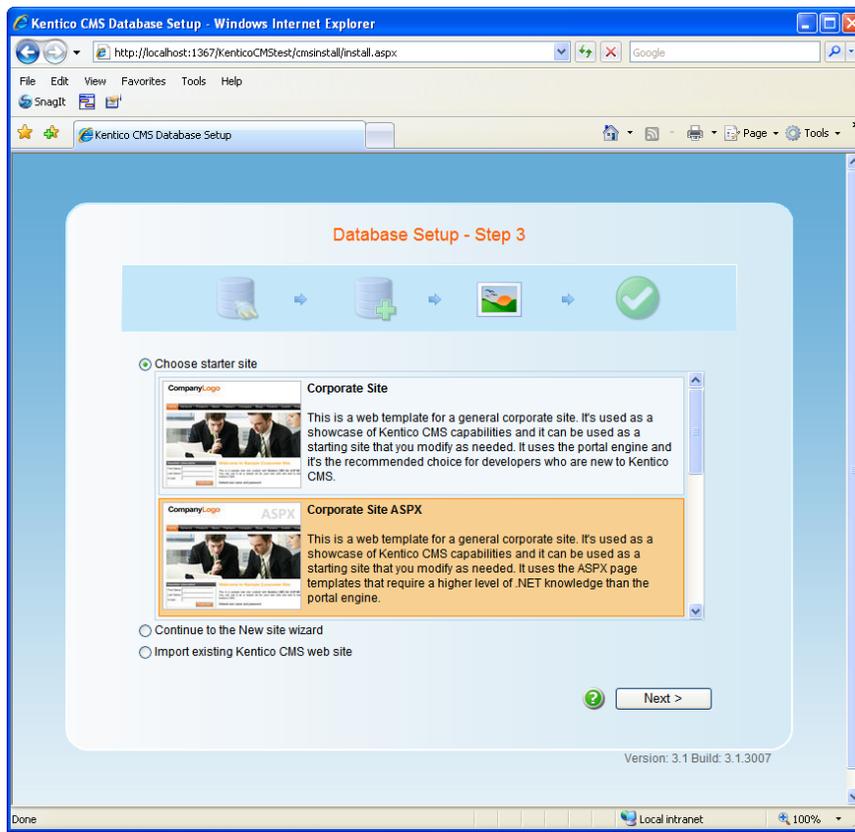
- In case you use **SQL Server account**, you need to enter the user name (such as **sa**) and password.
- In case you use **Windows authentication**, you need to ensure that the ASP.NET account of the name displayed in the brackets has an appropriate login name in your SQL Server.

The account must be granted with permissions for creating new databases or for creating database objects in an existing database.

Click **Next**.



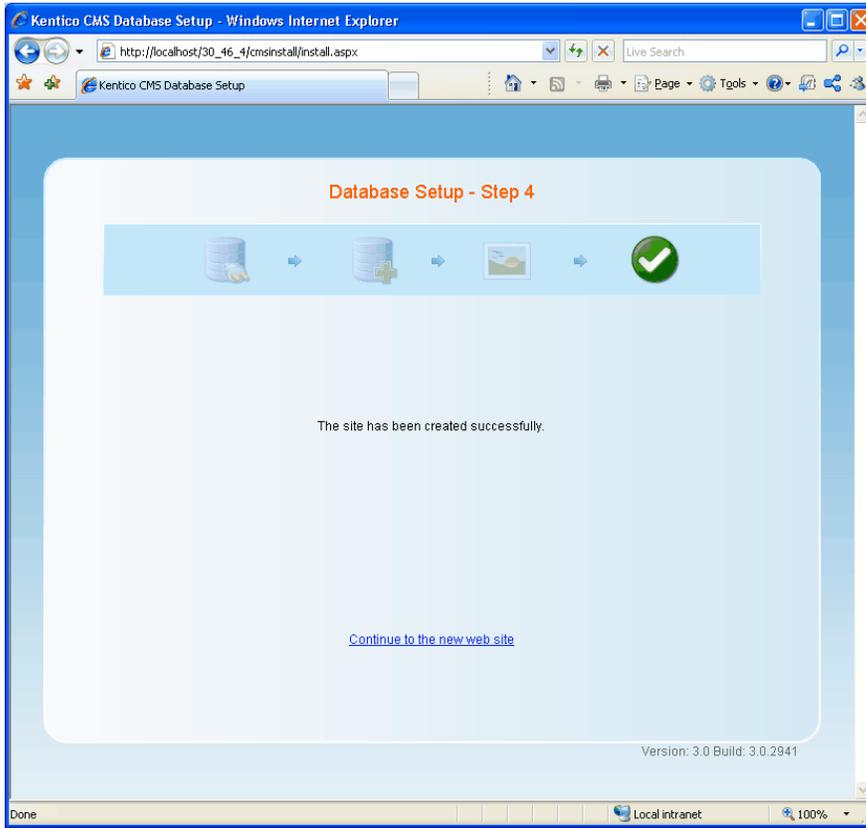
Choose to use an existing database or to create a new one. Click **Next**. If no error occurs, you should see the last step:



You can choose from the following Site options:

- **Choose a starter site:**
  - **Corporate Site (portal engine)** - this option installs the sample corporate site - it is **recommended** for most users, especially **for evaluators**.
  - **Corporate Site (ASPX templates)** - this option is recommended only for experienced ASP.NET developers who want to use ASPX page templates instead of portal templates.
  - **E-commerce site** - this sample site can be used as a starting point for creating your own e-shop and shows the possibilities of Kentico CMS's E-commerce module.
  - **Personal site** - this is a web template suitable for a simple personal site.
  - **Community site** - complex web template suitable for community webs, showing Kentico CMS's social networking features in practice.
  - **Blank site** - this is a blank site without any content; you will use it to create a new site from scratch.
  - **Blank site ASPX** - the same as above, but for ASPX poage templates.
- **Continue to the New site wizard** - this option is recommended if you're starting a new site from scratch.
- **Import an existing Kentico CMS web site** - use this option is you already created a web site with Kentico CMS and need to import it into the new installation (e.g. on the production server).

For the purpose of this guide, please choose to **create sample Corporate Site (ASPX templates)** and click **Next**. You will see the confirmation and a link to your new web site:



Click the link and you will be redirected to the title page of the sample Corporate Site:



**Sample web site**

The Corporate Site web site is only an example of a web site you can create with Kentico CMS. You have full control over the site structure, design, page layout and functionality as you will see in the following chapters.

**Part**



## 3 Managing content

### 3.1 User interface overview

Click the **Sign in to CMS Desk** link at the top of the web site or go to <http://<domain>/<virtualdirectory>/cmsdesk>. You will be asked for user name and password.



#### Default user name and password

The default user name is **administrator** with **blank password**.

It's highly recommended that you change the password before you publish the web site on the live server.

Once you sign in, you will see a splash screen, giving you some basic information. Click the **Continue** button, you will be redirected to the following page:

The screenshot shows the CMS Desk interface with several callout boxes pointing to specific features:

- Document actions:** Points to the toolbar containing icons for New, Delete, Copy, Move, Up, and Down.
- Main menu:** Points to the top navigation bar with tabs for Content, My Desk, Tools, and Administration.
- View mode selection:** Points to the Edit, Preview, and List buttons in the toolbar.
- Page editing mode:** Points to the WYSIWYG editor toolbar.
- WYSIWYG editor:** Points to the main content editing area.
- Content tree:** Points to the left-hand sidebar showing a hierarchical tree of site content.
- Editable/viewing space:** Points to the main content area where text and links are being edited.

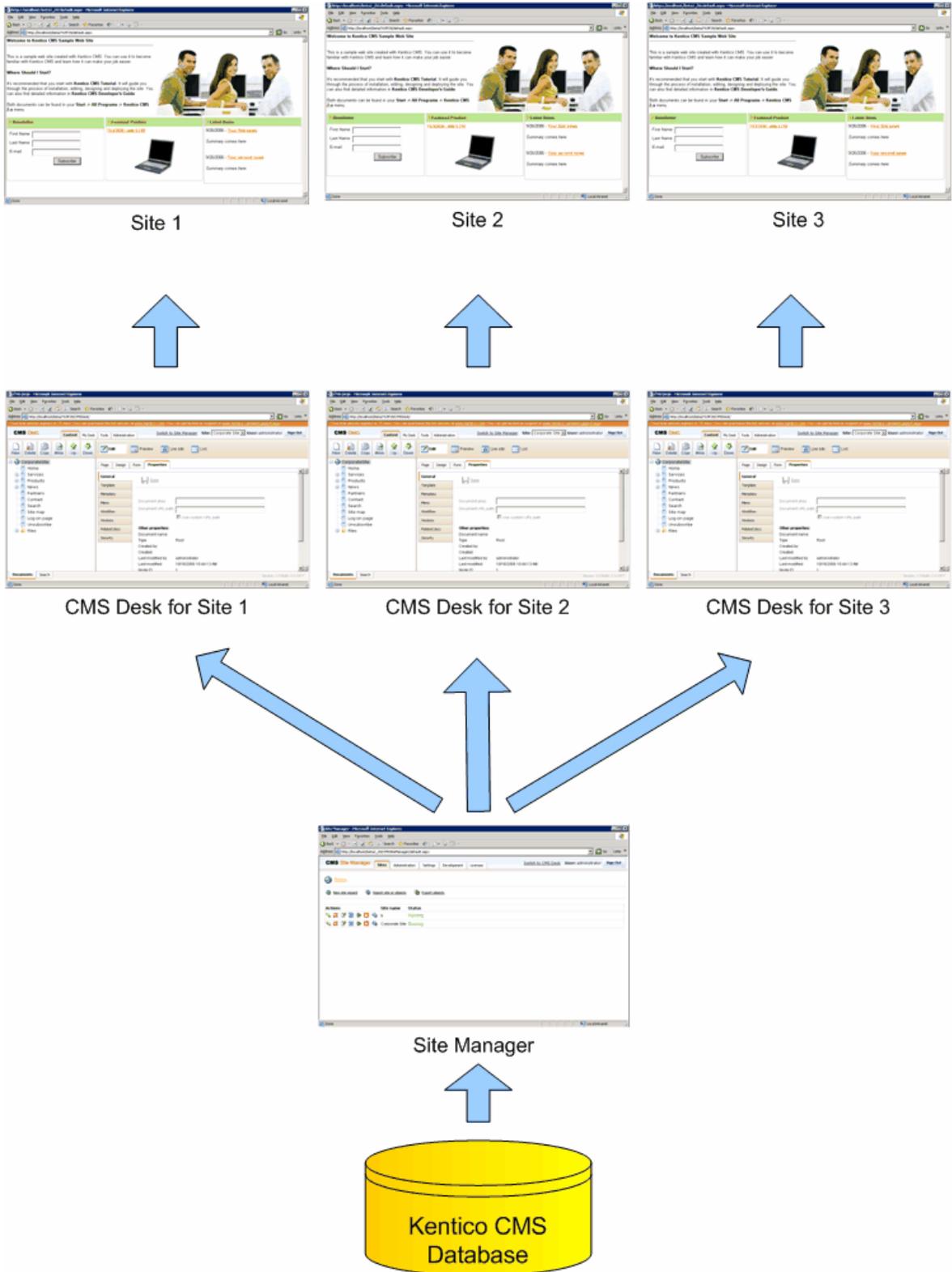
The interface includes a top navigation bar with 'Switch to Site Manager', 'Site: Corporate Site ASPX', and 'User: Global Administrator'. The main content area displays a 'CompanyLogo' and a 'Services' section with a list of services: Web Development, Web Design, and Network Administration. A search bar and a shopping cart icon are also visible.

The user interface consists of the following main sections:

- **Main menu** with Content, My Desk, Tools and Administration sections.
- **Document actions** toolbar with buttons for creating new documents, deleting, copying, moving and sorting documents.
- **View mode** selection that allows you to choose between editing, preview, live view and list view.
- **Content tree** that represents the site map of the web site and allows you to organize the structure of documents and choose document that appears on the right side of the screen.
- **Page editing mode** - you can choose to edit page content, design the page template, edit the document fields, product properties or document properties.
- **Editing/viewing space** - here you can view/edit the document selected in the content tree in the mode selected in the view mode toolbar and in the page editing mode toolbar.

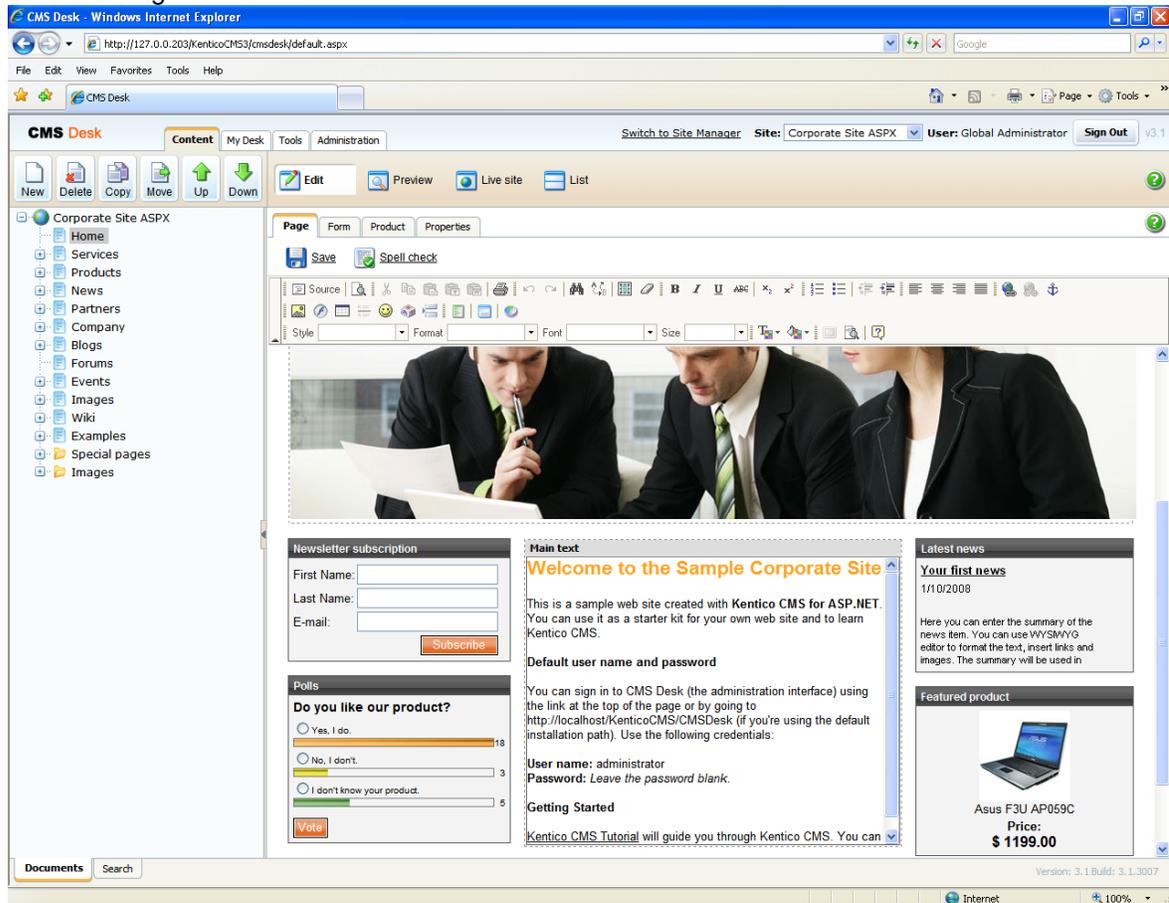
## CMS Desk and Site Manager

**CMS Desk** allows content editors to edit content of a single web sites. Developers and site administrators who need to manage settings, code and configuration of all web sites, can use also the **Site Manager** interface. The Site Manager interface is accessible either through the `<web project>/CMSsiteManager` URL or by clicking the **Switch to Site Manager** link at the top of the CMS Desk user interface. The following figure shows how the database, Site Manager, CMS Desk and web sites are related:



## 3.2 Editing home page content

Now we will modify the home page content. Click **Home** in the content tree. You will see a page like this on the right side:



The page is now displayed in the editing mode with editable region. Delete all content from the editable region and enter the following text:

This is my first text.

You can then use the WYSIWYG editor toolbar at the top of the page to change the formatting of the text like this:

*This is my **first** text.*

Click the **Save** button at the top of the page or press **CTRL+S** to save the changes.

Now click the **Live site** button in the main toolbar. You will see the modified version of the home page as it's displayed to the site visitors.



### Preview mode

If you click the **Preview** mode now, it will display the same content as the **Live site** mode. It works as a preview mode only if you set up workflow. Then, you can preview the latest modifications before they are published.

## 3.3 Creating a new page

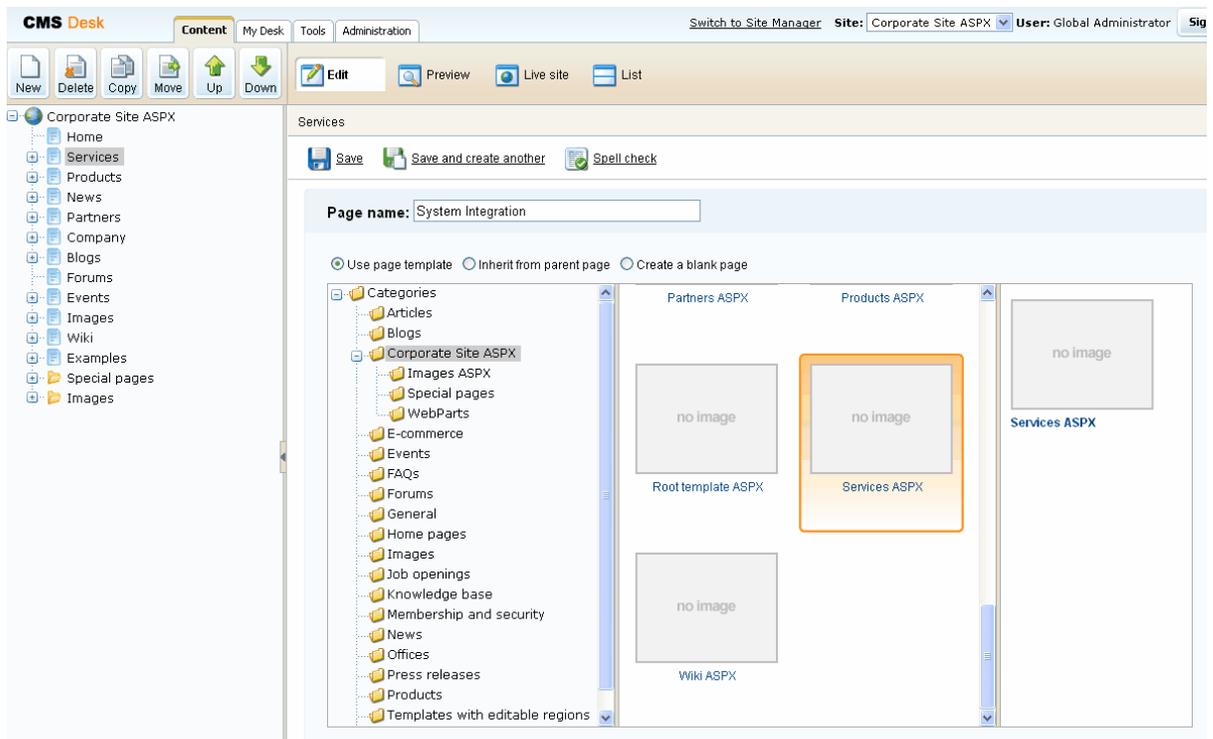
Now we will create a new page under the Services section. Click **Edit** in main toolbar to switch back to the editing mode. Click **Services** in the content tree. Click **New** in the main toolbar. You will see the following dialog that allows you to select the type of the document you want to create under the currently selected document:

### New document

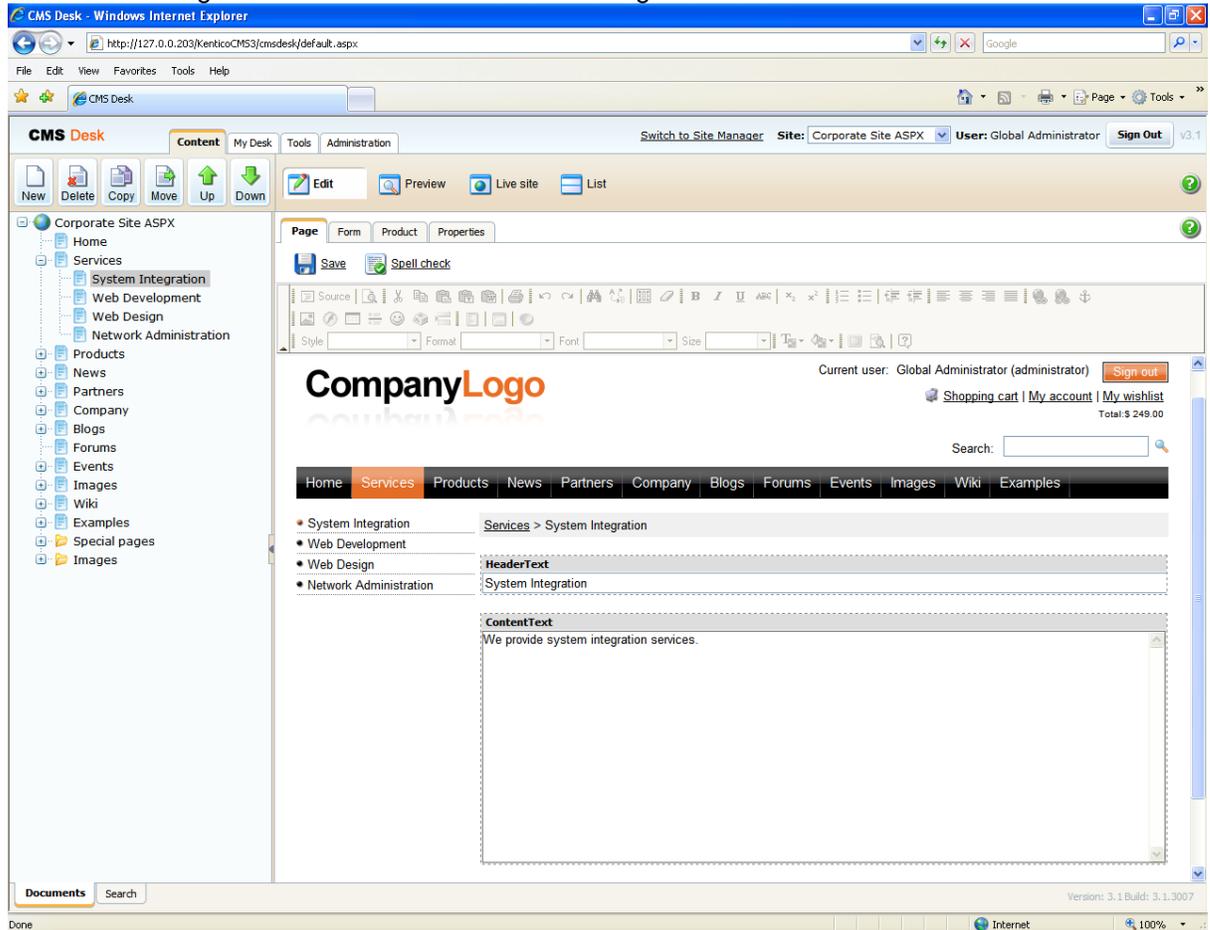
Please select new document type:

-  [Page \(menu item\)](#)
  -  [Article](#)
  -  [Blog](#)
  -  [Cell phone](#)
  -  [Event](#)
  -  [Event \(booking system\)](#)
  -  [FAQ](#)
  -  [File](#)
  -  [Folder](#)
  -  [Image gallery](#)
  -  [Job opening](#)
  -  [Knowledge base article](#)
  -  [Laptop](#)
  -  [News](#)
  -  [Office](#)
  -  [PDA](#)
  -  [Press release](#)
  -  [Product](#)
  -  [Simple article](#)
- 
-  [Link an existing document](#)

Click the **Page (menu item)** button. You will be redirected to the new page properties dialog. Enter *System Integration* in the **Page name** field and choose the **CorporateSite ASPX/Services ASPX** template:



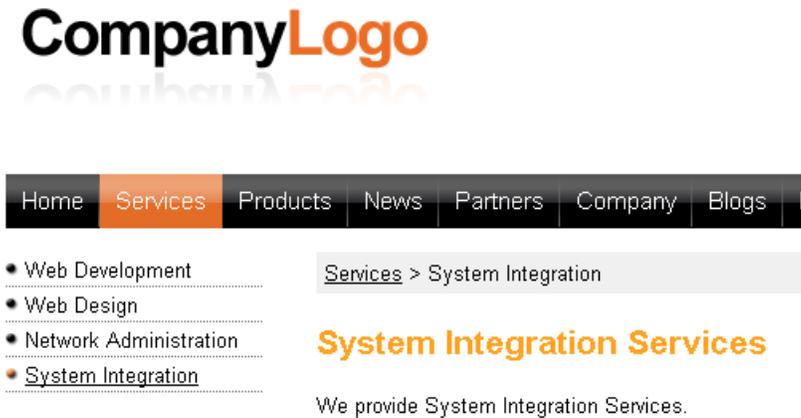
Click **Save** to create the new page. The page is now created in the content tree and you can edit page content on the right. Enter some text in the editable regions and click **Save**.



Now you may want to change the order of the items in the left menu. Click the **Down** button in the main toolbar three times. The *System Integration* item is moved at the bottom of the section:



Click **Live site** in the main toolbar. You will see your new page as it is displayed to site visitors. Please note that the **System Integration** item is placed at the end of the left menu as you specified:

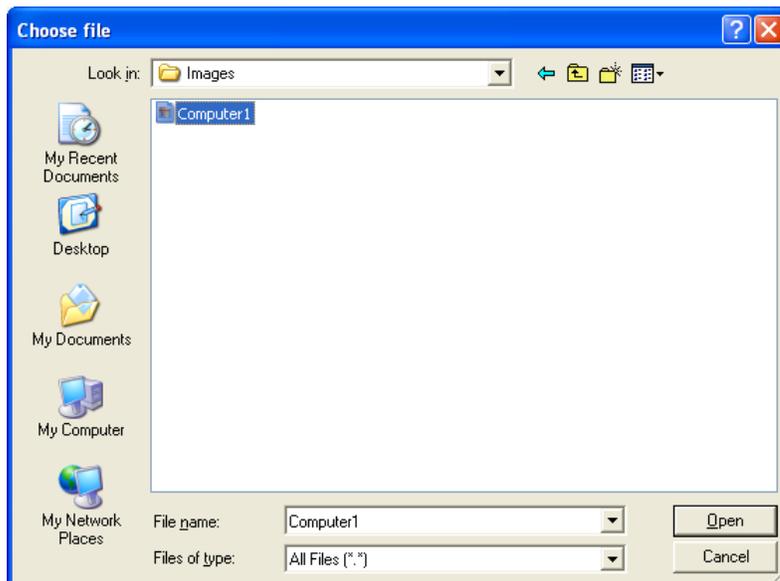


You have learned how to create a new page based on a pre-defined page template.

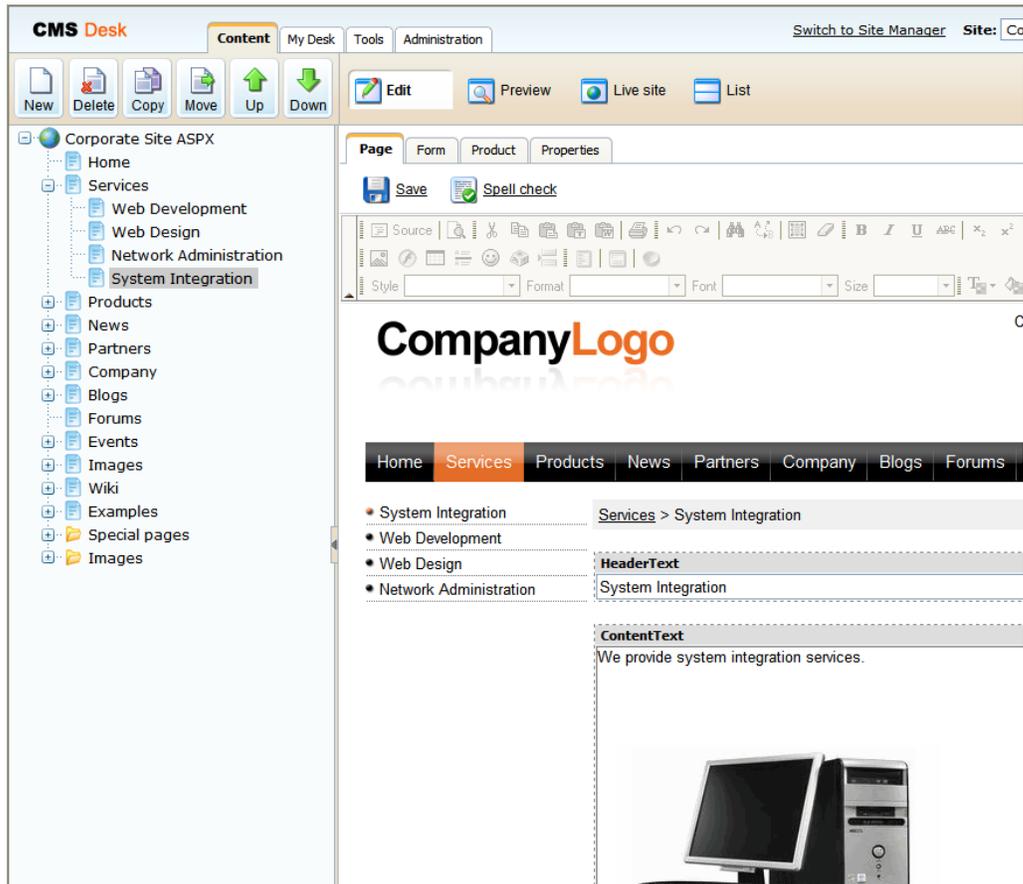
### 3.4 Inserting an image

Now we will upload and insert a new image to our new page. Click **Services -> System Integration** in the content tree. Switch to the **Edit -> Page** mode. Place the cursor into the main editable region, just below the text, and click the **Quickly insert image** (🖼️) in the WYSIWYG editor toolbar.

The browser's Choose file dialog opens. Locate some suitable image file and click **Open**.



The image will be pasted to the editable region so that the page looks like this:



Click **Save** to save the changes. Click **Live site** to see the new version of your page.

You have learned how to upload an image and insert it into the text.



#### Allowing pop-ups for the web site

If you're using a pop-up blocker, you may need to allow pop-up windows in your browser so that the Web part properties dialog as well as some other dialogs work correctly. This applies only to the administration interface, so the site visitors are not affected by this.

### 3.5 Creating a link

Now we will create a link between the **Services** page and our **System Integration** page. Click **Services** in the content tree and make sure you have the **Edit -> Page** mode selected.

Add a new item in the bulleted list called **System integration**:

Services

**HeaderText**  
Our Services

**ContentText**  
We provide professional services in the following areas:

- [Web development](#)
- [Web design](#)
- [Network administration](#)
- System integration

*This is a simple page with left menu and content on the right. You can Insert/Edit Link button in the editor toolbar.*

Select the whole line:

Services

**HeaderText**  
Our Services

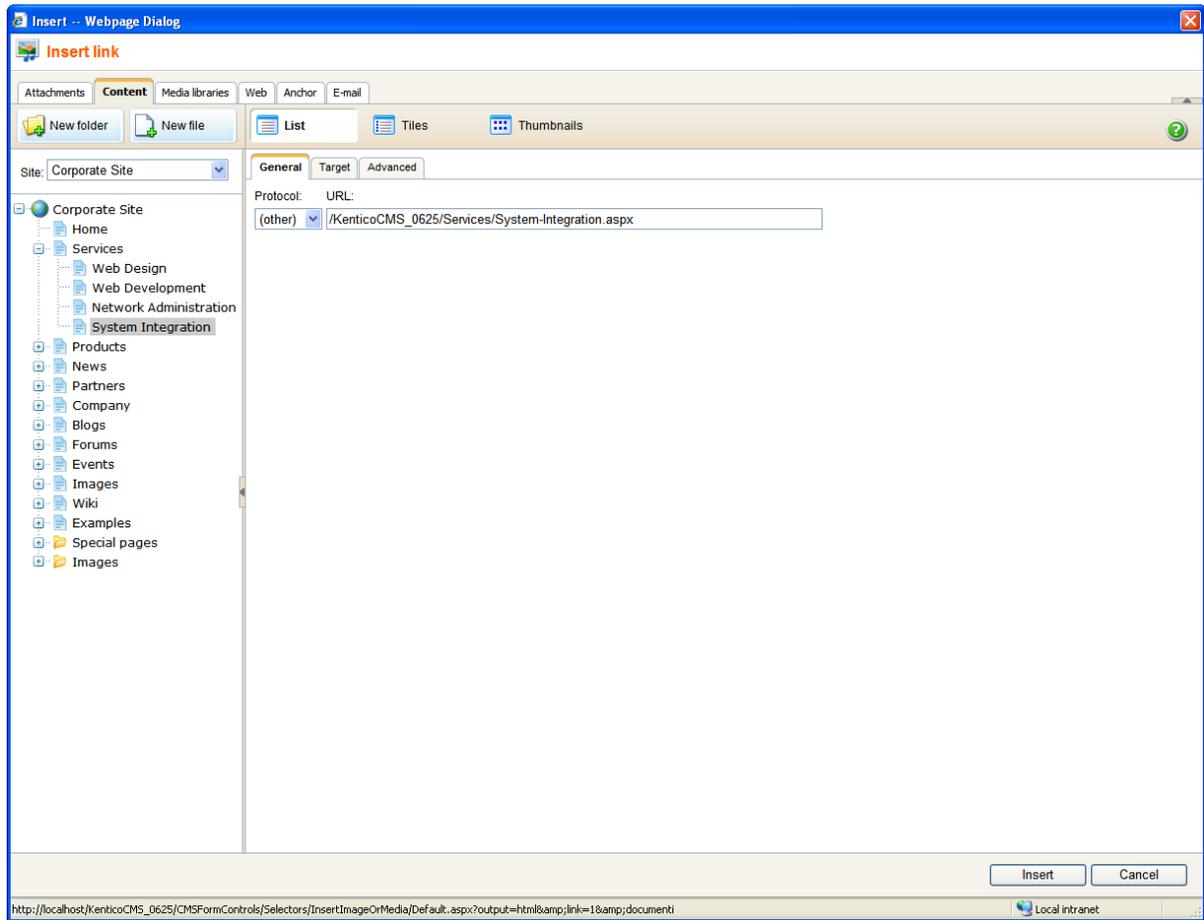
**ContentText**  
We provide professional services in the following areas:

- [Web development](#)
- [Web design](#)
- [Network administration](#)
- [System integration](#)

*This is a simple page with left menu and content on the right. You can Insert/Edit Link button in the editor toolbar.*

And click the **Insert/Edit Link** () button in the WYSIWYG editor toolbar.

The **Insert link** dialog opens.



Switch to the **Content** tab, select the **Services -> System Integration** page from the content tree and click **Insert**.

The text is now marked as a link:



The screenshot shows a web page editor interface. At the top, there is a grey header bar with the text "Services". Below the header is a dashed-line box containing the text "Our Services". Underneath that is another dashed-line box labeled "ContentText". Inside the "ContentText" box, the text reads: "We provide professional services in the following areas:" followed by a bulleted list of four items: "Web development", "Web design", "Network administration", and "System integration". Each item in the list is underlined. Below the list, there is a paragraph of text: "This is a simple page with left menu and content on the right. You ca. Insert/Edit Link button in the editor toolbar." The entire content area is enclosed in a dashed-line border.

Click **Save** and choose the **Live site** mode. When you click the **System integration** link now, you're redirected to the new page.

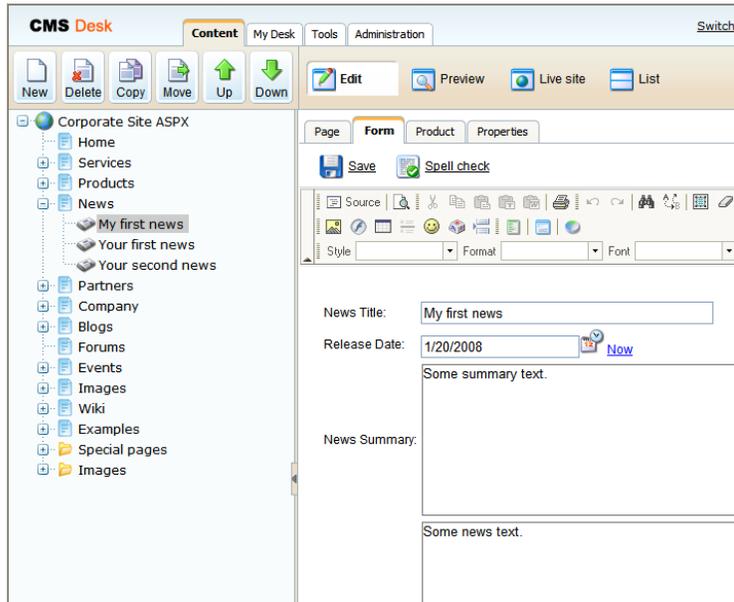
You have learned how to create a link between pages.

### 3.6 Creating a news item

Now you will learn how to create a news item. Click **Edit** in the main toolbar. Click **News** in the content tree and click **New**. Choose to create a new document of type **News**. You are redirected to the form that allows you to define news item sections: title, summary, full text and release date. Enter the following text:

- **News title:** My first news
- **Release date:** *click Now*
- **News summary:** Some summary text.
- **News text:** Some news text.

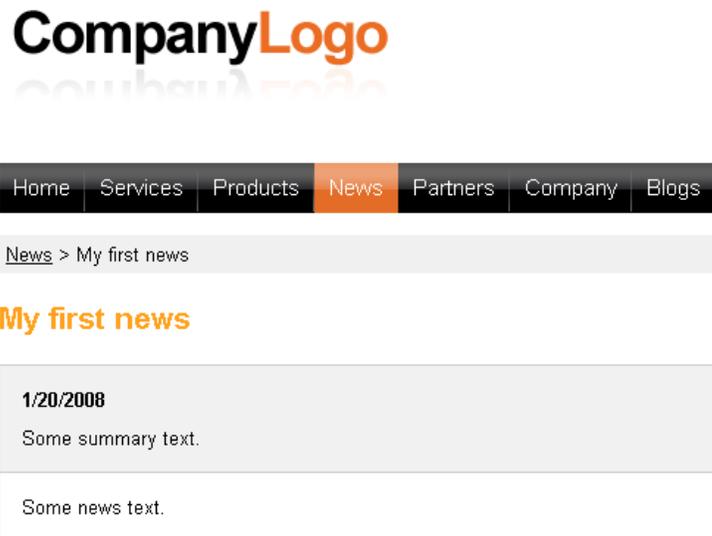
Click **Save** to save the new document.



The screenshot displays the Kentico CMS Desk interface. The top navigation bar includes 'Content', 'My Desk', 'Tools', and 'Administration'. The main toolbar contains 'New', 'Delete', 'Copy', 'Move', 'Up', 'Down', 'Edit', 'Preview', 'Live site', and 'List'. The left sidebar shows a content tree with 'Corporate Site ASPX' expanded, containing 'Home', 'Services', 'Products', 'News', 'Partners', 'Company', 'Blogs', 'Forums', 'Events', 'Images', 'Wiki', 'Examples', 'Special pages', and 'Images'. The 'News' folder is expanded, showing 'My first news', 'Your first news', and 'Your second news'. The main workspace is in 'Form' editing mode, showing a 'News Title' field with 'My first news', a 'Release Date' field with '1/20/2008' and a 'Now' button, a 'News Summary' field with 'Some summary text.', and a 'News Text' field with 'Some news text.'. The 'Form' tab is active, and the 'Save' button is visible.

As you can see, the editing mode is now set to **Form** instead of **Page**. It means you do not edit the editable regions on the page, but you edit the structured data related to the document. The **Form** tab is used for editing the **structured data related to the document**. The document fields are fully customizable for every document type.

When you click **Live site**, you will see the news item displayed using a pre-defined transformation in both News and News -> My first news page:



You have learned how to add a news item and how to use the editing form for structured documents.



### Page versus Form

There are two faces of document: content stored in editable regions on the page and data stored in form fields. The following table compares both approaches:

	Editable regions on the page	Form
<b>Content structure</b>	Simple content structure, only text content.	Complex content structure, typed data, such as text, date-time, numbers, etc.
<b>Validation</b>	Only basic validation rules for minimum and maximum length.	Complex validation rules, including regular expressions and custom form controls with custom validation code.
<b>Display</b>	The content is displayed in the context of the page as it is displayed in the editing mode.	The content is displayed using XSLT or ASCX transformations using special controls or web parts.
<b>Storage</b>	The content is stored in a single XML document in the document properties.	The content is stored in a separate database table. Each field has its own column. The data can be easily modified using SQL queries or API.
<b>Examples of use.</b>	Home page, contact page.  Generally: pages with simply structured or unstructured text-based content.  The editable regions are usually used for documents of type <b>Page (menu item)</b> .	News, product specification, event details, job opening, etc.  Generally: pages with structured content where you need to separate content from design and keep the content in its original data type.  The form-based content is usually used for documents of type <b>News, Product, Article</b> , etc.

**Part**

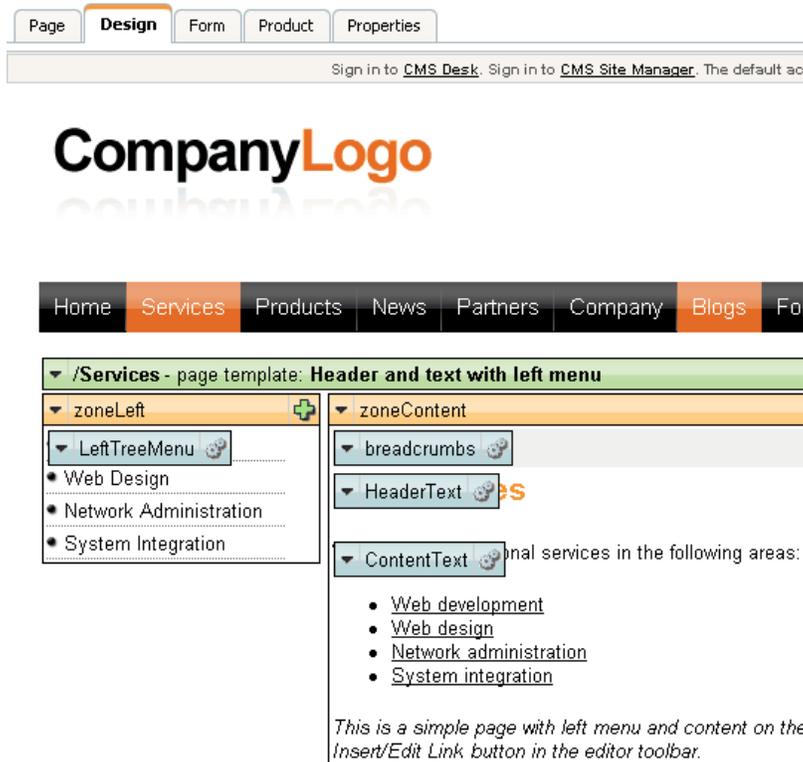


## 4 Site Development Overview

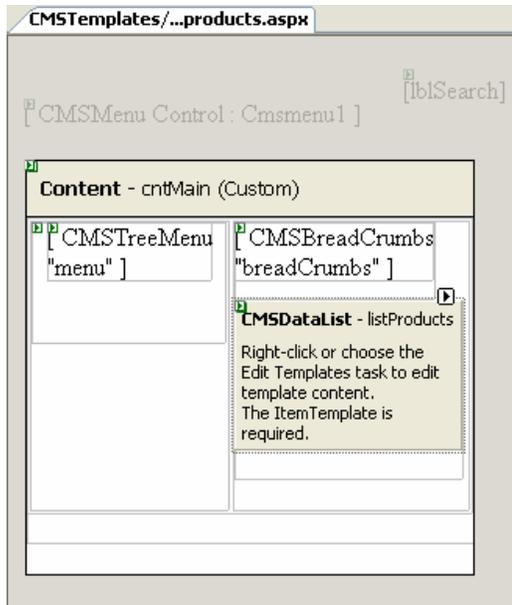
### 4.1 Site Development Overview

Kentico CMS provides two development models and you can choose which one suits you better:

- **Portal Engine** - this model allows you to build web site using a portal engine. It's the recommended way for most developers since it doesn't require programming and using Visual Studio. You can easily build web site using web parts in the **browser-based** user interface.



- **ASPX Templates** - this model can be chosen by advanced ASP.NET developers who prefer to create the web site using standard ASP.NET architecture and using standard development tools, such as **Visual Studio 2005**. You need to be familiar with ASP.NET development and have at least basic programming knowledge of C# or VB.NET.



Both approaches are fully supported and they provide the same level of flexibility and extensibility. We recommend that you use the portal engine model, but if you're a hard-core .NET developer and do not trust portal engines, you may want to use ASPX templates.

Both models can be combined in a single web site and you can e.g. enhance the portal engine web site with ASPX templates or even with your own ASPX pages and integrate your own applications.

The following table compares both portal engine and ASPX templates:

	Portal Engine	ASPX Template
<b>How you work</b>	<p>You build web site using the browser-based interface.</p> <p>No programming knowledge is required for common tasks.</p>	<p>You build ASPX page templates that are used to display content from Kentico CMS.</p> <p>At least basic programming knowledge of ASP.NET and either C# or VB.NET is required.</p>
<b>How you assemble pages</b>	<p>You use built-in or custom web parts that you place into customizable page layouts.</p>	<p>You use built-in or custom ASP.NET server controls that are placed on the ASPX pages. These are standard ASPX pages and they are part of the web site project that you can open in Visual Studio 2005.</p> <p>You can also place web parts (which are actually standard ASCX user controls) on the page templates if the functionality is not available as a server control.</p>
<b>Master pages and visual inheritance</b>	<p>Sub-pages inherit the content from the parent pages by default (so called "visual inheritance"). The inheritance can be optionally broken if you want to create a page without parent content.</p>	<p>All page templates (.ASPX pages) may use a master page, which is a standard ASP.NET 2.0 master page (.master page).</p> <p>The pages do not inherit content from their parents, they only inherit content from the master page (if it's used).</p>
<b>Custom code integration and extensibility</b>	<p>You can create your own user controls and web parts if you need to integrate a specific functionality.</p> <p>You can add any custom controls and code to the web parts or user controls that you use on your web site.</p> <p>You can also use standard ASPX pages within your portal engine-based web site.</p>	<p>You build standard ASPX pages with code-behind which means you can use any custom controls and code on the page in Visual Studio.</p>
<b>Advantages</b>	<ul style="list-style-type: none"> <li>• Easier and faster way of building a web site.</li> <li>• ASP.NET programming knowledge is not required for common tasks.</li> <li>• You can build the whole web site very quickly, only using the web browser.</li> </ul>	<ul style="list-style-type: none"> <li>• Standard ASP.NET architecture.</li> <li>• You can use your favorite development tools, such as Visual Studio 2005 for all changes.</li> </ul>
<b>Disadvantages</b>	<ul style="list-style-type: none"> <li>• Proprietary architecture and development.</li> </ul>	<ul style="list-style-type: none"> <li>• Requires ASP.NET programming knowledge.</li> </ul>



### Is Kentico CMS just another portal engine?

Now you may ask what's the difference between Kentico CMS and DotNetNuke or SharePoint.

Well, the main difference is the **flexibility**. Kentico CMS gives you a full control over:

- site structure
- site navigation
- page layout
- design
- content structure

Also, it's important to explain that Kentico CMS is a **content management system**, not only a portal engine. It provides features of advanced CMS systems, such as:

- content repository with a logical tree hierarchy of documents
- content/design separation
- custom document types with custom fields
- workflow and versioning
- content locking (check-out, check-in)
- multilingual content
- content preview and content staging
- document-level permissions with permission inheritance
- full-text search in all content
- document management features for uploaded files

Moreover, Kentico CMS comes with many **professional and flexible built-in modules out-of-the-box**, including Newsletters, On-line forms, Forums, E-commerce, Content Staging, Image gallery, Event calendar, Booking system, Blogs, Polls and others

It means you do not need to purchase third-party modules with inconsistent user and programming interface, but you get everything from a single source, with a complete documentation.

The rest of this tutorial explains the ASPX templates approach. If you want to use the ASPX templates, please read the Tutorial for ASPX page templates.

**Part**



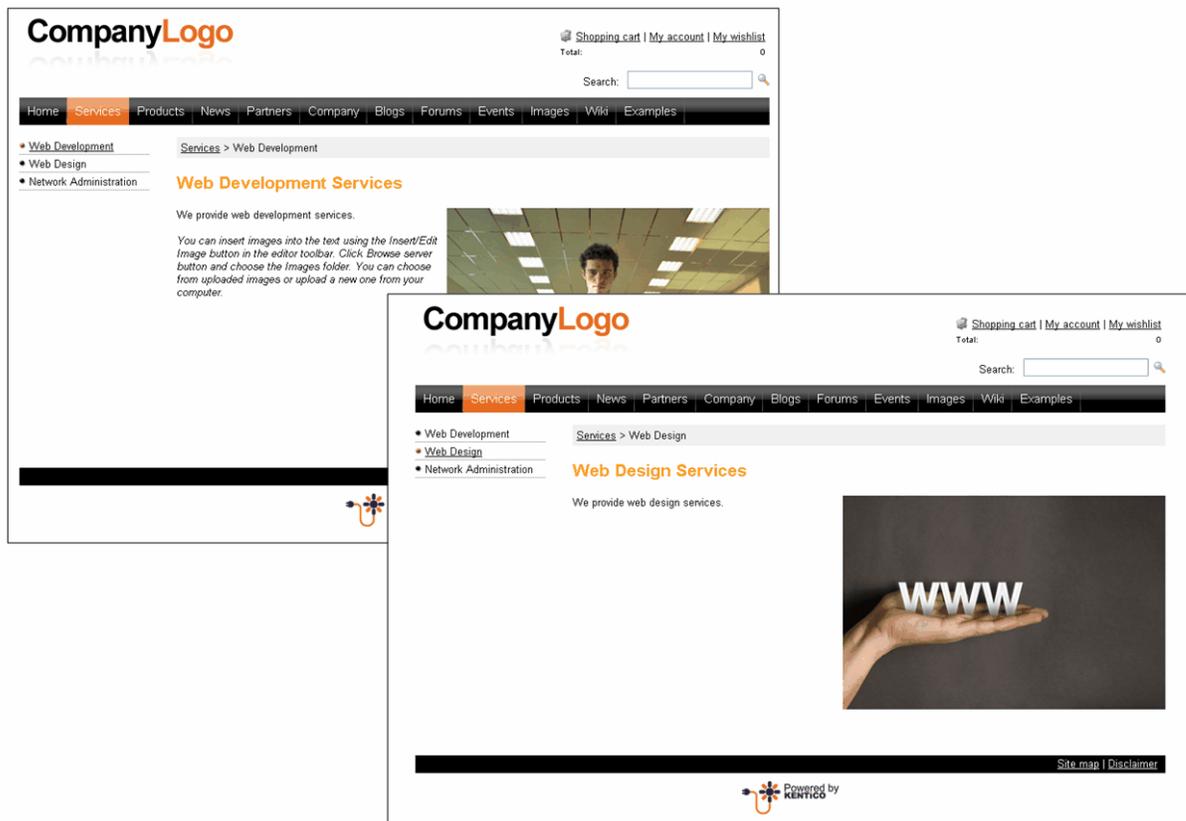
## 5 Creating pages using ASPX templates

### 5.1 ASPX page templates

If you're familiar with ASP.NET development in Visual Studio 2005, you may choose to develop web sites using standard ASPX page templates. ASPX page templates in Kentico CMS are standard ASP.NET pages that display content from Kentico CMS. They receive the **aliasPath** URL parameter that tells the page template which page should be displayed.

#### What is a page template?

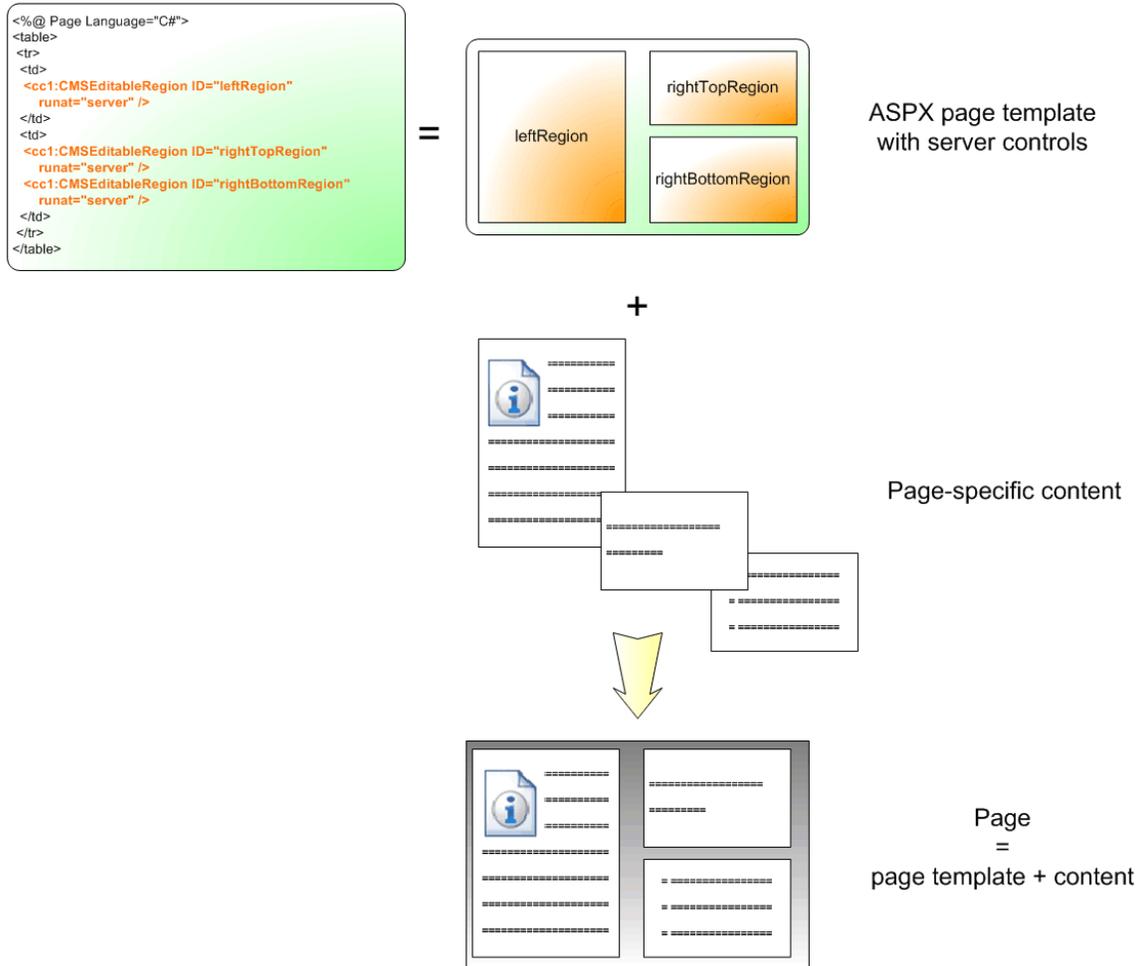
Every web page is based on some page template. The page template can be specific for a single web page ("ad hoc" page template) or it can be re-used for several pages. The following picture shows an example of two pages that use the same page template:



As you can see both of them use the same header, main menu, sub-menu, content structure and footer - they are based on the same **page template**. In this way, you can create multiple pages using the same design.

## What does the ASPX page template consist of?

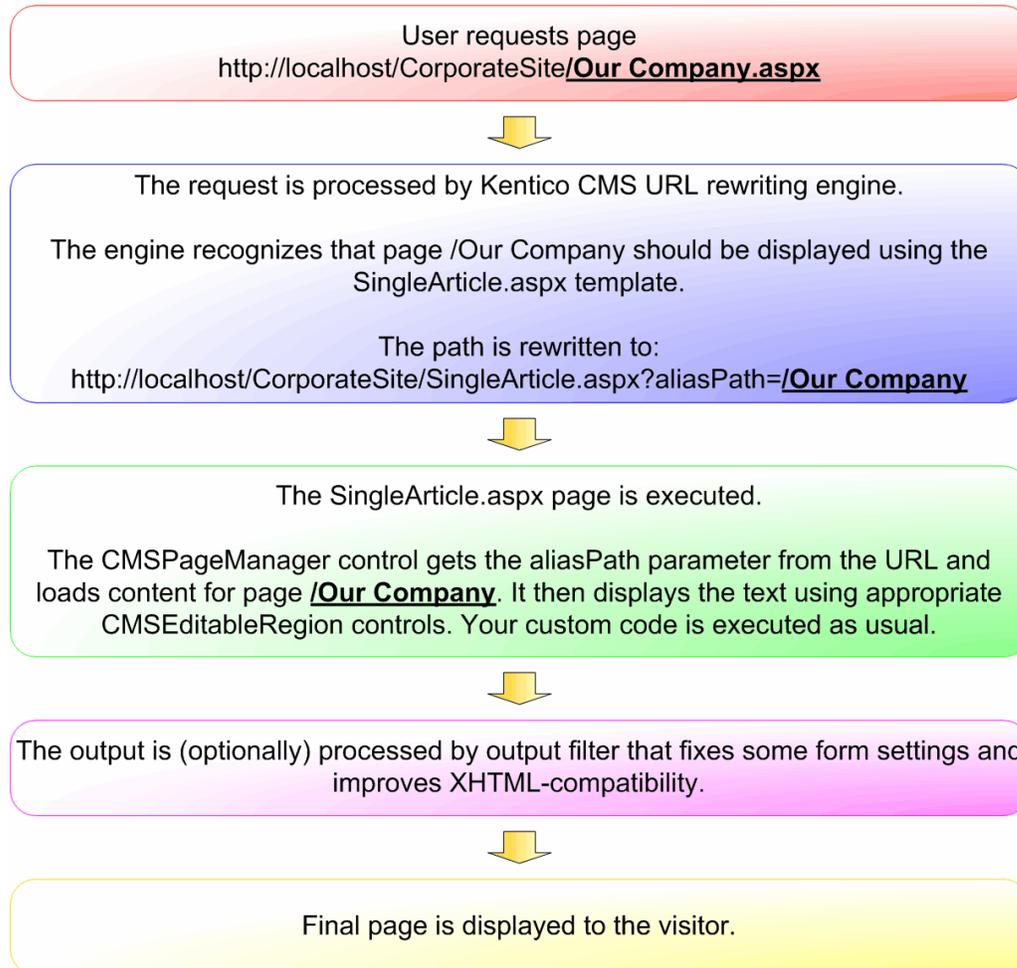
The page template is a combination of static HTML code and ASP.NET server controls (or user controls) that render dynamic content. The following figure illustrates how ASPX page template and page content are combined to display



As you can see, the ASPX page template is a standard page that may contain HTML code, CMS server controls and any other server control. You can also use code behind (in both VB.NET and C#) to modify page behavior and add custom functionality.

## How is the ASPX page template processed?

When a user requests some page, such as `/services/web-development.aspx`, the system calls the assigned page template with the **aliasPath** parameter that specifies what content (which page) should be displayed using the given template:



The built-in Kentico CMS controls understand the `aliasPath` parameter in the URL and render the appropriate content automatically.

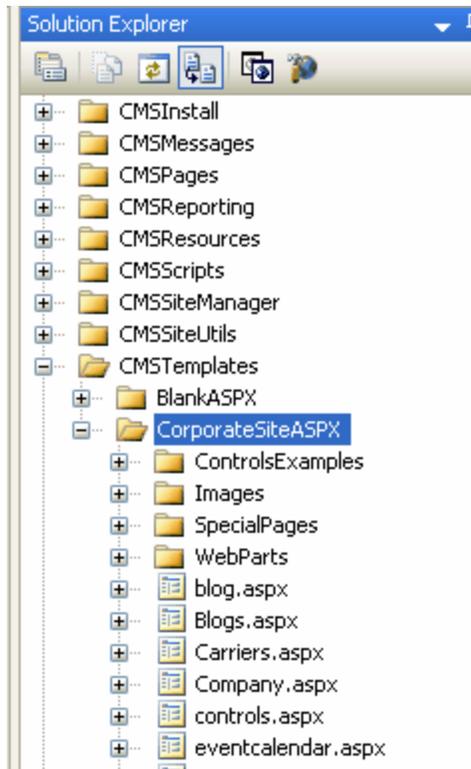
As you can see, the system uses a standard ASP.NET architecture. If you developed the web site without Kentico CMS, you would most likely use URLs like this: `/news.aspx?newsid=127` which is similar to `/news.aspx?aliaspath=/news/november news.aspx` URL used in Kentico CMS.

## 5.2 Creating a simple ASPX page template

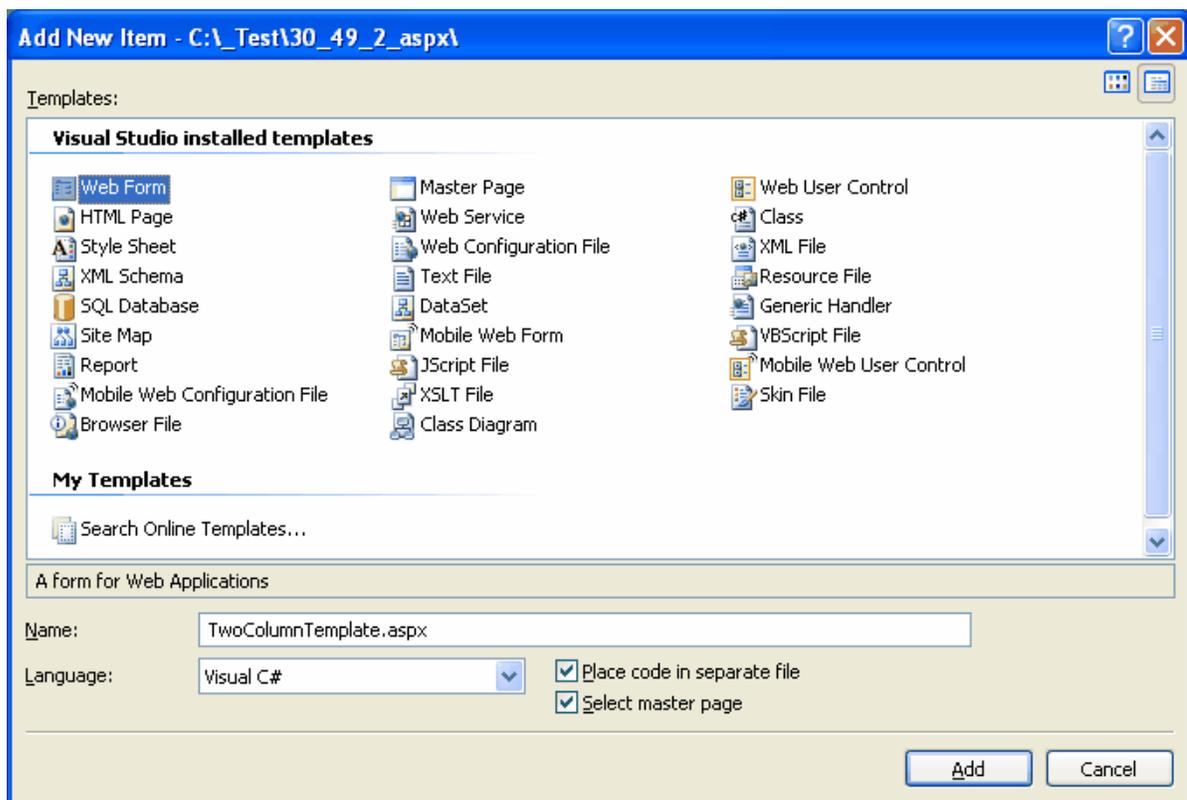
Now you will learn how to create a new ASPX page template. We will create a new **Contact** page with two columns that will contain editable regions.

Open the web project in Visual Studio 2005. You can open it either using the WebProject.sln file or using the **File -> Open -> Web Site** menu.

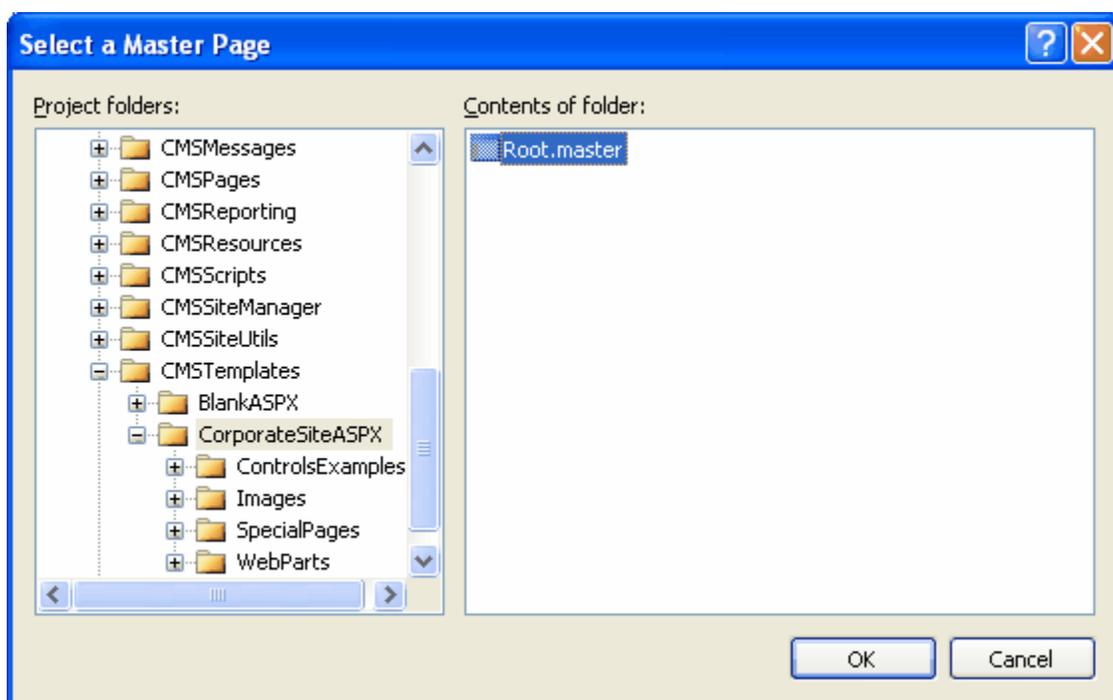
Now right-click the **CMSTemplates -> CorporateSiteASPX** folder in the Solution Explorer and choose **Add new item**:



Choose to create a new **web form** and call it **TwoColumnTemplate.aspx**, check the box **Select master page** and click **Add**.



Choose the master page **CMSTemplates/CorporateSiteASPX/root.master** and click **OK**.



## Writing the ASPX code

Add the following code above the `<asp:Content>` element. It allows you to use Kentico CMS controls in your page:

```
<%@ Register Assembly="CMS.Controls" Namespace="CMS.Controls" TagPrefix="cc1" %>
```

Add the following code inside the `<asp:Content>` element:

```
<table width="100%">
  <tr>
    <td width="50%">
      <cc1:CMSEditableRegion ID="txtLeft" runat="server" DialogHeight="400"
        RegionType="HtmlEditor" RegionTitle="Left column" />
    </td>
    <td width="50%">
      <cc1:CMSEditableRegion ID="txtText" runat="server" DialogHeight="400"
        RegionType="HtmlEditor" RegionTitle="Right column" />
    </td>
  </tr>
</table>
```

The `<asp:Content>` control specifies that this content will be loaded into the master page (that is defined in the `MainMenu.master` file). As you can see, you can use the standard concept of master pages that is available in ASP.NET 2.0.

The `<cc1:CMSEditableRegion>` control defines an editable region that will be displayed as an HTML editor in the editing mode. On the live site, it ensures displaying of the page content.

**Please note:** this example uses a table layout. If you prefer CSS layout, you can simply replace the surrounding HTML code with `<DIV>` elements. As you can see, you have full control over the content.

Now we need to modify the code behind. Switch to code behind and add the following namespace so that the `CMSEditableRegion` can be recognized:

```
using CMS.UIControls;
```

The last step is to modify the class from which our page is inherited. Change the following code:

```
public partial class CMSTemplates_CorporateSiteASPX_TwoColumnTemplate :
  System.Web.UI.Page
```

to the following:

```
public partial class CMSTemplates_CorporateSiteASPX_TwoColumnTemplate :
  TemplatePage
```

so that the page can be used as a page template in Kentico CMS.

## Registering the ASPX page as a page template

Now that we have created a new ASPX page, we need to register it in Kentico CMS as a page template, so that it can be used by content editors.

Sign in to **Site Manager** and go to **Development -> Page templates**. Click the **Corporate Site ASPX** folder and click **New template**. Enter the following values:

- **Template display name:** Two column template
- **Template code name:** TwoColumnTemplate

Click **OK**. Now enter the following value in the **File name** field:

```
~/CMSTemplates/CorporateSiteASPX/twocolumntemplate.aspx
```

It is the virtual path of our ASPX page.

The screenshot shows the Kentico CMS Site Manager interface. The top navigation bar includes 'CMS Site Manager', 'Sites', 'Administration', 'Settings', 'Development' (selected), 'Licenses', and 'Support'. The main content area is titled 'Page Templates' and contains a tree view on the left and a 'Page template properties' panel on the right. The tree view shows a hierarchy starting with 'Page Templates', then 'Corporate Site ASPX', and finally 'Two column template' (highlighted). The 'Page template properties' panel has tabs for 'General', 'Sites', 'Header', and 'Documents'. The 'General' tab is active and shows a 'Save' button and several input fields: 'Template display name' (Two column template), 'Template code name' (TwoColumnTemplate), 'Category' (Corporate Site ASPX), 'Template description' (empty text area), 'Thumbnail' (Upload: [ ] and Browse...), 'Template type' (Portal page and ASPX page, with ASPX page selected), and 'File name' (~/CMSTemplates/CorporateSiteASPX/tv).

Save the changes.

Now click the **Sites** tab and assign the page template to the new web site. Click **OK**.

### Page template properties

Templates ▶ Two column template

General **Sites** Header Documents

The changes were saved.

The page template is available for the following web sites:

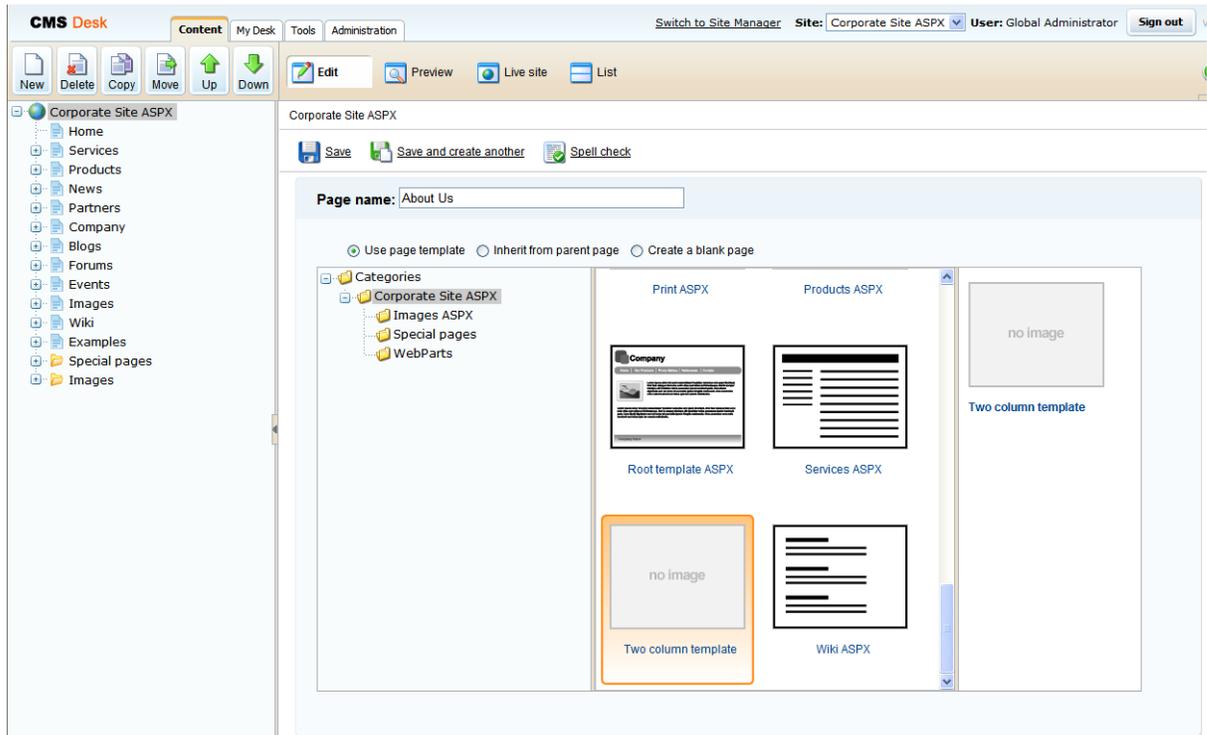
Select Site name

Corporate Site ASPX

OK

## Creating an About Us page based on the new page template

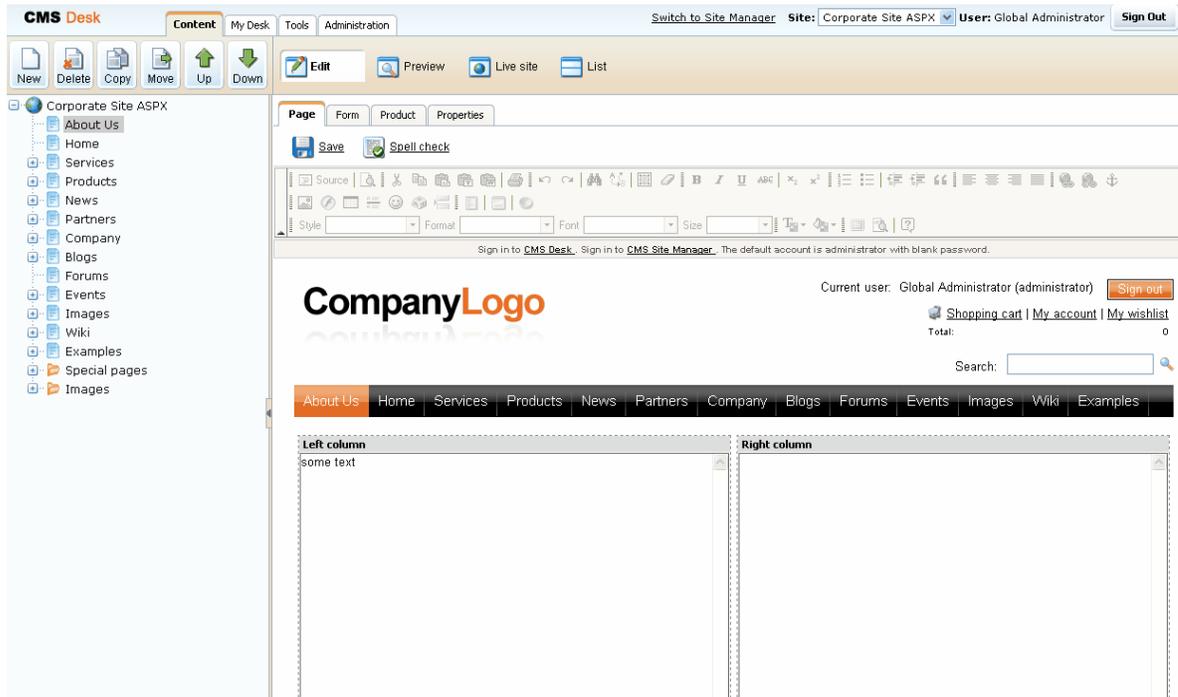
Go to **CMS Desk** -> **Content**. Click the root of the content tree (Corporate Site ASPX) and click **New** in the main menu. Choose to create a new **Page (menu item)**. Enter the page name **About Us** and choose to create a page using the page template **Corporate Site ASPX/Two column template**:



The screenshot shows the CMS Desk interface with the 'Content' tab selected. The 'New' button is clicked, opening a dialog for creating a new page. The 'Page name' field contains 'About Us'. The 'Use page template' radio button is selected. The 'Two column template' is highlighted in the template selection area. The 'Save' button is visible at the bottom left of the dialog.

Click **Save** to create the new page.

Click **Page** and you will see a page with editable regions like this:



Congratulations, you have just created your first page template based on ASPX page. Now you can enter some text and click **Save** to save the changes.



#### Please note

If you want to move the About Us page in another position in the menu, you can use the Up and Down arrows in the main toolbar to re-order the menu items.

## 5.3 Using the master pages

Kentico CMS allows you to use standard ASP.NET 2.0 master pages together with ASPX page templates. This is a very powerful concept, that allows you to share the same site header and footer with logo, main menu, search box, etc. over all pages without having to create these sections on each page template again and again.

The master pages are defined in files with extension **.master**. You can assign a single master page to each ASPX page. The master page must always contain the **ContentPlaceHolder** control like this:

```
<asp:ContentPlaceHolder ID="plcMain" runat="server">
```

The **ContentPlaceHolder** control specifies where the content of page templates should be loaded. So the master page typically contains the main logo and navigation and the content is displayed by ASPX pages loaded into the master page.

The following code sample defines a very simple master page:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Root.master.cs"
Inherits="CMSTemplates_CorporateSiteASPX_Root" %>
<%@ Register Assembly="CMS.PortalControls" Namespace="CMS.PortalControls"
TagPrefix="cc2" %>
<%@ Register Assembly="CMS.Controls" Namespace="CMS.Controls" TagPrefix="cc1" %>
<%=DocType%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <asp:literal runat="server" id="ltlTags" enableviewstate="false" />
</head>
<body class="<%=BodyClass%>" <%=BodyParameters%>>

    <form id="form1" runat="server">
        <cc1:CMSPageManager ID="CMSPageManager1" runat="server" />
        <cc1:CMSMenu ID="cmsmenu1" runat="server" CSSPrefix=";Sub"
Cursor="Pointer"
        HighlightAllItemsInPath="true"
        Layout="Horizontal"
        Padding="0"
        Spacing="1" />
        <asp:ContentPlaceHolder ID="plcMain" runat="server">
            </asp:ContentPlaceHolder>
    </form>
</body>

</html>
```

The **CMSPageManager** control ensures loading of the content from the database into the editable regions. The **CMSMenu** control displays a drop-down menu. The **ContentPlaceHolder** control defines where the content of sub-pages should be loaded.

In case that you are planning to use AJAX components on your site, you need to add the **ScriptManager** control after the **CMSPageManager** control.

```
<asp:ScriptManager ID="manScript" runat="server" />
```

We use **CMSMenu** and **CMSPageManager** controls on the page template, so we need to add the **CMS.UIControls** namespace in code behind:

#### [C#]

```
using CMS.UIControls;
```

#### [VB.NET]

```
Imports CMS.UIControls
```

The master page must be inherited from the `TemplateMasterPage`, so the class definition must look like this:

**[C#]**

```
public partial class CMSTemplates_CorporateSiteASPX_Root : TemplateMasterPage
```

**[VB.NET]**

```
Partial Class CMSTemplates_CorporateSiteASPX_Root  
    Inherits TemplateMasterPage
```

And you also need to put the following code to the master page code-behind class:

**[C#]**

```
protected override void CreateChildControls()  
{  
    base.CreateChildControls();  
  
    this.PageManager = this.CMSPageManager1;  
}  
  
protected override void OnPreRender(EventArgs e)  
{  
    base.OnPreRender(e);  
  
    this.ltlTags.Text = this.HeaderTags;  
}
```

**[VB.NET]**

```
Protected Overloads Overrides Sub CreateChildControls()  
    MyBase.CreateChildControls()  
  
    Me.PageManager = Me.CMSPageManager1  
End Sub  
  
Protected Overloads Overrides Sub OnPreRender(ByVal e As EventArgs)  
    MyBase.OnPreRender(e)  
  
    Me.ltlTags.Text = Me.HeaderTags  
End Sub
```

You should store master pages in the **CMSTemplates** folder together with page templates, so that they are exported with your web site.

**Part**



## 6 Managing styles and design

### 6.1 CSS styles

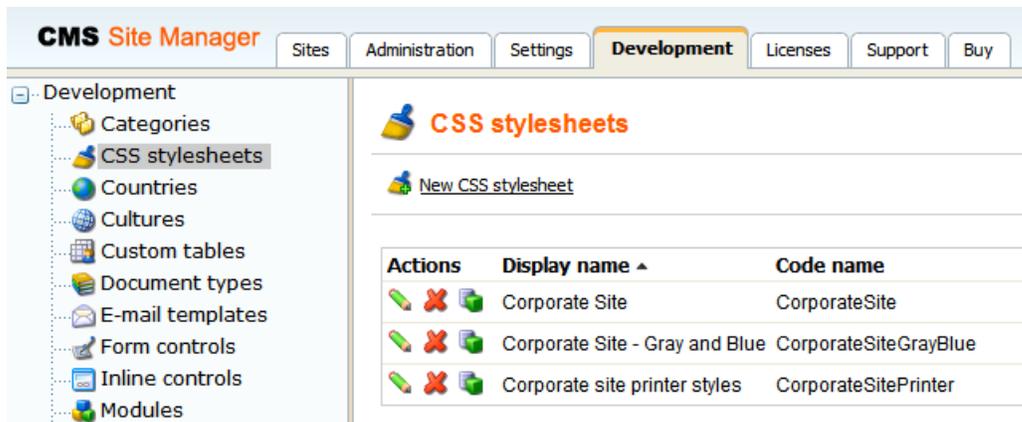
The design of the web site relies on standard CSS styles. Each web site has its global CSS stylesheet that can be chosen in **Site Manager -> Sites -> ... edit site ... -> General**. Here you can also choose a different CSS stylesheet used by WYSIWYG editors.

Besides, each page can overwrite the global CSS stylesheet by setting its own stylesheet in **CMS Desk -> Content -> ... edit some document ... -> Properties -> General**.

The CSS stylesheets can be managed in Site Manager. When you're in CMS Desk, you can easily switch to Site Manager by clicking the **Switch to Site Manager** link in the header:

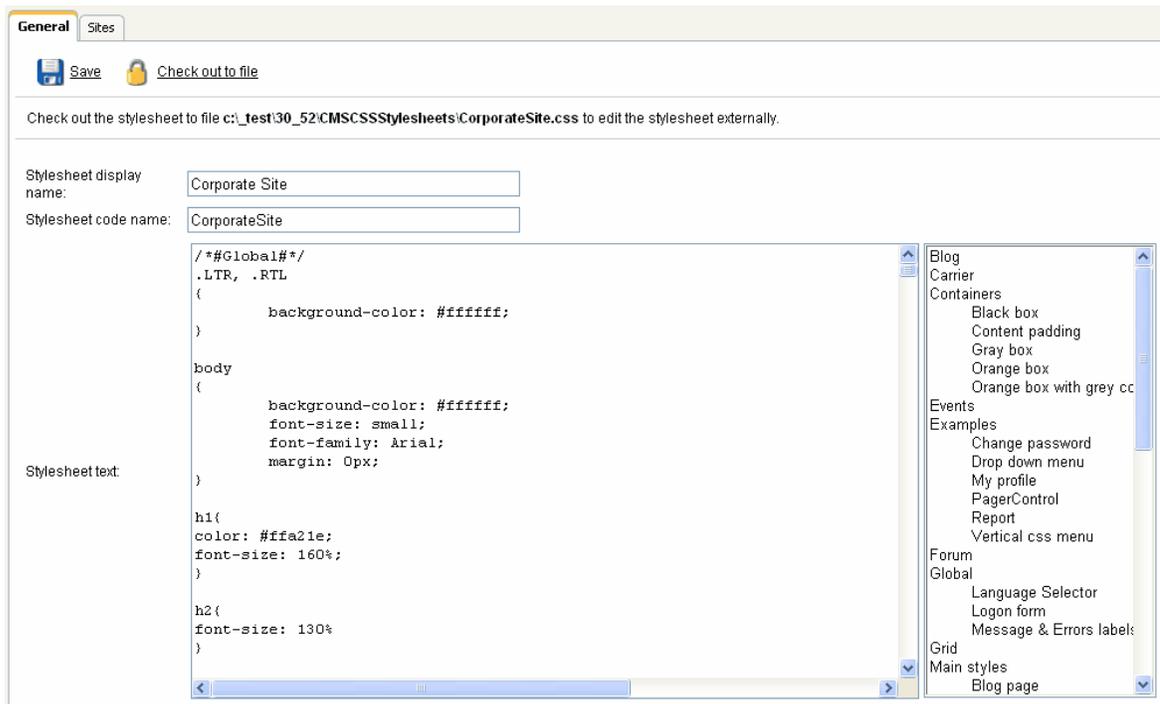


Then click **Development** in the **Site Manager** main menu and click **CSS stylesheets** in the left menu:



Actions	Display name ^	Code name
  	Corporate Site	CorporateSite
  	Corporate Site - Gray and Blue	CorporateSiteGrayBlue
  	Corporate site printer styles	CorporateSitePrinter

Edit  the Corporate Site stylesheet:



General Sites

Save Checkout to file

Checkout the stylesheet to file c:\\_test\30\_52\CMS\Stylesheets\CorporateSite.css to edit the stylesheet externally.

Stylesheet display name: Corporate Site

Stylesheet code name: CorporateSite

Stylesheet text:

```

/*#Global#*/
.LTR, .RTL
{
    background-color: #ffffff;
}

body
{
    background-color: #ffffff;
    font-size: small;
    font-family: Arial;
    margin: 0px;
}

h1{
    color: #ffa21e;
    font-size: 160%;
}

h2{
    font-size: 130%
}

```

Blog  
Carrier  
Containers  
  Black box  
  Content padding  
  Gray box  
  Orange box  
  Orange box with grey cc  
Events  
Examples  
  Change password  
  Drop down menu  
  My profile  
  PagerControl  
  Report  
  Vertical css menu  
Forum  
Global  
  Language Selector  
  Logon form  
  Message & Errors labels  
Grid  
Main styles  
  Blog page

### Example:

Change the background color of the .LTR, .RTL style to **#fef3b8**, click **OK** to save the changes. Switch back to **CMS Desk** and see the live site. The site background will be yellow.

**Please note:** you may need to right-click the page and choose Refresh (in Internet Explorer) in order to see the latest version of the style.



### .LTR, .RTL and body CSS classes

The .LTR and .RTL CSS classes are assigned to the BODY element for the left-to-right and right-to-left text direction respectively. You can use them to set the styles of the BODY element used on the live site without setting the BODY element in the WYSIWYG editor. It's useful if you need to set a dark background to the web site, but use a white background in the WYSIWYG editor text.

The body CSS class is applied to BODY element in both web site and WYSIWYG editor.



### Browser-dependant styles

The BODY element CSS class consists of the combination of LTR/RTL style (see the box above) and the browser version. It looks like this:

```
<body class="LTR IE7" >
```

This feature allows you to create styles that depend on the browser version. So if you define the styles like this:

```
.IE6 .MyClass  
{  
  height: 20px ;  
}  
.Opera .MyClass  
{  
  height: 18px ;  
}
```

the font size will be different for IE6 and Opera.

The class names are:

- IE: IE<major version>
- Opera: Opera
- Firefox: Gecko
- Safari: Safari

## 6.2 App themes

In some cases, you may leverage the built-in support for ASP.NET themes. You can use them for setting styles of controls that do not have their own CSS class name, such as Datagrid, Calendar or web parts with complex dialogs (logon form, registration form, ...).

The name of the theme folder under **App\_Themes** must be same as the code name of the site's CSS stylesheet. So if you use the **Green** stylesheet on your site, your theme must be stored in the **App\_Themes\green** sub-folder under your web project.

You need to add your skins to the default.skin file. Here's an example of CMSCalendar / Calendar web part skin:

```
<cms:CMSCalendar SkinID="EventCalendar" Runat="server">  
  <NextPrevStyle ForeColor="Red"></NextPrevStyle>  
  <WeekendDayStyle BackColor="#E0E0E0"></WeekendDayStyle>  
</cms:CMSCalendar>
```

The code above defines the look of the Event calendar control. You can see this control on the Events page in the sample Corporate Site.



### Where should I store web site design files?

It's recommended that you store all images or Flash movies that are part of the web site design template in folder `app_themes/<stylesheet code name>`. This ensures that the files are exported together with web site when your deploying it to some other server.

## 6.3 Menu design

Now you will learn how to change the design of the main menu. The main menu used in the sample Corporate Site is displayed using the **Drop-down menu** web part which is based on the **CMSMenu** server control.

The menu design depends primarily on the CSS styles. Here's an example of the CSS styles for the drop-down menu:

```
.horizontalmenu{
    background: #000 url(../App_Themes/CorporateSite/Images/
topMenuBackground.gif);
}

.horizontalCMSMenu
{
    height: 34px;
    color: #fff;
    padding: 3px;
    background: #000 url(../App_Themes/CorporateSite/Images/
topMenuBackground.gif);
}

.horizontalsubCMSMenu
{
    background: black;
    color: #fff;
}

.horizontalsubCMSMenu tr
{
    background: black;
}

.horizontalsubCMSMenuItem
{
    white-space: no-wrap;
    display: block;
}

...
```

As you can see these are standard CSS styles. You can modify the styles in the global CSS stylesheet of the given site.

The default menu looks like this:



Now we will change the background color of selected menu items to orange. Go to **Site Manager -> Development -> CSS stylesheets** and edit the **Corporate Site** stylesheet. Choose **Top menu styles -> Horizontal** in the right navigation panel. Change the highlighted line:

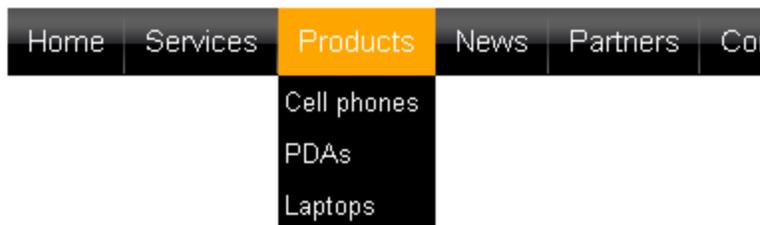
```
.horizontalCMSMenuItemMouseOver, .horizontalCMSMenuHighlightedMenuItem,  
.horizontalCMSMenuHighlightedMenuItemMouseOver,  
.horizontalCMSMenuHighlightedMenuItemMouseDown  
{  
    background: orange;  
}
```

Click **OK** to save changes. When go to the live site now, you will see a menu like this:



### Defining different styles for different menu levels

When you mouse-over the menu you will see that the sub-menus are displayed in different colors:



The sub-menu styles can be configured using the **CSS prefix** property of the **Drop-down menu** web part or the **CSSPrefix** property of the **CMSMenu** control.

In the sample Corporate Site project, the property is set to "**horizontal;horizontalsub**". It means that the CSS styles for the first level of the menu use styles like **horizontalCMSMenuItem**, while the sub-menus on the second and further levels are displayed using the CSS styles with prefix **horizontalsub**, such as **horizontalsubCMSMenuItem**. In this way, you can define different CSS styles for any level of the menu structure.

## Defining the style of a single menu item

Every document may have its own style that is used when the document is displayed in the menu. We will try to modify the style of the **Home** menu item. Go to **CMS Desk -> Content** and click **Home**. Click **Properties -> Menu**. Here you can define:

- **Menu caption** - the name of the document when it's displayed in the menu.
- **Show in navigation** - indicates if the document should be displayed in the navigation controls
- **Show in site map** - indicates if the document should be displayed in the site map
- **Menu item design** for standard, mouse-overed and highlighted menu item.

Enter the following value into the **Menu item style** value (under the Menu item design section): `background-color: red;` and click **Save**. Click **Live site**. Click **Services**. The **Home** menu item is now displayed in red:



**Part**



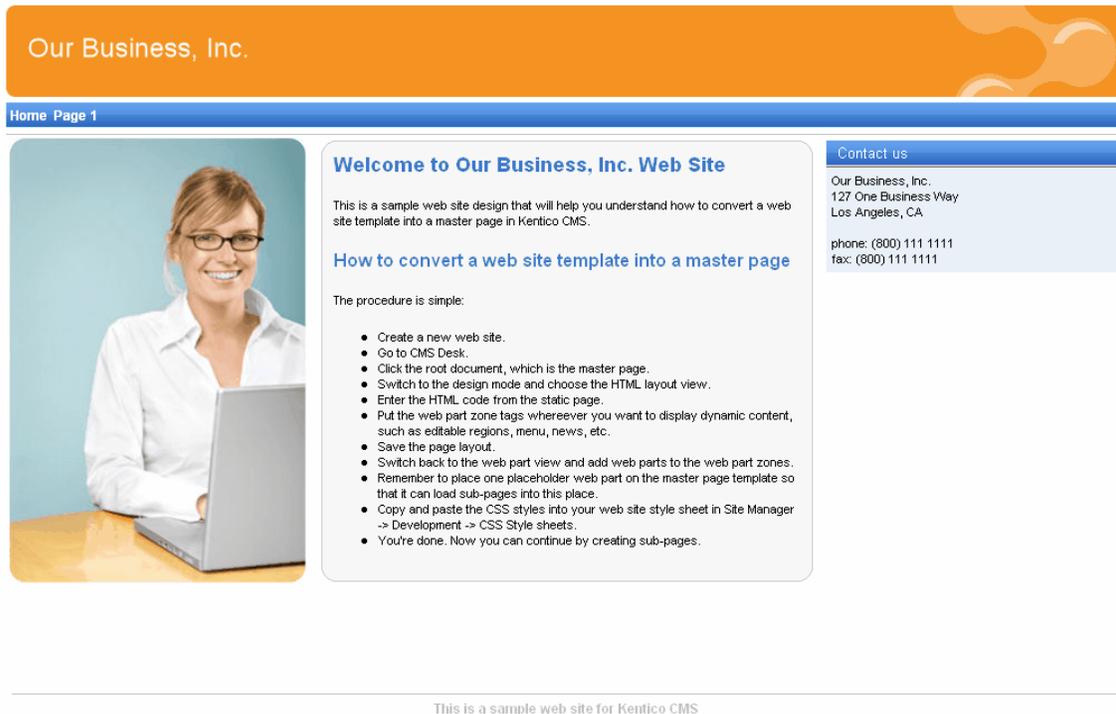
**VII**

## 7 Creating a new site using ASPX templates

### 7.1 Overview

This tutorial will guide you through the creation of a simple web site using ASPX page templates developed in Visual Studio 2005. You will learn how to define site structure, design, how to create your own pages and page templates.

During this tutorial, we will use a static web site template that is similar to what a developer gets from graphic designer. It looks like this:



You can find the static page template in folder `C:\Program Files\Kentico CMS\. The template consists of the home.htm file, styles folder and app_themes folder with images.`

### 7.2 Creating a new web site using wizard

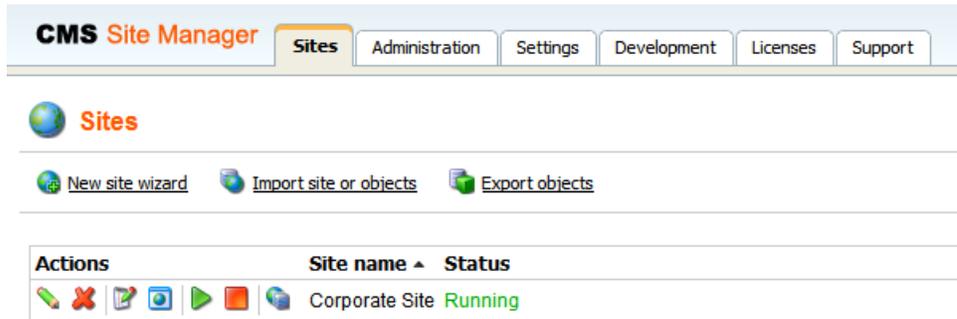
The following chapters suppose you have previously installed a sample CorporateSite on your computer. We will leave the existing web site and add a new web site that will run on `http://127.0.0.1`.



#### Multiple sites and VS2005 built-in web server

If you're using the **built-in web server in VS2005 instead of IIS**, you need to **stop the CorporateSite** site in the Site Manager -> Sites dialog first and then you can continue. Since the built-in web server doesn't support other domain than localhost, you will use the **localhost domain** again.

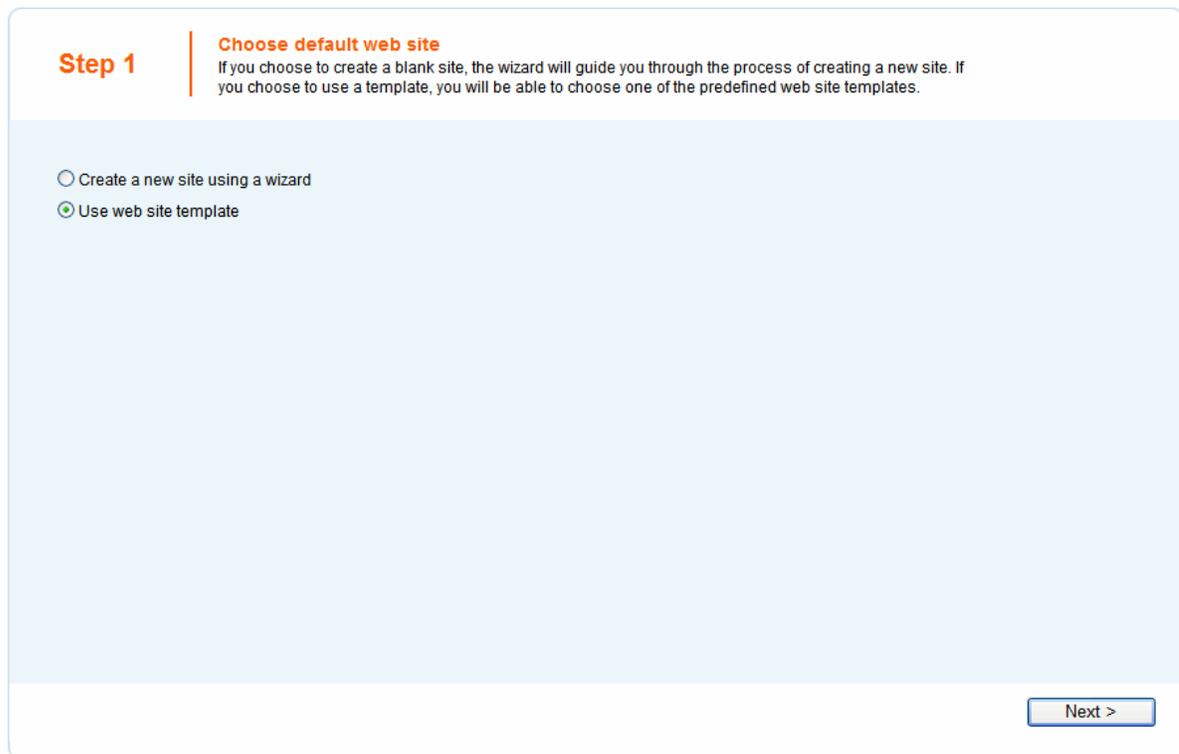
Sign in as Administrator to **Site Manager -> Sites**. Click **New site wizard** on the home page.



The screenshot shows the CMS Site Manager interface. At the top, there is a navigation bar with the following tabs: **Sites** (selected), Administration, Settings, Development, Licenses, and Support. Below the navigation bar, the **Sites** section is displayed, featuring a globe icon and the text "Sites". Underneath, there are three links: [New site wizard](#), [Import site or objects](#), and [Export objects](#). Below these links is a table with the following structure:

Actions	Site name ▲	Status
  	Corporate Site	Running

In the first step, choose to **create a new site using a wizard**. Click **Next**.



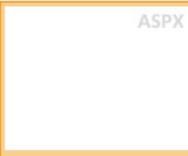
The screenshot shows the "Step 1" of the wizard, titled "Choose default web site". The instructions state: "If you choose to create a blank site, the wizard will guide you through the process of creating a new site. If you choose to use a template, you will be able to choose one of the predefined web site templates." There are two radio button options:

- Create a new site using a wizard
- Use web site template

A "Next >" button is located at the bottom right of the screen.

In the second step, choose the **Blank site ASPX** web site template: Click **Next**.

**Step 2** | **Choose web site template**  
Choose the predefined web site template that will be used for your new web site. The web site template may contain site structure, design, basic content, new document types and other settings.

	<b>Community Site</b> This is a web site template for a sample community site. Social networking features of Kentico CMS are used on the site to give you an idea of how they can be used on your web site. You can also modify this web site and use it as a base for the development of your own site. It uses the portal engine and is recommended for developers who are new to Kentico CMS.
	<b>Blank Site</b> This template is intended for developers who want to create a new web site from scratch. It uses the portal engine.
	<b>Blank Site ASPX</b> This template is intended for developers who want to create a new web site from scratch. It uses the ASPX page templates that require a higher level of .NET knowledge than the portal engine.

< Previous      Next >

In the third step, enter the following details:

- Site display name: **My web site**
- Site code name: **mysite**
- Domain: **127.0.0.1** - if you're using VS2005 built-in web server, set the default value to **localhost**

Click **Next**.

**Step 3**

**Enter new site settings**

Enter the display name and code name of the web site. The Domain field must contain the domain that you will use to access the web site during development (you may change it when the site goes live). The default culture is the main language of the web site.

Site display name:

Site code name:

Domain name:

[< Previous](#) [Next >](#)

In the fourth step, you're asked to select objects which should be imported to your new site. Do not change anything and click **Next**.

### Step 4

#### Objects selection

Please select objects which should be imported.

**All objects**

- Web site**
  - Documents
  - Tools
  - Administration
  - Settings
  - Development
- Global objects**
  - Tools
  - Administration
  - Development

#### Import objects

**Please note:** The import process may overwrite your existing objects. The existing objects are marked with \* and will be overwritten if checked.

Please select the object type from the tree if you wish to change the default selection. Click **Next** to start the import of selected objects.

**Global selection**

[Load default selection](#)   [Select all objects](#)   [Select only new objects](#)   [Deselect all objects](#)

**Import settings**

- Assign all objects to the imported site (recommended)
- Run the site after import
- Delete incomplete site when import fails
- Import files (recommended)
- Do not import objects where parent object is missing
- Import tasks (recommended)

< Previous   Next >

In the fifth step, the progress of objects import is displayed. Click **Next** after **Import has successfully finished** appears.

**Step 5** | **Import progress**  
Objects are being imported.

**Import has successfully finished**  
Starting site 'My web site'  
Copying objects files  
Importing 'Newsletters' objects  
Importing 'E-mail templates' objects  
Importing additional document properties  
Importing ACL items  
Importing ACLs  
Importing document/ (en-US)  
Importing 'Tag groups' objects  
Importing 'Scheduled tasks' objects  
Importing 'Role' objects  
Importing 'Web site' objects  
Importing 'Page templates' objects  
Importing 'Page template categories' objects  
Importing 'Settings' objects  
Importing 'Settings categories' objects

Import in progress

< Previous    Cancel    Next >

You will see the confirmation message.

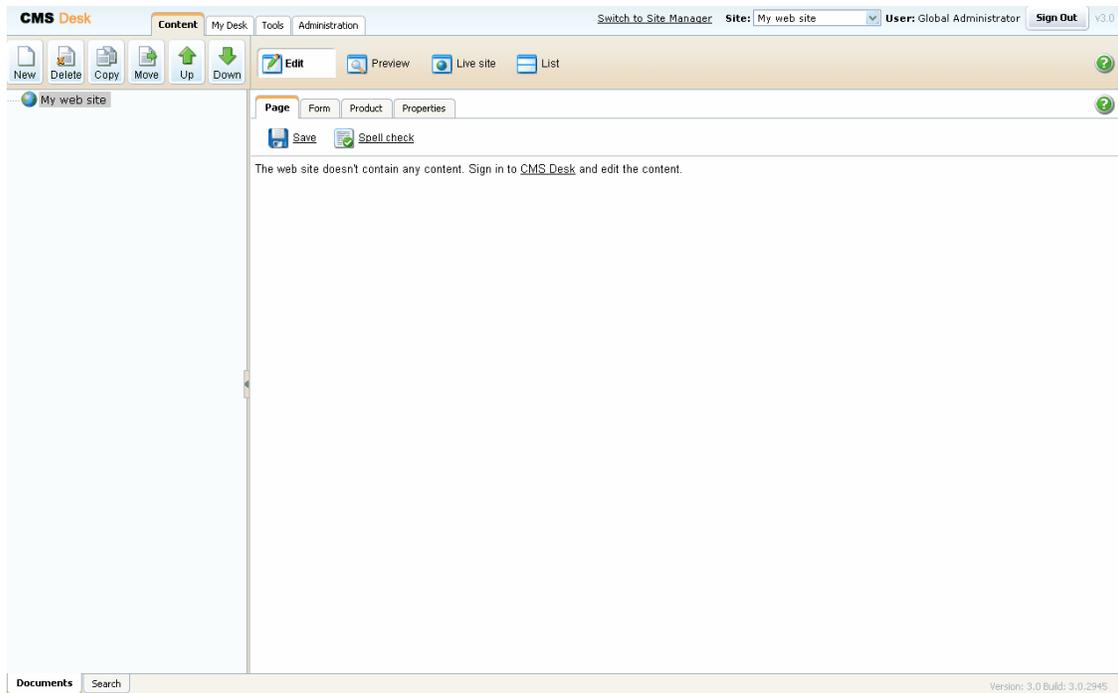
**Step 7** | **The web site has been created successfully**  
Now you can view and edit your new web site.

[Edit your new web site](#)

If this is your first web site, the default user account is administrator without password.

Finish

Click the **Edit your new web site** link. A new window with Kentico CMS Desk opens at domain 127.0.0.1. You need to sign in again since the authentication is not shared over different domains. After you sign in, you will see your new, empty web site:



The screenshot displays the Kentico CMS Desk interface. At the top, there is a navigation bar with tabs for 'Content', 'My Desk', 'Tools', and 'Administration'. The 'Content' tab is active. Below the navigation bar, there is a toolbar with icons for 'New', 'Delete', 'Copy', 'Move', 'Up', and 'Down'. The main content area shows a confirmation message: 'The web site has been created successfully. Now you can view and edit your new web site.' Below this message is a link 'Edit your new web site' and a note: 'If this is your first web site, the default user account is administrator without password.' At the bottom right of the main content area is a 'Finish' button. The interface also shows a 'Page' tab with sub-tabs for 'Form', 'Product', and 'Properties'. The 'Page' tab is active, and there are 'Save' and 'Spell check' buttons. The status bar at the bottom indicates 'Documents' and 'Search' on the left, and 'Version: 3.0 Build: 3.0.2945' on the right.

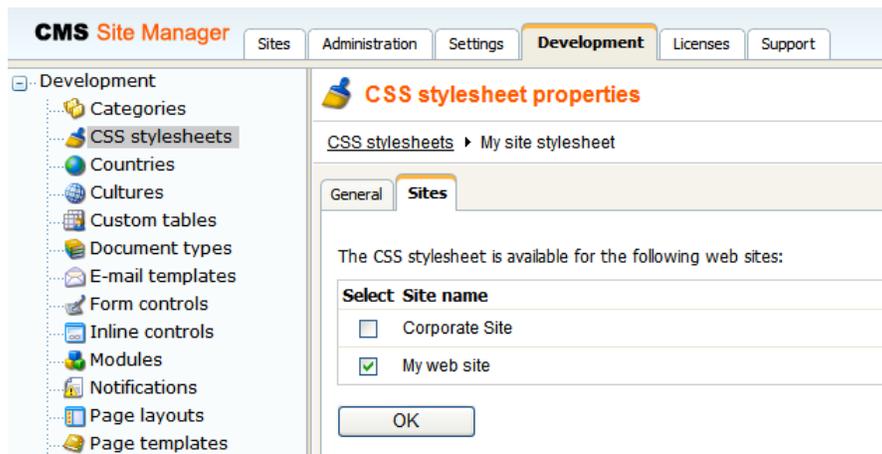
You have created the base for your new web site. In the next chapters, you will learn how to implement the required design.

## 7.3 Creating the CSS stylesheet

Before we start editing our new web site, we will prepare the CSS styles and images based on our web site template. Go to Site Manager -> Development -> CSS Stylesheets and click **New CSS stylesheet**. Enter the following values:

- Stylesheet display name: **My site stylesheet**
- Stylesheet code name: **MySite**
- Stylesheet text: copy and paste all text from the **SampleWebTemplate\Styles\main.css** file (you will find it in folder **C:\Program Files\Kentico CMS\<version>\CodeSamples**)

Click **OK**. Click the **Sites** tab and check the **My web site** box in the list and click **OK**. It assigns the stylesheet to your web site.



Go to **Site Manager** -> **Sites** and edit properties of **My web site**. Select **My site stylesheet** in the Site CSS stylesheet drop-down list and click **OK**. It ensures that the stylesheet is used on all pages of your new web site.

The screenshot shows the 'CMS Site Manager' interface with the 'Sites' tab selected. The 'Site Properties' dialog is open for the site 'mysite'. The 'General' tab is active, showing the following fields:

- Site display name: My web site
- Site code name: mysite
- Site domain name: 127.0.0.1
- Default content culture: English - United States
- Default visitor culture: (Automatic)
- Site CSS stylesheet: My site stylesheet
- Editor CSS stylesheet: (site stylesheet)
- Site description: (empty text area)

An 'OK' button is located at the bottom of the dialog.

Now copy the folder **SampleWebTemplate\app\_themes\MySite** to **<web project>\app\_themes**. It will ensure that the images are exported as a part of the web site if you decide to move the web site in future. Please note that the folder under `app_themes` must have the same name as the code name of the CSS stylesheet - **MySite**.



### CSS stylesheet URL and relative paths

We have adjusted the image paths in the CSS stylesheet so that they match the target folders in your new web site. In real life, you will need to adjust the paths manually. **The URLs of images in the CSS stylesheets are always relative to the CSS stylesheet.**

The URL of the CSS stylesheet is

```
<web project>/CMSPages/GetCSS.aspx?stylesheetname=MySite
```

which means, you need to link to files in the `app_themes` folder as

```
../app_themes/mysite/images/imagename.gif.
```

## 7.4 Opening and configuring the web project

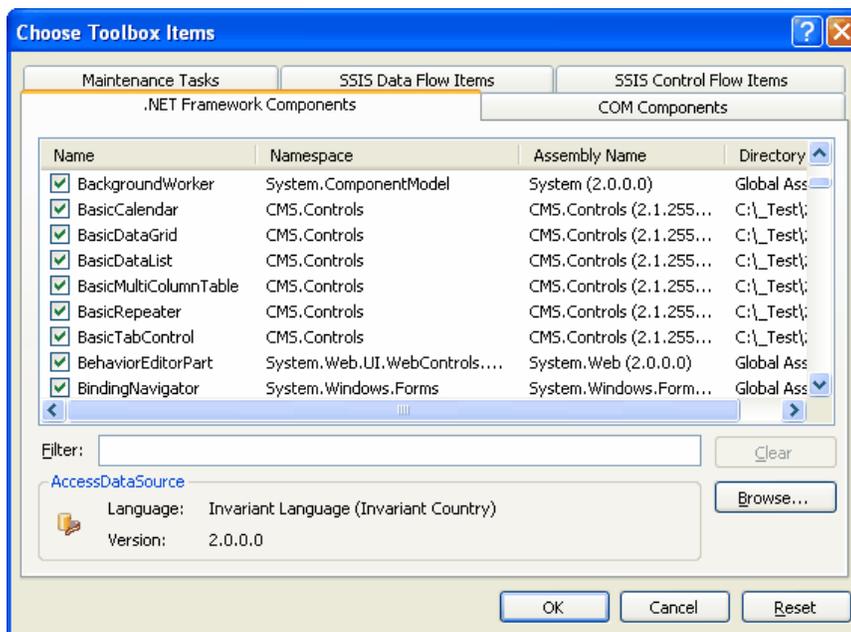
Open the web project in Visual Studio 2005. You can open it either using the **WebProject.sln** file or using the **File -> Open -> Web Site** menu.

Now we need to add Kentico CMS Controls to the Visual Studio Toolbox.

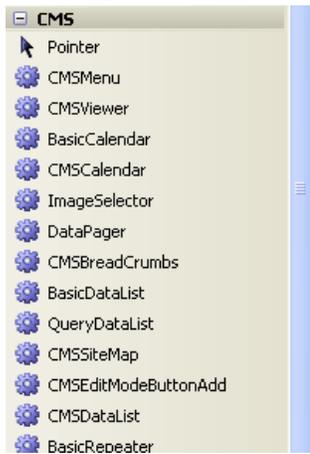
1. Open the web site project in Visual Studio and open some ASPX page.
2. Right-click the **Toolbox** and choose **Add tab** from the context menu.
3. Type the name of the new tab (e.g. CMS) and press Enter:



4. Right-click the new tab and choose **Choose items...** from the context menu.
5. In the **Choose Toolbox Items** dialog, click Browse and locate the **CMS.Controls.DLL** library in the **bin** folder under your web site. Click **Open** and then click **OK**.



## 6. The controls are now added to the Toolbox:

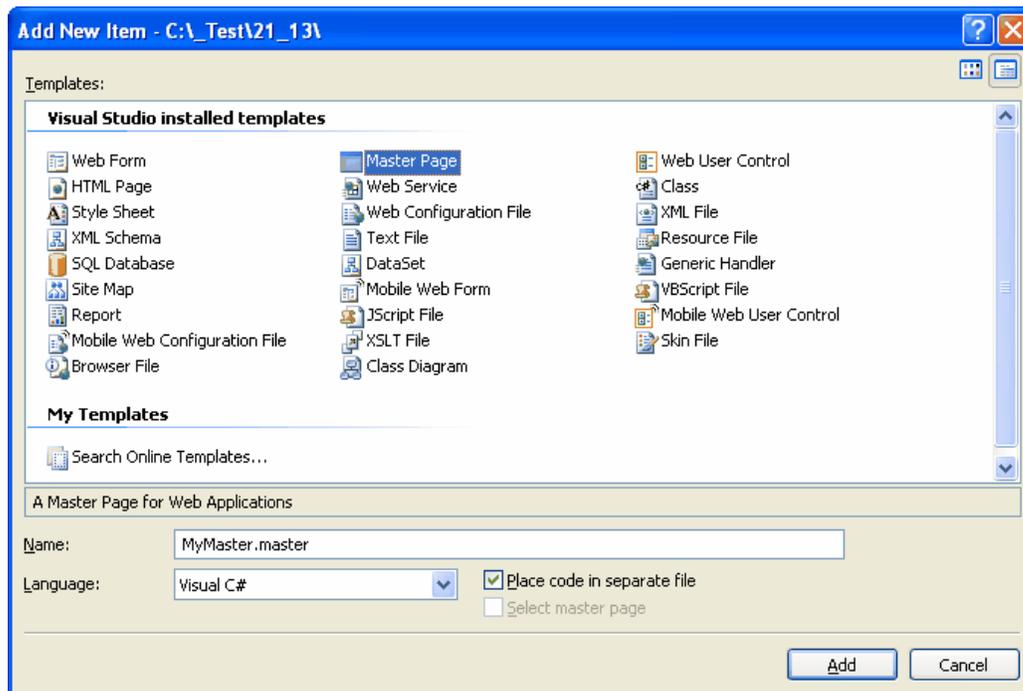


7. Now you can easily drag and drop the controls on your Web forms.

## 7.5 Master page

Open the web project in Visual Studio and right-click the **CMSTemplates** folder in the Solution Explorer window and create a new sub-folder **MySite**. Please note that the folder name must be the same as the code name of your site.

Right-click the **MySite** folder and choose **Add new item...** Choose to create a new master page and set its name to **MyMaster.master**. If you're a VB developer, you may want to choose Visual Basic in the **Language** drop-down list.



Replace all default ASPX code from the master page (in the Source view) except of the first line with `<%@ Master %>` directive with the following code:

```
<%@ Register Assembly="CMS.PortalControls" Namespace="CMS.PortalControls"
TagPrefix="cc2" %>
<%@ Register Assembly="CMS.Controls" Namespace="CMS.Controls" TagPrefix="ccl" %>
<%=DocType%>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <asp:literal runat="server" id="ltrlTags" enableviewstate="false" />
</head>
<body class="<%=BodyClass%>" <%=BodyParameters%>>
    <form id="form1" runat="server">
        <ccl:CMSPageManager ID="CMSPageManager1" runat="server" />
    </form>
</body>
</html>
```

The **CMSPageManager** control ensures loading of content from the database into the page.

In case that you are planning to use AJAX components on your site, you need to add the **ScriptManager** control after the CMSPageManager control.

```
<asp:ScriptManager ID="manScript" runat="server" />
```

Switch to code behind and add the reference to the CMS.UIControls namespace:

**[C#]**

```
using CMS.UIControls;
```

**[VB.NET]**

```
Imports CMS.UIControls
```

Change the class definition so that the master page inherits from TemplateMasterPage:

**[C#]**

```
public partial class CMSTemplates_MySite_MyMaster : TemplateMasterPage
```

**[VB.NET]**

```
Partial Class CMSTemplates_MySite_MyMaster
    Inherits TemplateMasterPage
```

Add the following code after the Page\_Load method and call both the CreateChildControls and the OnPreRender methods from the Page\_Load method:

**[C#]**

```
protected override void CreateChildControls()
{
    base.CreateChildControls();

    this.PageManager = this.CMSPageManager1;
}

protected override void OnPreRender(EventArgs e)
{
    base.OnPreRender(e);

    this.ltlTags.Text = this.HeaderTags;
}
```

**[VB.NET]**

```
Protected Overloads Overrides Sub CreateChildControls()
    MyBase.CreateChildControls()

    Me.PageManager = Me.CMSPageManager1
End Sub

Protected Overloads Overrides Sub OnPreRender(ByVal e As EventArgs)
    MyBase.OnPreRender(e)

    Me.ltlTags.Text = Me.HeaderTags
End Sub
```

Now switch to the **Source mode** (HTML mode) copy and paste the HTML code from the sample **home.htm** file (inside <body></body> tags) after the <cc1:CMSPageManager /> control in the master page.

However, we need only the logo, main menu and footer. So we will replace the <!-- main content --> ... <!-- /main content --> section of the HTML code with the following code:

```
<asp:ContentPlaceHolder ID="plcMain" runat="server"></asp:ContentPlaceHolder>
```

This is a standard ASP.NET control that ensures loading of pages into the master page.

So the added code will look like this:

```
<div class="MainDiv">

<!-- logo -->
<br />
<div class="Logo">
    &nbsp;
</div>

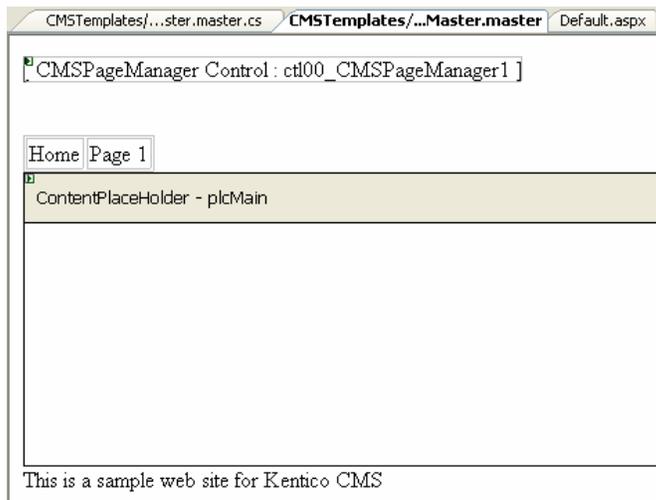
<!-- main menu -->
<div class="MainMenu">
    <table cellpadding="2" cellspacing="2" border="0">
        <tr>
            <td class="MainCMSMenuHighlightedMenuItem">Home</td>
            <td class="MainCMSMenuItem">Page 1</td>
        </tr>
    </table>
</div>

<!-- main content -->
<asp:ContentPlaceHolder ID="plcMain" runat="server"></asp:ContentPlaceHolder>
<!-- /main content -->

<!-- footer -->
<div class="Footer">
    This is a sample web site for Kentico CMS
</div>

</div>
```

When you switch to the **Design** tab, you should see a page preview like this:



**Save** the changes.



### Using CSS-based layout instead of tables

If you prefer using CSS-based layout, you can easily change the HTML code here and replace the tables with DIV elements. We use table-based layout by default since it's easier to understand, although we are aware of advantages of the CSS-based layout.

## 7.6 Main menu

Now we will add a dynamic **drop-down menu** to our master page. The drop-down menu can be implemented either by CMSMenu or CMSListMenu control. The first option is easier to use if you're not familiar with complex CSS styles, so we will use it now.

Please note: If you prefer using a drop-down menu based on the CSS styles and UL/LI elements, you can try to use the CMSListMenu later (you can find more details and examples in Kentico CMS Controls Reference).

Switch to the **Source** mode of the **MyMaster.master** page and drag and drop the **CMSMenu** control inside the `<div class="MainMenu">` element. Remove the original `<table>` element used for the static menu. The main menu section will look like this:

```
<!-- main menu -->
<div class="MainMenu">
    <ccl:CMSMenu ID="CMSMenu1" runat="server" />
</div>
```

Now switch back to the **Design** tab and set the following properties of the CMSMenu control:

- **Path:** /%
- **Layout:** Horizontal
- **CSSPrefix:** ;sub
- **Cursor:** Pointer

The **Path** property value specifies that the menu should start from the root of the site structure.

The **Layout** property allows you to choose between vertical and horizontal menu.

The **CSSPrefix** property specifies the names of CSS styles for the top menu (standard style names) and for sub-menus (all style names are prefixed with **sub**).

The **Cursor** property specifies the type of cursor when the user mouse-overs the menu.

**Save** the changes.

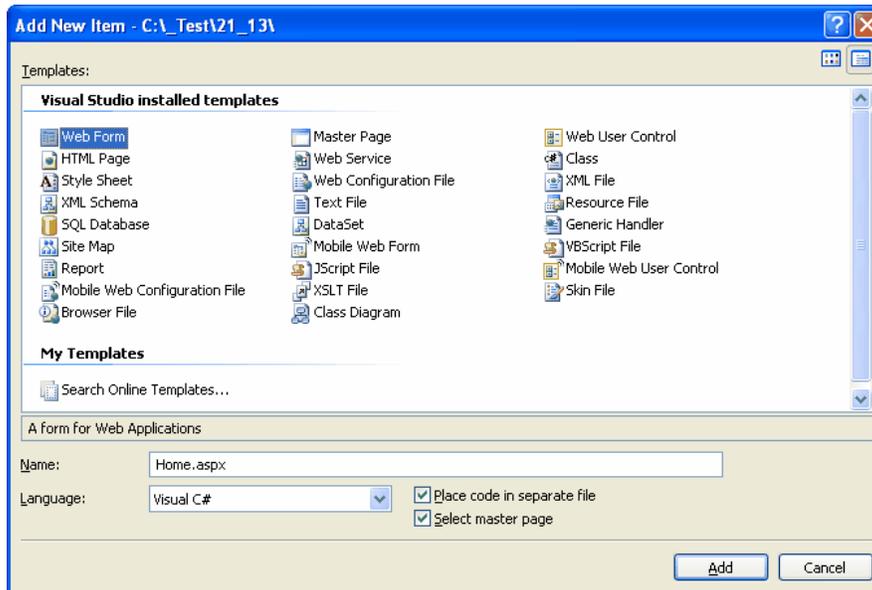


### Kentico CMS Controls and Web Parts

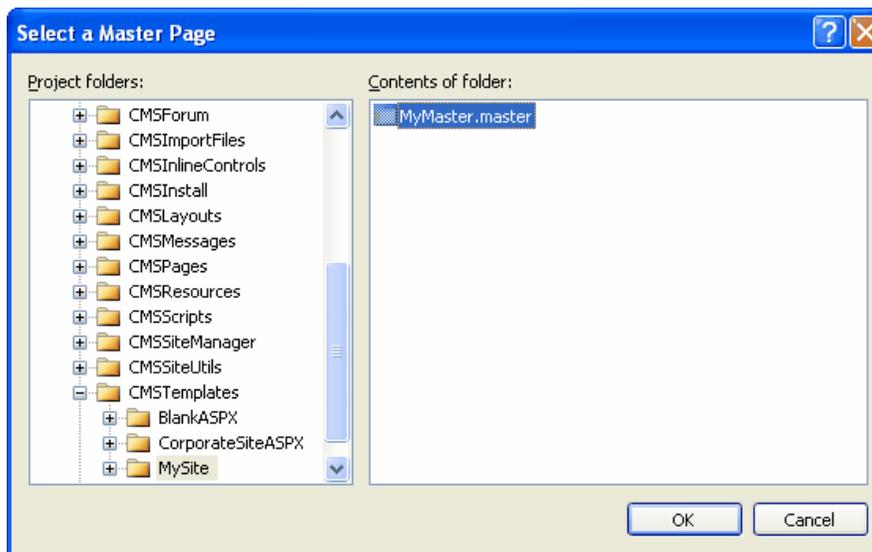
While Kentico CMS is delivered with a set of flexible server controls in the CMS.Controls.dll library, lots of functionality is only available in the web parts that are stored in the **CMSWebParts** folders. These web parts are standard ASCX user controls and they can be used on both portal engine templates and on ASPX pages. You only need to drag and drop the web parts on your ASPX page and set their properties in the Properties window of Visual Studio. All CMS controls have a relevant web part as well.

## 7.7 Home page

Now we will change the home page template to our web site. Right-click the **CMSTemplates/MySite** folder in the **Solution Explorer** and click **Add new item...** Choose to create a new page called **Home.aspx** and check the box **Select master page**:



Click **Add** and choose the **MyMaster.master** page in the **CMSTemplates/MySite** folder in the next dialog:



Copy the whole **<!-- main content -->** section from the **home.htm** file inside the **<asp:Content>** element. Now we will remove the text content with editable regions so that it can be managed by content editors:

- Remove the whole "Welcome to Our Business, Inc. Web Site..." text section.
- Remove the whole "Our Business, Inc. ..." content of the right box.

The complete code will look like this:

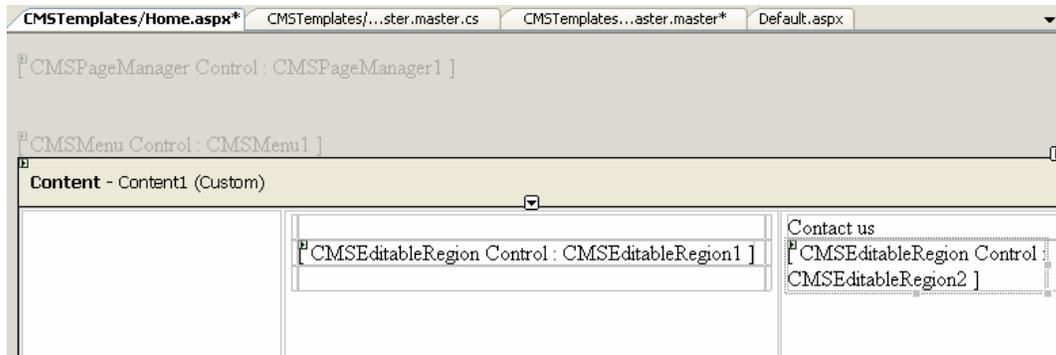
```
<%@ Page Language="C#" MasterPageFile="~/CMSTemplates/MySite/MyMaster.master"
AutoEventWireup="true"
CodeFile="Home.aspx.cs" Inherits="CMSTemplates_MySite_Home" Title="Untitled Page" %>
<asp:Content ID="Content1" ContentPlaceHolderID="plcMain" Runat="Server">
<!-- main content -->
<table style="width:100%;height:500px;border: 0px">
<tr valign="top">
<!-- left column -->
<td style="width:280px" class="HomePageLeftColumn">
</td>
<!-- center column -->
<td style="padding: 3px 5px 0px 5px;width:450px;">
<!-- center box -->
<table cellpadding="0" cellspacing="0" border="0" class="ContainerWithCorners"
width="100%">
<tr class="ContainerWithCornersRow">
<td class="ContainerWithCornersTopLeft">&nbsp;</td>
<td class="ContainerWithCornersTop">&nbsp;</td>
<td class="ContainerWithCornersTopRight">&nbsp;</td>
</tr>
<tr>
<td class="ContainerWithCornersLeft">&nbsp;</td>
<td class="ContainerWithCornersContent" valign="top">

</td>
<td class="ContainerWithCornersRight">&nbsp;</td>
</tr>
<tr class="ContainerWithCornersRow">
<td class="ContainerWithCornersBottomLeft">&nbsp;</td>
<td class="ContainerWithCornersBottom"></td>
<td class="ContainerWithCornersBottomRight">&nbsp;</td>
</tr>
</table>
</td>
<!-- right column -->
<td style="padding: 3px 0px 0px 5px;width:270px">
<!-- text box -->
<table cellpadding="0" cellspacing="0" style="width: 100%;margin-bottom: 10px;"
class="Blue">
<tr>
<td class="BoxTitle">Contact us
</td>
</tr>
<tr>
<td class="BoxArea" style="height: 19px">

</td>
</tr>
</table>
</td>
</tr>
</table>
<!-- /main content -->
</asp:Content>
```

Switch to the **Design** tab and drag and drop two **CMSEditableRegion** controls on the page:

- one into the center cell of the center box
- one into the bottom cell of the right box



Set the following properties of the **CMSEditableRegion** controls:

- **ID:** txtMain
- **DialogHeight:** 300
- **RegionType:** HtmlEditor
- **RegionTitle:** Main content
  
- **ID:** txtRight
- **DialogHeight:** 280
- **RegionType:** HtmlEditor
- **RegionTitle:** Right content

Switch to the **code behind** and add reference to the **CMS.UIControls** namespace:

**[C#]**

```
using CMS.UIControls;
```

**[VB.NET]**

```
Imports CMS.UIControls
```

You also need to change the class definition so that it inherits from the **TemplatePage** class:

**[C#]**

```
public partial class CMSTemplates_MySite_Home : TemplatePage
```

**[VB.NET]**

```
Partial Class CMSTemplates_MySite_Home  
    Inherits TemplatePage
```

Save the changes.

Our master page and page template for the home page are ready. Now we need to register the home page template in Kentico CMS. Open Kentico CMS in a web browser and go to **Site Manager (http://localhost/KenticoCMS/CMSSiteManager) -> Development -> Page templates**.

Click the root and click **New category**. Create a new category with name **My web site**.

Click **New template** and enter the following values:

- **Template display name:** Home page
- **Template code name:** HomePage

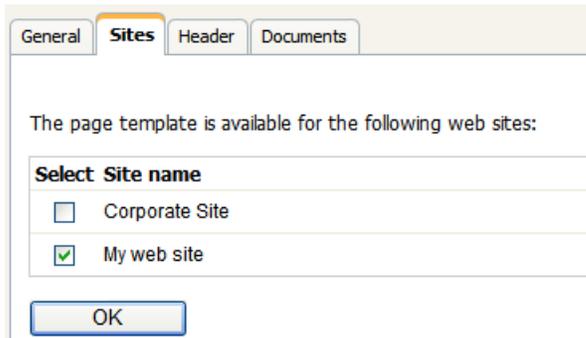
Click **OK** and set the following values on the **General** tab:

- **Template type:** ASPX page
- **File:** ~/CMSTemplates/MySite/home.aspx

The screenshot shows the Kentico CMS Site Manager interface. The top navigation bar includes 'Sites', 'Administration', 'Settings', 'Development', 'Licenses', 'Support', and 'Buy'. The 'Development' tab is active. On the left, a tree view shows the 'Page Templates' folder expanded, with 'My web site' and 'Home page' selected. The main area displays the 'Page template properties' for 'Home page' under the 'General' tab. The 'Save' button is visible. The configuration fields are as follows:

Template display name:	Home page
Template code name:	HomePage
Category:	My web site
Template description:	
Thumbnail:	Upload: <input type="text"/> <input type="button" value="Browse..."/>
Template type:	<input type="radio"/> Portal page <input checked="" type="radio"/> ASPX page
File name:	~/CMSTemplates/MySite/home.aspx

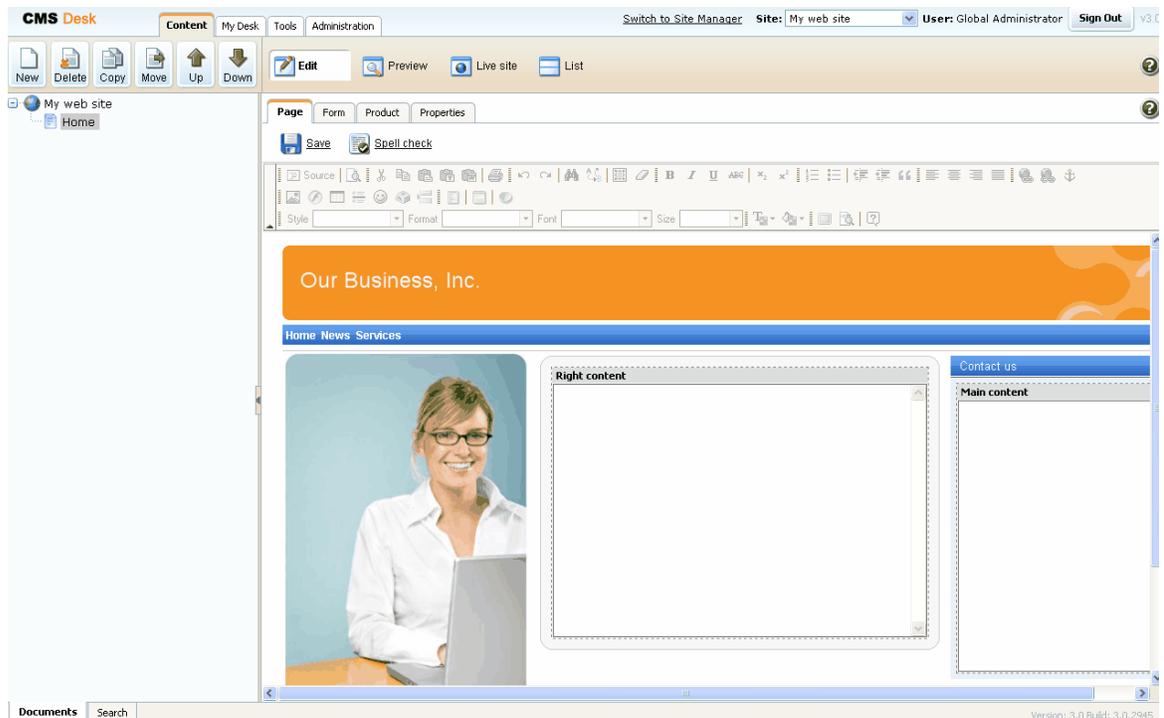
Click **OK** to save and click the **Sites** tab. Assign the new page template to your web site and click **OK**.



Go to **CMS Desk -> Content**. Click root and click **New**. Choose to create a new **Page (menu item)** and enter the following values:

- **Page name:** Home
- **Use page template:** My web site/Home page

Click **Save** and Click the **Page** tab. Now you can see that the page contains two editable regions.



Enter the following text:

- **Main text:** Here comes the welcome text.
- **Contact text:** Call 800 111 2222

Click **Save**. Click **Live site** and you will see the home page of your new web site.

## Configuring the web site home page

When the site visitor comes to the root of your web site (e.g. to <http://www.mydomain.com>), the system needs to know which page should be displayed as a home page. Go to **Site Manager -> Settings**, select **My web site**, click **Web site** and make sure the value **Default alias path** is set to **/Home**, which is the alias path of our new home page.

The screenshot shows the 'Settings' page for 'My web site' in the Kentico CMS Site Manager. The 'Default alias path' field is highlighted with a red box and contains the value '/Home'. Other settings include cache times, culture, and page titles.

Setting	Value	Inherit from global settings
Cache content (minutes)	10	<input checked="" type="checkbox"/>
Cache images (minutes)	10	<input checked="" type="checkbox"/>
Cache page info (minutes)	10	<input checked="" type="checkbox"/>
Combine with default culture	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Combine files with default culture	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Default alias path	/Home	<input type="checkbox"/>
Default culture of the content	Afrikaans - South Africa	<input type="checkbox"/>
Page description prefix		<input checked="" type="checkbox"/>
Page key words prefix		<input checked="" type="checkbox"/>
Page title format	{%prefix%} - {%pagetitle_orelse_name%}	<input checked="" type="checkbox"/>
Page title prefix	My Site	<input type="checkbox"/>
Page not found for non-published documents	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Page not found URL		<input checked="" type="checkbox"/>
Exclude document types from SQL search		<input checked="" type="checkbox"/>
Exclude documents from SQL search		<input checked="" type="checkbox"/>
Control element	div	<input checked="" type="checkbox"/>

## 7.8 News page

Now we will create the News section of our web site. Go to Visual Studio and create a new web form in the **CMSTemplates\MySite** folder, call it **NewsPage.aspx**, check the **Select master page** box and click **Add**. Choose the **MyMaster.master** master page and click **OK**.

Switch to the **Design** tab, drag and drop and configure the following controls:

### CMSBreadCrumbs

(no properties to be set)

### CMSRepeater

- ClassNames: cms.news
- TransformationName: cms.news.preview
- SelectedItemTransformationName: cms.news.default
- ItemSeparator: <hr />

Switch to the HTML mode and add the following HTML code between both controls:

```
<h1>News</h1>
```

When you switch back to the **Design** tab, you should see a page like this:



Switch to the **code behind** and add reference to the CMS.UIControls namespace:

**[C#]**

```
using CMS.UIControls;
```

**[VB.NET]**

```
Imports CMS.UIControls
```

You also need to change the class definition so that it inherits from the **TemplatePage** class:

**[C#]**

```
public partial class CMSTemplates_MySite_NewsPage : TemplatePage
```

**[VB.NET]**

```
Partial Class CMSTemplates_MySite_NewsPage  
    Inherits TemplatePage
```

Save all changes.

### Page Template Registration

Go to **Site Manager -> Development -> Page templates**, click the **My web site category** and click **New template**. Create a new page template with following details:

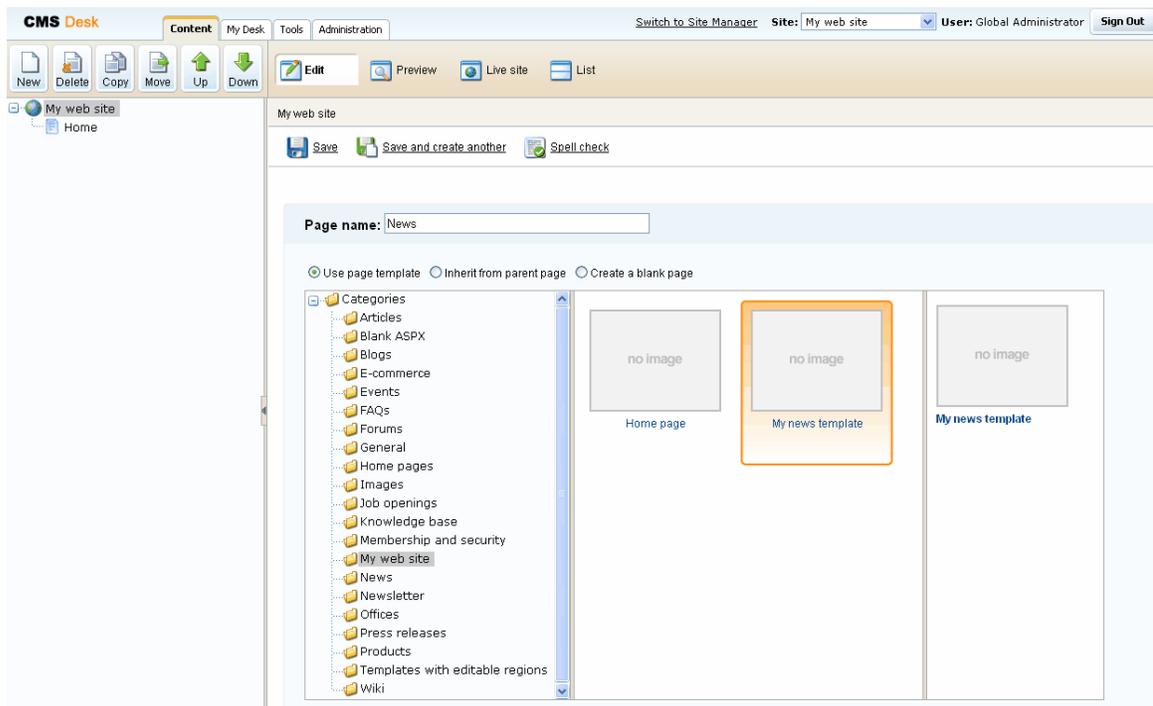
- Template display name: My news template
- Template code name: mynewstemplate

On the detail page, please choose:

- Template type: ASPX
- File name: ~/CMSTemplates/MySite/NewsPage.aspx

Switch to the **Sites** tab and assign the template to **My web site**.

Go to **CMS Desk** -> **Content**, click root and click **New**. Choose to create a new **Page (menu item)** using the **My web site/My news template page** template and call the page **News**.



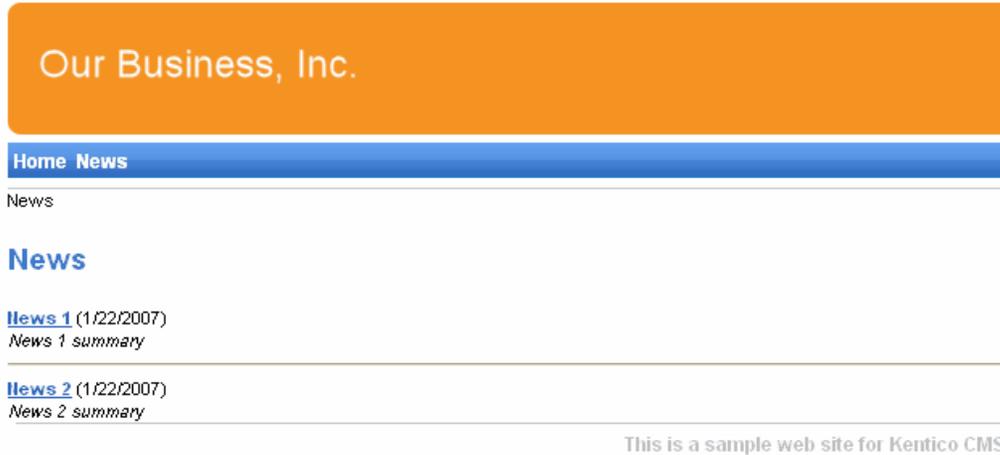
Click **Save**. Click the **News** page and click **New**, choose to create a **news** document and enter the following text:

- News title: News 1
- Release date: click the date-time picker and click **Now** and click **OK**.
- News summary: News 1 summary.
- News text: News 1 text.
- Publish from, to: leave the fields blank.

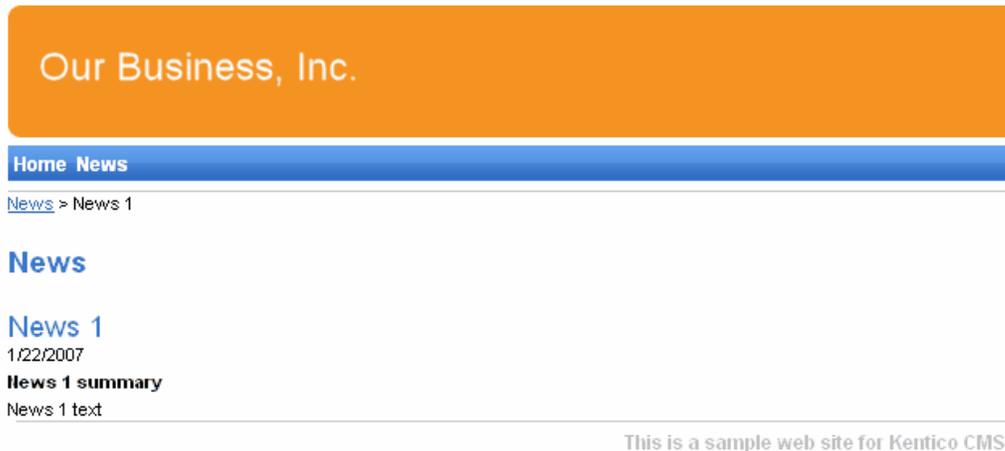
Click **Save and create another** and enter the following values:

- News title: News 2
- Release date: click the date-time picker and click **Now** and click **OK**.
- News summary: News 2 summary.
- News text: News 2 text.
- Publish from, to: leave the fields blank.

When you click **/News** and **Live site** now, you will see the list of news under the **News** section:



As you can see, the main page **/News** displays the list of the news items that are placed under it. This is an example of how the content is logically structured in Kentico CMS. When you click **/News/News 1** now, you will see the detail view:



The breadcrumbs now show you current path on the web site: **News > News 1**. The position is also reflected in the URLs:

- The URL of the News page is **/news.aspx**
- The URL of the News 1 page is **/news/news-1.aspx**

This makes the web site more accessible to both people and search engines, such as Google.

## How it works

1. You go to the **/News page**.
2. The **CMSRepeater** control checks if you have selected some particular news item (based on its `ClassNames` property value).
3. It finds out that you have selected a page document, so it looks for all underlying news documents and displays them as a list using the **cms.news.preview** transformation.

When you click on some particular news item, such as **/News/News 1**, the **NewsRepeater** web part uses the **cms.news.default** transformation instead and displays the detail view.



### Path expressions

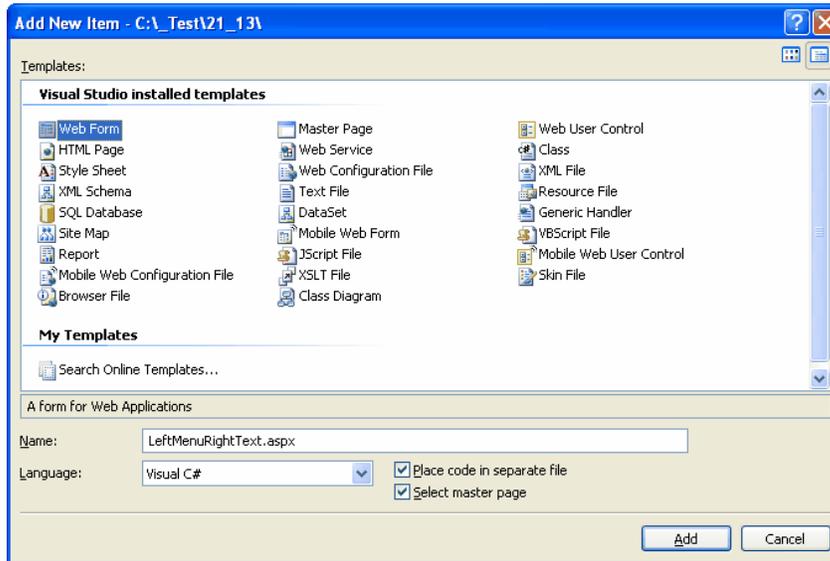
The Path property in web parts supports following special expressions that allow you select the content dynamically:

<code>/%</code>	All documents on the web site.
<code>/news/%</code>	All documents under <code>/News</code> .
<code>/news/news1</code>	News1 document.
<code>./%</code>	All items under the current document.
<code>./logo</code>	Logo document under the current document.
<code>./images/%</code>	All images under the <i>images</i> child document.
<code>../contacts/%</code>	All documents under the sibling document <i>contacts</i> .
<code>/{0}/%</code>	All documents under the current first level document.  Example: if the currently selected document is <code>/news/news1</code> the expression is evaluated as <code>/news/%</code>

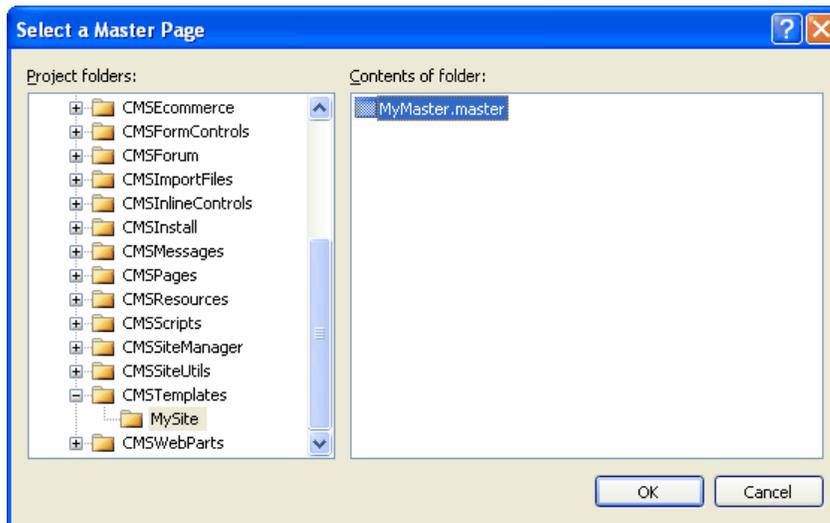
## 7.9 Services page

Now we will create a new site section for services. This site section will contain a left tree menu and a single editable region.

Go to Visual Studio and choose to create a new ASPX page in folder **CMSTemplates\MySite**. Call the page **LeftMenuRightText.aspx** and check the box **Select master page**:



Choose the **CMSTemplates\MySite\MyMaster.master** page on the next dialog:



Now enter the following HTML layout code inside the <asp:content> elements of the newly created page:

```
<table width="100%">
  <tr valign="top">
    <td width="20%">

    </td>
    <td width="80%">

    </td>
  </tr>
</table>
```

Switch to the **Design** tab and you will see a preview of the page, including the inherited master page.

Drag and drop the CMSTreeMenu control to the left column and the CMSEditableRegion control to the right column:



Set the following properties of the controls:

**CMSTreeMenu1:**

- **Path:** /{0}% (this means that the menu starts from the second level)
- **MenuItemImageURL:** ~/app\_themes/mysite/images/bullet.gif
- **MenuItemOpenImageURL:** ~/app\_themes/mysite/images/bullet.gif  
(the ~ character represents the root of the web site and it ensures that the image will be displayed correctly whether you run the web sites in the root or in a virtual directory)

**CMSEditableRegion1:**

- **RegionType:** HTMLEditor
- **Height:** 400
- **RegionTitle:** Main Text

Switch to the **code behind** and add reference to the CMS.UIControls namespace:

**[C#]**

```
using CMS.UIControls;
```

**[VB.NET]**

```
Imports CMS.UIControls
```

You also need to change the class definition so that it inherits from the **TemplatePage** class:

**[C#]**

```
public partial class CMSTemplates_MySite_LeftMenuRightText : TemplatePage
```

**[VB.NET]**

```
Partial Class CMSTemplates_MySite_LeftMenuRightText  
    Inherits TemplatePage
```

**Save** all changes.

Now we need to register our new page template. Go to **CMS Site Manager -> Development -> Page Templates** and click **My Web Site** and click **New template**. Enter the following values:

- **Template display name:** Left menu with right text
- **Template code name:** LeftMenuWithRightText

Click **OK**. Enter the following values on the **General** tab:

- **Template type:** ASPX page
- **File name:** ~/CMSTemplates/MySite/LeftMenuRightText.aspx

### Page template properties

Templates ▸ Left menu with right text

**General** Sites Header Documents

 Save

Template display name:

Template code name:

Category:  ▾

Template description:

Thumbnail: Upload:

Template type:  Portal page  ASPX page

File name:

Switch to the **Sites** tab and assign the template to your web site:

General **Sites** Header Documents

The page template is available for the following web sites:

**Select Site name**

Corporate Site

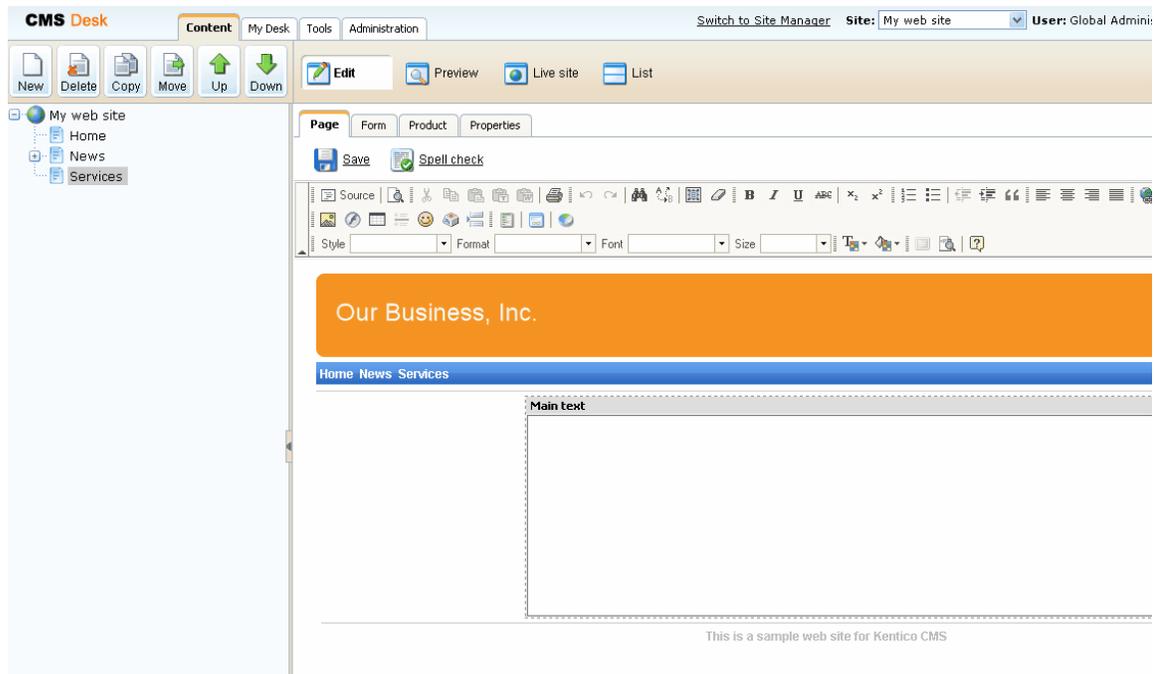
My web site

Now that we created the page template, we can start adding new pages based on this template. Go to **CMS Desk -> Content**, click the **root** and click **New**. Choose to create a new **Page (menu item)** and enter the following values:

- **Page name:** Services
- **Use page template:** My web site/Left menu with right text

Click **Save**.

Now you are redirected to the **Page** tab and you can enter some content on the Services page:



## Adding sub-pages

Click **New** in the main toolbar and choose to create a new page under the /Services page. Call the page **Service 1** and choose to use page template **My web site/Left menu with right text**. Click **Save**. Enter some text into the editable region and click **Save**.

## 7.10 Products page

### 7.10.1 Overview

Now we will add a new Products section displaying a list of computers and their technical specification. You will learn how to create a new document type *Computer* and how to display a list of computers on the site. You will also learn how write transformations.

### 7.10.2 New document type

Each document in the Kentico CMS repository is of some type, such as news, product, article, etc. Each document type has its own fields. Our document type will describe computer, so it will have a computer name, processor type, RAM size, disk size and a product image field.

Go to **Site Manager -> Development -> Document types** and click **New document type**. You are redirected to the New document type wizard. In the first step, enter the following values:

- **Document type display name:** *Computer (this name will be displayed to the users)*
- **Document type code name:** *custom.computer (custom is your namespace to distinguish your document types from system types that use the cms namespace, computer is the document type); you will use this value in web part properties later.*

The screenshot shows the 'New document type' wizard in the Kentico CMS Site Manager. The interface includes a top navigation bar with tabs for 'Sites', 'Administration', 'Settings', 'Development', 'Licenses', and 'Support'. The 'Development' tab is active. On the left, a tree view shows the 'Development' section expanded, with 'Document types' selected. The main content area is titled 'New document type' and shows 'Step 1: General'. Below the title, there is a instruction: 'Please enter document type display name (for users) and code name (it will be used in your code when necessary)'. The form contains two input fields: 'Document type display name' with the value 'Computer' and 'Document type code name' with the value 'custom' in the 'namespace' field and 'computer' in the 'document type' field. A 'Next' button is located at the bottom right of the form.

Click **Next**.

In step 2, you need to choose the name of the database table that will be used for storing computer details. You also need to enter the name of the primary key in this table. Leave the default values:

**Step 2**

**Data type**

Please choose document data type. If you choose a document type with custom attributes you will also need to supply names of the new database table and its primary key.

The document type has custom fields

Table name:

Primary key name:

The document type is only a container without custom fields

Click **Next**. The wizard has created a new database table for computers. Now you need to define the fields (columns of the table). Click **New attribute** to create a new field. Enter the following values:

- **Attribute name:** ComputerName
- **Attribute type:** Text
- **Attribute size:** 200
- **Field caption:** Computer name
- **Field type:** Text box

Click **OK**. Click **New attribute**. Enter:

- **Attribute name:** ComputerProcessorType
- **Attribute type:** Text
- **Attribute size:** 200
- **Field caption:** Processor type
- **Field type:** Drop-down list, choose to use **Options**
- **Options:** enter the following options, one per line:  
Athlon;Athlon  
Pentium XEON;Pentium XEON  
Pentium Core 2 Duo;Pentium Core 2 Duo

Click **OK**. Click **New attribute**. Enter:

- **Attribute name:** ComputerRamSize
- **Attribute type:** Integer Number
- **Field caption:** RAM (MB)
- **Field type:** Text box

Click **OK**. Click **New attribute**. Enter:

- **Attribute name:** ComputerHddSize
- **Attribute type:** Integer Number
- **Field caption:** HDD (GB)
- **Field type:** Text box

Click **OK**. Click **New attribute**. Enter:

- **Attribute name:** ComputerImage
- **Attribute type:** File
- **Allow empty value:** check the box
- **Field caption:** Image
- **Field type:** Upload file

Click **OK**.

**Step 3** | **Fields**  
Please define custom attributes of the document type and their appearance in the editing form. You can define attributes, such as product number, product weight, press release text, etc.

The changes were saved.

computerID	
ComputerName	
ComputerProcessorType	
ComputerRamSize	
ComputerHddSize	
<b>ComputerImage</b>	

Database

Attribute name: ComputerImage

Attribute type: File

Attribute size:

Allow empty value:

Attribute default value:

Display attribute in the editing form

Field

Field caption: Image

Field type: Upload file

OK

Next

Click **Next**.

Now you need to choose the field that will be used as document name. Choose the **ComputerName** field. It means that when you create a new computer document, its name will be automatically taken from the ComputerName value and this value will appear in site navigation and in CMS Desk content tree.

**Step 4** | **Additional settings**  
Please choose the source field that will be used as a document name. You can choose either one of the custom fields or you can choose to use document name as a separate field.

Document name source:

Click **Next**. In step 5, you need to select the document types under which the computers can be added in the content tree. Check only the **Page (menu item)** value, which means the editors will be able to create computer documents only under some page, not under article or news document in the content tree.

**Step 5** | **Parent types**  
Please select document types under which this document template can be placed.

<input type="checkbox"/>	FAQ	CMS.Faq
<input type="checkbox"/>	File	CMS.File
<input type="checkbox"/>	Folder	CMS.Folder
<input type="checkbox"/>	Job opening	CMS.Job
<input type="checkbox"/>	Knowledge base article	CMS.KBArticle
<input type="checkbox"/>	Laptop	CMS.Laptop
<input type="checkbox"/>	News	CMS.News
<input type="checkbox"/>	Office	CMS.Office
<input checked="" type="checkbox"/>	Page (menu item)	CMS.Menuitem
<input type="checkbox"/>	PDA	CMS.Pda
<input type="checkbox"/>	Press release	CMS.PressRelease

Click **Next**. In step 6, you need to choose which web sites will use this document type. Check **My web site**.

**Step 6** | **Sites**  
Please select sites where this document type can be used:

Select Site name

- Corporate Site
- My web site

Next

Click **Next**. In Step 7, you are asked to specify how documents of this type should be indexed for searching and displayed in search results. Select the following values in the drop-downs:

- **Title field:** ComputerName
- **Content field:** DocumentContent
- **Image field:** ComputerImage
- **Date field:** DocumentCreatedWhen

Leave the default values for the rest of the options and click **Next**.

**Step 7** | **Search options**  
Please set search fields for Smart search module.

Title field: ComputerName

Content field: DocumentContent

Image field: ComputerImage

Date field: DocumentCreatedWhen

Set automatically

Field name	Content	Searchable	Tokenized	Custom search name
ComputerID	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
ComputerName	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text"/>
ComputerProcessorType	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text"/>

Next

The wizard has finished the configuration of the new document type. It has automatically created not only the database table, but also the SQL queries for SELECT, INSERT, UPDATE, DELETE operations and a default transformation.

**Step 8** | The wizard has finished

The setup has finished the following steps:

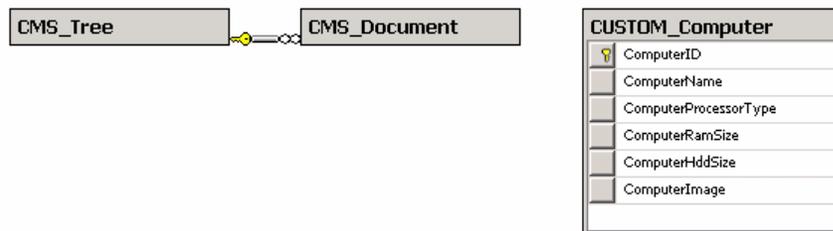
- ▶ The new document type was created.
- ▶ The new editing form was created.
- ▶ The document types were added among allowed child types of the new document type.
- ▶ The sites were selected where this document type can be used.
- ▶ The default queries were created.
- ▶ The default ASCX transformations were created.
- ▶ The default permission names were created.
- ▶ The default icon was created.
- ▶ Document smart search specification was created.

You have learned how to define a new document type.



### How the content is stored

As you already know, the new document type Computer has its own database table. Each document is stored in three tables: **CMS\_TREE** (tree structure), **CMS\_Document** (document properties and metadata) and the custom table - in this case **CUSTOM\_Computer**:



The system automatically ensures all operations on these tables. The advantage of this storage is that it's very fast and **you can easily write standard SQL SELECT queries to retrieve data** from the repository (i.e. from the Microsoft SQL Server database).

### 7.10.3 Transformations

Now that we have created a new document type, we need to prepare the transformations that will be used for displaying product details in a list and in a detail view.

In the **Computer** document type properties dialog, click the **Transformations** tab:

 **Document Type Properties**

Document Types ▸ Computer

General Fields Form **Transformations** Queries Child types Sites Alternative forms Search fields Documents

 New Transformation

Transformation name: LIKE

**Actions Transformation name** ▲

-   Default
-   Preview

As you can see, the wizard has created some default transformations. We will use them for our detail view. Edit the **Default** transformation, clear the default code and enter the following code:

```
<h1>
  <## Eval("ComputerName") %>
</h1>
<table>
  <tr>
    <td>
      Processor:
    </td>
    <td>
      <## Eval("ComputerProcessorType") %>
    </td>
  </tr>
  <tr>
    <td>
      RAM (MB):
    </td>
    <td>
      <## Eval("ComputerRamSize") %>
    </td>
  </tr>
  <tr>
    <td>
      HDD (GB):
    </td>
    <td>
      <## Eval("ComputerHddSize") %>
    </td>
  </tr>
  <tr>
    <td>
      Image:
    </td>
    <td>
      <## GetImage("ComputerImage") %>
    </td>
  </tr>
</table>
```

Click **Save**. As you can see the transformation code is the standard ItemTemplate code that you may already know from ASP.NET 2.0 Repeater and DataList controls. It combines HTML code with ASP.NET commands and data binding expressions. You can use several built-in functions, such as **GetImage** that simplify some tasks. You can find the list of the most important functions directly under the transformation code.

Now we will create transformation for the list of computers. Go back to the transformation list and edit the **Preview** transformation. Clear the default code and enter the following code:

```
<div style="text-align:center;border: 1px solid #CCCCCC">
<h2>
<a href="<%# GetDocumentUrl() %>"><%# Eval("ComputerName") %></a>
<h2>
?maxsize=120" />
</div>
```

Click **OK**.

Please note how the link to the document is created:

```
<a href="<%# GetDocumentUrl() %>"><%# Eval("ComputerName") %></a>
```

It consists of standard HTML tags for links and it inserts the URL and link text dynamically.

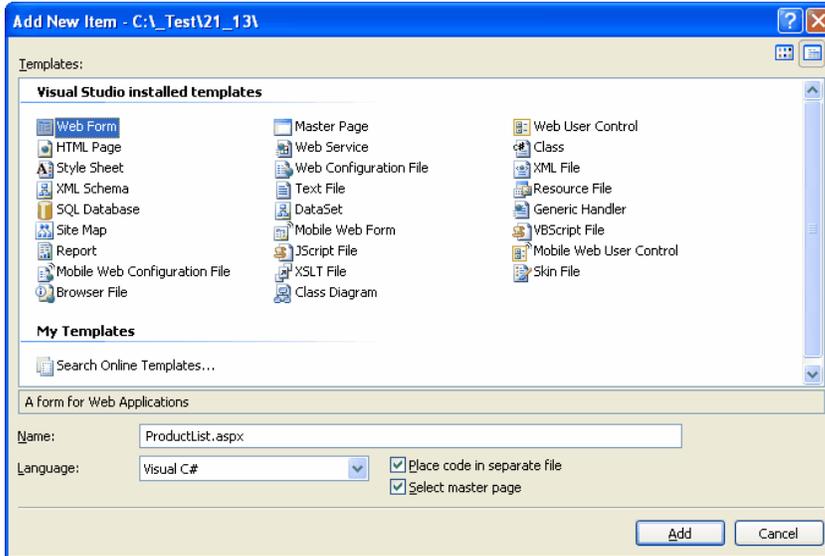
Similarly, you can create an image tag with parameter that ensures automatic resize of the longest side to 120 pixels on the server side:

```
?maxsize=120" />
```

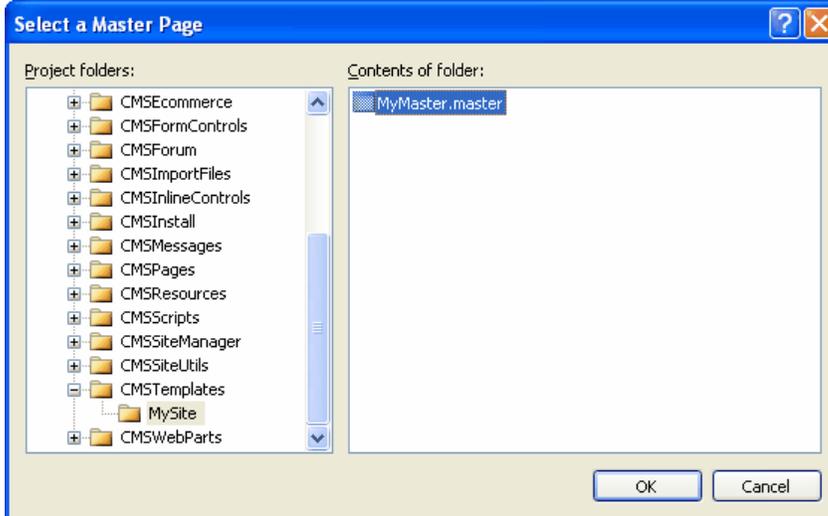
You have learned how to write transformations for displaying the content of structured documents.

### 7.10.4 Page template

Now we get to the final step of this chapter: publishing computer specifications on your web site. Go to Visual Studio and choose to create a new ASPX page in folder **CMSTemplates\MySite**. Call the page **ProductList.aspx** and check the box **Select master page**:



Choose the **CMSTemplates\MySite\MyMaster.master** page on the next dialog:



Switch to the **Design** tab and you will see a preview of the page, including the inherited master page. Drag and drop the CMSBreadCrumbs and CMSDataList controls on the page:



Set the following properties of the controls:

#### **CMSDataList**

- **ClassNames:** custom.computer (the document types to be displayed)
- **OrderBy:** ComputerName ASC
- **TransformationName:** custom.computer.preview
- **SelectedItemTransformationName:** custom.computer.default
- **RepeatColumns:** 2

Switch to the **code behind** and add reference to the CMS.UIControls namespace:

**[C#]**

```
using CMS.UIControls;
```

**[VB.NET]**

```
Imports CMS.UIControls
```

You also need to change the class definition so that it inherits from the **TemplatePage** class:

**[C#]**

```
public partial class CMSTemplates_MySite_ProductList : TemplatePage
```

**[VB.NET]**

```
Partial Class CMSTemplates_MySite_ProductList  
    Inherits TemplatePage
```

Save all changes.

Now we need to register our new page template. Go to CMS Site Manager -> Development -> Page Templates and click **My Web Site** and click **New template**. Enter the following values:

- **Template display name:** Product list
- **Template code name:** ProductList

Click **OK**. Enter the following values on the General tab:

- **Template type:** ASPX page
- **File name:** ~/CMSTemplates/MySite/ProductList.aspx

### Page template properties

Templates ▶ Product list

General Sites Header Documents

 Save

Template display name:

Template code name:

Category:  ▼

Template description:

Thumbnail: Upload:

Template type:  Portal page  ASPX page

File name:

Switch to the **Sites** tab and assign the template to your web site:

General **Sites** Header Documents

The page template is available for the following web sites:

Select Site name

Corporate Site

My web site

Now that we created the page template, we can start adding new pages based on this template. Go to **CMS Desk -> Content**, click the **root** and click **New**. Choose to create a new **Page (menu item)** and enter the following values:

- **Page name:** Products
- **Use page template:** My web site/Product list

Click **Save**.

Now we need to enter some computer details. Click **/Products** in the content tree and click **New**. Choose to create a new computer. Enter the following value:

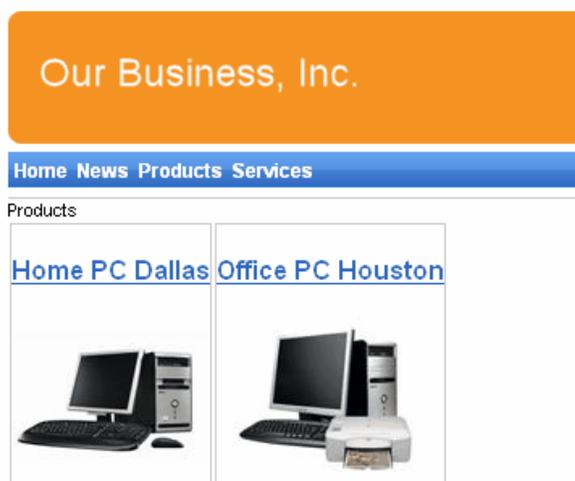
- **Computer name:** Home PC Dallas
- **Processor type:** Athlon
- **RAM (MB):** 512
- **HDD (GB):** 80
- **Image:** upload some image (you can find sample images in folder <Kentico CMS installation> \CodeSamples\SampleWebTemplate\Computer\_Images)
- **Publish from/to** - leave the values blank

Click **Save and create another** and enter:

- **Computer name:** Office PC Houston
- **Processor type:** Pentium Core 2 Duo
- **RAM (MB):** 1024
- **HDD (GB):** 120
- **Image:** upload some image (you can find sample images in folder <Kentico CMS installation> \CodeSamples\SampleWebTemplate\Computer\_Images)
- **Publish from/to** - leave the values blank

Click **Save**.

Now, when you click **/Products** you will see a page like this:



When you click on some link, you will see computer details:



You have learned how to define a new document type and how to publish its documents on the web site.

## 7.11 Search page

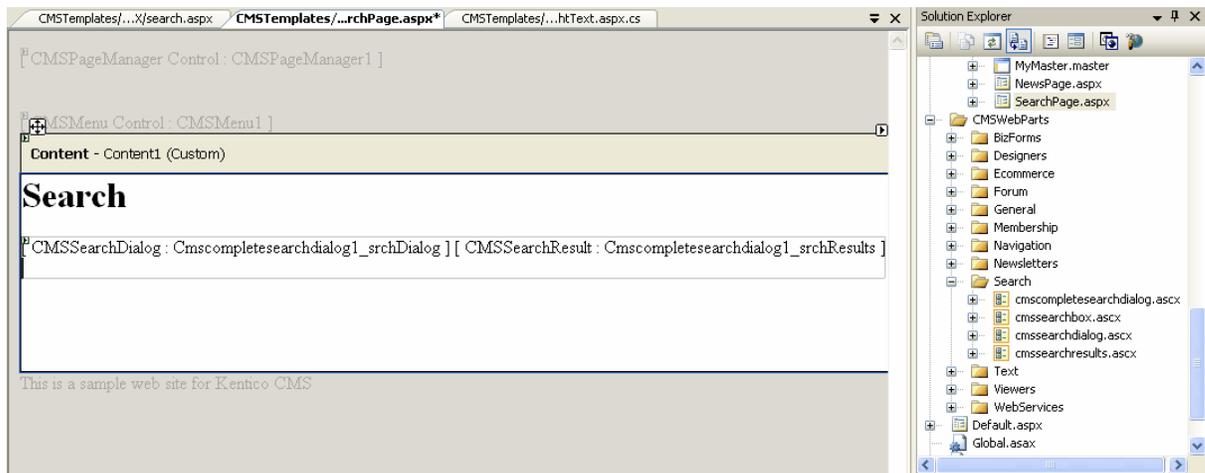
Kentico CMS allows you to full-text search all documents in Kentico CMS repository. You can also configure it so that it searches the text inside uploaded documents, such as PDF, DOC or XLS documents. The configuration is described in **Developer's Guide -> Installation and deployment -> Configuration of full-text search in files**. It's not necessary to configure it at this moment since we will use the database search only.

Go to Visual Studio and choose to create a new ASPX page in folder **CMSTemplates\MySite**. Call the page **SearchPage.aspx** and check the box **Select master page**. Choose the **CMSTemplates\MySite\MyMaster.master** page on the next dialog.

Add the following HTML code inside the <asp:content> element:

```
<h1>Search</h1>
```

Switch to the **Design** tab and drag and drop the user control **CMSWebParts/search/cmscompletesearchdialog.ascx** under the Search header:



Set its properties:

- **TransformationName:** cms.root.searchresults
- **ShowSearchScope:** false

Switch to the **code behind** and add reference to the CMS.UIControls namespace:

**[C#]**

```
using CMS.UIControls;
```

**[VB.NET]**

```
Imports CMS.UIControls
```

You also need to change the class definition so that it inherits from the **TemplatePage** class:

**[C#]**

```
public partial class CMSTemplates_MySite_SearchPage : TemplatePage
```

**[VB.NET]**

```
Partial Class CMSTemplates_MySite_SearchPage
    Inherits TemplatePage
```

Save the changes. Now we need to register our new page template. Go to **CMS Site Manager -> Development -> Page Templates** and click **My Web Site** and click **New template**. Enter the following values:

- **Template display name:** Search page
- **Template code name:** searchpage

Click **OK**. Enter the following values on the General tab:

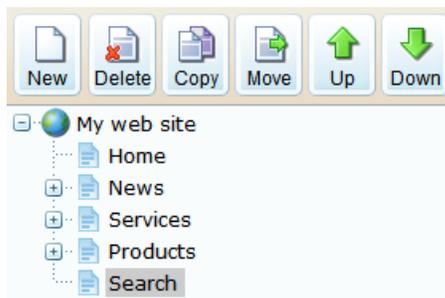
- **Template type:** ASPX page
- **File name:** ~/CMSTemplates/MySite/SearchPage.aspx

Switch to the **Sites** tab and assign the template to **My web site** site.

Go to **CMS Desk -> Content**, click the **root** and click **New**. Choose to create a new **Page (menu item)** and enter the following values:

- **Page name:** Search
- **Use page template:** My web site/Search page

Click **Save**. Click the **Down** arrow in the main toolbar until you move the **Search** page at the end of the list.



Click the new **/Search** page. You will see the search page. Enter **PC** in the **Search for** box and click **Go**.

## Our Business, Inc.

[Home](#) [News](#) [Services](#) [Products](#) [Search](#)

### Search

Search for:

Search mode:  ▼

[Home PC Dallas](#)  
Path: /Products/Home PC Dallas

[Office PC Houston](#)  
Path: /Products/Office PC Houston

---

This is a sample web site for Kentico CMS

When you click some search result, you are redirected to the appropriate document.



#### Modifying the search results format

If you prefer a different design of the search results, you can modify the format in **Site Manager -> Development -> Document types -> Root -> Transformations -> searchresults** transformation.

## 7.12 Secured section for partners

Kentico CMS allows you to create secured site sections that can be accessed only by users who have a valid user name and password. We will create a simple page for partners that can be accessed only by registered users.

Go to **CMS Desk -> Content**, click root and click **New**. Choose to create a new **Page**. Enter page name **Partners** and choose to use the page template **My web site/Left menu with right text**. Click **Save**. Click the **Down** arrow in the main toolbar until you move the **Partners** page at the end of the list.

Click **Page** and enter the following text: This is a secured page for partners. Click **Save**.

Click **Properties -> Security**. In the **Access** section of the dialog, click **Yes** and click **OK**. This will ensure that the page can be accessed only by authenticated users.

Now we need to create the logon page. Go to Visual Studio and choose to create a new ASPX page in folder **CMSTemplates\MySite**. Call the page **LogonPage.aspx** and check the box **Select master page**. Choose the **CMSTemplates\MySite\MyMaster.master** page on the next dialog.

Add the following HTML code inside the `<asp:content>` element:

```
<table border="0" width="100%">
  <tr valign="top">
    <td style="width:50%">
    </td>
    <td style="width:50%">
    </td>
  </tr>
</table>
```

Switch to the **Design** tab and drag and drop the user control **CMSWebParts/Membership/LogonForm.aspx** inside the left column and the **CMSWebParts/Membership/RegistrationForm.aspx** user control into the right column. Set their properties:

### LogonForm1

- **AllowPasswordRetrieval:** true
- **SendEmailFrom:** <your e-mail address>

### RegistrationForm1

- **AssignRoles:** CMSPublicUsers
- **EnableUserAfterRegistration:** true

Switch to the **code behind** and add reference to the CMS.UIControls namespace:

**[C#]**

```
using CMS.UIControls;
```

**[VB.NET]**

```
Imports CMS.UIControls
```

You also need to change the class definition so that it inherits from the **TemplatePage** class:

**[C#]**

```
public partial class CMSTemplates_MySite_LogonPage : TemplatePage
```

**[VB.NET]**

```
Partial Class CMSTemplates_MySite_LogonPage  
    Inherits TemplatePage
```

**Save** the changes. Now we need to register the logon page. Go to **Site Manager -> Development -> Page templates**, choose category **My web site** and register a new page template with following values:

- **Template display name:** Logon page
- **Template code name:** LogonPage
- **Template type:** ASPX page
- **File name:** ~/CMSTemplates/MySite/logonpage.aspx

**Save** the changes, switch to the **Sites** tab and assign the template to **My web site**. Now go to **CMS Desk -> Content**, click **Root**, click **New** and choose to create a new folder. Call the folder **Special pages**. Create a new page under the **Special pages** folder, call it **Logon** and choose the page template **My web site -> Logon page**. Click **Save**.

## Configuring the logon page

We will need to configure the system so that it uses our new logon page. Go to **Site Manager -> Settings**, choose **My web site** in the drop-down list, click **Security** and set the value **Secured areas logon page** to **~/Special-pages/Logon.aspx**, which is the relative URL of our logon page (from the web application root). Click **Save**.

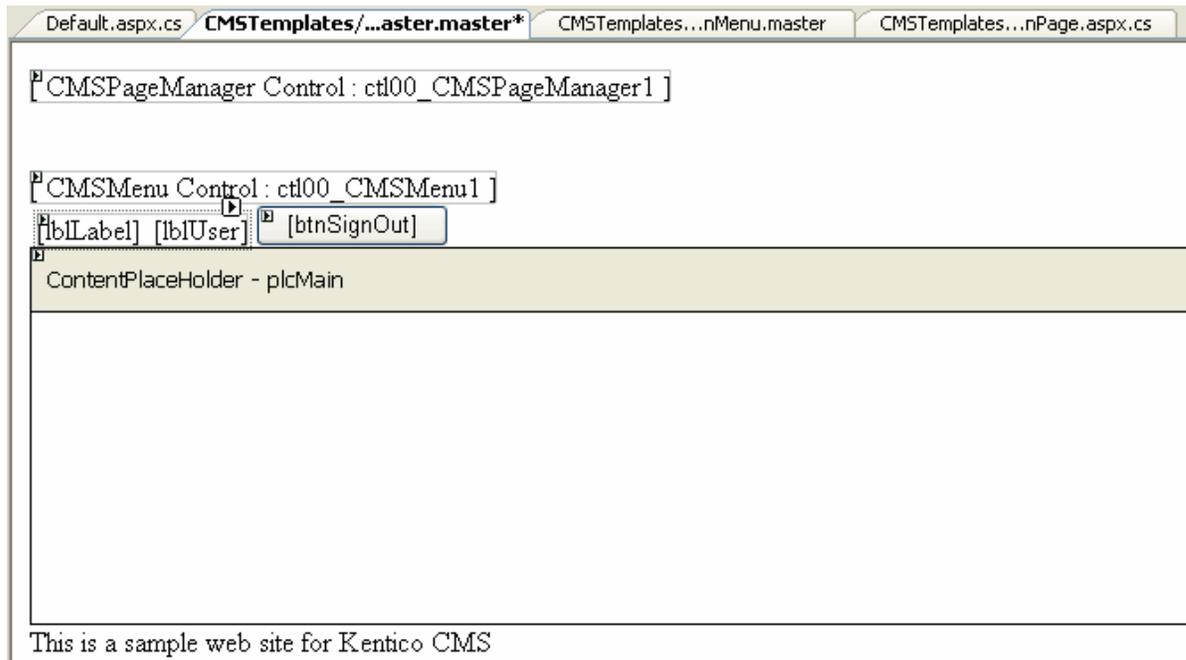
The screenshot shows the 'Settings' page for 'My web site' in the Kentico CMS Site Manager. The left sidebar contains a tree view of settings categories, with 'Security' selected. The main content area displays the 'Security' settings, including:

- Send password e-mails from:   Inherit from global settings
- Check page permissions:   Inherit from global settings
- Website logon page URL:   Inherit from global settings
- Access denied page URL:   Inherit from global settings
- Enable banned IPs:   Inherit from global settings
- Redirect banned IPs to URL:   Inherit from global settings
- Enable flood protection:   Inherit from global settings
- Flood interval:   Inherit from global settings

Buttons for 'Save', 'Reset these settings to default', and 'Export these settings' are visible at the bottom of the settings list.

## Adding the Sign out button

Now we will add the "current user name" and "sign out" controls to our master page. Open the master page **MyMaster.master** in Design mode in Visual Studio and drag and drop the **CMSWebParts\Membership\CurrentUser.ascx** and **SignOutButton.ascx** controls just under the **CMSMenu** control:



Set the following property values of these controls:

### CurrentUser1

- **ShowOnlyWhenAuthenticated:** true
- **CssClass:** CurrentUser

### SignOutButton1

- **ShowOnlyWhenAuthenticated:** true

Now go to **Site Manager -> Development -> CSS Stylesheets** and add the following code to the end of the stylesheet used by your site:

```
.CurrentUser
{
  color: black;
}
```

Save the changes.

Sign out. Click Partners in the main menu. You are redirected to the logon page:

The screenshot shows the logon page for 'Our Business, Inc.'. At the top is an orange header with the company name. Below it is a blue navigation bar with links: Home, News, Services, Search, Partners. The main content area contains two columns of login forms. The left column has fields for 'User name:' and 'Password:', a 'Remember me' checkbox, a 'Log on' button, and a 'Forgotten password' link. The right column has fields for 'First name:', 'Last name:', 'E-mail:', 'Password:', and 'Confirm password:', along with a 'Register' button. At the bottom, a small text line reads 'This is a sample web site for Kentico CMS'.

Now you need either to sign in as administrator or sign up and create a new account. After you sign in successfully, you will see the Partners page content together with **Sign out** button:

The screenshot shows the 'Partners' page after a successful login. The orange header and blue navigation bar are present. Below the navigation bar, it displays 'Current user: administrator (administrator)' and a 'Sign out' button. A small text line below reads 'This is a secured page for partners.'

You have learned how to secure part of the web site so that it's only accessible for registered users.



### Displaying personalized content based on user's permissions

Kentico CMS also allows you to display a personalized content based on user's read permissions. You can e.g. grant gold partners with Read permission for the Gold partners section and then only the gold partners will see the menu item and page content.

You can find more details on personalized content in **Developer's Guide -> Security, permissions and personalization -> Displaying personalized content.**

You have just finished the sample web site.

**Part**



## 8 Further steps

### 8.1 Further steps

This is the end of the Kentico CMS Tutorial. If you need any further details, you will find them in **Kentico CMS Developer's Guide**. It covers also other advanced topics, such as:

- Multi-lingual content
- Multi-site configuration
- Workflow and versioning
- Security administration
- Deployment to the live web site
- Newsletters, BizForms and other modules
- Kentico CMS API and extensibility
- and many other features.

If you cannot find some information, please feel free to contact us at <http://www.kentico.com/Support.aspx>