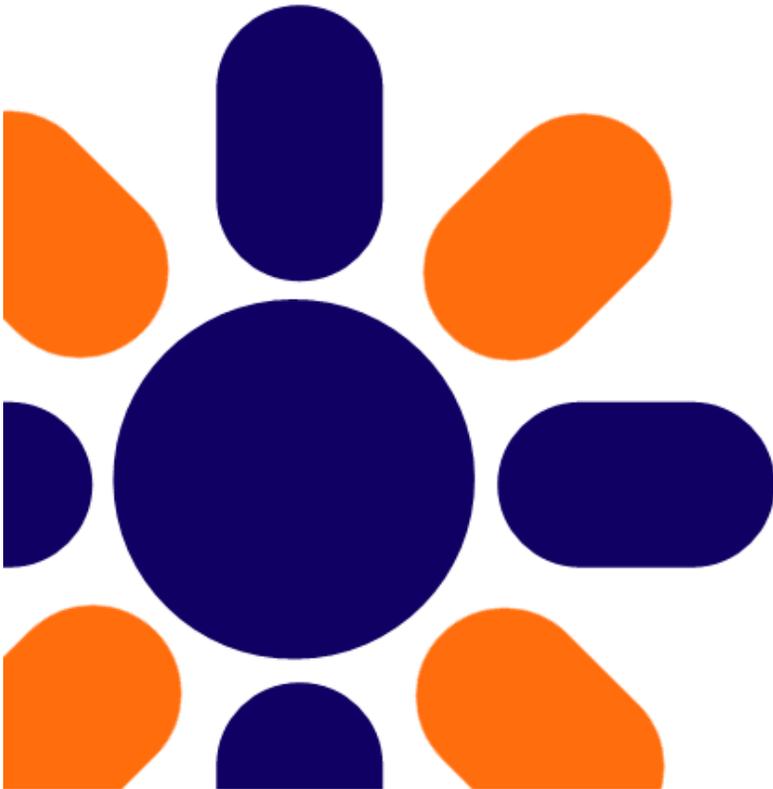


# Kentico CMS Tutorial 4.1



# 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 the portal engine</b>	<b>41</b>
1 Page templates.....	41
2 Overview of web parts.....	43
3 Modifying the home page structure.....	45
4 Creating a new page template.....	49
5 Re-using the page template.....	52
6 Visual inheritance.....	54
7 Configuring visual inheritance.....	56
8 Web part containers.....	60
9 Adding custom code to your web site.....	62
<b>Part VI Developing web parts</b>	<b>65</b>
1 Web part development overview.....	65
2 Creating a simple web part.....	68
3 Web part modifications.....	72
<b>Part VII Managing styles and design</b>	<b>74</b>

---

1	CSS styles .....	74
2	App themes.....	76
3	Menu design.....	77
<b>Part VIII Creating a new site using the portal engine</b>		<b>81</b>
1	Overview .....	81
2	Creating a new web site using a wizard.....	82
3	Creating a CSS stylesheet.....	92
4	Master page.....	94
5	Main menu.....	100
6	Page placeholder.....	101
7	Home page.....	102
8	Container for home page text.....	105
9	News page.....	106
10	Services page.....	110
11	Products page.....	113
	Overview .....	113
	New document type .....	113
	Transformations .....	119
	Page template .....	122
12	Search page.....	126
13	Secured section for partners.....	127
<b>Part IX Further steps</b>		<b>132</b>
1	Further steps.....	132

**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>

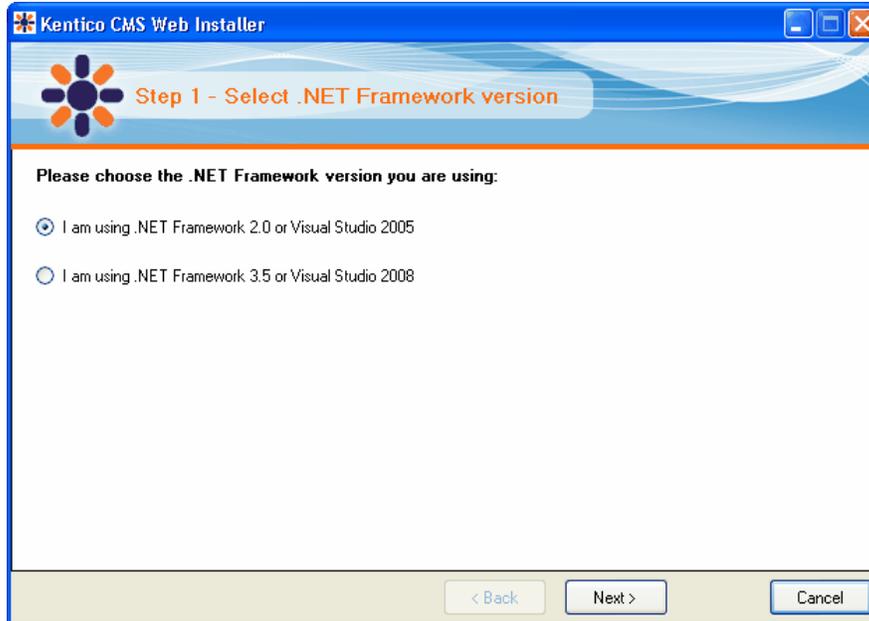
Run KenticoCMS.exe and follow the installation wizard:



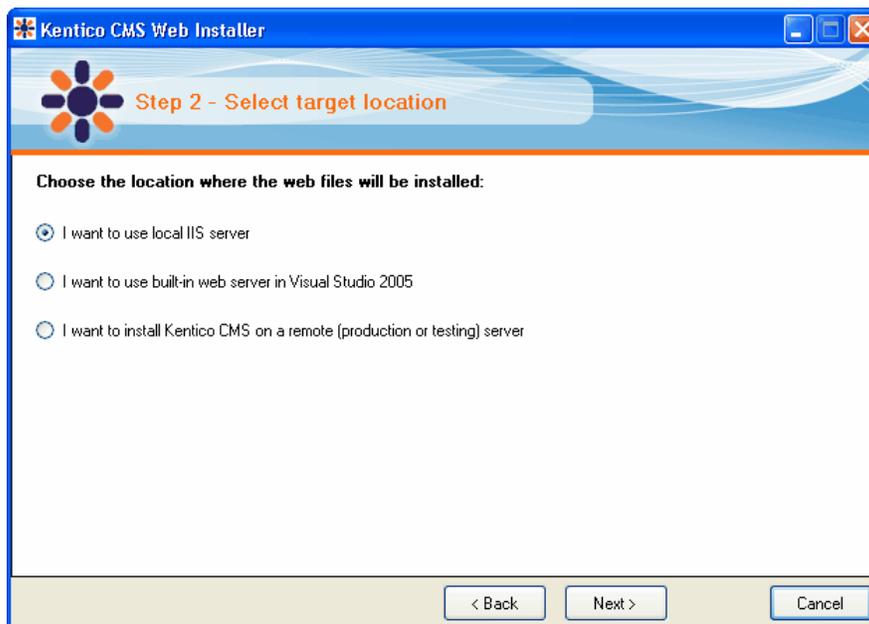
Read and accept the license agreement and click **Next**. Choose the installation location of the binary files and documentation on your disk. After the setup completes the installation, choose to **Launch Kentico CMS Web Installer** and click **Finish**.

## 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**.

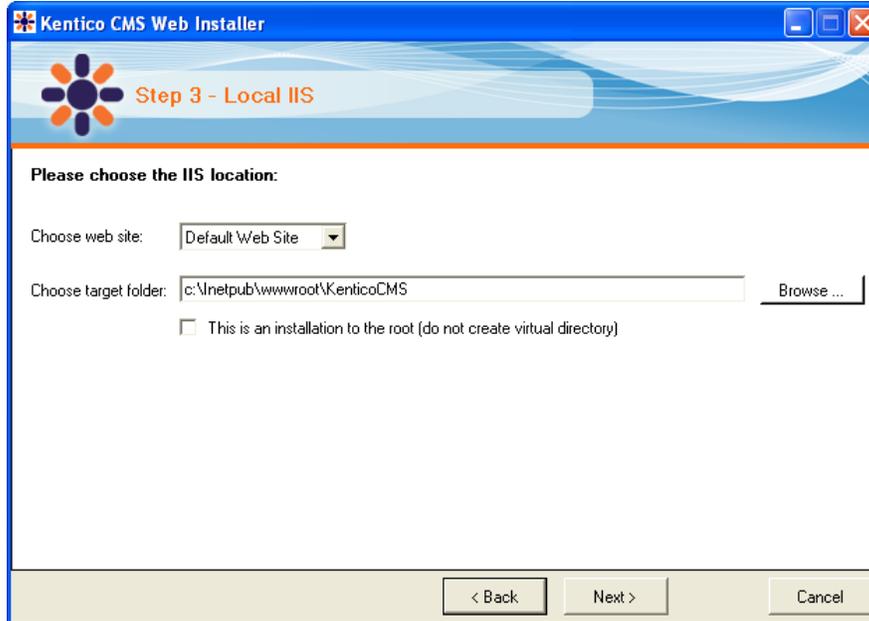


Choose to use either IIS server or Visual Studio 2005 built-in web server (if you do not have IIS installed). Click **Next**.

## 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.



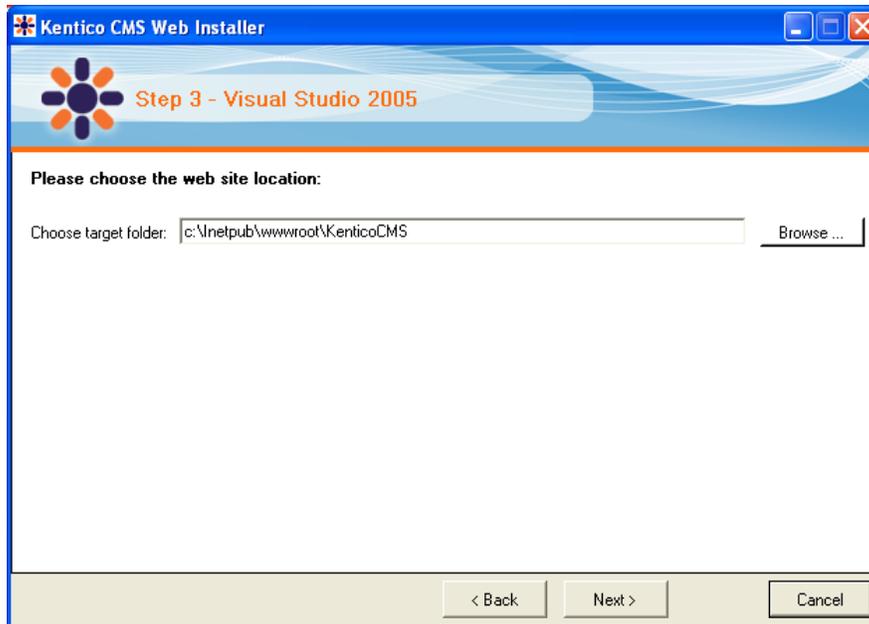
The screenshot shows the 'Kentico CMS Web Installer' window at 'Step 3 - Local IIS'. The window title is 'Kentico CMS Web Installer'. The main content area is titled 'Please choose the IIS location:'. It contains the following elements:

- 'Choose web site:' with a dropdown menu showing 'Default Web Site'.
- 'Choose target folder:' with a text box containing 'c:\inetpub\wwwroot\KenticoCMS' and a 'Browse ...' button.
- An unchecked checkbox labeled 'This is an installation to the root (do not create virtual directory)'.

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**.

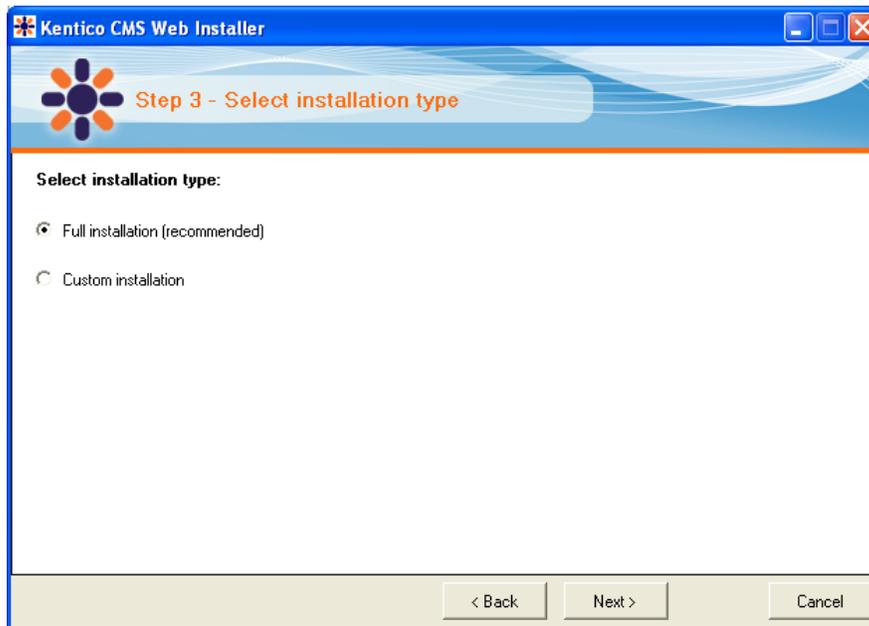


## Type of installation

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

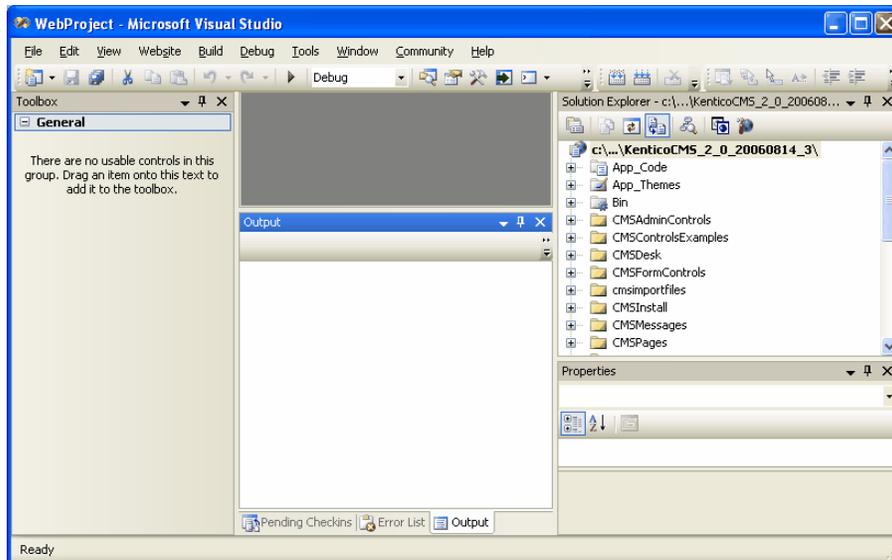
Click **Next** to start the installation process.



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.

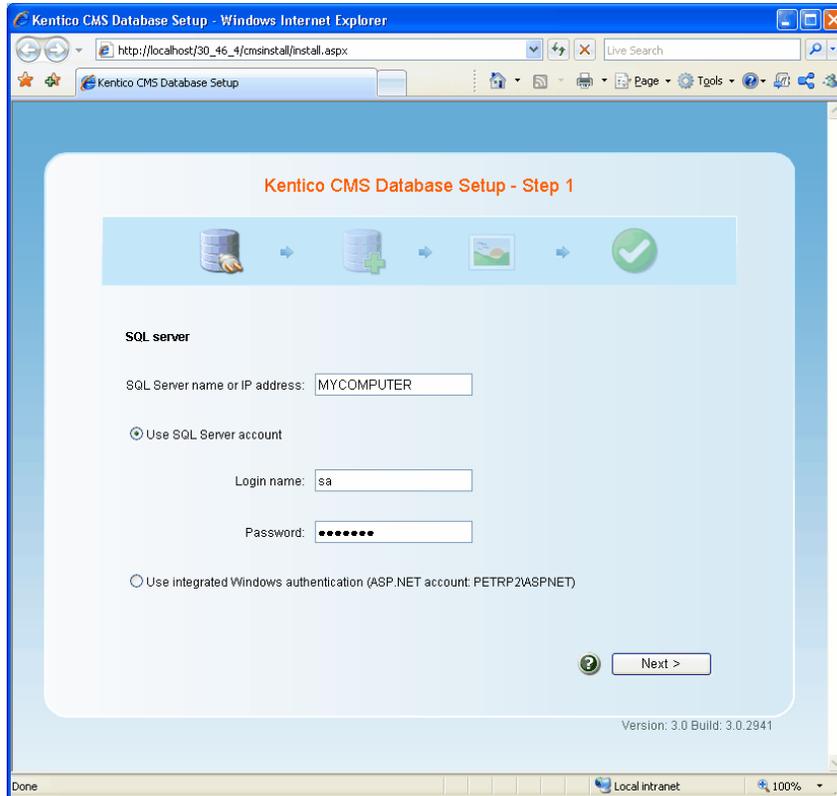


### When you cannot open the web site in Visual Studio 2005

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.



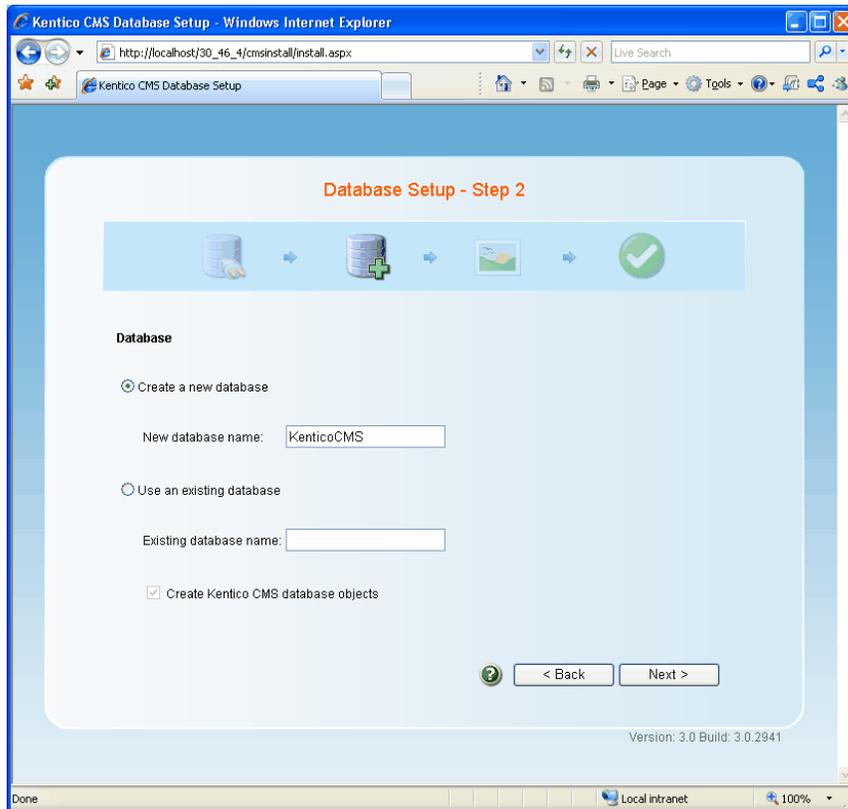
Choose the SQL Server name or IP address. If you're using SQL Server 2005 Express Edition, the default server name is `.\SQLEXPRESS` or `(local)\SQLEXPRESS`.

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

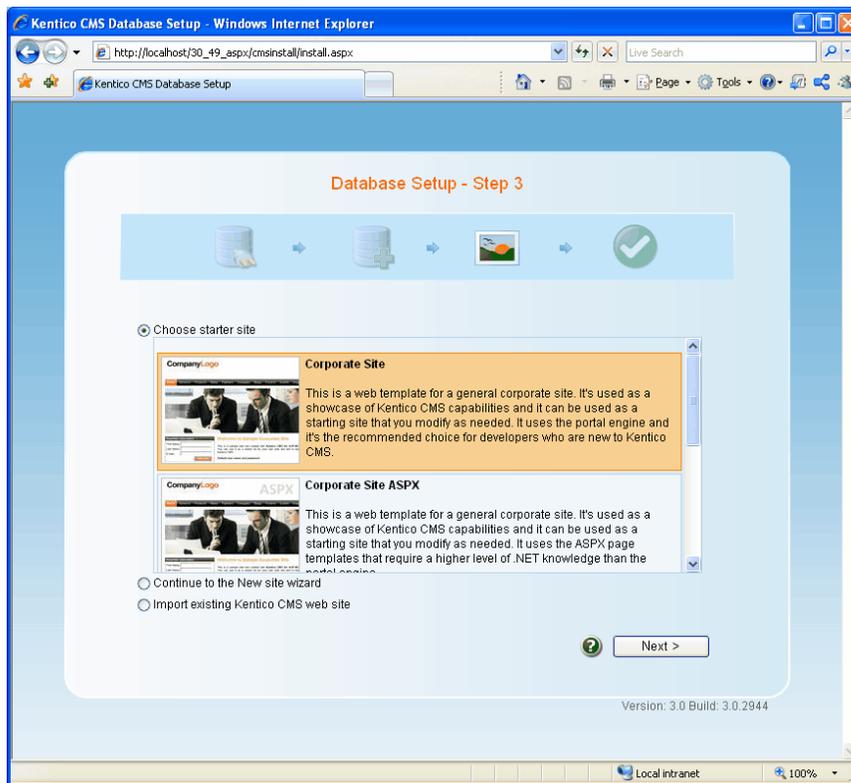
- In case you use **SQL Server account**, you need to enter the user name (such as **sa**) and password.
- In case you use **Windows authentication**, you need to ensure that the ASP.NET account of the name displayed in the brackets has an appropriate login name in your SQL Server.

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

Click **Next**.



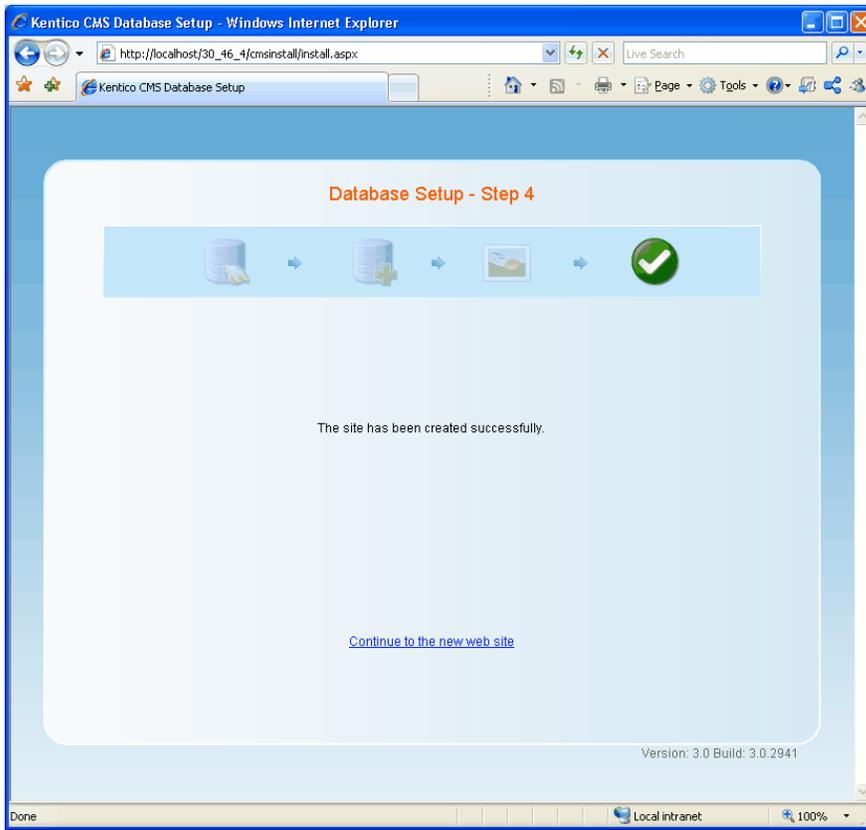
Choose to use an existing database or to create a new one. Click **Next**. If no error occurs, you should see the last step:



You can choose from the following 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 page 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 if 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** 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 Kentico CMS Desk interface. The interface is divided into several sections, each highlighted with a yellow callout box:

- Document actions:** Located at the top left, containing icons for New, Delete, Copy, Move, Up, and Down.
- Main menu:** Located at the top center, containing links for Edit, Preview, Live site, and List.
- View mode selection:** Located at the top right, containing a dropdown menu for Site (Corporate Site) and a Sign Out button.
- Page editing mode:** Located in the middle left, containing tabs for Page, Design, Form, Product, and Properties.
- WYSIWYG editor toolbar:** Located in the middle right, containing a rich set of text and image editing tools.
- Content tree:** Located on the left side, showing a hierarchical view of the site structure (Corporate Site, Home, Services, Products, News, Partners, Company, Blogs, Forums, Events, Images, Wiki, Examples, Special pages, Images).
- Page in editing mode:** The main content area, showing a preview of a page with a header, navigation menu, and content blocks.
- Editable region:** A specific area within the page content, showing a list of services and a text block.

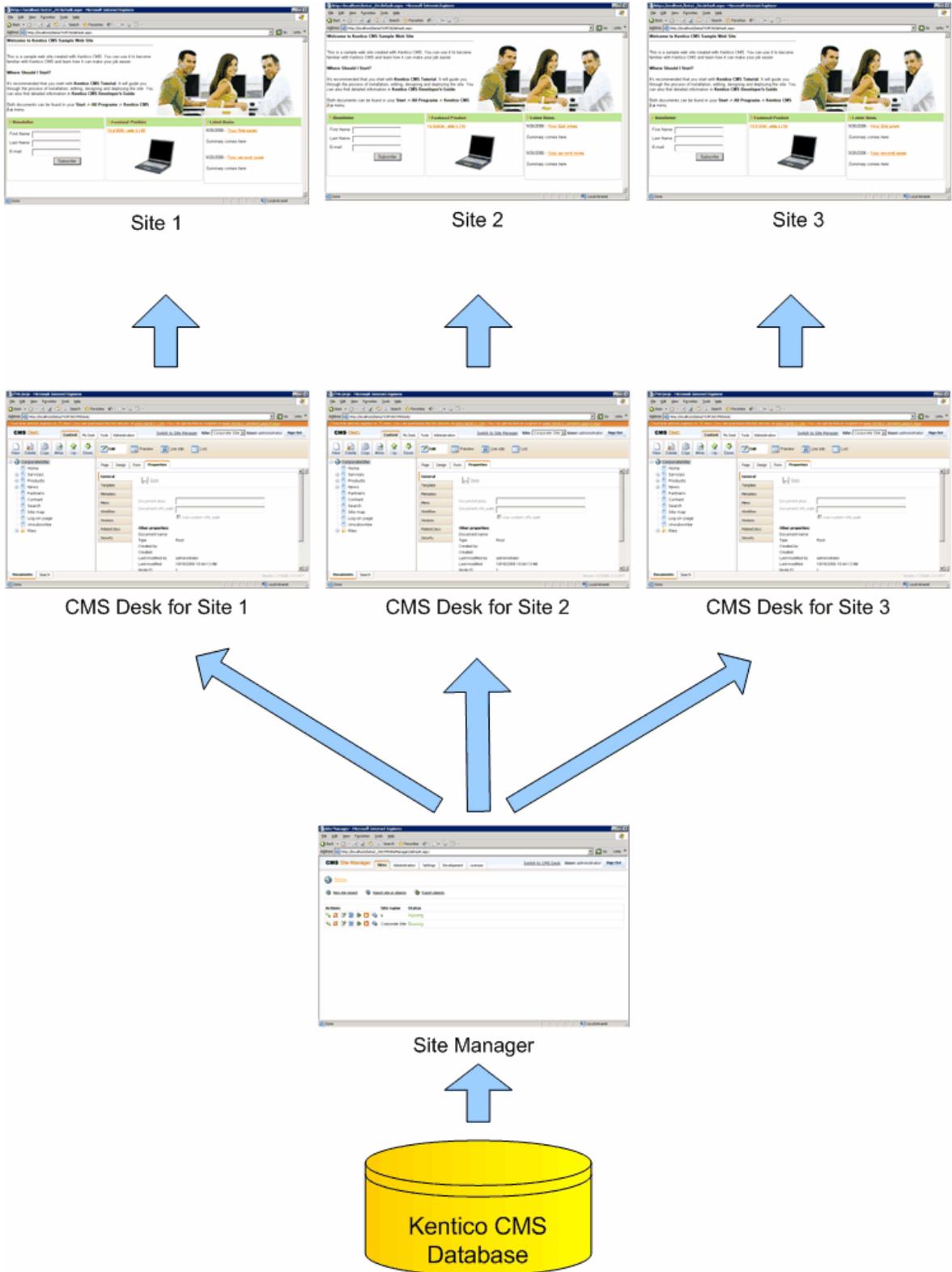
The interface also displays user information (Current user: Global Administrator (administrator)), a shopping cart, and a search bar.

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.

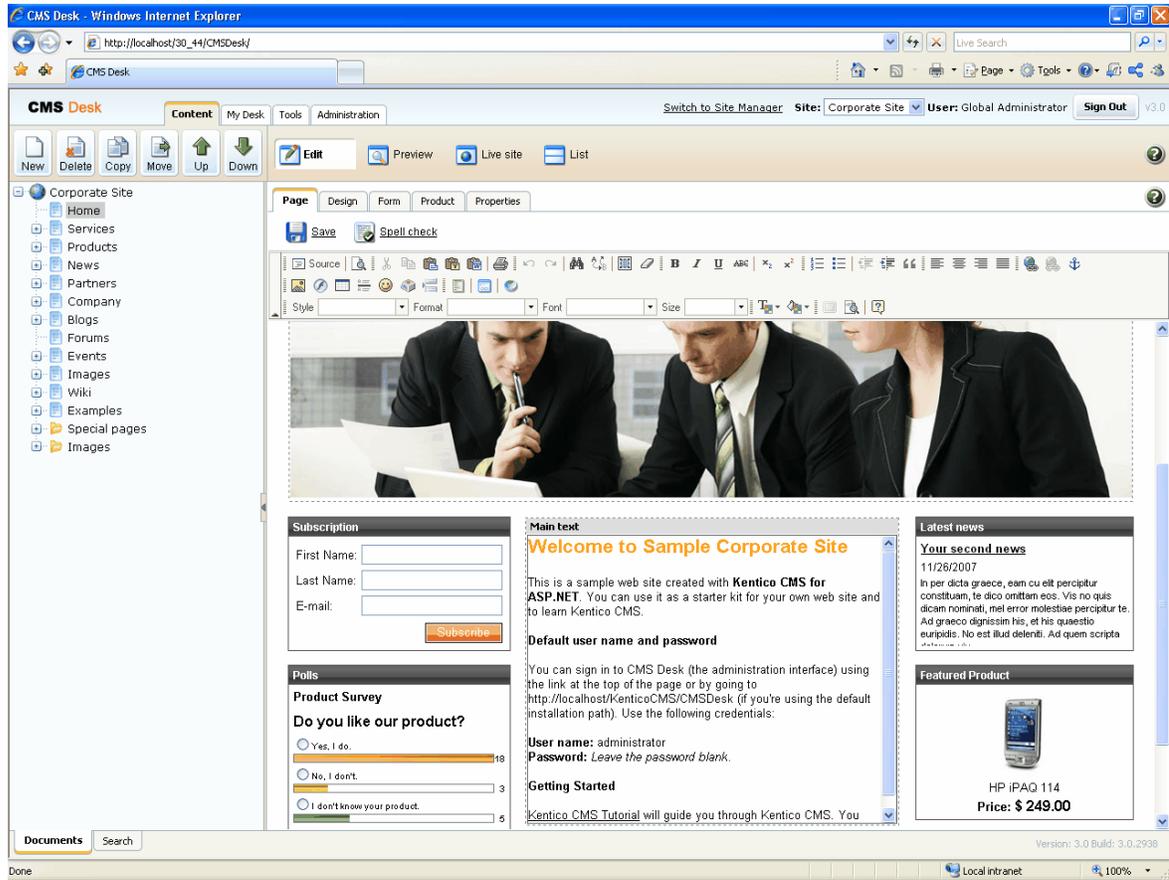
## CMS Desk and Site Manager

**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:

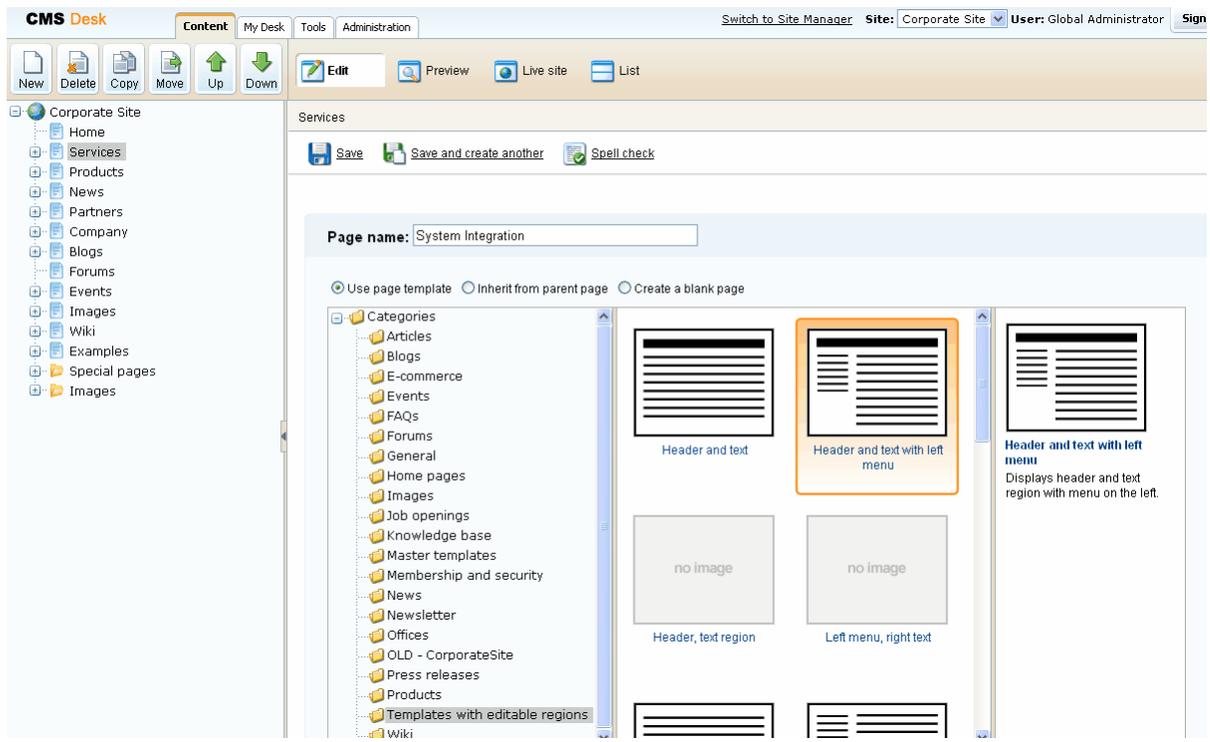
### New document

Please select new document type:

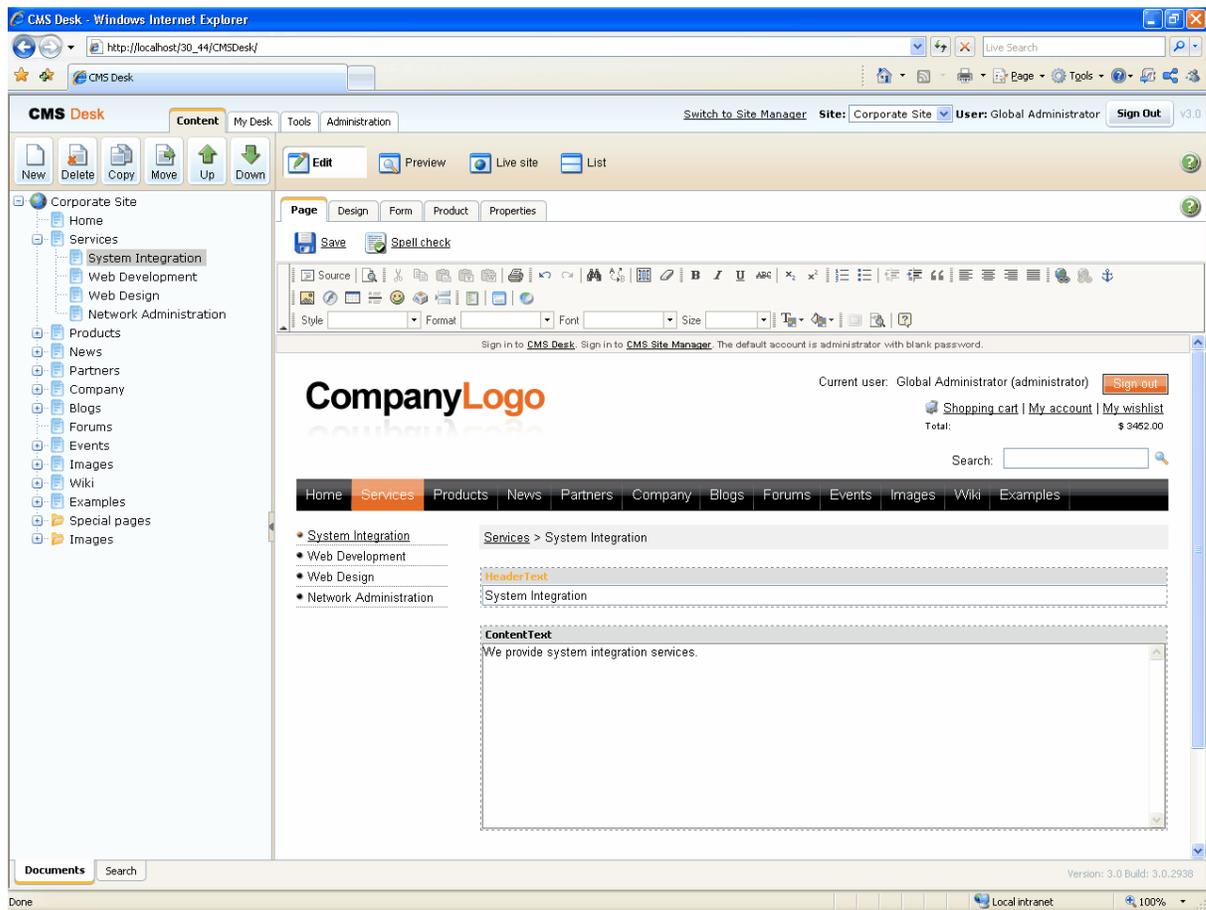
-  [Page \(menu item\)](#)
  -  [Article](#)
  -  [Blog](#)
  -  [Cell phone](#)
  -  [Event](#)
  -  [Event \(booking system\)](#)
  -  [FAQ](#)
  -  [File](#)
  -  [Folder](#)
  -  [Image gallery](#)
  -  [Job opening](#)
  -  [Knowledge base article](#)
  -  [Laptop](#)
  -  [News](#)
  -  [Office](#)
  -  [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 **Templates with editable regions -> Header and text with left menu** template:



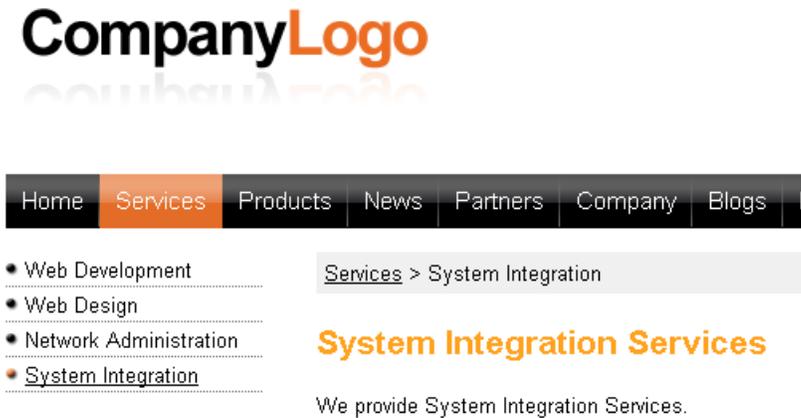
Click **Save** to create the new page. The page is now created in the content tree and you can edit page content on the right. Enter some text in the editable regions and click **Save**.



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:

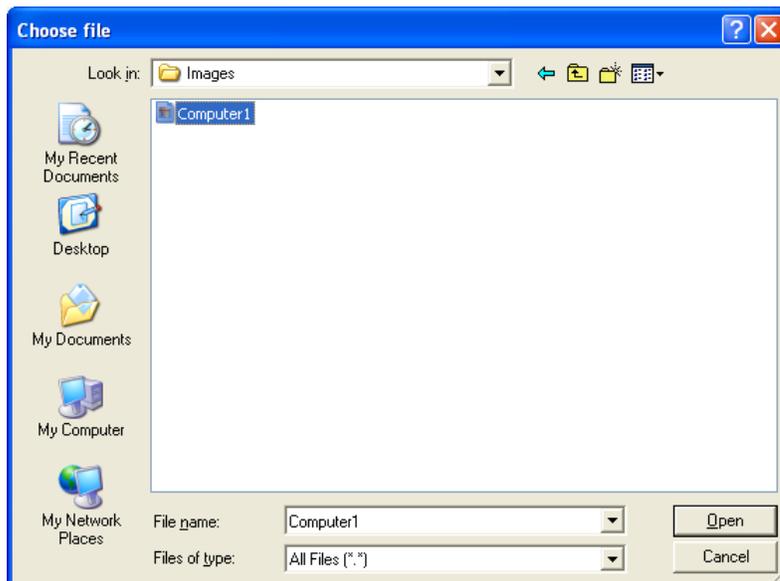


You have learned how to create a new page based on a pre-defined page template.

### 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:

The screenshot displays the Kentico CMS Desk interface. On the left is a tree view of the site structure, with 'System Integration' selected under 'Services'. The main editing area shows a page with a 'CompanyLogo' at the top, a navigation menu where 'Services' is highlighted, and a breadcrumb trail 'Services > System Integration'. Below the breadcrumb are two editable regions: 'HeaderText' containing 'System Integration Services' and 'ContentText' containing 'We provide System Integration Services.' and an image of a computer monitor and tower. The interface includes various toolbars for editing, saving, and previewing.

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

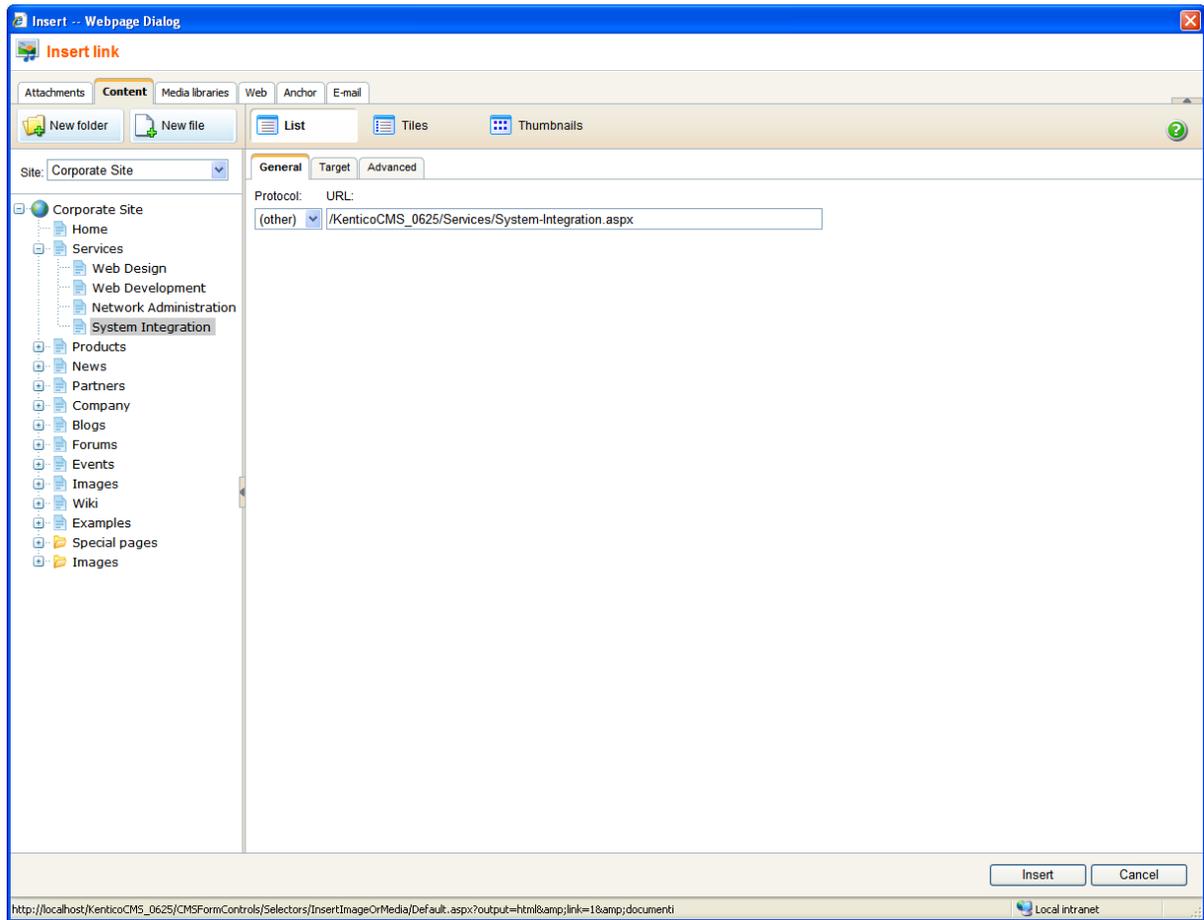
**ContentText**  
We provide professional services in the following areas:

- [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 italicized text: "This is a simple page with left menu and content on the right. You ca. Insert/Edit Link button in the editor toolbar."

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

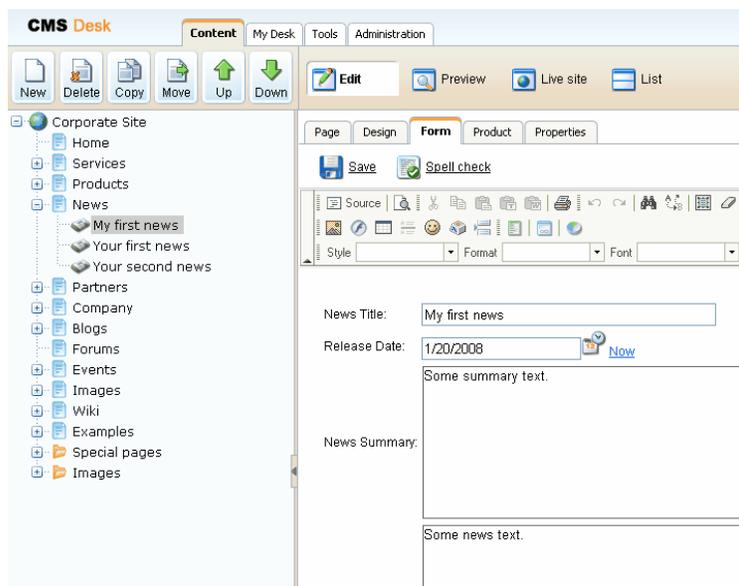
You have learned how to create a link between pages.

## 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.



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.

The screenshot shows the Kentico CMS Design view for a page titled "/Services". At the top, there are tabs for "Page", "Design" (selected), "Form", "Product", and "Properties". Below the tabs, there is a navigation bar with links for "Home", "Services" (selected), "Products", "News", "Partners", "Company", "Blogs", and "For". The main content area is divided into two zones: "zoneLeft" and "zoneContent".

**zoneLeft** contains a "LeftTreeMenu" web part with a tree view showing:

- Web Design
- Network Administration
- System Integration

**zoneContent** contains three web parts: "breadcrumbs", "HeaderText", and "ContentText".

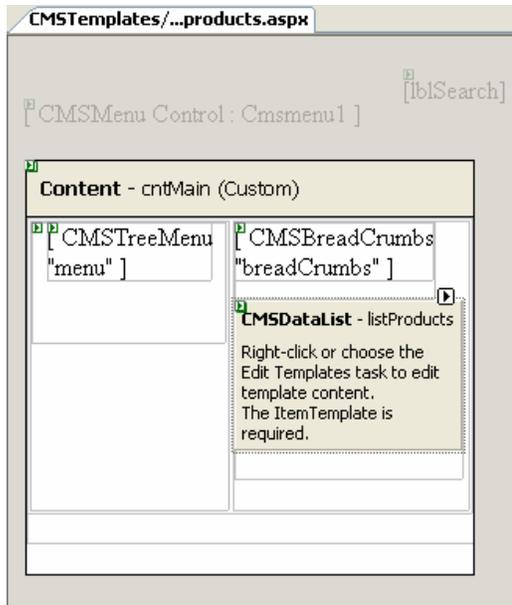
The "HeaderText" web part displays the text "CompanyLogo".

The "ContentText" web part displays the text "Final services in the following areas:" followed by a bulleted list:

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

At the bottom of the ContentText web part, there is a note: "This is a simple page with left menu and content on the Insert/Edit Link button in the editor toolbar."

- **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 portal engine approach. If you want to use the portal engine, please read the Tutorial for portal engine.

**Part**



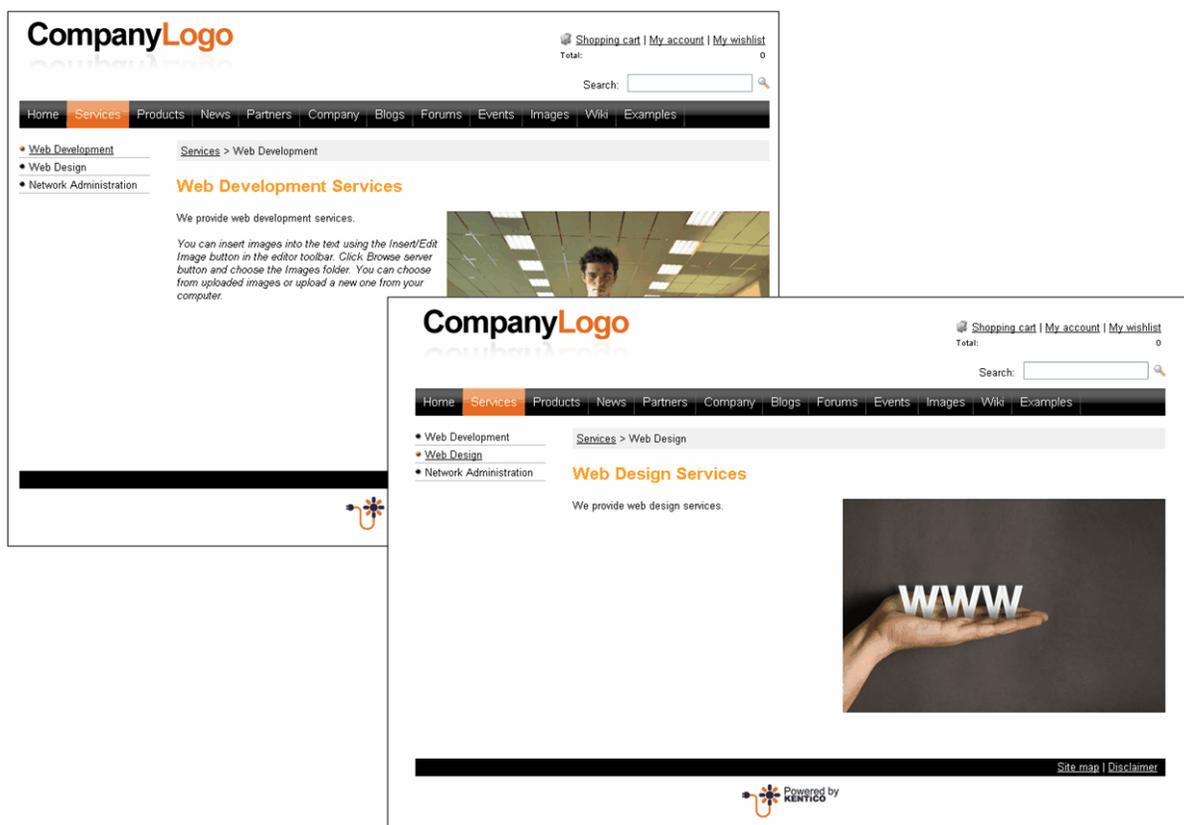
## 5 Creating pages using the portal engine

### 5.1 Page templates

Kentico CMS portal engine provides a highly flexible and extensible framework that allows you to build complex dynamic web sites using built-in and custom web parts. This chapter explains how the portal engine works.

#### What is a page template?

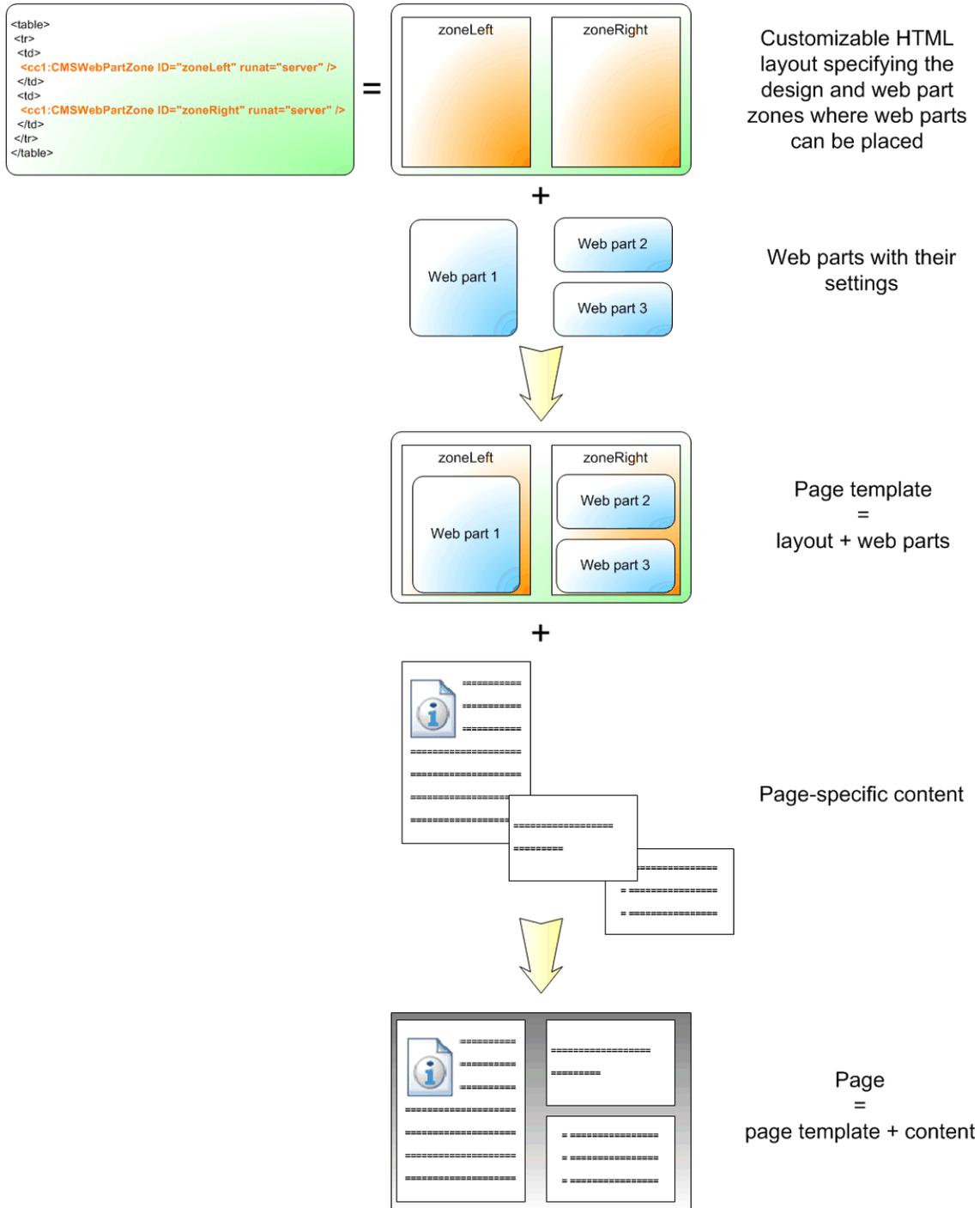
Every web page is based on some page template. The page template can be specific for a single page (so called "ad hoc" page template) or it can be re-used for several pages. The following picture shows an example of two pages that use the same page template:



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 a page template consist of?

The page template is a combination of page layout (HTML code) and web parts. The following figure illustrates the structure of the page template and how it's used to display a page:



As you can see, the **page layout** is a simple, fully customizable HTML code that defines the design of the page and it contains web part zones. The **web part zones** can contain web parts. The **web parts** display page content or provide some functionality, such as input form. The layout and the web parts

together define the **page template**.

When you add **page specific content** to the page template, you get the final **page**.

## 5.2 Overview of web parts

Now you will learn how to modify the page template. Click the **Home** page and switch to the **Design** tab. You will see the structure of the home page template. The main logo and main menu are inherited from the root page. The "active" part marked with green box contains the yellow web part zones and blue web parts:



The page is based on the page template *Home page with top image and three columns*. The **page template** consists of **web parts** that are placed inside **web part zones**. The web part zones are organized in the page layout (HTML layout) that specifies the placement of the zones within the page.

When you click the **Edit layout** button  in the green header of the active page template, you can edit the page layout:

The screenshot shows the 'Page layout' editor interface. At the top, there are tabs for 'Page', 'Design', 'Form', 'Product', and 'Properties'. A 'Display web part content' checkbox is checked. The current user is identified as 'Global Administrator (administrator)'. Below the tabs, there are 'Save' and 'Check out to file' buttons. A message prompts the user to check out the layout to a specific file path to edit it externally. Below this, there is a text box containing ASPX code for a three-column layout. The code defines three zones: 'zoneTop', 'zoneLeft', and 'zoneRight', each using the `<cc1:CMSWebPartZone ID="zoneName" runat="server" />` control. The layout is styled with CSS floats and a clear:both rule.

```

<div class="homeTopImage3cols">
  <!-- Top image -->
  <div class="zoneTop">
    <cc1:CMSWebPartZone ID="zoneTop" runat="server" />
  </div>
  <!-- Left column -->
  <div class="zoneLeft" style="float: left;">
    <cc1:CMSWebPartZone ID="zoneLeft" runat="server" />
  </div>
  <!-- Center column -->
  <div class="zoneCenter" style="float: left;">
    <cc1:CMSWebPartZone ID="zoneCenter" runat="server" />
  </div>
  <!-- Right column -->
  <div class="zoneRight" style="float: right;">
    <cc1:CMSWebPartZone ID="zoneRight" runat="server" />
  </div>
  <div style="clear:both;" />
</div>

```

As you can see, the page layout is a simple HTML (or rather ASPX) code that contains controls called **CMSWebPartZone**. Each of them must have a unique ID. You can also see that you can use any HTML code for page formatting - it's up to you whether you use tables or CSS layout.

### Editing the layout using an external editor

You can check out the file to the disk using the Check out to file button, edit it using Visual Studio or some other editor and then check it in.



#### Access control

The document can be opened in the **Design** tab only by user who is marked as a **global administrator** or is granted with **Design web site** permission for the **CMS Content** module. It cannot be accessed by content editors.

## 5.3 Modifying the home page structure

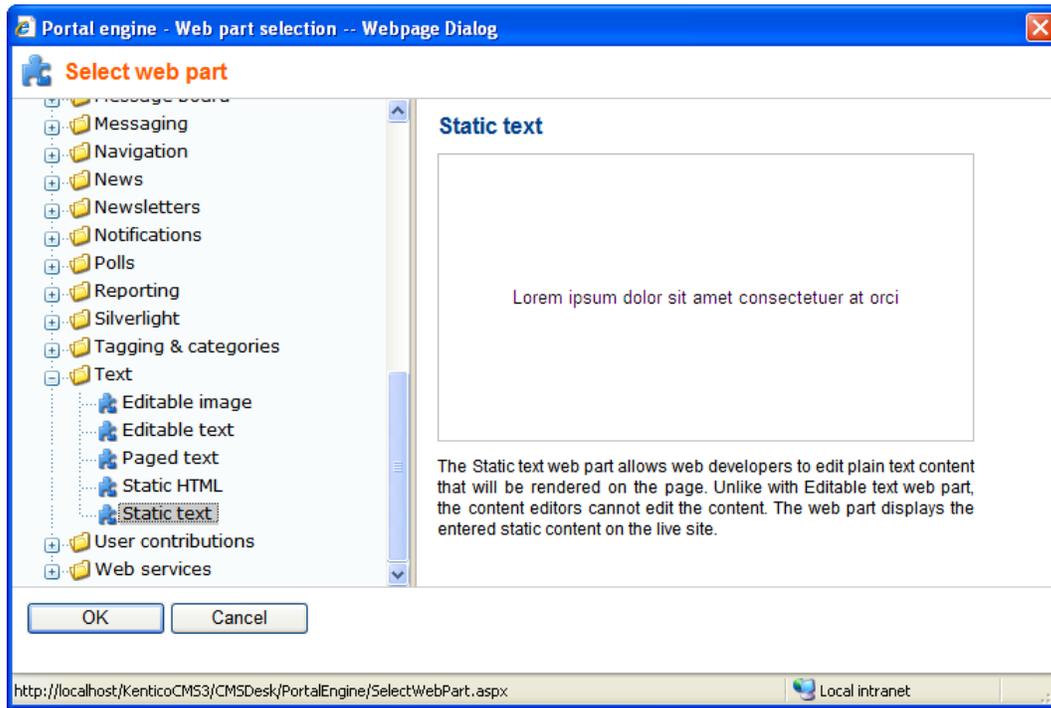
Now you will learn how you can modify the home page structure. We will replace the polls on the left with a static text.

Click **Home** in the content tree and click the **Design** tab. Right-click the **Poll** web part and click **remove** and confirm the action.

The screenshot shows the CMS Desk interface. The top navigation bar includes 'Content', 'My Desk', 'Tools', and 'Administration'. Below this is a toolbar with icons for 'New', 'Delete', 'Copy', 'Move', 'Up', 'Down', 'Edit', 'Preview', 'Live site', and 'List'. The left sidebar shows a content tree for 'Corporate Site' with 'Home' selected. The main workspace is in 'Design' mode, showing a page layout with a header image of a man thinking. Below the image are two zones: 'zoneLeft' and 'zoneCenter'. The 'zoneLeft' zone contains a 'NewsletterSubscription' web part with input fields for 'First Name', 'Last Name', and 'E-mail', and a 'Subscribe' button. Below it is a 'Poll' web part with a 'Vote' button and a context menu open over it. The context menu options are 'Configure', 'Up...', 'Down...', 'Move to zone ...', 'Duplicate web part', and 'Remove'. The 'zoneCenter' zone contains a 'MainContentText' web part with sample text and login information.

Now click the **Add web part** (+) icon in the **zoneLeft** zone.

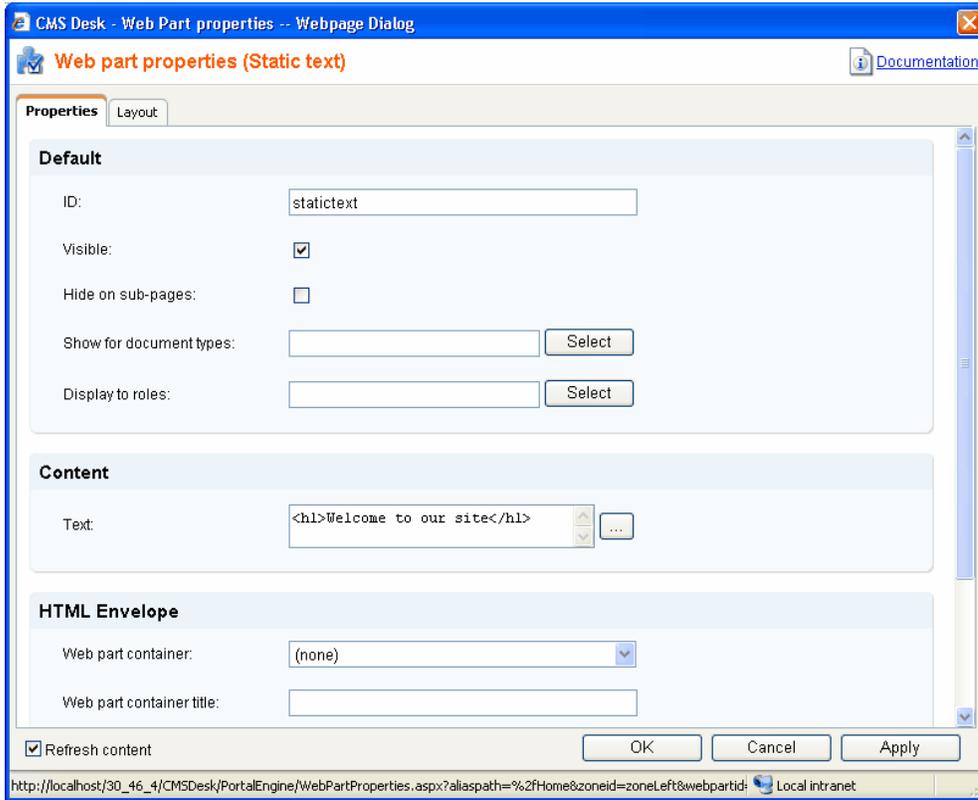
The **Select web part** dialog opens. Choose web part **Text** -> **Static text** and click **OK**.



The web part is added to the web part zone and the **Web part properties** dialog opens. Enter the following text in the Text property:

```
<h1>Welcome to our site</h1>
```

The dialog now looks like this:



Click **OK** to save changes. Click **Live site** in the main toolbar and see the new design of the home page:



**Subscription**

First Name:

Last Name:

E-mail:

## Welcome to our site

## Welcome to Sample Corporate Site

This is a sample web site created with **Kentico CMS for ASP.NET**. You can use it as a starter kit for your own web site and to learn Kentico CMS.

### Default user name and password

You can sign in to CMS Desk (the administration interface) using the link at the top of the page or by going to <http://localhost/KenticoCMS/CMSDesk> (if you're using the default installation path). Use the following credentials:

**User name:** administrator

**Password:** *Leave the password blank.*

### Getting Started

[Kentico CMS Tutorial](#) will guide you through Kentico CMS. You can also get help at <http://www.kentico.com/Support.aspx>.

Now when you click **Edit** in the main toolbar, the home page is displayed in the editing mode, but the *Welcome to our site* text cannot be edited, since it is displayed using the **Static text** web part that doesn't enable content editors to edit the text (the text can be edited only by the web developer). If you want to create an editable region, you would need to add the **Editable text** web part.

You have learned how to modify the structure of an existing page template and how to work with web parts.



**Tip: Testing the changes without switching to the view mode**

You can quickly test the changes you make to the web part values by opening/refreshing the web site in a new browser window. However, you will need to use a **different browser** (e.g. IE for design and FireFox for testing) since the view mode is stored in cookies (if you use different browsers, they use different cookies).

It's also useful to use tabbed browsing support in IE 7.0 and Firefox for easy switching between the **CMS Desk** and **Site Manager** interface.

## 5.4 Creating a new page template

Now you will learn how to create a new page from scratch without using a pre-defined page template. We will create a new **About Us** page with two columns that will contain editable regions.

Click the root in the content tree and click **New**. Choose to create a new **Page (menu item)**.

Enter the page name **About Us** and choose to **create a blank page** using the **Two columns** layout. Make sure that the **Copy this layout to my page template** box is checked - it will ensure that your page template can use a modified version of the pre-defined layout without modifying other pages.

Corporate Site

Save Save and create another Spell check

Page name:

Use page template  Inherit from parent page  Create a blank page

Blank master page layout  
Corporate Site home page layout  
Corporate Site Master Page with Top Menu  
Grid 2x2 cells  
Grid 2x2 cells (CSS)  
Grid 3x2 cells  
Grid 3x2 cells (CSS)  
Simple  
Three columns  
Three columns (CSS)  
Top row, three columns, bottom row  
Top row, two columns, bottom row  
**Two columns**  
Two columns - 20/80  
Two columns - 20/80 (CSS)  
Two columns - 30/70  
Two columns - 30/70 (CSS)  
Two columns - 70/30  
Two columns - 70/30 (CSS)  
Two columns - 80/20  
Two columns - 80/20 (CSS)  
Two columns (CSS)  
Two columns, three columns  
Two columns, three columns (CSS)

no image

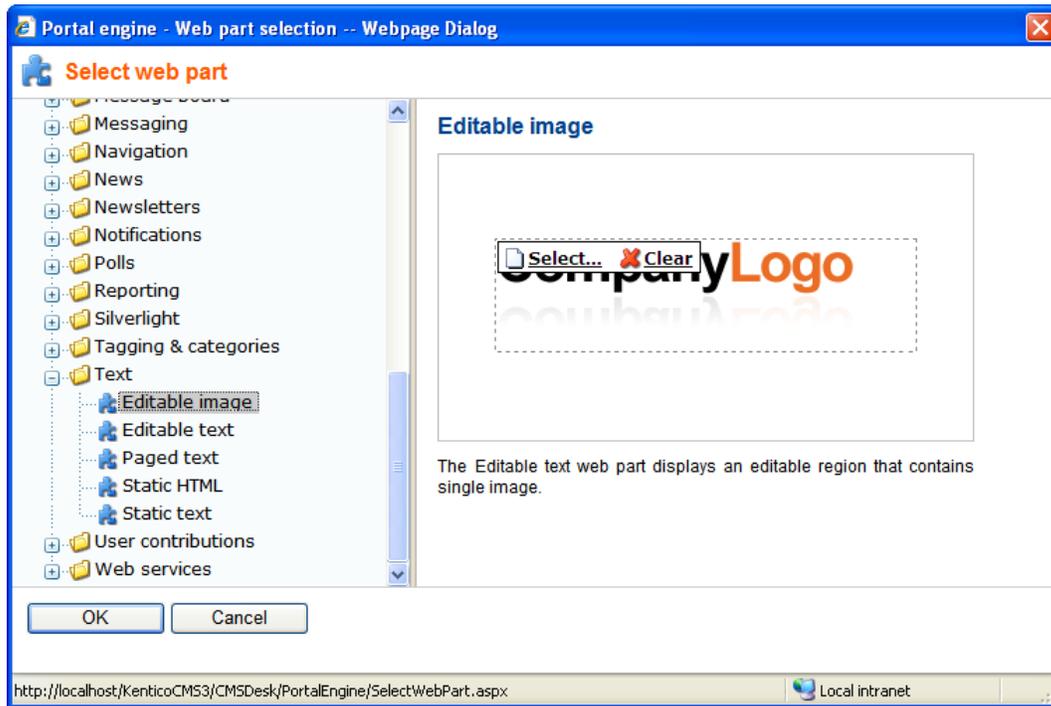
**Two columns**  
Two columns with web part zones.

Copy this layout to my page template

Click **Save** to create the new page. Use the **Down** () button to move the page to the bottom of the content tree.

The page cannot be edited now since it doesn't contain any editable regions. We will add an editable image region on the left and two editable regions on the right. Switch to the **Design** tab and click the **Add web part** () icon in the **zoneLeft** web part zone.

Choose the **Text -> Editable image** web part and click **OK**.



Set the following values:

- **Web part control ID:** LeftImage
- **Image title:** Left image

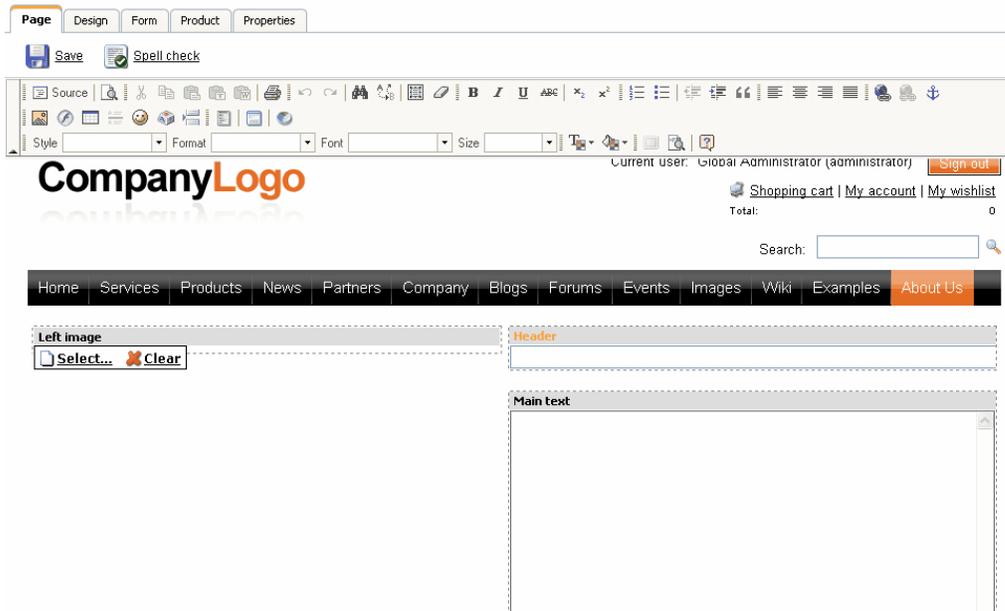
Add another web part to the **zoneRight** zone: Choose the **Text -> Editable text** web part and click **OK**. Set the following property values:

- **Web part control ID:** HeaderText
- **Editable region title:** Header
- **Editable region type:** Text box
- **Content before:** <h1>
- **Content after:** </h1>

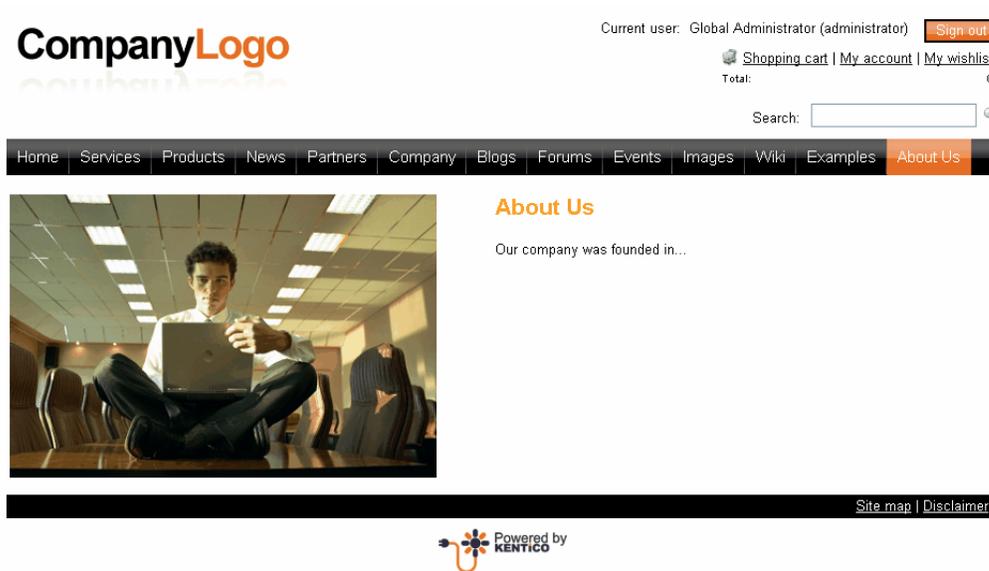
Click **OK**. Add another **Editable text** web part to the right web part zone and set the following properties:

- **Web part control ID:** MainText
- **Editable region title:** Main text
- **Editable region type:** HTML Editor
- **Editable region width:** (leave empty)
- **Editable region height:** 400

Switch to the **Page** tab and you will see a page with editable regions like this:



Now you can enter some content and choose some image using the **Select** button. Click **Save** and view the page in the **Live site** mode:



Congratulations, you have just created your first page template using web parts.

## 5.5 Re-using the page template

In the previous chapter, you have learned how to create a new page with its own, page-specific page template. This type of page template is called an **ad-hoc page template**.

Now, you will learn how to convert the page-specific page template into a **re-usable page template** that can be used for several pages with the same layout, but different content.

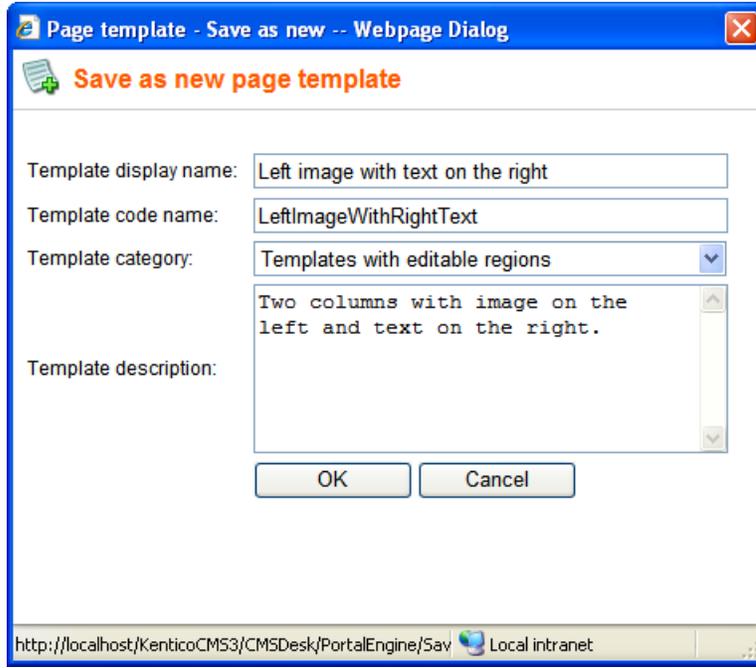
Go to **CMS Desk -> Content**. Click the new **About Us** page and switch to the **Properties** tab. Click **Template**. Now you can see a dialog like this:

The screenshot shows the 'Properties' tab of the Kentico CMS interface. On the left is a vertical navigation menu with the following items: General, URLs, **Template** (highlighted), Metadata, Categories, Menu, Workflow, Versions, Related docs, Linked docs, Security, and Attachments. The main content area is titled 'Properties' and contains the following elements:

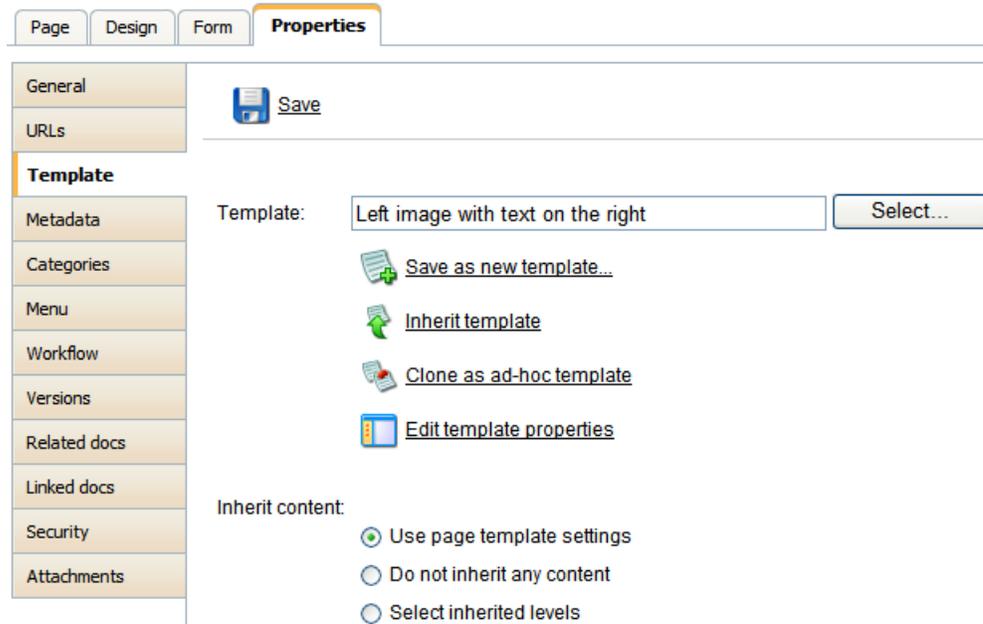
- A 'Save' button with a floppy disk icon.
- A 'Template:' label followed by a text input field containing 'Ad-hoc: About us' and a 'Select...' button.
- Three action buttons with icons:
  - 'Save as new template...' with a document and plus icon.
  - 'Inherit template' with a document and arrow icon.
  - 'Edit template properties' with a document and gear icon.
- An 'Inherit content:' label followed by three radio button options:
  - Use page template settings
  - Do not inherit any content
  - Select inherited levels

Click **Save as new template** and enter the following values:

- **Template display name:** Left image with text on the right
- **Template code name:** LeftImageWithRightText
- **Template category:** Templates with editable regions
- **Template description:** Two columns with image on the left and text on the right.



Click **OK** and click **Save** on the **Template** dialog. The dialog now looks like this:



You have just created a new re-usable page template. If you try to create a new page now, the page template will be offered and you can create another page with the same structure as the About Us page.



### Modifying shared page templates

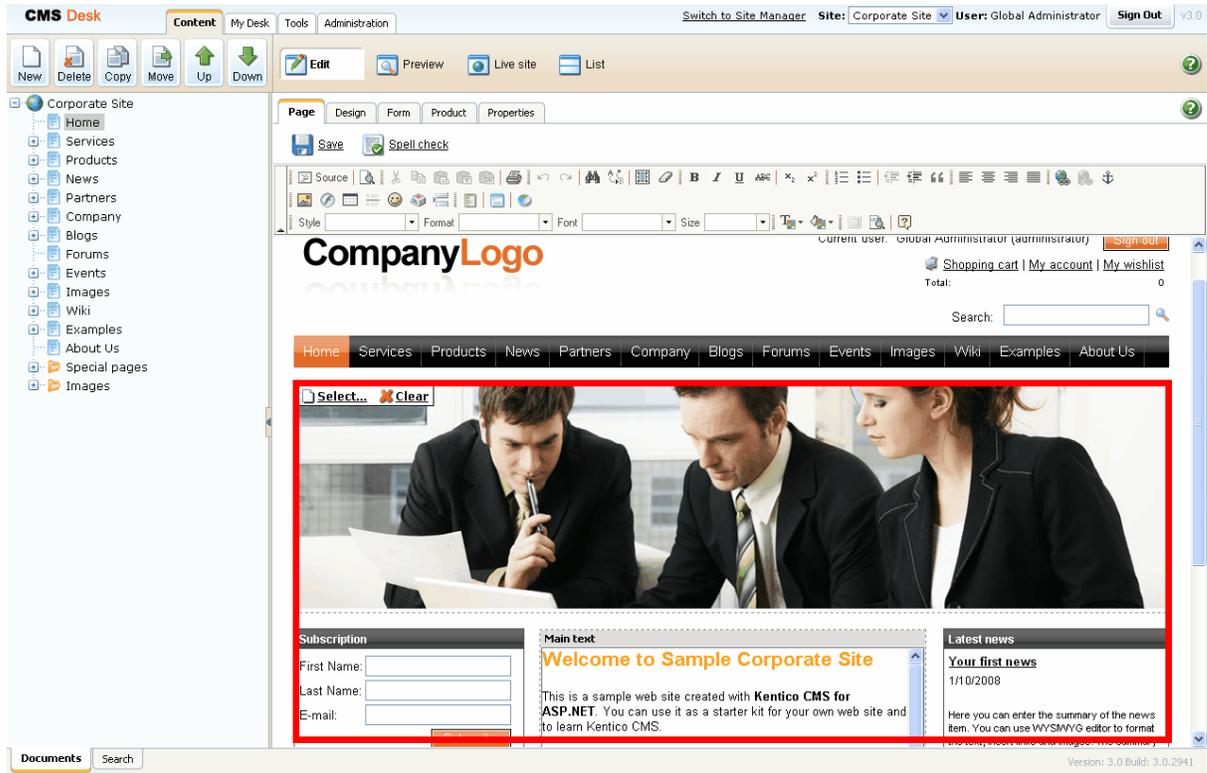
Please note that when you modify the re-usable page template, the changes will affect all pages that use this page template!

## 5.6 Visual inheritance

The content of sub-pages is displayed using **page placeholder** web parts. The impact of this approach is that the sub-page content is "nested" inside the content of upper pages. The following picture shows the root (master page) without nested content:

The screenshot displays the Kentico CMS Desk interface. The top navigation bar includes "CMS Desk", "Content", "My Desk", "Tools", and "Administration". The user is logged in as "Global Administrator" on the "Corporate Site". The interface shows a sidebar with a tree view of the site structure, including "Home", "Services", "Products", "News", "Partners", "Company", "Blogs", "Forums", "Events", "Images", "Wiki", "Examples", "About Us", "Special pages", and "Images". The main content area is in "Edit" mode, showing a "Page" design view with a "Company Logo" placeholder. The footer contains a navigation menu with links for "Home", "Services", "Products", "News", "Partners", "Company", "Blogs", "Forums", "Events", "Images", "Wiki", "Examples", and "About Us". The version number "3.0.2941" is visible in the bottom right corner.

... and here's the home page nested inside the master page:



Please note how the pages are structured in the content tree: the root (master page) is parent of the Home page which is loaded inside the master page. You can do the nesting recursively at any level of the content tree. The important component of page nesting is the **Page placeholder** web part. This web part must be placed in the master page and it specifies the place where the underlying page will be inserted. The Page placeholder in the Corporate Site root loads the sub-pages between the main menu and footer:

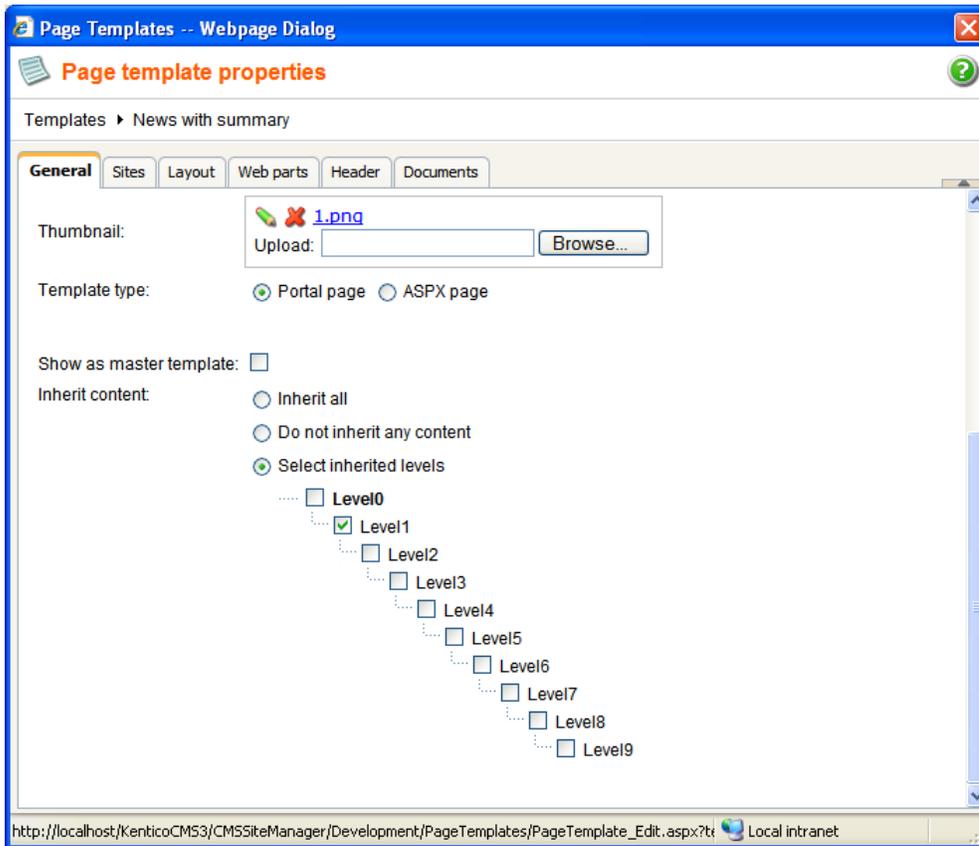


## 5.7 Configuring visual inheritance

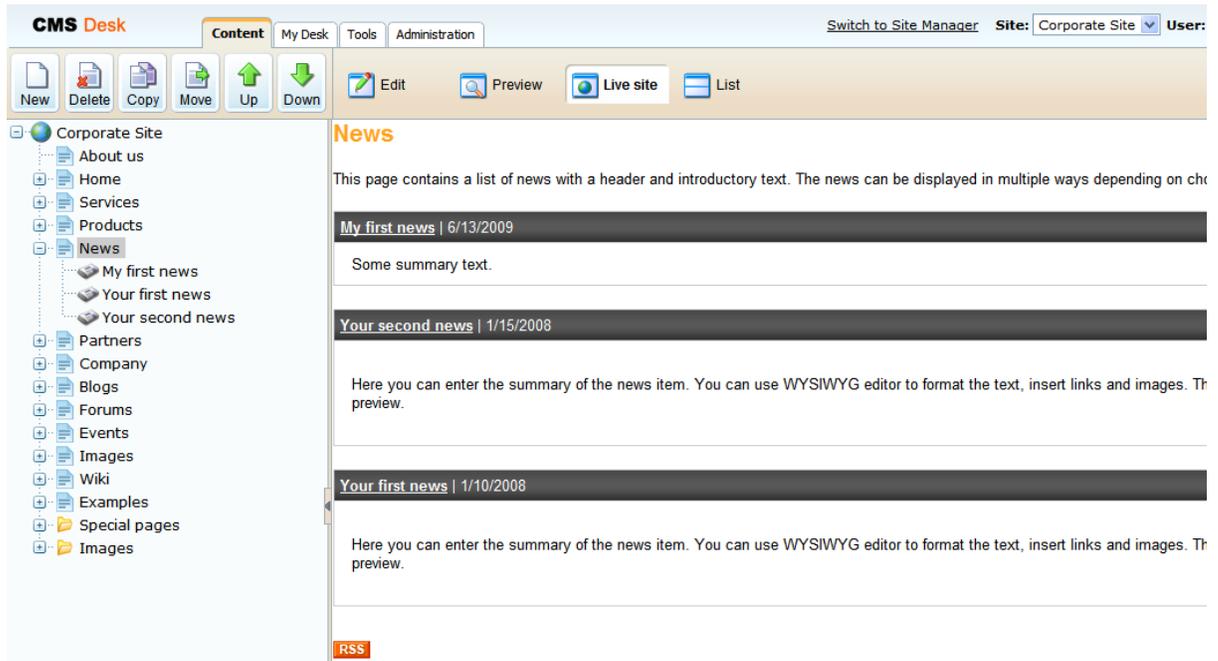
In some cases, you may want to hide some parts of the parent page. There are several ways how to achieve that:

### Using the "Inherit content" property of the page template

Click the **/News** page and click **Properties -> Template -> Edit template properties**. Now you can set the **Inherit content** value to **Selected inherited levels** and check only the **Level 1** box. It means that only the content from first level of content hierarchy will be displayed and the master page (root) is not inherited. Click **OK** to save the changes.

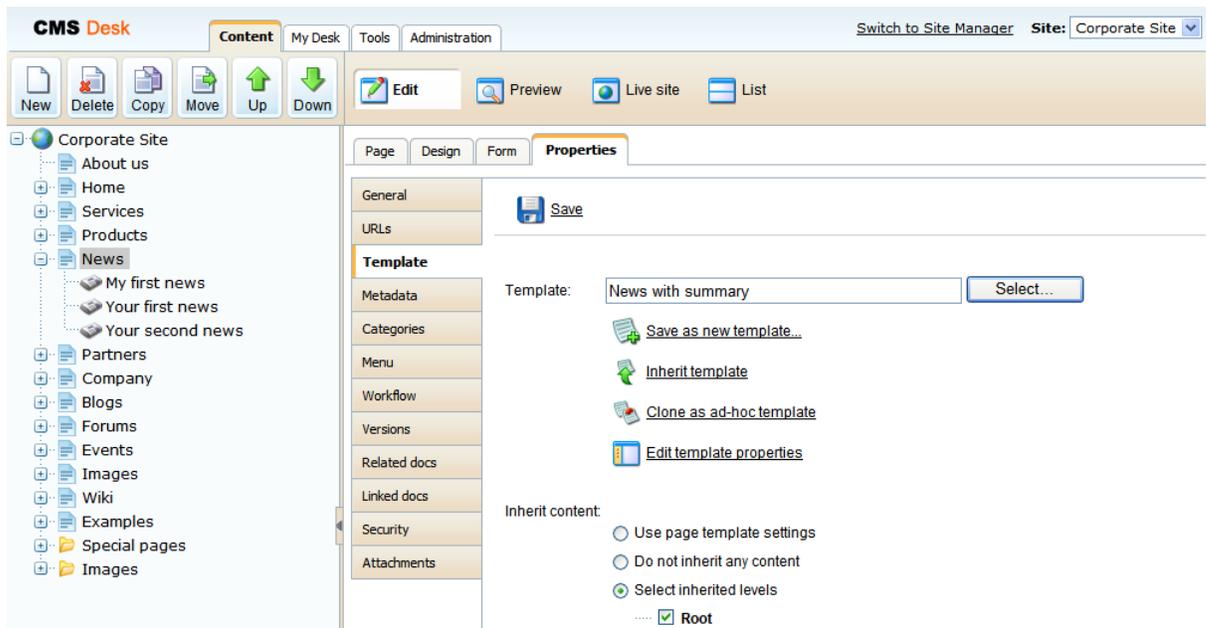


The page will look like this:



Set the value back to inheritance from **Level0** only and click **Save**.

Similarly, you can set the content inheritance on **page level** using the **Properties -> Template** dialog. The content inheritance settings you configure for the page override the page template settings:

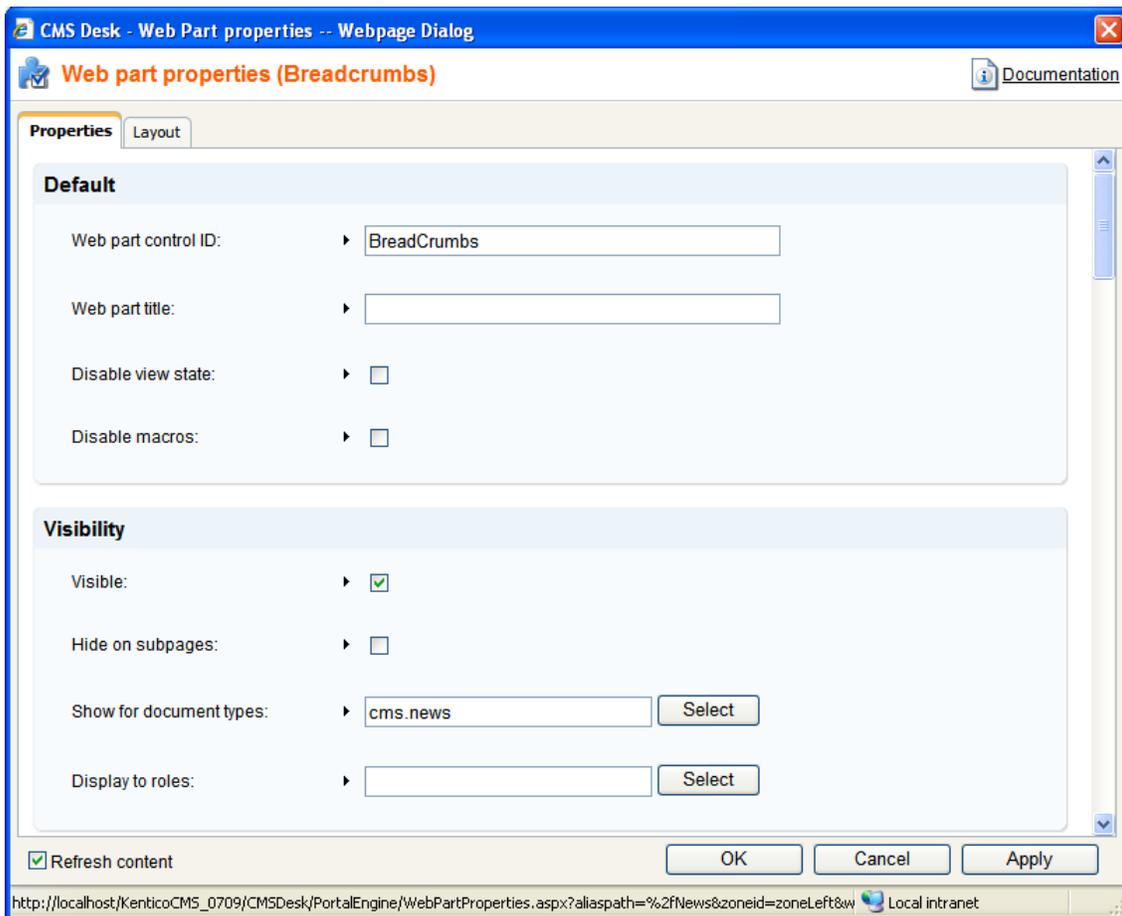


## Using the "Hide on sub-pages" web part property

Every web part has the property called **Hide on sub-pages**. When you check the box, the web part is hidden if some sub-page is displayed.

## Using the "Show for document types" web part property

The **Show for document types** property allows you to define a list of document types for which the web part will be displayed. Click **/News**, click **Design** and choose to **Configure** (⚙️) the **BreadCrums** web part. The **Show for document types** field contains the **cms.news** value. Do not make any changes and click **Cancel**.

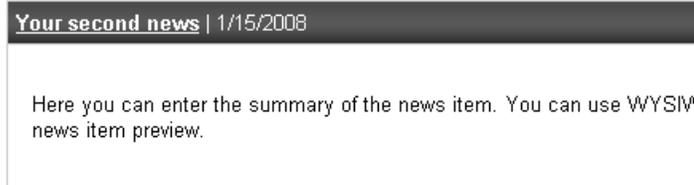


Go to the live site, to the **News** page. The breadcrumbs navigation is hidden since the currently selected document type is a page:



## News

This page contains a list of news with a header and introductory text. The n part.

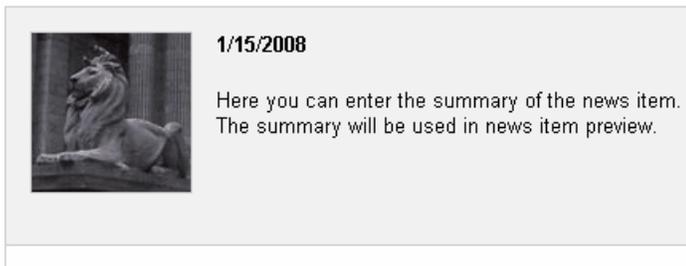


When you go to some particular news item, the currently selected document is a news item (cms.news document type) and the breadcrumbs navigation is displayed:



[News](#) > Your second news

## Your second news



You have learned how to break inheritance of content and how to display content based on the current document type.

## 5.8 Web part containers

The web parts can display only its own content or they can be rendered with a "container". The container consists of HTML code that is rendered before and after the web part.

Switch to **CMS Desk** and click **Live site**. The **Newsletter subscription** web part uses the black container:

With container:	Without container:
	

The container is useful when you need to organize the page layout into separate boxes. You can choose the web part container when you configure the web part in the **Design** tab.

The containers can be edited in the **Site Manager -> Development -> Web part containers** section.

**CMS Site Manager** Sites Administration Settings **Development** Licenses Support Buy

Development

- Categories
- CSS stylesheets
- Countries
- Cultures
- Custom tables
- Document types
- E-mail templates
- Form controls
- Inline controls
- Modules
- Notifications
- Page layouts
- Page templates
- Relationship names
- System tables
- Tag groups
- Time zones
- UI cultures
- Web part containers**
- Web parts
- Web templates
- Workflows

**Container properties**

Web part containers ▶ Black box

**General** Sites

Display name:

Code name:

Text before web part:

```
<div class="blackBox">
<h2 class="title">{%ContainerTitle%}</h2>
<div class="content">

</div>
</div>
```

Here's an example of the **Black box** container code:

**Text before web part:**

```
<div class="blackBox">
<h2 class="title">{%ContainerTitle%}</h2>
<div class="content">
```

**Text after web part:**

```
</div>
</div>
```

As you can see the container is a simple HTML code. It can use DIVs or a TABLE element. The `{%ContainerTitle%}` expression indicates where the container title should be displayed. It's dynamically substituted when the web part is displayed.

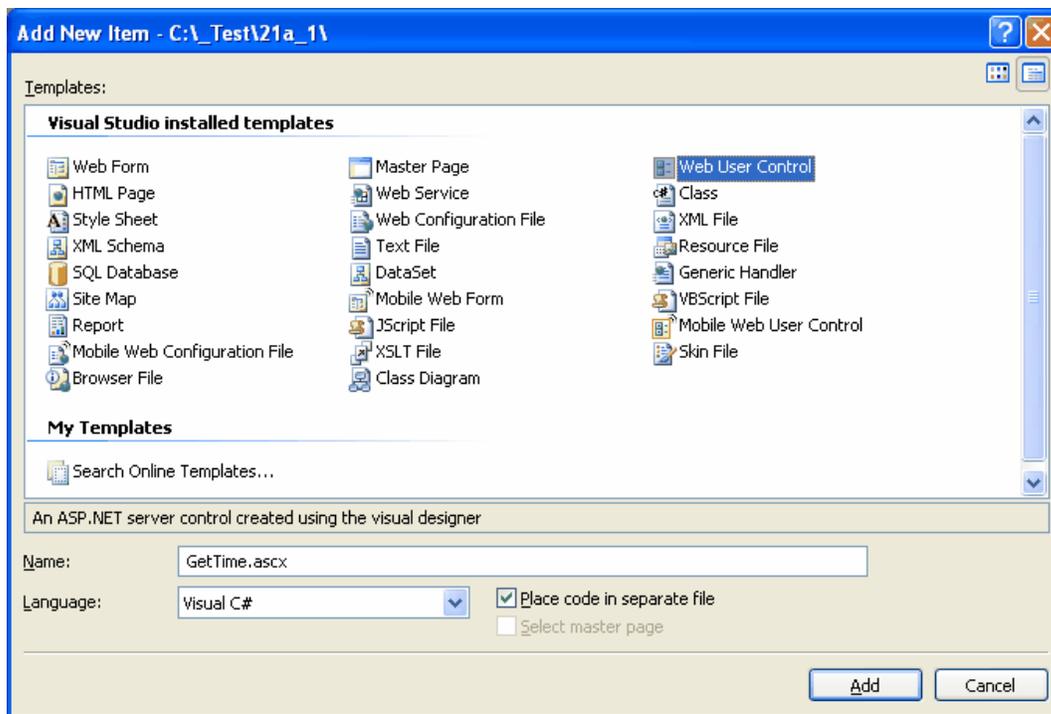
## 5.9 Adding custom code to your web site

The easiest way how to insert custom code into the portal engine-based web site is using the standard ASCX user controls. This chapter will show you how to do that. If you're not familiar with Visual Studio development, you can skip this chapter.

### Current time example

In this example, we will create a simple user control (ASCX) using Visual Studio 2005 (or 2008) and integrate it into our home page.

Open the web site project using the WebProject.sln file that is placed in the folder where you deployed the web site. Right-click the web project in the Solution Explorer window and click the **Add new item...** option. Choose to create a new Web User Control and set its name to **GetTime.ascx**. You can set the programming language option to either Visual C# or Visual Basic.



Click **Add**. Switch to the **Design** tab and drag and drop the following controls and set their properties:

#### **Button** control:

- ID: Button1
- Text: Get time

#### **Label** control:

- ID: Label1
- Text: <clear the value>



Double-click the **Get time** button and enter the following code to the Button1\_Click method:

#### [C#]

```
Label1.Text = DateTime.Now.ToString();
```

#### [VB.NET]

```
Label1.Text = DateTime.Now.ToString()
```

This code ensures that the label displays current date and time when the button is clicked. It's not necessary to compile the project - the user controls are compiled at run time.

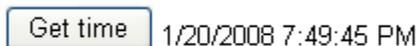
**Save** both the ASCX and ASCX.CS files.

## Adding the user control on the page

Sign in to Kentico CMS Desk, click the Home page and click **Design**. Add web part of type **General** -> **User control** to the left zone. Choose to add the General/User control web part. Enter the following value in the User control virtual path property:

```
~/GetTime.ascx
```

The ~ character represents the root of your web application. Click **OK**. Click the **Live site** mode and now you can see the user control inserted into the page. When you click the **Get time** button now, the current date and time is displayed next to the button:



## User controls versus web parts

Another option how to insert custom code into the page is creating your own web part. A web part is basically also an ASCX user control, but it inherits some standardized properties and methods from the CMSAbstractWebPart class. You will build web parts in case you need to create re-usable, parameterized user controls. The web parts are described in the next chapter.

**Part**



## 6 Developing web parts

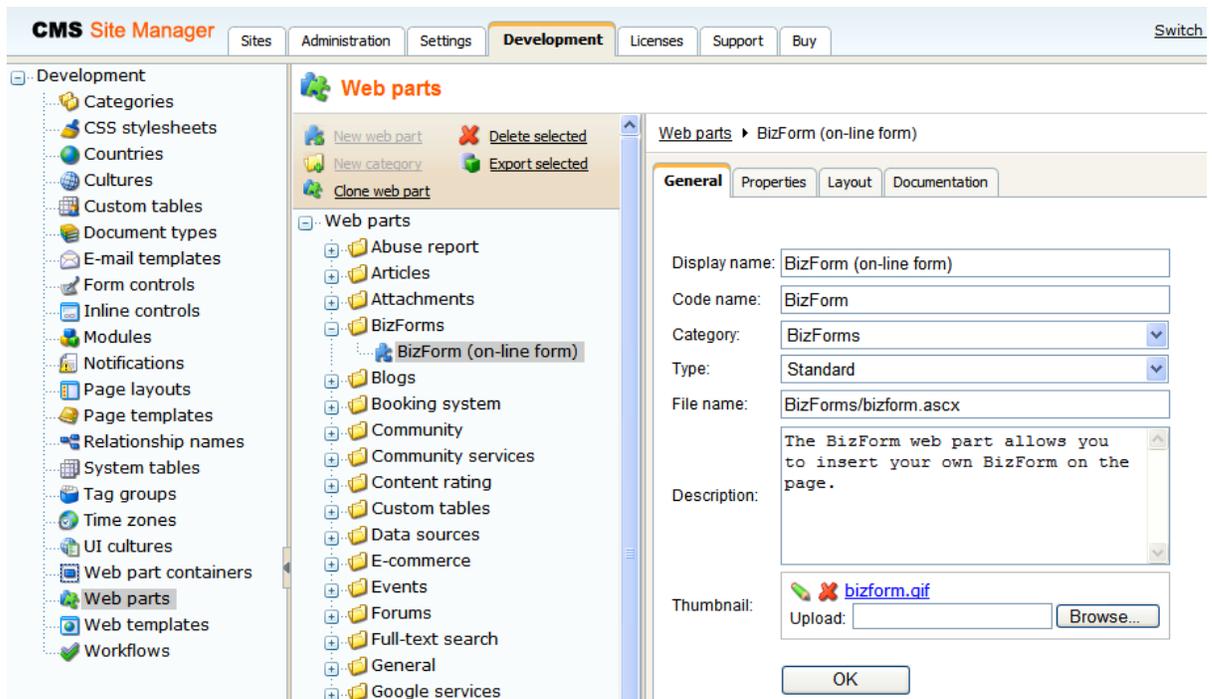
### 6.1 Web part development overview

**Web parts allow you to add your custom .NET code inside the web site.** They allow you to display content or add some functionality to your web site.

A web part is a user control (ASCX) that inherits from the **CMSAbstractWebPart** class. You can easily create your own web parts using Visual Studio 2005/2008 or Visual Web Developer 2005/2008 Express Edition.

Web parts can be managed in **Site Manager -> Development -> Web parts**. Each web part has the following properties on the **General** tab:

Display name	The name of the web part displayed to the users.
Code name	The name of the site used in the code.
Category	Here you can choose the category of the web part catalog where the web part is placed.
Type	Type of the web part. There are the following types of web parts: <b>Standard</b> - typical web parts displaying some content <b>Data source</b> - do not display any content, only provide data to be displayed by a connected web part <b>Filter</b> - can be connected to a data source and enables users to limit the data provided by it
File name	Contains a relative path to the user control that implements the web part. The path starts from the CMSWebParts folder. Example: <code>Search/cmscompletesearchdialog.ascx</code> It's recommended that you organize the web parts on the disk in the same way as in the categories.
Description	Description.
Thumbnail	Image used in the web part catalog.



On the **Properties** tab, you can define the web part properties and how they appear in the **Web part properties** configuration dialog. Each web part has the following default properties that are not displayed on this tab. These properties are created automatically when the web part is defined. However, if you define some of these properties via this tab, you can modify its default behaviour and attributes of the property (e.g. for it not to be displayed, etc.).

Default	
WebPartControllID	ID of the web part on the page.
WebPartTitle	Title of the web part displayed in CMS Desk's Design tab.
DisableViewState	Indicates if ViewState is disabled for the web part.
DisableMacros	Indicates if macro resolving is disabled for the web part.
Visibility	
Visible	Indicates if the web part should be displayed.
HideOnSubPages	Indicates if the web part should be hidden on sub-pages.
ShowForDocumentTypes	The list of document type code names separated with a semicolon (;) for which the web part should be displayed (if they are selected as the current document).
DisplayToRoles	The list of roles the web part should be displayed to.
HTML Envelope	
Container	Container (box) displayed around the web part. This value contains the code name of the container defined in <b>Development -&gt; Web part containers</b> .
ContainerTitle	Container title displayed if you specify the container.
ContentBefore	HTML content displayed before the web part. You can use it to display

	some header or add some encapsulating code, such as <div> or <table> element for required layout.
ContentAfter	HTML content displayed after the web part. You can use it to close the tags contained in the ContentBefore value, such as </div> or </table> element for required layout.
<b>AJAX</b>	
Use update panel	Indicates if AJAX UpdatePanel container is used for the web part.
<b>Time zones</b>	
Time zone	Time zone used for the web part content.
Custom time zone	Optional custom time zone that can be used specifically only for this web part.
<b>Partial caching</b>	
Partial cache minutes	Number of minutes for that the content of the web part remains cached.
Partial cache dependencies	List of cache keys on which the cached keys depend. When the cache item changes, the cache of the web part is also deleted. Each line contains one item.

The structure of the web part, content before/after and container is following:



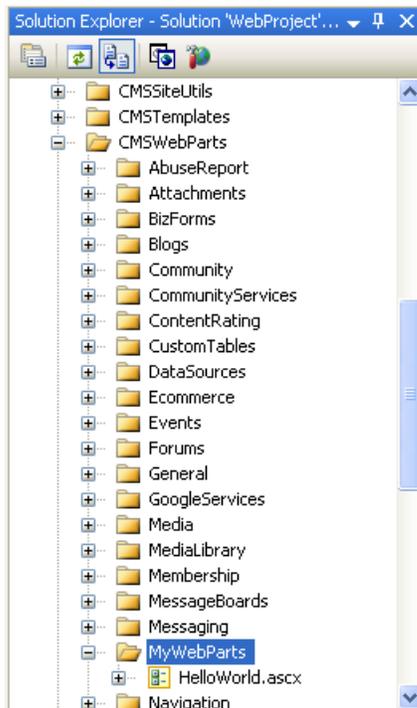
#### Storing files related to a web part

If your web part consists of several files (such as ASCX controls, images, js scripts, etc.), you should place these files in the subfolder under the folder where your main web part ASCX file is placed. If the name of the web part is MyWebPart.ascx, the subfolder name must be MyWebPart\_Files. It will ensure that the additional files are exported/imported correctly when you move your web site or when you distribute the web part to other developers.

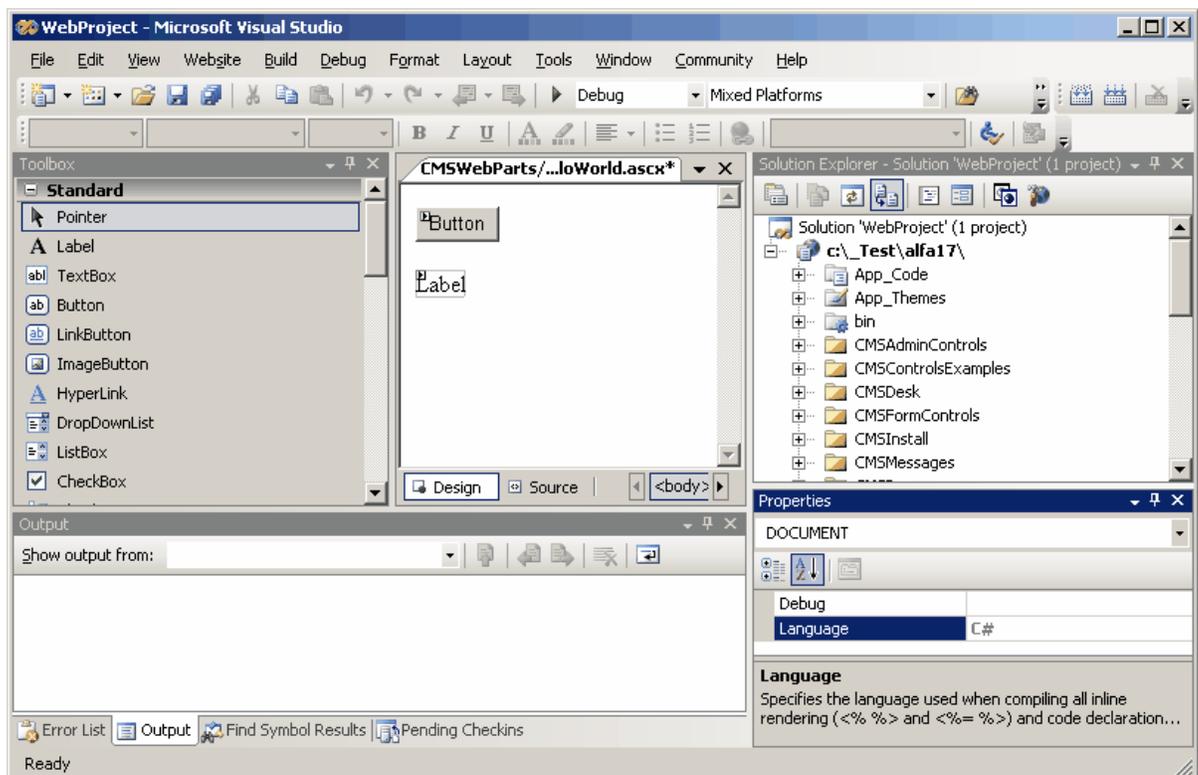
## 6.2 Creating a simple web part

This chapter will guide you through the process of creating a very simple "Hello world" web part that displays a label and a button. When the button is clicked, it updates the current time displayed in the label.

1. Open the web project in Visual Studio 2005 (or Visual Web Developer) using the **WebProject.sln** file or using **File -> Open -> Web site** in Visual Studio.
2. Right-click the **CMSWebParts** folder in the **Solution Explorer** window and choose **New Folder**. Rename the folder to **MyWebParts**.
3. Right-click the **MyWebParts** folder and choose **Add New Item**. Choose to create a new **Web User Control** and call it **HelloWorld.ascx**.



4. Display the **HelloWorld** control in **Design** tab. Drag and drop a new Button control and a new Label control on the form:



5. Double-click the Button control and add the following code to the Button1\_Click method:

**[C#]**

```
Label1.Text = DateTime.Now.ToString();
```

**[VB.NET]**

```
Label1.Text = DateTime.Now.ToString()
```

6. Add the following line to the beginning of the code:

**[C#]**

```
using CMS.PortalControls;
```

**[VB.NET]**

```
Imports CMS.PortalControls
```

7. Change the following line:

**[C#]**

```
public partial class CMSWebParts_MyWebParts_HelloWorld :  
    System.Web.UI.UserControl  
  
to  
  
public partial class CMSWebParts_MyWebParts_HelloWorld : CMSAbstractWebPart
```

**[VB.NET]**

```
Partial class CMSWebParts_MyWebParts_HelloWorld  
    Inherits System.Web.UI.UserControl  
  
to  
  
Partial class CMSWebParts_MyWebParts_HelloWorld  
    Inherits CMSAbstractWebPart
```

It ensures that the user control behaves as a web part.

Please make sure that inheritance is set to Inherits="CMSWebParts\_MyWebParts\_HelloWorld" in the HelloWorld.ascx file, as Visual Studio does not always change it automatically.

8. Add the following code to the Page\_Load method:

**[C#]**

```
Button1.Text = (string) PartInstance.GetValue("ButtonText");
```

**[VB.NET]**

(Visual Basic.NET doesn't create the Page\_Load method automatically, so you need to add the whole method:)

```
protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) _  
    Handles Me.Load  
    Button1.Text = CType(PartInstance.GetValue("ButtonText"), String)  
End Sub
```

It sets the button text to the value configured in Kentico CMS Desk.

9. **Save** all changes.

10. Open **Site Manager -> Development -> Web parts**, click the root and click **New category**. Enter *My web parts* into the **Category display name** field, *MyWebParts* into the **Category name** field and click **OK**.

11. Click the new category and click **New web part**. Click the **Create a new web part** radio button and enter the following values:
  - Web part display name: Hello world
  - Web part code name: HelloWorld
  - Web part file name: MyWebParts/HelloWorld.ascx

Click **OK**.

12. Switch to **CMS Desk**.

13. Click the **Services** page in the content tree and switch to the **Design** tab.

14. Add the new **Hello world** web part to the left zone and set the value of the **Button text** field to *Hello world!*

15. Now switch to the **Live site** mode. You will see the button with text Hello world! When you click it, the label displays current date and time:



1/21/2008 11:55:54 AM

You have learned how to create a simple web part.



**Tip: Displaying content on the web part**

You can also use Kentico CMS Controls on the web part to display content from Kentico CMS.

## 6.3 Web part modifications

### Web part cloning

If you need to copy or modify the behavior of some web part, you can clone the web part. It ensures that your changes won't be overwritten with the nearest upgrade.

You can clone a web part in **Site Manager -> Development -> Web parts**. Choose the original web part and click **Clone web part**. Then, you can modify the web part code in Visual Studio.

### Web part layout

Custom web part layouts allow you to modify the design of web parts. It's useful if you need to change the look of some dialog or hide some field.

Go to **CMS Desk**, choose the **Design** tab, open web part properties and switch to the **Layout** tab. Here you can choose to create a new layout and modify its HTML code. Please note: you always need to keep all ASP.NET controls in the layout. If you need to hide some of the controls, just add the `Visible="False"` attribute to the control.

### Web part inheritance

Web part inheritance allows you to create web parts with pre-set values for particular purpose. You can, for example, create a specialized web part for displaying news, inherited from the Repeater web part. You will define default values for the relevant properties so that the Repeater web part displays news items without additional configuration. It may speed-up your work.

You can create an inherited web part by going to **Site Manager -> Development -> Web parts**, clicking **New web part** and choosing to **inherit** from an existing web part.

**Part**



## 7 Managing styles and design

### 7.1 CSS styles

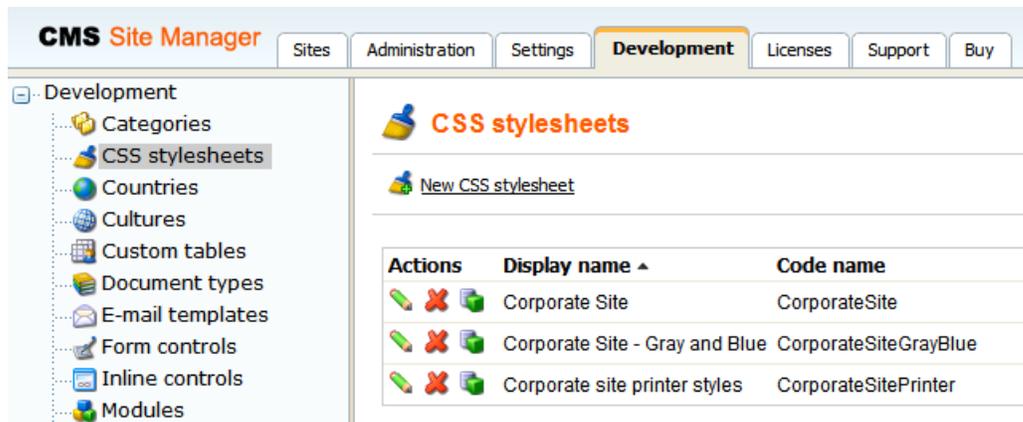
The design of the web site relies on standard CSS styles. Each web site has its global CSS stylesheet that can be chosen in **Site Manager -> Sites -> ... edit site ... -> General**. Here you can also choose a different CSS stylesheet used by WYSIWYG editors.

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

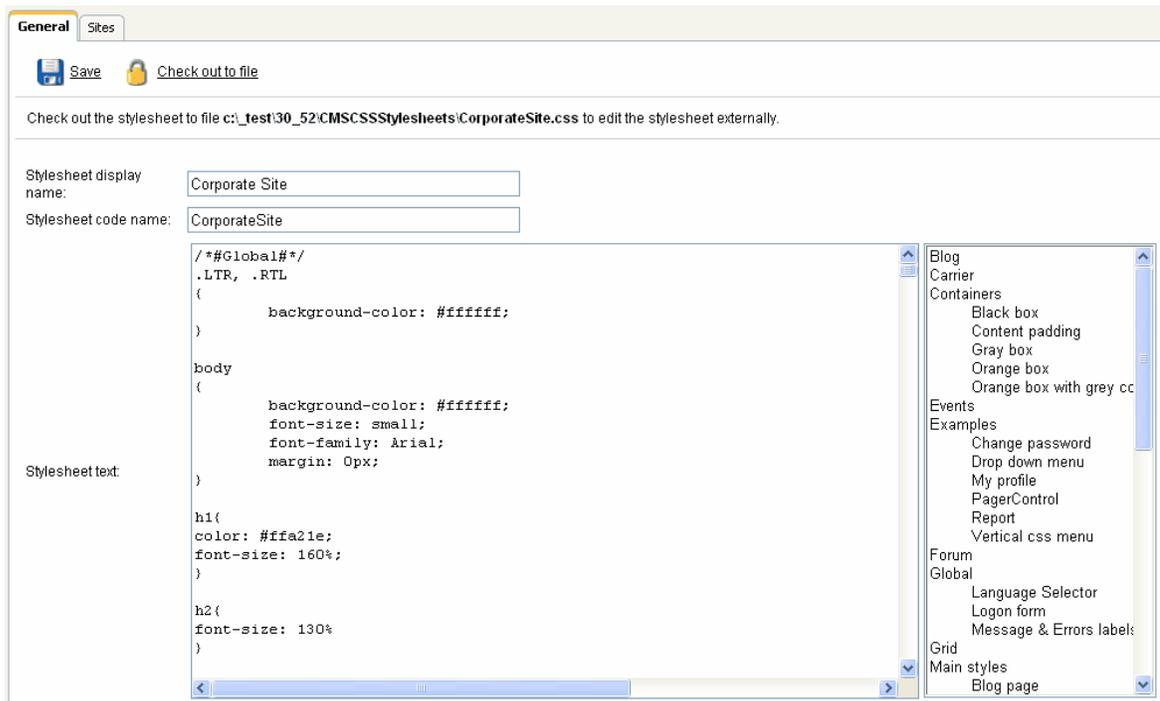
The CSS stylesheets can be managed in Site Manager. When you're in CMS Desk, you can easily switch to Site Manager by clicking the **Switch to Site Manager** link in the header:



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



Edit  the **Corporate Site** stylesheet:



General Sites

Save Checkout to file

Checkout the stylesheet to file c:\\_test\30\_52\CMSCSSStylesheets\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

## 7.2 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.

## 7.3 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**

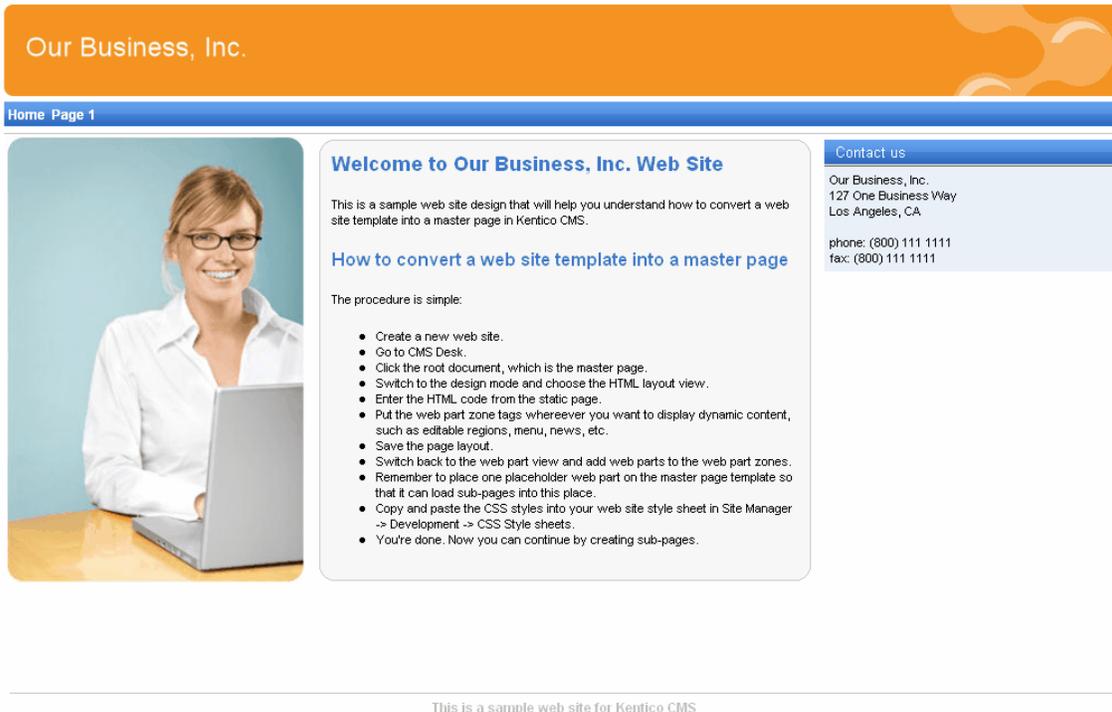


## 8 Creating a new site using the portal engine

### 8.1 Overview

This part of the tutorial will guide you through the creation of a simple web site. You will learn how to define site structure and design and 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 a graphic designer. It looks like this:



You can find the static page template in folder **C:\Program Files\Kentico CMS\<version>\CodeSamples\SampleWebTemplate**. The template consists of the home.htm file, styles folder and app\_themes folder with images.

## 8.2 Creating a new web site using a wizard

The following chapters assume that you have previously installed a sample Corporate Site 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 Visual Studio 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: **CMS Site Manager**, **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** (with a globe icon), **Import site or objects** (with a folder icon), and **Export objects** (with a document icon). Below these links, there 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**.

**Step 1** | **Choose default web site**

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.

Create a new site using a wizard

Use web site template

Next >

In the second step, enter the following details:

- **Site display name:** My web site
- **Site code name:** mysite
- **Domain:** 127.0.0.1 - if you're using Visual Studio built-in web server, set the default value to localhost
- **Default culture:** English - United States *(by choosing a different default culture, you can achieve displaying date, time and numeric values in culture-specific format)*

Click **Next**.

**Step 2** | **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:

Site culture:  ▼

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

**Step 3** | **Objects selection**  
Please select objects which should be imported.

**All objects**

- Web site
  - Documents
  - 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 fourth step, the progress of objects import is displayed. Click **Next** when **Import has successfully finished** appears.

**Step 4** | **Import progress**  
Objects are being imported.

**Import has successfully finished**  
Starting site 'My web site'  
Copying objects files  
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 'Settings' objects  
Importing 'Settings categories' objects

Import in progress

In the fifth step, you can choose the master page. You can change or modify it at any time later. For now, please choose the **Blank master page** option.

**Step 5** | **Select master page**  
The master page defines the layout of the main menu, logo and content placeholders. You can change it at any time later.

**Blank master page**  
Generic default page template.

**Company**  
**Top logo and left menu**  
Master page template with logo on the top and menu on the left side.

**Company**  
**Top logo and menu**  
Master page for the Sample web site.

Next >

Click **Next**. In the sixth step, you can design the site map of your new web site. Click **New** and add a new page called **Home** and choose page template **Templates with editable regions -> Simple text**. Click **OK** to save it.

**Step 6** | **Define basic site structure**  
Define the site map of your new web site. The pages you create will be displayed in the site menu. Every page must have some template specified or it can inherit page template from the parent page.

New Delete Up Down

My web site  
Home

**Page properties**

Page name:

Page template:

Click **My web site** and click **New**. Add a new page called **News** and choose page template **News** -> **News list**. Click **OK** to save it.

**Step 6** | **Define basic site structure**  
Define the site map of your new web site. The pages you create will be displayed in the site menu. Every page must have some template specified or it can inherit page template from the parent page.

New Delete Up Down

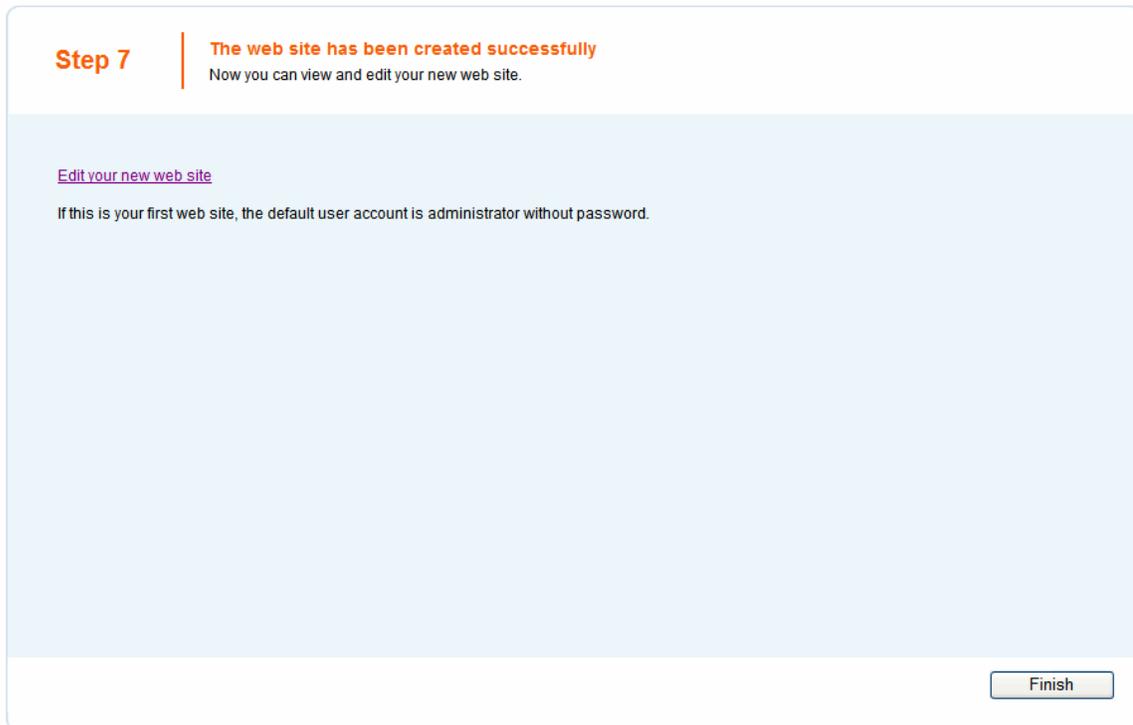
My web site  
Home  
News

**Page properties**

Page name:

Page template:

Click **Next**. You will see a confirmation message:



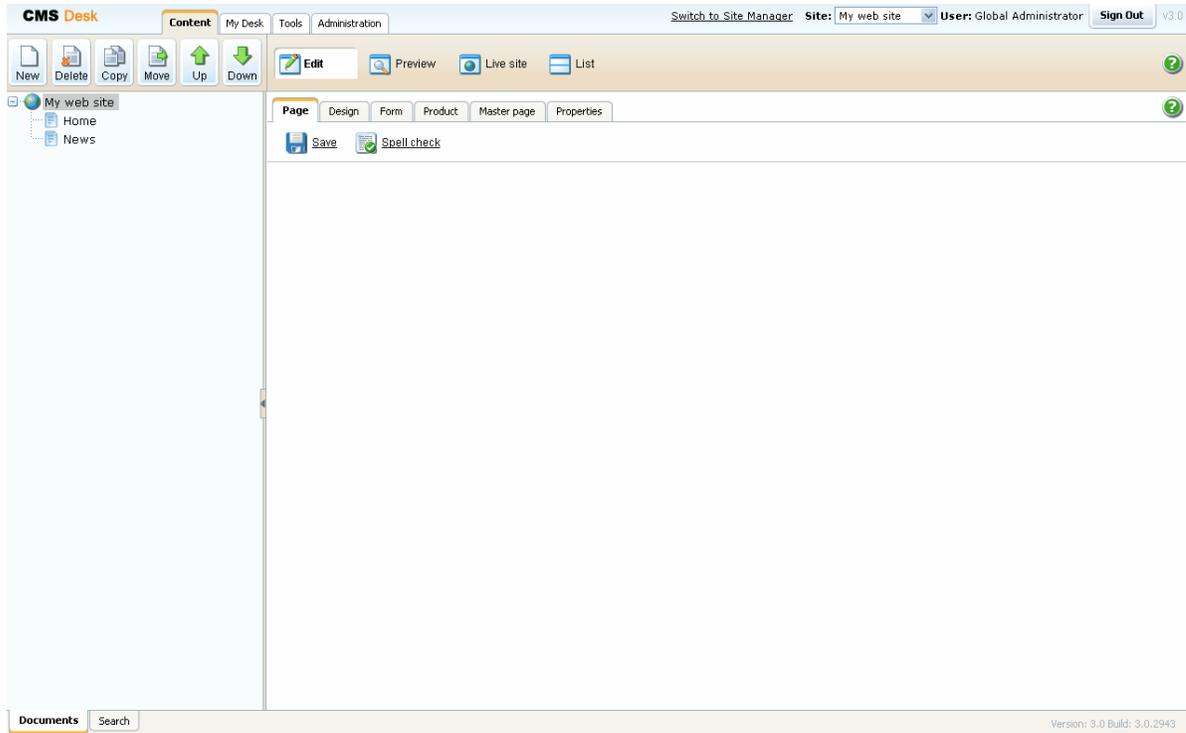
**Step 7** | **The web site has been created successfully**  
Now you can view and edit your new web site.

[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 (user name administrator, blank password) since the authentication is not shared over different domains. After you sign in, you will see your new, empty web site:

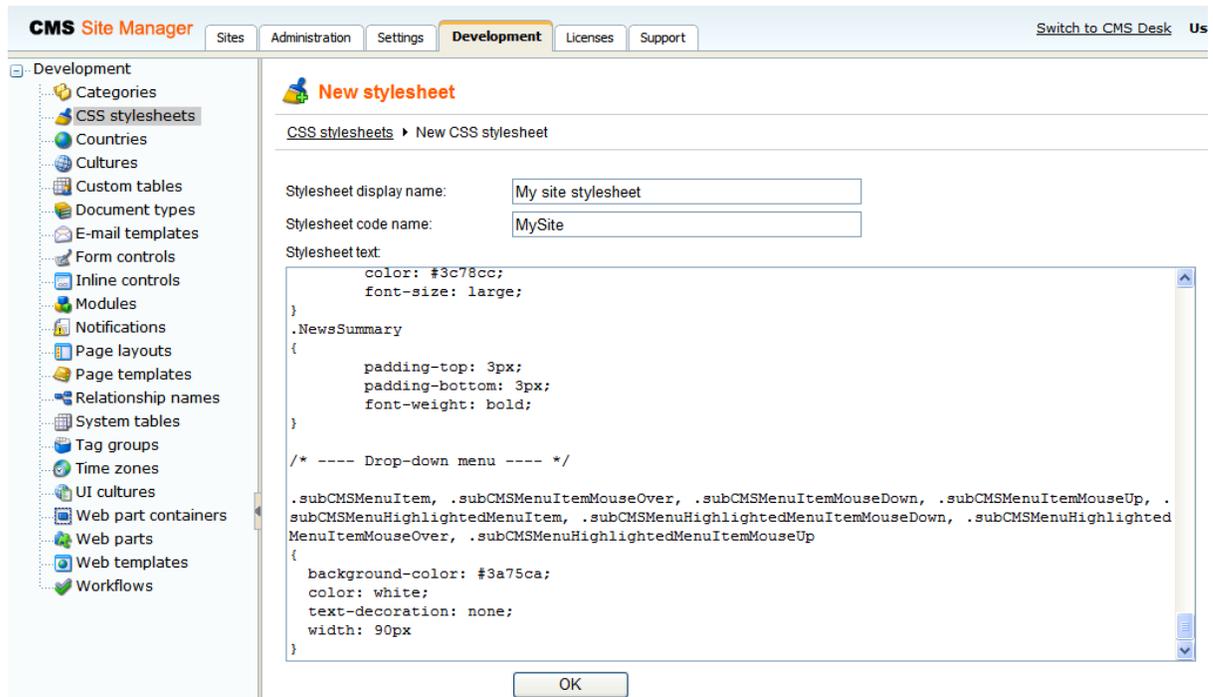


You have created the base for your new web site. In the next chapters, you will learn how to implement the required design.

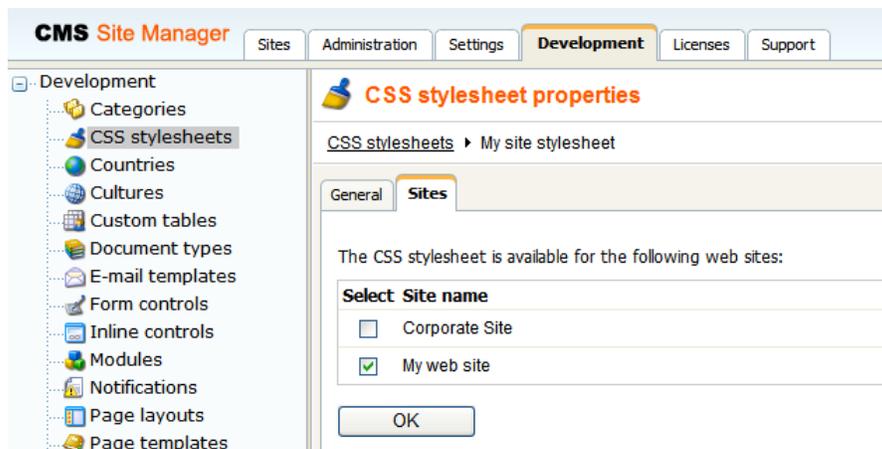
## 8.3 Creating a CSS stylesheet

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:

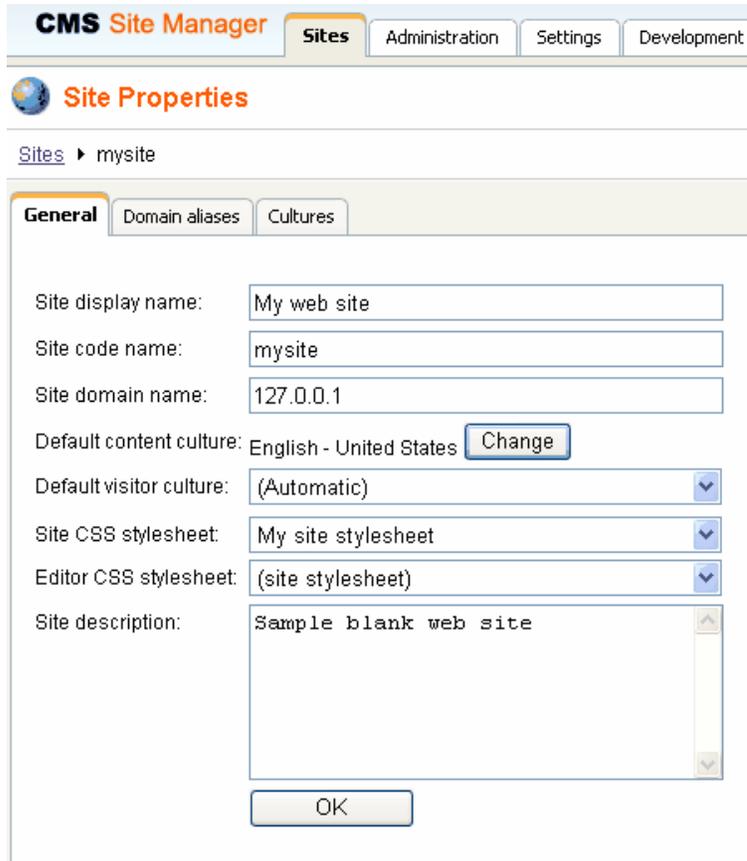
- Stylesheet display name: **My site stylesheet**
- Stylesheet code name: **MySite**
- Stylesheet text: copy and paste all text from the **SampleWebTemplate\Styles\main.css** file (you will find it in folder **C:\Program Files\Kentico CMS\<version>\CodeSamples**)



Click **OK**. Click the **Sites** tab and make sure the stylesheet is assigned to **My 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.



**CMS Site Manager** Sites Administration Settings Development

**Site Properties**

Sites ▶ mysite

**General** Domain aliases Cultures

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: Sample blank web site

Now copy the folder **SampleWebTemplate\app\_themes\MySite** to **<web project>\app\_themes**. The folder contains graphics for this web site template. 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 sample 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

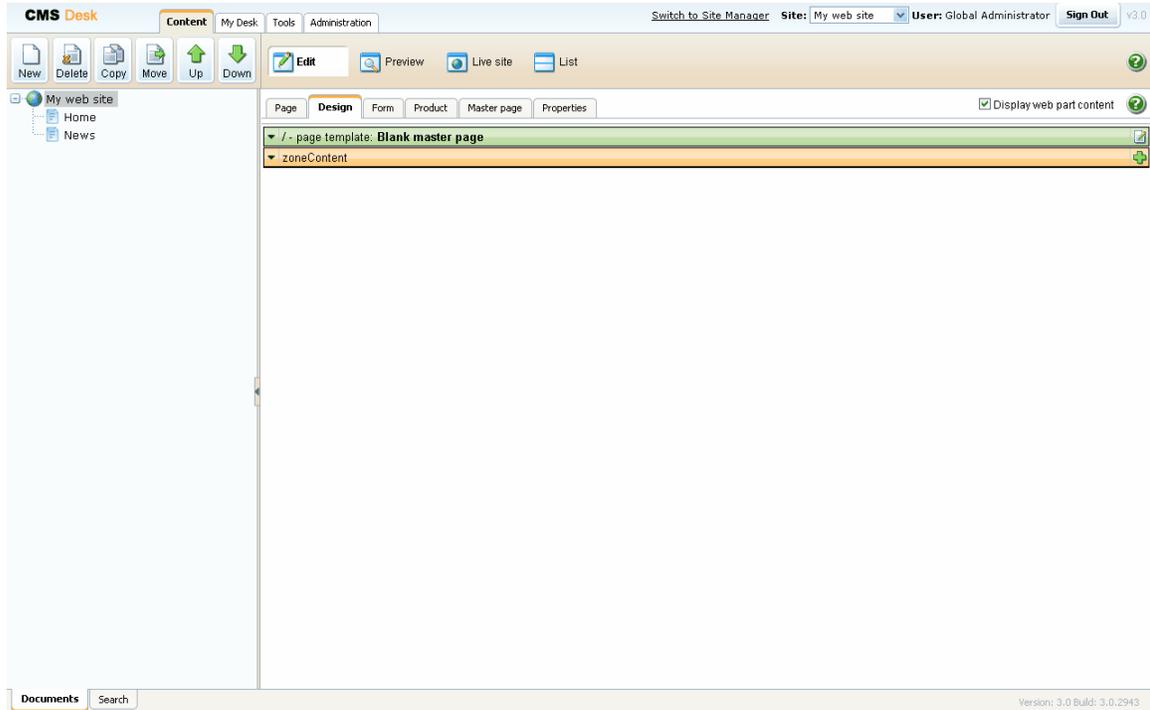
`<web project>/CMSPages/GetCSS.aspx?stylesheetname=MySite`

which means, you need to link to files in the app\_themes folder like this:

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

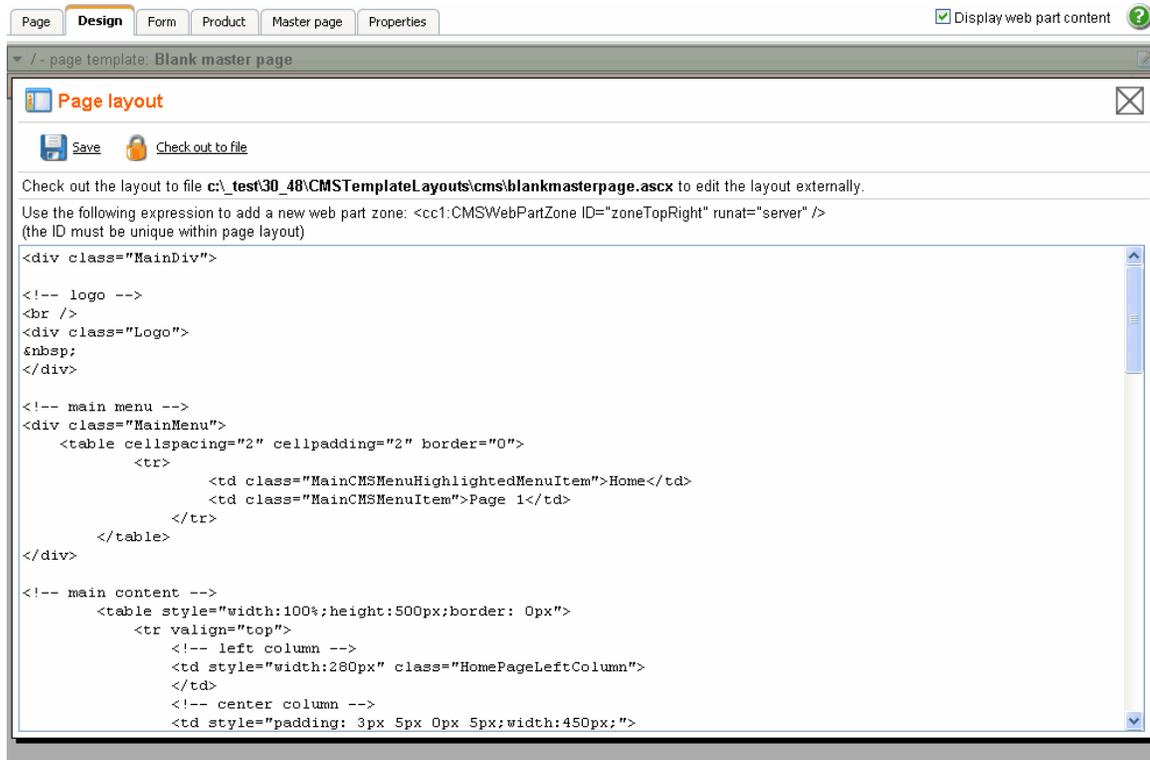
## 8.4 Master page

The site master page is defined by the root document. Switch to **CMS Desk**, click the root of the content tree and switch to the **Design** tab. You will see the blank page template:



As you can see, there are no web parts at this moment. Click the **Edit layout** (📄) icon in the green box to edit the **HTML layout**.

Clear the text box content and copy and paste the HTML code inside the <BODY> element from file **home.htm** (it's part of the web site template located in folder C:\Program Files\Kentico CMS\<version>\CodeSamples\SampleWebTemplate) into the text box:



The screenshot shows the 'Page layout' editor in Kentico CMS. The title bar indicates the page template is 'Blank master page'. The editor has tabs for 'Page', 'Design', 'Form', 'Product', 'Master page', and 'Properties'. A 'Display web part content' checkbox is checked. The main area contains the following HTML code:

```
<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 -->
<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;">
```

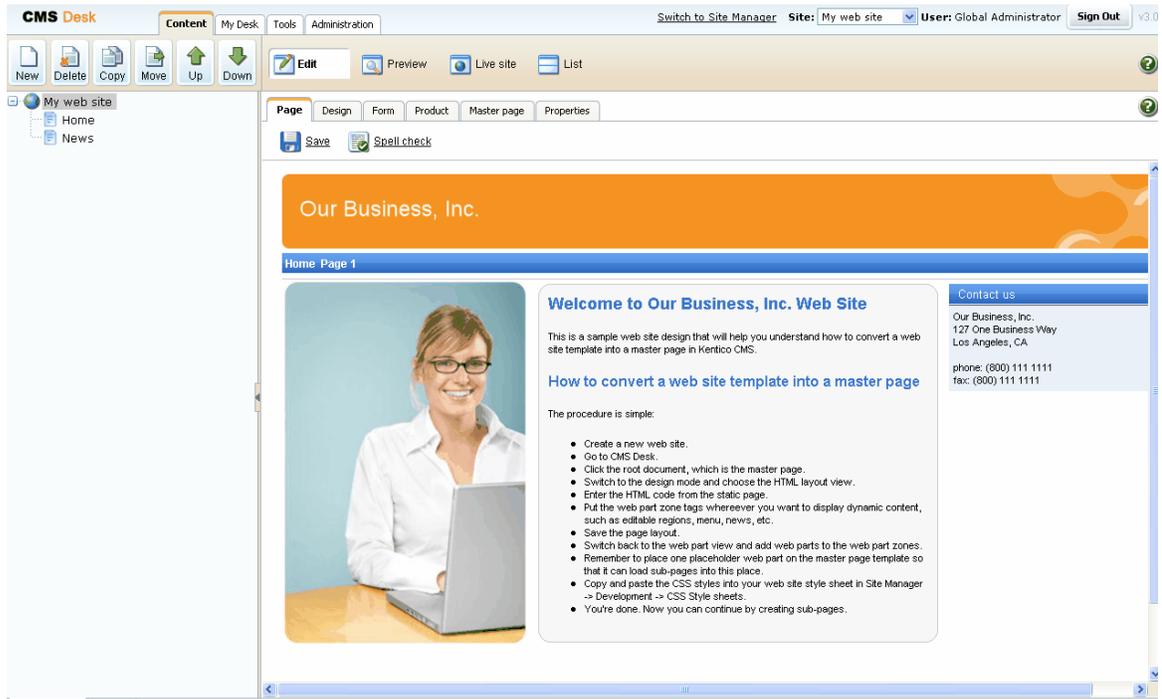
Click **Save**.



### 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. You may also need to modify page layouts in **Site Manager -> Development -> Page layouts** so that they use CSS-based layout. We use table-based layout by default since it's easier to understand, although we are aware of advantages of the CSS-based layout.

Click the **Page** tab now and you can see your new web site design (it's still static):



## Adding web part zones

Now we need to replace static text with web part zones, so that we can put dynamic web parts on the page. Click **Design** and click **Edit layout** (📄).

Locate the `<!-- main menu -->` section in your page template and replace the following code:

```
<!-- 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>
```

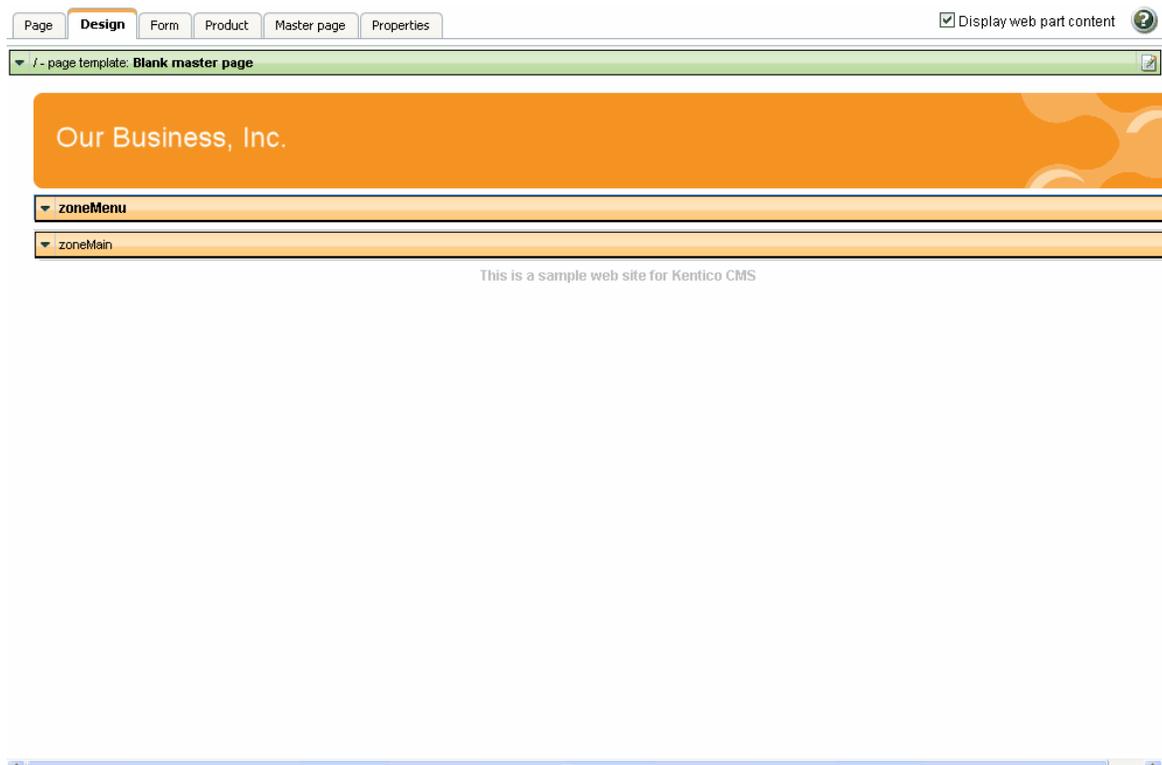
with:

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

Now replace the main content section between `<!-- main content -->` and `<!-- /main content -->` lines with:

```
<ccl:CMSWebPartZone ID="zoneMain" runat="server" />
```

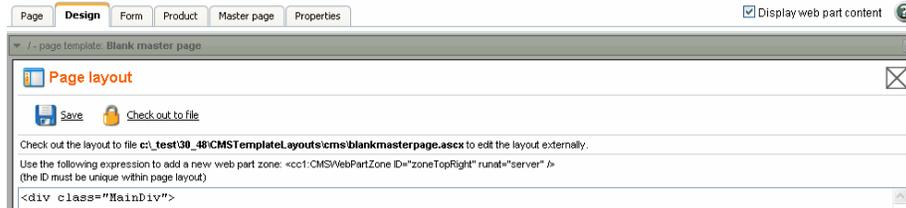
Click **Save**. After you close the HTML layout, you will see the new layout with web part zones:





### Tip 1: Editing HTML layout code in an external editor

If you prefer to edit the HTML layout code in some external editor, such as Visual Studio or Dreamweaver, you can Check-out the code to file on your disk using the **Check out to file** button in the HTML layout editor:



Now you can open the file using the path displayed on the screen in your favorite editor. You can modify the HTML code and preview the changes on the live site. After you finish the changes, you can check in the code back to the system and make the changes available to all users.

**Please note:** If you check-in the file or undo the check-out, the file is deleted on your disk.

You can use the same concept with transformations and CSS stylesheets as well.



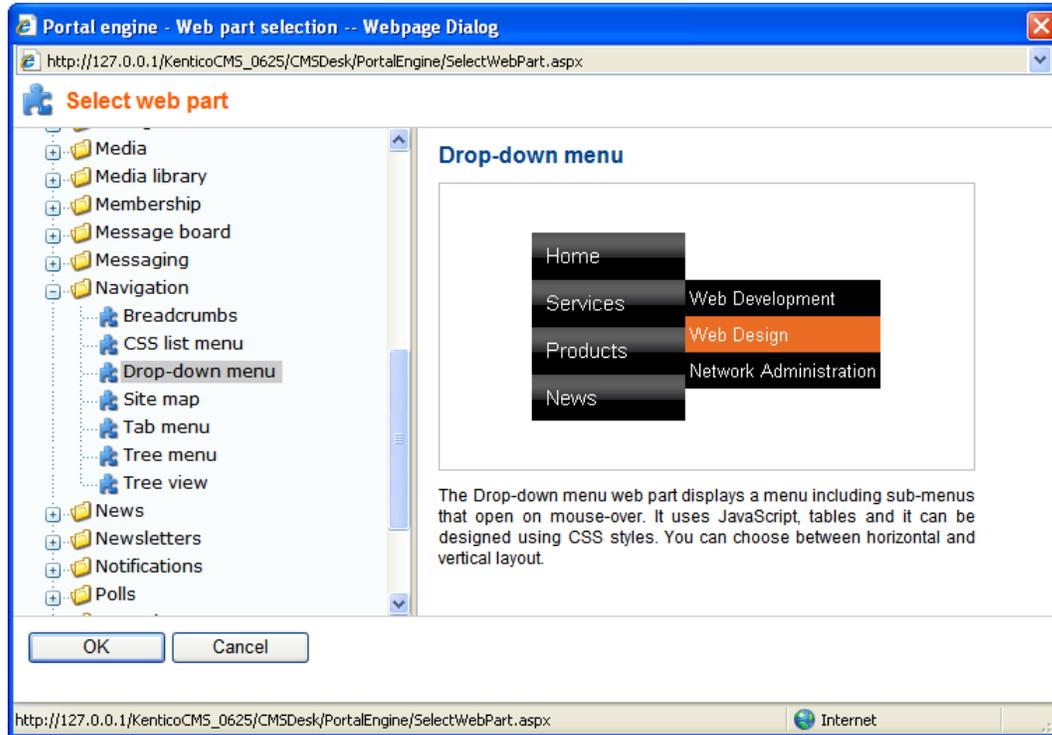
## Tip 2: Master page tab

The **Master page** tab allows you to add custom **DOCTYPE** directive, custom **HEAD section elements** and custom **BODY attributes**. It also allows you to edit the master page HTML layout directly (this is the same code that you edit in the **HTML layout** mode of the **Design view**).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>
My Site - </title>
<meta http-equiv="pragma" content="no-cache" />
<link type="text/css" rel="stylesheet" href="/30_48/CMSPages/GetCSS.aspx?stylesheetname=MySite" />
</head>
<body class="LTR DesignMode IE7">
<div class="MainDiv">
<!-- logo -->
<br />
<div class="Logo">
&nbsp;
</div>
<!-- main menu -->
<div class="MainMenu">
<cc1:CMSWebPartZone ID="zoneMenu" runat="server" />
</div>
<!-- main content -->
<cc1:CMSWebPartZone ID="zoneMain" runat="server" />
<!-- /main content -->
<!-- footer -->
<div class="Footer">
</div>
</body>
</html>
```

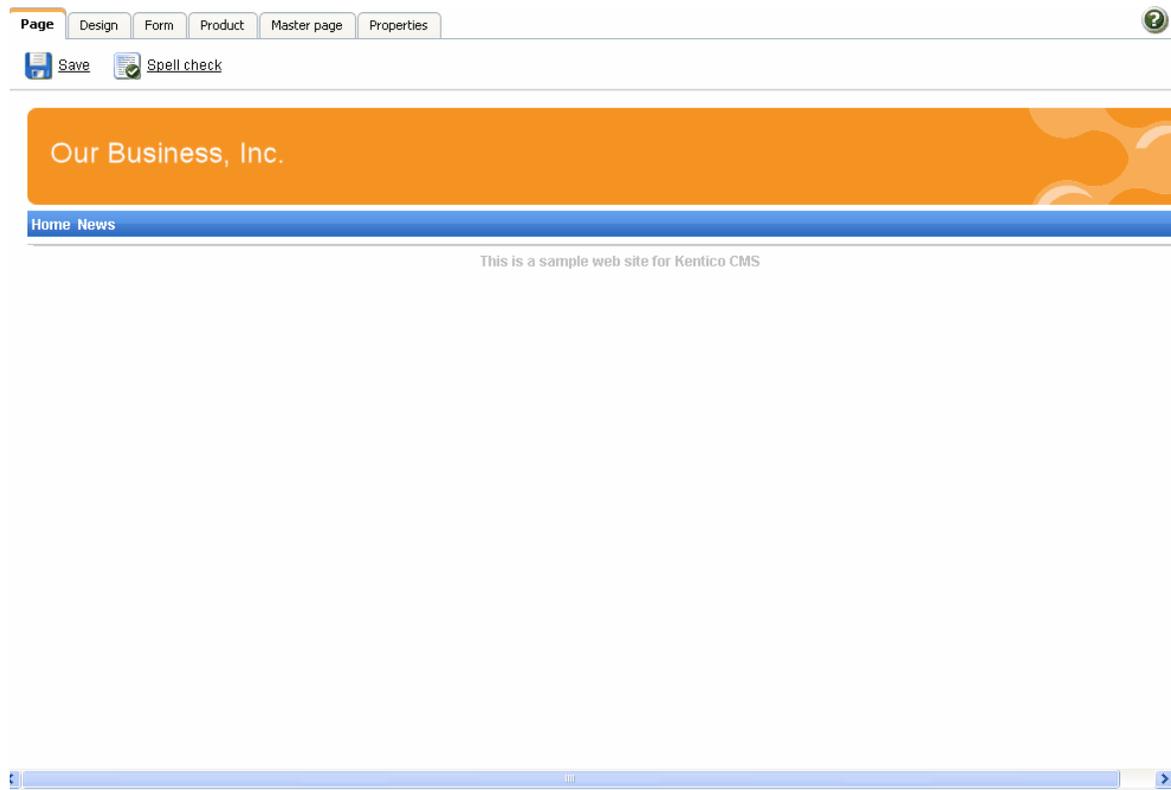
## 8.5 Main menu

Now we will add the main menu. Switch to the **Design** tab of the root. Add (+) the **Navigation/Drop-down menu** web part to the **zoneMenu** zone:



Click **OK**. The **Web part properties** dialog opens. Set the value of **CSS prefix** to `;sub` - it means that the menu CSS classes for second menu level will have a prefix `sub*`. Set the value **Design/Layout** to **Horizontal**. Click **OK**.

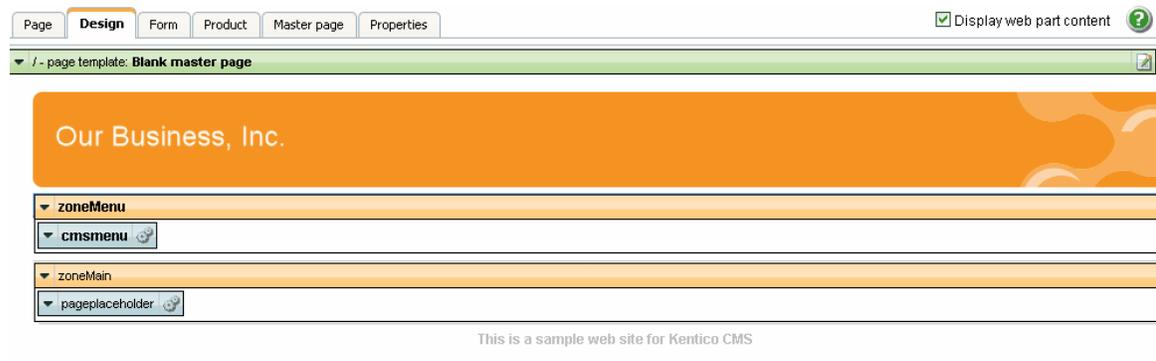
When you switch to the **Page** tab, you will see the page with a dynamic menu that displays the pages - Home and News:



## 8.6 Page placeholder

Now we need to add a **Page placeholder** web part that ensures loading of sub-pages (such as Home or News page) into the master template. There can be only single page placeholder web part in a single page template.

Switch to the **Design** tab and add the **General/Page placeholder** web part into the **zoneMain** web part zone. Click **OK**. You do not need to make any changes to the page placeholder settings.



## 8.7 Home page

Now we will create the home page. Delete the existing **/Home** page. Click the root and click **New** to create a new **Page (menu item)** document. Enter the page name **Home** and choose to **create a blank page** using the **Simple** layout:

The screenshot shows the Kentico CMS Desk interface. At the top, there are tabs for 'Content', 'My Desk', 'Tools', and 'Administration'. The 'Content' tab is active. Below the tabs is a toolbar with icons for 'New', 'Delete', 'Copy', 'Move', 'Up', 'Down', 'Edit', 'Preview', 'Live site', and 'List'. The main area is titled 'My web site' and contains a list of items, including 'News'. Below the list, there are buttons for 'Save', 'Save and create another', and 'Spell check'. The 'Page name' field is set to 'Home'. There are three radio buttons: 'Use page template', 'Inherit from parent page', and 'Create a blank page', with 'Create a blank page' selected. A list of page layouts is shown, with 'Simple' selected. To the right of the list is a preview area showing a 'no image' placeholder. At the bottom, there is a checkbox labeled 'Copy this layout to my page template' which is checked.

Click **Save** to create the page.

Now we will define a new HTML layout for our home page. Click the Home page. Switch to the **Design** tab and click the **Edit layout**  icon. Replace the default code with following code:

```
<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;">
      <ccl:CMSWebPartZone ID="zoneCenter" runat="server" />
    </td>
    <!-- right column -->
    <td style="padding: 3px 0px 0px 5px;width:270px">
      <ccl:CMSWebPartZone ID="zoneRight" runat="server" />
    </td>
  </tr>
</table>
```

Click **Save** to save the new layout and close the HTML layout editor. Add the **Text -> Editable text** web part into the **zoneCenter** web part zone. Set its properties:

- Editable region title: **Main text**
- Editable region height: **450**

Click **OK**. Add another **Text -> Editable text** web part into the **zoneRight** web part zone and click **OK**. Set its properties:

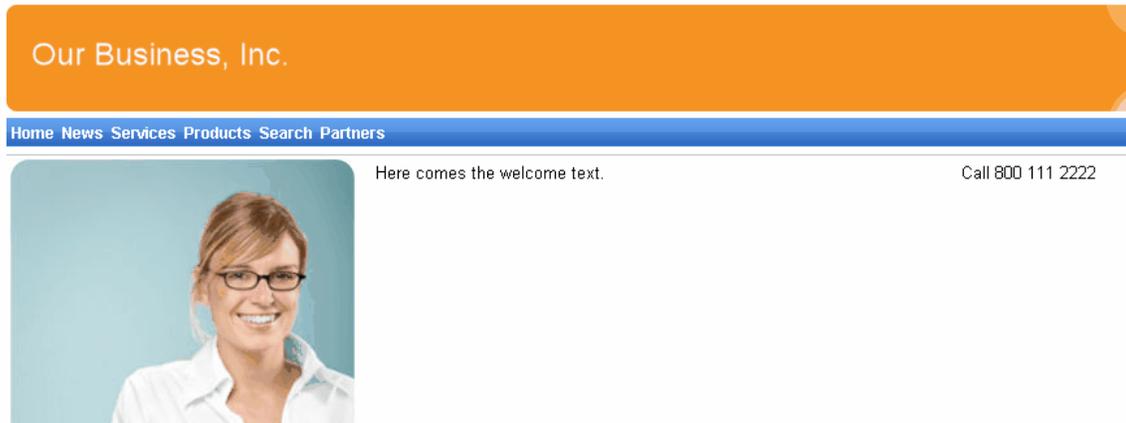
- Editable region title: **Contact text**
- Editable region height: **100**

Click **OK**. 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** in the **Site** drop-down menu, 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. If not, please uncheck the **Inherit from global settings** box, enter the value and click **Save**.

The screenshot shows the 'Settings' page for 'My web site'. The 'Default alias path' is set to '/Home'. The 'Inherit from global settings' checkbox for this setting is unchecked. Other settings include:

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"/>

## 8.8 Container for home page text

Now we will create the gray and blue containers for our home page text. Such a container can be re-used for any other web part later. If you do not wish to use a container, you could alternatively insert the surrounding code into the HTML layout of your page template.

Go to **Site Manager -> Development -> Web part containers** and click **New container**. Enter the following values:

- Display name: **My web site gray box**
- Code name: **MyWebSiteGrayBox**
- Text before web part:

```
<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">
```

- Text after web part:

```

    </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>
```

Click **OK**. Switch to the **Sites** tab and make sure that the container is assigned to **My web site**. Now create another container:

- Display name: **My web site blue box**
- Code name: **MyWebSiteBlueBox**
- Text before web part:

```
<table cellpadding="0" cellspacing="0" style="width: 100%;" class="Blue">
  <tr>
    <td class="BoxTitle">{%ContainerTitle%}&nbsp;</td>
  </tr>
  <tr>
    <td class="BoxArea">
```

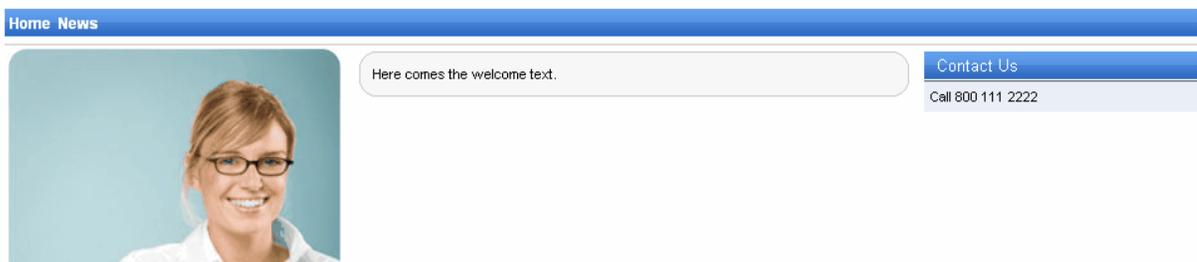
- Text after web part:

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

Now go back to **CMS Desk -> Content ->** select **Home** page -> **Design** tab and configure the **editabletext** web part in the middle column. Set its **Web part container** property to **My web site gray box** and click **OK**.

Configure the **editabletext1** web part in the right column and set the **Web part container** property to **My web site blue box**. Set the **Web part container title** property to *Contact Us*.

When you go to the **Live site** now, you can see the gray box around the home page text and blue box around the Contact Us text:



## 8.9 News page

Click **/News** in the content tree. The News page is ready to use since it uses the pre-defined **News list** page template. 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.
- Teaser, Publish from, to: leave the fields blank.

Click **Save and create another**. Enter the following values:

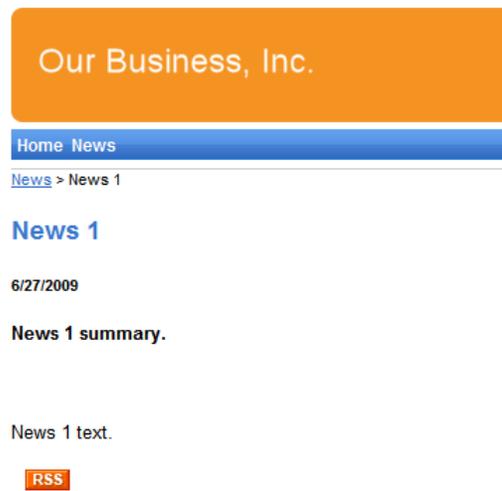
- News title: News 2
- Release date: click the date-time picker and click **Now** and click **OK**.
- News summary: News 2 summary.
- News text: News 2 text.
- Teaser, Publish from, to: leave the fields blank.

Click **Save**.

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:

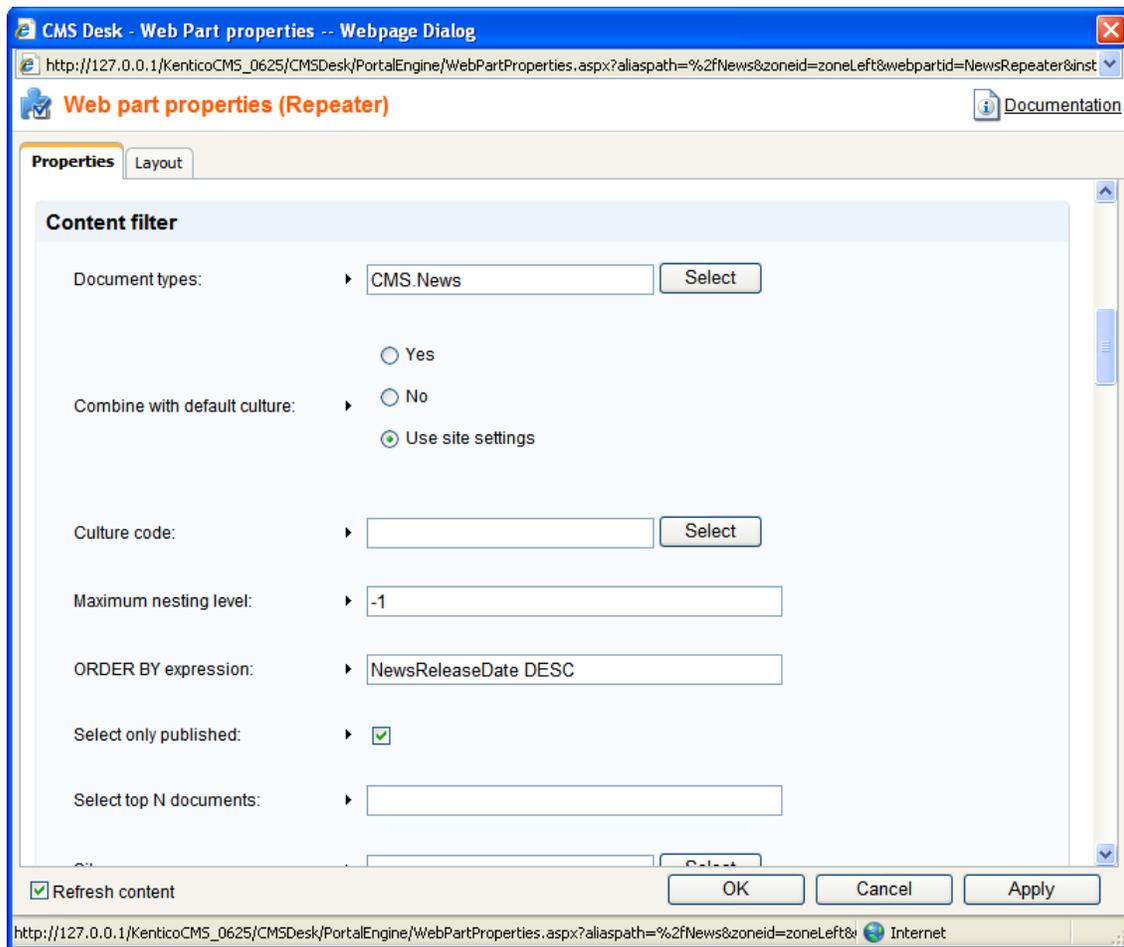


The breadcrumbs now show the current path on the web site: **News > News 1**. The position is also reflected in the URLs:

- 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.

You might wonder how the listings are displayed by Kentico CMS. Let's explain it on this page: Click **/News**, click **Edit** and choose the **Design** tab. Edit the properties of the **NewsRepeater** web part. This web part ensures displaying of the list of documents. It's actually based on the standard ASP.NET Repeater control that you may already know.



The most important properties are these:

- **Path** - this property says where the content that should be displayed is located in the content tree. If you leave the value empty, it searches for documents under the current page.
- **Document types** = cms.news - this property says what type of documents should be displayed (news/products/jobs/etc.)
- **Transformation** = cms.news.preview - this property says which transformation should be used for the items displayed in the listing. The transformation is actually the ItemTemplate section that you normally use with Repeater control. The transformations can be managed in **Site Manager -> Development -> Document types -> ... edit document type ... -> Transformations** dialog.
- **Selected item transformation** = cms.news.default - this property says which transformation should be used for the detailed view of the selected document.
- **Paging** - if you plan to have too many news items on a single page, you can configure automatic paging using this property.

### How it works

1. You go to the **/News page**.
2. The **NewsRepeater** web part checks if you have selected some particular news item (based on its Document types 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.
4. 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 details.



### Path expressions

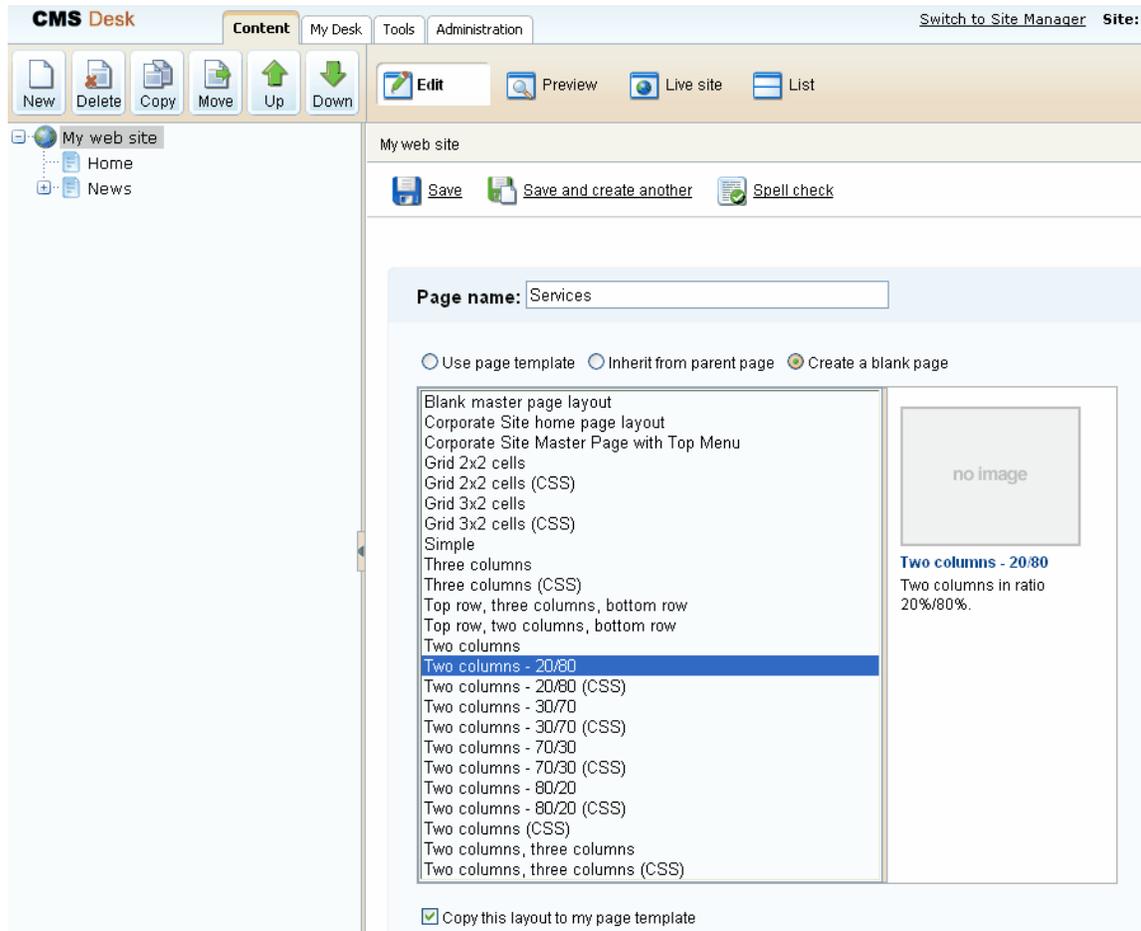
The Path property in web parts supports following special expressions that allow you select the content dynamically:

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

## 8.10 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.

Click the root of the content tree and click **New**. Choose to create a new **Page (menu item)**. Enter page name **Services** and choose to create a blank page using the **Two columns - 20/80** layout. Leave the box "Copy this layout to my page template" checked.



Click **Save**. Click **Design** and add the **Navigation -> Tree menu** web part into the **zoneLeft** zone. Set the following menu properties:

- **Content -> Path:** `/{0}/%` (this means that the menu starts from the second level)
- **Design -> Item image URL:** `~/app_themes/mysite/images/bullet.gif`
- **Design -> Open item image URL:** `~/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)*

Click **OK**.

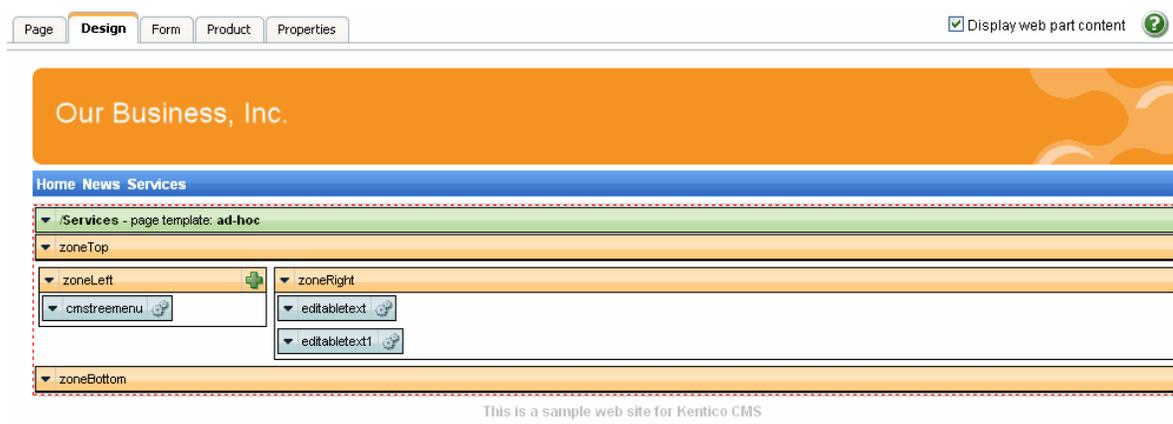
Add the **Text/Editable text** web part into the **zoneRight** zone and set the following values:

- **Editable region title:** Header
- **Editable region type:** Text box
- **Editable region width:** 600
- **Content before:** <h1>
- **Content after:** </h1>

Click **OK**. Add another web part of type **Text/Editable text** below the first one and set the following web part properties:

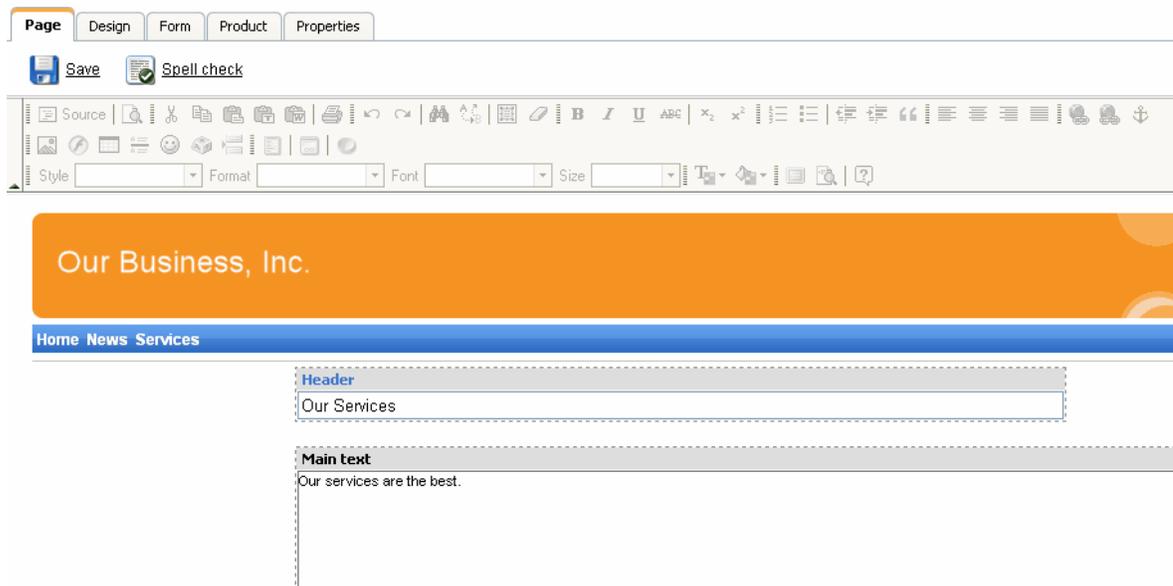
- **Editable region title:** Main text
- **Editable region type:** HTML editor
- **Editable region height:** 300

Click **OK**. The design tab of the page now looks like this:



## Entering text

Click the **Page** tab and enter some text into editable regions. Click **Save**.



## Creating a re-usable template

Now we will save this page as a re-usable template so that we can use it for sub-pages. Go to **Properties -> Template** and click **Save as new page template**. Enter the following values:

- **Template display name:** Services page template
- **Template code name:** ServicesPageTemplate
- **Template category:** General.

Click **OK** and click **Save** on the **Template** dialog.

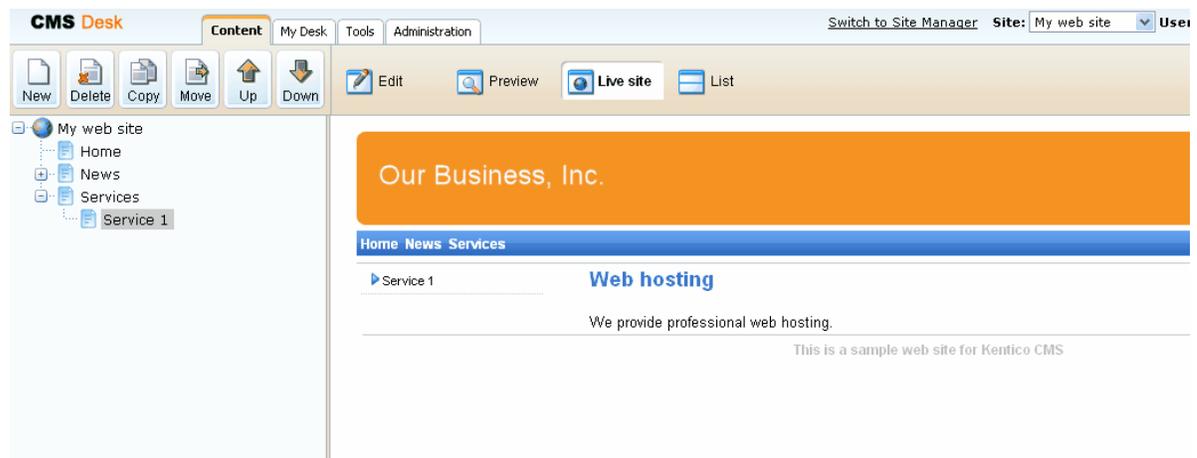
## Adding sub-pages

Click **New** in the main toolbar and choose to create a new **Page (menu item)** under the /Services page. Call the page **Service 1** and choose the **General/Services page template** template. Click **Save**. Enter the following text:

- **Header:** Web hosting
- **Main text:** We provide professional web hosting.

Click **Save**.

Click **Live site**. You will see a page like this:



In this chapter, you have learned how to create a new page template with editable regions from scratch and how to re-use it for multiple pages.

## 8.11 Products page

### 8.11.1 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.

### 8.11.2 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 'Development' tab is active, and the 'Document types' sub-tab is selected. The wizard is at 'Step 1: General'. The instructions state: 'Please enter document type display name (for users) and code name (it will be used in your code when necessary)'. The 'Document type display name' field contains 'Computer'. The 'Document type code name' field is split into two parts: 'namespace' containing 'custom' and 'document type' containing 'computer'. A 'Next' button is located at the bottom right of the form area.

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.

Attribute Name	Attribute Type	Field Caption	Field Type	Allow Empty Value	Display in Editing Form
computerID					
ComputerName					
ComputerProcessorType					
ComputerRamSize					
ComputerHddSize					
<b>ComputerImage</b>	<b>File</b>	<b>Image</b>	<b>Upload file</b>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

**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:

[Next](#)

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

[Next](#)

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:

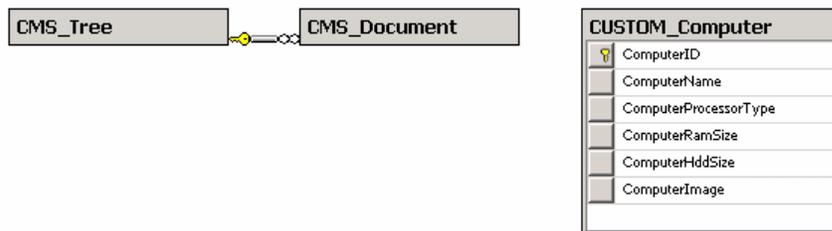
- ▶ The new document type was created.
- ▶ The new editing form was created.
- ▶ The document types were added among allowed child types of the new document type.
- ▶ The sites were selected where this document type can be used.
- ▶ The default queries were created.
- ▶ The default ASCX transformations 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).

### 8.11.3 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>
?maxsidesize=120" />
</div>
```

Click **Save**.

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.

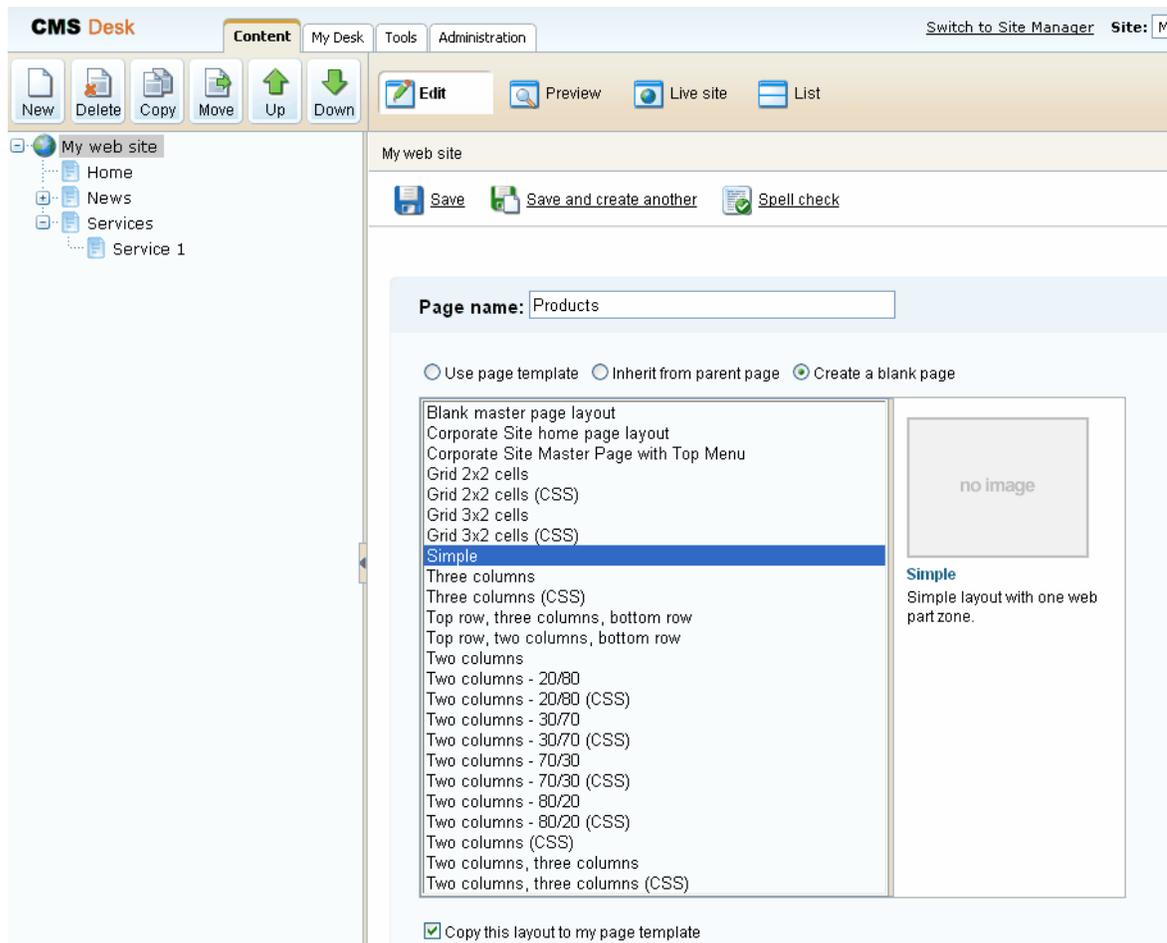
Similarly, you can create an image tag with parameter that ensures automatic resize of the longest side to 120 pixels on the server side:

```
<img src<## GetFileUrl("ComputerImage") %>?maxsidesize=120" />
```

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

### 8.11.4 Page template

Now we get to the final step of this chapter: publishing computer specifications on your web site. Go to **CMS Desk -> Content**, click **New** and choose to create a new **Page (menu item)** document. Enter page name **Products** and choose to **create a blank page** using the **Simple** layout.



Click the **Down** button in the toolbar until you move the new page at the end of the list.

Switch to the **Design** tab and add the **Navigation -> Breadcrumbs** web part into the **zoneLeft** zone. Leave the default values and click **OK**.

Then, add the **Listing and viewers -> Datalist** web part. Set the following property values of this web part:

- **Document types:** custom.computer
- **ORDER BY expression:** ComputerName ASC
- **Transformation:** custom.computer.preview
- **Selected item transformation:** custom.computer.default
- **Show New button:** yes
- **New button text:** Add new computer
- **Show edit and delete buttons:** yes

Click **OK**.

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 the following values:

- **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**.

The screenshot shows the CMS Desk interface with the 'Form' editor active. The left sidebar displays a content tree with 'My web site' expanded to show 'Home', 'News', 'Services', and 'Products'. Under 'Products', 'Office PC Houston' and 'Home PC Dallas' are listed. The main editor area has tabs for 'Page', 'Design', 'Form', 'Product', and 'Properties'. The 'Form' tab is selected, showing fields for 'computerID' (2), 'Computer name' (Office PC Houston), 'Processor type' (Pentium Core 2 Duo), 'RAM (MB)' (1024), and 'HDD (GB)' (120). The 'Image' field shows 'Computer2.jpg' with an 'Upload' button and a 'Browse...' button. The 'Publish from' and 'Publish to' fields are empty, each with a 'Now' button.

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

The screenshot displays a web page for 'Our Business, Inc.' with a blue navigation bar containing 'Home', 'News', 'Services', and 'Products'. Below the navigation bar, there is a 'Products' section with a '+ Add new computer' button. Two product cards are shown: 'Home PC Dallas' and 'Office PC Houston'. Each card features an image of a computer system and 'Edit' and 'Delete' buttons. At the bottom of the page, a footer reads 'This is a sample web site for Kentico CMS'.

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

## Our Business, Inc.

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

[Products](#) > Home PC Dallas

### Home PC Dallas

Processor: Athlon

RAM (MB): 512

HDD (GB): 80

Image:



This is a sample web site for

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

## 8.12 Search page

Kentico CMS allows you to perform index-based search all documents in Kentico CMS repository. For this to work, you need to have a search index created. Go to Site Manager -> Administration -> Smart search and choose to create a **New index** (🗑️). In the following dialog, enter the following details:

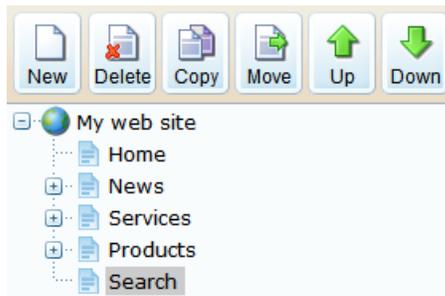
- **Display name:** My web site
- **Code name:** MyWebSite
- **Analyzer type:** Standard
- **Assign index to website MyWebSite:** enabled

Click **OK**. You will be redirected to the index's editing interface. Switch to the **Index** tab and click **Add allowed content** (📄). In the following dialog, enter **/%** into the **Path** field and click **OK**. Switch to the **Sites** tab and make sure that the index is assigned to **My web site**. Click **Save**. Switch to the **Cultures** tab and choose the default culture of your site (typically English - United States). Click **Save**.

Finally, switch to the **General** tab and choose to **Rebuild** (🔄) the index. Once the index gets rebuilt, the site is prepared to be searched.

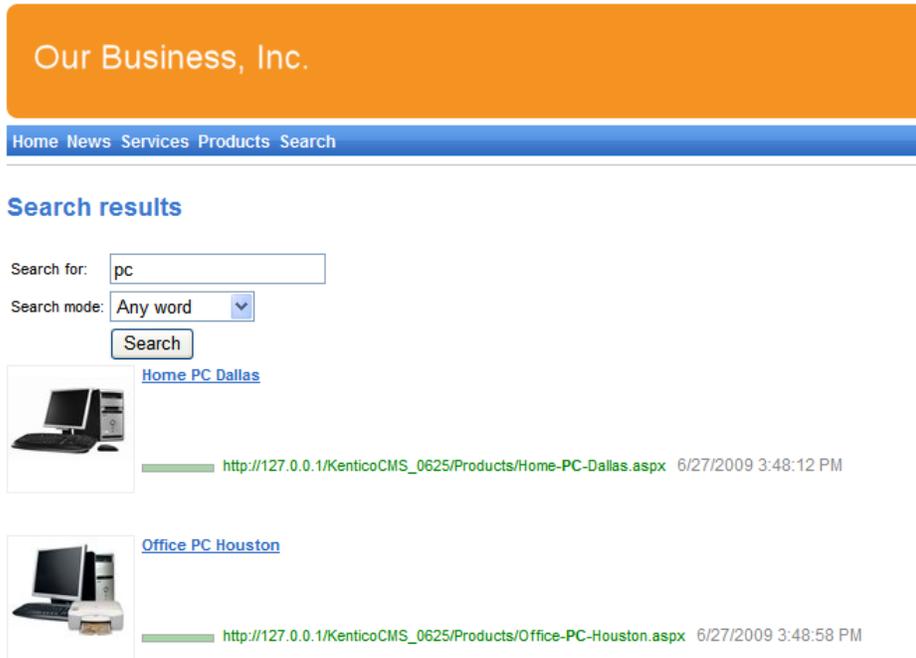
Now we need to create the search page. Click root, click **New** and choose to create a new **Page (menu item)**. Enter page name **Search** and choose the **General/Search** page template. Click **Save**.

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



Switch to the **Design** tab, choose to **Configure** (⚙️) the **SearchDialogWithResults** web part and choose the previously created index in its **Indexes** property. Click **OK**.

Now switch to the live site and view the page. Enter **PC** in the **Search for** box and click **Search**.



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.

## 8.13 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 (menu item)**. Enter page name **Partners** and choose to use the page template **Templates with editable regions -> Simple text**. Click **Save**. Click the **Down** arrow in the main toolbar until you move the **Partners** page at the end of the list.

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

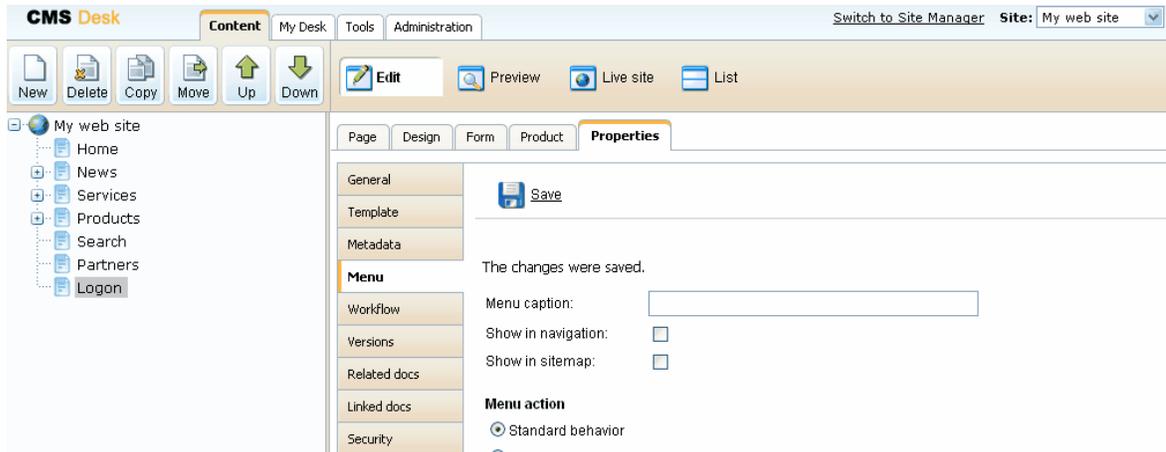
Click **Save**.

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

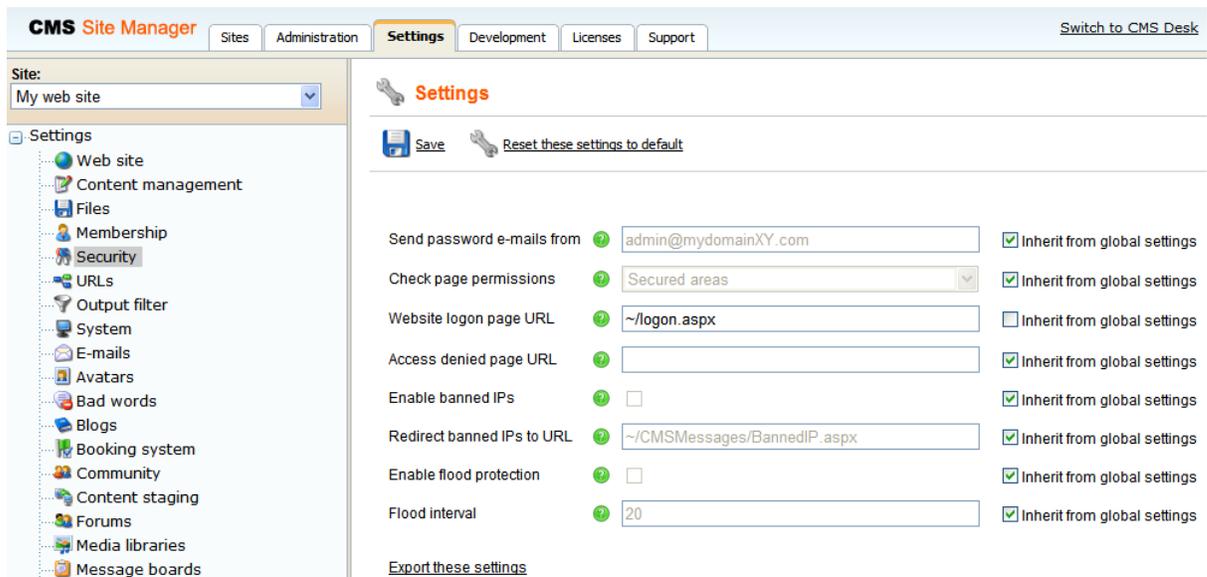
## Configuring the logon page

Now we need to create the logon page. We will use a pre-defined page template that you can further customize if necessary. Click **root** and click **New**. Choose to create a new **Page (menu item)**. Enter page name **Logon** and choose the page template **Membership and security -> Log-on page with registration form**. Click **Save**.

Click **Properties -> Menu** and uncheck boxes **Show in navigation** and **Show in sitemap**. Click **Save** and use the **Down** button to move the **Logon** page to the end of the document list in the content tree.



We will also 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 **Website logon page URL** to **~/logon.aspx**, which is the relative URL of our logon page (from the web application root). Click **Save**.



## Configuring the Sign out button

Go to **CMS Desk** -> **Content** -> select the root document -> click **Design**, add the following web parts to the **zoneMain** web part zone and set their properties:

### Membership -> Current user

- Content before: `<div style="text-align:right">`

### Membership -> Sign out button

- Content after: `</div>`

Then sort the web parts in the **zoneMain** zone in the following order (you can drag and drop web parts using mouse):

- Current user
- Sign out button
- Page placeholder

The screenshot shows the CMS Desk interface in Design view. The page title is "Our Business, Inc.". The page structure is as follows:

- Page: Design (selected), Form, Product, Master page, Properties
- Page template: Blank master page
- zoneMenu
  - cmsmenu
- zoneMain
  - currentuser
  - SignOutButton
  - pageplaceholder

The page content area contains the text "Our Business, Inc." and a footer that says "This is a sai".

**Sign out** of CMS Desk. 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, Products, Search, Partners. The main content area is divided into two sections. On the left, under the heading 'Log on', there are input fields for 'User name:' and 'Password:', a 'Remember me' checkbox, a 'Log on' button, and a link for 'Forgotten password'. On the right, under the heading 'Not a member yet? Sign up now!', there are input fields for 'First name:', 'Last name:', 'E-mail:', 'Password:', and 'Confirm password:', along with a 'Register' button.

This is a sample web site for Kentico CMS

Now you need either to sign in as administrator or register as a new user. 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. The navigation bar now includes a 'Service 1' button. The main content area has a message: 'This is a secured page for partners.' In the top right corner, it displays 'Current user: administrator (administrator)' and a 'Sign out' button. The footer text 'This is a sample web site for Kentico CMS' is also visible.

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**



## 9 Further steps

### 9.1 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>