

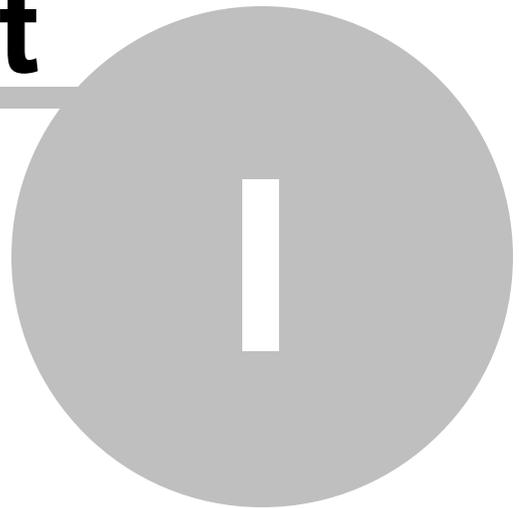
Kentico CMS 6.0 Tutorial

Table of Contents

Introduction	5
Kentico CMS Overview.....	5
Installation	7
Prerequisites.....	7
Setup installation.....	8
Web application installation.....	8
Database setup and Corporate Site.....	12
Managing content	19
User interface overview.....	19
Editing home page content.....	21
Creating a new page.....	23
Inserting an image.....	27
Creating a link.....	28
Creating a news item.....	31
Site Development Overview	35
Site Development Overview.....	35
Creating pages using the portal engine	40
Page templates.....	40
Modifying a page template.....	42
Modifying the home page structure.....	43
Customizing pages using widgets.....	47
Creating a new page template.....	53
Re-using the page template.....	56
Visual inheritance.....	58
Configuring visual inheritance.....	61
Web part containers.....	65
Adding custom code to your web site.....	67
Developing web parts and widgets	70
Web part development overview.....	70
Creating a simple web part.....	73
Web part modifications.....	79

Creating widgets.....	80
Managing styles and design	85
CSS styles.....	85
App themes.....	87
Menu design.....	88
Creating a new site using the portal engine	91
Overview	91
Creating a new web site using the wizard.....	91
Creating a CSS stylesheet.....	98
Master page.....	100
Main menu.....	105
Page placeholder.....	106
Home page.....	107
Container for home page text.....	109
News page.....	111
Services page.....	114
Products page.....	117
Overview	117
New document type	117
Transformations	123
Page template	125
Search page.....	128
Secured section for partners.....	130
Further steps	136
Further steps.....	136

Part



Introduction

1 Introduction

1.1 Kentico CMS Overview

Kentico CMS for ASP.NET helps you create powerful dynamic websites 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 websites.

This document was written for evaluators and new users. It's intended for developers who create the websites. 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:

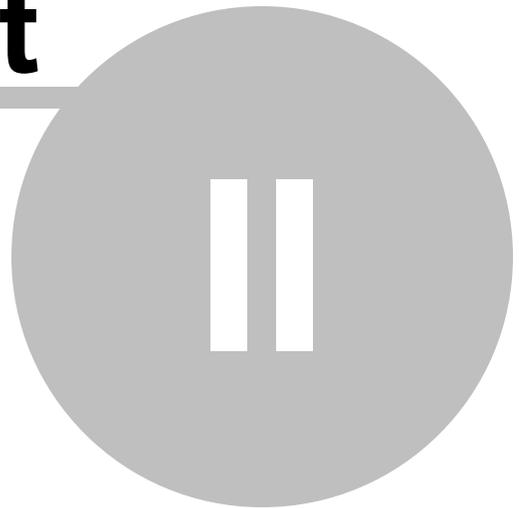
- Developer's Guide
- Controls Reference
- Web parts or Widgets Reference
- API Reference
- Database Reference



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



Installation

2 Installation

2.1 Prerequisites

Before you start the installation, please make sure you have the following software installed. Other configurations may work too, but the system was not tested on them.

Server-side Requirements

- Windows XP, 2003, 2008, 2008 R2, Windows Vista Home Premium/Business/Enterprise/Ultimate or Windows 7 (both 32bit and 64bit)
- Microsoft .NET Framework 3.5 SP1 or higher
- Microsoft Internet Information Services (IIS) or Visual Studio/Visual Web Developer 2005/2008 built-in web server
- Microsoft SQL Server 2005, 2008 (including free SQL Server Express Edition)

Hosting Requirements

- ASP.NET 3.5 SP1 (or higher) and Microsoft SQL Server 2005/2008 support
- Medium-trust or full-trust permissions for the ASP.NET application
- If the server uses medium trust, ASP.NET AJAX 1.0 must be installed on the server.
- If the application uses .NET Framework 3.5 SP1 and is hosted in a medium trust environment, it is necessary to have [Microsoft Chart Controls](#) installed on the server.
- It's recommended that your hosting plan comes with 125 MB or more memory and 100+ MB database.

You can use your favorite hosting provider or choose from our [hosting partners](#).

Development Tools

If you want to create custom web parts or integrate custom code, you need **Visual Studio 2005/2008/2010** or **Visual Web Developer 2005/2008/2010 Express Edition**.

Supported Client Browsers for Content Editors

- Internet Explorer 7, 8, 9
- Firefox 3.6, 4.0, 5.0
- Chrome 12
- Safari 4.0, 5.0, 5.1

Supported Client Browsers for Site Visitors

- Internet Explorer 6.0+
- Firefox 3.6+
- Chrome 12+
- Safari 4.0+
- Opera 10.50+

(the visitor browser requirements depend on functionality used on the website)

Required experience

Although Kentico CMS allows you to create dynamic websites without programming, you may need to create custom web parts or add custom code when developing a more complex website. You should be able to create a simple application in ASP.NET 3.5 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:

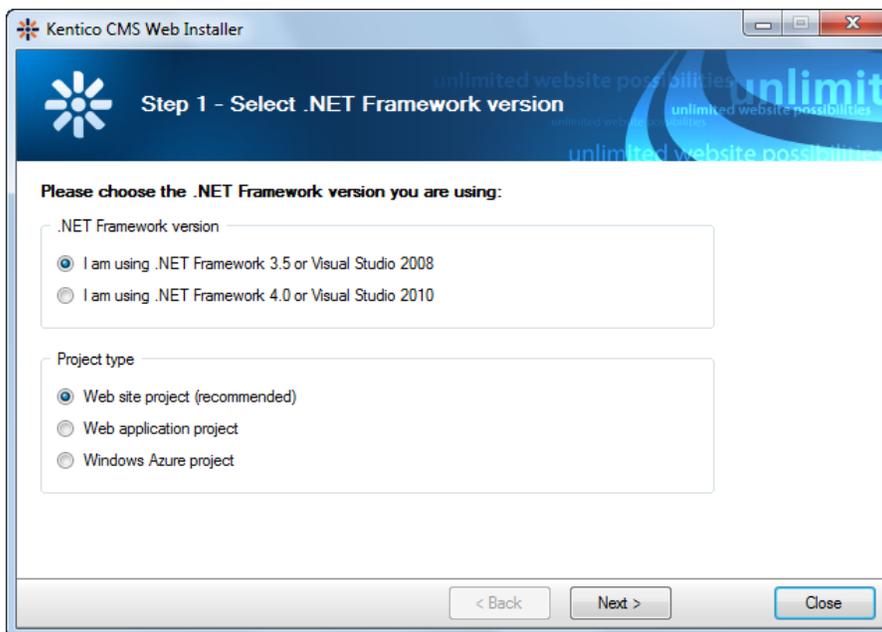


Read and accept the license agreement and click **Next**. Choose the installation location of the binary files and documentation on your disk. After the setup completes the installation, check **Launch Kentico Web Installer** and click **Finish**.

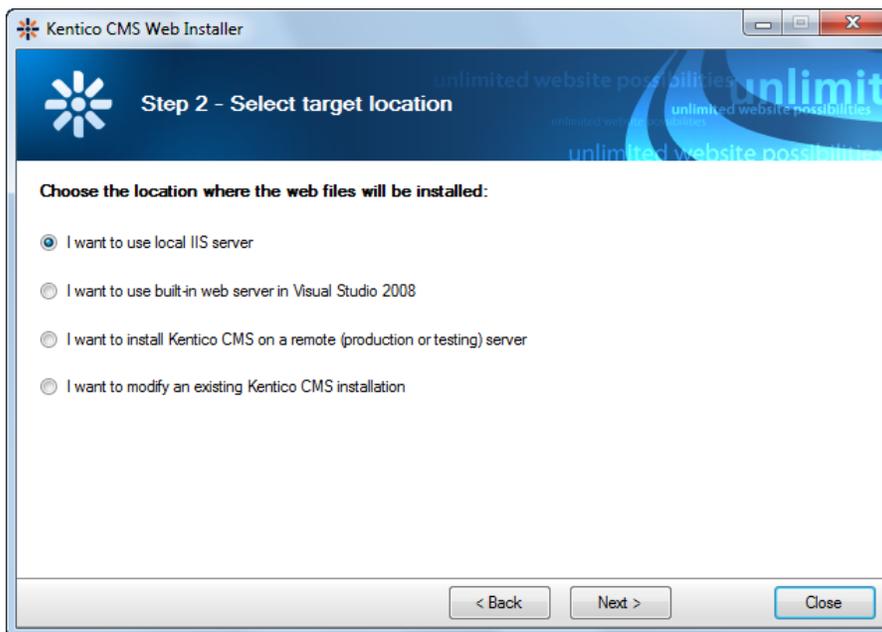
2.3 Web application installation

Now you should see the Kentico Web Installer. If it has not started automatically, you can always run it from **Start menu -> All Programs -> Kentico CMS -> Kentico Web Installer**.

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



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

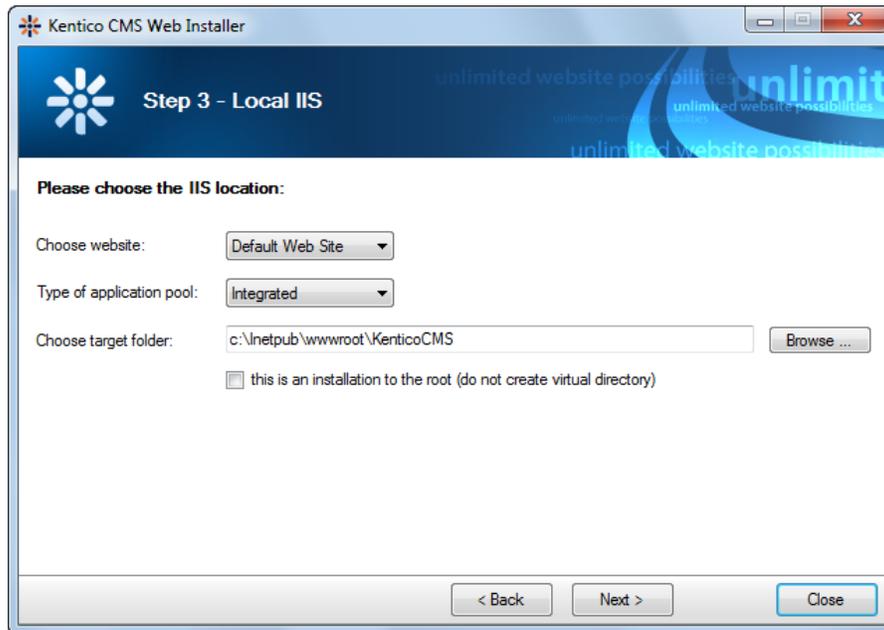


IIS installation

If you choose the IIS server, you can choose the website 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**.

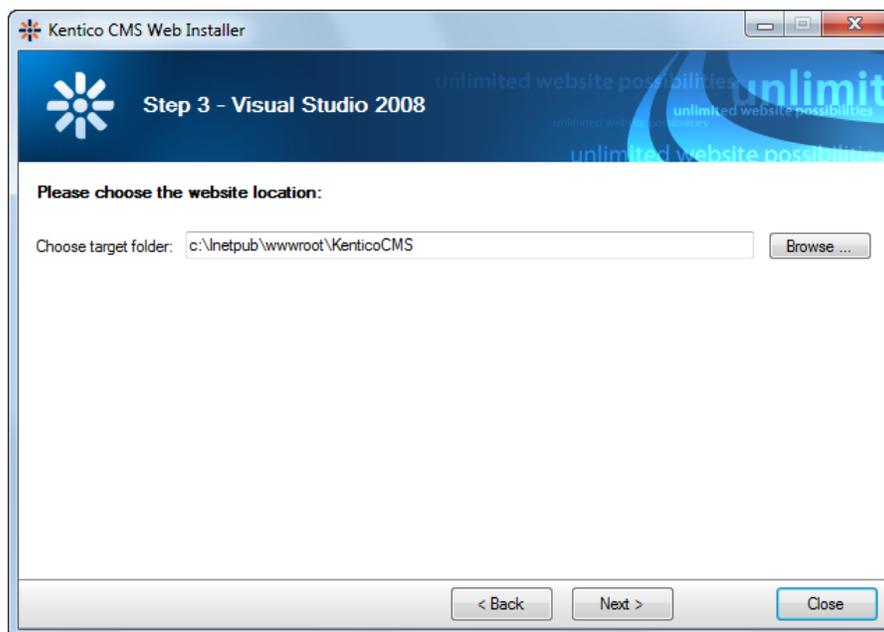
Please note: if you are installing Kentico CMS into the root of your website (such as <http://www.domain.com>) and do not wish to create a virtual directory (such as <http://www.domain.com/cms>), please

check the **This is an installation to the root (do not create virtual directory)** check-box.



Visual Studio installation

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

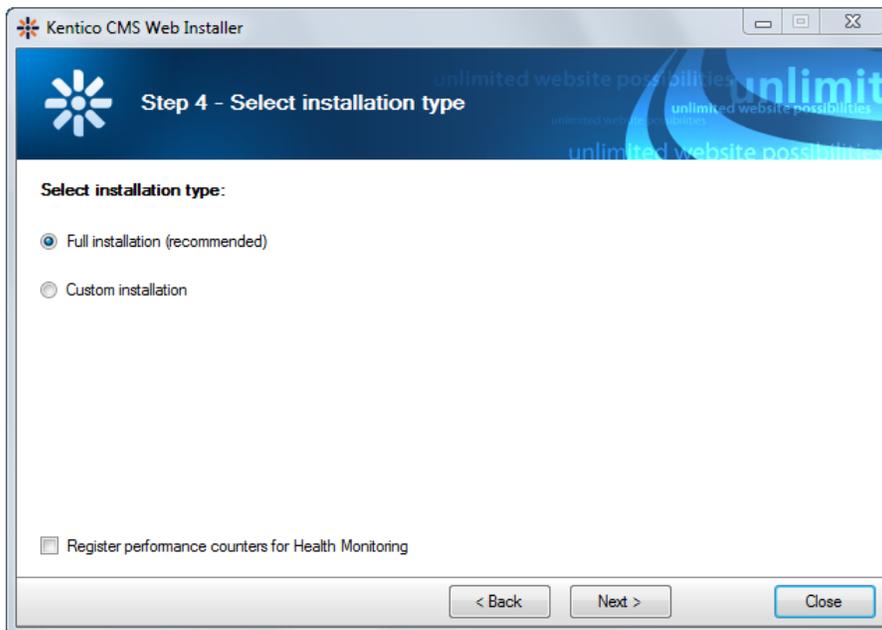


Type of installation

No matter if you chose the IIS or VS installation, the next step after specifying the target folder is the selection of the 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.

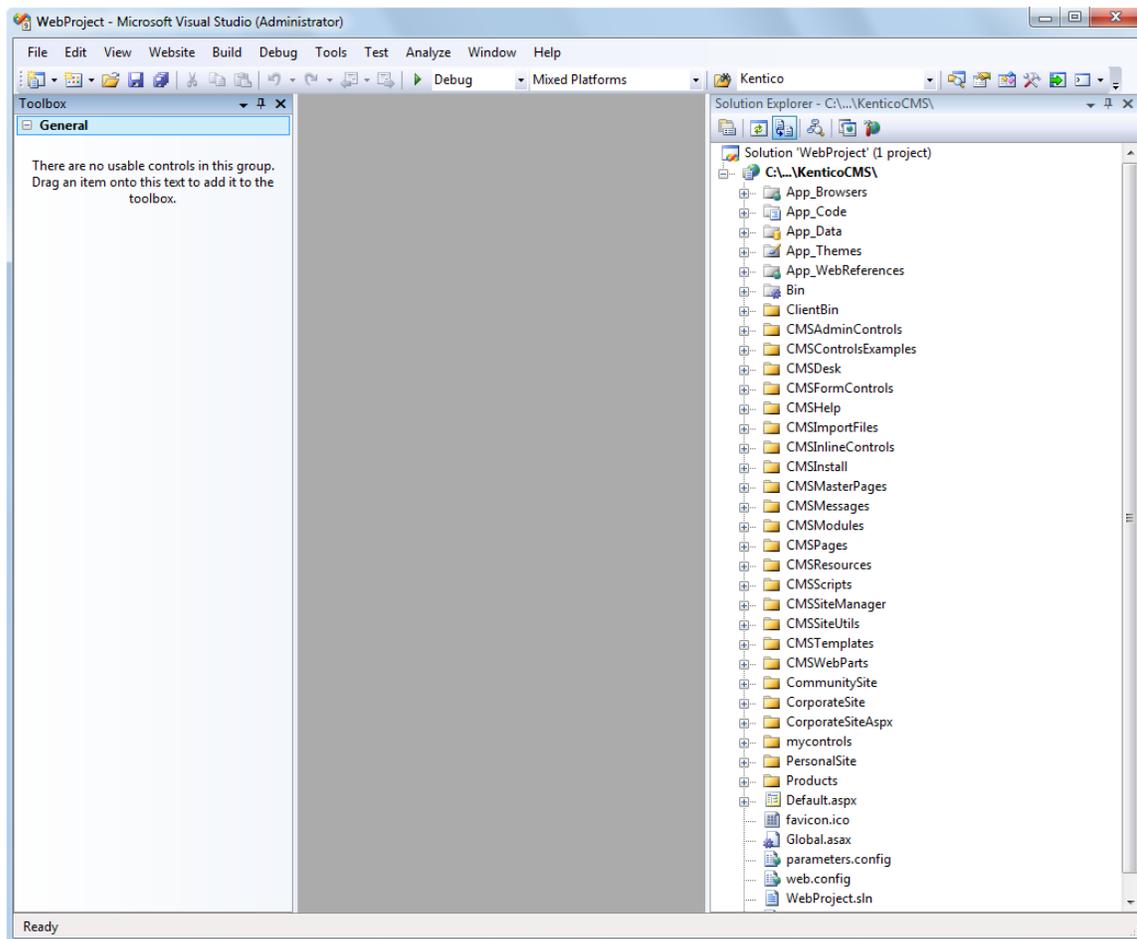
Click **Next** to start the installation process.



After the setup copies all files, you will see a link that opens the web application in your browser or the web project in Visual Studio. **Click the link.**

Opening the website in Visual Studio

If you chose Visual Studio installation, the project is opened in Visual Studio:



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



If you cannot open the website in Visual Studio

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.

In the first step, choose the SQL Server name or IP address. If you are using SQL Server 2005 Express Edition, the default server name is `.\SQLEXPRESS` or `(local)\SQLEXPRESS`.

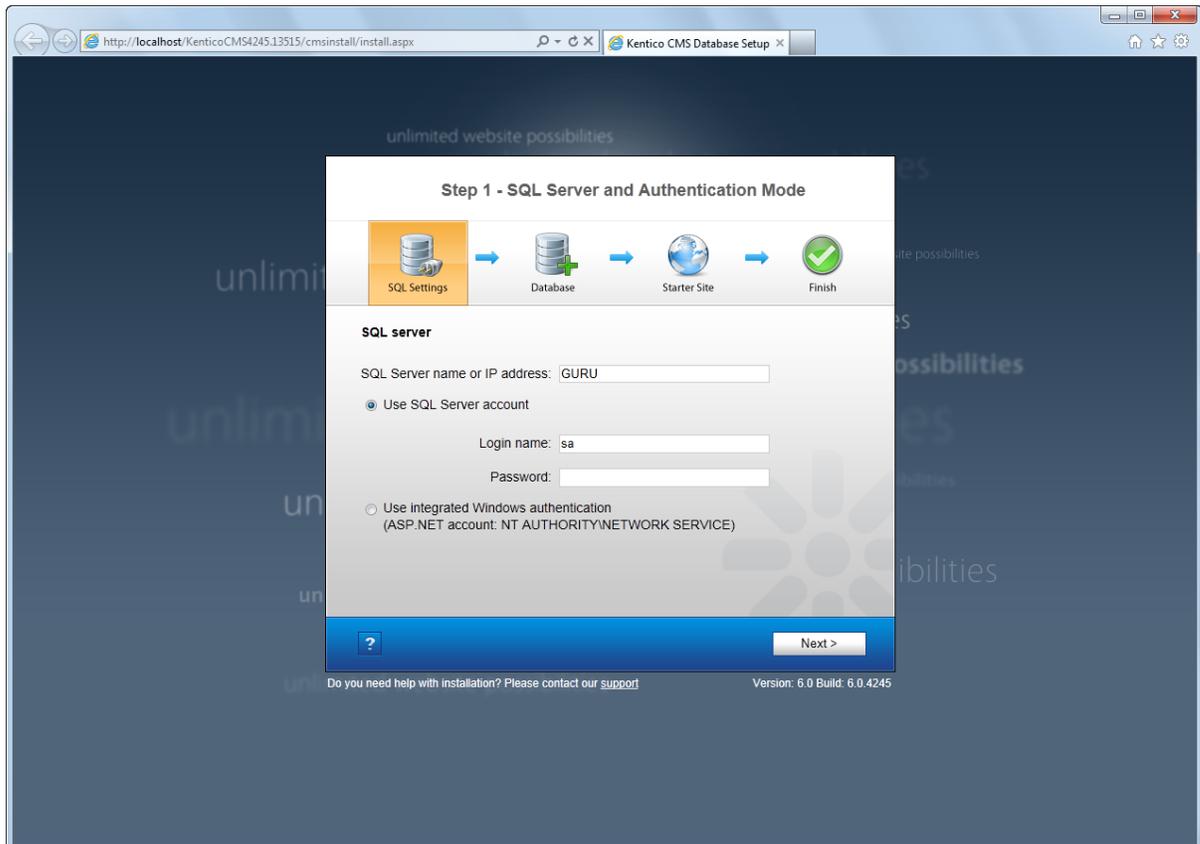
You can use either SQL Server authentication (recommended) or integrated Windows authentication.

- 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.

Permissions for creating a new databases or for creating database objects in an existing database must be granted to the account.

Click **Next**.



In the **Database Instance** step, choose **Create a new database**, enter the name of the new database into the **New database name** field and click **Next**.

Step 2 - Database Instance



Database

Create a new database

New database name:

Use an existing database

Existing database name:

Create Kentico CMS database objects

[Show advanced options](#)

[?](#)

The database creation log will be displayed.

Step 3 - Database Creation Log

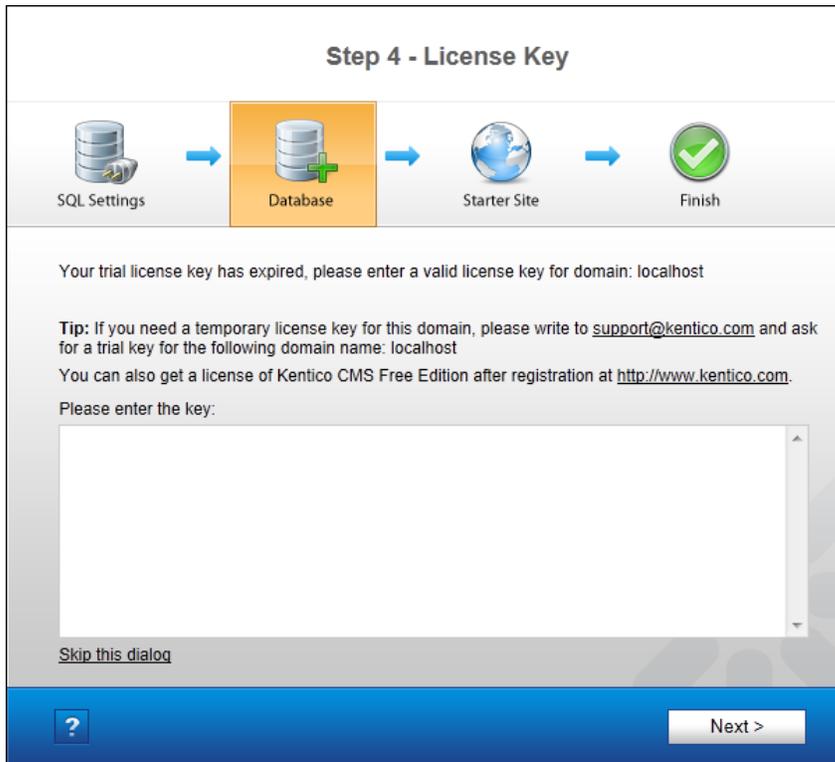


Creating database objects...

```
proc_forums_forum_setpermissions
proc_forums_forum_removedependencies
proc_forums_forum_movenodeup
proc_forums_forum_movenodedown
proc_forums_forum_initnodeorders
proc_forums_forumpost_updatethreadcounts
proc_forums_forumpost_updatepostcounts
proc_forums_forumpost_unstickthread
proc_forums_forumpost_stickthread
proc_forums_forumpost_removedependencies
proc_forums_forumpost_moveupstickythread
proc_forums_forumpost_movedownstickythread
proc_forums_forumgroup_removedependencies
proc_forums_forumgroup_movenodeup
proc_forums_forumgroup_movenodedown
```

When the database is created, you will be asked to enter your license key. If you do not have a license

key yet, click **Next** to continue in trial mode. The functionality of the trial mode is the same as the full version.



In the **Starter Site** step, you can choose from the following options:

- **Choose a starter site:**
 - **Corporate Site (portal engine)** - this option installs the sample corporate site. This is **recommended** for most users, especially **for evaluators**.
 - **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.
 - **Intranet Portal** - ready-to-use solution for company intranets with support of departments, workgroups, project management, etc.
 - **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 are starting a new site from scratch.
- **Import an existing Kentico CMS website** - use this option if you already created a website with Kentico CMS and need to import it into the new installation (e.g. on the production server).

For the purposes of this guide, please select the sample **Corporate Site** and click **Next**. You will see the confirmation and a link to your new website:

Step 5 - Starter Site



SQL Settings → Database → Starter Site → Finish

Choose starter site

Corporate Site

This is a web template for a general corporate site. It's used as a showcase of Kentico CMS capabilities and it can be used as a starting site that you modify as needed. It uses the portal engine and it's the recommended choice for developers who are new to Kentico CMS.

E-commerce Site

This is a web template for a simple E-commerce site. It's used as a showcase of Kentico CMS E-commerce module capabilities and it can be used as a starting site that you modify as needed. It uses the portal engine and it's the recommended choice for developers who are new to Kentico CMS.

Continue to the New site wizard

Import existing Kentico CMS website

[Next >](#)

A log showing the creation of the site will be displayed. When it's finished, you will be shown the final step that you can see in the screenshot below. Click the **Continue to the new web site** link.

Step 7 - Finished

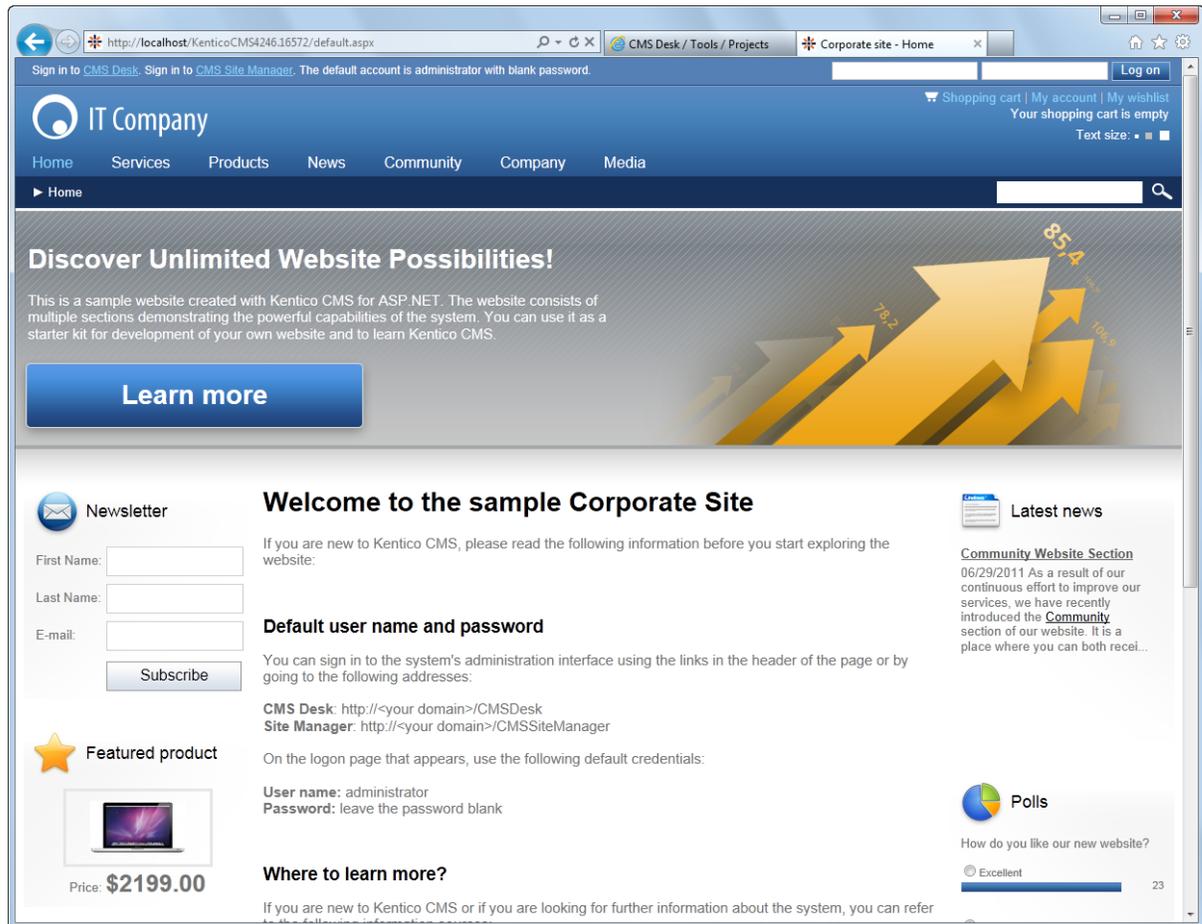


SQL Settings → Database → Starter Site → Finish

The site has been created successfully.

[Continue to the new website](#)

You will be redirected to the title page of the sample Corporate Site:



The screenshot shows a web browser window displaying the homepage of a sample Corporate Site. The browser address bar shows the URL: `http://localhost/KenticoCMS4246.16572/default.aspx`. The page header includes the site logo "IT Company" and navigation links: Home, Services, Products, News, Community, Company, Media. A shopping cart icon indicates "Your shopping cart is empty".

The main content area features a large banner with the text "Discover Unlimited Website Possibilities!" and a "Learn more" button. Below the banner, there are several sections:

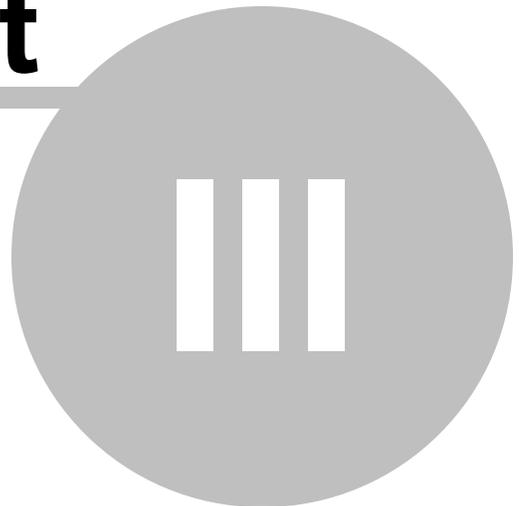
- Newsletter:** A form with fields for "First Name", "Last Name", and "E-mail", and a "Subscribe" button.
- Welcome to the sample Corporate Site:** A section providing information for new users, including default user name and password, and links to CMS Desk and Site Manager.
- Default user name and password:** A section stating the default user name is "administrator" and the password should be left blank.
- Where to learn more?:** A section providing information for new users or those looking for further information about the system.
- Latest news:** A section titled "Community Website Section" dated 06/29/2011, mentioning improvements to the website's services.
- Featured product:** A section featuring a product image and a price of "\$2199.00".
- Polls:** A section titled "How do you like our new website?" with a poll showing "Excellent" as the selected option and a count of 23.



Sample website

The Corporate Site website is only an example of a website 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



Managing content

3 Managing content

3.1 User interface overview

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

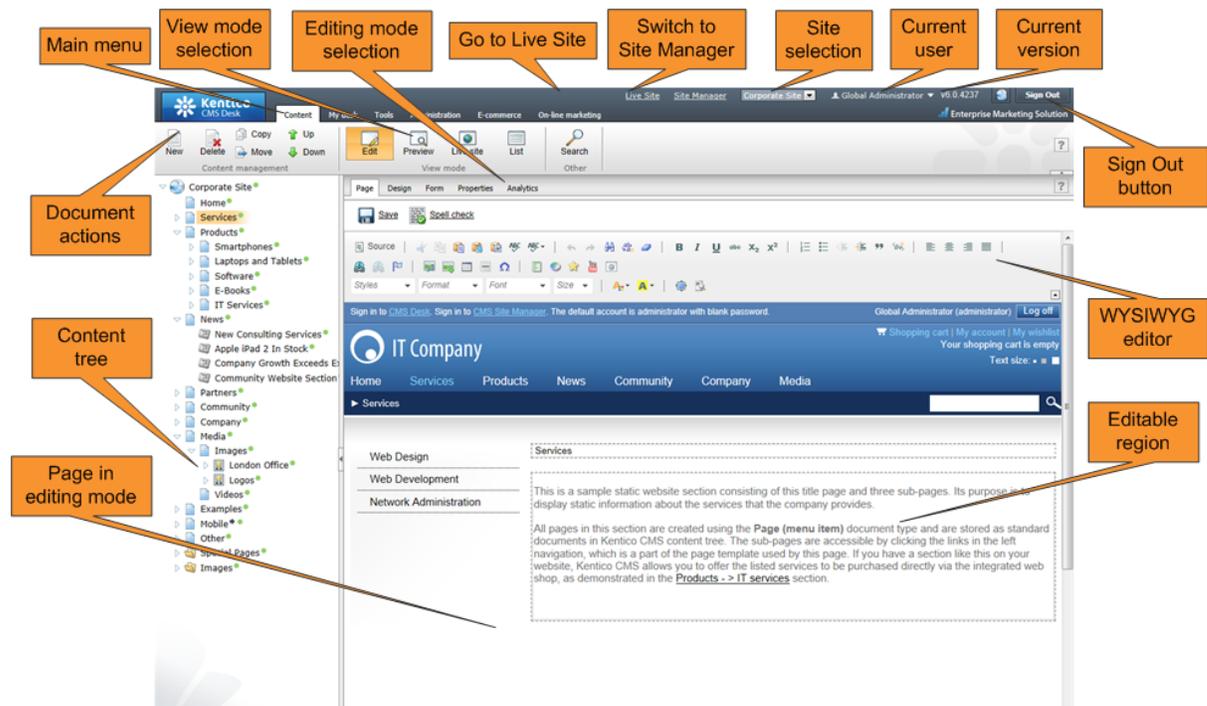


Default user name and password

The default user name is **administrator**, the default password is **blank (no password)**.

It's highly recommended that you change the password before you publish the website 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 user interface consists of the following main sections and features:

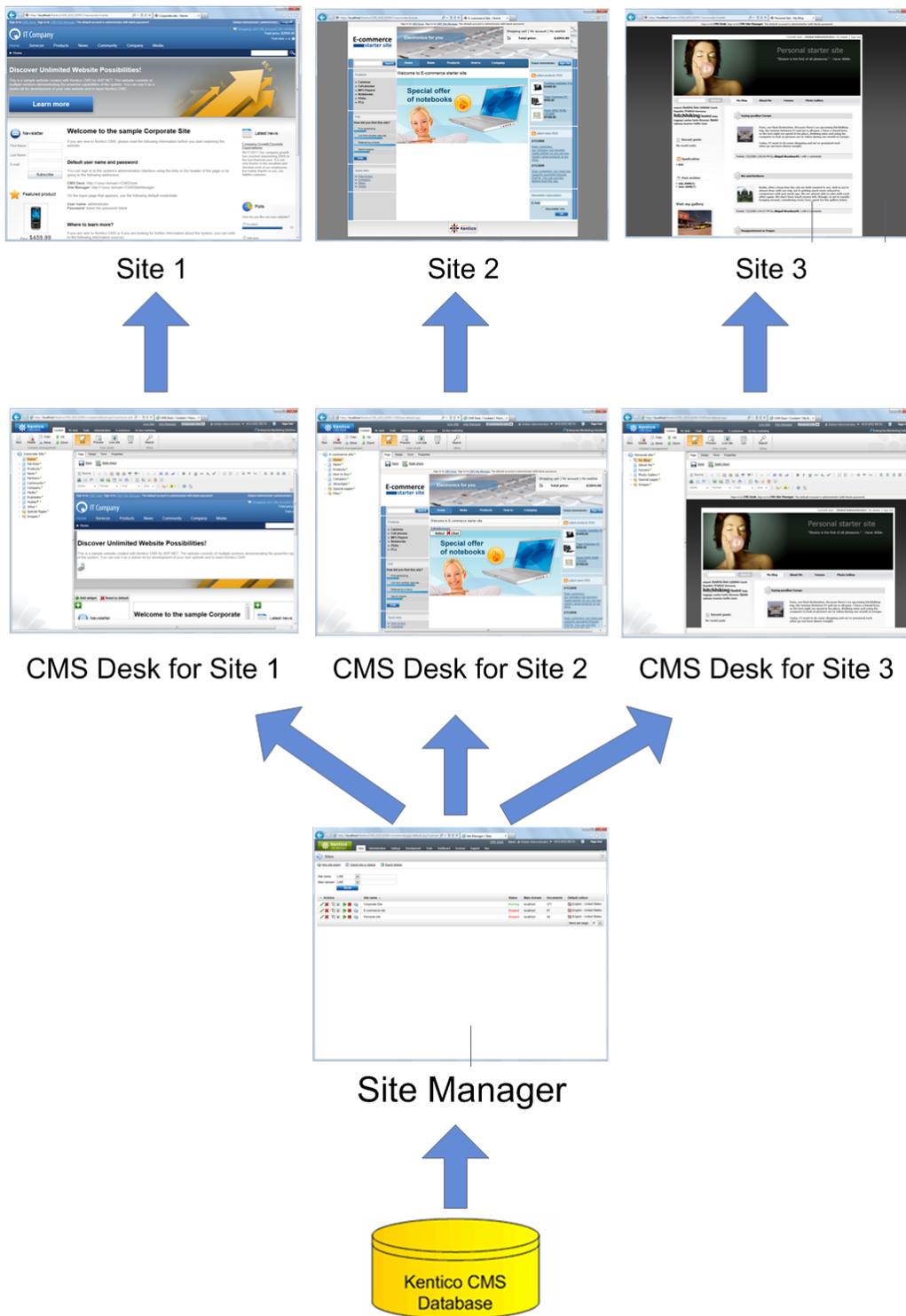
- **Main menu** with Content, My Desk, Tools and Administration tabs.
- **Content tree** that represents the site map of the website and allows you to organize the structure of documents and choose document that appears on the right side of the screen.
- **Document actions** toolbar with buttons for creating new documents, deleting, copying, moving and sorting documents.

- **View mode selection** - allows you to choose between editing, preview, live site view and list view.
- **Editing mode selection** - you can choose to edit page content, design the page template, edit the document fields, product properties or document properties.
- **Live Site** - this action redirects you to the title page of the currently edited website, logged under the same user account that you used to log into the user interface. This is a more convenient way than using the **Sign out** button and logging in on the live site afterwards.
- **Site Manager** - redirects you to Site Manager, the other part of the system's user interface. This option is only available to global administrators.
- **Site selection** - this drop-down list is used to select the currently edited website. Only those websites that the current user can edit are available in the drop-down list.
- **Current user** - user name of the current user.
- **Current version** - version of Kentico CMS.
- **Sign Out button** - clicking this button log you off the user interface and redirects you to the title page of the live site. This button is only displayed if *Forms authentication* is used. When using *Windows authentication*, the link is not displayed.
- **WYSIWYG editor** - allows you to edit text stored in **Editable regions**, change text formatting or insert graphics into the text.
- **Page in editing mode** - this is where you can view and edit the document selected in the content tree, in the mode selected in the view mode and editing mode toolbar.

CMS Desk and Site Manager

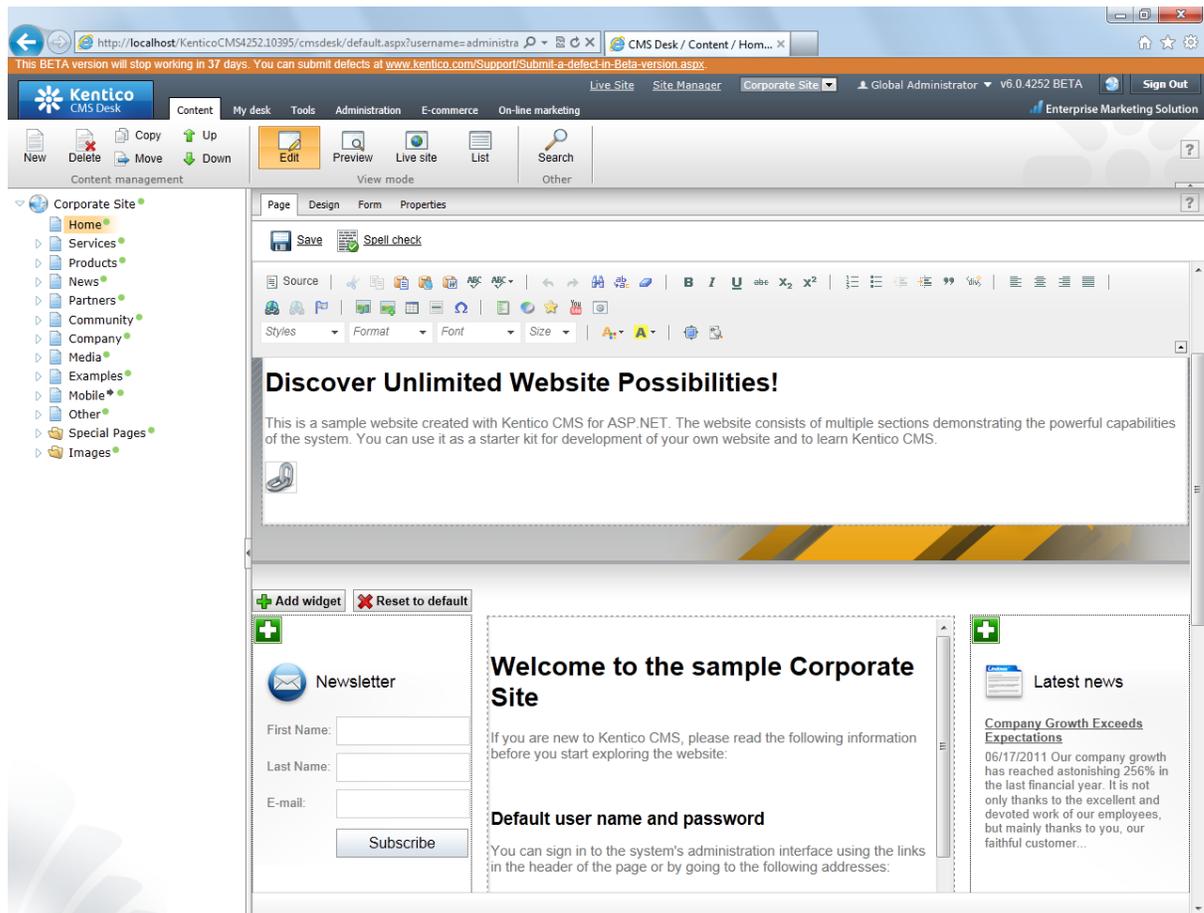
CMS Desk allows content editors to edit content of a single website. Developers and site administrators who need to manage settings, code and configuration of all websites, can also use the **Site Manager** interface.

The Site Manager interface is accessible either through the **<http://<domain>/<virtualdirectory>/KenticoCMS/CMSSiteManager>** URL, by clicking the **Site Manager** link at the top of the CMS Desk user interface or the **Sign in to Site Manager** link at the top of the live site. The following figure shows how the database, Site Manager, CMS Desk and websites 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 editing mode with an 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 the 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 document you want to create under the currently selected document:

 **New document**

Please select new document type:

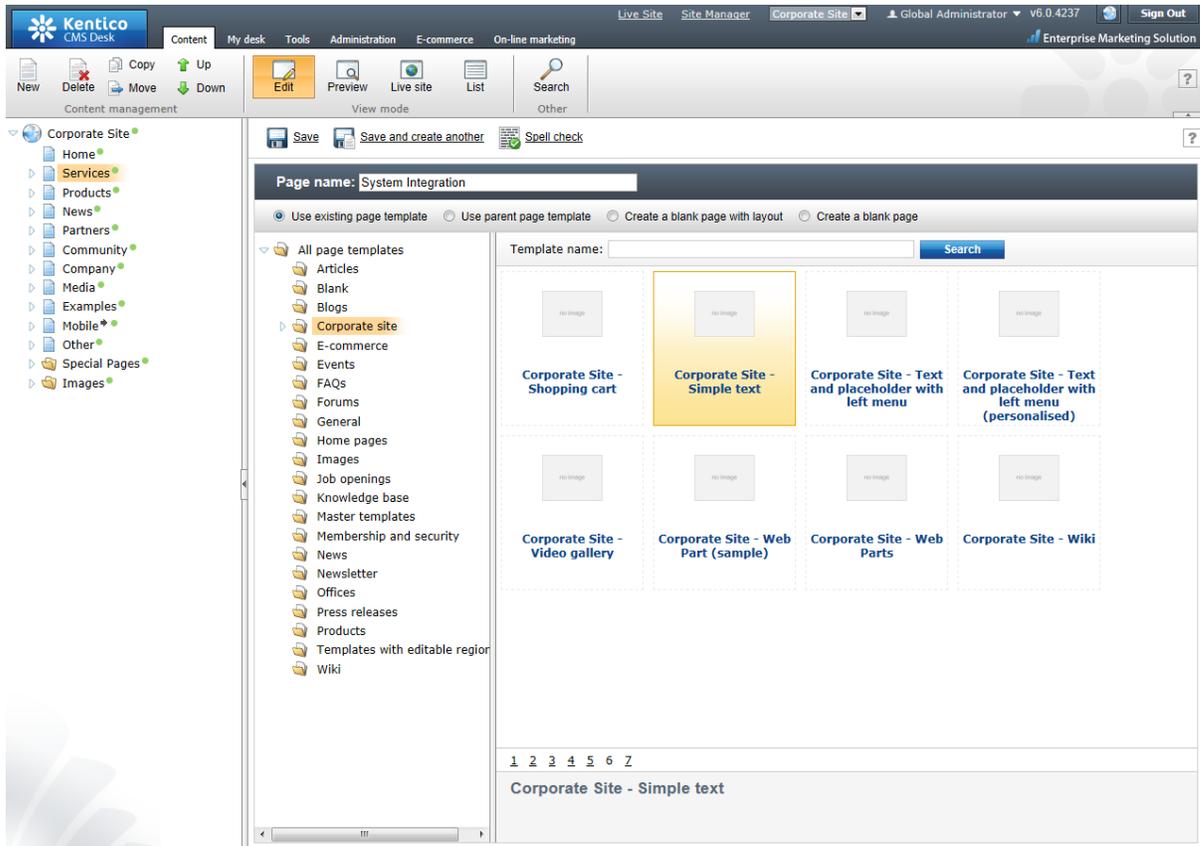
Document type name :

-  [Page \(menu item\)](#)
-  [Article](#)
-  [Blog](#)
-  [Cell phone](#)
-  [E-book](#)
-  [Event](#)
-  [Event \(booking system\)](#)
-  [FAQ](#)
-  [File](#)
-  [Folder](#)
-  [Image gallery](#)
-  [IT Service](#)
-  [Job opening](#)
-  [Knowledge base article](#)
-  [Laptop](#)
-  [News](#)
-  [Office](#)
-  [PDA](#)
-  [Press release](#)
-  [Product](#)
-  [Product - Cell phone](#)
-  [Product - Laptop](#)
-  [Simple article](#)
-  [Smartphone](#)
-  [Software](#)

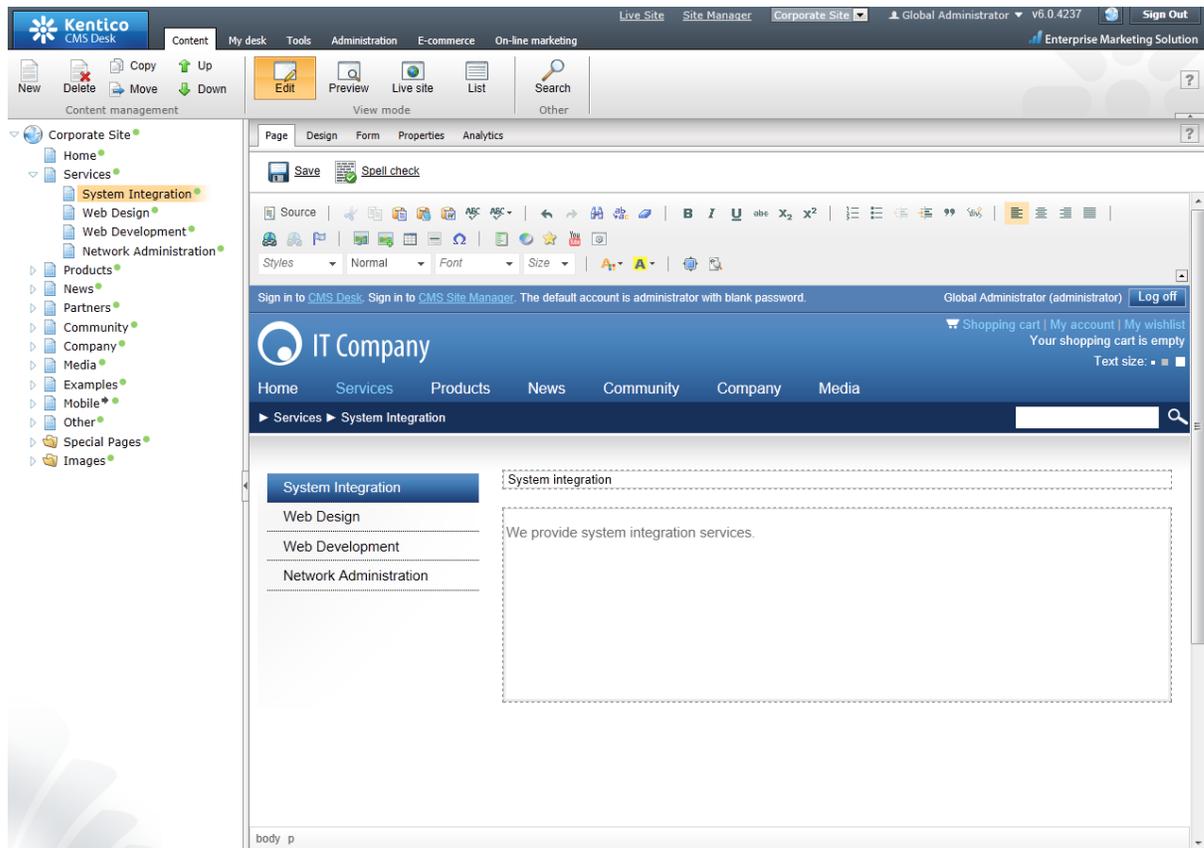
⏪ < 1 2 > ⏩

 [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 **Corporate Site -> Corporate Site - Simple text** template:



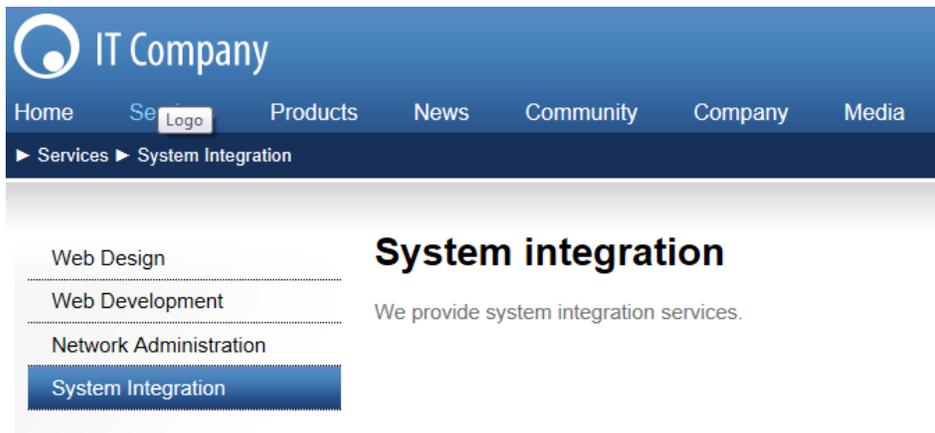
Click  **Save** to create the new page. The page is now created in the content tree and you can edit page content on the right. Enter some text in the editable regions and click  **Save** again.



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

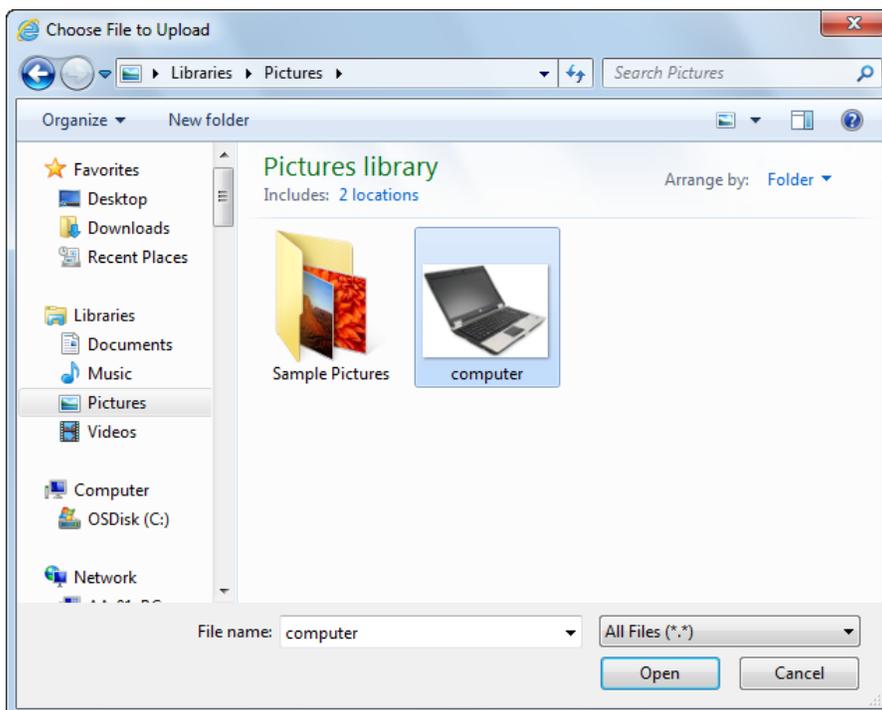


You have learned how to create a new page based on a pre-defined page template.

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 6.0 interface. The top navigation bar includes 'Live Site' and 'Site Manager' links. The main toolbar contains 'Save', 'Spell check', 'Source', and 'Live site' buttons. The content area shows a navigation menu with 'System Integration' highlighted, and a text block with the text 'We provide system integration services.' and an image of a laptop.

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.



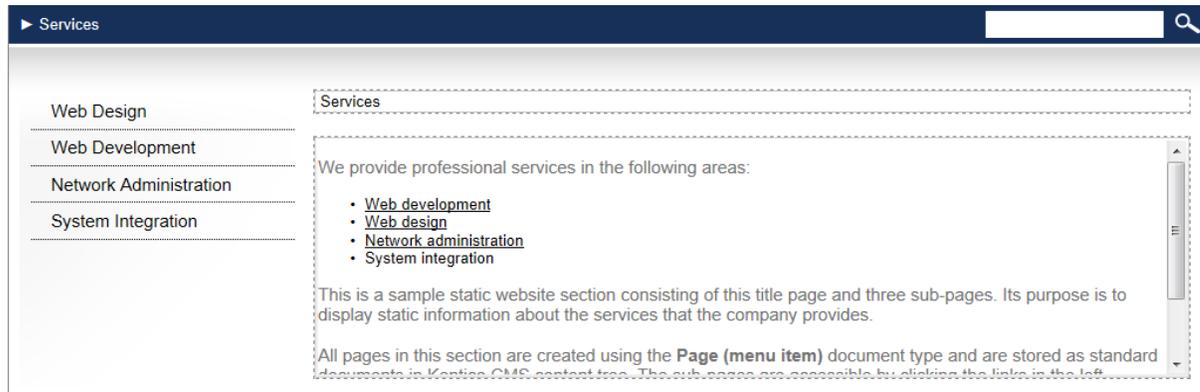
Allowing pop-ups for the website

If you are 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.

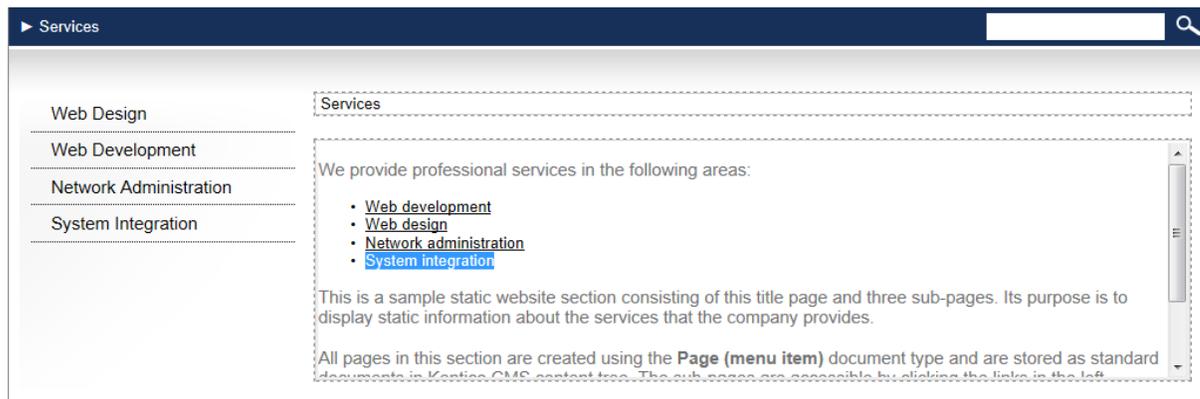
3.5 Creating a link

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.

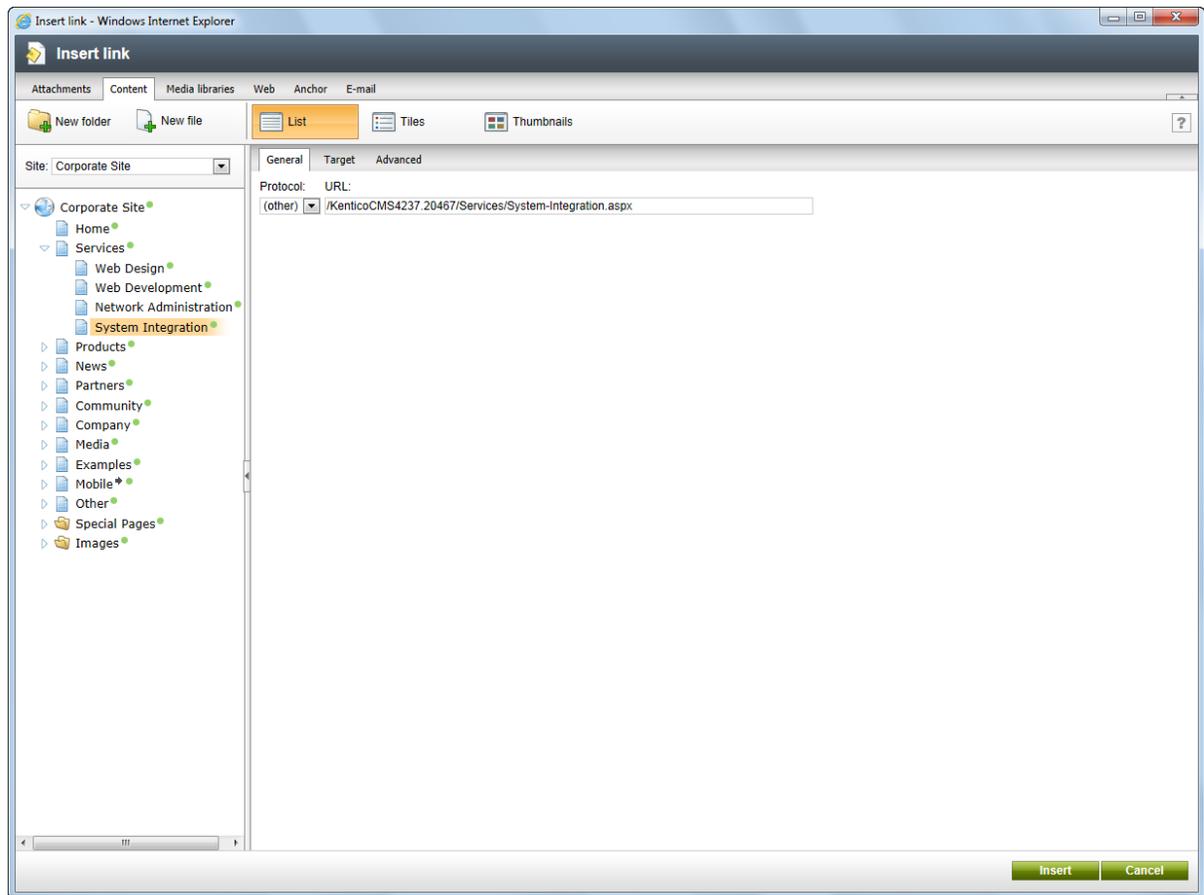
Add a new item in the bulleted list called **System integration**:



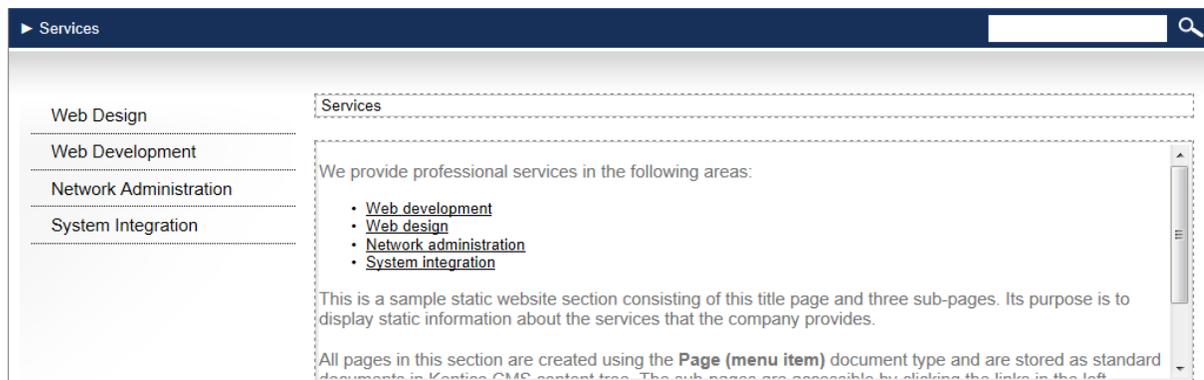
Select the whole line:



... 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:



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

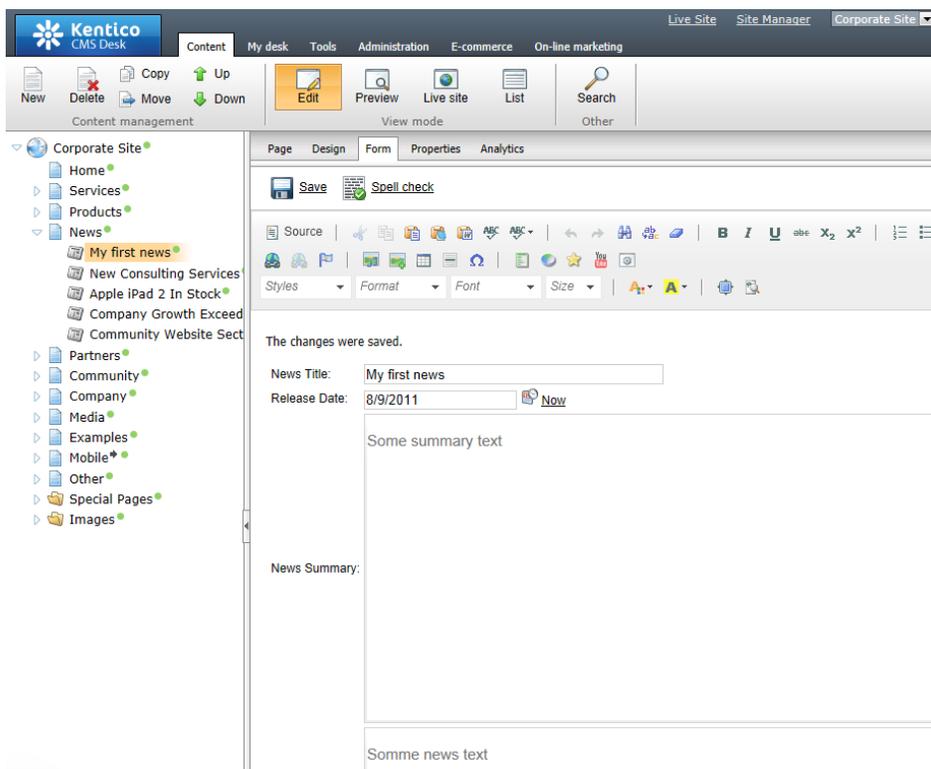
You have learned how to create a link between pages.

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 editable regions on the page, but rather 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 on both the **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 aspects of a 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
Content structure	Simple content structure, only text content.	Complex content structure, typed data, such as text, date-time, numbers, etc.
Validation	Only basic validation rules for minimum and maximum length.	Complex validation rules, including regular expressions and custom form controls with custom validation code.
Display	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.
Storage	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.
Examples of use.	Home page, contact page. Generally: pages with	News, product specification, event details, job openings, etc.

	<p>simply structured or unstructured text-based content.</p> <p>The editable regions are usually used for documents of type Page (menu item).</p>	<p>Generally: pages with structured content where you need to separate content from design and keep the content in its original data type.</p> <p>Form-based content is usually used for documents of type News, Product, Article, etc.</p>
--	--	--

Part

IV

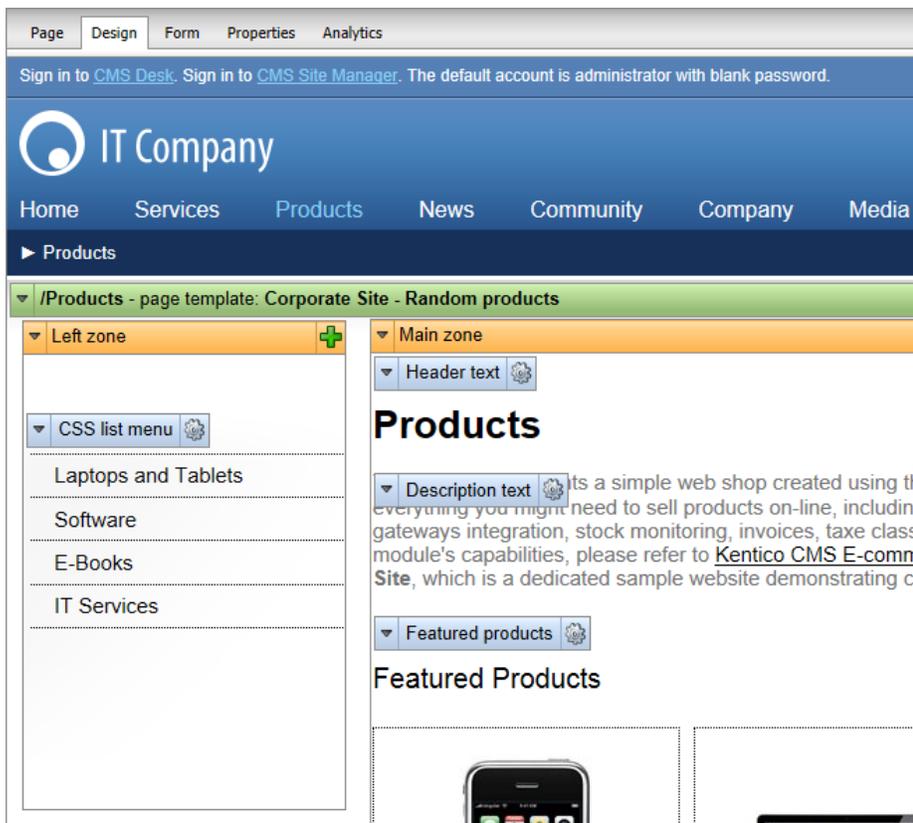
Site Development Overview

4 Site Development Overview

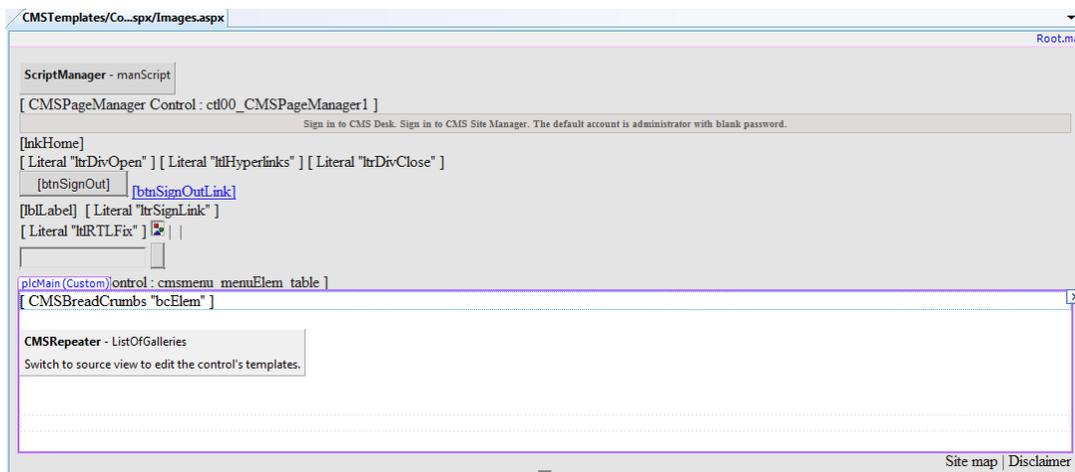
4.1 Site Development Overview

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

- **Portal Engine** - this model allows you to build websites 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 websites using web parts in the **browser-based** user interface.



- **ASPX Templates** - this model can be chosen by advanced ASP.NET developers who prefer to create the website using standard ASP.NET architecture and using standard development tools, such as **Visual Studio**. 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 are a hard-core .NET developer and do not trust portal engines, you may want to use ASPX templates.

If required, both models can be combined on a single website. For example, you can integrate ASPX templates into a portal engine website, or even custom ASPX pages implementing your own applications. On the other hand, special areas can be defined on ASPX templates where editing through the portal engine is possible.

The following table compares the portal engine and ASPX templates:

	Portal Engine	ASPX Template
How you work	<p>You build websites 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>
How you assemble pages	<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 website project that you can open in Visual Studio.</p> <p>You can also place web parts (which are actually standard ASCX user controls) on page templates if the functionality is not available as a server control.</p>
Master pages and visual inheritance	<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 master page (.master file).</p>

		The pages do not inherit content from their parents, they only inherit content from the master page (if it's used).
Custom code integration and extensibility	<p>You can create your own user controls and web parts if you need to integrate specific functionality.</p> <p>You can add any custom controls and code to the web parts or user controls that you use on your website.</p> <p>You can also use standard ASPX pages within your portal engine-based website.</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.
Advantages	<ul style="list-style-type: none"> • Easier and faster way of building a website. • ASP.NET programming knowledge is not required for common tasks. • You can build the whole website very quickly, using only the web browser. 	<ul style="list-style-type: none"> • Standard ASP.NET architecture. • You can use your favorite development tools, such as Visual Studio for all changes.
Disadvantages	<ul style="list-style-type: none"> • Proprietary architecture and development. 	<ul style="list-style-type: none"> • Requires ASP.NET programming knowledge.



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 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

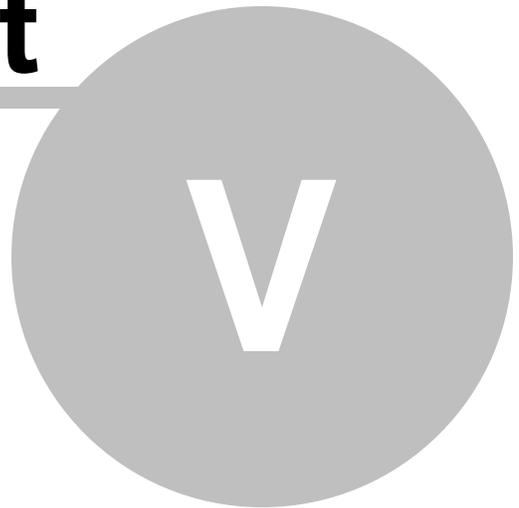
- document management features for uploaded files

Moreover, Kentico CMS comes with many **professional and flexible built-in modules out-of-the-box**, including Newsletters, On-line forms, Forums, E-commerce, Staging, Image gallery, Event calendar, Events, 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 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 the portal engine.

Part



Creating pages using the portal engine

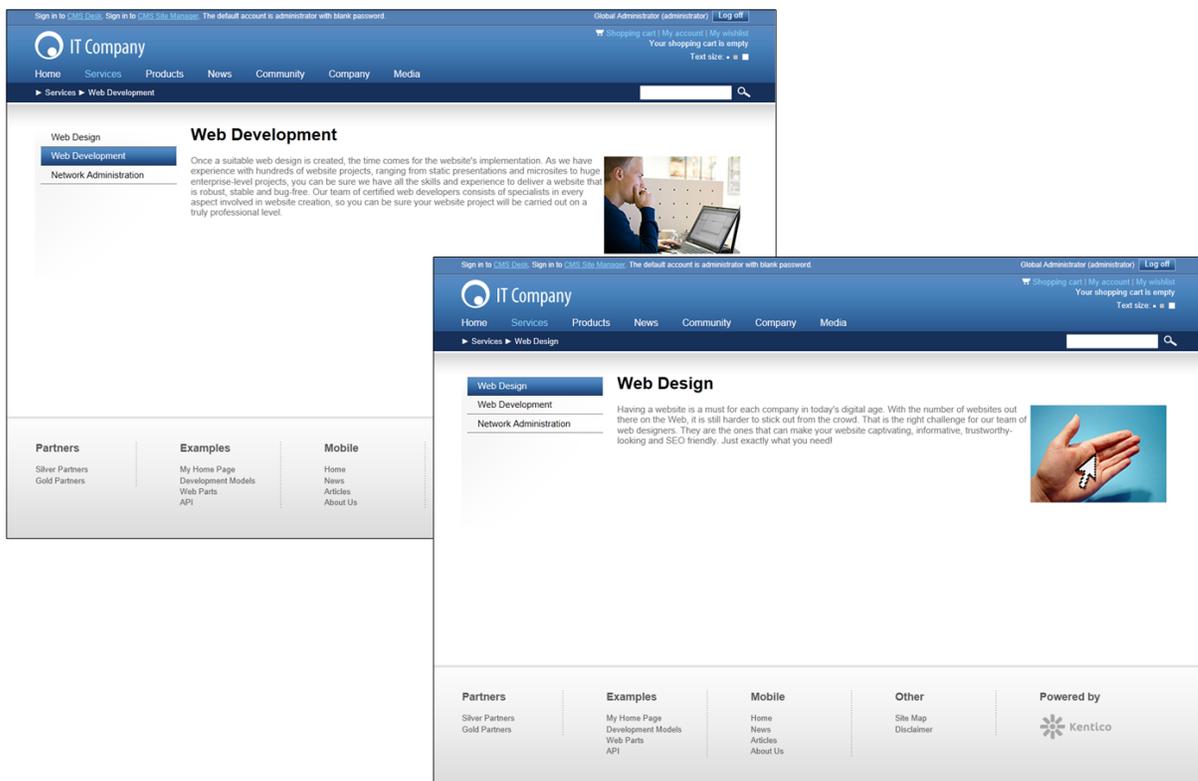
5 Creating pages using the portal engine

5.1 Page templates

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

What is a page template?

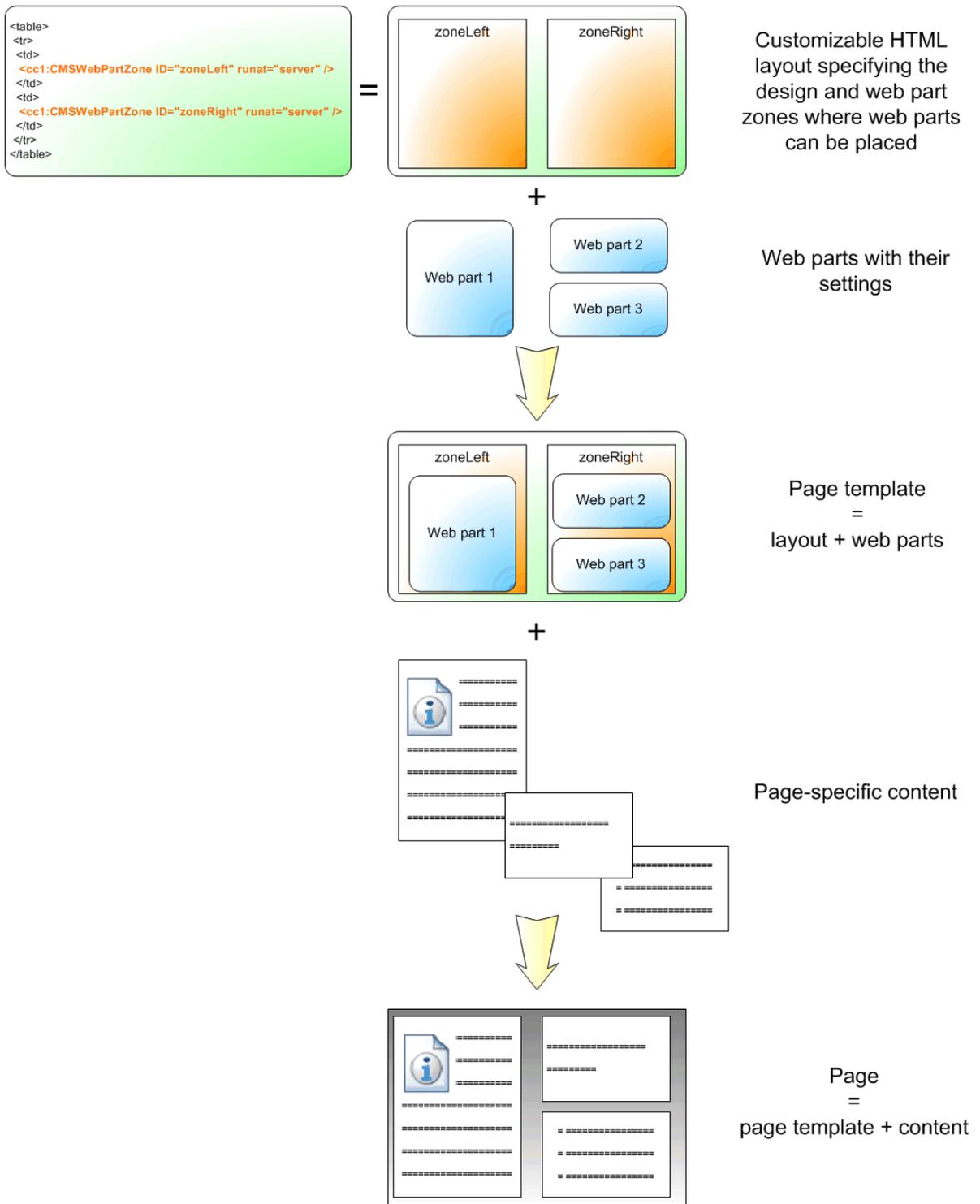
Every web page is based on some page template. The page template can be specific for a single page (so called "ad hoc" page template) or it can be re-used for several pages. The following picture shows an example of two pages that use the same page template:



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?

A page template is a combination of page layout (ASCX or 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 piece of simple, fully customizable HTML code that defines the design of the page and 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 forms. The layout and the web parts together define the **page template**.

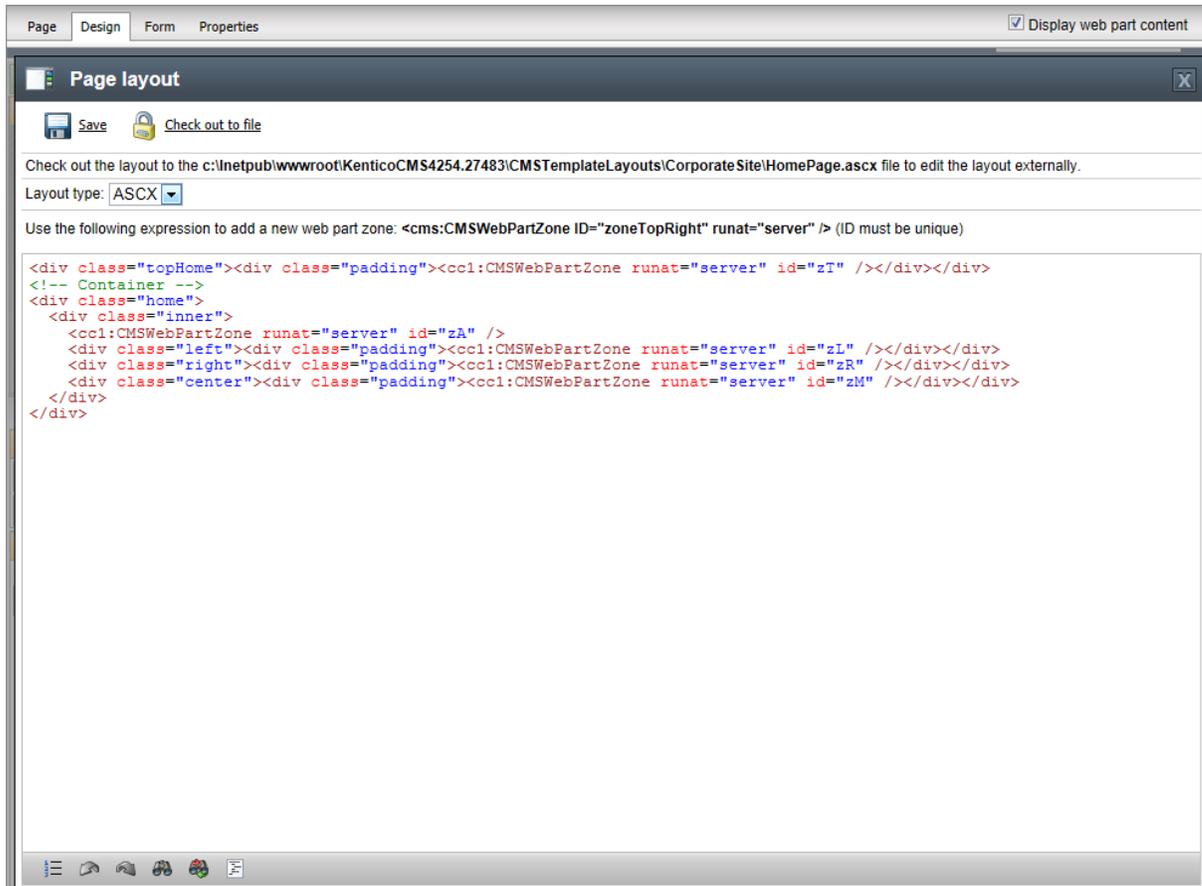
When you add **page specific content** to the page template, you get the final **page**.

5.2 Modifying a page template

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" area designated by a green box contains the orange 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 according to 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 layout of the page template:



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 a CSS-based layout.

Editing the layout using an external editor

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



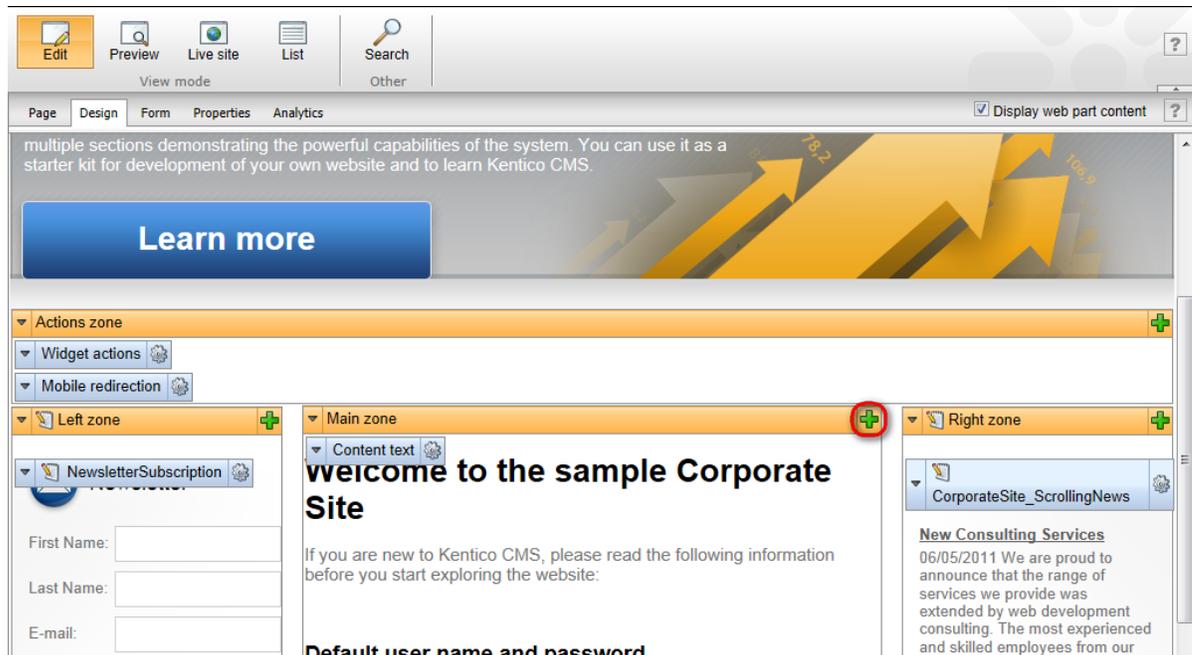
Access control

Documents can be opened on the **Design** tab only by users who are marked as **global administrators** or have the the **Design website** permission for the **CMS Content** module assigned to them. It cannot be accessed by content editors.

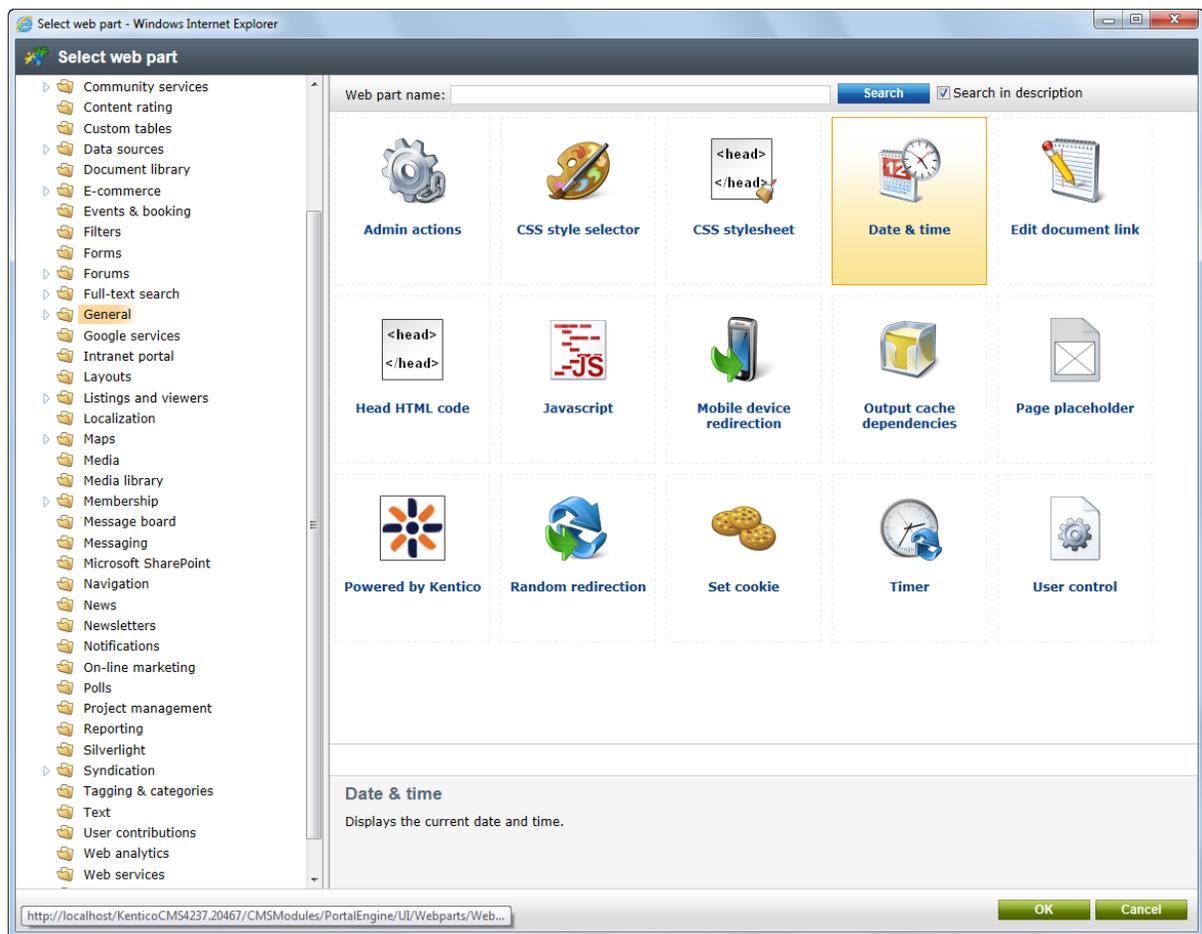
5.3 Modifying the home page structure

Now you will learn how you can modify the home page structure. We will add a web part displaying the current time to the bottom of the page.

Click **Home** in the content tree and click the **Design** tab. We will add another web part to the **zoneCenter** web part zone. Click the **Add web part (+)** icon in the top right corner of the zone.



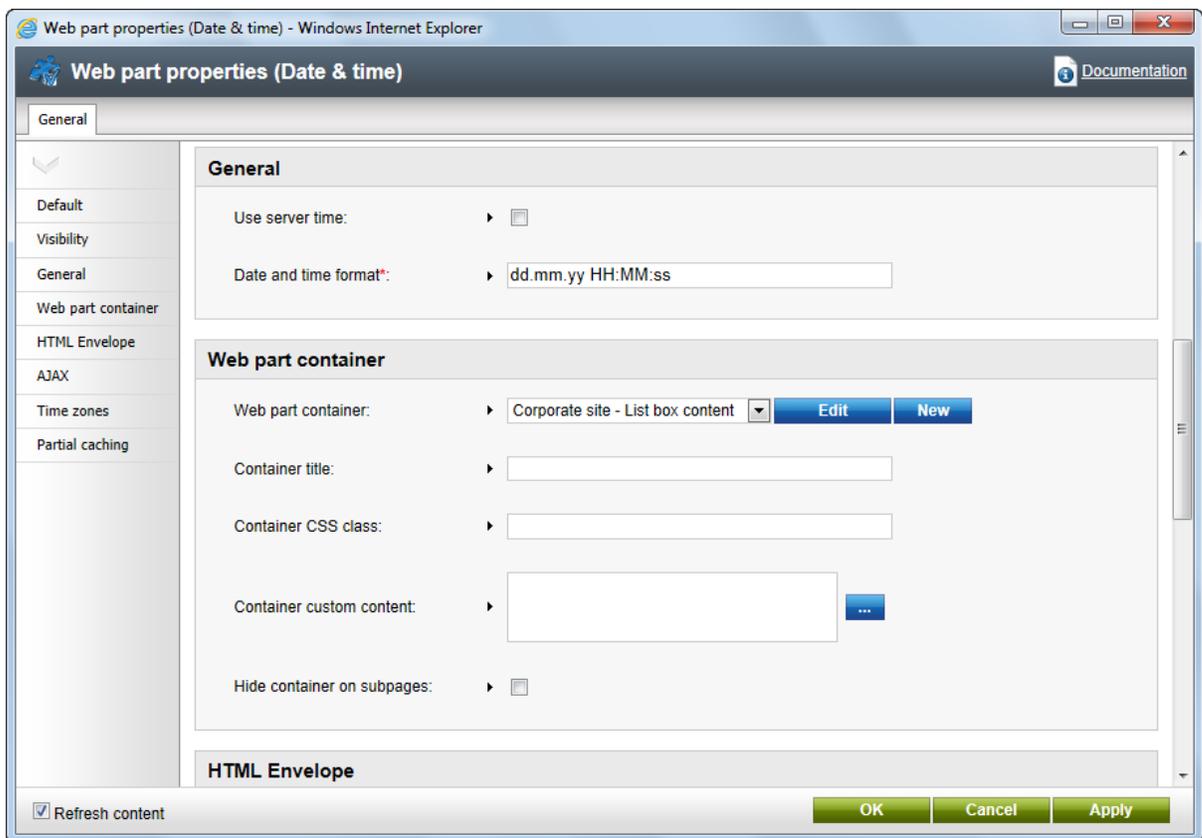
The **Select web part** dialog will be opened. Choose the **General -> Date & time** web part and click **OK**.



Next, the **Web part properties** dialog will be opened. Enter the following values:

- **Web part container:** Corporate Site - List box content

The dialog now looks like this:

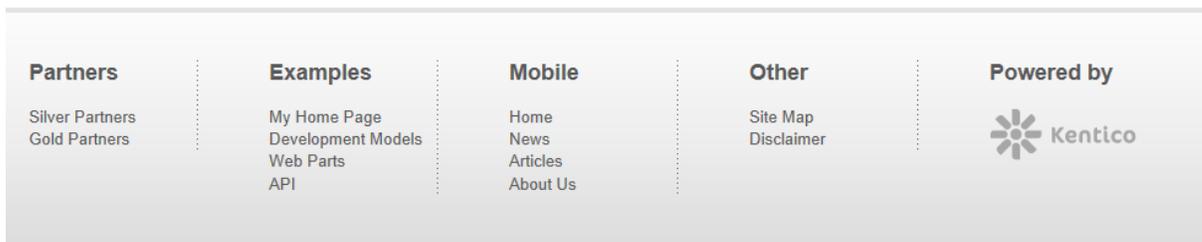


Click **OK** to save the changes and add the web part to the zone. Click **Live site** in the main toolbar to view the new design of the home page:

API Examples

Clicking this button will take you to the Support -> API examples section of Site Manager where you can view and execute examples of code utilizing Kentico CMS API.

08.08.11 19:12:55



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 properties of a web part by opening/refreshing the website in a new browser window.

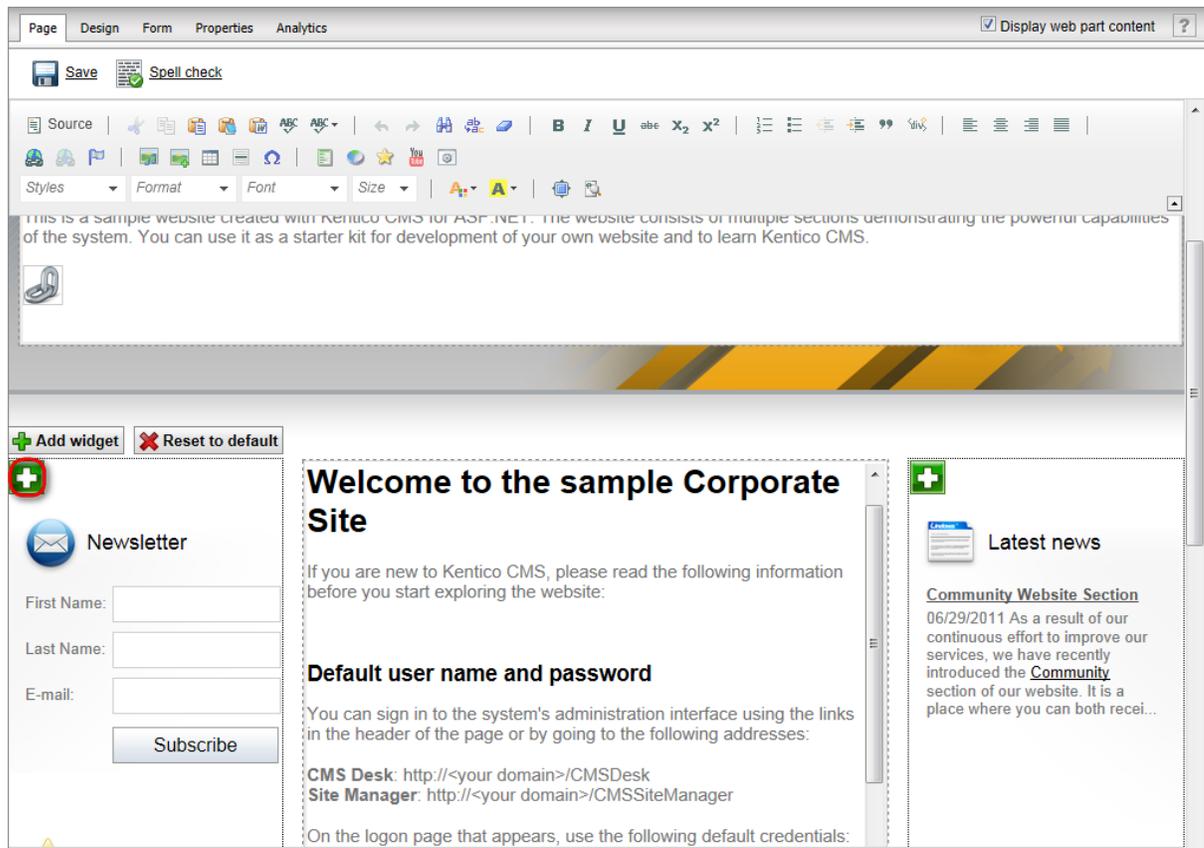
It's also useful to use tabbed browsing support for easy switching between the **CMS Desk** and **Site Manager** interface.

5.4 Customizing pages using widgets

Now return to **Edit** mode and view the **Home** page on the **Design** tab. You may have noticed that the **Left zone** and **Right zone** zones are marked with an additional icon (📄).

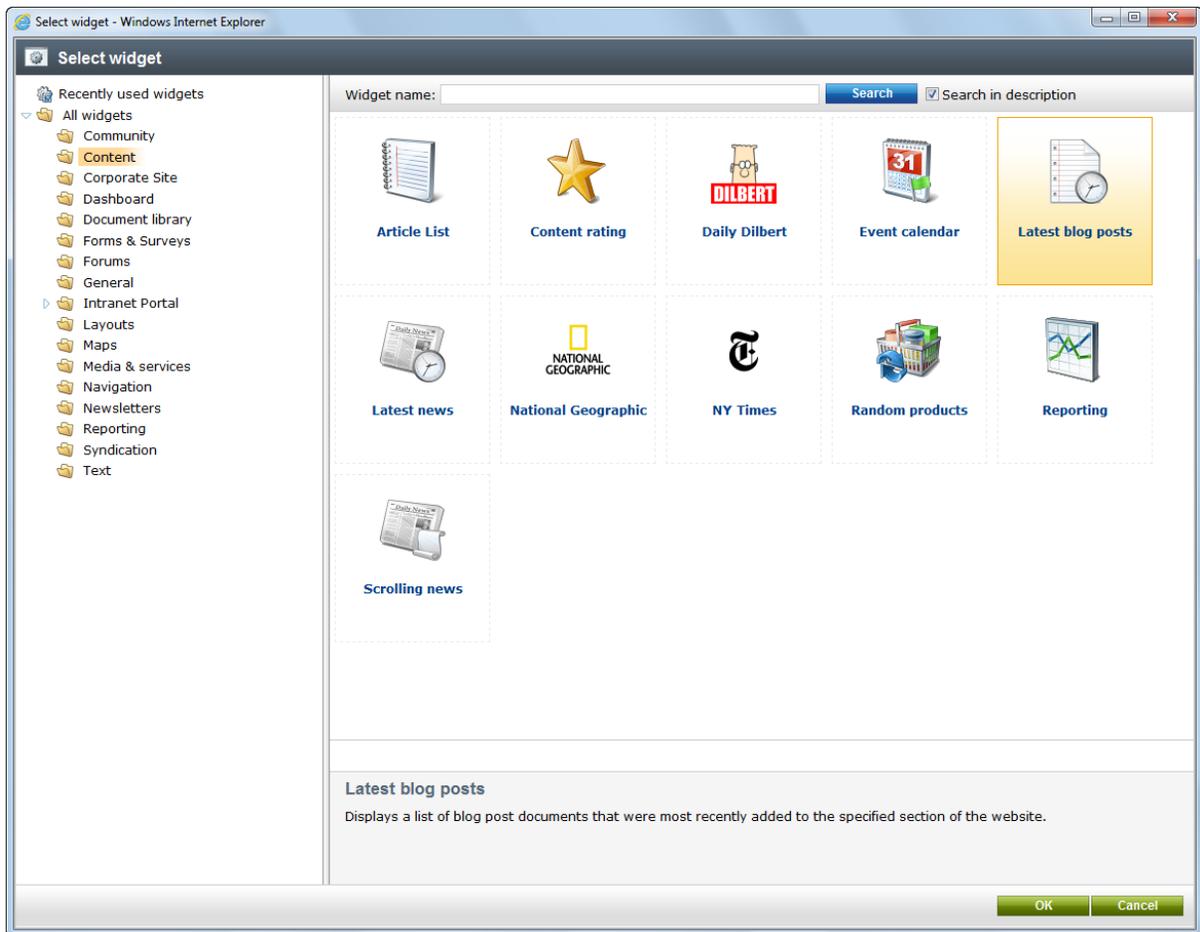
This identifies them as zones that contain *widgets* of a certain type rather than web parts. Widgets are components placed into page template zones just like web parts, but they provide the option of page customization for various kinds of website users, not just administrators or designers. These two zones are configured to allow customization by page editors, which means their content can be managed on the **Page** tab of **CMS Desk**.

Switch over to the **Page** tab and click the **Add widget** (📄) button in the top left corner of the outlined widget zone to place another widget onto the page.



The screenshot displays the Kentico CMS 6.0 editor interface. At the top, there are tabs for 'Page', 'Design', 'Form', 'Properties', and 'Analytics', along with a 'Display web part content' checkbox. Below the tabs is a toolbar with 'Save' and 'Spell check' buttons, followed by a rich text editor with various formatting options like bold, italic, underline, and text color. The main content area shows a sample website layout with a yellow and black diagonal banner. Below the banner, there are three widget areas: a 'Newsletter' widget on the left with input fields for 'First Name', 'Last Name', and 'E-mail', and a 'Subscribe' button; a central 'Welcome to the sample Corporate Site' widget with introductory text and default user credentials; and a 'Latest news' widget on the right with a 'Community Website Section' article snippet.

The **Select widget** dialog will be opened. You may notice that the content is similar as in the web part selection dialog, but with less available items. This is because every widget is based on an existing web part. Select the **Content -> Latest blog posts** widget and click **OK**.



Enter the following values into the widget's properties:

- **Web part container:** Black box
- **Container title:** Latest blog posts

Leave the remaining properties in their default state and click **OK** to add the widget to the page. Switch to **Live site** mode using the main toolbar to view the modified design of the home page. As you can see, pages with widget zones may be easily altered by content editors.



User name: administrator
Password: leave the password blank

Where to learn more?

If you are new to Kentico CMS or if you are looking for further information about the system, you can refer to the following information sources:

Kentico DevNet

On-line portal for Kentico CMS developers, providing news, blogs, discussion forums, knowledge base, FAQs, videos and many more.

Kentico CMS Documentation

Full documentation of the system available on-line at Kentico DevNet. You can download a large variety of guides, references and brochures aimed at different target audiences and various aspects of using the system.

Examples

The Examples section of this website contains live examples of all Kentico CMS web parts and a sample home page customizable by individual users using widgets.

API Examples

Clicking this button will take you to the Support - > API examples section of Site Manager where you can view and execute examples of code utilizing Kentico CMS API.

Latest blog posts

Remote Management

In this blog post, I will share some remarks regarding communication between our former New York Office and the newly setup London Office. Posted on 3/23/2011 3:12:26 PM

Expanding to Europe

In this blog post, I will try to share some of my impressions of the recent expansion of our operations to the Old Continent. Posted on 3/21/2011 5:57:47 PM



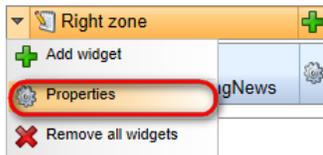
Polls

How do you like our new website?

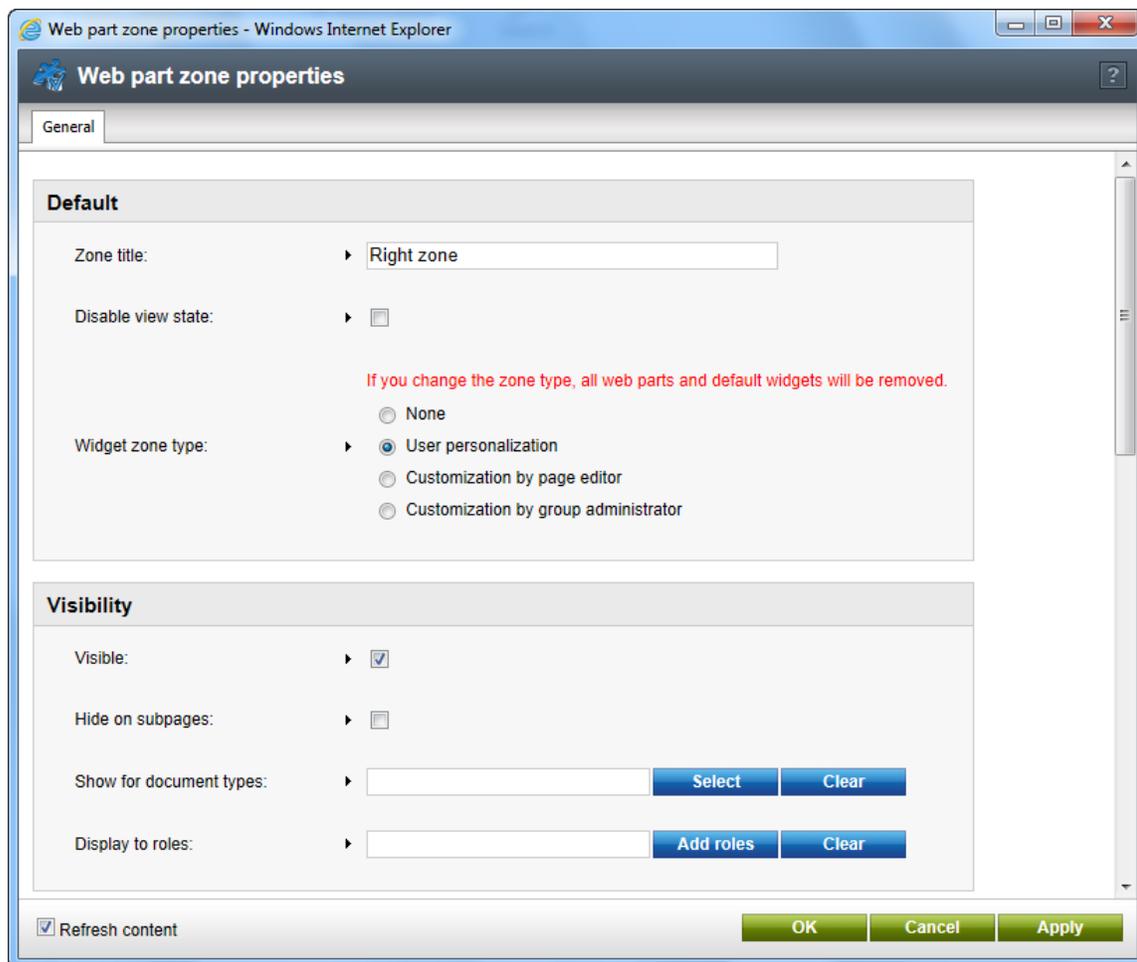


Vote

Now we will try out a different type of widget zone. Switch back to **Edit** mode and the **Design** tab. Expand the menu (▼) of the **Right zone** and select **Properties** to configure the zone.



Set the **Widget zone type** property to *User personalization*.



Click **OK**. The zone will now be editable by registered users directly on the live site. Please be aware that the content of a zone is removed when its zone type is changed.

Switch to the live site, hover over the location of the right zone and click the **Add widget** (🛠️) button.

IT Company

Shopping cart | My account | My wishlist
Your shopping cart is empty
Text size: ■ ■ ■

Home Services Products News Community Company Media

Home

Discover Unlimited Website Possibilities!

This is a sample website created with Kentico CMS for ASP.NET. The website consists of multiple sections demonstrating the powerful capabilities of the system. You can use it as a starter kit for development of your own website and to learn Kentico CMS.

[Learn more](#)

Newsletter

First Name:

Last Name:

E-mail:

Welcome to the sample Corporate Site

If you are new to Kentico CMS, please read the following information before you start exploring the website:

Default user name and password

You can sign in to the system's administration interface using the links in

Select the **Content -> Latest news** widget and click **OK**. In the properties dialog, leave the default values and click **OK** again. The widget will be displayed on the page.

Newsletter

First Name:

Last Name:

E-mail:

Welcome to the sample Corporate Site

If you are new to Kentico CMS, please read the following information before you start exploring the website:

Default user name and password

You can sign in to the system's administration interface using the links in the header of the page or by going to the following addresses:

Latest news

[Community Website Section](#)
6/29/2011
As a result of our continuous effort to improve our services, we have recently introduced the [Community](#) section of our website. It is a place where you can both receive interesting information about the company from various communication channels and express your opinions and thoughts yourselves.
[Company Growth Exceeds](#)

However, changes made to user widget zones do not affect the global design of the page. These types of zones are used to create personalized versions of the page only for the specific user. Sign out and log in as a different user (enter the username **Andy** with a blank password). View the home page on the live site and you will see that the zone is displaying its default content (empty in this case) and the widget previously added under the administrator account is not visible. Every user can choose their own content for the zone from the available selection of widgets.

Sign out and log back in as the global administrator.

Managing widgets on the Design tab



Widgets of all types may also be added or configured directly on the **Design** tab in the same way as web parts. There, the **default** content of widget zones can be set for the template, which may often be different than the currently displayed content.

5.5 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 with layout** 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 changing other pages.

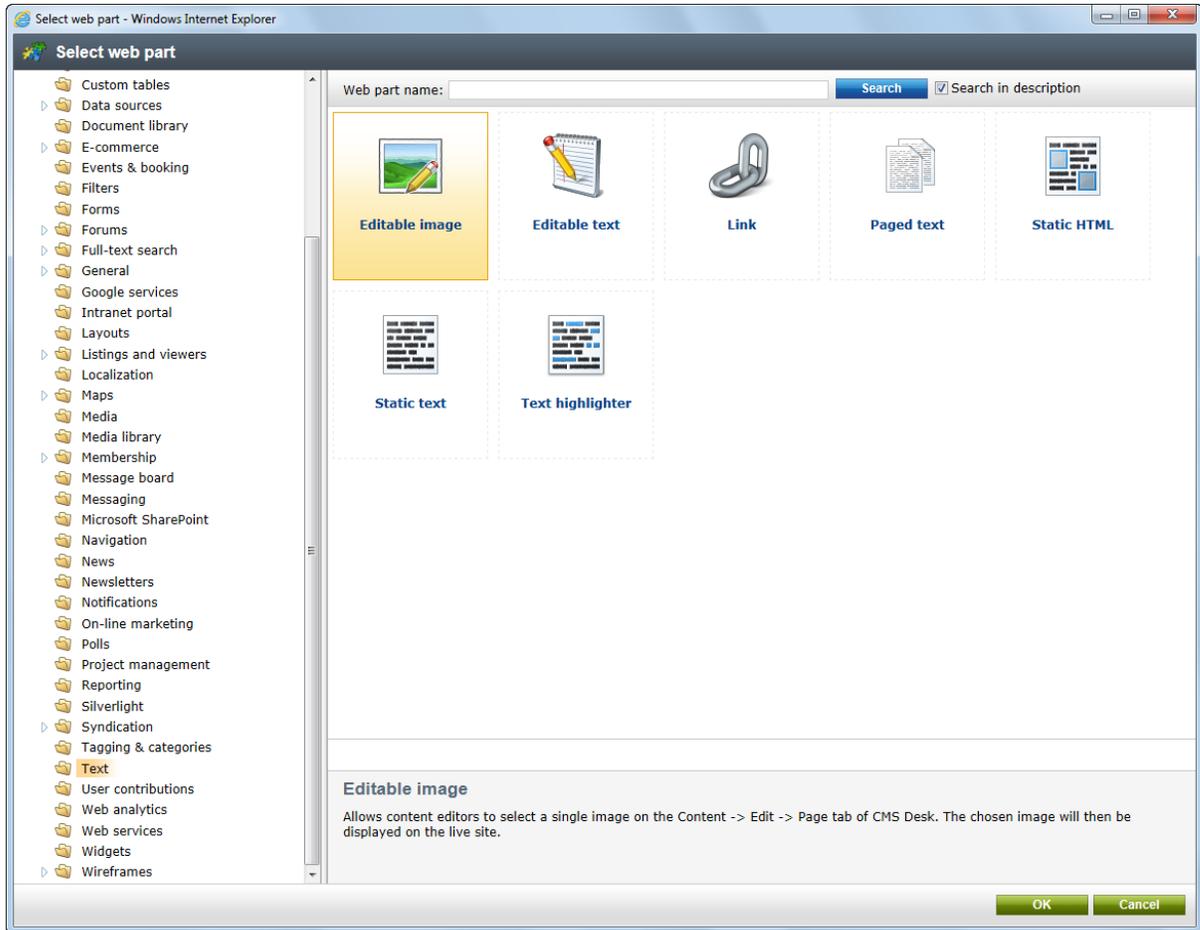
The screenshot shows the page creation interface. At the top, there are buttons for 'Save', 'Save and create another', and 'Spell check'. Below these is a 'Page name' field containing 'About us'. There are four radio button options: 'Use existing page template', 'Use parent page template', 'Create a blank page with layout' (which is selected and circled in red), and 'Create a blank page'. Below the radio buttons is a 'Layout name' field and a 'Search' button. A grid of layout icons is displayed, with the 'Two columns' layout highlighted in yellow. Below the grid, there are pagination controls '1 2 3' and a preview section for the 'Two columns' layout, which includes the text 'Two columns with web part zones.' At the bottom, there is a checkbox labeled 'Copy this layout to my page template' which is checked and circled in red.

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 on the right of 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
- **Content before:** <h1>
- **Content after:** </h1>

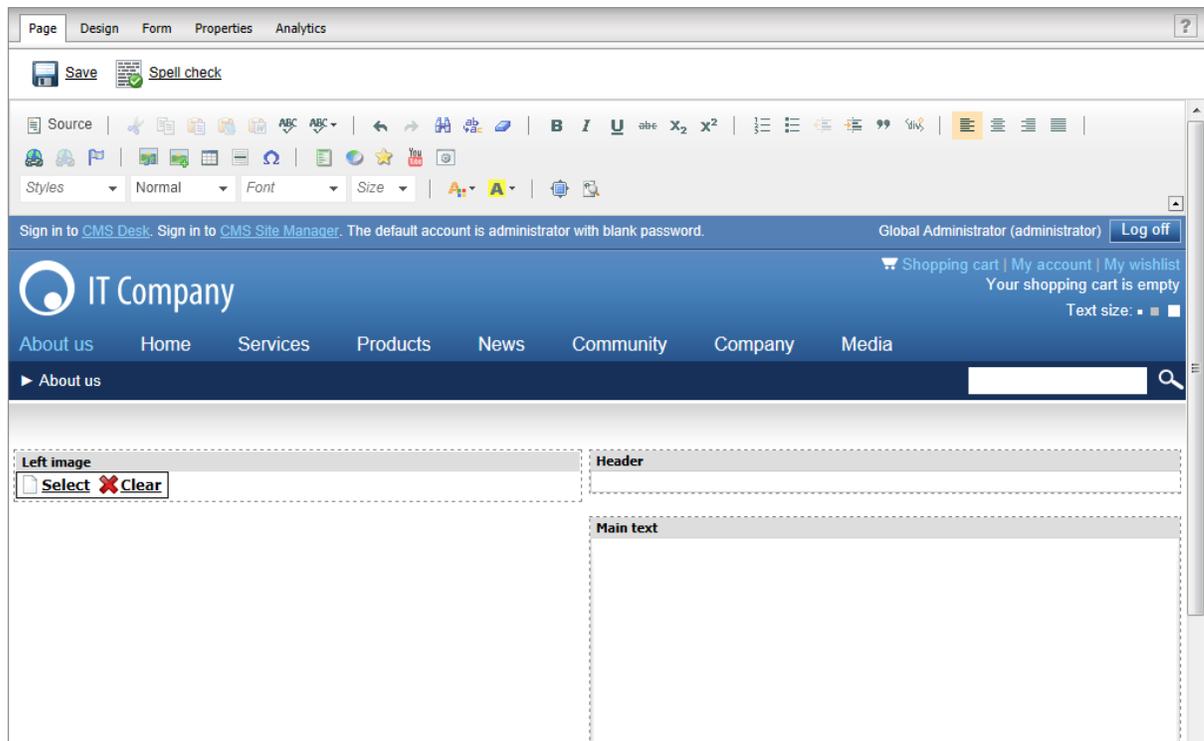
Add another web part to the **zoneRight** zone: Choose the **Text -> Editable text** web part and click **OK**. Set the following property values:

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

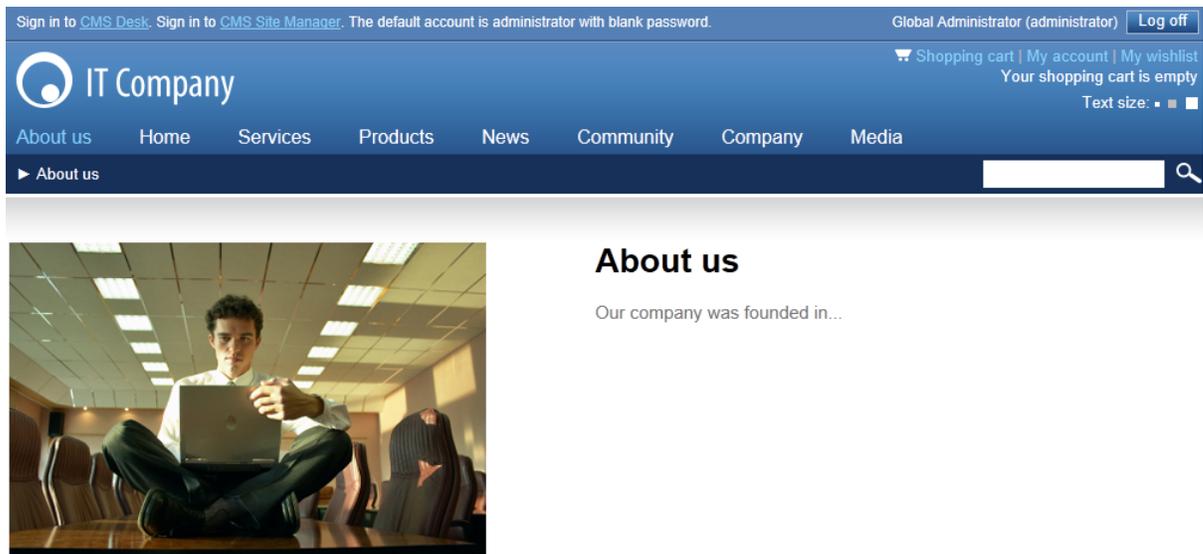
Click **OK**. Add another **Editable text** web part to the right web part zone and set the following properties:

- **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:



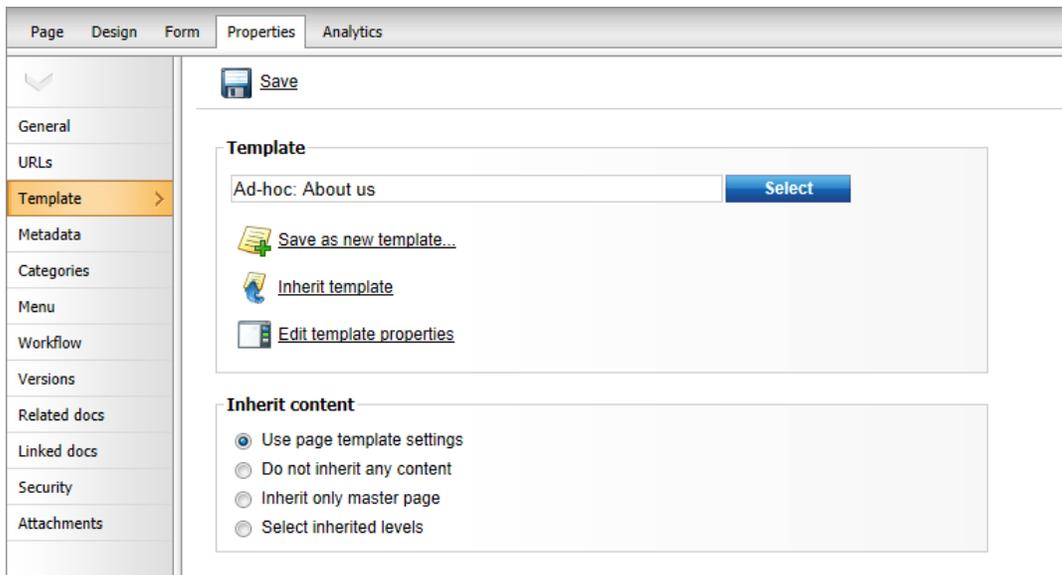
Congratulations, you have just created your first page template using web parts.

5.6 Re-using the page template

In the previous topic, 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**.

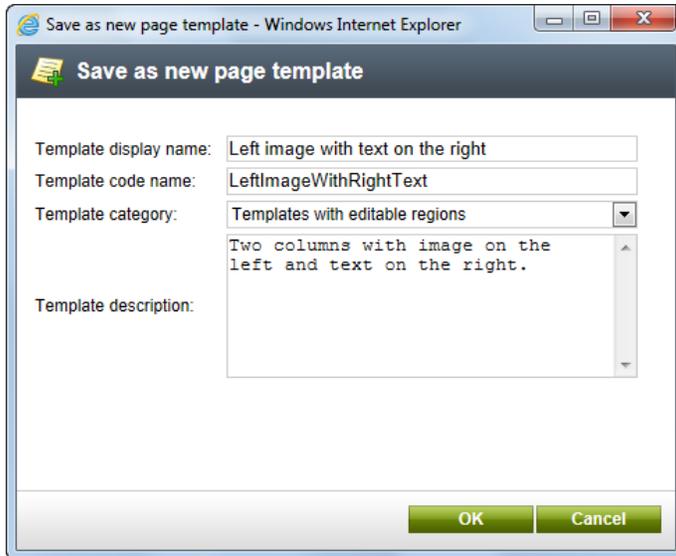
Now, you will learn how to convert the page-specific page template into a **re-usable page template** that can be used for several pages with the same layout, but different content.

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:

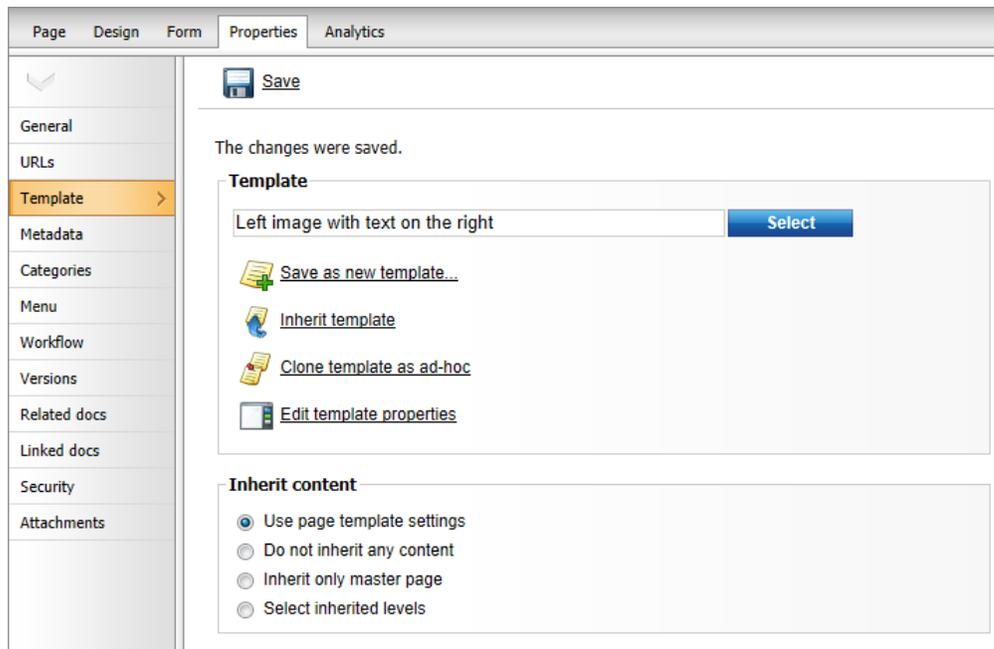


Click **Save as new template** and enter the following values:

- **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** above 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!

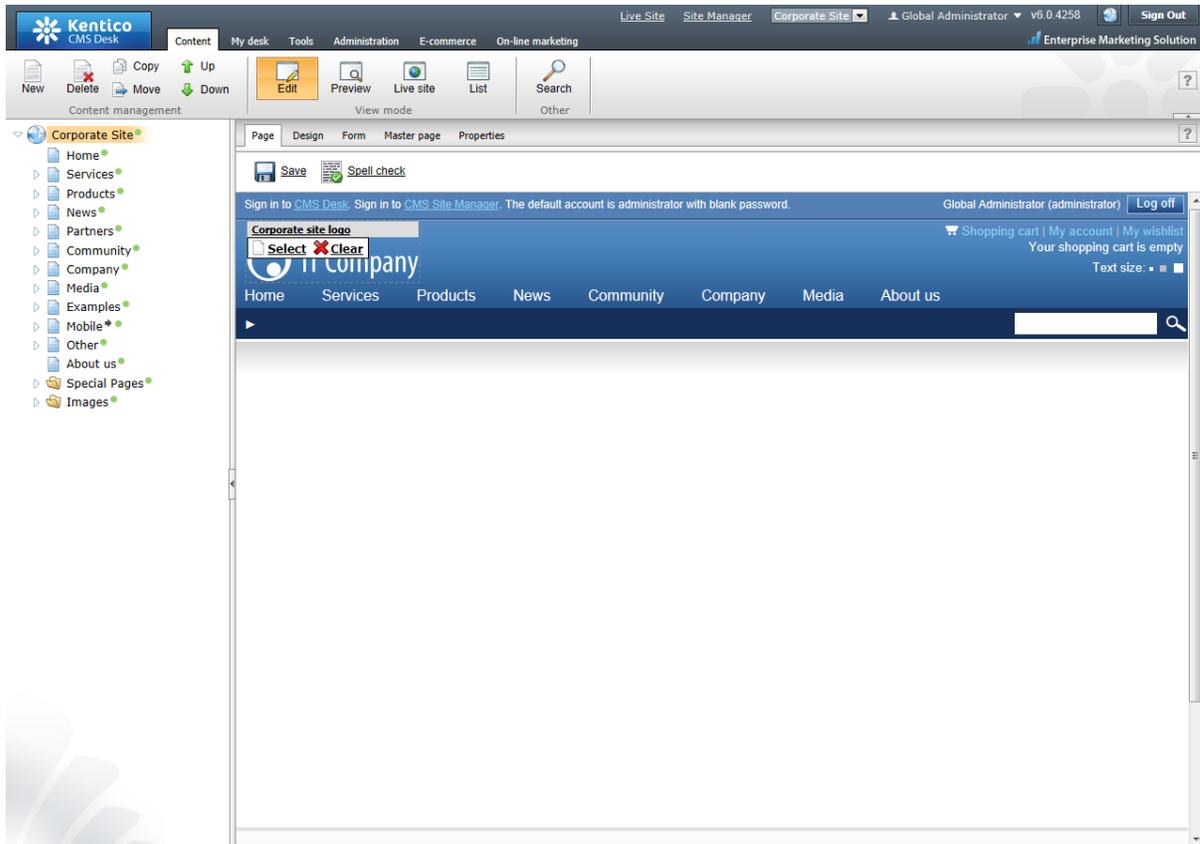


Important!

Please keep in mind that when you create a page based on an ad-hoc page template and later delete the page, the corresponding ad-hoc page template is deleted as well and **cannot be restored**.

5.7 Visual inheritance

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



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

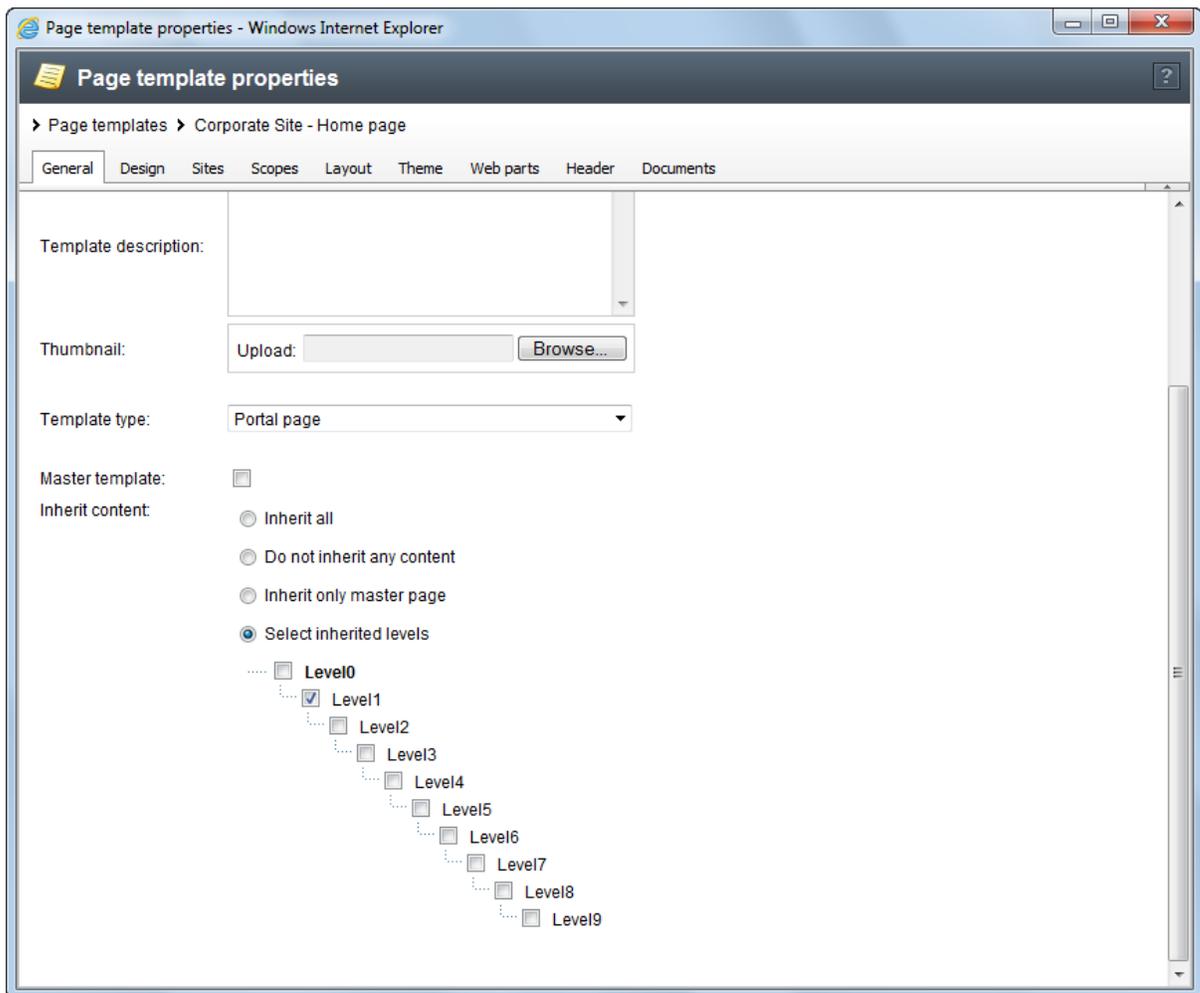
Please note how the pages are structured in the content tree: the root (master page) is the parent of the Home page which is loaded inside the master page. You can perform nesting recursively at any level of the content tree. A necessary component for page nesting is the **Page placeholder** web part.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

5.8 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 **Select inherited levels** and check only the **Level 1** box. It means that only the content from the first level of the content hierarchy will be displayed and the root master page (Level 0) is not inherited. Click **Save** to save the changes.



The page will look like this:

The screenshot displays the Kentico CMS Desk interface. The top navigation bar includes 'Content', 'My desk', 'Tools', 'Administration', 'E-commerce', and 'On-line marketing'. Below this is a toolbar with icons for 'New', 'Delete', 'Move', 'Copy', 'Up', 'Down', 'Edit', 'Preview', 'Live site', 'List', and 'Search'. The left sidebar shows a tree view of the site structure, with 'News' selected under 'Corporate site'. The main content area shows the 'News' section, which includes a text block explaining that news are standard Kentico CMS documents and can be scheduled or approved. Below this is a 'News List' section with a search box and a list of news items. The first item is titled 'Community Website Section' and includes a small image of people and a brief description.

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

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

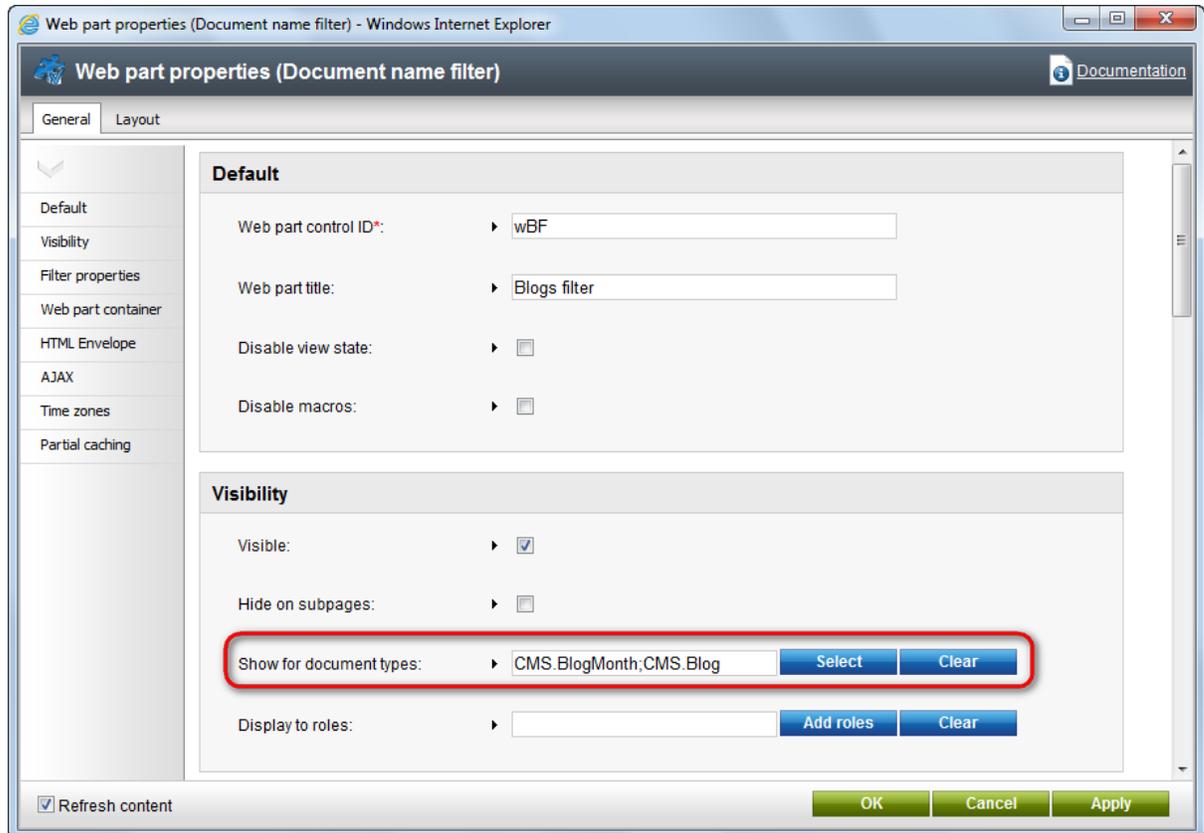
The screenshot shows the 'Properties' dialog box in Kentico CMS, specifically the 'Template' tab. The left sidebar lists various property categories, with 'Template' selected. The main area shows a 'Save' button at the top. Below it, the 'Template' section has a dropdown menu set to 'Corporate Site - News' and a 'Select' button. Underneath are four options: 'Save as new template...', 'Inherit template', 'Clone template as ad-hoc', and 'Edit template properties'. The 'Inherit content' section has four radio button options: 'Use page template settings', 'Do not inherit any content', 'Inherit only master page', and 'Select inherited levels'. The 'Select inherited levels' option is selected, and a tree view below it shows 'Root' checked.

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

Every web part has a property called **Hide on sub-pages**. When you check the box, the web part is hidden on pages where it is inherited from a parent page.

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. To see how it works, select **/Community/Blogs/Brad-Summers-Blog** in the content tree. On the **Design** tab, click **Configure** (⚙️) for the **Header text**, **Description text** or **Blogs filter** web part. All of them have the **Show for document types** property set to *CMS.BlogMonth;CMS.Blog*. This means that the web part will only be displayed on blog month and blog documents, not on blog posts which are stored under blog months. Click **Cancel**.



Still viewing the **Brad Summers Blog** document, switch to the **Live site** mode. The header text, description text and filter are all displayed above the repeater displaying blog posts.

► Community ► Blogs ► Brad Summers Blog

Forums Blogs 27 Events Wiki

Brad Summers Blog

Hi, my name is Brad Summers and I am the head of web development in our company. I decided to start this blog in order to share the most interesting remarks and ideas that I come across during my day-to-day work. I will share all sorts of interesting information related to activities of our company and to web development in general. I believe that it will be interesting reading for all our customers, partners and all other individuals interested in web development. And of course, you can post comments on each blog post in case that you want to share your opinion, have something to add or if you want to raise a discussion related to a post's topic.

Blog post name: Search



[Remote Management](#)

In this blog post, I will share some remarks regarding communication between our former New York Office and the newly setup London Office.

Brad Summers | 3/23/2011 3:12:26 PM | [2 comments](#)

When you display some particular blog post, the web parts are not displayed because *CMS.BlogPost* is not among the enumerated document types.



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

5.9 Web part containers

Web parts or widgets can be configured to display only their own content or they can be rendered within a "container". The container consists of HTML code that is placed before and after the web part.

Switch to **CMS Desk**, select **Home** and click **Live site**. The **Newsletter subscription** widget uses the Black box container:

With container:	Without container:
<p>Blog post Details</p> <p>Remote Management In this blog post, I will share some remarks regarding communication between our former New York Office and the newly setup London Office.</p> <p>Expanding to Europe In this blog post, I will try to share some of my impressions of the recent expansion of our operations to the Old Continent.</p> <p>Brad Summers 03/23/2011 2 comment(s)</p> <p>Brad Summers 03/21/2011 2 comment(s)</p>	<p>Remote Management In this blog post, I will share some remarks regarding communication between our former New York Office and the newly setup London Office.</p> <p>Expanding to Europe In this blog post, I will try to share some of my impressions of the recent expansion of our operations to the Old Continent.</p> <p>Brad Summers 03/23/2011 2 comment(s)</p> <p>Brad Summers 03/21/2011 2 comment(s)</p>

Container are useful when you need to organize the page layout into separate boxes. You can choose a container for web parts or widgets when you configure their properties.

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

The screenshot shows the Kentico Site Manager interface. The left sidebar lists various development options, with 'Web part containers' selected. The main area displays the 'Container properties' dialog for a 'Black box' container. The dialog has tabs for 'General', 'Theme', and 'Sites'. The 'General' tab is active, showing a 'Save' button and two input fields: 'Display name:' (Black box) and 'Code name:' (BlackBox). Below these fields, there are two text areas for HTML code. The top area, labeled 'HTML before web part:', contains the following code:

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

The bottom area, labeled 'HTML after web part:', contains the closing tags:

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

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

HTML before web part:

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

HTML after web part:

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

As you can see the container is composed of simple HTML code. Div or Table elements can be used together with CSS classes to implement the container. The CSS classes may either be defined in the site stylesheet or locally in the **CSS styles** field.

The `{%ContainerTitle%}` macro expression indicates where the container title should be displayed. The text of the title may be specified when configuring the properties of a web part and it is dynamically resolved when the web part is displayed.

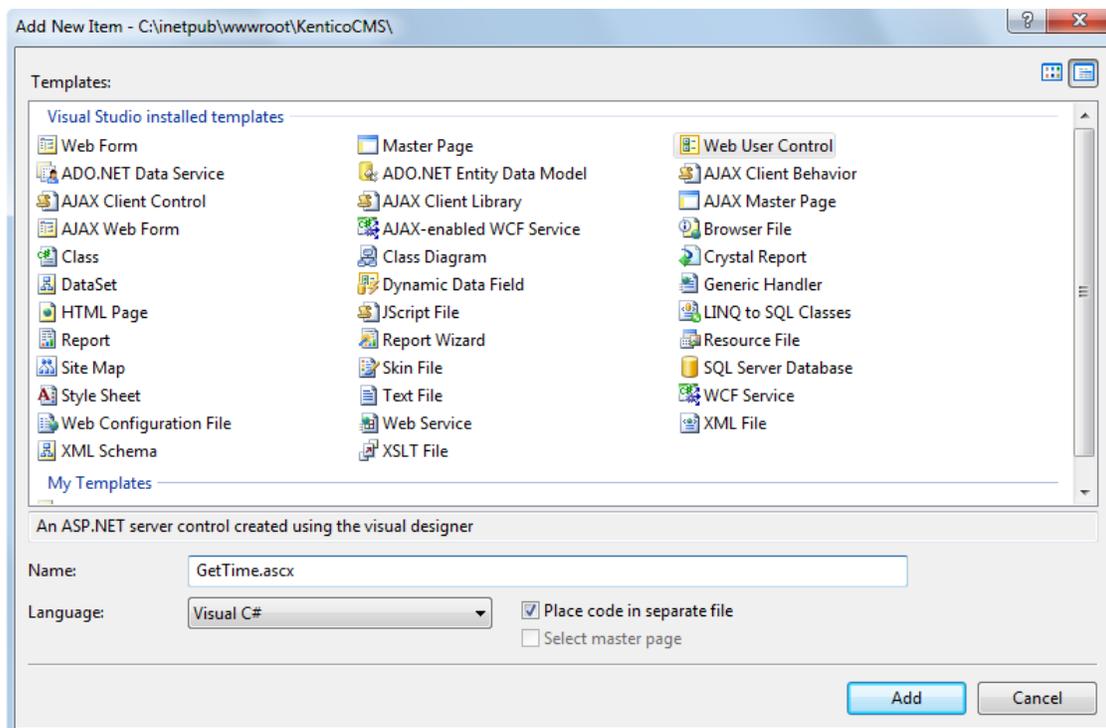
5.10 Adding custom code to your web site

The easiest way how to insert custom code into a portal engine-based website is using standard ASCX user controls. This topic will show you how to do this. If you are not familiar with Visual Studio development, you can skip this topic.

Current time example

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

Open the website project using the **WebProject.sln** file that is placed in the folder where you deployed the website. Right-click the web project root 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.



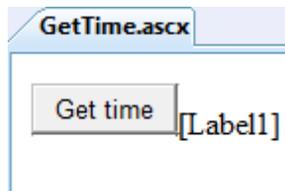
Click **Add**. Switch to the **Design** tab, drag and drop the following controls and set their properties:

Button control:

- **ID:** Button1
- **Text:** Get time

Label control:

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



Double-click the **Get time** button and add the following code to the *Button1_Click* method:

[C#]

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

[VB.NET]

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

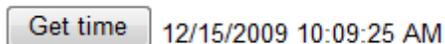
This code ensures that the label displays the current date and time when the button is clicked. It's not necessary to compile the project — user controls are compiled at run time.

Save both the ASCX and ASCX.CS files.

Adding the user control on the page

Sign in to Kentico CMS Desk, select the Home page and click **Design**. Add **+** a web part of type **General -> User control** to the **zoneCenter** web part zone. Enter the following value into 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 onto 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 onto a page is to create 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 should build web parts in case you need to create re-usable, parameterized user controls. The development of web parts is described in the next chapter.

Part

VI

Developing web parts and widgets

6 Developing web parts and widgets

6.1 Web part development overview

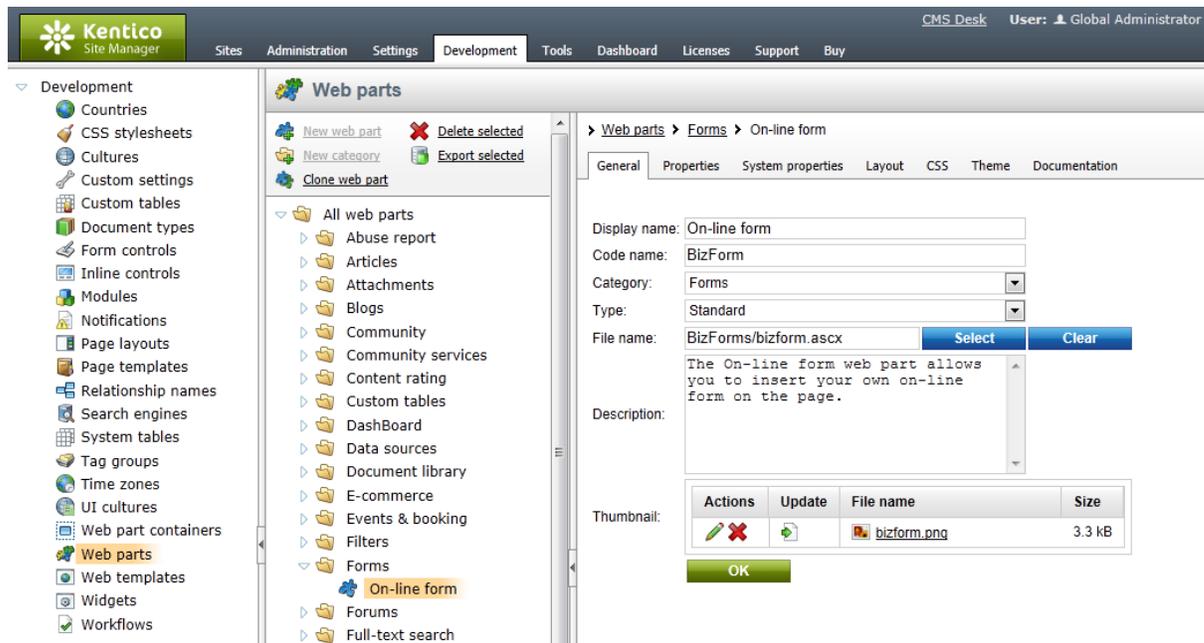
Web parts represent a block of content or combination of content and functionality. They are the basic building blocks of portal engine page templates. Using existing web parts, users with the appropriate permissions can build or modify the structure of pages directly from a browser by using the **CMS Desk - > Content -> Edit -> Design** interface. All web parts must be placed into web part zones.

A web part is a user control (ASCX) that inherits from the **CMSPortalControls.CMSAbstractWebPart** class. You can easily create your own web parts using Visual Studio or Visual Web Developer Express Edition and use them to add your custom .NET code onto the pages of the website.

For complete documentation of all web parts and their properties, please see the [Kentico CMS Web Parts](#) reference.

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 in the administration interface.
- **Code name** - the name of the web part used in website code.
- **Category** - here you can choose the category in the web part catalog where the web part will be placed.
- **Type** - the type of the web part. The following web part types are available:
 - **Standard** - typical web parts displaying some content.
 - **Data source** - do not display any content, only provide data to be displayed by a connected web part.
 - **Filter** - can be connected to a data source and enables users to limit the data provided by it.
 - **Placeholder** - used for the **General -> Layout -> Page placeholder** web part, which specifies an area where the content of sub-pages should be displayed.
 - **Invisible** - are not displayed on the page at all and usually perform some type of background task.
 - **Basic** - basic web parts without partial caching and AJAX UpdatePanel support.
 - **Widget only** - these web parts are only intended to serve as base templates for widgets and are not available in the web part catalog.
- **File name** - contains the relative path to the user control that implements the web part. The path starts from the *CMSWebParts* folder. It is recommended to organize the web parts on the disk in the same way as in the categories. Example: *Search/cmscompletesearchdialog.ascx*
- **Description** - a text description of the web part that will be displayed in the web part catalog.
- **Thumbnail** - the 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. These properties are loaded automatically when the web part is defined and their default values can be specified on the **System properties** tab. If you wish to modify more than the default value of one of these properties, such as its behaviour and attributes (e. g. for it not to be displayed in the web part configuration dialog) you can do so by adding and configuring it on the **Properties** tab.

Default

- **Web part control ID** - serves as an identifier for the web part. This ID must be unique within the context of each page template. The value of this property may only contain alphanumeric characters and the underscore character (_).
- **Web part title** - title of the web part displayed on the *Design* tab of CMS Desk. If empty, the value of the *Web part control ID* property is used for this purpose.
- **Disable view state** - indicates if view state should be disabled for the web part.
- **Disable macros** - if checked, macros contained in the values of the web part's properties will no longer be resolved.

Visibility

- **Visible** - indicates if the web part should be displayed.
- **Hide on subpages** - indicates if the web part should be hidden on sub-pages. If checked, the web part will not be displayed on documents that inherit the web part from a parent document.
- **Show for document types** - contains a list of document types on which the web part should be displayed. If the currently selected document uses the page template containing the web part, but its type is not specified by this property, the web part will be hidden. The document types in the list must be specified by their code names and separated by semicolons (;). If empty, the web part will be displayed on all document types.
- **Display to roles** - contains a list of roles to which the web part should be displayed. This may be used to implement documents with specific functionality for different types of users. The roles in the

list must be specified by their code names and separated by semicolons (;). If empty, the web part will be displayed to all users.

Web part container

- **Web part container** - specifies the name of the container (box) to be displayed around the web part. Only the containers defined at *Site Manager* -> *Development* -> *Web part containers* can be selected. The selected container can be edited directly by using the **Edit** button.
- **Container title** - sets a title for the container (if one is specified for the web part). This title is displayed only if the `{%ContainerTitle%}` macro is used in the code of the container.
- **Container CSS class** - CSS class used for the web part's container. Applied only if the `{%ContainerCSSClass%}` macro is used as the value of the *Class* attribute in the code of the container.
- **Container custom content** - custom content to be displayed in the web part's container. Applied only if the `{%ContainerCustomContent%}` macro is used in the code of the container.

HTML Envelope

- **Content before** - HTML content placed before the web part. Can be used to display a header or add some encapsulating code, such as `<div>` or `<table>` elements to achieve the required layout.
- **Content after** - HTML content placed after the web part. Can be used to display a footer or close the tags contained in the *ContentBefore* value, such as `</div>` or `</table>` elements.

AJAX

- **Use update panel** - indicates if an AJAX UpdatePanel container should be used for the web part.

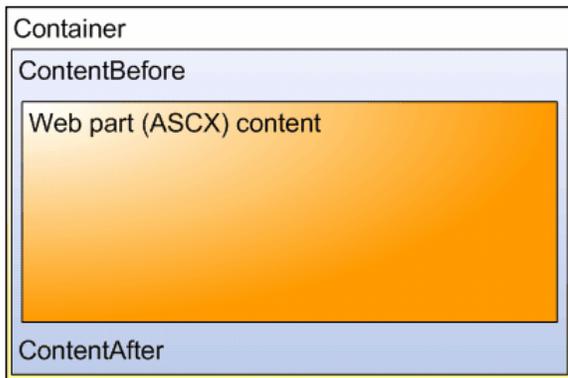
Time zones

- **Time zone** - specifies the type of time zone used for the content of the web part. The following types are available:
 - **Inherit** - inherits the time zone settings from the Page placeholder web part used to display the page template containing this web part (typically the one on the master page).
 - **Server** - server time zone settings will be used by the web part.
 - **Web site** - website time zone settings will be used by the web part.
 - **User** - time zone settings of individual users will be used by the web part.
 - **Custom** - some other time zone will be used based on the selection done in the *Custom time zone* property.
- **Custom time zone** - may be used to assign a custom custom time zone specifically for the content of this web part. If selected, the time zone will be used regardless of current user or website time zone settings.

Partial caching

- **Partial cache minutes** - sets the number of minutes for which the output HTML code of the web part should remain cached. This process is similar to full-page caching, but only for the code of the web part specifically. If left empty or set to 0, partial caching will not be used for the web part.
- **Partial cache dependencies** - contains a list of cache keys on which the partial cache of the web part depends. When the specified cache items change, the partial cache of the web part is deleted. Each line may only contain a single item. If the *Use default cache dependencies* box is checked, the default dependencies will be used, which include all possible object changes that could affect the specific web part.

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



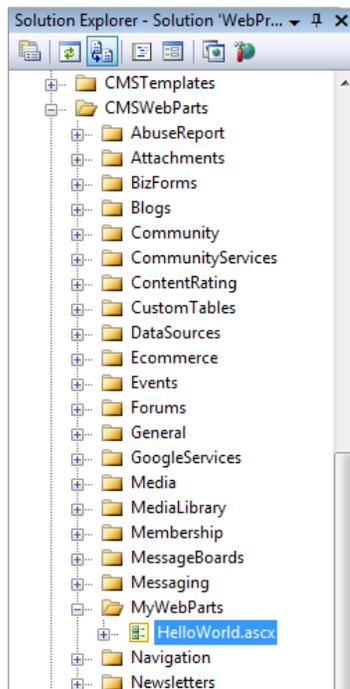
Storing files related to a web part

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

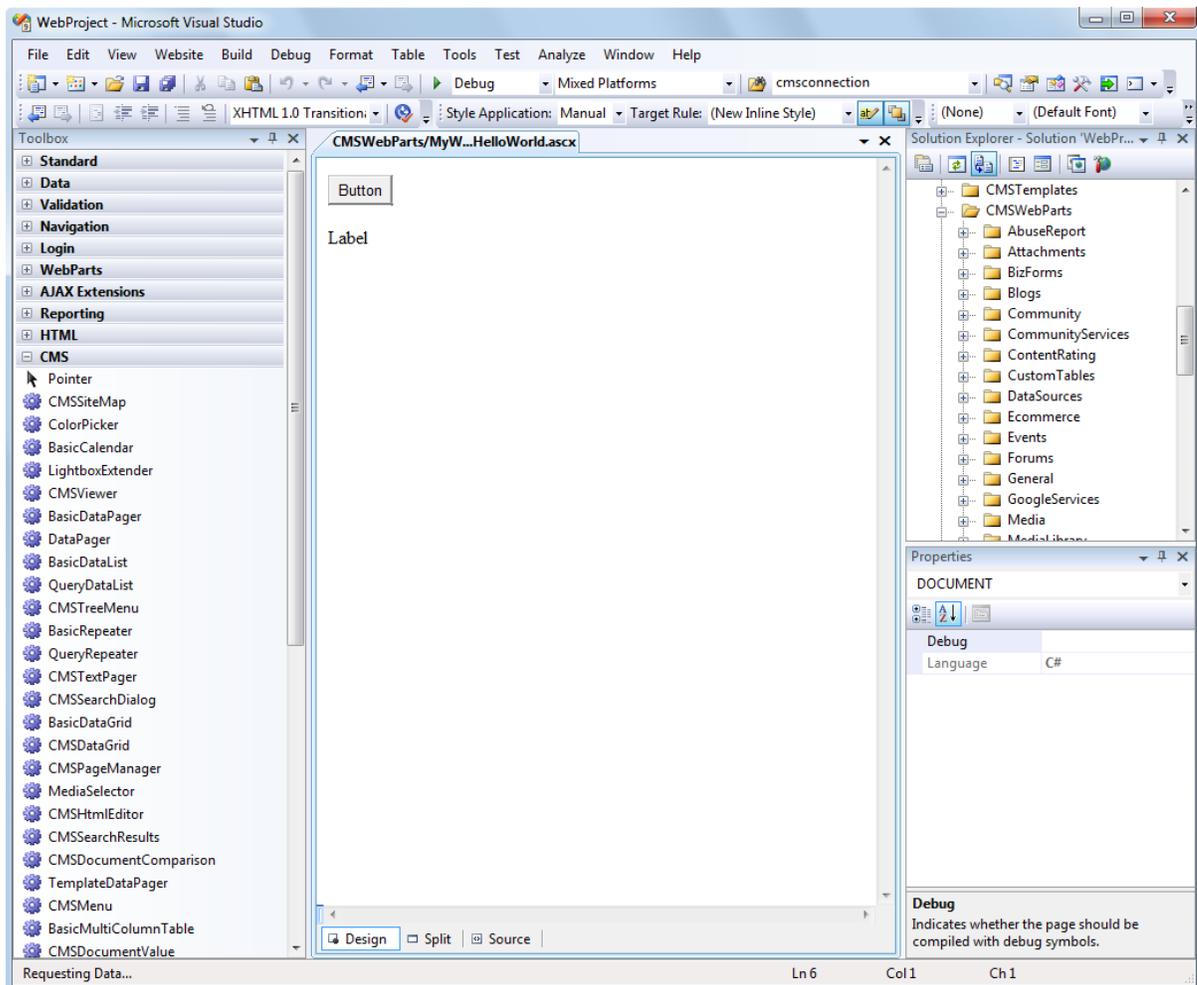
6.2 Creating a simple web part

This topic 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 (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 on the 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
```

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

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**.

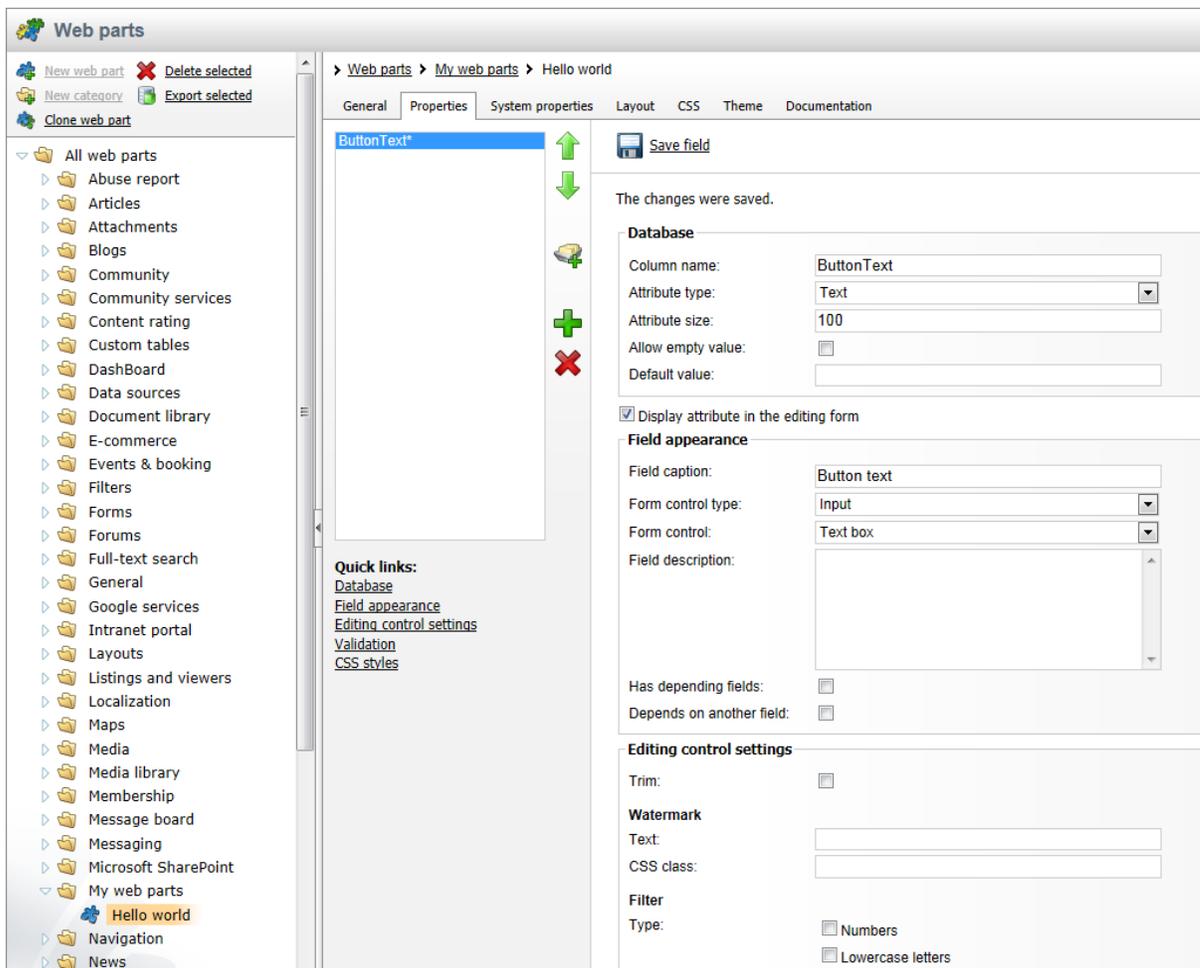
11. Select the new category and click  **New web part**. Choose to **create a new web part** and enter the following values:

- **Display name:** Hello world
- **Code name:** HelloWorld
- **File name:** ~/CMSWebParts/MyWebParts/HelloWorld.ascx

Click **OK**.

12. Switch to the **Properties** tab and add the following properties using the **Add attribute (+)** button:

- **Column name:** ButtonText
- **Attribute type:** Text
- **Attribute size:** 100
- **Field caption:** Button text
- **Form control type:** Input
- **Form control:** Text box

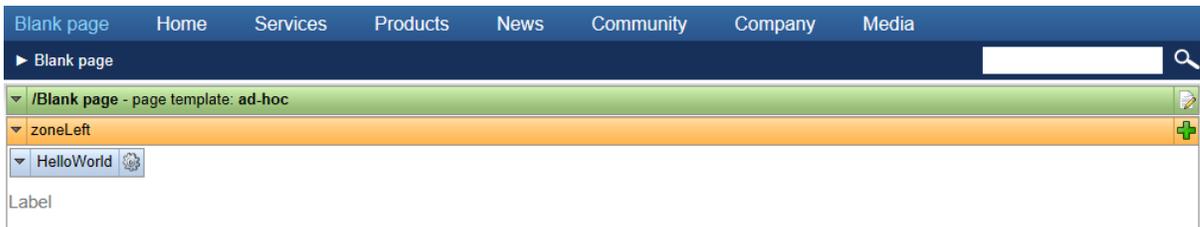


Click **OK**.

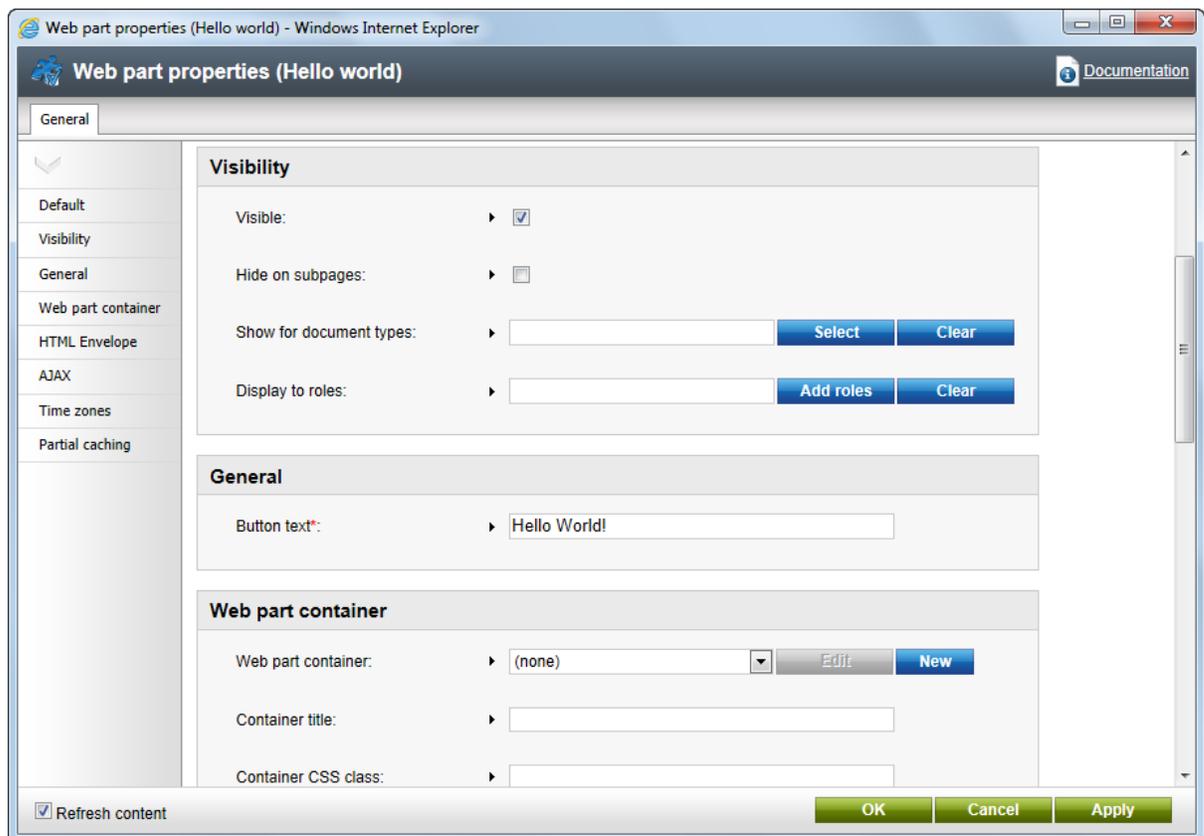
13. Switch to **CMS Desk**.

14. Create a new **blank page** using the **Simple** layout (or any other layout) under the root and switch to the **Design** tab.

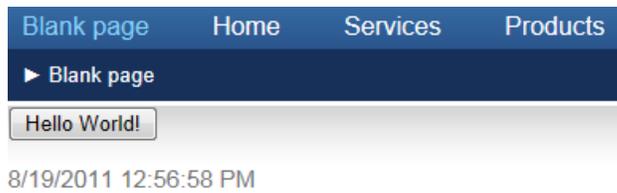
15. Click **Add web part** (+) in the upper right corner of the web part zone and choose to add the **Hello world** web part:



16. The Web part properties dialog of the **HelloWorld** web part appears, set the value of the **Button text** field to *Hello world!*



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



You have learned how to create a simple web part.



Tip: Displaying content in the web part

You can also use Kentico CMS Controls in the code of the web part (in the ASCX control) to display content from Kentico CMS in a customized form.

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's 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 appearance of a dialog or hide a field.

Go to **CMS Desk**, choose the **Design** tab, open the web part's properties dialog 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

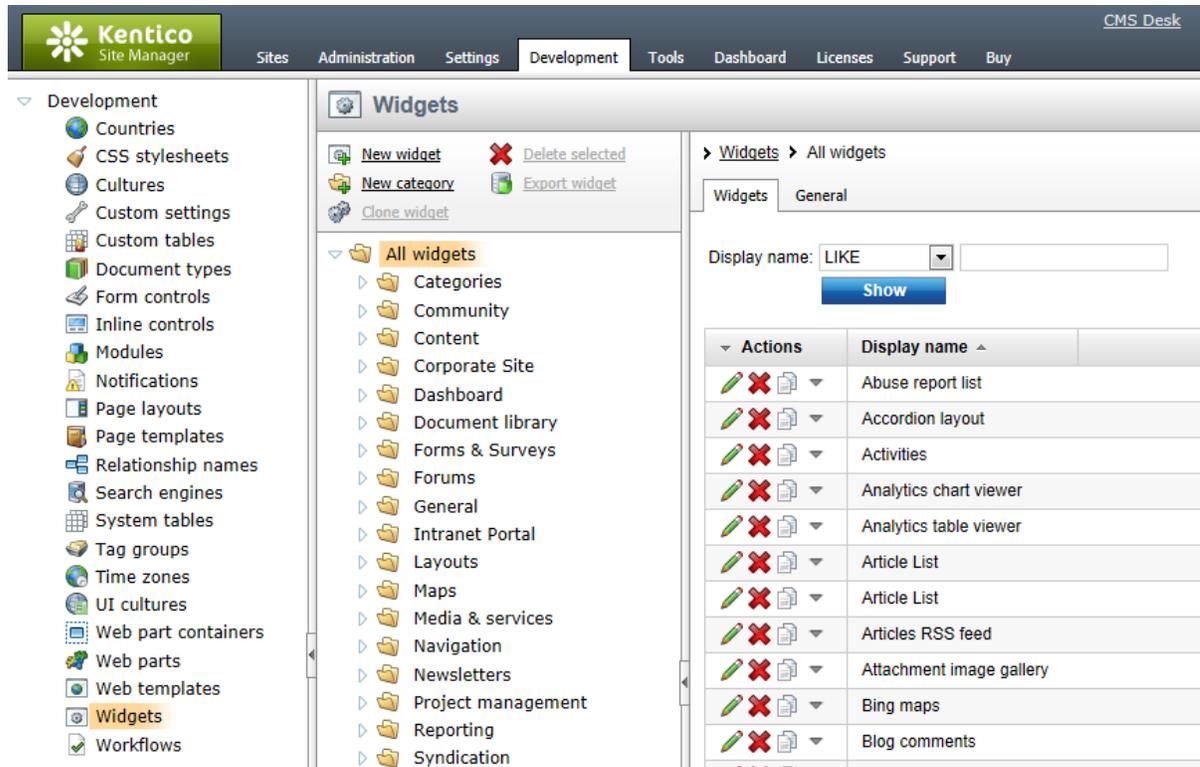
Web part inheritance allows you to create web parts with pre-set values for a particular purpose. You can, for example, create a specialized web part for displaying news, inherited from the Repeater web part. You must define default values for the relevant properties so that the Repeater web part displays 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**.

6.4 Creating widgets

The basic functionality provided by a widget is determined by the parent web part that is assigned to it. This topic will lead you through the process of creating and configuring a widget that will allow content editors to place their own editable text regions onto pages with editor widget zones.

1. To start, open **Site Manager** -> **Development** -> **Widgets** to access the administration interface used to manage widgets.



The screenshot shows the Kentico CMS 6.0 Site Manager interface. The top navigation bar includes 'Sites', 'Administration', 'Settings', 'Development', 'Tools', 'Dashboard', 'Licenses', 'Support', and 'Buy'. The left sidebar shows the 'Development' menu with 'Widgets' selected. The main area displays the 'Widgets' management interface, including a list of widget categories and a table of widgets.

Development

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

Widgets

New widget Delete selected
New category Export widget
Clone widget

Widgets > All widgets

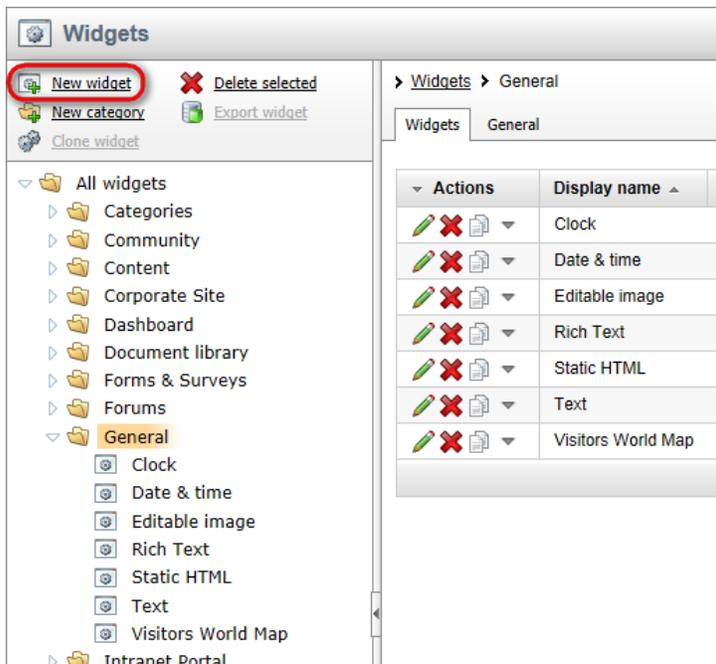
Widgets General

Display name: LIKE Show

Actions	Display name ^
	Abuse report list
	Accordion layout
	Activities
	Analytics chart viewer
	Analytics table viewer
	Article List
	Article List
	Articles RSS feed
	Attachment image gallery
	Bing maps
	Blog comments

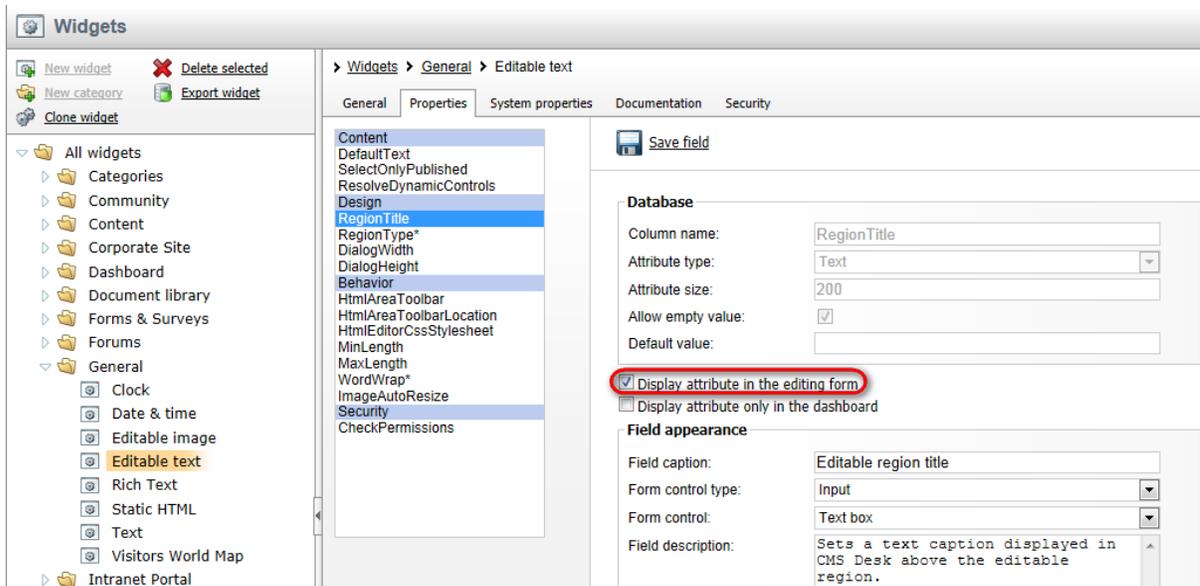
Widgets are grouped into categories. It is important to note that categories are only used for the purposes of organization. The category under which a widget is stored does not affect where the widget can be used or its functionality.

2. Select and expand the **General** category and click the **New widget** link.



A web part selection dialog will be displayed, where you can choose the web part on which the widget will be based. If you ever need to create a widget with completely new custom functionality, you must first develop the appropriate web part. Select the **Text -> Editable text** web part and click the **OK** button to confirm the selection. The widget will now be created.

3. Switch to the **Properties** tab of the new widget, where all the properties taken from the parent web part are listed. By default, none of these properties are available for editing by users when creating or configuring the widget. Select the **RegionTitle** property, check the **Display attribute in the editing form** box and click the **Save field** button. Next, do the same for the **DialogHeight** property.



This setting ensures that the given property will be displayed in the widget configuration dialog. It is recommended to carefully consider which properties should be set as visible, since many widgets are

meant to be accessed by users who are not administrators, even directly on the live site in some cases. Certain properties could be used to configure the widget in an undesirable way, view administration related data or are simply unnecessary for the intended purpose of the widget. A configuration dialog with a small amount of properties will also provide a simpler, easier to use interface.

4. Now open the **Security** tab in order to specify where and by what type of users the widget can be used. By default, the widget is forbidden for all zone types and is not usable by any users. Check the **This widget can be used in editor zones** box and select the **Authenticated users** option.

The screenshot shows the 'Widgets' configuration interface in Kentico CMS. The left sidebar shows a tree view of widget categories, with 'General' expanded and 'Editable text' selected. The main area shows the 'Security' tab for the 'Editable text' widget. The 'Allowed for' section has a table with the following settings:

	Allowed for
This widget can be used in group zones	<input type="checkbox"/>
This widget can be used in editor zones	<input checked="" type="checkbox"/>
This widget can be used in user zones	<input type="checkbox"/>
This widget can be used in dashboard zones	<input type="checkbox"/>
This widget can be used as inline widget	<input type="checkbox"/>
Authenticated users	<input checked="" type="radio"/>
Global Admin only	<input type="radio"/>
Authorized roles	<input type="radio"/>

Below the table, there is a note: 'Please select the authorized roles (available only when you select the "Authorized roles" option above):'. A dropdown menu for 'Site:' is set to 'Corporate site'.

The widget will now be available for all registered users when managing editor type zones. In addition to these security settings, users must also be allowed to access the interface where the given type of widgets is managed. So in this case (page editor zones), access to **CMS Desk** and modify permissions for page content are also required.



Security settings and existing widget instances

Changing the security settings of a widget will not affect the instances of the widget that are already placed on pages by users.

If the permissions for an existing widget are removed by an administrator, it will still be displayed. However, adding new instances of that widget will not be possible.

5. The configuration is now complete and the widget is ready to be placed on a page. Switch to **CMS Desk**, select the **Home** page in the content tree, view it on the **Edit -> Page** tab, click the **Add widget** (🟢) button and choose the newly created **General -> Editable text** widget. In the **Widget properties** dialog, enter the following values into the two properties that were allowed:

- **Editable region title:** Left editable text
- **Editable region height:** 100

The fields in the **Widget container** section are system properties and are available for all editor widgets. You may leave them empty in this case. Now click **OK** and an editable text area will be added to the

page as shown below.

The screenshot shows a web design tool interface with the following components:

- Page Header:** Tabs for Page, Design, Form, and Properties.
- Toolbar:** Includes 'Save' and 'Spell check' buttons, and a rich text editor with icons for source, undo, redo, bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, and image.
- Form Elements:** 'Add widget' and 'Reset to default' buttons.
- Newsletter Widget:**
 - Icon: Envelope with a plus sign.
 - Title: 'Newsletter'.
 - Fields: 'First Name:', 'Last Name:', and 'E-mail:'.
 - Button: 'Subscribe'.
- Left Editable Text:** A box containing the text 'Sample text...'.
- Main Content Area:**
 - Section Header:** 'Welcome to the sample Corporate Site'.
 - Text:** 'If you are new to Kentico CMS, please read the following information before you start exploring the website:'.
 - Section Header:** 'Default user name and password'.
 - Text:** 'You can sign in to the system's administration interface using the links in the header of the page or by going to the following addresses:'.
 - Links:**
 - CMS Desk:** <http://<your domain>/CMSDesk>
 - Site Manager:** <http://<your domain>/CMSSiteManager>
 - Text:** 'On the logon page that appears, use the following default credentials:'.
 - Credentials:**
 - User name:** administrator
 - Password:** leave the password blank

Part



VII

Managing styles and design

7 Managing styles and design

7.1 CSS styles

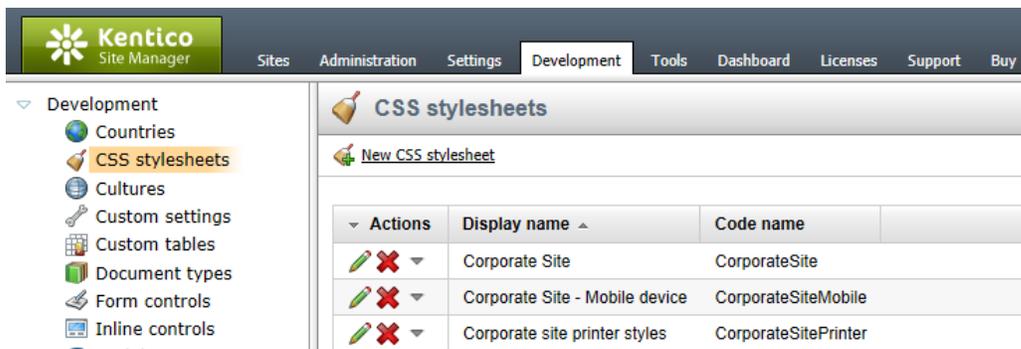
The design of the website relies on standard CSS styles. Each website has its global CSS stylesheet that can be chosen in **Site Manager -> Sites -> ... edit site ... -> General**. Here you can also choose a different stylesheet for the site's WYSIWYG editors.

In addition, each page can override the global website stylesheet. You can assign a stylesheet to a specific page by editing its corresponding document in **CMS Desk -> Content -> Properties -> General** and using the **CSS stylesheet** selector.

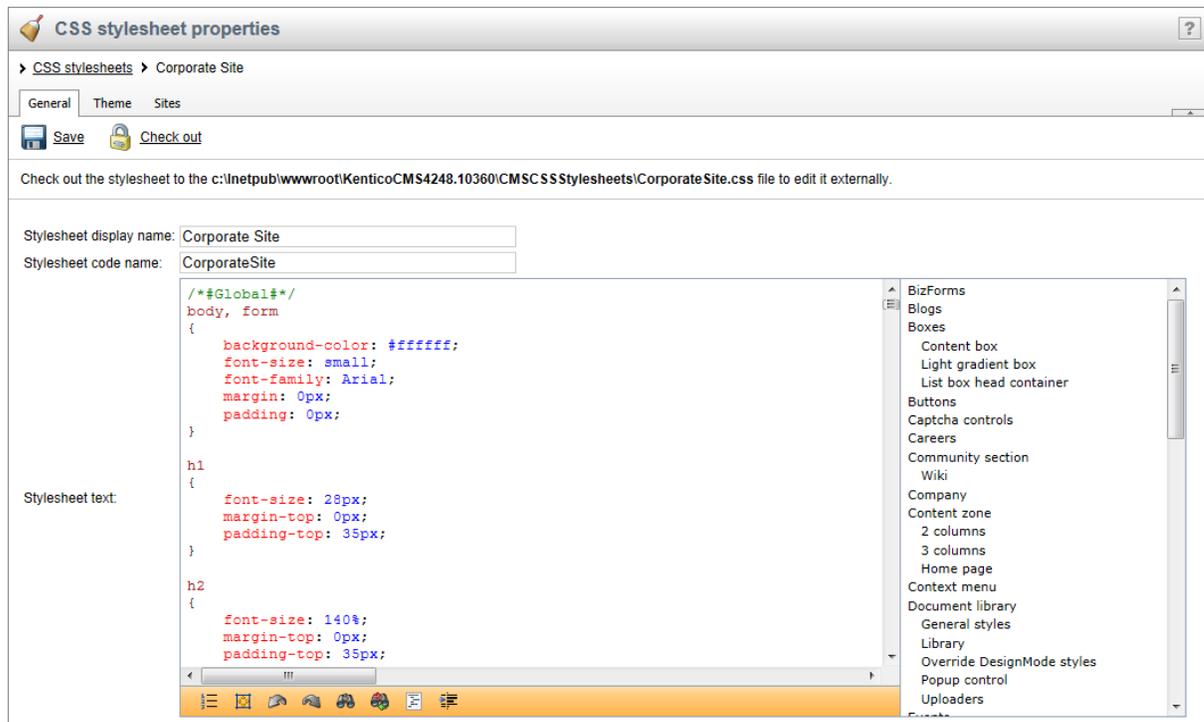
The actual content of the CSS stylesheets can be managed in Site Manager. When you are in CMS Desk, you can easily switch to Site Manager by clicking the **Site Manager** link in the header:



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



Edit () the **Corporate Site** stylesheet:



CSS stylesheet properties

> CSS stylesheets > Corporate Site

General Theme Sites

Save Check out

Check out the stylesheet to the c:\inetpub\wwwroot\KenticoCMS4248.10360\CMSCSSStylesheets\CorporateSite.css file to edit it externally.

Stylesheet display name: Corporate Site

Stylesheet code name: CorporateSite

Stylesheet text:

```

/*#Global#*/
body, form
{
    background-color: #ffffff;
    font-size: small;
    font-family: Arial;
    margin: 0px;
    padding: 0px;
}

h1
{
    font-size: 28px;
    margin-top: 0px;
    padding-top: 35px;
}

h2
{
    font-size: 140%;
    margin-top: 0px;
    padding-top: 35px;
}

```

Right-hand pane (Tree View):

- BizForms
- Blogs
- Boxes
 - Content box
 - Light gradient box
 - List box head container
- Buttons
- Captcha controls
- Careers
- Community section
- Wiki
- Company
- Content zone
 - 2 columns
 - 3 columns
 - Home page
- Context menu
- Document library
 - General styles
- Library
- Override DesignMode styles
- Popup control
- Uploaders

Please note: you may need to refresh your browser in order to see the latest version of the style.



Browser and language-specific styles

Pages automatically have CSS classes assigned to their `<body>` element according to the characteristics of their language (its text direction and specific culture) and depending on the browser in which they are currently viewed. For example:

```
<body class="LTR IE IE9 ENUS">
```

As you can see, four types of classes are added:

- **Text direction** - the *LTR* class is assigned for left-to-right languages and *RTL* for right-to-left.
- **Browser type** - this class is added according to the browser in which the page is currently opened. The following classes are used:

Browser	Class name(s)
Internet Explorer	IE
Firefox	Gecko
Google Chrome	Chrome, Safari
Safari	Safari
Opera	Opera

- **Browser version** - the class name is the same as for the browser type, but with the number of the browser's major version appended, e.g. *IE9*, *Gecko5* etc.
- **Culture** - the name of the class is added based on the culture code of the page's content (without the hyphen), for example *ENUS* for pages using the *en-US* culture.

This feature allows you to style page elements differently according to the browsing environment of the current visitor. You can define styles for any combination of the classes mentioned above.

For example, you can add the following into a website's stylesheet:

[CSS]

```
.IE8 .MyFont
{
    font-size: 20px;
}

.Opera .MyFont
{
    font-size: 18px;
}
```

Now elements styled using the *MyFont* class will have a different font size when viewed in the Internet Explorer 8 or Opera (all versions) browsers.

7.2 App themes

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

Themes must be defined in a folder located under the **App_Themes** directory. The name of this folder needs to be the same as the code name of the site's CSS stylesheet. So if you use the *Green* stylesheet on your site, your themes must be stored in the *App_Themes\Green* sub-folder.

Skins for your controls must be added to the **Default.skin** file under this folder. Here's an example of a skin for the **CMSCalendar** control / **Calendar** web part:

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

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



Where should I store website design files?

It's recommended to store all images, flash movies and other resources that are part of the website design template in the **App_Themes/<stylesheet code name>** folder. This ensures that the files are exported together with the stylesheet if you deploy 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 on 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 applied CSS styles. Here's an example of the CSS styles for the drop-down menu:

```
.zoneMenu .CMSListMenuUL
{
    list-style: none;
    margin: 0px;
    padding: 0px;
    position: relative;
}

.zoneMenu .CMSListMenuUL li
{
    float: left;
    padding: 0px 22px 0px 0px;
}

.zoneMenu .CMSListMenuUL li a
{
    color: #fff;
    text-decoration: none;
    display: block;
    height: 23px;
    font-size: 16px;
    line-height: 23px;
    padding: 0px 8px;
    border: 1px solid transparent;
    font-family: Arial;
}

.zoneMenu .CMSListMenuUL .CMSListMenuHighlightedLIfirst a,
.zoneMenu .CMSListMenuUL .CMSListMenuLIfirst a
{
    padding-left: 0px;
}

...
```

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. Select **Top menu styles -> Horizontal** in the right navigation panel. Change the highlighted line:

```
.zoneMenu .CMSListMenuHighlightedLI a,  
.zoneMenu .CMSListMenuHighlightedLIfirst a  
{  
  color: #8cd2f8 !important;  
  background-color: orange;  
  text-decoration: none;  
}
```

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



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 select **Home** from the content tree. Click **Properties -> Menu**. Here you can define:

- **Menu caption** - the name of the document when it's displayed in a menu.
- **Show in navigation** - indicates if the document should be displayed by navigation controls.
- **Show in sitemap** - indicates if the document should be displayed in the sitemap.
- **Menu actions** - sets special behaviour for when the menu item is clicked by a user. The link can be disabled, a JavaScript command can be executed, or a specified URL can be opened.
- **Menu design** - sets styles applied to the document's menu item (standard, mouse-overed and highlighted).

Enter the following value into the **Menu item style** value (in 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



VIII

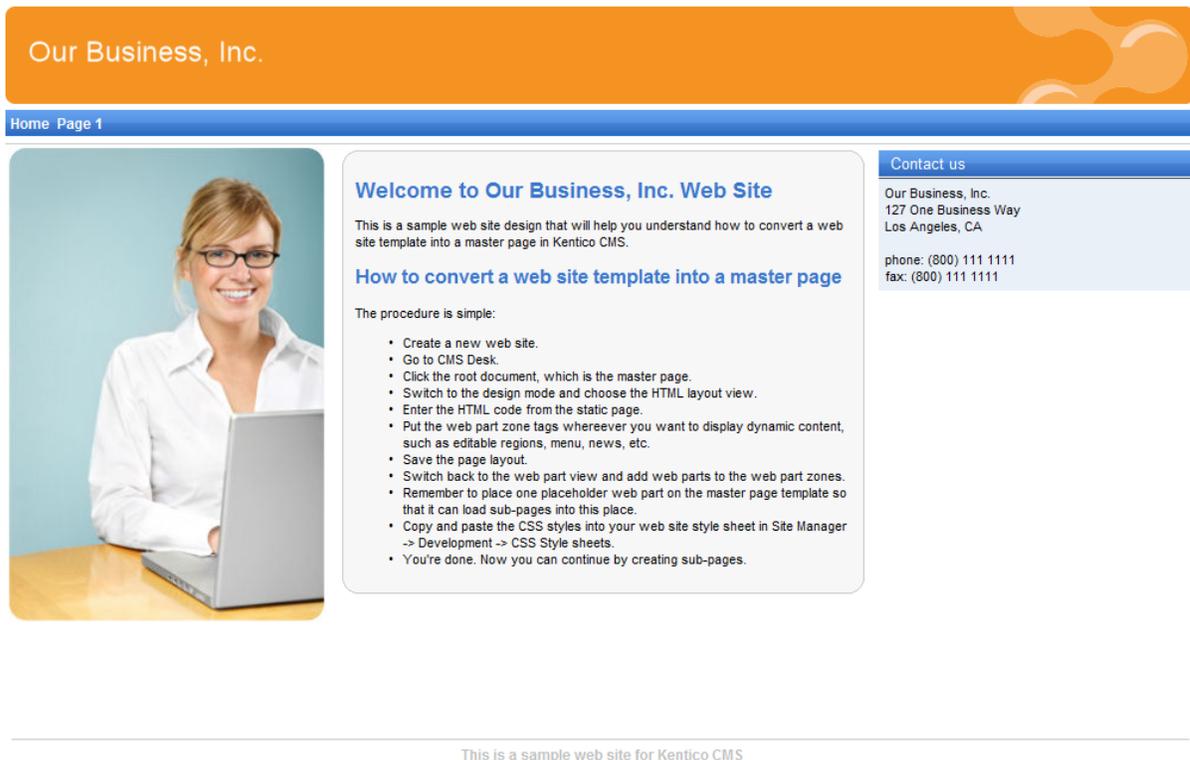
Creating a new site using the portal engine

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 website. 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 website 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 the **C:\Program Files\Kentico CMS\<version>\CodeSamples\SampleWebTemplate** folder. The template consists of the home.htm file, a styles folder and an app_themes folder with images.

8.2 Creating a new web site using the wizard

The following topics assume that you have previously installed a sample Corporate Site. We will leave the existing website and add a new website that will run on `http://127.0.0.1`.

Multiple sites and Visual Studio's built-in web server

If you are 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 any domain other than localhost,

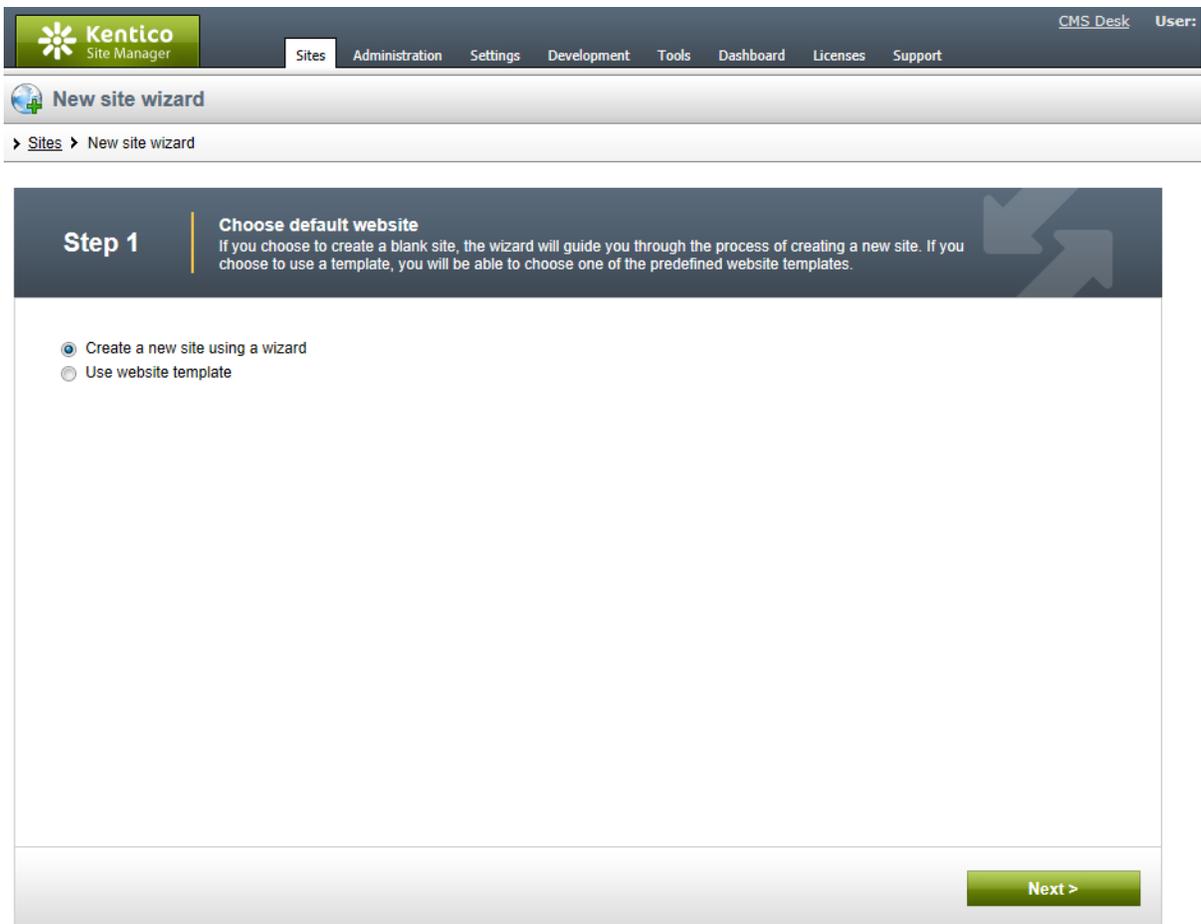
you will use the **localhost** domain again.

Sign in as Administrator to **Site Manager** -> **Sites** and click  **New site wizard**.



Actions	Site name	Status	Main domain	Documents	Default culture
    	Corporate Site	Running	localhost	526	English - United States

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



Step 1 | **Choose default website**
 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 website templates.

Create a new site using a wizard
 Use website template

Next >

In the second step, enter the following details:

- **Site display name:** My website
- **Site code name:** mysite
- **Domain name:** 127.0.0.1 (if you are using Visual Studio built-in web server, set the default value to *localhost*)
- **Site 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 website. The Domain field must contain the domain that you will use to access the website during development (you may change it when the site goes live). The default culture is the main language of the website.

Site display name:

Site code name:

Domain name:

Site culture:

< Previous Next >

In the third step, you are 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

- Website**
 - 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
- Do not import objects where parent object is missing
- Import tasks (recommended)
- Import files (recommended)
- Import global folders
- Import assembly files

< Previous Next >

In the fourth step, the progress of object import is displayed.

Step 4

Import progress

Objects are being imported.

Importing 'Roles' objects
Importing 'Website' objects
Importing 'CSS stylesheets' objects
Importing 'Settings' objects
Importing 'Settings categories' objects

Objects are being imported

< Previous **Cancel** Next >

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	Blank master page Generic default page template.
no image	Corporate Site - Master page
no image	Corporate Site Mobile - Master page
	Top logo and left menu Master page template with logo on the top and menu on the left side.

[Next >](#)

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

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

New Delete Up Down
Content management

My web site
Home

Page properties

Page name: Home
Page template: Simple text Select... Inherit template

OK

Next >

Click **My website** and click **New**. Add a new page called **News** and select the **News -> News list** page template. Click **OK** to save it.

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

New Delete Up Down
Content management

My web site
Home
News

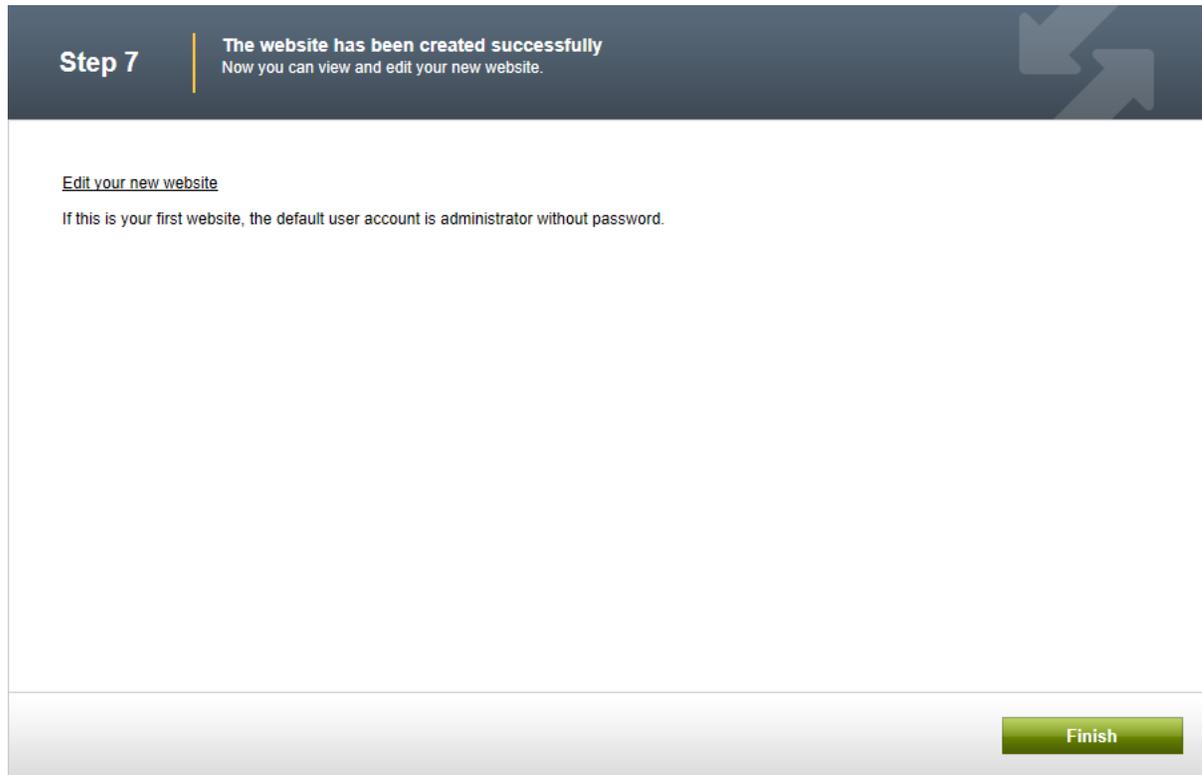
Page properties

Page name: News
Page template: News list Select... Remove template

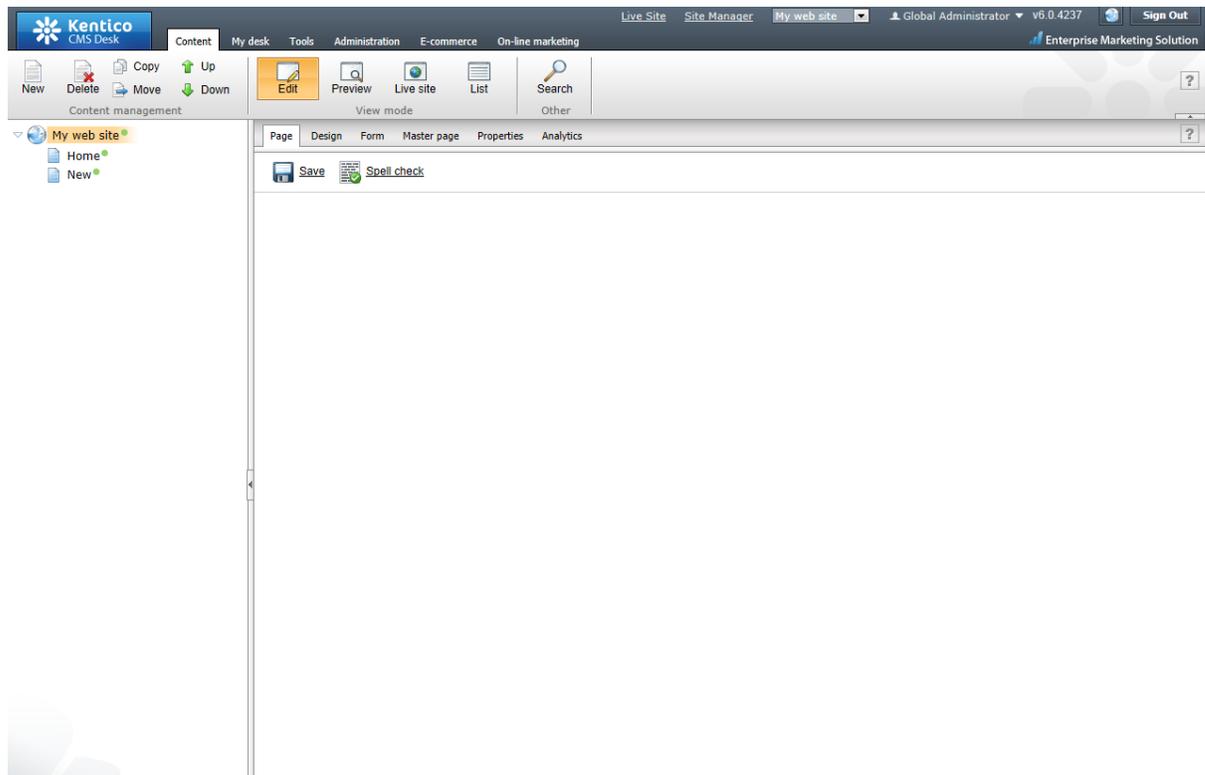
OK

Next >

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



Click the **Edit your new website** 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 authentication is not shared over different domains by default. After you sign in, you will see your new, empty website:

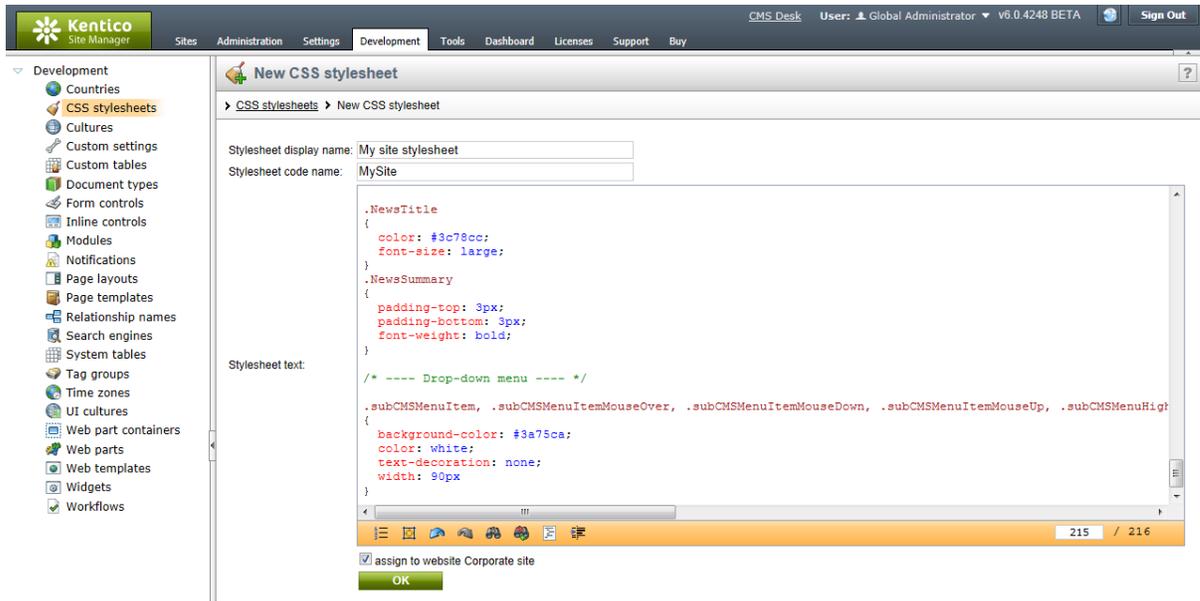


You have created the base for your new website. In the next topics, you will learn how to implement the required design.

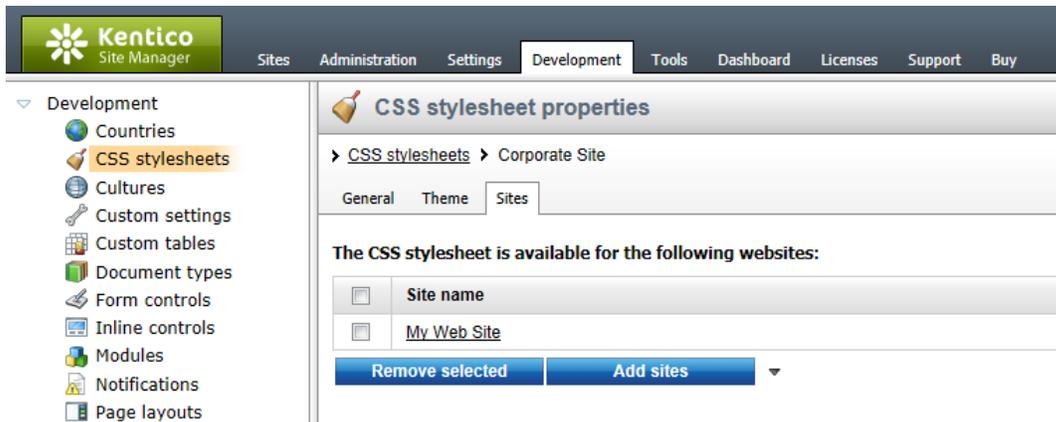
8.3 Creating a CSS stylesheet

Before we start editing our new website, we will prepare the CSS styles and images based on our website 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 **C:\Program Files\Kentico CMS\<version>\CodeSamples\SampleWebTemplate\Styles\main.css**



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



Go to **Site Manager -> Sites** and **Edit** (✎) the properties of **My website**. Select **My site stylesheet** in the **Site CSS stylesheet** drop-down list and click **OK**. This will ensure that the stylesheet is used on all pages of your new website.

Kentico Site Manager

Sites Administration Settings Development Tools Dashboard

Site properties

> Sites > My Web Site

General Domain aliases Cultures Off-line mode

Site display name: My Web Site

Site code name: MyWebSite

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 **C:\Program Files\Kentico CMS\<version>\CodeSamples\SampleWebTemplate\app_themes\MySite** folder to **<your web project folder>\App_themes**. The folder contains graphics for this website template. It will ensure that the images are exported as a part of the website if you decide to move the website in the 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 website. In real-world scenarios, you will need to adjust the paths manually. **The URLs of images in the CSS stylesheets are always relative to the location of the web project.**

The URL of the CSS stylesheet is:

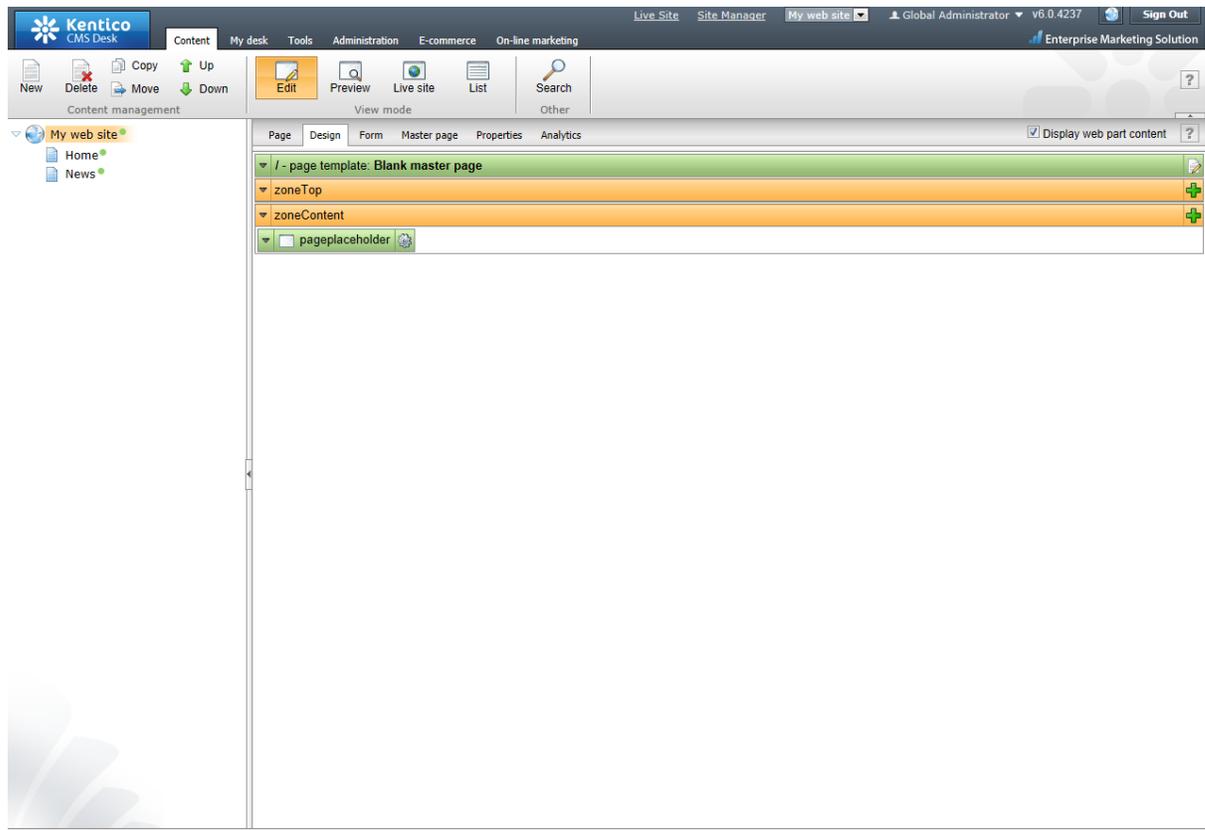
```
<web project>/CMSPages/GetResource.ashx?stylesheetname=MySite
```

which means that you need to link to files in the **App_themes** folder like in the example below:

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

Page Design Form Master page Properties Analytics Display web part content ?

Page layout

Save Check out to file

Check out the layout to the c:\inetpub\wwwroot\KenticoCMS4237.20467\CMSTemplateLayouts\cms\blankmasterpage.ascx file to edit the layout externally.

Layout type: ASCX

Use the following expression to add a new web part zone: <cms:CMSWebPartZone ID="zoneTopRight" runat="server" /> (ID must be unique)

```
<div class="MainDiv">
<!-- logo -->
<br />
<div class="Logo">
 
</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;">
<!-- center box -->
<table cellpadding="0" cellspacing="0" border="0" class="ContainerWithCorners" width="100%">
<tr class="ContainerWithCornersRow">
<td class="ContainerWithCornersTopLeft">&nbsp;&nbsp;&nbsp;</td>
<td class="ContainerWithCornersTop">&nbsp;&nbsp;&nbsp;</td>
<td class="ContainerWithCornersTopRight">&nbsp;&nbsp;&nbsp;</td>

```

2 / 100

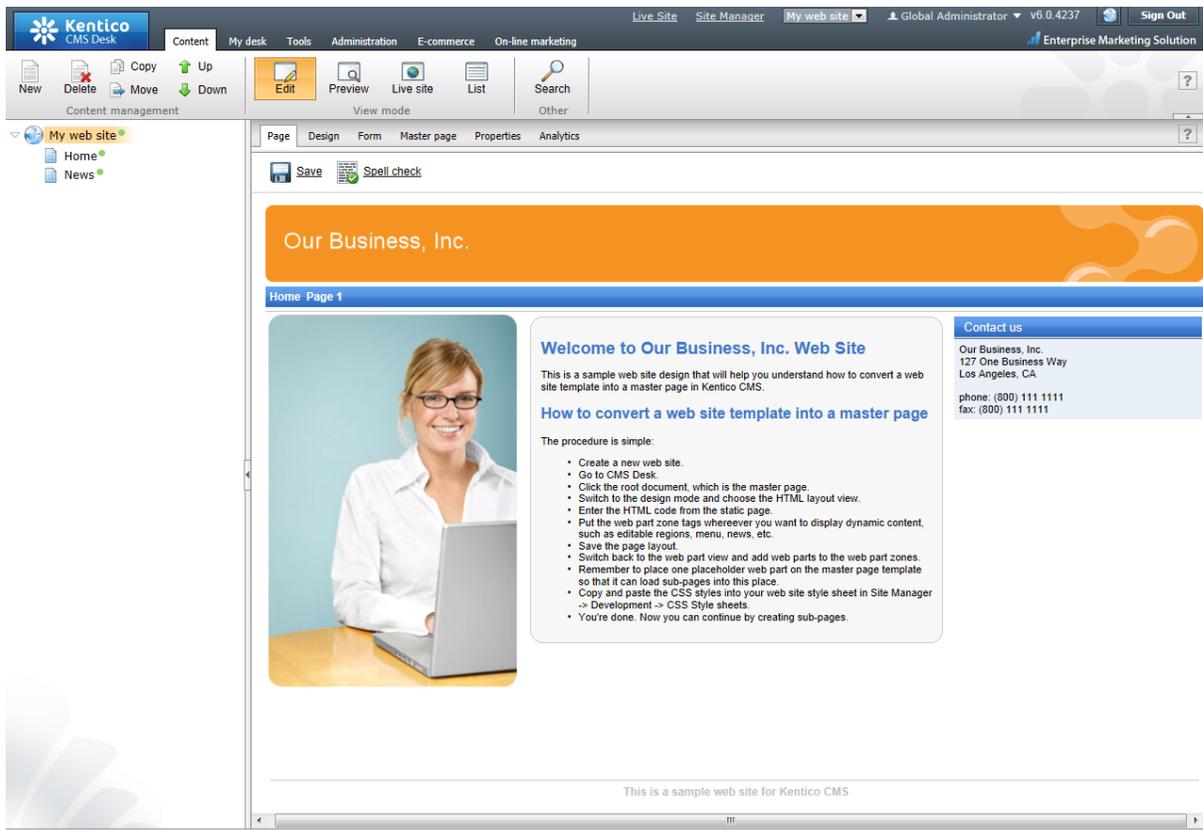
Click **Save**.



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

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



Adding web part zones

Now we need to replace the static html code with web part zones, so that we can place 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">
  <c1:CMSWebPartZone ID="zoneMenu" runat="server" />
</div>
```

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

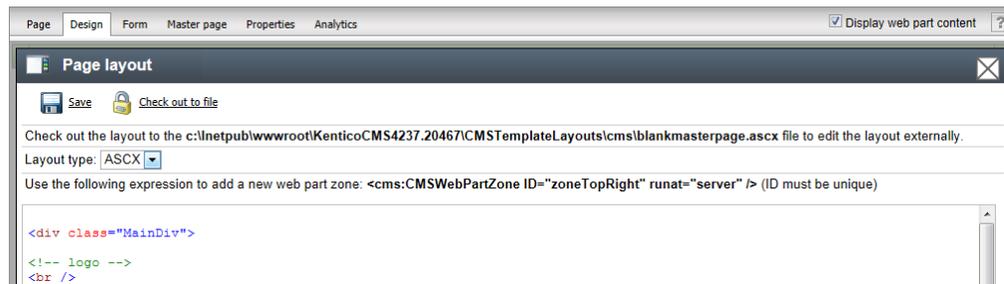
```
<cc1:CMSWebPartZone ID="zoneMain" runat="server" />
```

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 a file on your local 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 will be deleted from 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 a 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 edited in **HTML layout** mode on the **Design** tab).

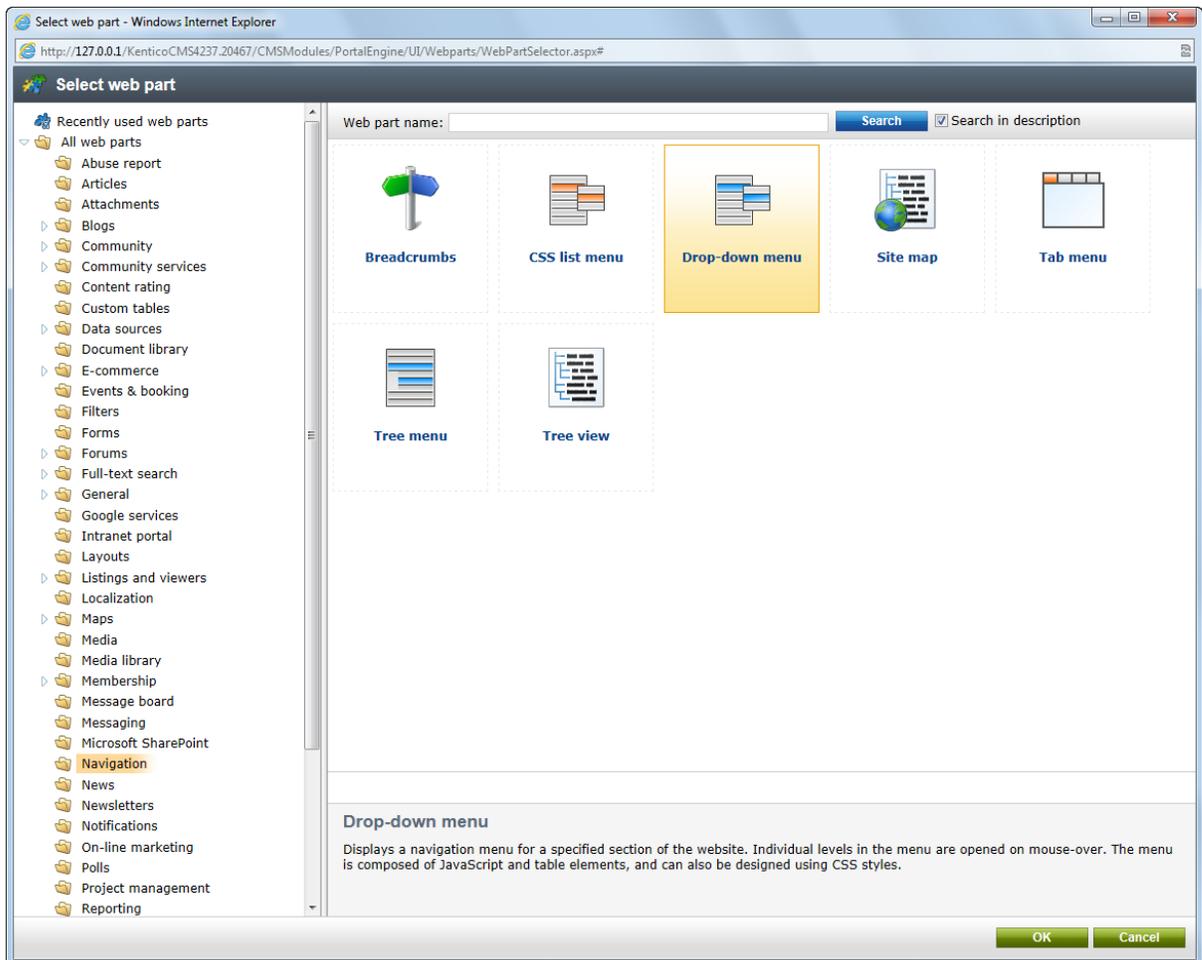
```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
<html>
<head>
<title>
Corporate site - </title>
<meta http-equiv="pragma" content="no-cache" />
<link href="/KenticoCMS_4248.10360/CMSPages/GetResource.ashx?stylesheetname=CorporateSite" type="text/css"
rel="stylesheet"/>
<link href="/KenticoCMS_4248.10360/CMSPages/GetResource.ashx?
stylesheetfile=/KenticoCMS_4248.10360/App_Themes/CorporateSite/Dialogs.css" type="text/css" rel="stylesheet"/>
</head>
<body class="LTR IE IE9 ENUS"
Layout type: ASCX
Use the following expression to add a new web part zone: <cms:CMSWebPartZone ID="zoneTopRight" runat="server" /> (ID must be unique)
<!-- Top info -->
<div class="zoneTopInfo">
<div class="inner">
<div class="left"><oc1:CMSWebPartZone ID="zTI" runat="server" /></div>
<div class="right"><oc1:CMSWebPartZone ID="zTIR" runat="server" /></div>
<div class="clear"></div>
</div>
</div>
<div class="zoneTopWrap">
<div class="inner">
<!-- Logo -->
<div class="zoneLogo">
<oc1:CMSWebPartZone ID="zL" runat="server" />
</div>
<!-- Top zone -->
<div class="zoneTop">
<oc1:CMSWebPartZone ID="zT" runat="server" />
</div>
</div>
</body>
</html>

```

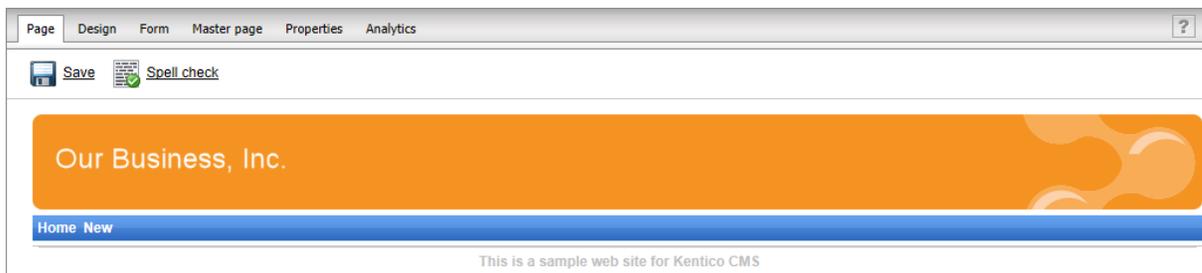
8.5 Main menu

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 will be opened. Set the value of the **CSS prefix** property to ; *sub*. This means that CSS classes used by the menu for the second menu level will have the *sub** prefix. Set the **Design -> Layout** property to **Horizontal**. Click **OK**.

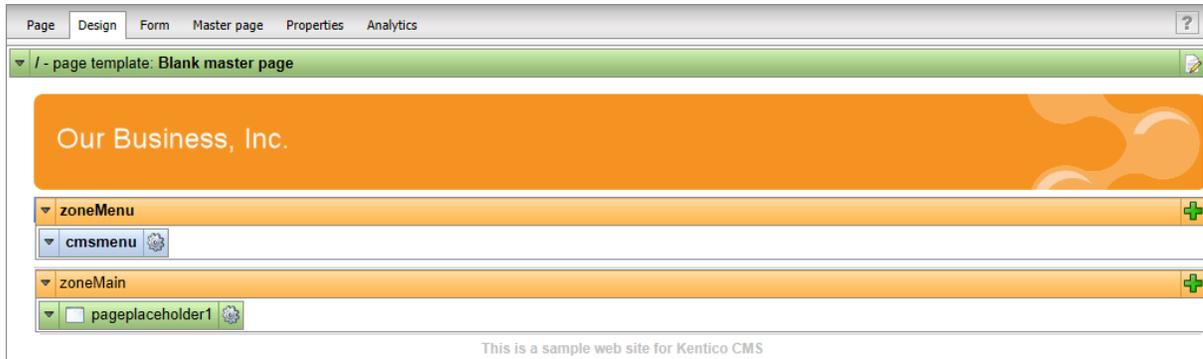
When you switch to the **Page** tab, you will see a page with a dynamic menu that displays the child pages: **Home** and **News**.



8.6 Page placeholder

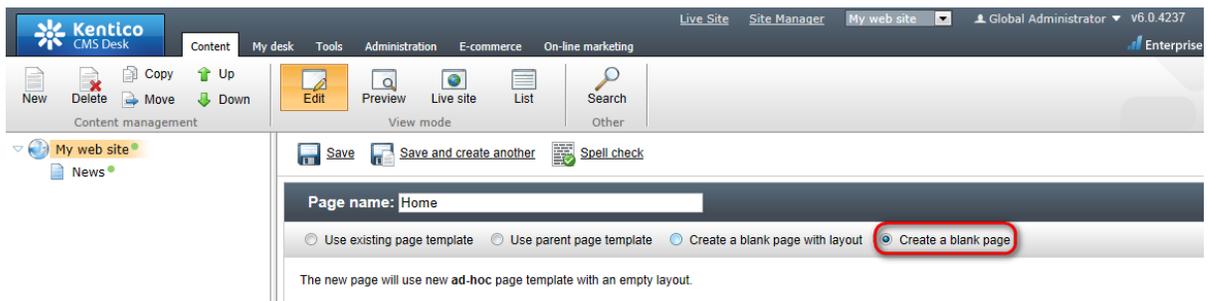
Now we need to add a **Page placeholder** web part that ensures the loading of sub-pages (such as the Home or News page) into the master template.

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 **Home** into the **Page name** field and choose the **Create a blank page** option.



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 the 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;">
      <c1:CMSWebPartZone ID="zoneCenter" runat="server" />
    </td>
    <!-- right column -->
    <td style="padding: 3px 0px 0px 5px;width:270px">
      <c1:CMSWebPartZone ID="zoneRight" runat="server" />
    </td>
  </tr>
</table>
```

```
</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:

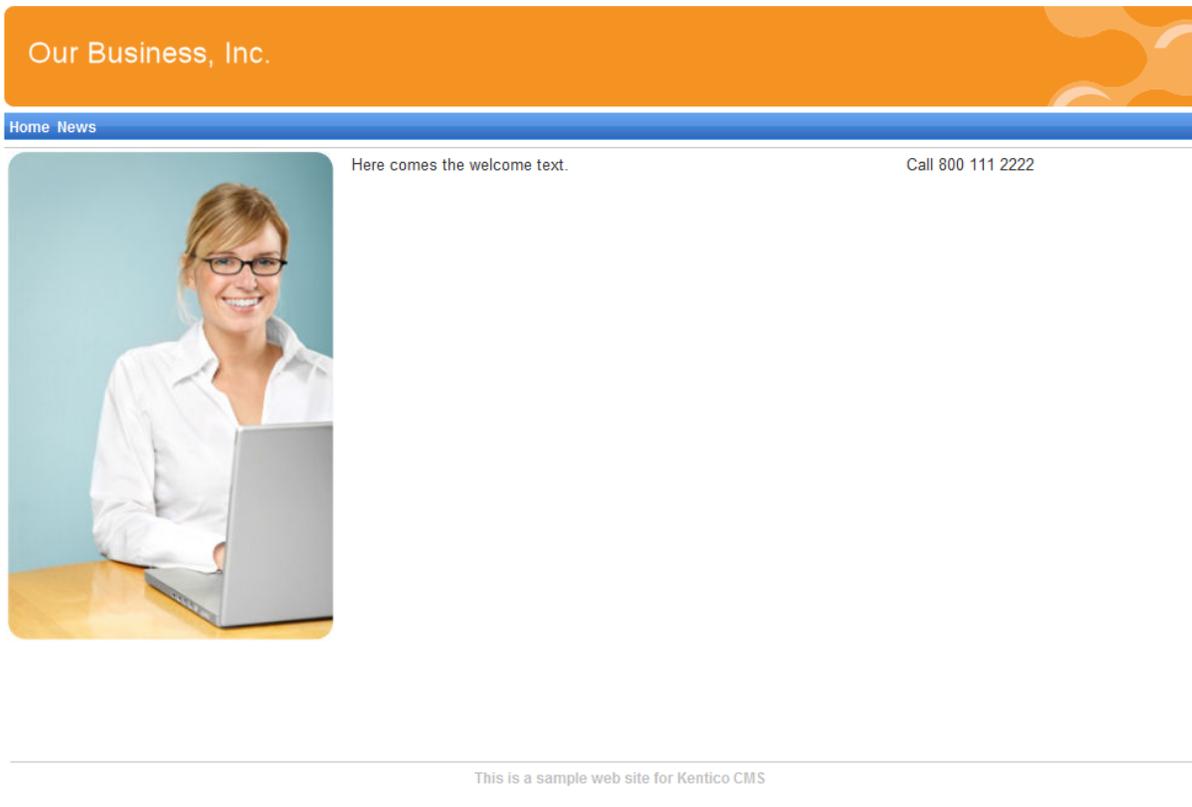
- **Editable region title:** Contact text
- **Editable region height:** 100

Click **OK**. Select 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 website:



Configuring the website home page

When a site visitor comes to the root of your website (e.g. to *http://www.example.com*), the system needs to know which page should be displayed by default as the home page. Go to **Site Manager -> Settings**, select **My website** in the **Site** drop-down menu, click the **Content** setting category and make sure the value of **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 Kentico Site Manager interface. The left sidebar contains a tree view of settings categories, with 'Content' selected. The main content area is titled 'Content' and contains several sections: 'Page not found', 'Multilingual', 'Web site content', and 'Metadata'. In the 'Web site content' section, the 'Default alias path' field is highlighted with a red box and contains the value '/Home'. The 'Inherit from global settings' checkbox for this field is unchecked. Other fields in the 'Web site content' section include 'Page not found for non-published documents', 'Page not found URL', and 'Log page not found exception'. The 'Metadata' section includes fields for 'Page key words prefix', 'Page description prefix', 'Page title format', 'Page title prefix', and 'Control element'.

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 website gray box
- **Code name:** MyWebSiteGrayBox
- **HTML 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">

```

- **HTML 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 website**. Now create another container:

- **Display name:** My website blue box
- **Code name:** MyWebSiteBlueBox
- **HTML 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:**

```

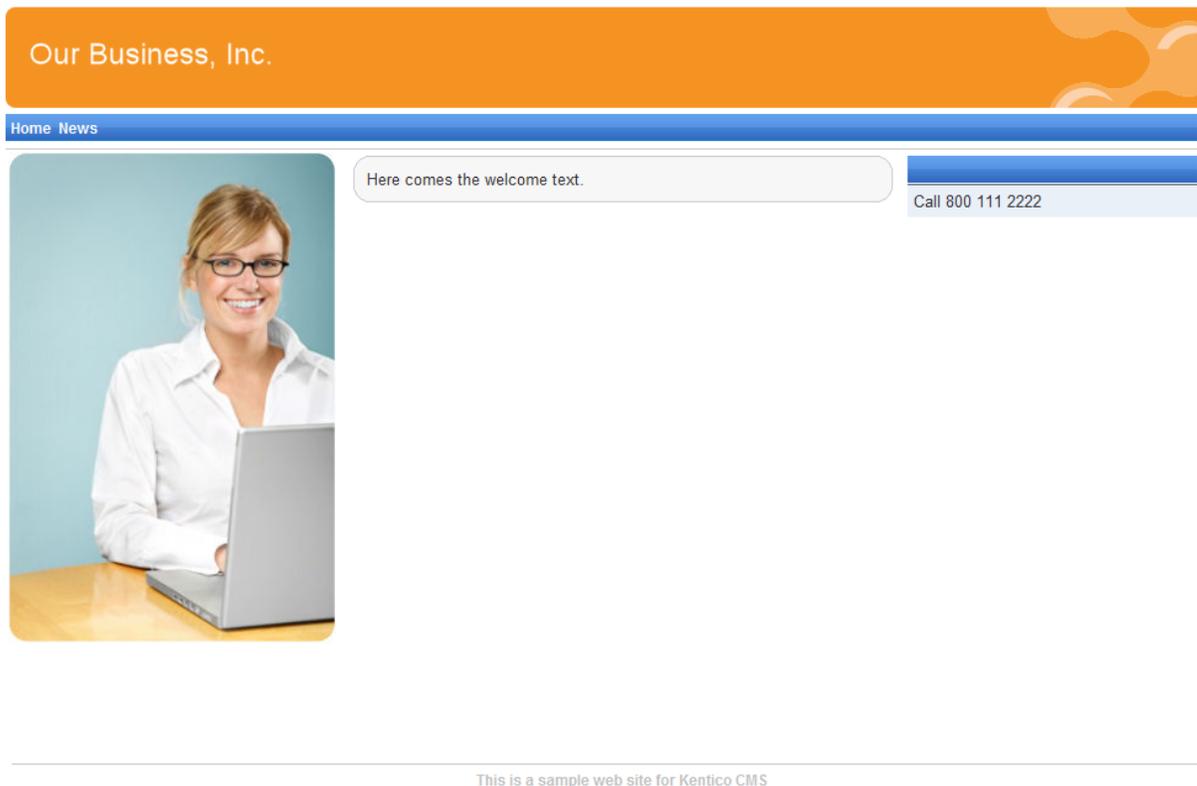
  </td>
</tr>
</table>

```

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

Configure the **editabletext1** web part in the right column and set the **Web part container** property to *My website 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 **News**, choose to create a **News** document and enter the following text:

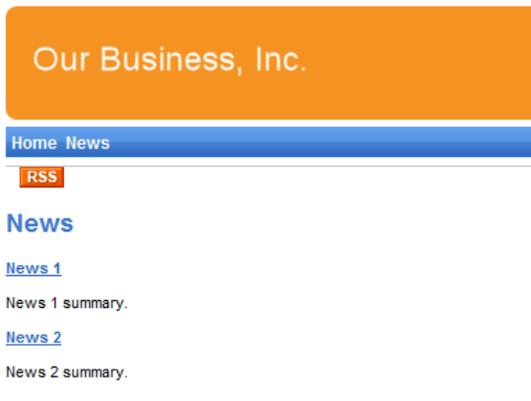
- **News title:** News 1
- **Release date:** click the date-time picker and click **Now** and click **OK**.
- **News summary:** News 1 summary.
- **News text:** News 1 text.
- **Teaser, Publish from/to:** leave the fields blank.

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 **/News** page 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:

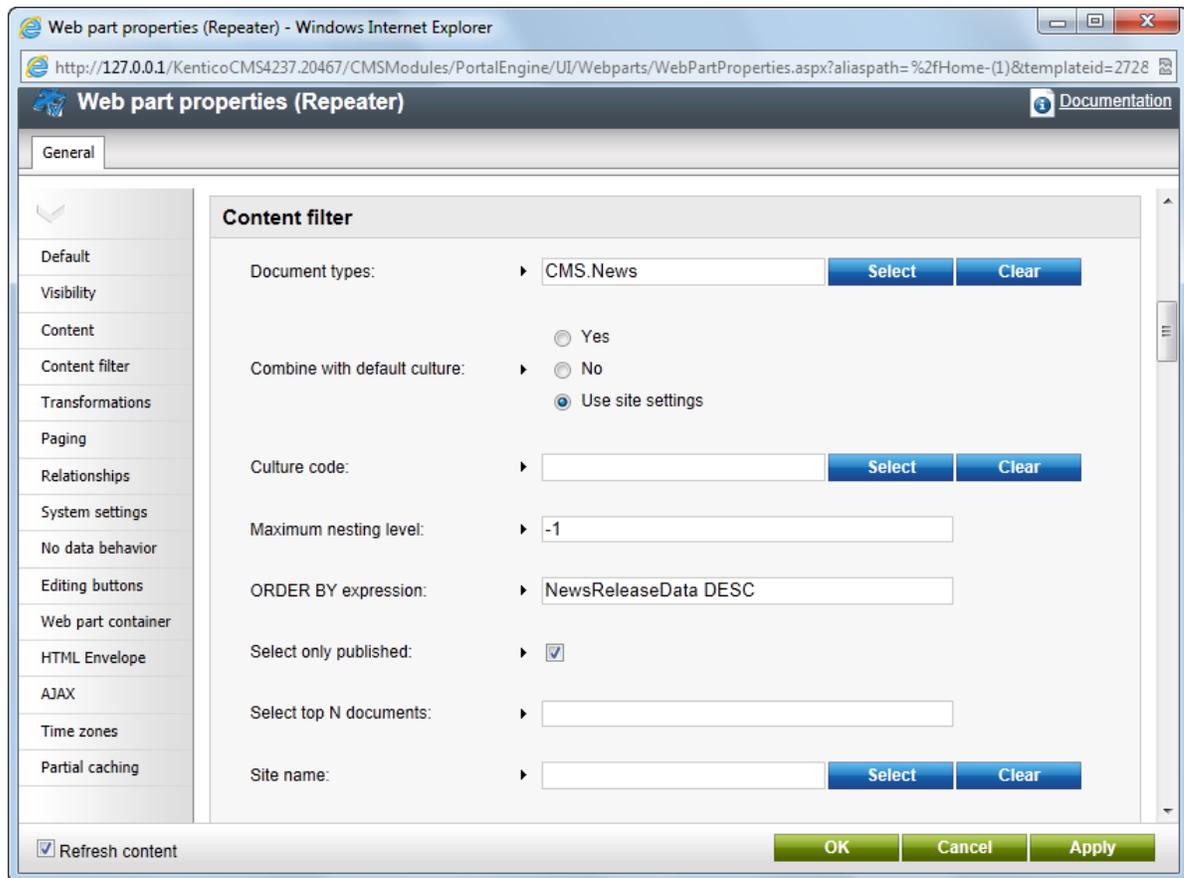


The breadcrumbs now show the current path on the website: **News > News 1**. The position is also reflected in the URLs:

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

This makes the website 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 displays 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 to 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 to render the displayed items. The transformation works similar to the ItemTemplate element that you would normally use with a Repeater control. Transformations can be managed in the *Site Manager -> Development -> Document types -> ... edit document type ... -> Transformations* interface.
- **Selected item transformation** = cms.news.default - this property says which transformation should be used for the detail view of the selected document.
- **Paging** - if you plan to have a large amount of 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 select some particular news item, such as **/News/News 1**, the **NewsRepeater** web part uses the **cms.news.detail** transformation instead and displays the details.



Path expressions

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

/%	All documents on the website.
/news/%	All documents under /News.
/news/news1	The News1 document.
./%	All items under the current document.
./logo	The Logo document under the current document.
./images/%	All images under the Images child document.
../contacts/%	All documents under the Contacts document on the same content level.
/{{0}}/%	All documents under the current first level document. <u>Example:</u> 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 *Services* as the **Page name** and choose to **Create a blank page with layout** using the **Two columns - 20/80** layout. Leave the **Copy this layout to my page template** box checked.

Page name:

Use existing page template
 Use parent page template
 Create a blank page with layout
 Create a blank page

Layout name:

 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

1 2 3

Select layout

Copy this layout to my page template

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 website and it ensures that the image will be displayed correctly whether you run the websites 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:

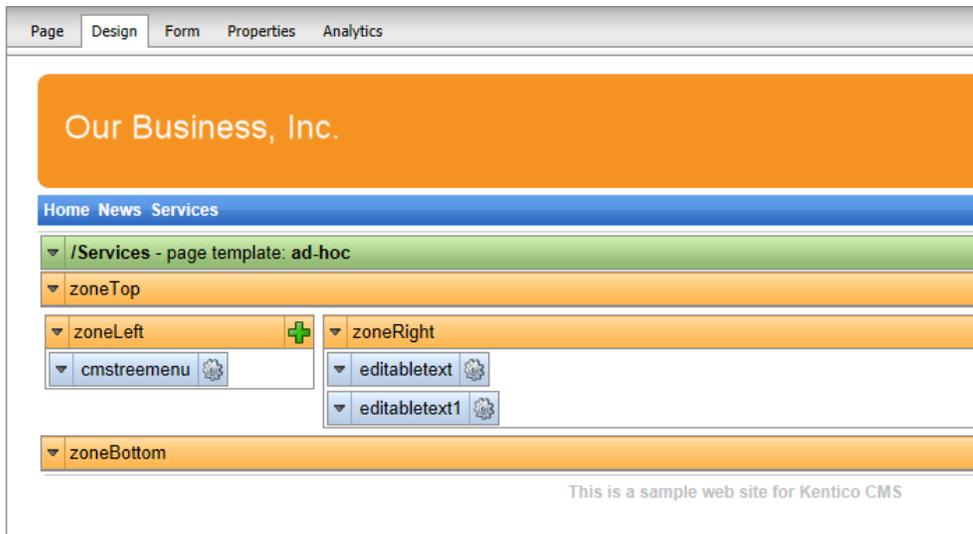
- **Editable region title:** Header
- **Editable region type:** Text box
- **Editable region width:** 600
- **Content before:** <h1>
- **Content after:** </h1>

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

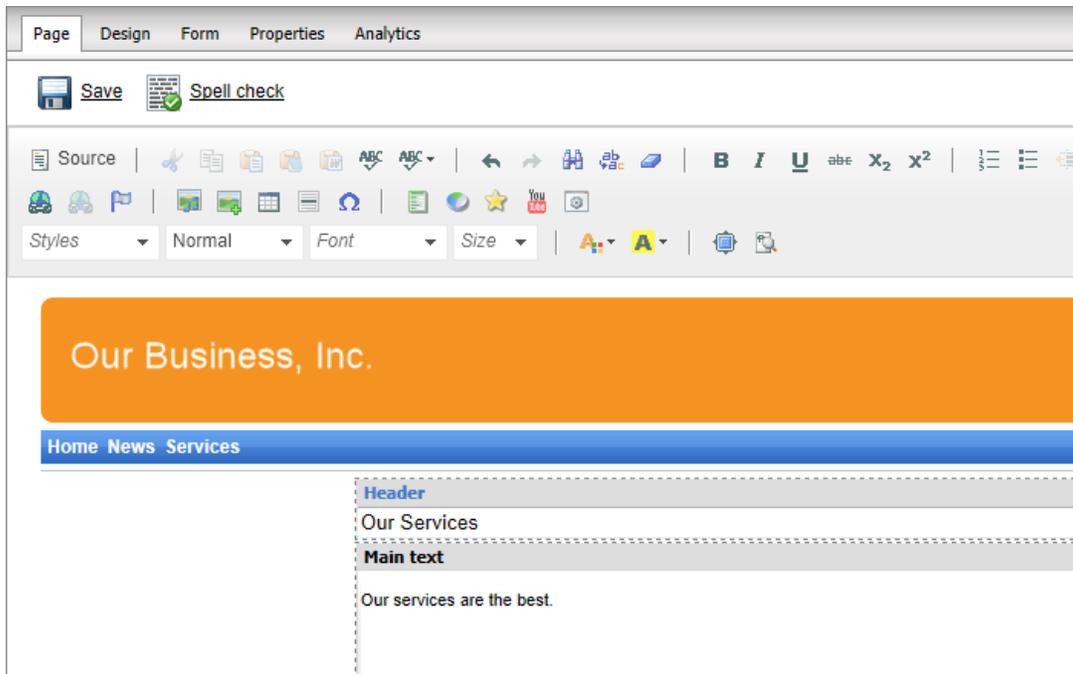
- **Editable region height:** 300

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



Entering text

Click the **Page** tab and enter some text into the 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** in the template dialog.

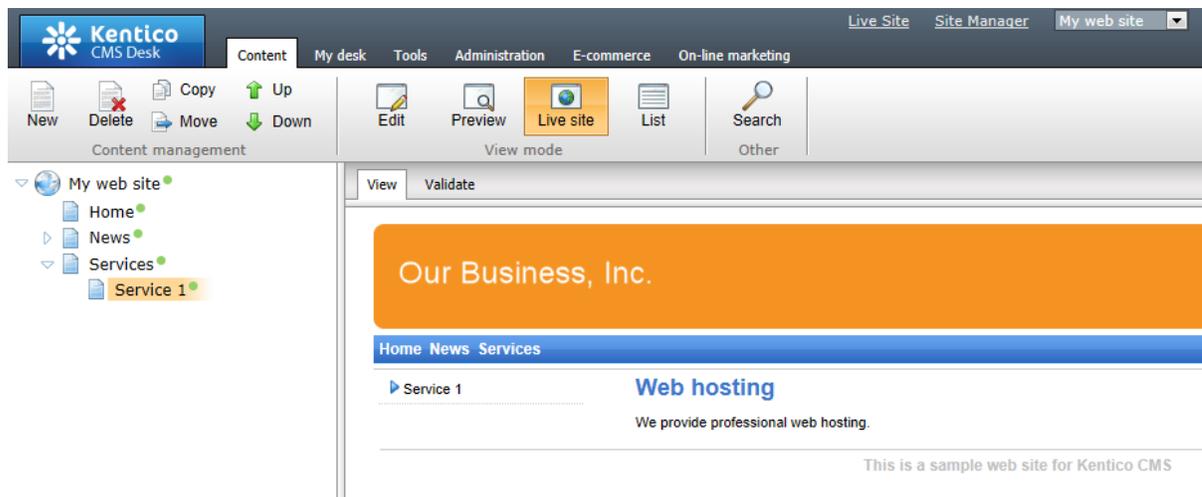
Adding sub-pages

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 topic, 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 *Computer* document type and how to display a list of computers on the site. You will also learn how to write transformations.

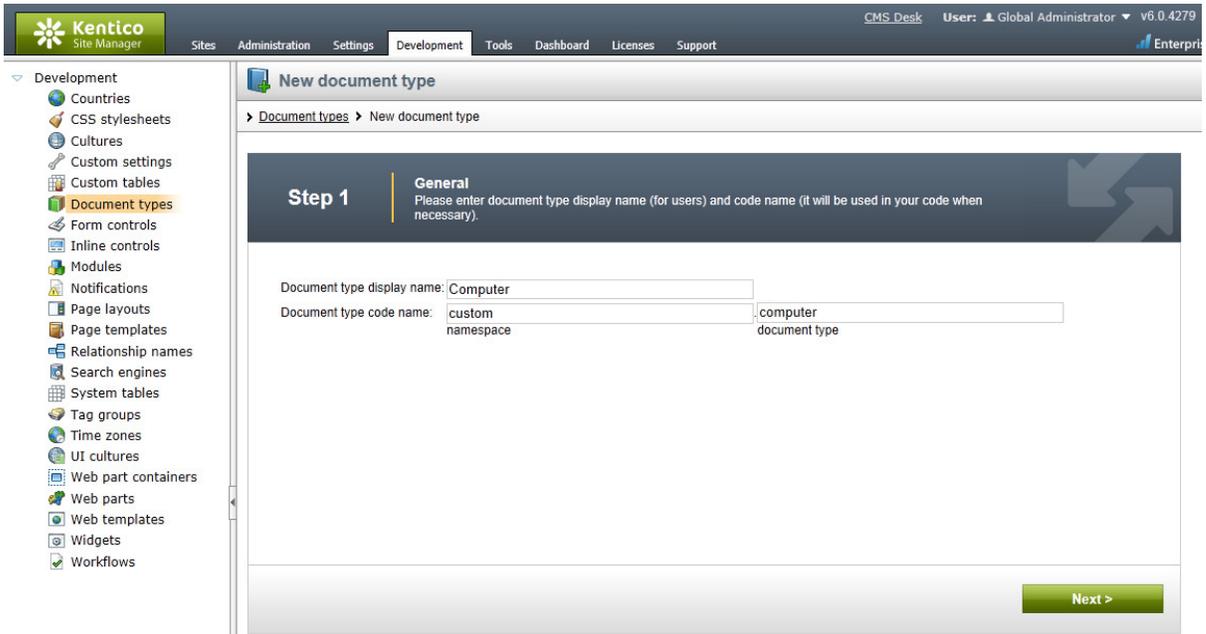
8.11.2 New document type

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 a 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 will be 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 in the administration interface*)
- **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 Kentico CMS 6.0 Site Manager interface. The top navigation bar includes 'Sites', 'Administration', 'Settings', 'Development', 'Tools', 'Dashboard', 'Licenses', and 'Support'. The 'Development' menu is expanded, showing various options like 'Countries', 'CSS stylesheets', 'Cultures', 'Custom settings', 'Custom tables', 'Document types', 'Form controls', 'Inline controls', 'Modules', 'Notifications', 'Page layouts', 'Page templates', 'Relationship names', 'Search engines', 'System tables', 'Tag groups', 'Time zones', 'UI cultures', 'Web part containers', 'Web parts', 'Web templates', 'Widgets', and 'Workflows'. The 'Document types' option is highlighted. The main content area is titled 'New document type' and shows 'Step 1: General'. Below the title, there is a prompt: 'Please enter document type display name (for users) and code name (it will be used in your code when necessary)'. There are two input fields: 'Document type display name' with the value 'Computer' and 'Document type code name' with the value 'custom.namespace.computer.document.type'. A 'Next >' button is located at the bottom right of the form.

Click **Next**.

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

 **New document type**

> [Document types](#) > New document type

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:

Inherits fields from document type:

The document type is only a container without custom fields

[Next >](#)

Click **Next**.

The wizard has created a new database table for computers. Now you need to define the fields of the document type (columns of the table). Use the **New attribute** (+) button to create the following fields. For each field, enter the values, click  **Save field** and repeat the procedure until you have all the listed fields defined.

- **Column name:** ComputerName
- **Attribute type:** Text
- **Attribute size:** 200
- **Field caption:** Computer name
- **Form control type:** Input
- **Form control:** Text box

- **Column name:** ComputerProcessorType
- **Attribute type:** Text
- **Attribute size:** 200
- **Field caption:** Processor type
- **Form control type:** Input
- **Form control:** Drop-down list
- **Editing control settings -> Data source:** select *Options* and enter the following items into the text area, one per line:
 Athlon;Athlon
 Pentium XEON;Pentium XEON
 Pentium Core 2 Duo;Pentium Core 2 Duo

- **Column name:** ComputerRamSize
- **Attribute type:** Integer number
- **Field caption:** RAM (MB)

- **Form control type:** Input
- **Form control:** Text box

- **Column name:** ComputerHddSize
- **Attribute type:** Integer number
- **Field caption:** HDD (GB)
- **Form control type:** Input
- **Form control:** Text box

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

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.

ComputerID*
ComputerName*
ComputerProcessorType*
ComputerRamSize*
ComputerHddSize*
ComputerImage

Save field

The changes were saved.

Database

Column name: ComputerImage
Attribute type: File
Attribute size:
Allow empty value:

Display attribute in the editing form

Field appearance

Field caption: Image
Form control type: Uploader
Form control: Upload file
Field description:

Quick links:
[Database](#)
[Field appearance](#)
[Editing control settings](#)
[CSS styles](#)

Next >

Click **Next**.

Now you need to choose the field that will be used as the 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 the 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. Click the **Add document types** button and add the **Page (menu item)** document type, which means the editors will be able to create computer documents only under pages, not under article or news documents in the content tree.

Step 5

Parent types

Please select document types under which this document template can be placed.

<input type="checkbox"/>	Document type name
<input type="checkbox"/>	Page (menu item) (CMS.Menuitem)

[Remove selected](#) [Add document types](#)

[Next >](#)

Click **Next**.

In step 6, you need to choose which websites will use this document type. Click the **Add sites** button and choose **My website** in the following dialog.

Step 6

Sites
 Please select sites where this document type can be used:

<input type="checkbox"/>	Site name
<input type="checkbox"/>	My website

Remove selected
Add sites

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:

Content field:

Image field:

Date field:

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"/>
ComputerRamSize	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
ComputerHddSize	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input 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.

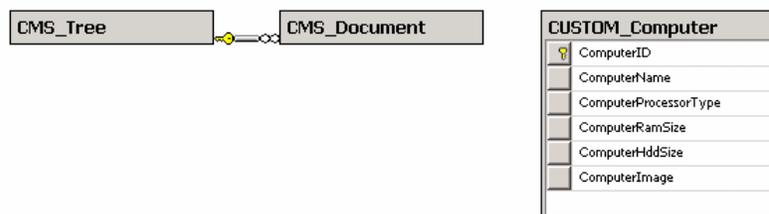
Finish

You have learned how to define a new document type.



How the content is stored

As you already know, the new *Computer* document type has its own database table. Each document is stored in three tables: **CMS_TREE** (tree structure), **CMS_Document** (document properties and metadata) and the specific table of the given document type, 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 list and in detail view mode.

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

Document type properties

> Document types > Computer

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

New transformation New hierarchical transformation

Actions	Transformation name	Transformation type
	AtomItem	ASCX
	Default	ASCX
	Preview	ASCX
	RSSItem	ASCX

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>
```

```
</table>
```

Click  **Save**. As you can see the transformation code is similar to standard ItemTemplate code that you may already know from ASP.NET Repeater and DataList controls. It combines HTML code with ASP.NET commands and data binding expressions. You can also use 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 a transformation for viewing computers in list mode. Go back to the transformation list and edit the **Preview** transformation. Clear the default code and enter the following code:

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

Click  **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.

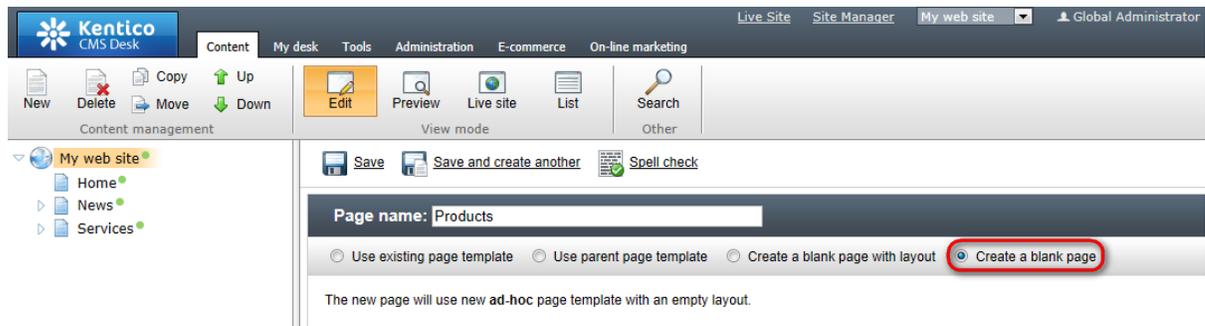
Similarly, you can create an image tag with a parameter that ensures automatic resizing of the longest side to 120 pixels on the server side:

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

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

8.11.4 Page template

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



Click the **Down** button in the toolbar until you move the new page to 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**.

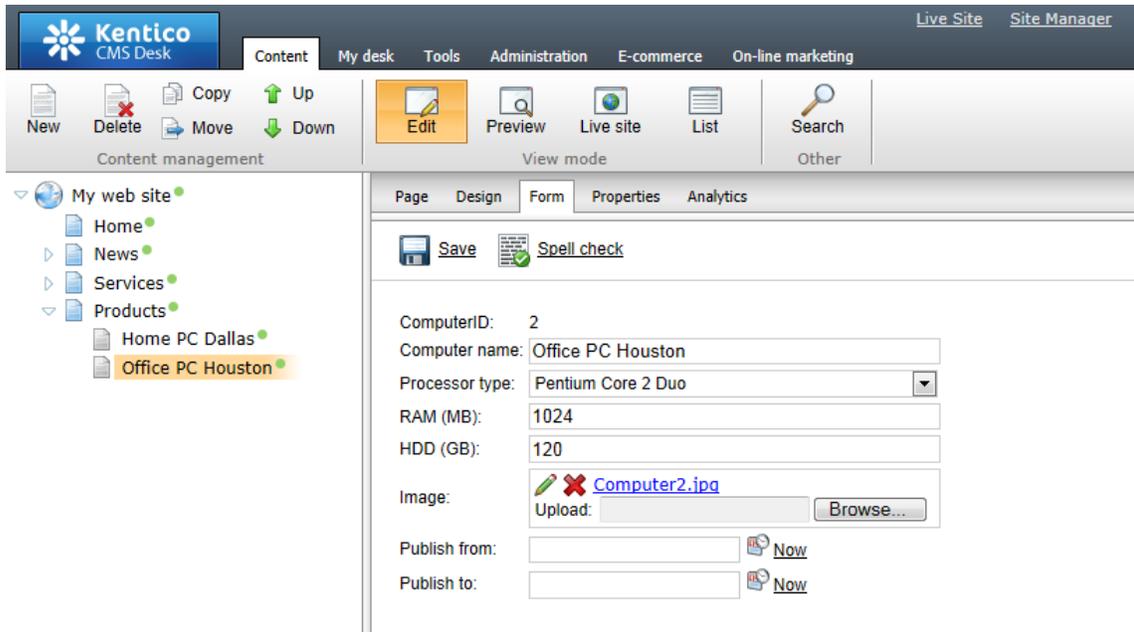
Now we need to enter some computer details. Click **/Products** in the content tree and click **New**. Choose to create a new **Computer**. Enter the following values:

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

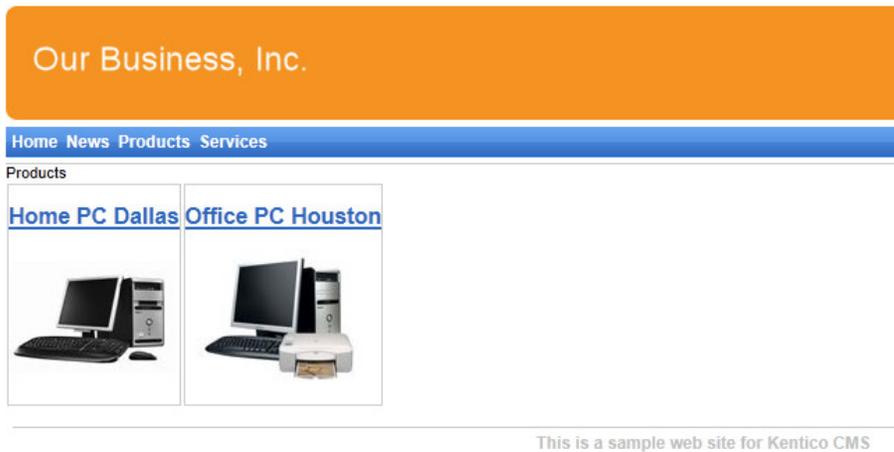
Click **Save and create another** with the following values:

- **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 the *<Kentico CMS installation>\CodeSamples\SampleWebTemplate\Computer_Images* folder)
- **Publish from/to** - leave the values blank

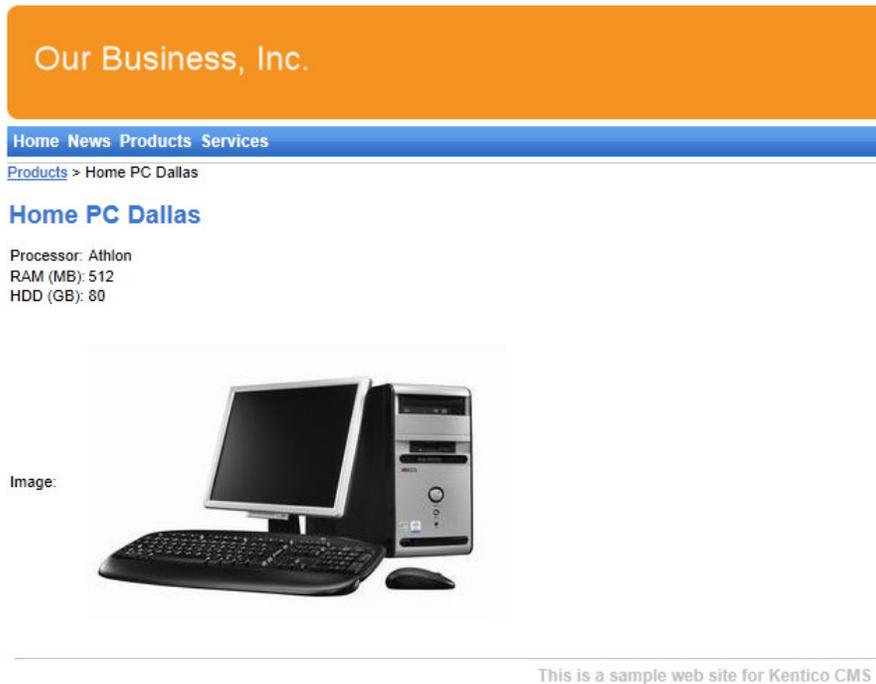
Click  **Save**.



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



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



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

8.12 Search page

Kentico CMS allows you to perform index-based searching through all documents in the 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 website
- **Code name:** MyWebSite
- **Index type:** Documents
- **Analyzer type:** Standard
- **Stop words:** Default
- **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 website**. Switch to the **Cultures** tab and choose the default culture of your site (typically English - United States).

Finally, switch to the **General** tab and **Rebuild** () the index. Once the index gets rebuilt, the documents on the site are prepared to be searched.

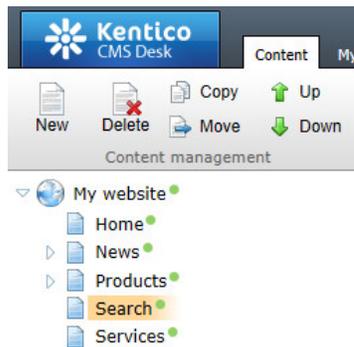
Searching through uploaded text files



It is possible to configure the CMS to search the text inside uploaded files, such as PDF, DOC or XLS documents. The configuration is described in **Developer's Guide -> Installation and deployment -> Additional configuration tasks -> Configuration of full-text search in files**. It's not necessary to configure it at this moment since we will only use the document content search.

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

Click the **Down** arrow in the main toolbar until you move the **Search** page to 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** into the **Search for** box and click **Search**.

Our Business, Inc.

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

Search results

Search for:

Search mode: Any word ▼



Home PC Dallas

http://127.0.0.1/KenticoCMS/Products/Home-PC-Dallas.aspx 12/16/2009 9:15:07 AM



Office PC Houston

http://127.0.0.1/KenticoCMS/Products/Office-PC-Houston.aspx 12/16/2009 9:15:32 AM

This is a sample web site for Kentico CMS

When you click a search result, you will be 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 by editing the **Site Manager -> Development -> Document types -> Root -> Transformations -> SmartSearchResults** (or **SmartSearchResultsWithImages**) 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**, select the root and click **New**. Choose to create a new **Page (menu item)**. Enter *Partners* as the page name and choose to use the **Templates with editable regions -> Simple text** page template. Click  **Save**. Click the **Down** arrow in the main toolbar until you move the **Partners** page to the end of the list.

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

Click  **Save**.

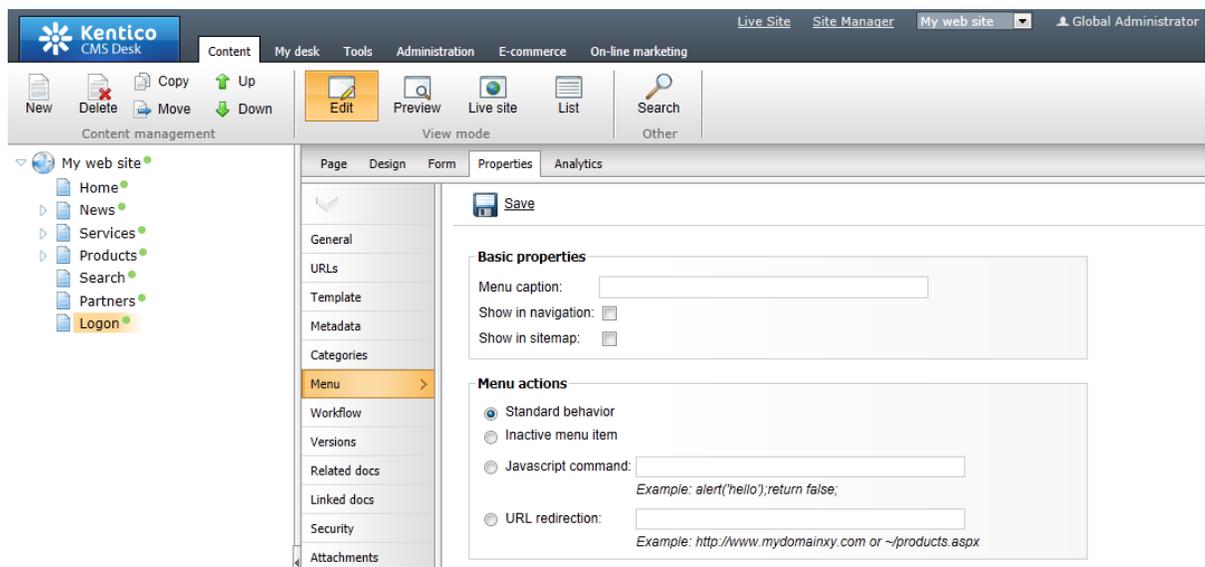
Click **Properties -> Security**. In the **Access** section of the dialog, select **Yes** in the **Requires**

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 *Logon* as the page name and choose the **Membership and security -> Log-on page with registration form** page template. Click **Save**.

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



You will also need to configure the system so that it uses the new logon page. Go to **Site Manager -> Settings**, choose **My website** in the drop-down list, click **Security & Membership**, clear the **Inherit from global settings** checkbox for the **Website logon page URL** field and set its value to `~/logon.aspx`, which is the relative URL of the logon page (from the web application root). Click **Save**.

The screenshot shows the Kentico CMS 6.0 Administration interface. The top navigation bar includes 'Sites', 'Administration', 'Settings', 'Development', 'Tools', 'Dashboard', 'Licenses', and 'Support'. The user is logged in as 'Global Administrator' (v6.0.4279). The current page is 'Security & Membership' under the 'Settings' tab. The left sidebar shows a tree view of settings categories, with 'Security & Membership' selected. The main content area is divided into several sections:

- General:**
 - Administrator's e-mail: Inherit from global settings
 - Send membership reminder (days): Inherit from global settings
- Registrations:**
 - Reserved user names: Inherit from global settings
 - Registration requires e-mail confirmation: Inherit from global settings
 - Registration requires administrator's approval: Inherit from global settings
 - Delete non-activated user after (days): Inherit from global settings
 - Require unique user e-mails: Inherit from global settings
- On-line users:**
 - Update on-line users (minutes): Inherit from global settings
- Content:**
 - Check page permissions: Inherit from global settings
 - Website logon page URL: Inherit from global settings
 - Access denied page URL: Inherit from global settings
- Administration:**
 - Enable UI personalization: Inherit from global settings

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

Configuring the Sign out button

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

Membership -> Current user

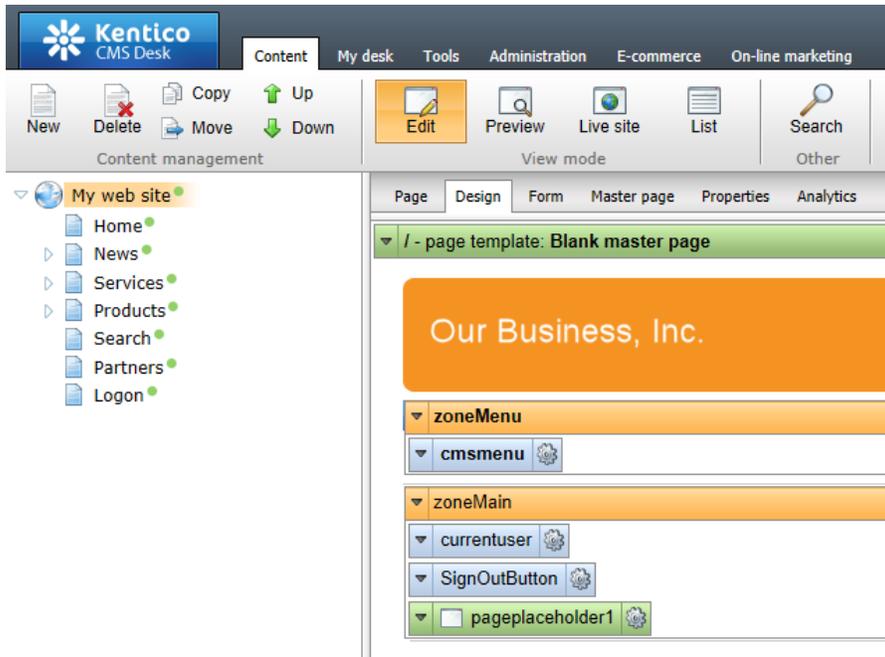
- **Content before:** `<div style="text-align:right">`

Membership -> Sign out button

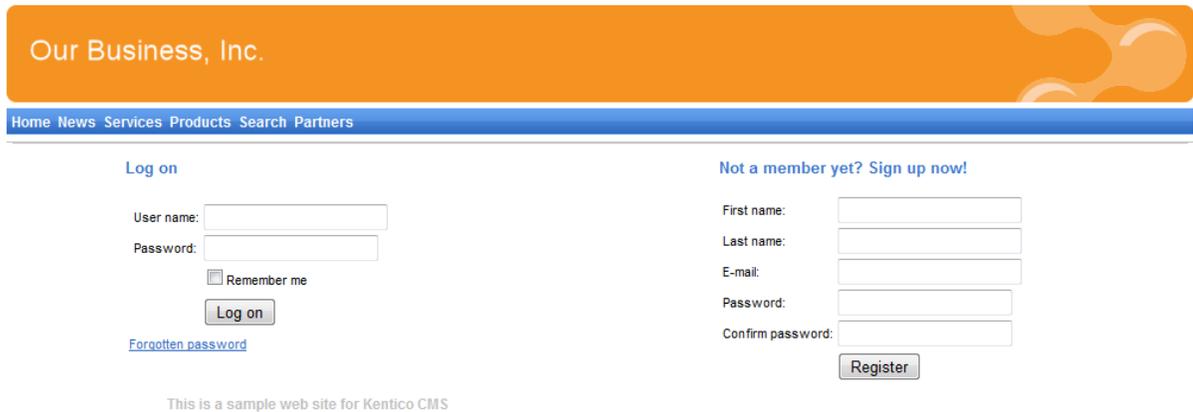
- **Content after:** `</div>`

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



Sign out of CMS Desk. Click **Partners** in the main menu. You will be redirected to the logon page:



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



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



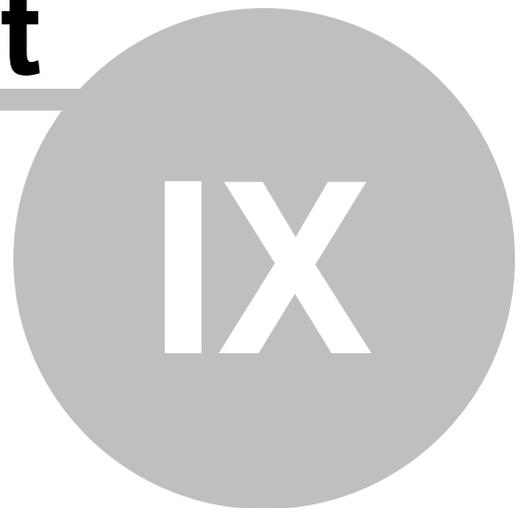
Displaying personalized content based on a user's permissions

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

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

You have just finished the sample website.

Part



Further steps

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 also covers other advanced topics, such as:

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

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

Index

- A -

App themes 87

- C -

container 109
CSS styles 85
custom code
 adding 67

- D -

Document type 117

- F -

Further steps 136

- H -

home page
 creating 107
 editing content 21
 modifying the structure 43
 widget customization 47

- I -

image
 inserting 27
installation
 setup 8
 web application 8

- K -

Kentico CMS Overview 5

- L -

link 28
logon page
 configuring 131

- M -

Main menu 105
master page
 editing 100
menu design
 overview 88
minimum requirements 7

- N -

new page 23
new site
 CSS stylesheet 98
 using the portal engine 91
 wizard 91
news item 31
News page 111

- O -

Overview 5

- P -

Page placeholder 106
page template
 creating 53
 modifying 42
 overview 40
 re-using 56
prerequisites 7
Products page
 new document type 117
 overview 117
 page template 125
 transformations 123

- S -

Search page 128
Secured section 130
Services page 114
Setup 12
Sign out button
 configuring 132
Site Development Overview 35
Support 5, 136
system requirements 7

- T -

Technical support 5, 136
text container 109
Transformations 123

- U -

User interface overview 19

- V -

visual inheritance
 configuring 61
 overview 58

- W -

web installer 8
web part
 cloning 79
 container 65
 creating 73
 development overview 70
 inheritance 79
 layout 79
 modifications 79
web part container 109
web part zone
 adding 103
widgets
 creating 80
 customizing pages 47