

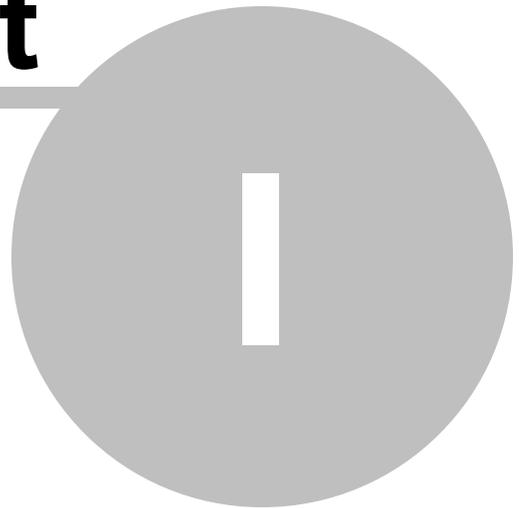
Kentico CMS 7.0 Tutorial

Table of Contents

Introduction	5
Kentico CMS Overview.....	5
Installation	7
Prerequisites.....	7
Setup installation.....	8
Web application installation.....	9
Database setup and Corporate Site.....	11
Managing content	17
User interface overview.....	17
Editing home page content.....	20
Creating a new page.....	21
Inserting an image.....	25
Creating a link.....	26
Creating a news item.....	28
Site development overview	32
Site development overview.....	32
Creating pages using the portal engine	37
Page templates.....	37
Modifying page templates.....	39
Modifying the home page structure.....	40
Customizing pages using widgets.....	45
Creating new page templates.....	51
Re-using page templates.....	54
Visual inheritance.....	56
Configuring visual inheritance.....	58
Web part containers.....	63
Adding custom code to your website.....	65
Developing web parts and widgets	68
Web part development overview.....	68
Creating a simple web part.....	71
Web part modifications.....	77

Creating widgets.....	78
Managing styles and design	84
CSS styles.....	84
App themes.....	86
Menu design.....	87
Creating a new site using the portal engine	91
Overview	91
Creating a new web site using the wizard.....	91
Creating a CSS stylesheet.....	99
Master page.....	102
Main menu.....	106
Page placeholder.....	108
Home page.....	108
Container for home page text.....	111
News page.....	113
Services page.....	116
Products page.....	120
Overview	120
New document type	120
Transformations	127
Page template	129
Search page.....	132
Secured section for partners.....	135
Further steps	141
Further steps.....	141

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:

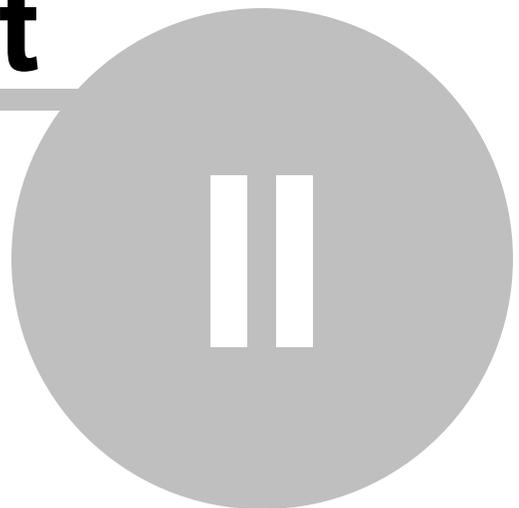
- Developer's Guide
- Controls Reference
- Web parts or Widgets Reference
- API Reference
- Database Reference



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

The following configurations are supported by Kentico CMS. Other configurations may work too, but the system was not tested on them.

Server-side Requirements

- Windows Vista Home Premium/Business/Enterprise/Ultimate, Windows 7 (both 32bit and 64bit) or Windows 8 (both 32bit and 64bit), or Windows Server 2008, 2008 R2, 2012.
- Microsoft .NET Framework [3.5 SP1](#) or [4.0](#) or higher.
- Microsoft Internet Information Services (see the table below) or Visual Studio/Visual Web Developer 2008/2010/2012 built-in web server.
- Microsoft SQL Server 2005, 2008, 2008 R2, 2012 (including free SQL Server Express Edition [2005/2008/2012](#)).

Internet Information Services overview

Internet Information Services version	Operating system	Details and installation instructions
IIS 5.1	Windows XP Professional	not supported
IIS 6.0	Windows Server 2003	not supported
IIS 7.0	Windows Vista	IIS 7 Installation and Deployment
	Windows Server 2008	
IIS 7.5	Windows 7	
	Windows Server 2008 R2	
IIS 8.0	Windows 8	Installing IIS on Windows Server 2012
	Windows Server 2012	

Hosting Requirements

- ASP.NET 3.5 SP1 (or higher) and Microsoft SQL Server 2005/2008/2012 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 250 MB or more memory and 100+ MB database.

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

Windows Azure

Kentico CMS fully supports the Microsoft Windows Azure platform, including SQL Azure, Azure Storage and other services. Windows Azure SDK 1.7 is required.

Development Tools

If you want to create custom web parts or integrate custom code, you need **Visual Studio 2008/2010/2012** or free **Visual Web Developer 2008/2010/2012 Express Edition**.

Supported Client Browsers for Content Editors

- Internet Explorer 8, 9
- Firefox 4.0+
- Chrome 12
- Safari 4.0+ or Firefox 4.0+ on Mac OS

Supported Client Browsers for Site Visitors

- Internet Explorer 6.0+
- Firefox 1.0.5+
- Chrome 12+
- Mozilla 1.7.1+
- Netscape 7.1+
- Opera 7.52+
- Safari or Firefox on Mac OS
- Mobile browsers, such as Safari on iPhone, are supported as well, but some features may be limited by browser capabilities.

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

Required experience

Although Kentico CMS allows you to create dynamic websites without programming, you may want to create custom web parts or add custom code when developing more complex websites. Developers should be able to create a simple ASP.NET application using Visual Studio and have some experience with relational databases and SQL, so that the flexibility of Kentico CMS can be fully put to use.

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:



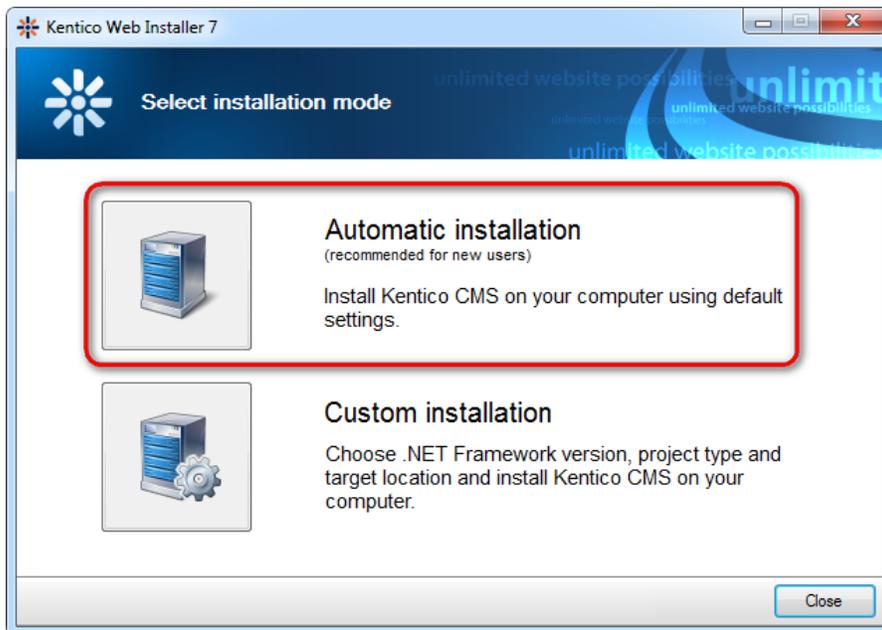
Read and accept the license agreement and click **Next**. Choose the installation location of the binary files and documentation on your disk. After the setup completes the installation, check **Launch Kentico Web Installer** and click **Finish**.

2.3 Web application installation

The Kentico Web installer opens. If it has not started automatically, you can always run it from **Start menu -> All Programs -> Kentico CMS -> Kentico Web Installer**.

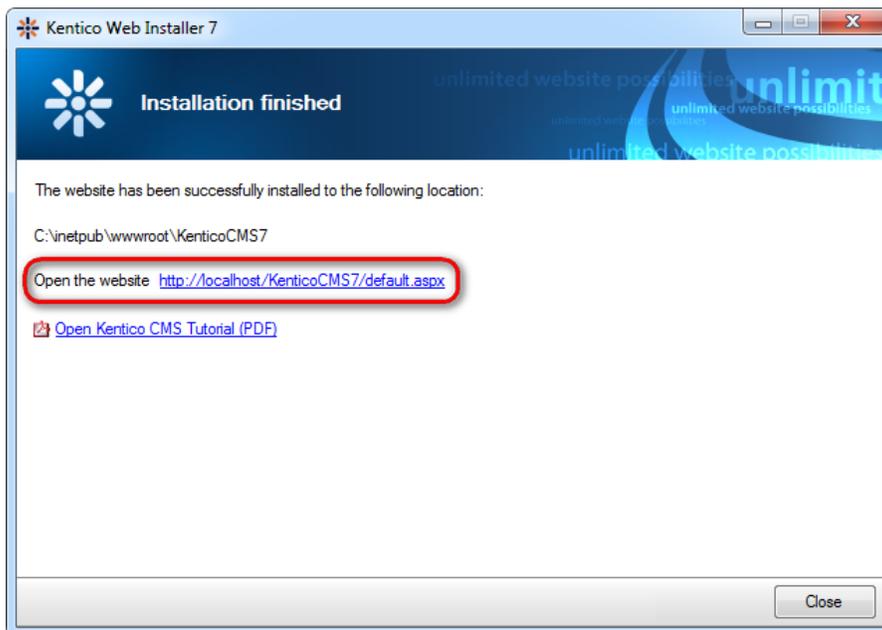
Selecting installation mode

1. In the installer, select the **Automatic installation** option.



Note that **Automatic installation** chooses .NET framework based on the framework that is available in your system, first checking for .NET 4.0, .NET 4.5 and finally 3.5. *Web site project* and default IIS *inetpub\wwwroot* installation folder are selected. If you want to make these selections yourself, refer to [Developer's guide -> Installation and deployment -> Installation procedure -> Web installer](#) for more information on the **Custom installation** option.

2. Once the installer copies the necessary files, click the **Open the website** link.



3. Continue to the [Database setup](#) topic.

2.4 Database setup and Corporate Site

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

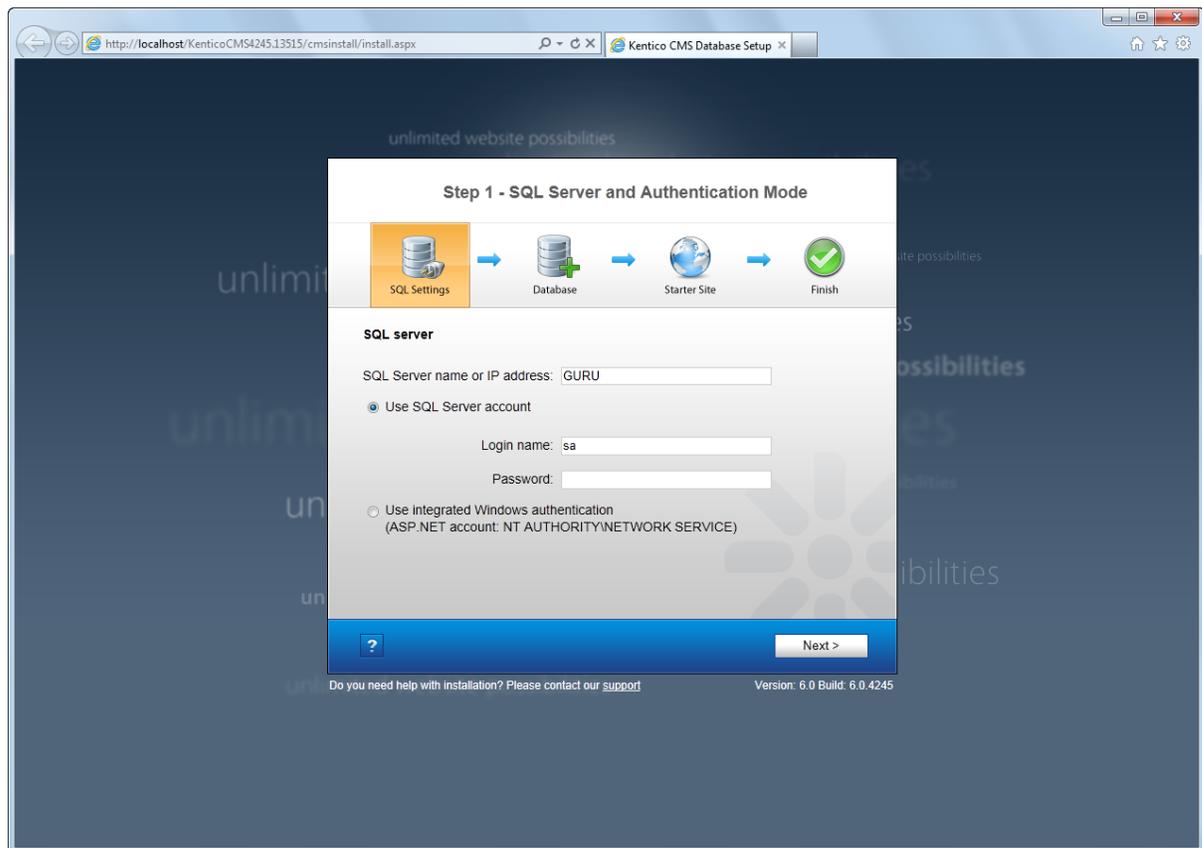
In the first step, choose the SQL Server name or IP address. If you are using SQL Server 2005 Express Edition, the default server name is `.\SQLExpress` or `(local)\SQLExpress`.

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



SQL Settings → Database → Starter Site → Finish

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



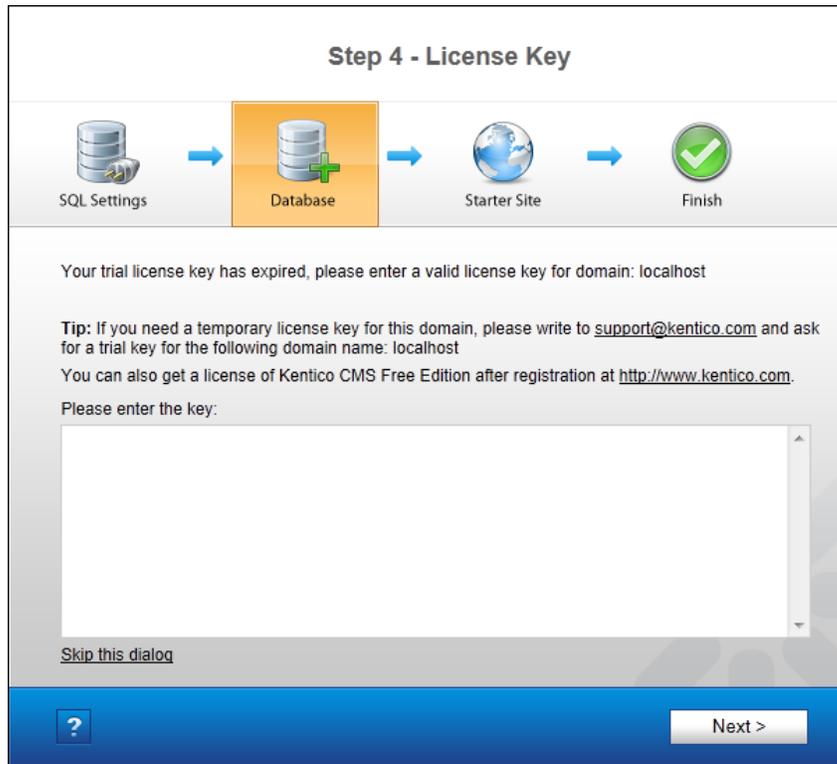
SQL Settings → Database → Starter Site → Finish

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



The progress bar shows four steps: SQL Settings, Database, Starter Site, and Finish. The Starter Site step is currently selected and highlighted in orange.

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 on **Continue to the new web site**.

Step 7 - Finished

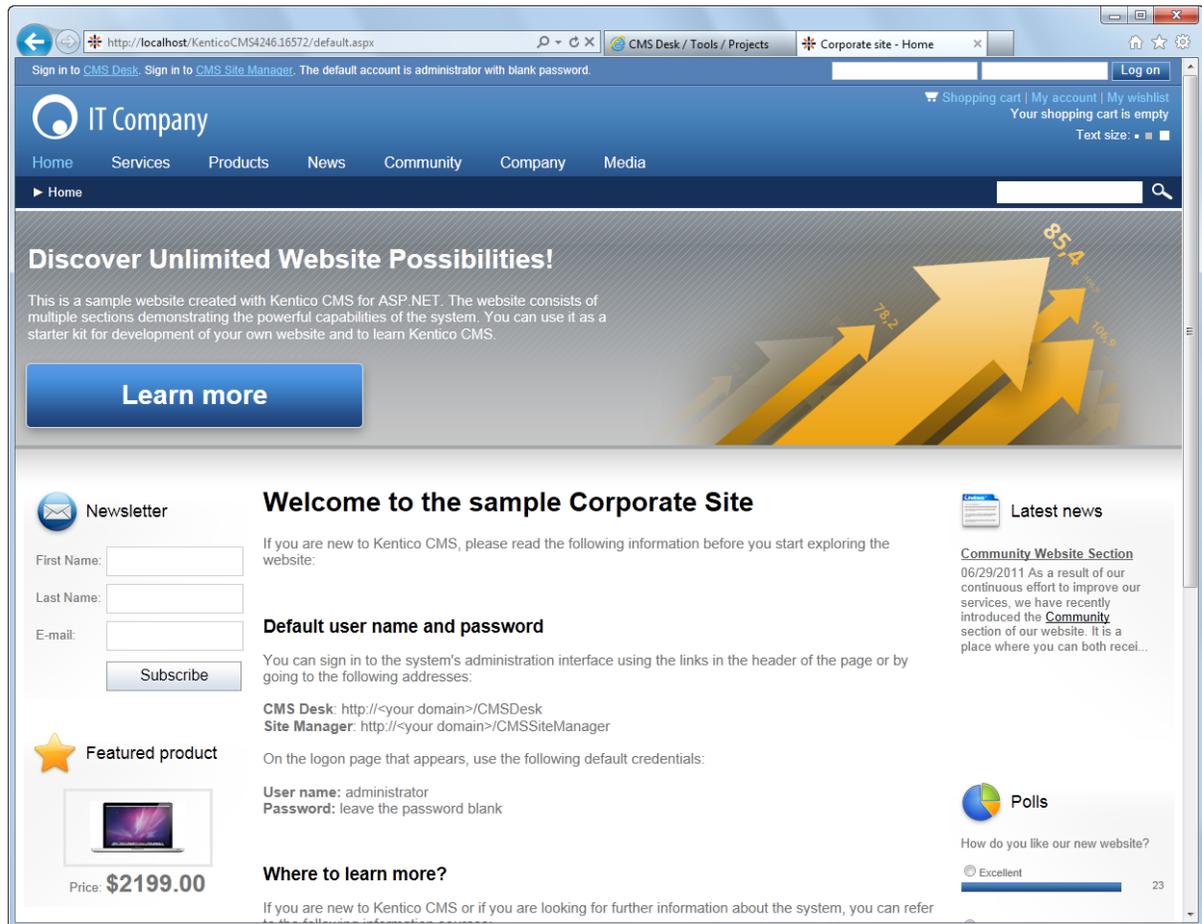


The progress bar shows four steps: SQL Settings, Database, Starter Site, and Finish. The Finish step is currently selected and highlighted in orange.

The site has been created successfully.

[Continue to the new website](#)

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



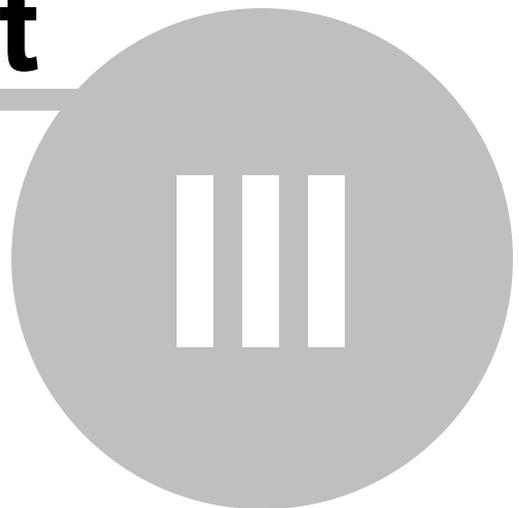
The screenshot shows a web browser window displaying the homepage of a sample Corporate Site. The browser's address bar shows the URL `http://localhost/KenticoCMS4246.16572/default.aspx`. The page header includes the site logo "IT Company" and navigation links for Home, Services, Products, News, Community, Company, and Media. A shopping cart icon indicates an empty cart. 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: a "Newsletter" sign-up form with fields for First Name, Last Name, and E-mail; a "Welcome to the sample Corporate Site" section with default user name and password information; a "Featured product" section showing a laptop for \$2199.00; and a "Where to learn more?" section. On the right side, there is a "Latest news" section with a "Community Website Section" article and a "Polls" section asking "How do you like our new website?" with an "Excellent" option selected.



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

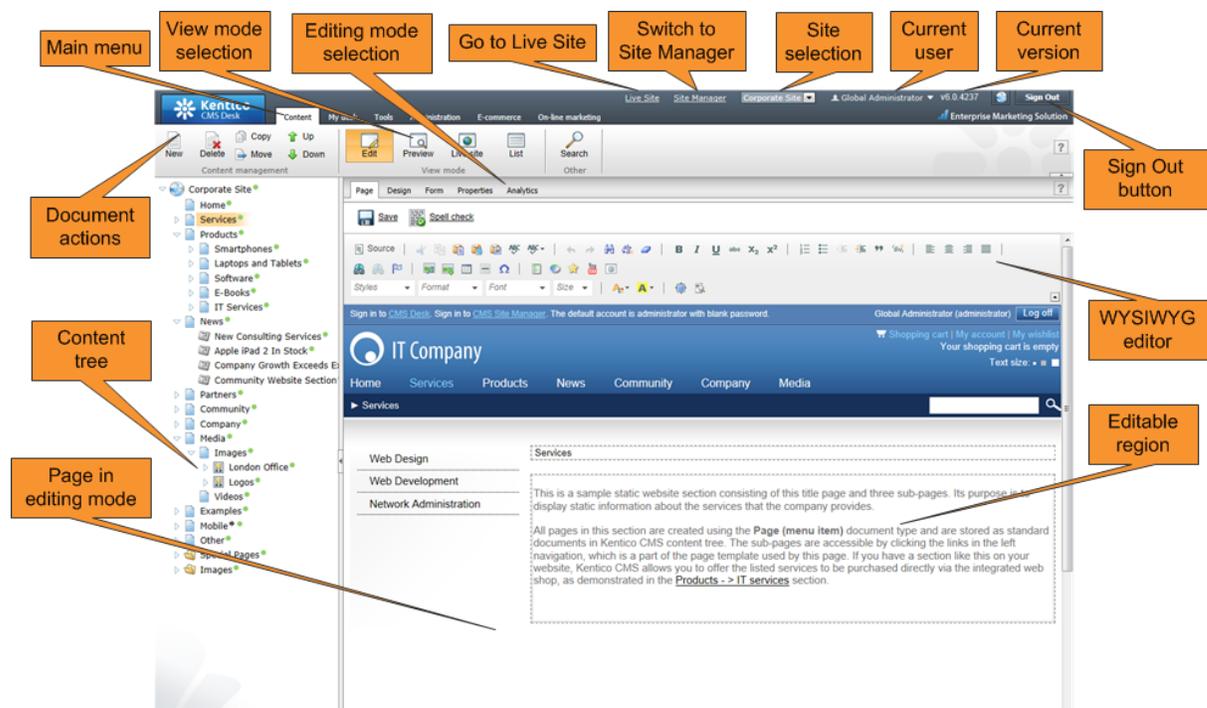


Default user name and password

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

It is highly recommended to change the password before publishing the website on a live server.

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



The CMS Desk user interface consists of the following main sections and features:

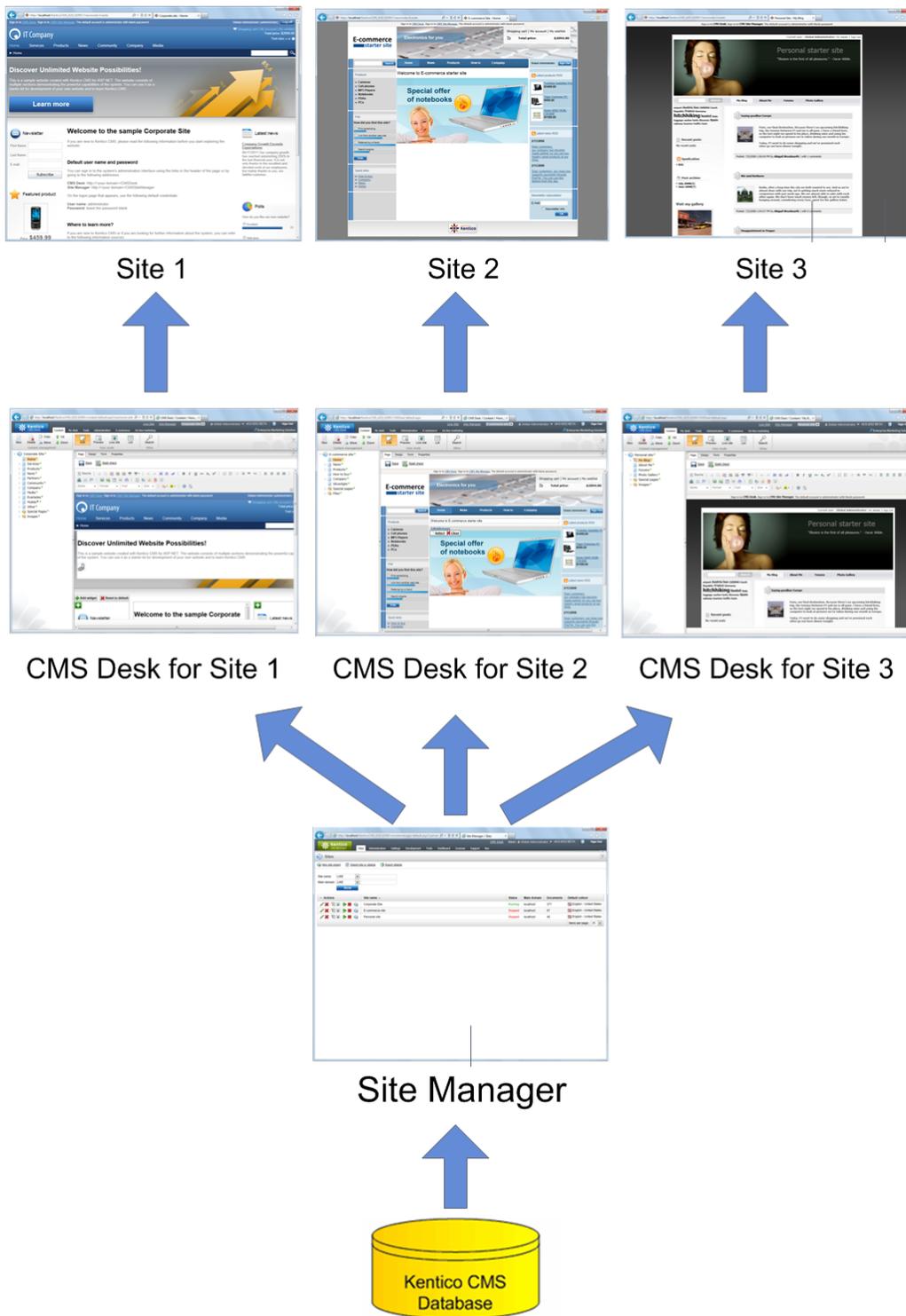
- **Main menu** with Content, My Desk, Tools, Administration, E-commerce and On-line marketing tabs.
- **Content tree** that represents the site map of the website. It allows you to organize the site's structure or select a specific document for editing in the content section.
- **Document actions** 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 CMS Desk. 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 administration interface. This option is only available for 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 list.
- **Current user** - displays the name of the current user.
- **Current version** - version number of the Kentico CMS installation.
- **Sign Out button** - clicking this button logs you out of 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, change its formatting and insert graphics or other items. It is available for **Editable regions** on the *Page* tab, as well as when editing document fields on the *Form* tab.
- **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 manage the content of a specific website. Developers and site administrators who need to manage global settings, code and components that affect all websites can also use the **Site Manager** interface.

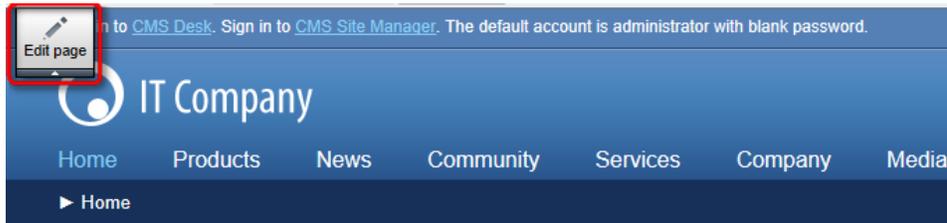
The Site Manager interface is accessible either through the **<http://<domain>/<virtualdirectory>/cmssitemanager>** URL, by clicking **Site Manager** at the top of the CMS Desk user interface or **Sign in to Site Manager** at the top of a live site. The following figure shows how the database, Site Manager, CMS Desk and websites are related:



On-site editing

In addition to the two main administration interfaces, Kentico CMS also provides a way to edit page content directly while browsing on the live website. Authorized editors can access on-site editing mode by going to the <http://<domain>/<virtualdirectory>/cmsedit> URL, or by clicking the **Edit page** button

displayed in the corner of pages.



All actions available in on-site editing mode can also be done through the CMS Desk interface. It simply provides an alternative way to edit websites.

This tutorial demonstrates all operations in CMS Desk — it is recommended to first become familiar with CMS Desk and the general website content structure before using on-site editing. If you are interested in learning more about on-site editing, please see the [Content management -> On-site editing](#) chapter of the Kentico CMS Developer's Guide.

3.2 Editing home page content

Now we will modify the home page content. Click the **Home** document in the content tree. You will see a page as shown below.

The page is now displayed in editing mode and it contains two editable regions. Editable regions on the **Page** tab can be identified by a dotted outline. Delete all content from one of the regions and enter the following text instead:

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 on 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 select **Preview** mode now, it will display the same content as the **Live site** mode. It works as a preview for documents that are affected by workflow or are scheduled to be published in the future. In these cases, 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**

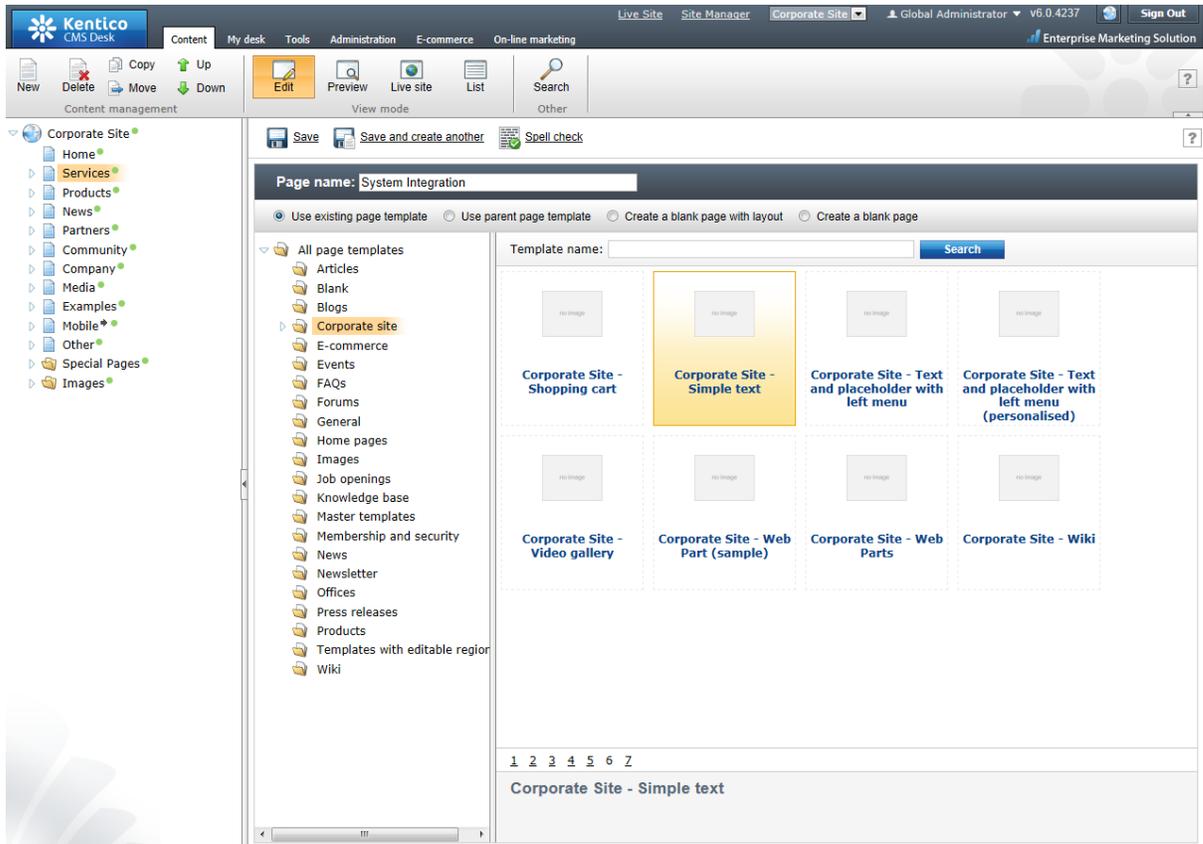
Please select new document type:

-  [Page \(menu item\)](#)
-  [Wireframe](#)

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

 [Link an existing document](#)

Click the **Page (menu item)** option. 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.

The screenshot displays the Kentico CMS 7.0 interface. On the left, the content tree shows the following structure:

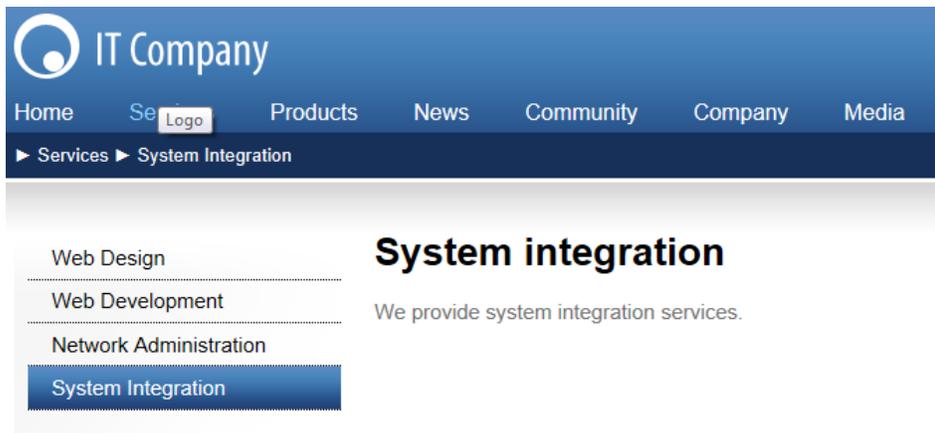
- Corporate Site
 - Home
 - Services
 - System Integration
 - Web Design
 - Web Development
 - Network Administration
 - Products
 - News
 - Partners
 - Community
 - Company
 - Media
 - Examples
 - Mobile
 - Other
 - Special Pages
 - Images

The main editor area shows a preview of the 'System Integration' page. The page has a blue header with the 'IT Company' logo and navigation links: Home, Services, Products, News, Community, Company, Media. The main content area contains a 'System Integration' section with a list of services: Web Design, Web Development, and Network Administration. Below the list is a text box containing the text: 'We provide system integration services.'

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 will be moved to the bottom of the section.

- Services
 - Web Design
 - Web Development
 - Network Administration
 - System Integration

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.

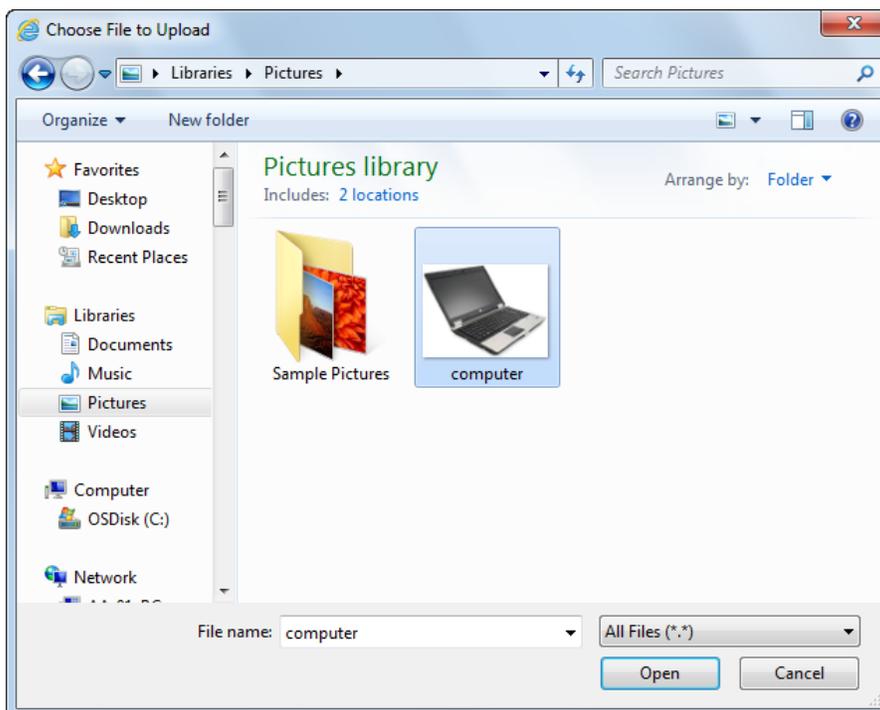


You have learned how to create a new page based on a pre-defined page template.

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 media** (📎) 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 7.0 interface. The top navigation bar includes 'Live Site' and 'Site Manager' links. The main toolbar contains 'Save', 'Spell check', 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.



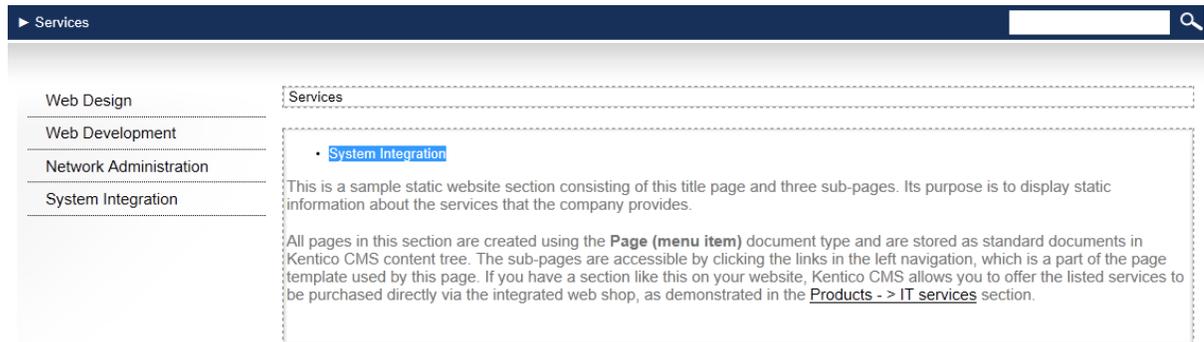
Allowing pop-ups for the website

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.

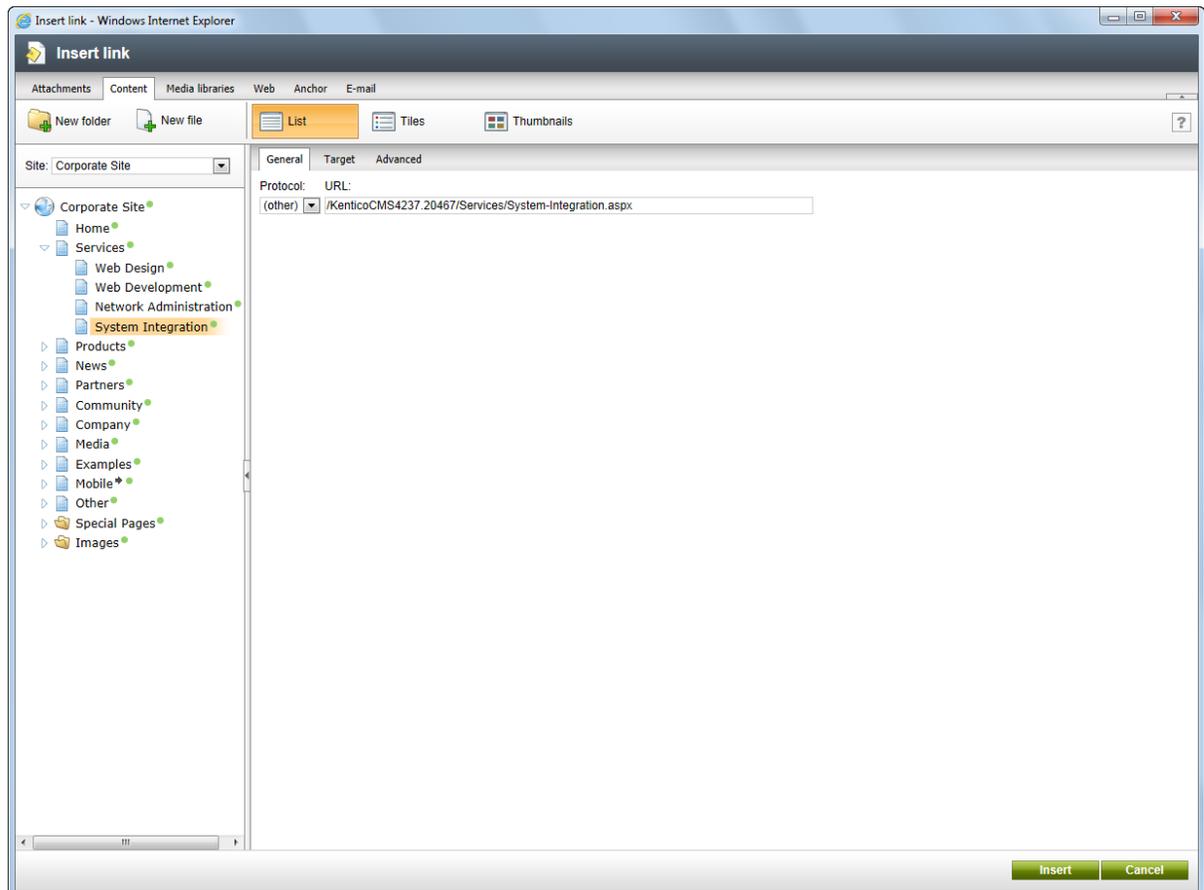
3.5 Creating a link

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 bulleted list item called **System integration** and 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 (underlined).

Click  **Save** and choose the **Live site** mode. Now, when you click on **System integration**, you are redirected to the new page.

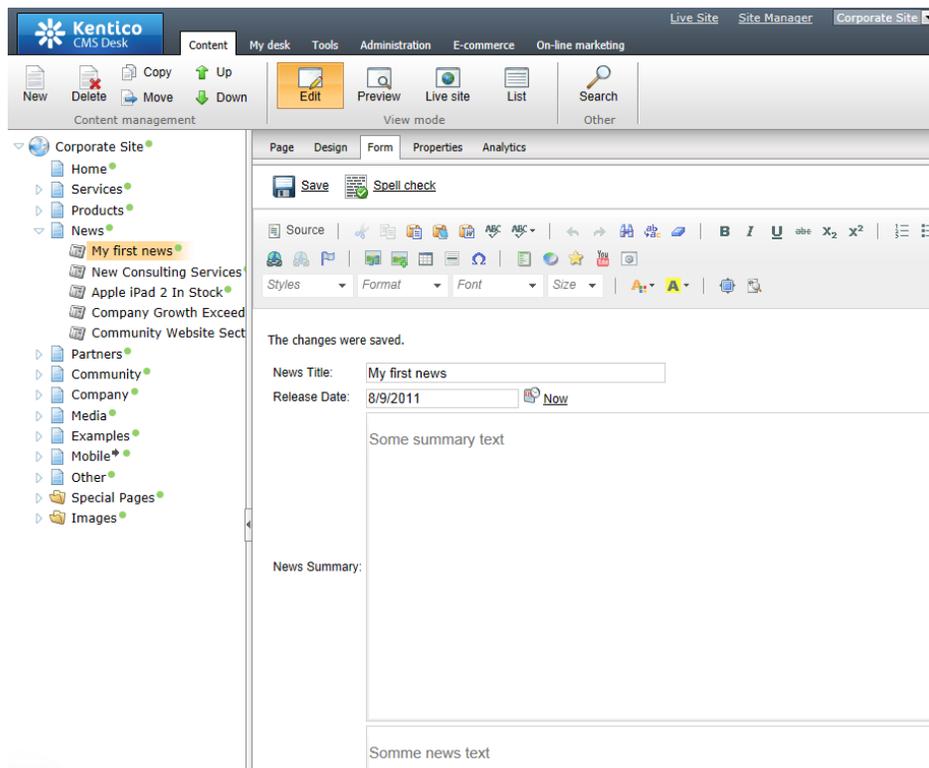
You have learned how to create a link between pages.

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:



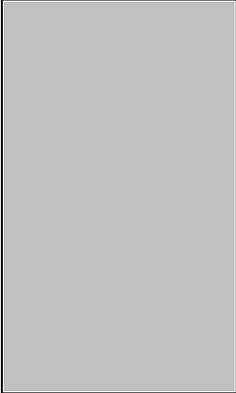
You have learned how to add a news item and how to use the editing form for structured documents.



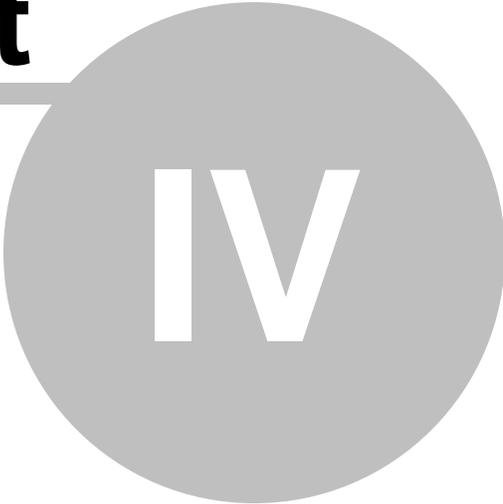
Page versus Form

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

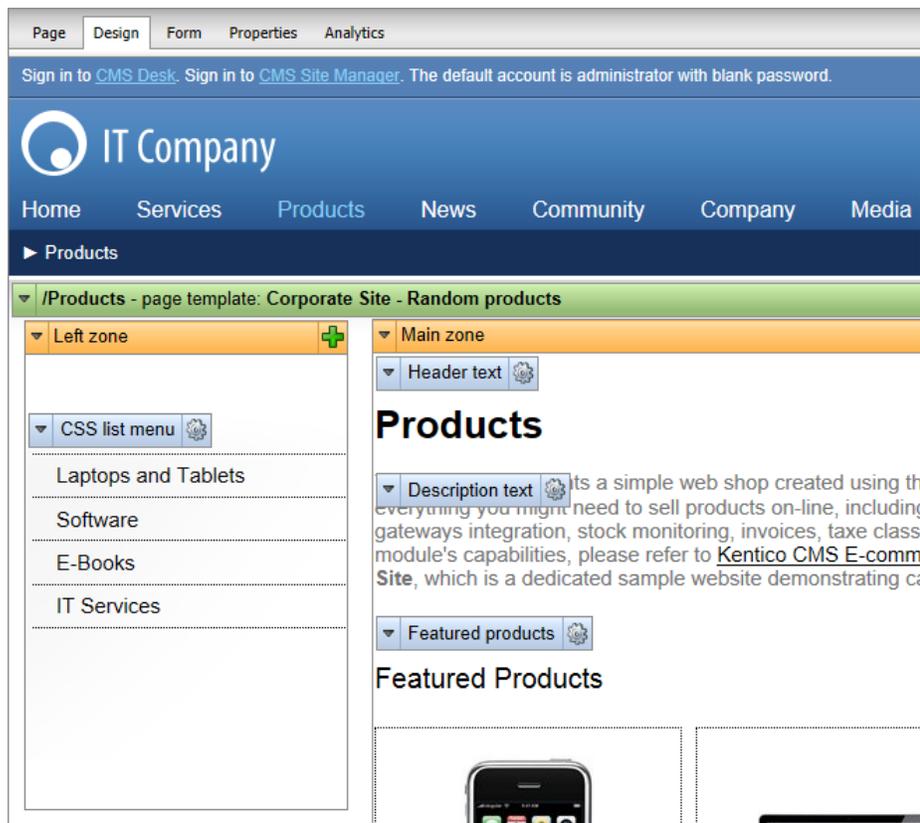
Site development overview

4 Site development overview

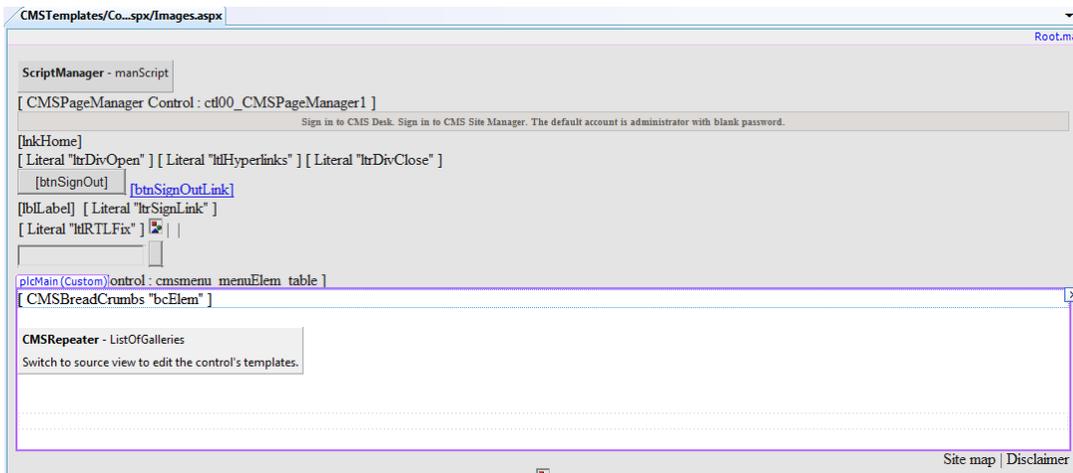
4.1 Site development overview

Kentico CMS provides two basic 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 using the portal engine, but if you are an advanced .NET developer or wish to integrate existing functionality built on standard ASP.NET architecture, you may want to use ASPX templates. It is also possible to create websites or specific pages using the Model-View-Controller architectural pattern (based on the ASP.NET MVC framework), but this is beyond the scope of this basic tutorial. Please see the [MVC development model](#) chapter in the Developer's Guide for more information.

If required, the available development 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 may be defined on ASPX templates where editing can be done through the portal engine.

The following table compares the portal engine and ASPX templates:

	Portal Engine	ASPX Template
How you work	<p>You build websites and design their pages using a browser-based interface.</p> <p>No programming knowledge is required for common tasks.</p>	<p>You build ASPX pages (web forms) 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 and place them onto the ASPX pages. These are standard ASPX pages that are part of the web project, so you can also work with their code behind files.</p> <p>It is also possible to place web parts (which are actually ASCX user controls) on the page templates if the required server control is not available.</p>
Master pages and visual inheritance	<p>Sub-pages inherit content from their parent pages by default (so called "visual inheritance"). The inheritance</p>	<p>Page templates may inherit content from a master page, which works just like a standard ASP.NET master page</p>

	can optionally be broken if you want to create a page without parent content.	(.master file). Pages do not inherit content from their parents in the website content hierarchy, they only inherit from the master page (if it is used).
Custom code integration and extensibility	You can create your own user controls or web parts (ASCX files with a portal engine interface) if you need to integrate custom functionality. Any custom controls or code can be added to the web parts placed on the website. You can also use standard ASPX pages within your portal engine-based website.	You build standard ASPX pages with code behind files, which means you can place any custom controls and code onto the page.
Advantages	<ul style="list-style-type: none"> • Easier and faster to build a website. • ASP.NET programming knowledge is not required for common tasks. • You can build the whole website very quickly, using only a web browser. 	<ul style="list-style-type: none"> • Standard ASP.NET architecture. • You can use your favorite development tools, such as Visual Studio.
Disadvantages	<ul style="list-style-type: none"> • Proprietary architecture and development process. 	<ul style="list-style-type: none"> • Requires ASP.NET programming knowledge. • The design of the web pages cannot be fully managed via the browser-based administration interface.



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

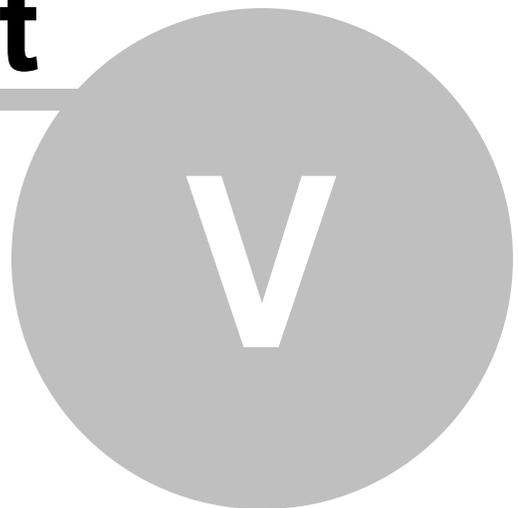
- custom document types with custom fields
- workflow and versioning
- content locking (check-out, check-in)
- multilingual content
- content preview and content staging
- document-level permissions with permission inheritance
- full-text search in all content
- document management features for uploaded 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 ASPX templates, please read the Tutorial for ASPX page templates.

Part



Creating pages using the portal engine

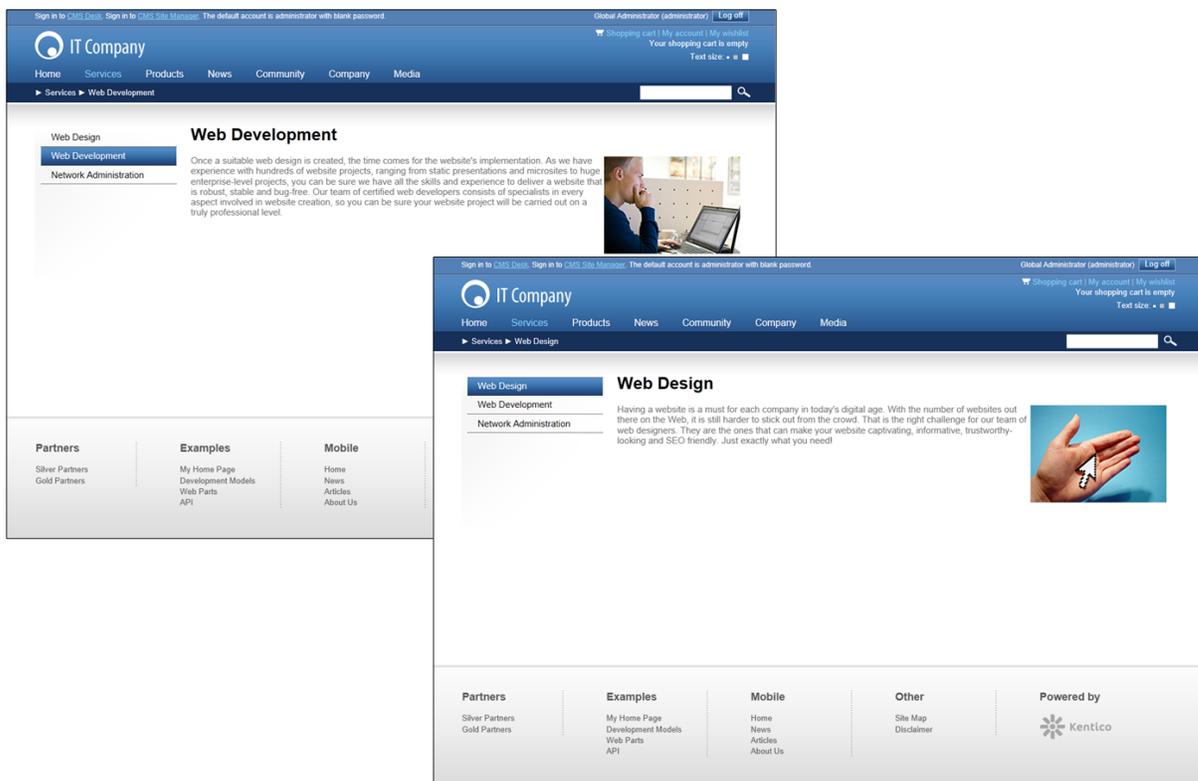
5 Creating pages using the portal engine

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 or custom web parts. This chapter explains how the portal engine works.

What is a page template?

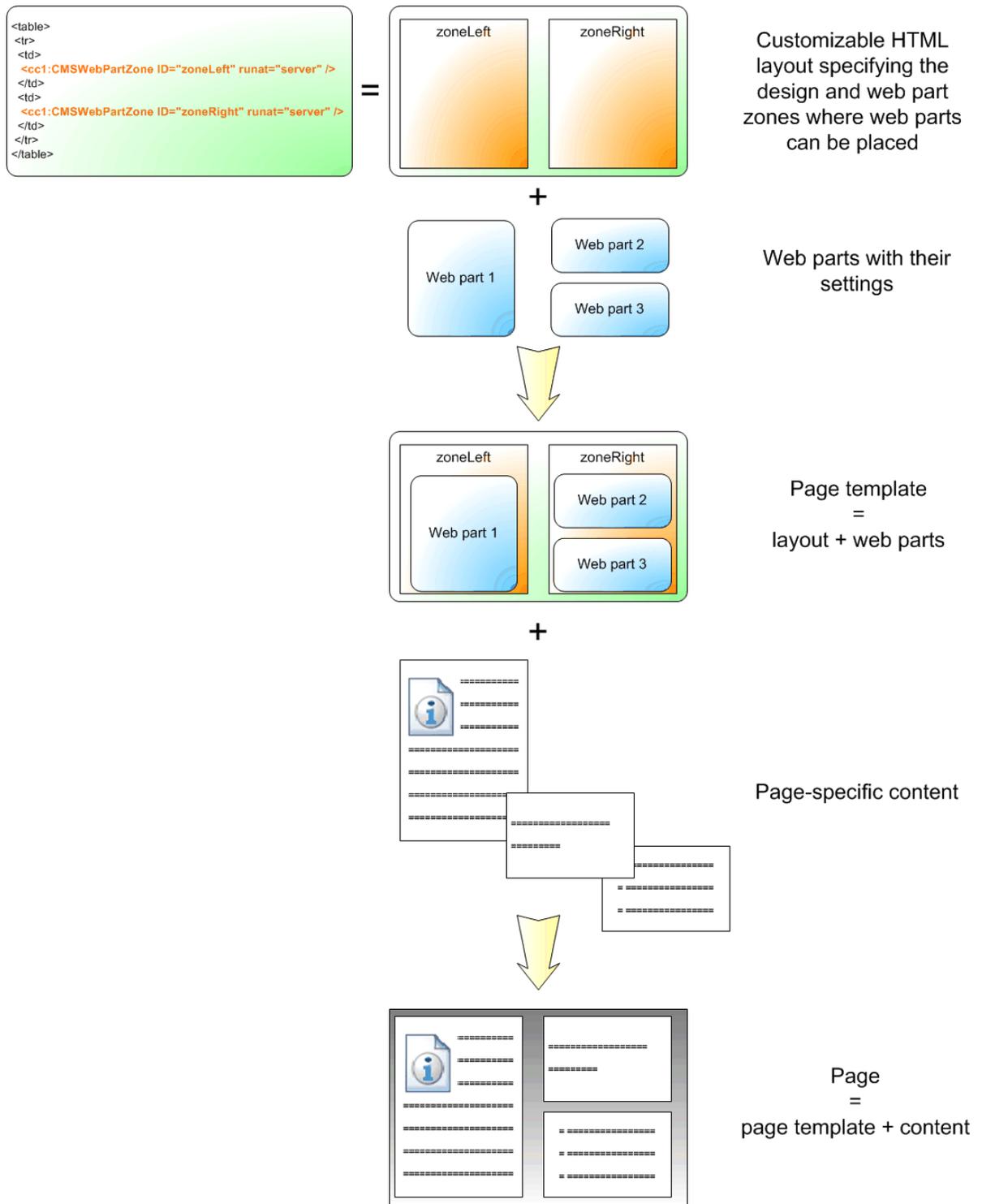
Every web page is based on a page template. The page template can be specific for a single page (so called "ad hoc" page template) or it can be re-used for several pages. The following picture shows an example of two pages that use the same page template:



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 a 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:

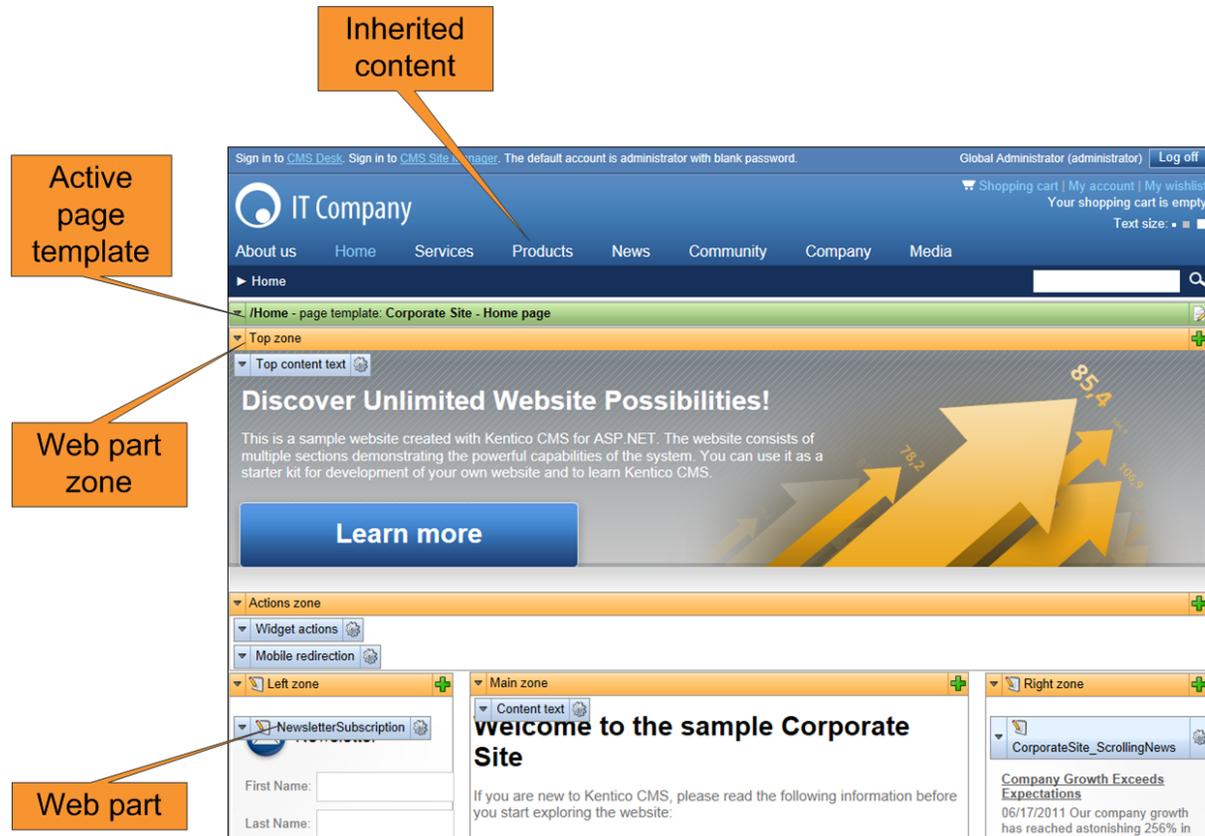


The **page layout** is a piece of fully customizable HTML code that defines the design of the page and contains web part zones. The **web part zones** represent areas where web designers can place web parts. The **web parts** display page content or provide some functionality, such as input forms. The 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 page templates

Now you will learn how to work with the layout of a page template. Select 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 header, contains the orange web part zones and blue web parts.



The page is based on the page template *Corporate Site - Home page*. The **page template** consists of **web parts** that are placed inside **web part zones**. The placement of these web part zones is specified by the page layout (HTML layout) of the template.

Clicking the **Edit layout** (🔧) button in the green header of the active page template allows you to modify the layout of the template.

Page template layout properties

Save Preview

Changes affect all pages that use the current template.

Layout element: Web part zone Layout type: ASCX

```
<%@ Control Language="C#" ClassName="Simple" Inherits="CMS.PortalControls.CMSAbstractLayout" %>
<%@ Register Assembly="CMS.PortalControls" Namespace="CMS.PortalControls" TagPrefix="cms" %>

<div class="topHome"><div class="padding"><cms:CMSWebPartZone runat="server" ZoneID="zT" /></div></div>
<!-- Container -->
<div class="home">
  <div class="inner">
    <cms:CMSWebPartZone runat="server" ZoneID="zA" />
    <div class="left"><div class="padding"><cms:CMSWebPartZone runat="server" ZoneID="zL" /></div></div>
    <div class="right"><div class="padding"><cms:CMSWebPartZone runat="server" ZoneID="zR" /></div></div>
    <div class="responsiveClear"></div>
    <div class="center"><div class="padding"><cms:CMSWebPartZone runat="server" ZoneID="zM" /></div></div>
  </div>
</div>
```

1 / 12

Add CSS styles

Close

The page layout consists of basic HTML (or ASPX) code. It contains **CMSWebPartZone** controls, which define the web part zones on the template. Each of them must have a unique ID. It is possible to format the layout using any type of HTML code — it's up to you whether you use tables or a CSS-based layout.

Working with the layout code

You can use the selector above the code editor to quickly **Insert** web part zones and other advanced layout elements.

By clicking the **Preview** button in the header of the editing dialog, you can edit the layout code side-by-side with a preview of how the changes affect the live site version of the page.



Access control

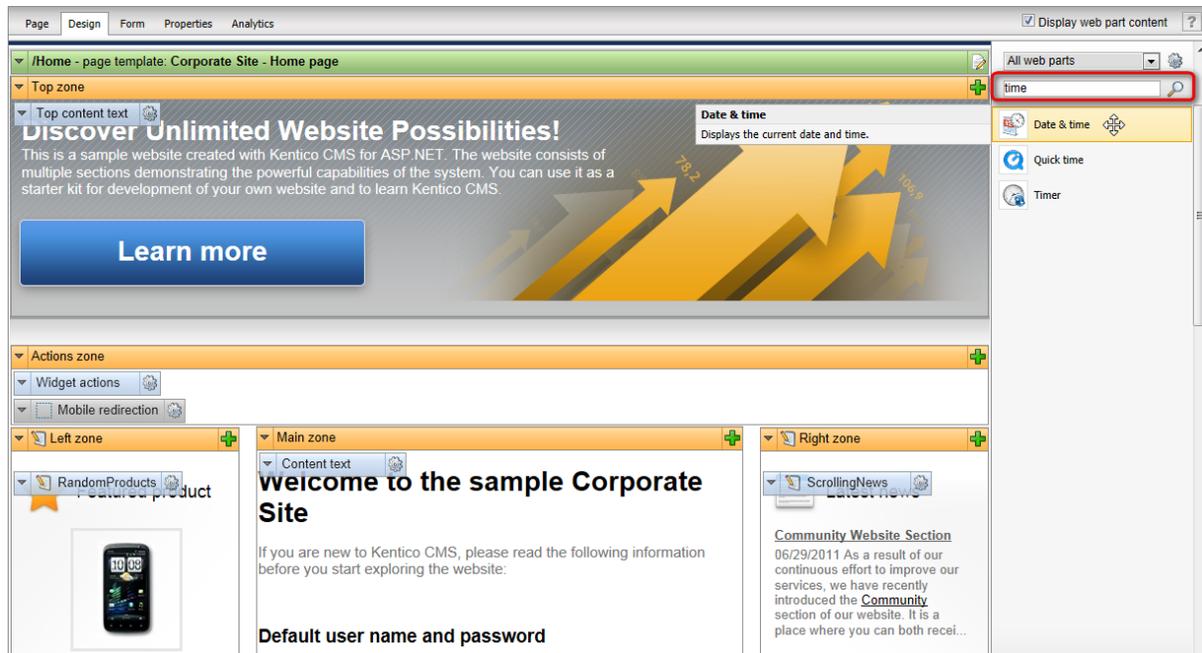
Only users who are marked as **global administrators** or have the *Design web site* permission assigned for the **Design** module may open documents on the **Design** tab and edit page layouts. This functionality cannot be accessed by regular content editors.

5.3 Modifying the home page structure

Now you will learn how to define the content of a page template's web part zones. Specifically, we will add a web part displaying the current time to the template used by the Home page.

Make sure you are in **Edit** mode, click on **Home** in the content tree and select the **Design** tab. You can

easily insert web parts using the toolbar displayed on the right side of the tab (by default). To find the appropriate web part, type the word *time* into the search textbox (🔍) at the top of the toolbar. This will limit the listed web parts to those that have *time* in their name.



Now hover over the **Date & time** web part, hold down the mouse button and drag it from the toolbar to the bottom of the **Actions zone**, below the *Mobile redirection* web part. After you drop the web part into the zone, click the **Configure** (⚙️) button on the right side of this web part to open the **Web part properties** dialog.

Here, scroll down to the **Web part container** property and select *Corporate Site - List box content* as the value.

Web part properties (Date & time)

General | Layout

General

Use server time:

Date and time format*:

Web part container

Web part container: [Edit](#) [New](#)

Container title:

Container CSS class:

Container custom content: [...](#)

Hide container on subpages:

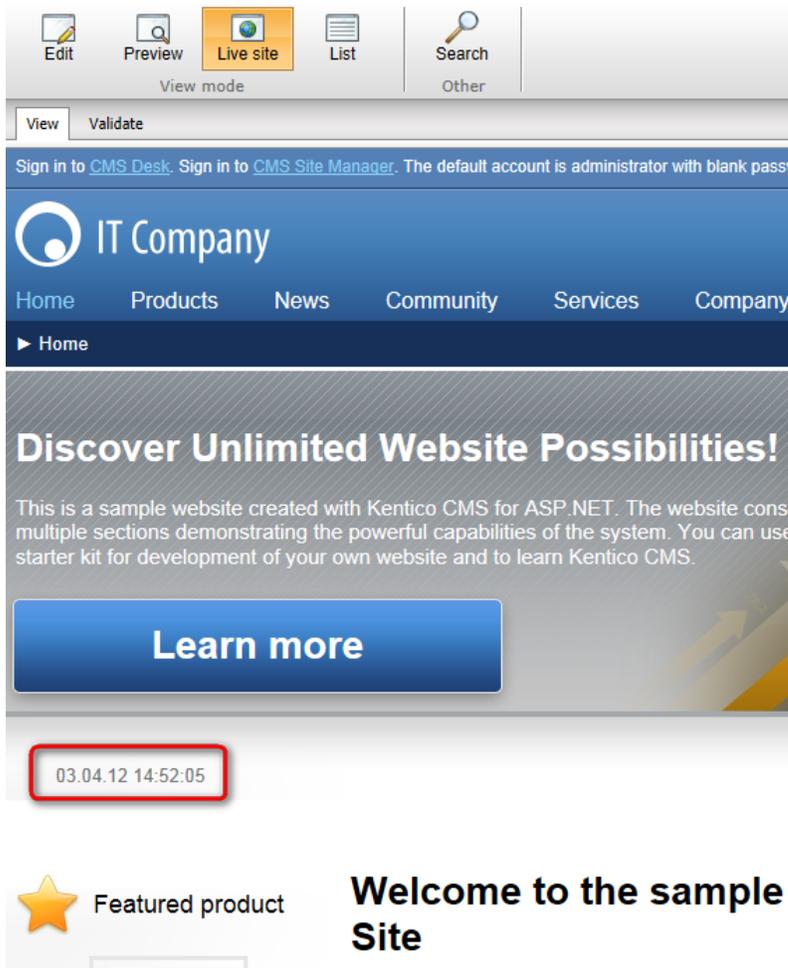
HTML Envelope

Content before: [...](#)

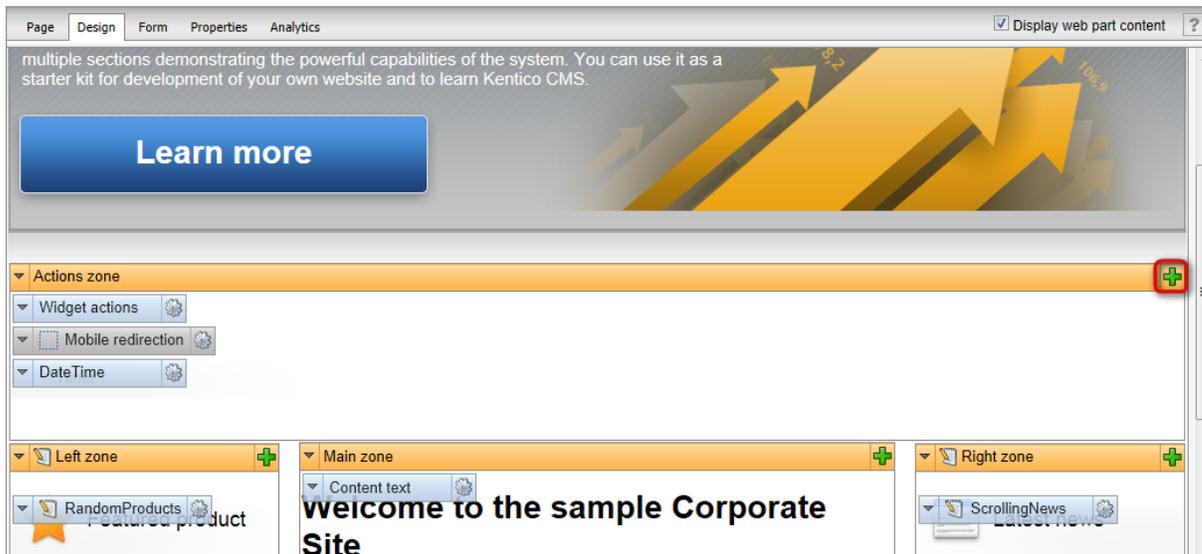
Refresh content

[OK](#) [Cancel](#) [Apply](#)

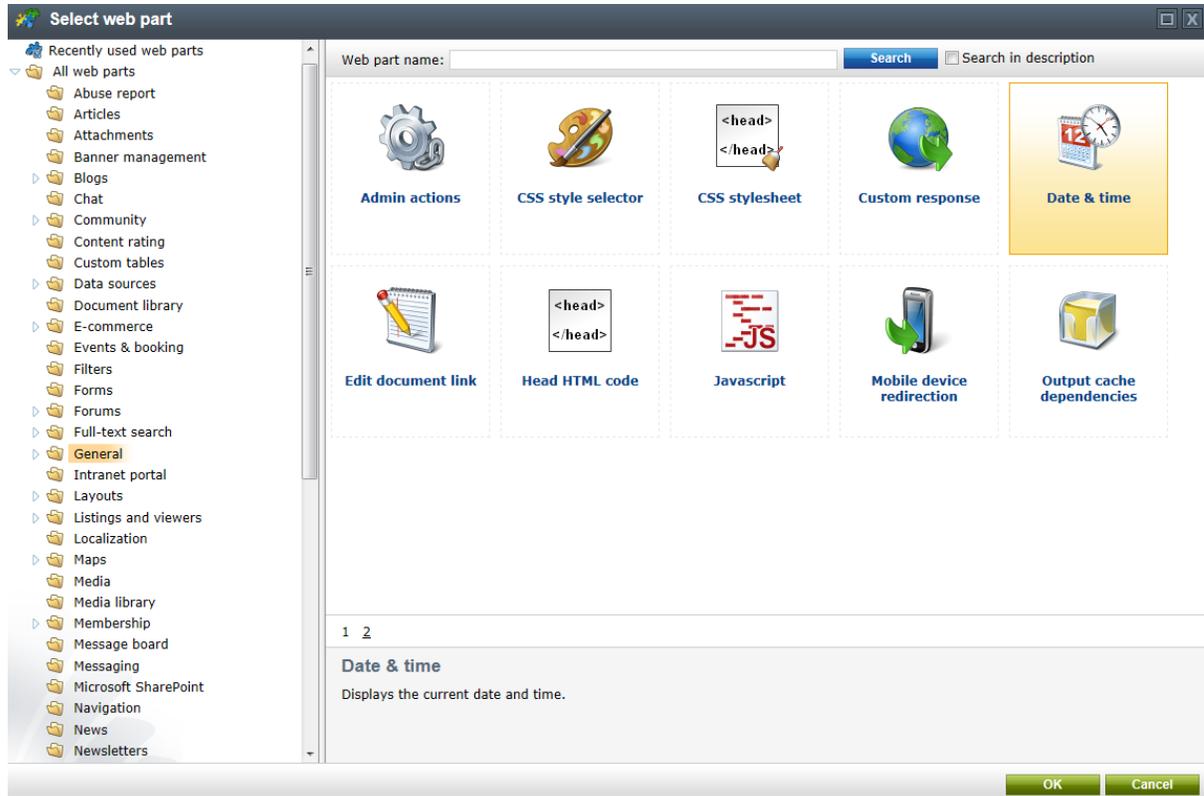
Click **OK** to save the changes and the web part will be added to the zone. Click on **Live site** in the main view mode menu and you will be able to see how the new web part affects the appearance of the Home page.



Return to **Edit** mode and view the **Home** page on the **Design** tab again. There is an alternative way of adding web parts if you do not wish to use the toolbar. You can instead click on the **Add web part (+)** icon in the top right corner of the zone where you want to insert the web part.



This opens the **Select web part** dialog, where you can look through the catalog of all available web parts. For example, the previously used **Date & time** web part can be found in the **General** category.



Because there is no reason to add the web part again, click **Cancel** to close the dialog.

Both ways of inserting web parts achieve the same result. When instructed to add a web part to a page template in the remaining parts of this tutorial, you can choose whichever method you prefer — either the toolbar or the zone action buttons.

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



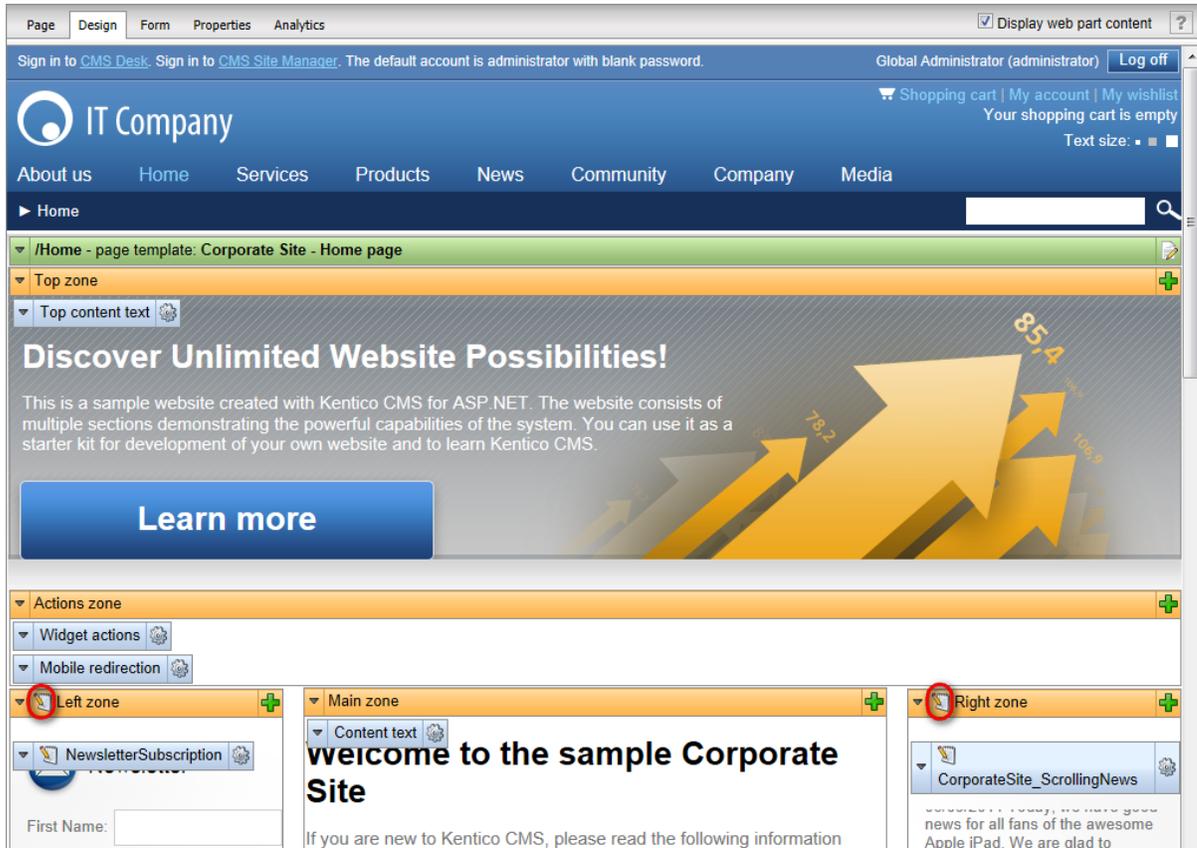
Tip: Testing changes without switching the view mode

You can quickly test the changes you make to the structure of the website by opening or refreshing the given page in a different browser. Please note that the view mode will still be changed if you use two tabs of the same browser.

Tabbed browsing support can however be useful for easy switching between the **CMS Desk** and **Site Manager** interface.

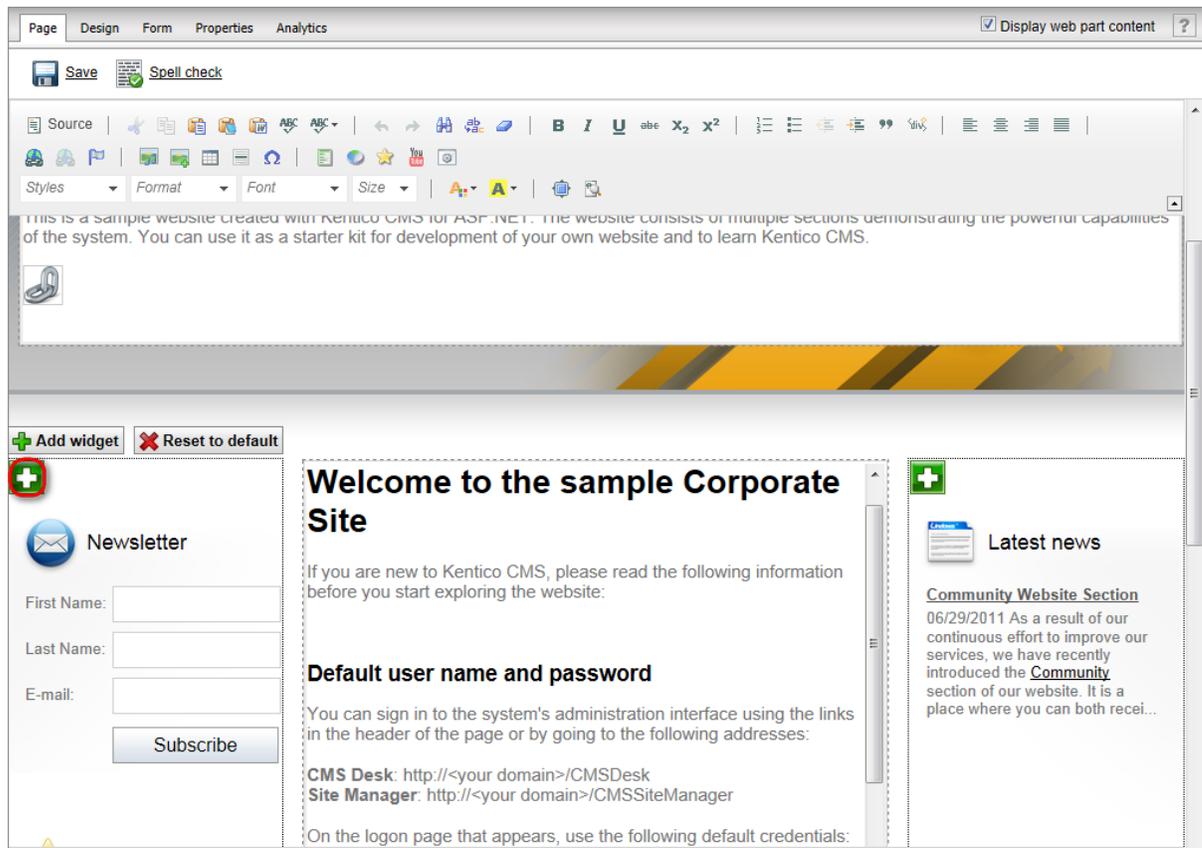
5.4 Customizing pages using widgets

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 (🔧).



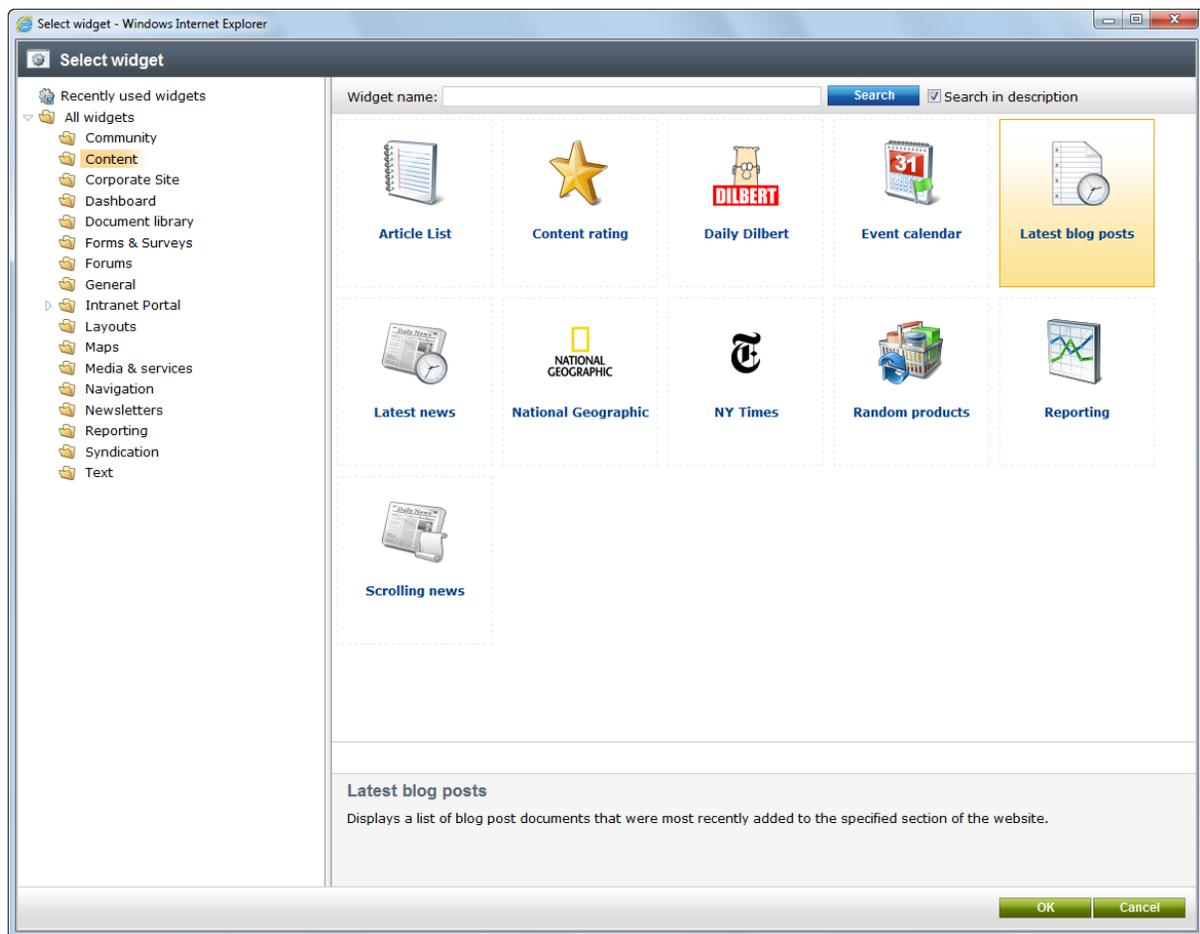
This identifies them as zones that contain *widgets* of a certain type rather than web parts. Widgets are components placed into page template zones just like web parts, but they provide the option of page customization for various kinds of website users, not just administrators or designers. These two 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 7.0 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 striped 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 to the web part selection dialog, but with less items available. 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:

- **Widget container:** Black box
- **Widget 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



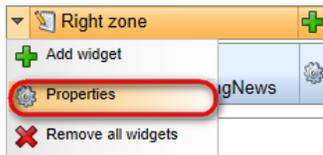
Polls

How do you like our new website?

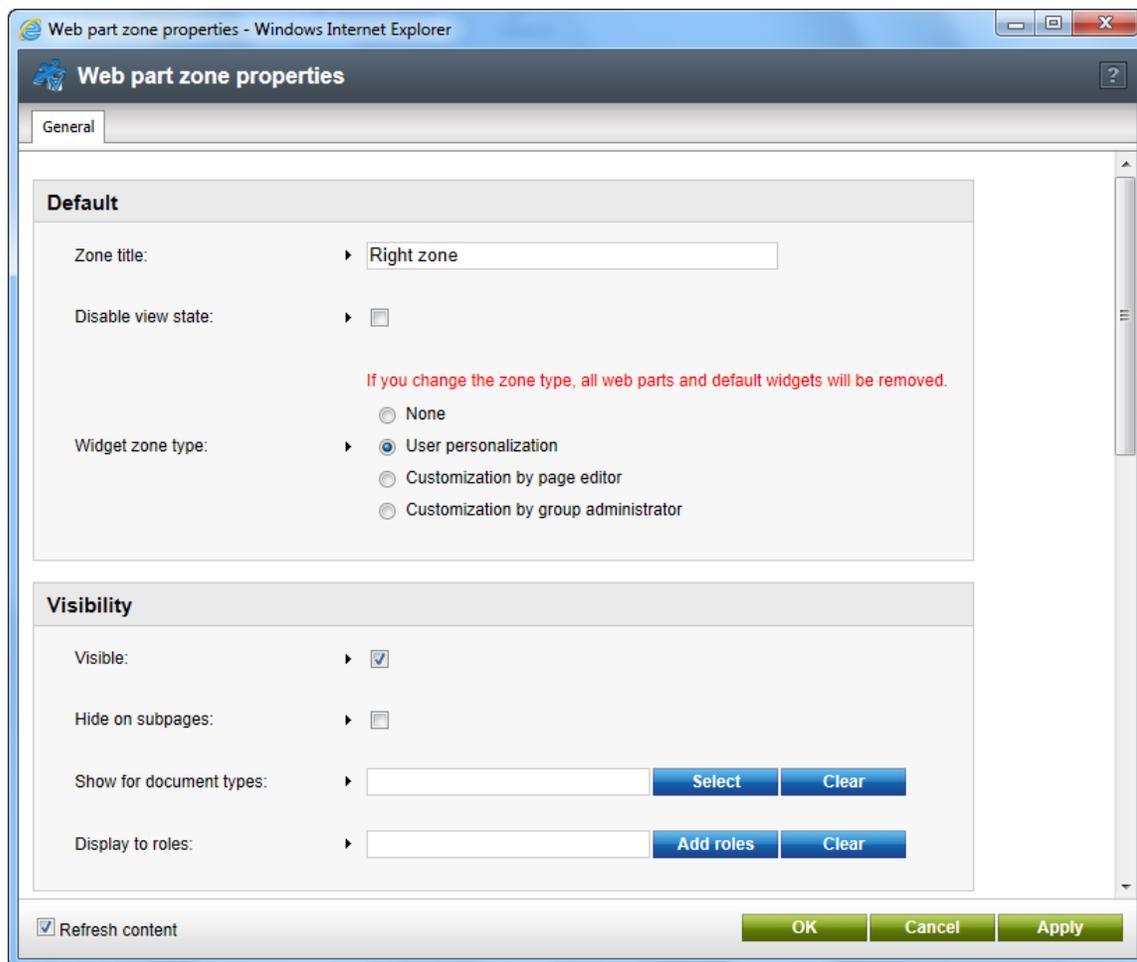


Vote

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:

[Subscribe](#)

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:

[Subscribe](#)

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 from the currently displayed content.

5.5 Creating new page templates

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.

1. Click the root in the content tree and click **New**. Choose to create a new **Page (menu item)**.
2. Type *About us* as the **Page name** and choose to **Create a blank page with layout** using the **Two columns** layout (found on the second page). Make sure that the **Copy this layout to my page template** box is checked — it ensures that your page template uses a modified version of the pre-defined layout without changing other pages.

Save Save and create another Spell check ?

Page name: About us

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

Layout name: Search

 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

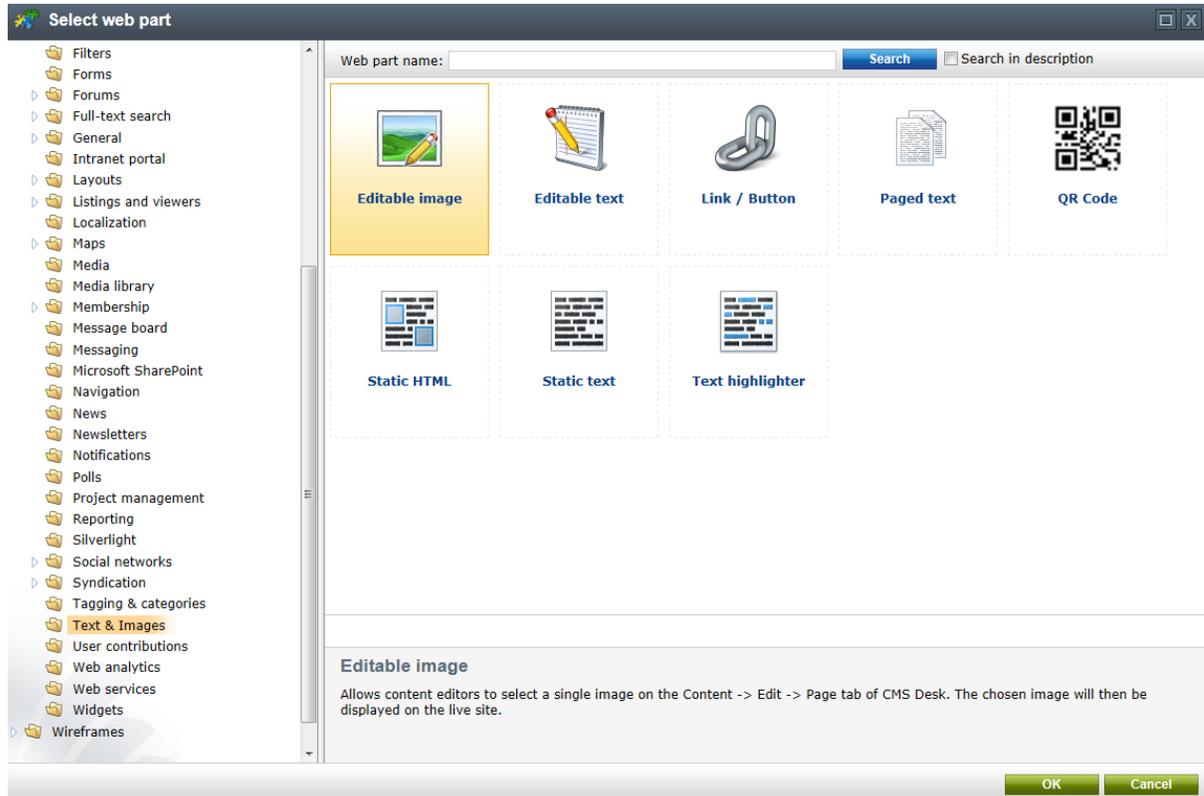
Two columns
Two columns with web part zones.

Copy this layout to my page template

3. Click **Save** to create the new page. 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 text regions on the right

4. Switch to the **Design** tab and click the **Add web part (+)** icon on the right of the **zoneB** web part zone.

5. Choose the **Text & Images -> 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>

Click **OK**.

6. Add another web part to the **zoneC** zone: Choose the **Text & Images -> 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**.

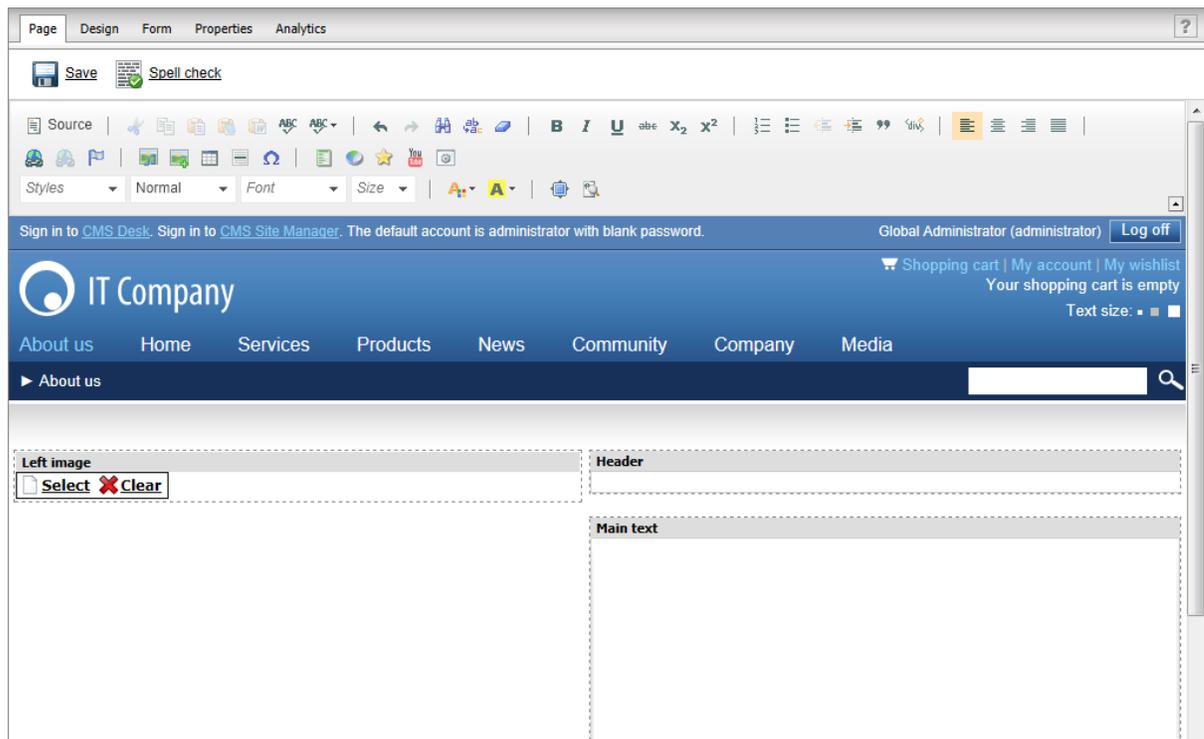
7. Add another **Editable text** web part to the right web part zone and set the following properties:

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

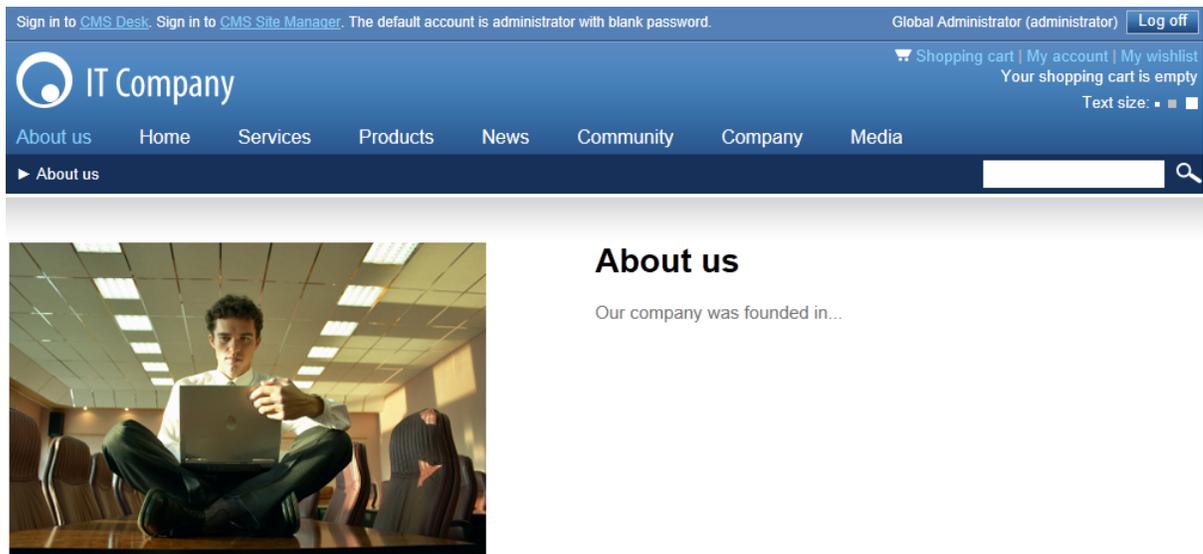
Click **OK**.

Result

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



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



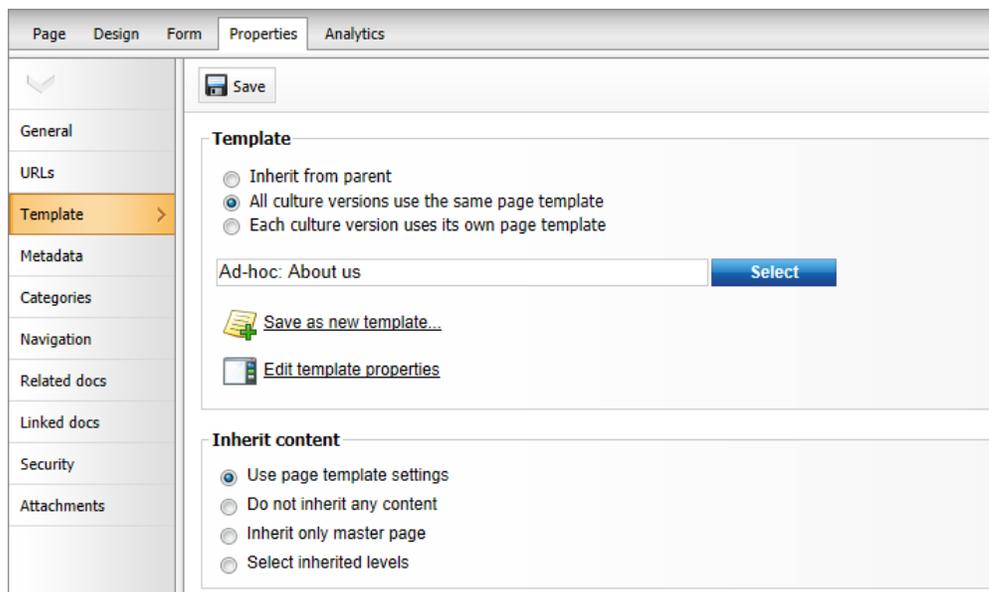
Congratulations, you have just created your first page template using web parts.

5.6 Re-using page templates

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

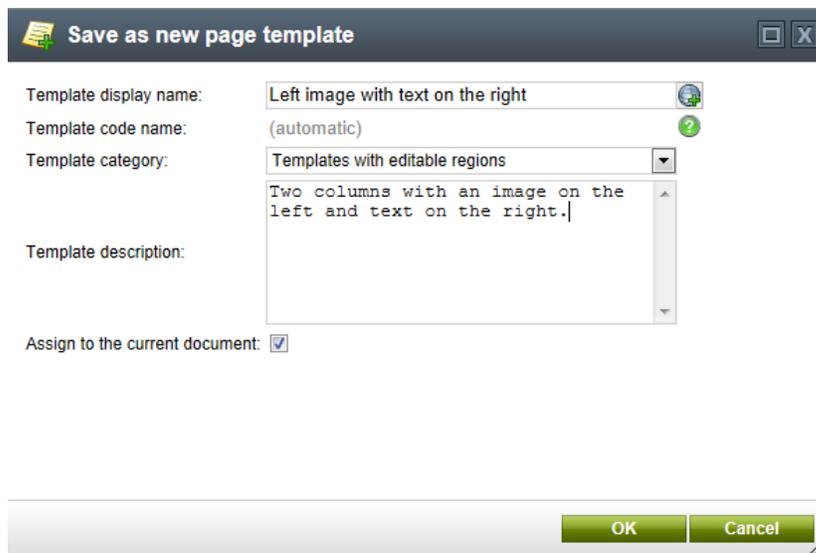
Now, you will learn how to convert the page-specific template into a **re-usable page template** that can be used to create several pages with the same general layout, but different content.

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



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

- **Template display name:** Left image with text on the right
- **Template category:** Templates with editable regions
- **Template description:** Two columns with an image on the left and text on the right.
- **Assign to the current document:** Yes (leave it checked)



Confirm the creation of the template by clicking **OK**.

Result

You have just created a new re-usable page template. If you try to add a new page now, the page template is available among the offered options in the **Templates with editable regions** category and you can easily create another page with the same structure as the *About Us* page.

Because the **Assign to the current document** option was checked, the system also automatically assigns this template to the current page instead of the previous ad-hoc template.



Modifying shared page templates

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

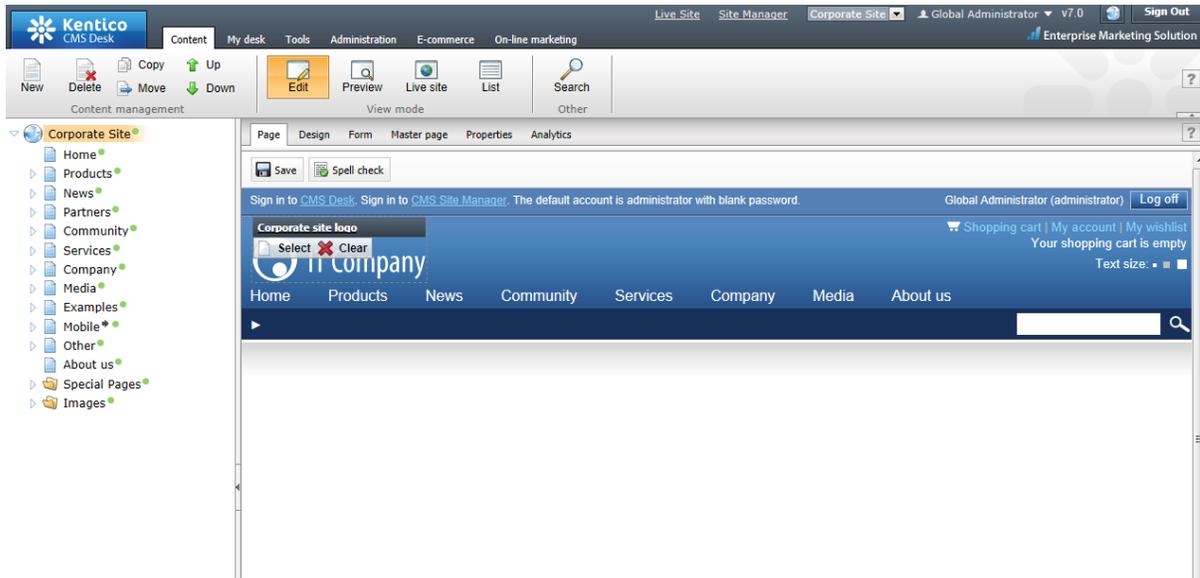


Important!

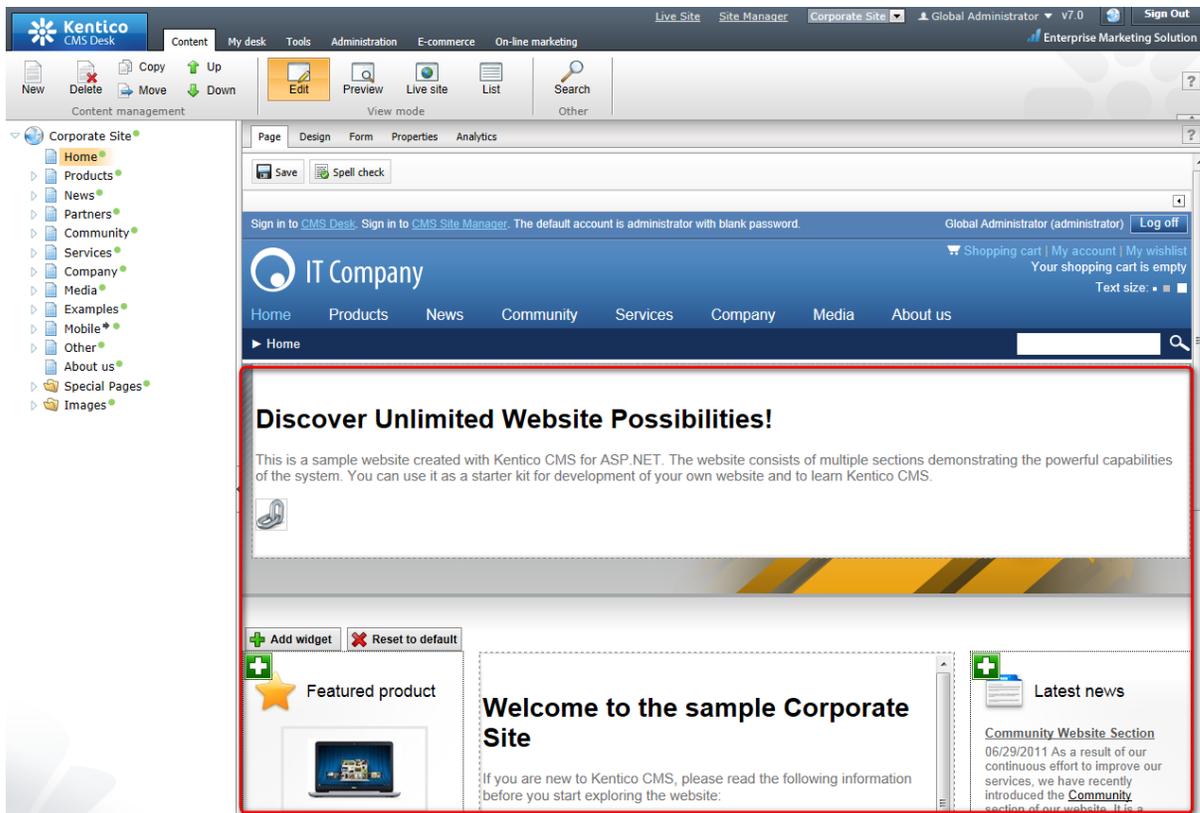
Please keep in mind that when you create a page based on an ad-hoc page template and later delete it, the corresponding ad-hoc template will be deleted as well and **cannot be restored**.

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:

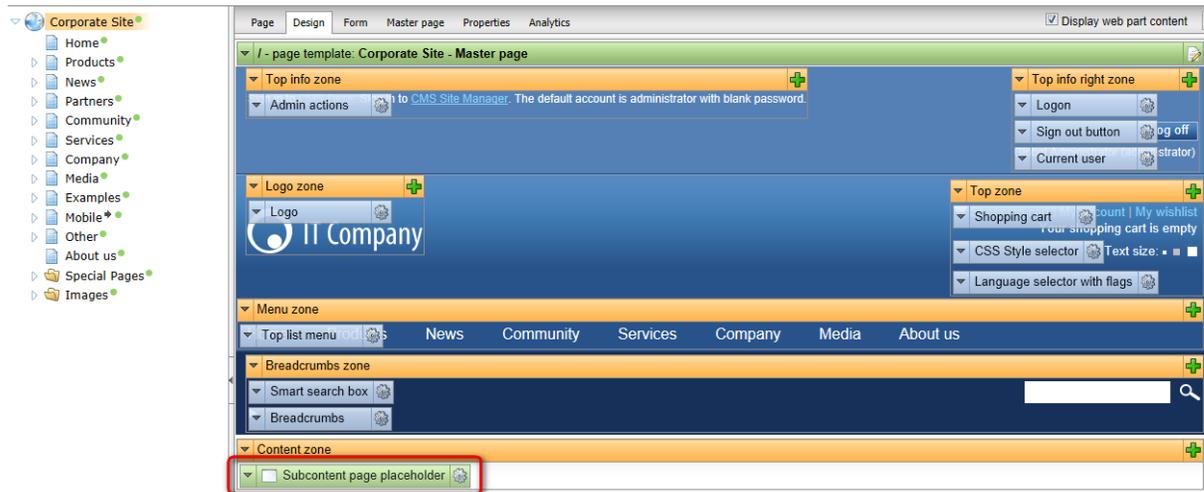


... and here's the home page nested inside the master page:



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:

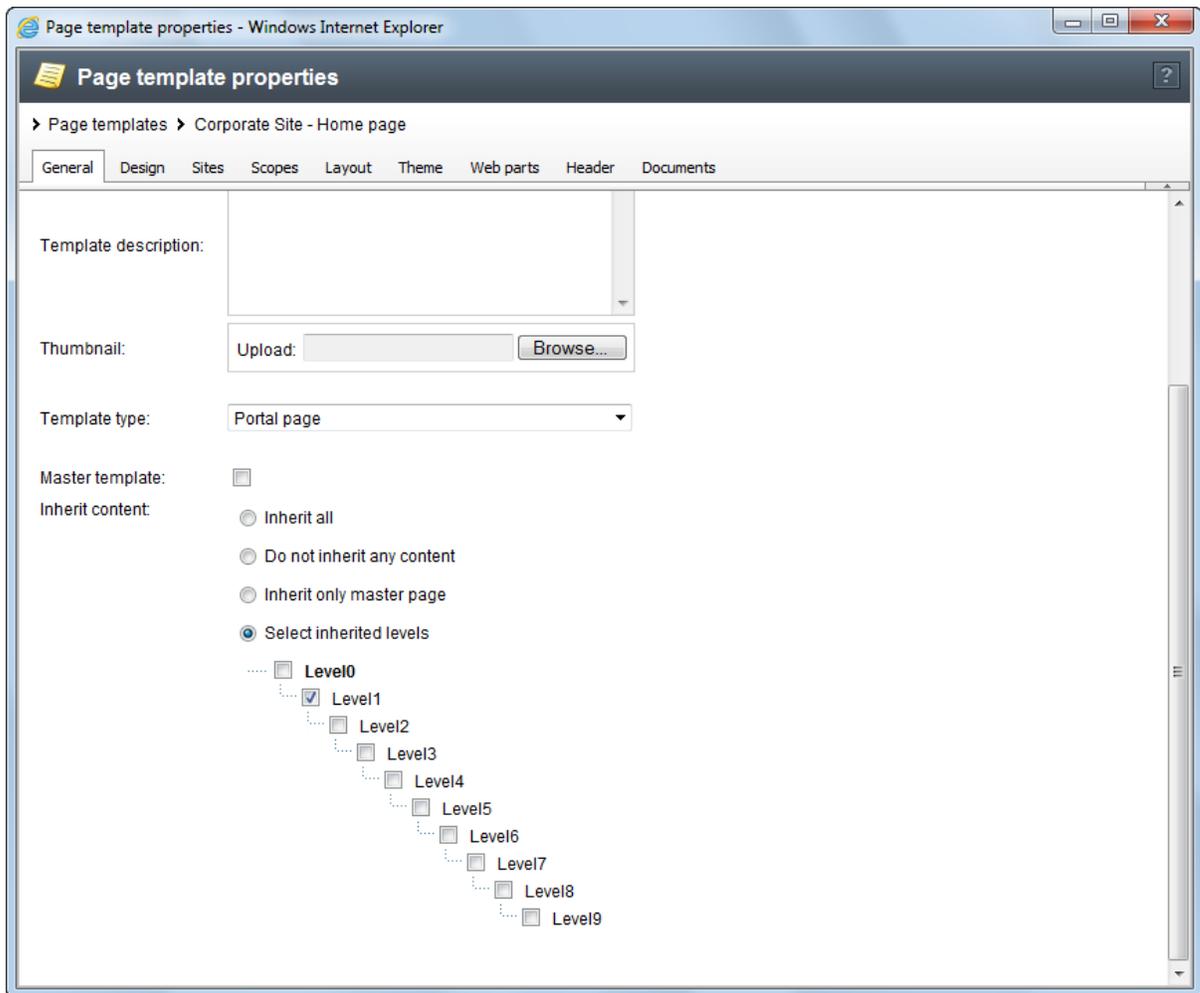


5.8 Configuring visual inheritance

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

Select the `/News` page, go to its **Properties -> Template** tab and click the  **Edit template properties** action. Now you can set the **Inherit content** value to *Select inherited levels* and check only the **Level 1** box. This 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 confirm 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'. The 'Content' menu is expanded, showing options like 'New', 'Delete', 'Move', 'Copy', 'Up', 'Down', 'Edit', 'Preview', 'Live site', 'List', and 'Search'. The left sidebar shows a tree view of the 'Corporate site' with folders for 'Home', 'Services', 'Products', 'News', 'Partners', 'Community', 'Company', 'Media', 'Examples', 'Mobile', 'Other', 'Special Pages', and 'Images'. The 'News' folder is selected, and the main content area shows a 'News List' with a search box and a single news item titled 'Community Website Section' by Brad Summers, dated 6/29/2011 12:00:00 AM.

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

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

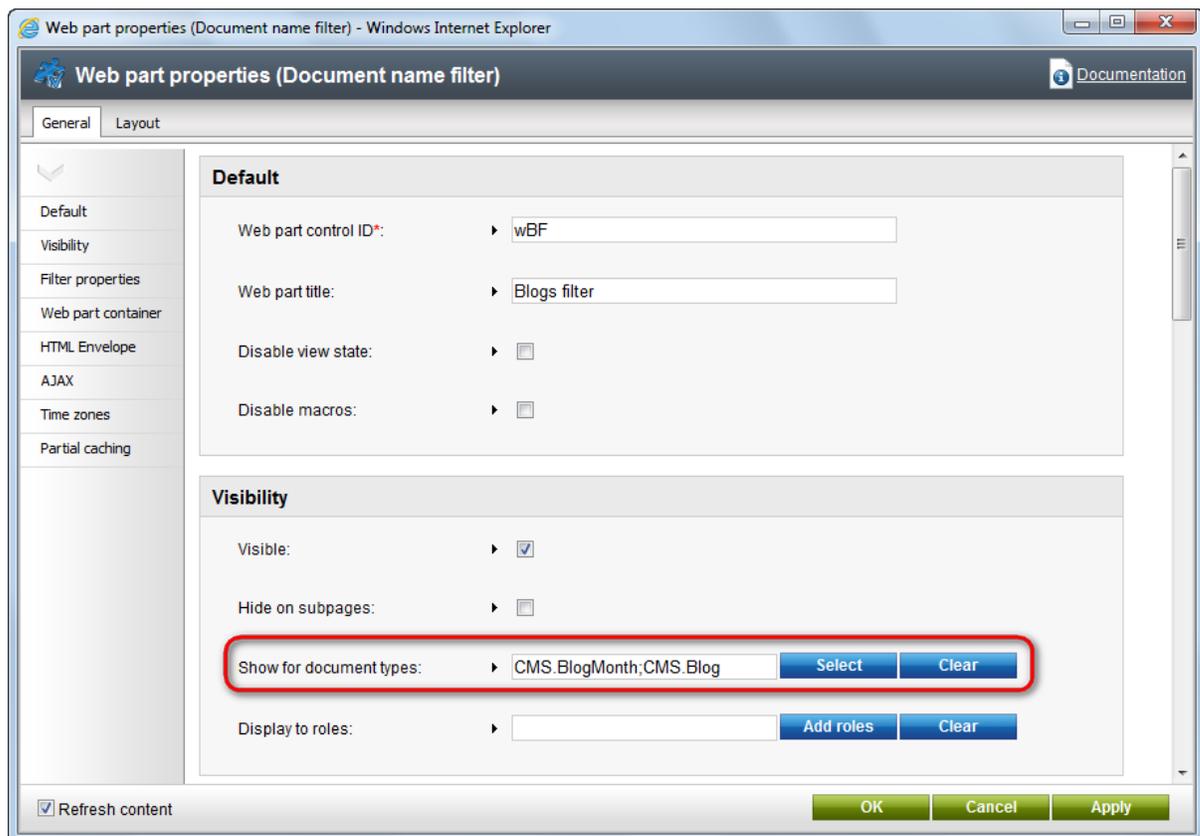
The screenshot shows the 'Properties' dialog box in Kentico CMS. The 'Template' tab is selected, and the 'Template' section is expanded. The 'Inherit content' section is also expanded. The 'Template' section shows three radio button options: 'Inherit from parent', 'All culture versions use the same page template', and 'Each culture version uses its own page template'. The 'Each culture version uses its own page template' option is selected. Below these options is a text box containing 'Corporate Site - News' and a 'Select' button. There are also links for 'Save as new template...', 'Clone template as ad-hoc', and 'Edit template properties'. The 'Inherit content' section shows 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 checkbox for 'Root' is 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/Andrew Jones 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 **Andrew Jones 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 ► Andrew Jones Blog

Blogs 27 Events Forums Wiki

Andrew Jones Blog

Hi, my name is Andrew Jones 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.

Andrew Jones | 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.

► Community ► Blogs ► Brad Summers Blog ► March 2011 ► Expanding to Europe

Forums Blogs 27 Events Wiki

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.



As you could already get to know from the News section, we have recently opened a new office in London, United Kingdom. The office has already started its operation and first projects are to be delivered soon, so I finally found some time to share my impressions from its setup.

When the idea to expand our operations to Europe first came to mind, everyone was quite skeptical about it. However, after some market research, we found out that there is a place for a company with our know-how in the European market and it was decided to make the idea come alive. I, as the head of web development in our company, was assigned the goal of overseeing the setup of the European branch and to participate in hiring of the first employees.

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 the **Community** document in the content tree and click **Live site**. The repeater web part displaying the latest blog posts uses the *Corporate Site - Content box* container:

With container:	Without container:
<div style="border: 1px solid #ccc; padding: 5px;"> <div style="background-color: #f0f0f0; padding: 2px; border-bottom: 1px solid #ccc;"> ▶ Blog post Details </div> <div style="padding: 5px;"> <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> </div> <div style="border-left: 1px dashed #ccc; padding-left: 5px; margin-left: 10px;"> <p>Brad Summers 03/23/2011 2 comment(s)</p> <p>Brad Summers 03/21/2011 2 comment(s)</p> </div> </div>	<div style="padding: 5px;"> <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> </div> <div style="border-left: 1px dashed #ccc; padding-left: 5px; margin-left: 10px;"> <p>Brad Summers 03/23/2011 2 comment(s)</p> <p>Brad Summers 03/21/2011 2 comment(s)</p> </div>

Containers 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.

Here's an example of the **Corporate site - Content box** container's code:

HTML code:

```
<div class="contentBox">
  <div class="head"><h3 class="{ContainerCSSClass%}">{ContainerTitle%}</h3>
  <div class="details">Details</div>
</div>
<div class="content">
  -Web part placeholder-
</div>
</div><div class="clear"></div>
```

As you can see the container is composed of simple HTML code. The position of the actual web part content is represented in the code by a placeholder, so you can add both opening and closing elements. 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. The same applies to the `{ContainerCSSClass%}` macro, which can be used to dynamically set the CSS class of the container.

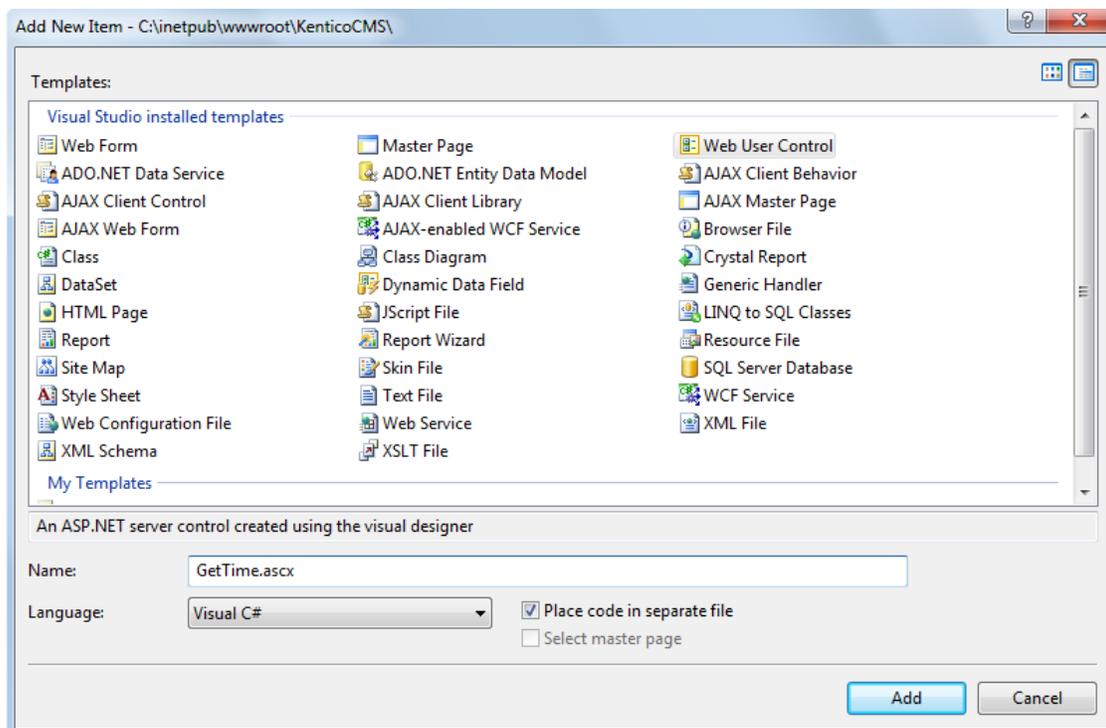
5.10 Adding custom code to your website

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.



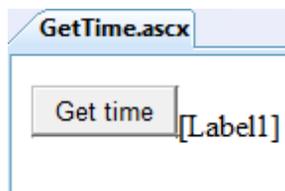
Click **Add**. Switch to the **Design** tab, drag and drop the following controls and set their properties:

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()
```

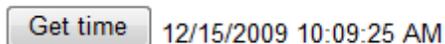
This code ensures that the label displays the current date and time when the button is clicked. It's not necessary to compile the project — user controls are compiled at run time.

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 **General -> User control** web part to the **Main zone**. Enter the following value into the web part's **User control virtual path** property: *~/GetTime.ascx*

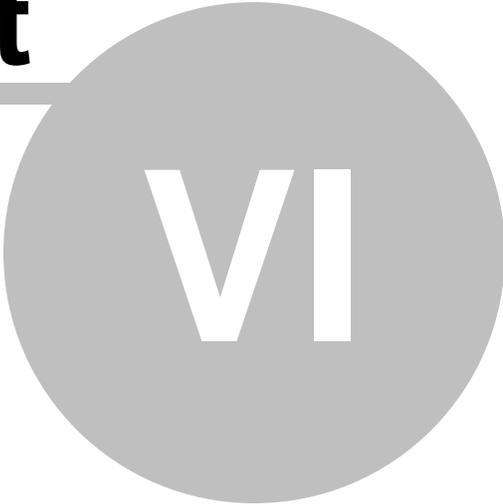
The ~ character represents the root of your web application. Click **OK**. Click the **Live site** mode and now you can see the user control inserted onto the page. When you click the **Get time** button now, the current date and time is displayed next to the button:



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 cases where 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.

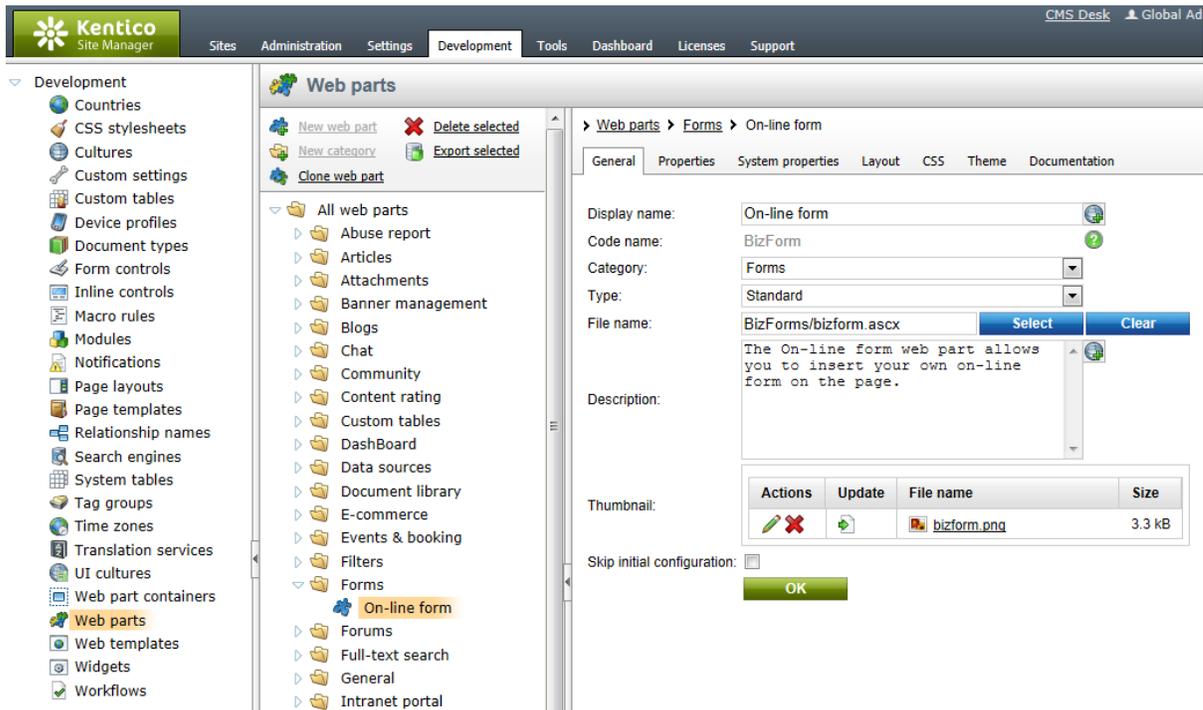
A web part is a user control (ASCX) that inherits from the **CMSPortalControls.CMSAbstractWebPart** class. You can easily create your own web parts using Visual Studio or Visual Web Developer Express Edition and use them to add your custom .NET code onto the pages of the website.

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 settings 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** - sets the type of the web part, which affects its behavior and properties. Different web part types are also marked with different colors and icons on the *Design* tab of *CMS Desk*. 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 another connected web part.
 - **Filter** - can be connected to a data source and enables users to limit the range of 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.
 - **Layout** - these web parts can be used to generate a specific layout for page content by defining additional web part zones.
 - **Widget only** - these web parts are only intended to serve as base templates for widgets and are not available in the web part catalog.
 - **Wireframe** - special type of web parts used when defining wireframe schematics for pages. They are only for internal purposes and are not displayed on the live site. Wireframes can be created for pages on the *Wireframe* tab in the *Edit* mode of *CMS Desk*, either for dedicated wireframe documents or other document types that contain a wireframe definition.
- **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 selection catalog and as a tooltip in the web part toolbar.
- **Thumbnail** - used to upload an image that will represent the web part in the selection catalog and web part toolbar.
- **Skip initial configuration** - if enabled, the property configuration dialog will be skipped when adding the web part to a page in *CMS Desk*. This can be convenient, particularly in the case of web parts

that are typically used with their default property values.



On the **Properties** tab, you can define the web part properties and how they appear in the **Web part properties** configuration dialog.

All standard web parts have 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 behavior and settings (e.g. for it not to be displayed in the web part configuration dialog) you can do so by overriding it on the **Properties** tab and configuring it differently.

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.

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.

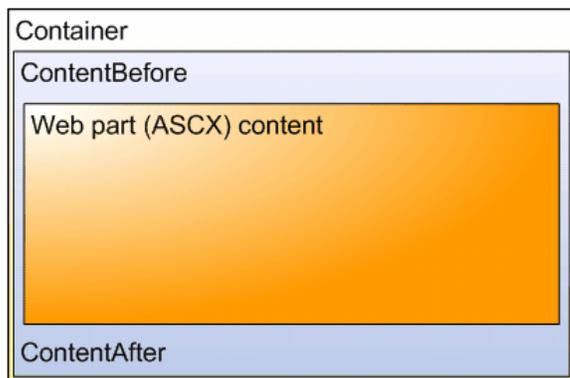
Web part container

- **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.
- **Hide container on subpages** - if enabled, the container will not be displayed around the web part on child documents that inherit it through visual inheritance. For example, this allows you to add a container for a master page only.

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.

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



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 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.

Performance

- **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.
- **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.

Output filter

The settings in this section are used to configure additional processing for the output code generated by specific instances of the web part. If you are interested in this topic, please see [Development -> Page processing and URLs -> Output filters](#) in the Kentico CMS Developer's Guide.



Storing files related to a web part

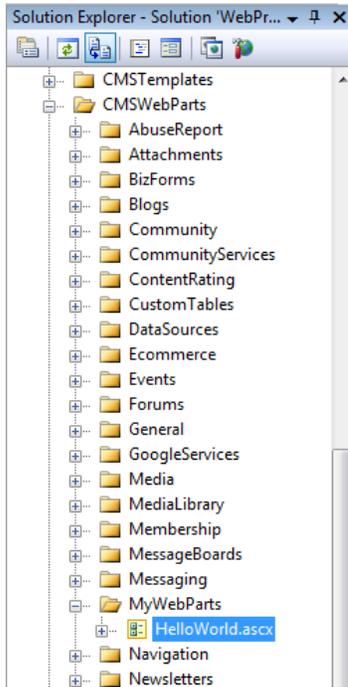
If your web part consists of several files (such as other ASCX controls, images, js scripts, etc.), you should place these files in a sub-folder under the folder where your main web part ASCX file is placed. If the code name of the web part is **MyWebPart**, the sub-folder's name must be **MyWebPart_Files**.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

6.2 Creating a simple web part

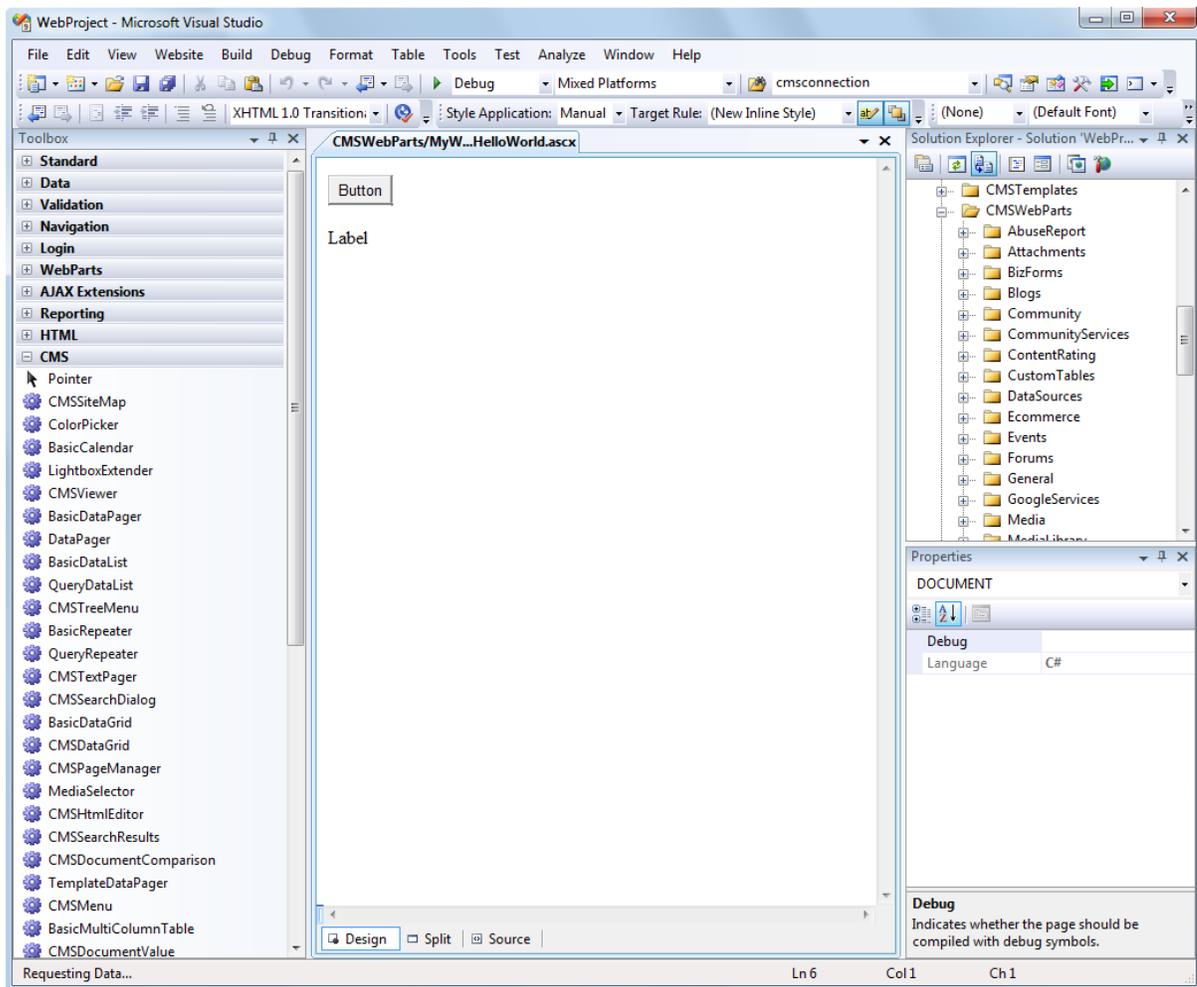
This topic will guide you through the process of creating a very simple "Hello world" web part that displays a label and a button. When the button is clicked, it updates the current time displayed in the label.

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  **Save**.

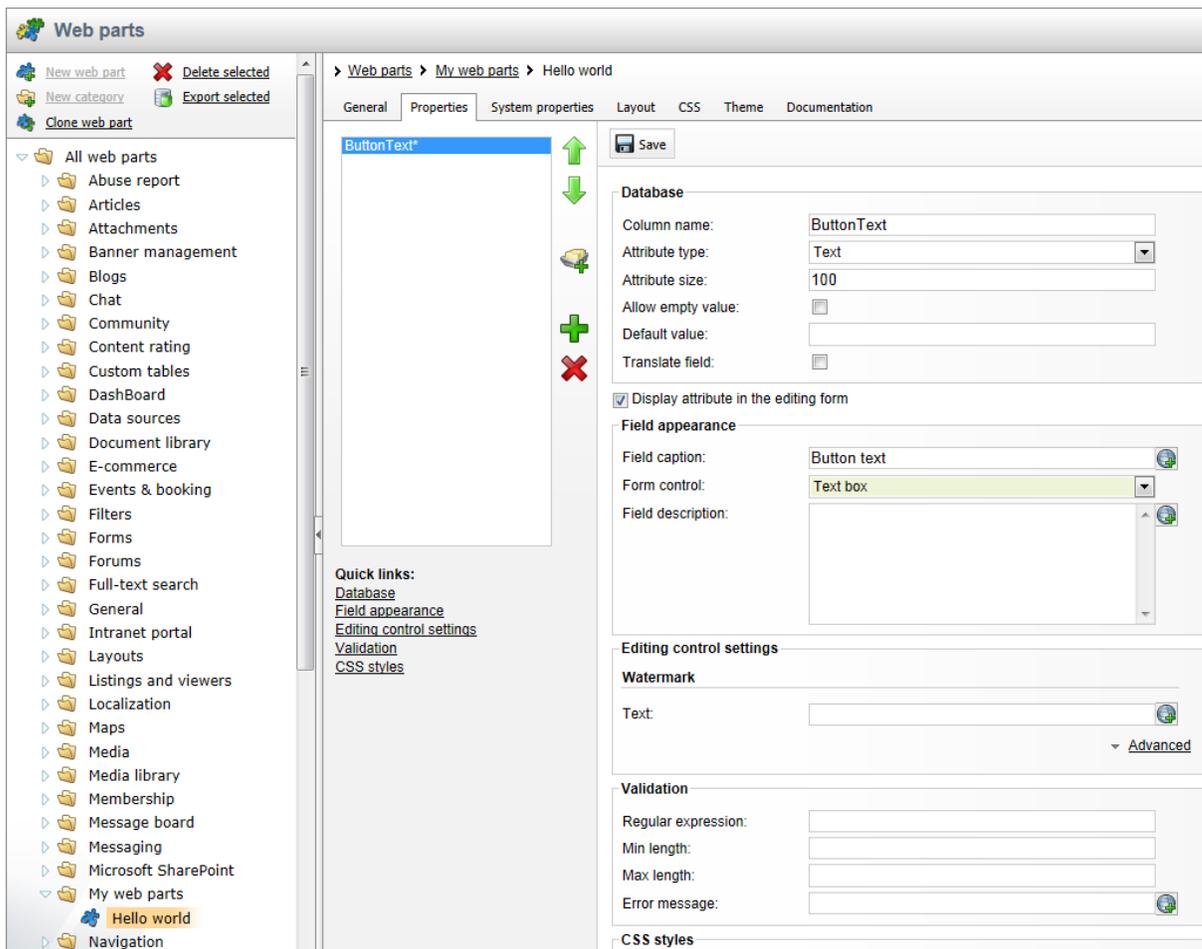
11. Select the new category and click  **New web part**. Choose to **Create a new web part** and enter the following values:

- **Display name:** Hello world
- **Code name:** HelloWorld
- **File path:** ~/CMSWebParts/MyWebParts/HelloWorld.ascx
- **Generate the code files:** false (we have already prepared the web part's source files in the previous steps, so there is no need to generate them)

Click  **Save**.

12. Switch to the **Properties** tab and add the following property using the **New attribute** () button:

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



The screenshot shows the 'Web parts' interface in SharePoint. The 'Properties' tab is selected for the 'Hello world' web part. The 'ButtonText*' property is highlighted in the list. The 'Database' section is expanded, showing the following configuration:

- Column name: ButtonText
- Attribute type: Text
- Attribute size: 100
- Allow empty value:
- Default value:
- Translate field:

The 'Field appearance' section is also expanded, showing:

- Field caption: Button text
- Form control: Text box
- Field description:

The 'Editing control settings' section is expanded, showing:

- Watermark: Text:
- Validation: Regular expression: , Min length: , Max length: , Error message:
- CSS styles:

Click  **Save**.

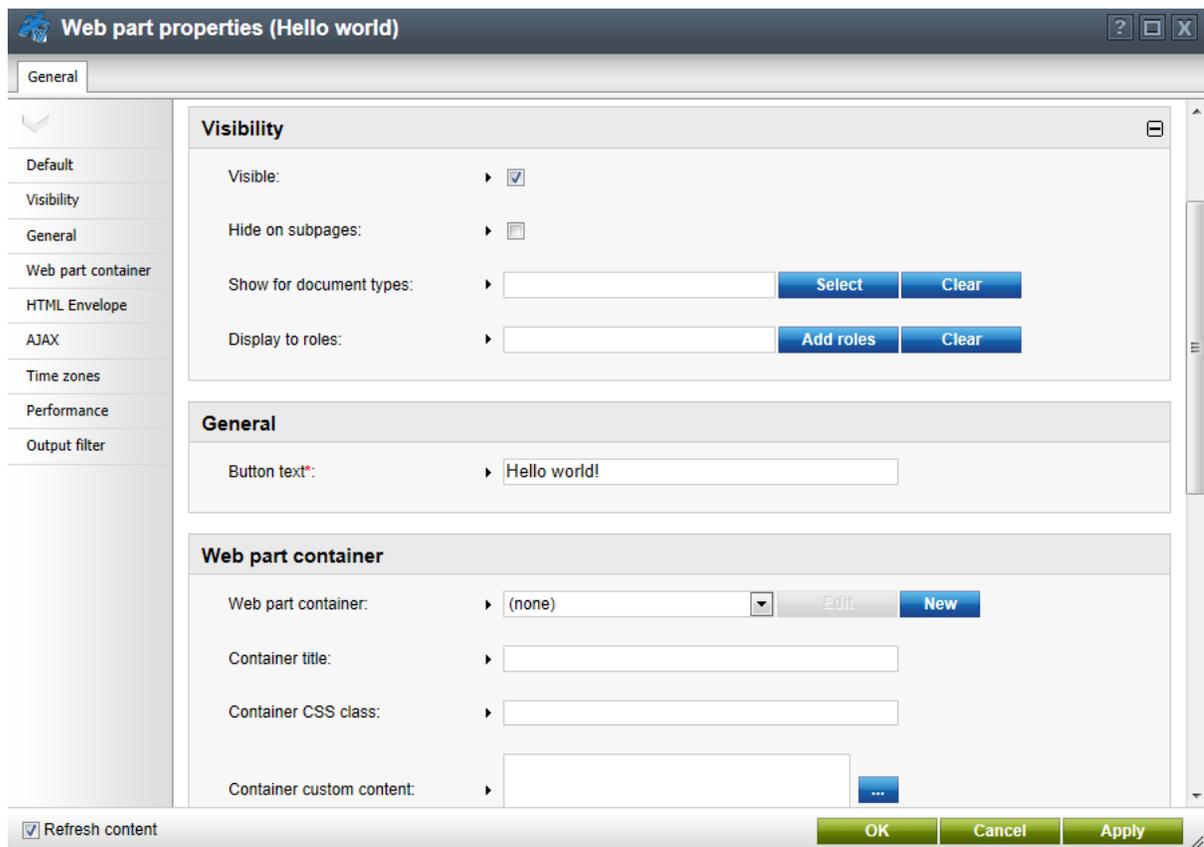
13. Switch to **CMS Desk**.

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

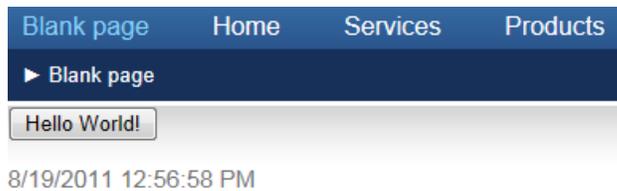
15. Click **Add web part** (+) in the upper right corner of the web part zone and choose to add the **Hello world** web part:



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



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 retrieved from Kentico CMS in a customized format.

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.

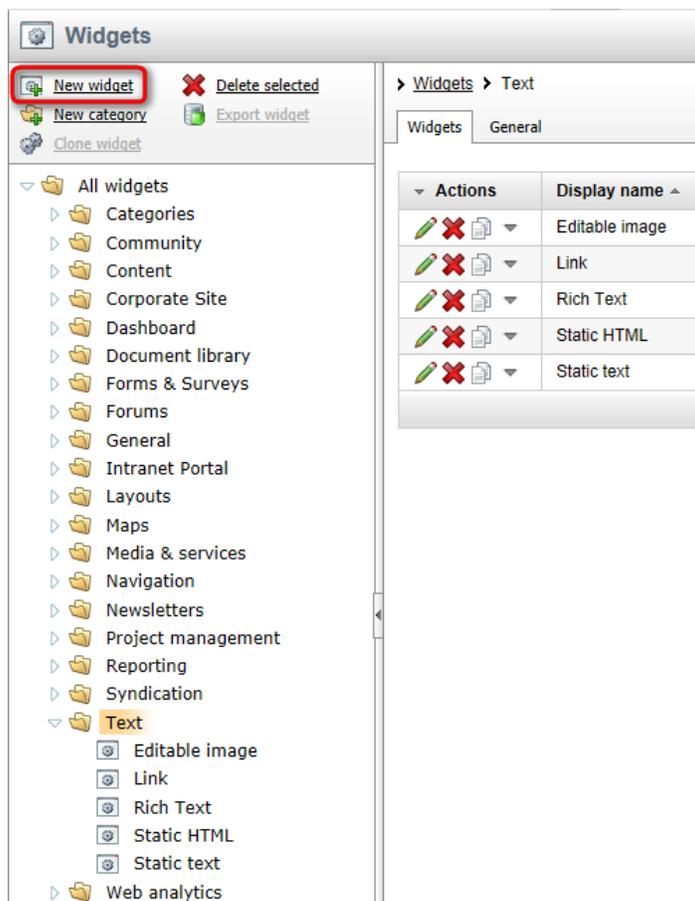
1. To start, open **Site Manager** -> **Development** -> **Widgets** to access the administration interface used to manage widgets.

The screenshot shows the Kentico CMS 7.0 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 widget instances.

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 **Text** category and click the **New widget** button.



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 & Images -> 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** button. Next, do the same for the **DialogHeight** property.

The screenshot shows the Kentico CMS Widgets configuration interface. On the left, a tree view shows the widget hierarchy: All widgets > Text > Editable text. The 'Editable text' widget is selected. The main area is divided into tabs: General, Properties, System properties, Documentation, and Security. The 'Properties' tab is active, displaying a list of properties for the 'Editable text' widget. The 'RegionTitle' property is selected, and its configuration is shown in the 'Database' section. The 'Display attribute in the editing form' checkbox is checked and highlighted with a red box. Other sections include 'Field appearance' and 'Editing control settings'.

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.

> [Widgets](#) > [Text](#) > Editable text

General Properties System properties Documentation Security

	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"/>

Please select the authorized roles (available only when you select the "Authorized roles" option above):

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 permission 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 **Text -> 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 displays the Kentico CMS 7.0 Design view interface. At the top, there are tabs for 'Page', 'Design', 'Form', and 'Properties'. Below these are buttons for 'Save' and 'Spell check'. A rich text editor toolbar is visible, containing various icons for text formatting (bold, italic, underline, text color, background color), alignment, and list creation. Below the toolbar are dropdown menus for 'Styles', 'Normal', 'Font', and 'Size', along with color pickers and a font size input field.

The main editing area is divided into two sections:

- Left sidebar:** Contains a '+ Add widget' button and a 'Reset to default' button. Below these is a 'Newsletter' widget with a mail icon, a 'First Name:' input field, a 'Last Name:' input field, an 'E-mail:' input field, and a 'Subscribe' button. Below the widget is a 'Left editable text' section with the placeholder text 'Sample text...'.
- Main content area:** Contains the following text:
 - 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:
 - CMS Desk:** <http://<your domain>/CMSDesk>
 - Site Manager:** <http://<your domain>/CMSSiteManager>
 - On the logon page that appears, use the following default credentials:
 - User name:** administrator
 - Password:** leave the password blank

Part

VII

Managing styles and design

7 Managing styles and design

7.1 CSS styles

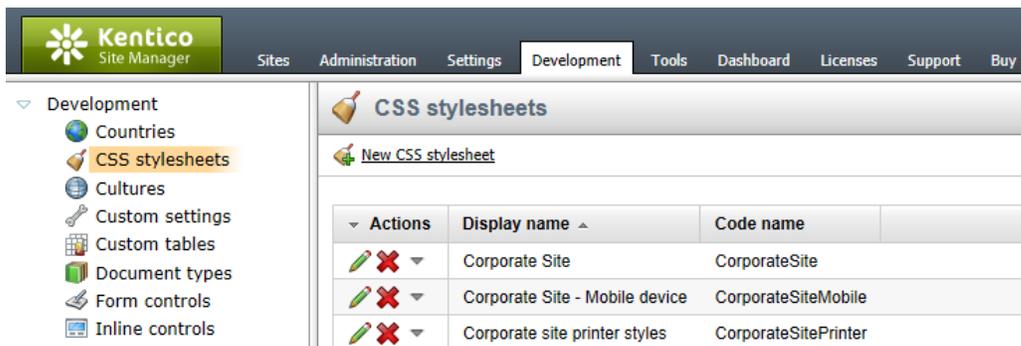
The design of the website relies on standard CSS styles. Each website has its global CSS stylesheet that can be selected in **Site Manager -> Sites -> ... edit site ... -> General**. Here you can also choose a different stylesheet for the site's WYSIWYG editors.

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 chosen stylesheet can be accessed by clicking the **Edit** button.

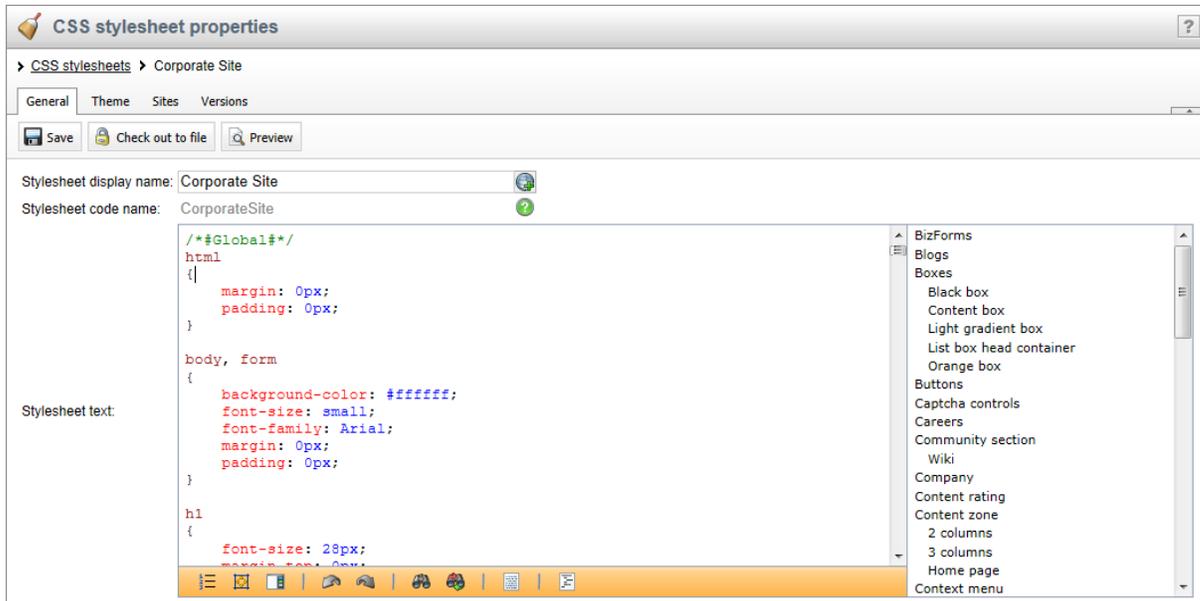
All stylesheets in the system may also be managed in Site Manager. When you are in CMS Desk, you can easily switch to Site Manager by clicking the **Site Manager** link in the header:



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



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



When modifying a stylesheet, you can use the  **Preview** button in the header to view the website's pages side-by-side with the CSS code. This allows you to immediately check how changes affect the appearance of the site. Please note that while the preview works immediately, you may need to clear your browser cache before the new CSS design is reflected on the actual live site.



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 navigation menu. The main menu used on the sample Corporate Site is displayed using the **CSS list menu** web part which is based on the **CMSListMenu** server control.

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

[CSS]

```
.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. The styles can be modified in the global stylesheet of the given site.

Go to **Site Manager -> Development -> CSS stylesheets** and edit (✎) the **Corporate Site** stylesheet. Then click the **Preview** button in the header of the **General** tab, which will allow you to view the site's pages while working with the stylesheet's code. Switch to a **Horizontal layout** using the appropriate button on the preview toolbar. You also need to specify a page for the preview, so for example enter */Home* into the path textbox on the toolbar and **Refresh** (🔄) the page section.

To allow easy navigation, the CSS code is separated into blocks that can be accessed using the panel on the right side of the editor. Select the **Top menu** bookmark and the editor will scroll down to the definitions of the CSS classes used to style the menu (starting with those listed above). In the page preview section below, you can see the default appearance of the website's main menu.

The screenshot shows the 'CSS stylesheet properties' window for the 'Corporate Site'. The 'General' tab is active, showing CSS code for the top menu. The code includes styles for the menu container and list items. A right-hand panel lists various site features, with 'Top menu' selected. Below the editor, a preview of the website's main menu is shown, featuring the 'IT Company' logo and navigation links: Home, Products, News, Community, Services, Company, and Media. The 'Top menu' is highlighted in orange in the preview.

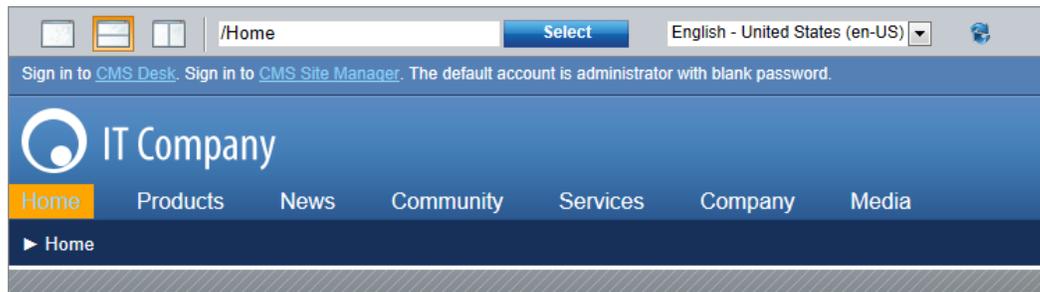
Now we will change the background color of highlighted menu items to orange. Scroll down in the CSS code until you find the class definition shown below and insert the highlighted line:

[CSS]

```
.zoneMenu .CMSListMenuHighlightedLI a,
.zoneMenu .CMSListMenuHighlightedLIfirst a
{
    color: #8cd2f8 !important;
}
```

```
background-color: orange;  
text-decoration: none;  
}
```

Click  **Save** to confirm the change. The page preview section will automatically be refreshed and you will be able to see the modified menu.



Of course, the new style will also be applied to the live site.

Defining the style of a single menu item

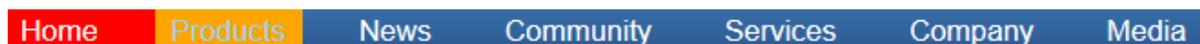
Every document may have its own style that is used when the document is displayed in a menu. We will try to modify the style of the **Home** menu item. Go to **CMS Desk -> Content** and select **Home** from the content tree. Then switch to the **Properties -> Navigation** tab. Here you can define the following settings:

- **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 web parts and controls, i.e. included in menus.
- **Show in sitemap** - indicates if the document should be included in the website's sitemap.
- **Menu actions** - determines what happens when the document's 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.
- **Search & SEO** - can be used to exclude the document from searches and set up advanced properties related to the website's Google sitemap.
- **Menu design** - the fields in this section allow you to set styles applied to the particular document's menu item. The styles can be specified for three different scenarios: standard, when a user hovers over the menu item, and when the given document is currently selected (highlighted).

Enter the following value into the **Menu item style** value (in the **Menu item design** section):

background-color: red;

Confirm the change by clicking  **Save**. To view the result, switch to **Live site** mode and select **Products** in the website's menu. The **Home** menu item will now be displayed in red:



Part

VIII

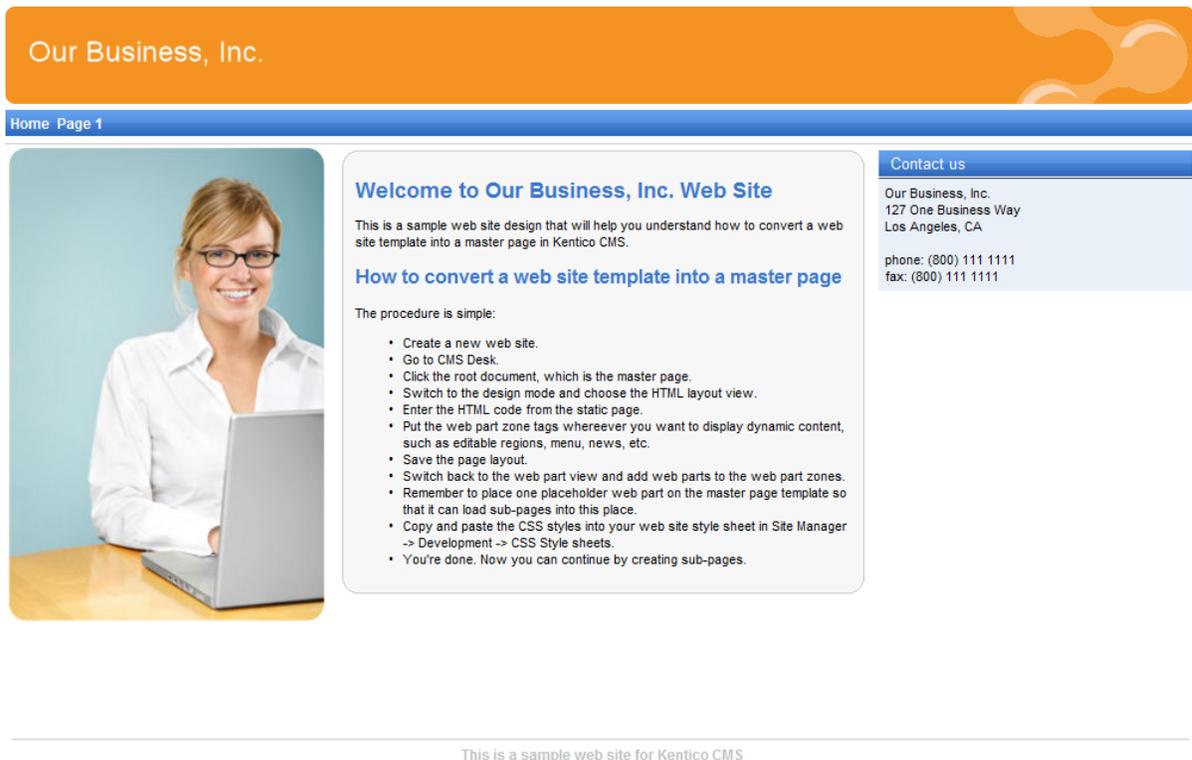
Creating a new site using the portal engine

8 Creating a new site using the portal engine

8.1 Overview

This part of the tutorial guides 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 the sample Corporate Site. We will leave the existing website and add a new site running under the *http://127.0.0.1* domain.



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 other domain than

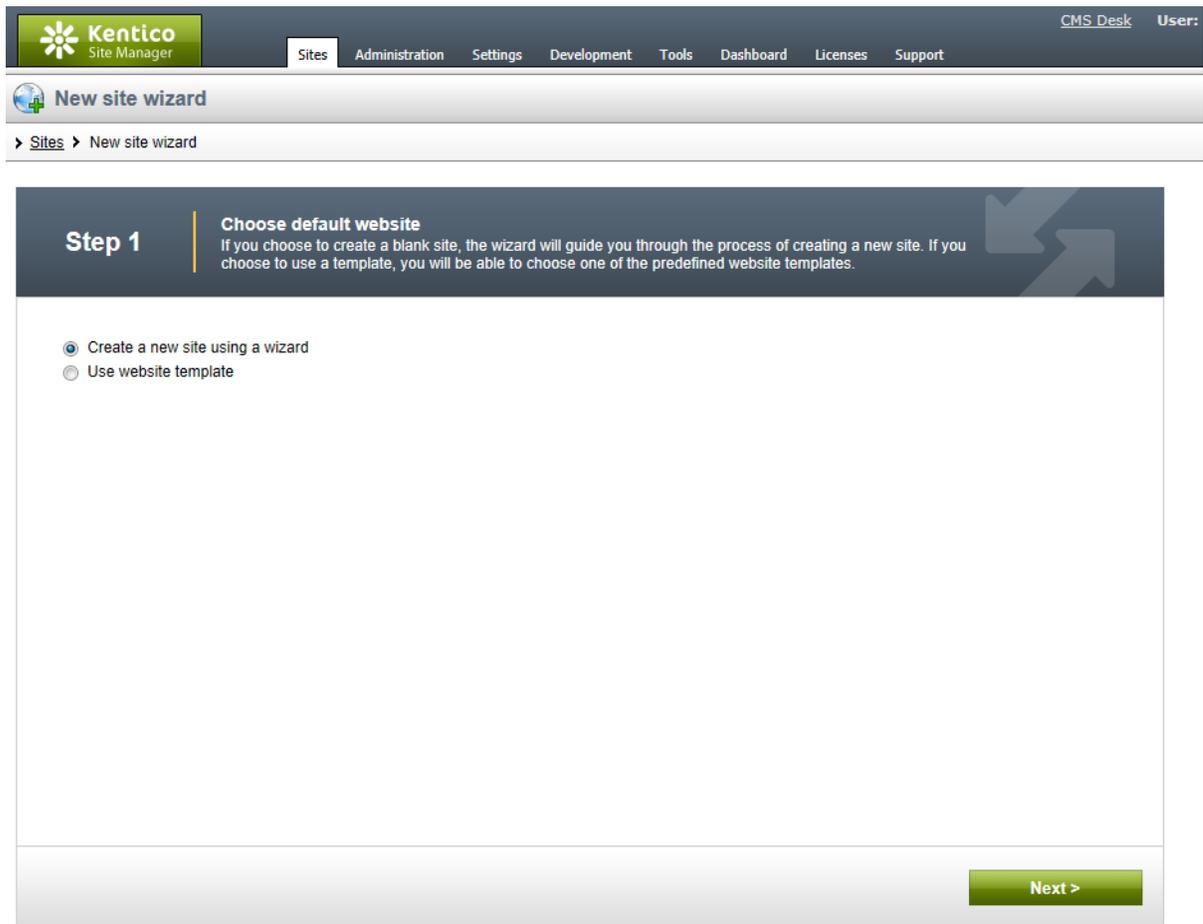
localhost, you will use the *localhost* domain again.

1. Sign in as *Administrator* and go to **Site Manager -> Sites**.
2. Click  **New site wizard**.



The **New site wizard** opens.

3. Select **Create a new site using a wizard**.



Click **Next**.

4. Enter the following details for the website:

- **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 **Domain name** value to *localhost*)
- **Site culture:** English - United States (the default culture determines how the website displays date, time and numeric values based on the different culture-specific format)

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 >](#)

Click **Next**.

5. The third step of the wizard allows you to select which objects the system imports into the 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 >

The fourth step displays the progress of object import.

Step 4

Import progress

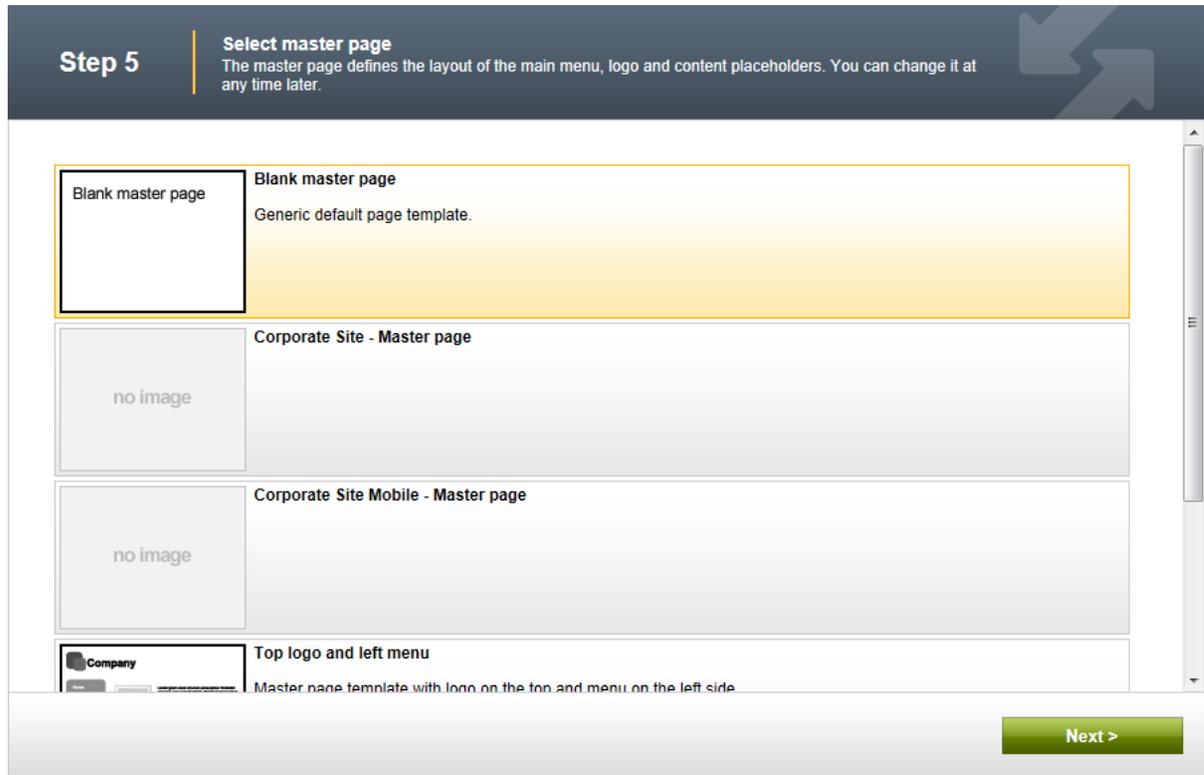
Objects are being imported.

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

Objects are being imported

< Previous **Cancel** Next >

6. Choose a master page template for the website. You can change or modify it later at any time. For now, select the **Blank master page**.



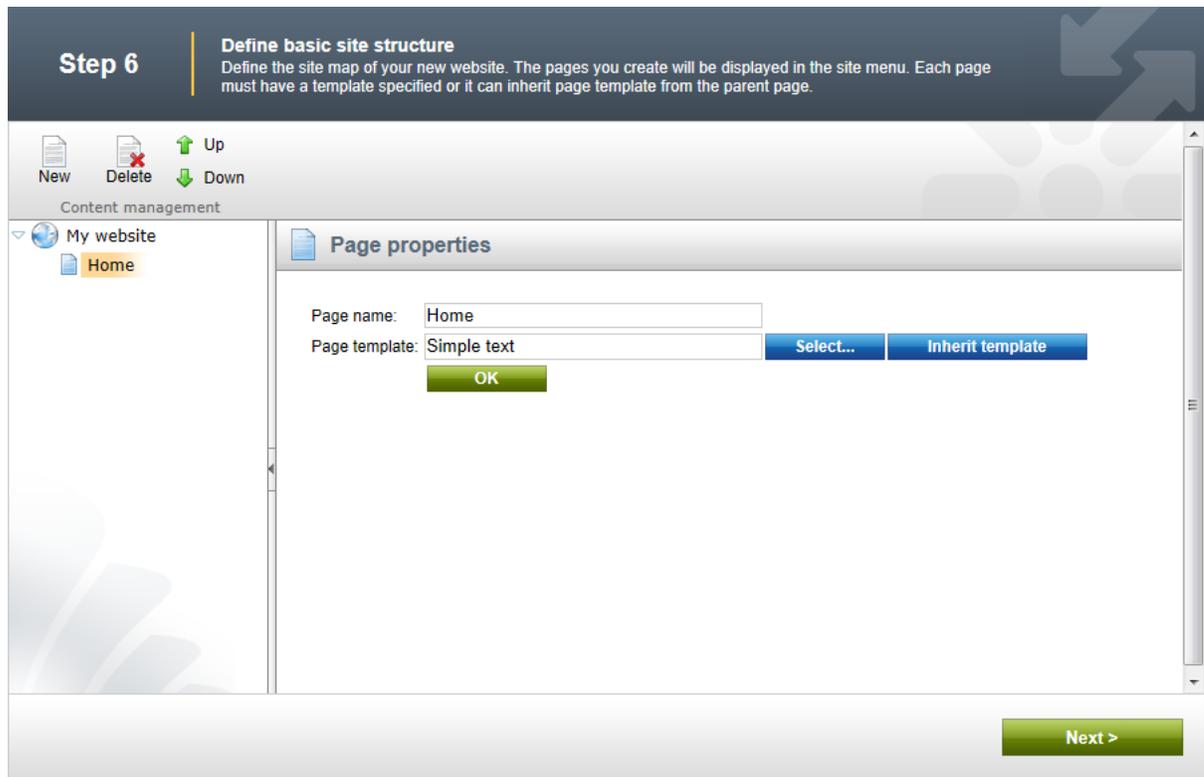
Click **Next**.

In **Step 6**, design the initial structure of the new website.

7. Click **New** and enter the following details:

- **Page name:** Home
- **Page template:** Click **Select**. Choose the **Templates with editable regions** category, select the **Simple text** page template and click **OK**.

Click **OK** to save the page.



8. Select the root page (**My website**), click **New** and enter the following details:

- **Page name:** News
- **Page template:** Click **Select**. Choose the **News** category, select the **News list** page template and click **OK**.

Click **OK** to save the page.

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 website
Home
News

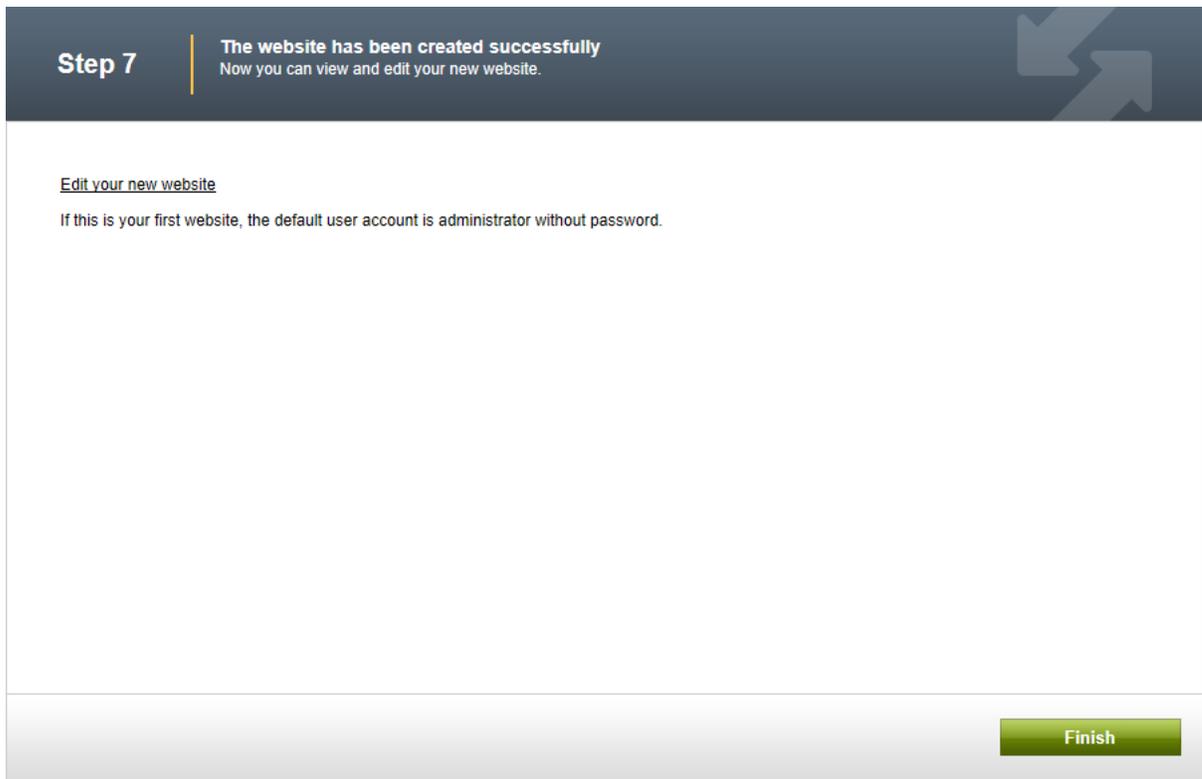
Page properties

Page name: News
Page template: News list Select... Inherit template
OK

Next >

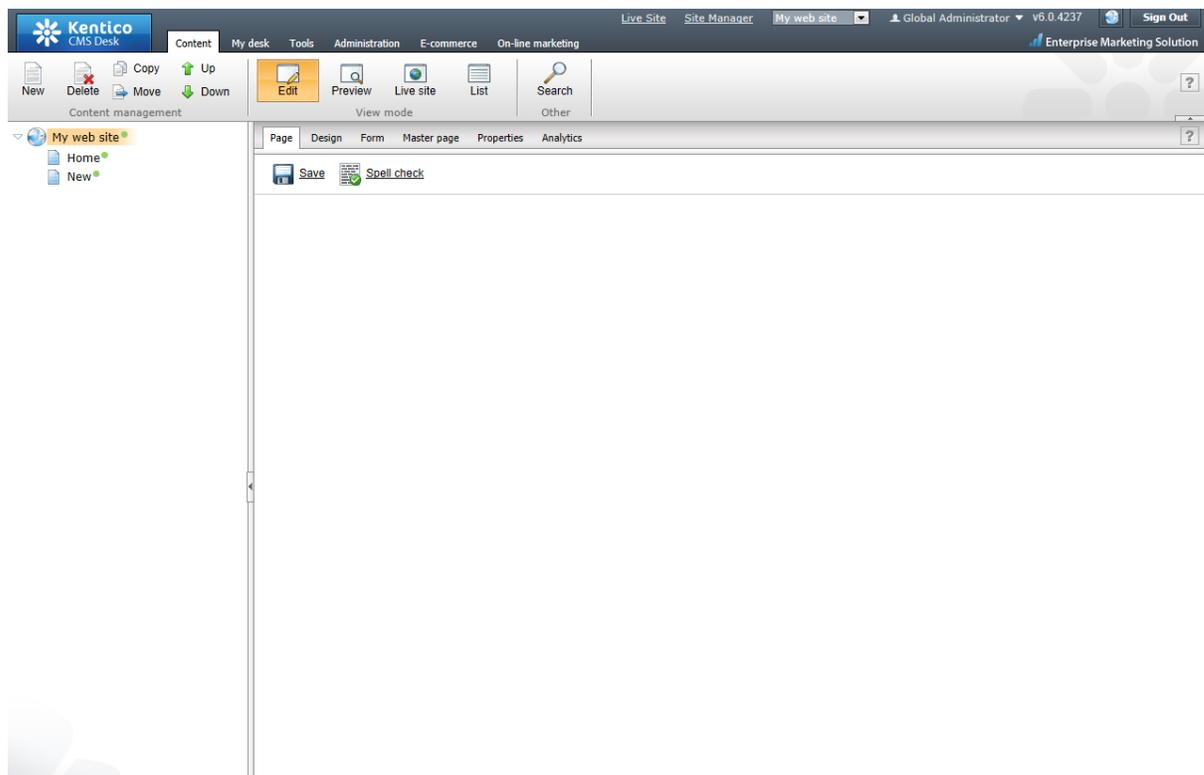
Click **Next**.

9. Click **Edit your new website**.



The screenshot shows a dark blue header bar with the text "Step 7" on the left and "The website has been created successfully" followed by "Now you can view and edit your new website." on the right. In the top right corner of the header, there are two overlapping grey arrows pointing towards the top right. Below the header is a large white area containing a blue underlined link "Edit your new website" and a line of text: "If this is your first website, the default user account is administrator without password." At the bottom right of the white area, there is a green button with the text "Finish".

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 website.



You have created the base for your new website. The next topics describe how to implement the required design.

8.3 Creating a CSS stylesheet

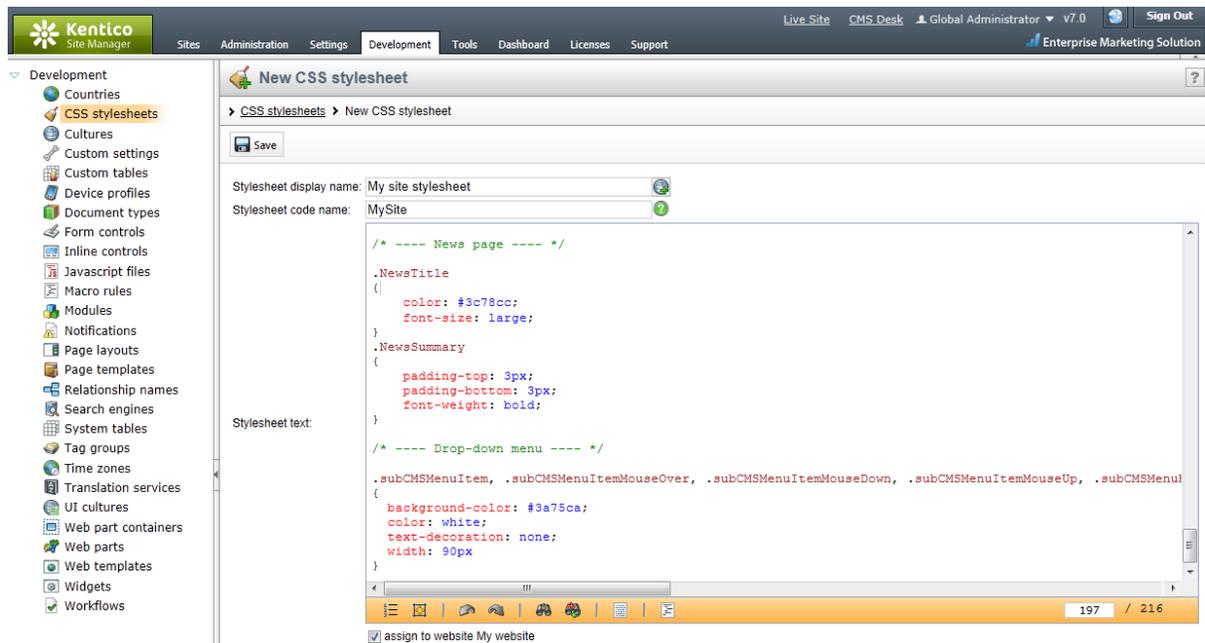
Before you start editing your new website, prepare a new CSS stylesheet based on the styles and images of the sample website template.

1. Go to **Site Manager -> Development -> CSS Stylesheets**.

2. Click **New CSS stylesheet**.

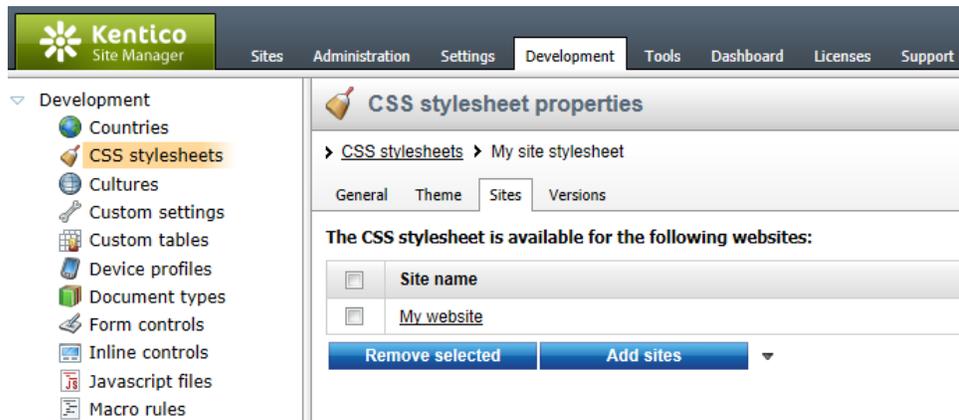
3. Enter the following values:

- **Stylesheet display name:** My site stylesheet
- **Stylesheet code name:** MySite
- **Stylesheet text:** copy and paste all code from *C:\Program Files\Kentico CMS\<version>\CodeSamples\SampleWebTemplate\Styles\main.css*



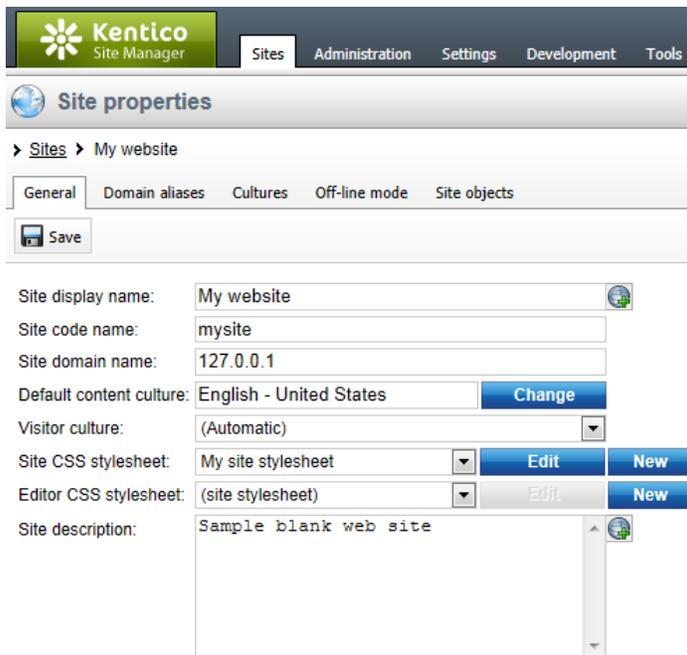
4. Click **Save**.

5. Switch to the **Sites** tab and assign the stylesheet to **My website**.



6. Navigate to **Site Manager** -> **Sites** and **Edit** () the properties of **My website**.

7. On the **General** tab, select *My site stylesheet* as the **Site CSS stylesheet**.



The screenshot shows the 'Site properties' page in the Kentico Site Manager. The page is titled 'Site properties' and has a breadcrumb trail 'Sites > My website'. Below the breadcrumb, there are tabs for 'General', 'Domain aliases', 'Cultures', 'Off-line mode', and 'Site objects'. A 'Save' button is visible. The 'General' tab is active, showing the following fields:

- Site display name: My website
- Site code name: mysite
- Site domain name: 127.0.0.1
- Default content culture: English - United States (with a 'Change' button)
- Visitor culture: (Automatic)
- Site CSS stylesheet: My site stylesheet (with 'Edit' and 'New' buttons)
- Editor CSS stylesheet: (site stylesheet) (with 'Edit' and 'New' buttons)
- Site description: Sample blank web site

8. Click  **Save**. This ensures that all pages of your new website load the appropriate stylesheet.

9. Copy the **SampleWebTemplate\app_themes\MySite** folder to the **App_Themes** folder under your web project.

The folder contains graphics for this website template. This location ensures that the images are exported as part of the website if you decide to move the website in the future. 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
```

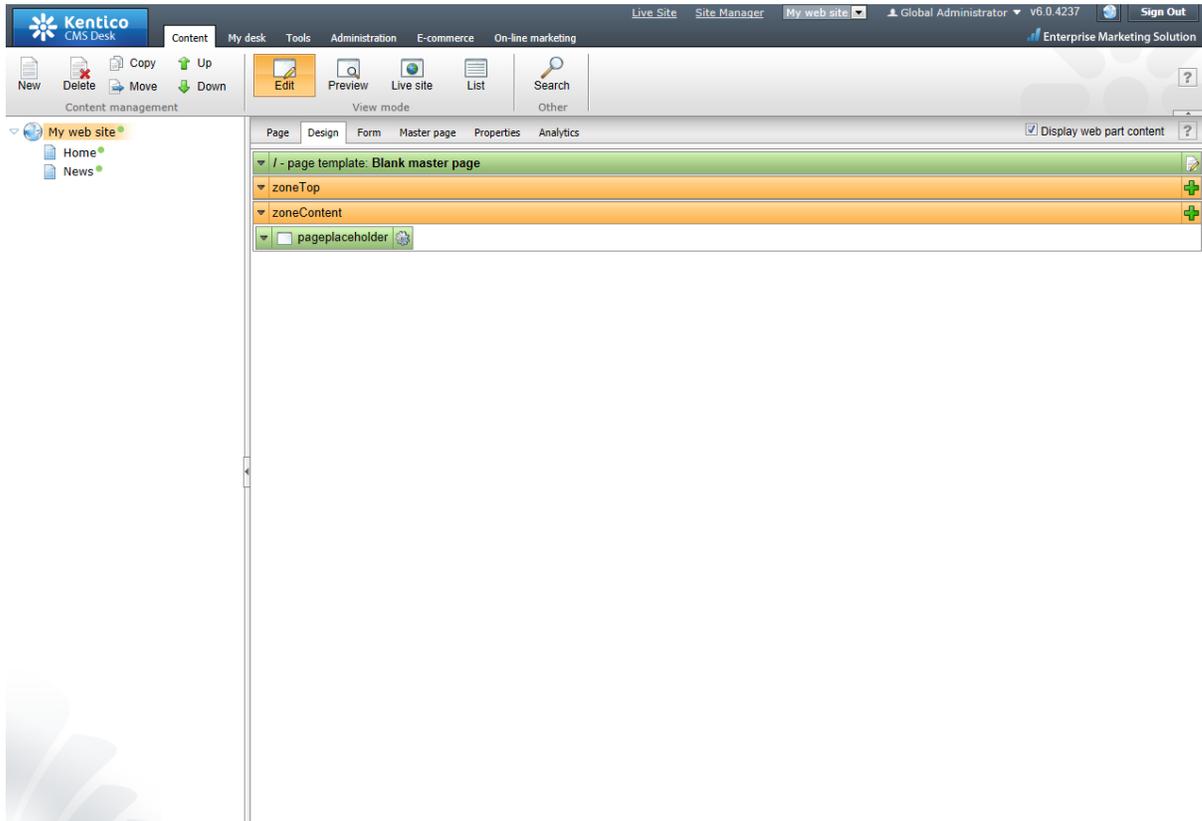
which means that you need to link to files in the **App_Themes** folder like in the example below:

```
../app_themes/mysite/images/imagenam.gif
```

8.4 Master page

The master page allows you to define content that can be shared by all pages on the website, such as a header, navigation menu and footer. The site's root document represents the main master page.

1. Open **CMS Desk**, select the root of the content tree (*My website*) and switch to the **Design** tab. The default design uses a blank page template with two web part zones.



2. Right-click the **pageplaceholder** web part in the bottom zone and select **Remove**. The *Page placeholder* is an important master page component, but we will add it again once the final page layout of the master page is prepared.

3. Click **Edit layout** (🔧) in the page template header (green box) to modify the template's *page layout*.

4. Clear the default layout content. Open the sample **home.htm** file (in `C:\Program Files\Kentico CMS\<version>\CodeSamples\SampleWebTemplate`) and copy the HTML code from inside the `<body>...</body>` tags. Paste this code into the page layout editor.

Page template layout properties

Layout element: Web part zone Insert Layout type: ASCX

```
<%@ Control Language="C#" ClassName="Simple" Inherits="CMS.PortalControls.CMSAbstractLayout" %>
<%@ Register Assembly="CMS.PortalControls" Namespace="CMS.PortalControls" TagPrefix="cms" %>

<div class="MainDiv">

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

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

<!-- main content -->
<table style="width:100%;height:500px;border: 0px">
  <tr valign="top">
    <!-- left column -->
    <td style="width:280px" class="HomePageLeftColumn">
</td>
    <!-- center column -->
    <td style="padding: 3px 5px 0px 5px;width:450px;">
      <!-- center box -->
      <table cellpadding="0" cellspacing="0" border="0" class="ContainerWithCorners" width="100%">
        <tr class="ContainerWithCornersRow">
          <td class="ContainerWithCornersTopLeft">&nbsp;</td>
          <td class="ContainerWithCornersTop">&nbsp;</td>
          <td class="ContainerWithCornersTopRight">&nbsp;</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

Add CSS styles

Close

5. Click **Save** and **Close** the window.

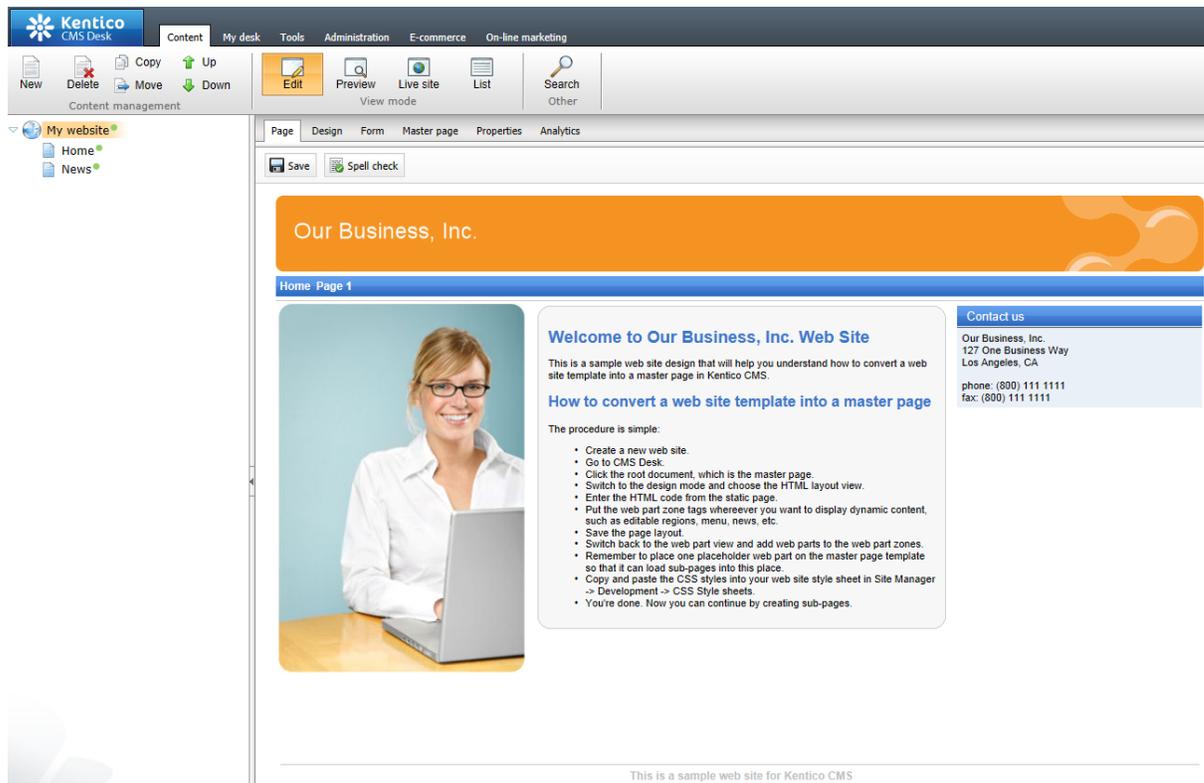


Using CSS-based layout instead of tables

If you prefer using a CSS-based layout, you can modify the HTML code here and replace the tables with other elements (<div>, , etc.). The examples use a table-based layout by default since it is easier to understand.

6. Switch to the **Page** tab.

Here you can view the new website design. For now, all content is entirely static and determined by the page layout code.



Adding web part zones

To make the master page editable through the portal engine, you need to replace the static HTML code in the page layout with web part zones.

1. Return to the **Design** tab and click **Edit layout** (🔗).
2. Locate the `<!-- main menu -->` section in the layout code and delete the entire table inside the `<div class="MainMenu">` element.
3. Place your cursor inside the `<div class="MainMenu">` element and click the **Insert** button above the editor. This adds a web part zone control into the layout.
4. Change the value of the web part zone's **ZoneID** property to `zoneMenu`. The main menu section should now match the following code:

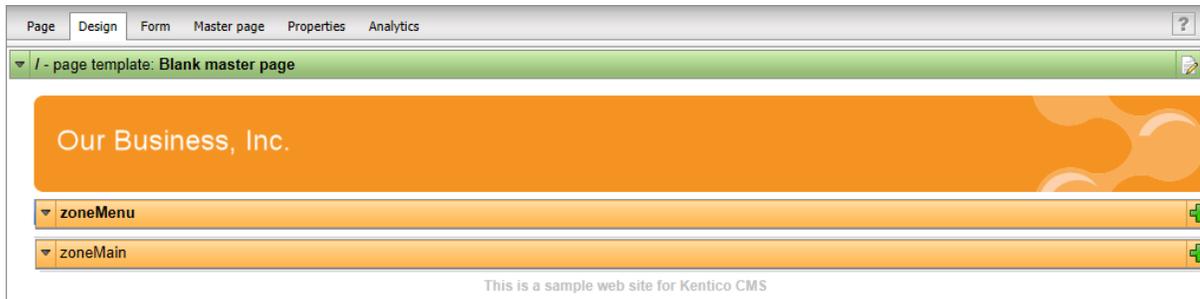
```
<!-- main menu -->
<div class="MainMenu">
    <cms:CMSWebPartZone ZoneID="zoneMenu" runat="server" />
</div>
```

5. Delete the entire content section between the `<!-- main content -->` and `<!-- /main content -->` lines and **Insert** another web part zone instead.
6. Set the second web part zone's **ZoneID** to `zoneMain`:

```
<!-- main content -->
    <cms:CMSWebPartZone ZoneID="zoneMain" runat="server" />
<!-- /main content -->
```

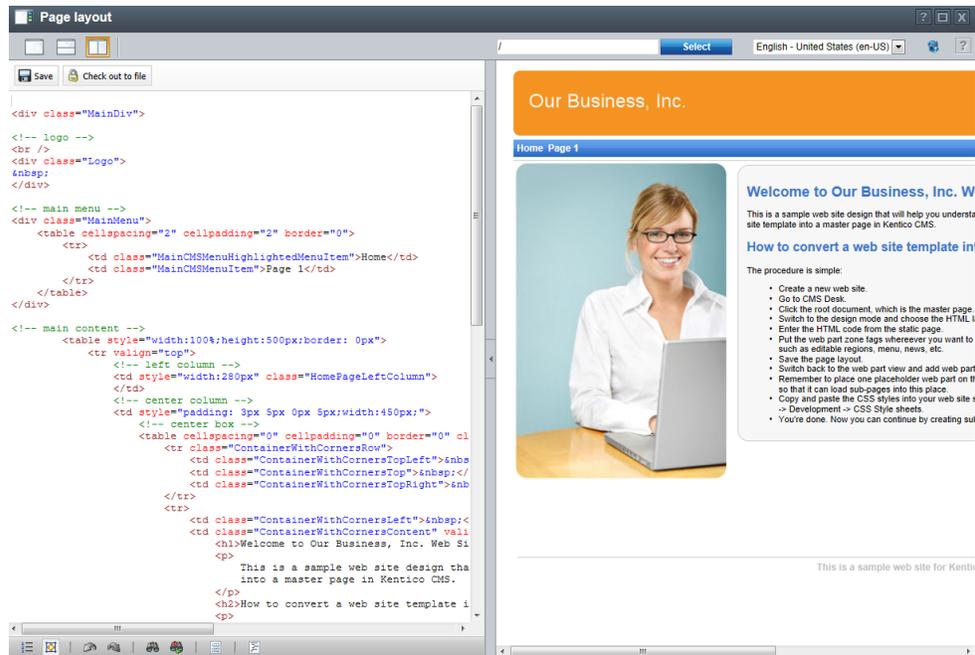
7. Click  **Save** and **Close** the window.

The **Design** tab now shows the new layout with web part zones.



Tip 1: Previewing the layout

By clicking the  **Preview** button in the header of the layout editing dialog, you can modify the code side-by-side with a preview of how the changes affect the live site version of the page.





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's layout (this is the same code that you edited on the **Design** tab).

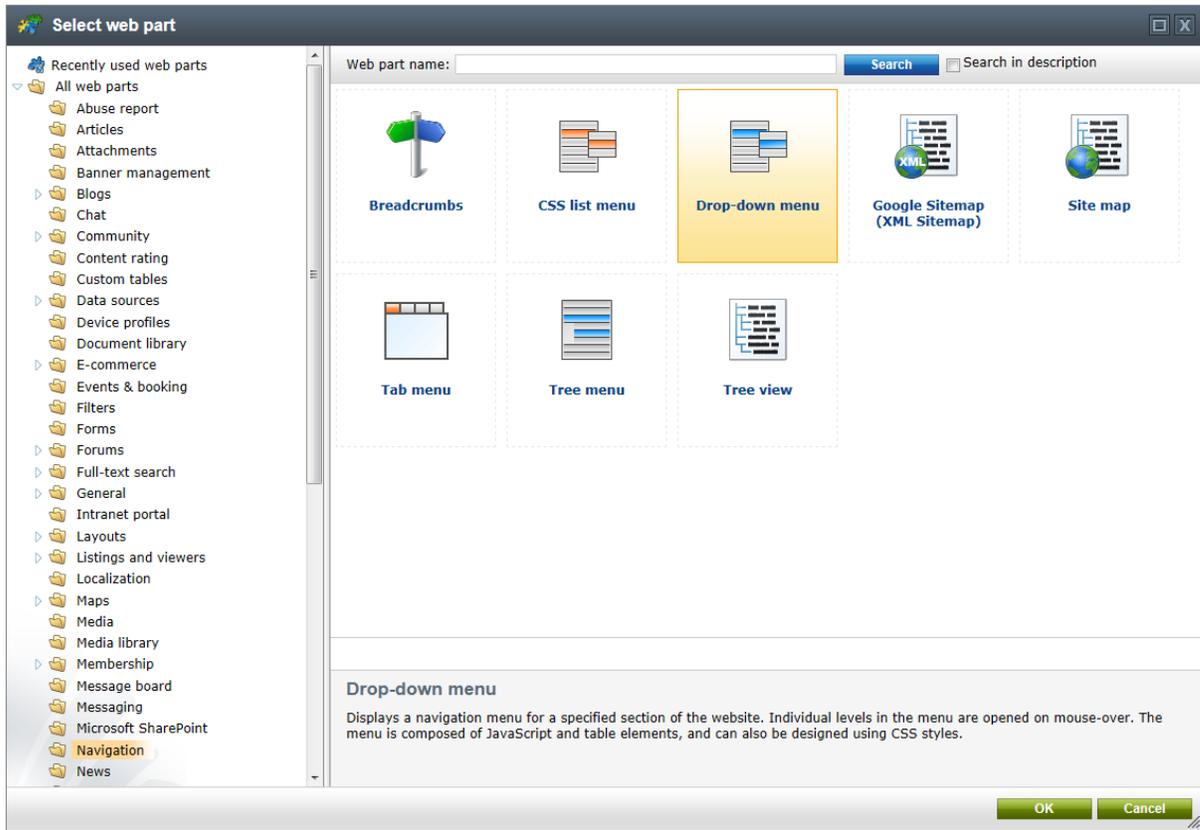
The screenshot shows the 'Master page' tab in the Kentico CMS interface. The top navigation bar includes 'Page', 'Design', 'Form', 'Master page', 'Properties', and 'Analytics'. Below the navigation bar are buttons for 'Save', 'Edit template properties', and 'Preview'. The main area displays the following HTML code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1-
<html>
<head>
  <title>
  - /</title>
  <meta http-equiv="pragma" content="no-cache" />
  <link href="/KenticoCMS_7.0.4490/CMSPages/GetResource.ashx?stylesheetname=CorporateSite" type="text/cs
  <link href="/KenticoCMS_7.0.4490/CMSPages/GetResource.ashx?stylesheetfile=/KenticoCMS_7.0.4490/App_Th
</head>
<body class="LTR IE IE9 ENUS ContentBody">
  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">
    &nbsp;
    </div>
    <!-- main menu -->
    <div class="MainMenu">
      <ccl:CMSWebPartZone ID="zoneMenu" runat="server" />
    </div>
    <!-- main content -->
    <ccl:CMSWebPartZone ID="zoneMain" runat="server" />
    <!-- /main content -->
    <!-- footer -->
    <div class="Footer">
      This is a sample web site for Kentico CMS
    </div>
  </div>
</body>
</html>
```

8.5 Main menu

Now we will add a dynamic menu to the [master page](#).

1. Select the root document (*My website*) in the CMS Desk content tree and open the **Design** tab.
2. Click **Add (+)** in the corner of the **zoneMenu** zone.
3. Select the **Navigation -> Drop-down menu** web part.



4. Click **OK**. The **Web part properties** dialog opens.

5. Set the following properties of the web part (in the **Design** section):

Property	Value	Description
CSSPrefix	;sub	Allows you to add prefixes before the names of the CSS classes applied to the menu. The <i>;sub</i> value uses unmodified class names for the main (first) menu level and the sub prefix for the second level and all other sub-levels.
Layout	Horizontal	Sets a horizontal layout for the menu.

6. Click **OK**.

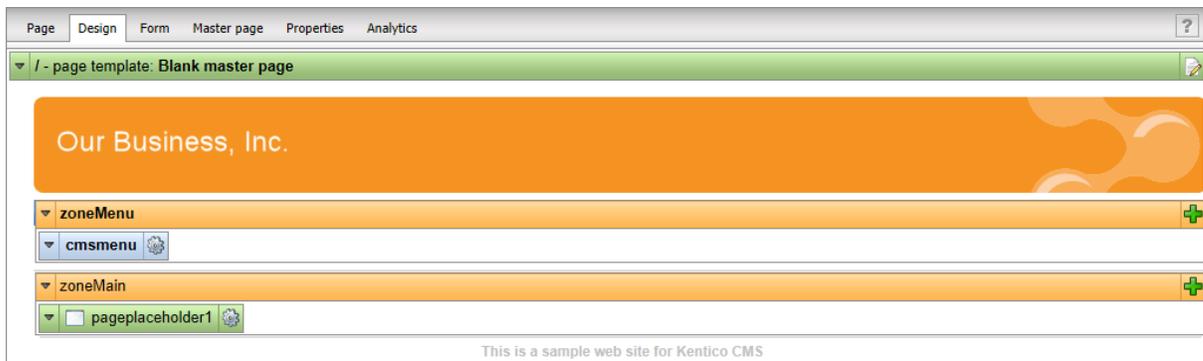
If you switch to the **Page** tab, the page now contains a dynamic menu that displays the child pages: **Home** and **News**.



8.6 Page placeholder

Now you need to add a **Page placeholder** web part that ensures the loading of child pages (such as Home or News) inside the master template.

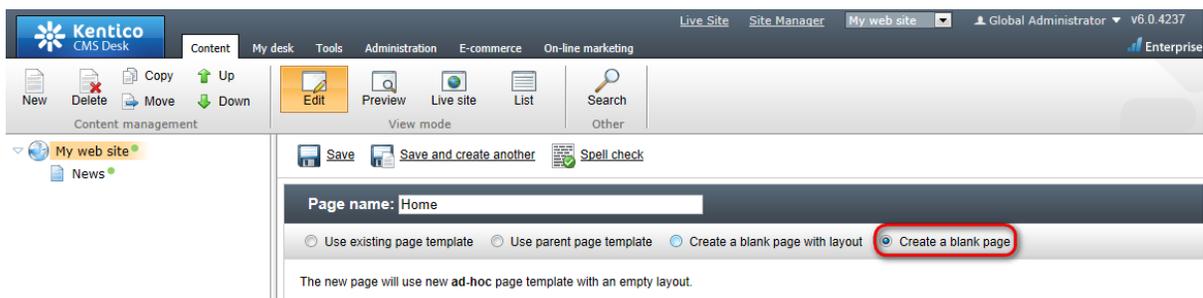
1. Select the root document (*My website*) in the CMS Desk content tree and open the **Design** tab.
2. Click **Add (+)** in the corner of the **zoneMain** zone.
3. Select the **General -> Page placeholder** web part and click **OK**.
4. Click **OK** again. You do not need to make any changes to the page placeholder settings.



8.7 Home page

This topic describes how to create the home page of the website.

1. Select the current **/Home** page and click **Delete** in the main CMS Desk menu. Confirm the deletion of the page in the **Delete document** dialog.
2. Select the root document (*My website*) and click **New**.
3. Choose the **Page (menu item)** document type.
4. Type in *Home* as the **Page name** and choose the **Create a blank page** option.



Click **Save** to create the page.

5. Switch to the **Design** tab and click **Edit layout** (.
6. Replace the default code with the following:

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

Click  **Save** to apply the new layout and **Close** the window.

7. Add the **Text & Images -> Editable text** web part into the **zoneCenter** web part zone. Set its properties:

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

8. Add another **Editable text** web part into the **zoneRight** web part zone. Set its properties:

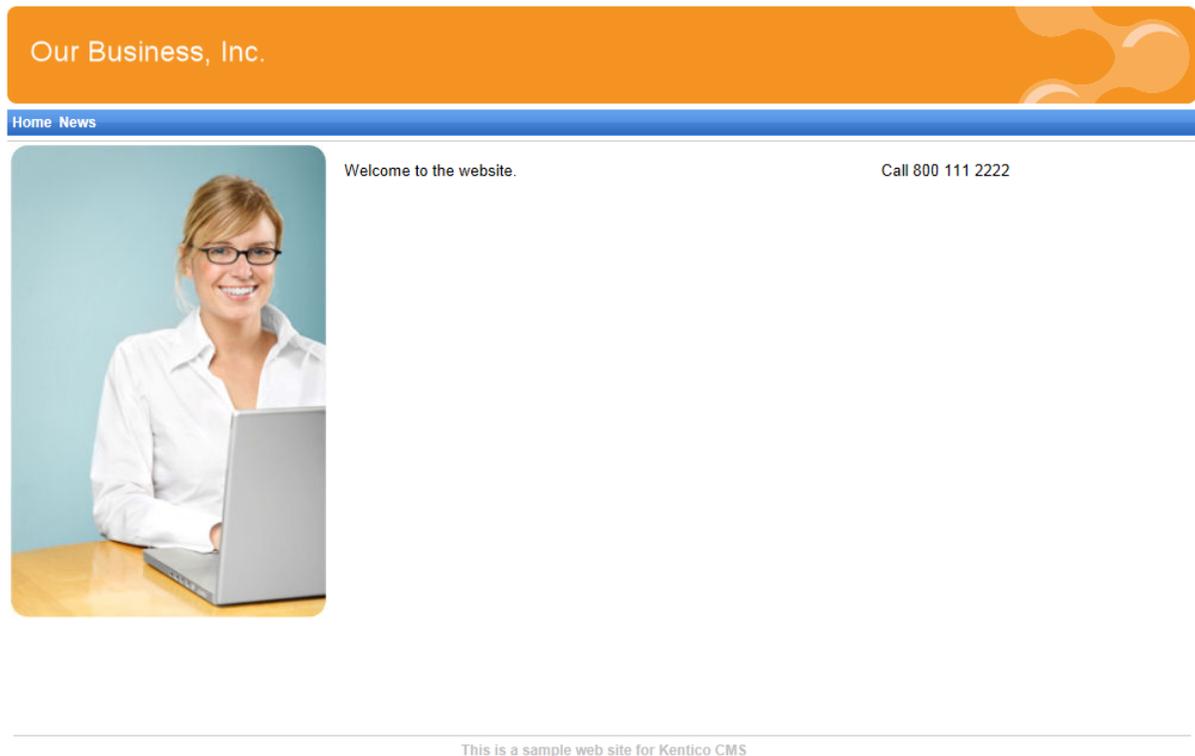
- **Editable region title:** Contact text
- **Editable region height:** 100

9. Open the **Page** tab. The page now contains two editable regions. Enter the following text:

- **Main text:** Welcome to the website.
- **Contact text:** Call 800 111 2222

Click  **Save**.

To view the new Home page of your website, switch to **Live site** mode.



Configuring the website's home page

When a visitor arrives on the root URL of the website (i.e. its domain name, for example *http://www.example.com*), the system needs to know which page to display as the home page.

To set the path of the website's default home page:

1. Go to **Site Manager -> Settings** and click the **Content** category in the settings tree.
2. Select **My website** in the **Site** drop-down menu.
3. Uncheck the **Inherit from global settings** box next to the **Default alias path** setting and type in */Home*, which is the alias path of your new home page.

The screenshot shows the Kentico Site Manager interface. The 'Site' dropdown is set to 'My website'. The 'Content' settings page is displayed, showing various configuration options. The 'Page not found' section includes settings for non-published documents, URL, and exceptions. The 'Multilingual' section includes settings for default content culture, combining with default culture, and combining files with default culture. The 'Web site content' section includes a setting for the default alias path, which is highlighted with a red box and set to '/Home'.

4. Click **Save**.

When visitors access the website without specifying the URL of a particular page, the system automatically displays the Home page.

8.8 Container for home page text

Now we will create gray and blue containers for our home page text. You can reuse containers for any other web parts later. If you do not wish to use web part containers, you can alternatively insert the surrounding code directly into the HTML layout of your page template.

Creating the web part containers

1. Go to **Site Manager -> Development -> Web part containers** and click **New container**.

2. Enter the following values:

- **Display name:** My website gray box
- **HTML code:**

```
<table cellpadding="0" cellspacing="0" border="0" class="ContainerWithCorners"
width="100%">
  <tr class="ContainerWithCornersRow">
    <td class="ContainerWithCornersTopLeft">&nbsp;</td>
    <td class="ContainerWithCornersTop">&nbsp;</td>
    <td class="ContainerWithCornersTopRight">&nbsp;</td>
  </tr>
  <tr>
    <td class="ContainerWithCornersLeft">&nbsp;</td>
    <td class="ContainerWithCornersContent" valign="top">
      ?
    </td>
    <td class="ContainerWithCornersRight">&nbsp;</td>
```

```

</tr>
<tr class="ContainerWithCornersRow">
  <td class="ContainerWithCornersBottomLeft">&nbsp;</td>
  <td class="ContainerWithCornersBottom"></td>
  <td class="ContainerWithCornersBottomRight">&nbsp;</td>
</tr>
</table>

```



Working with the web part placeholder

The "?" character in the code above determines the position of the web part placeholder when you paste it into the **HTML code** field.

3. Click  **Save**.
4. Switch to the **Sites** tab and assign the container to **My website**.
5. Return to the main container list and create another container with the following values:
 - **Display name:** My website blue box
 - **HTML code:**

```

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

```

6. Click  **Save** and assign the container to **My website** on the **Sites** tab.

Assigning the containers to web parts

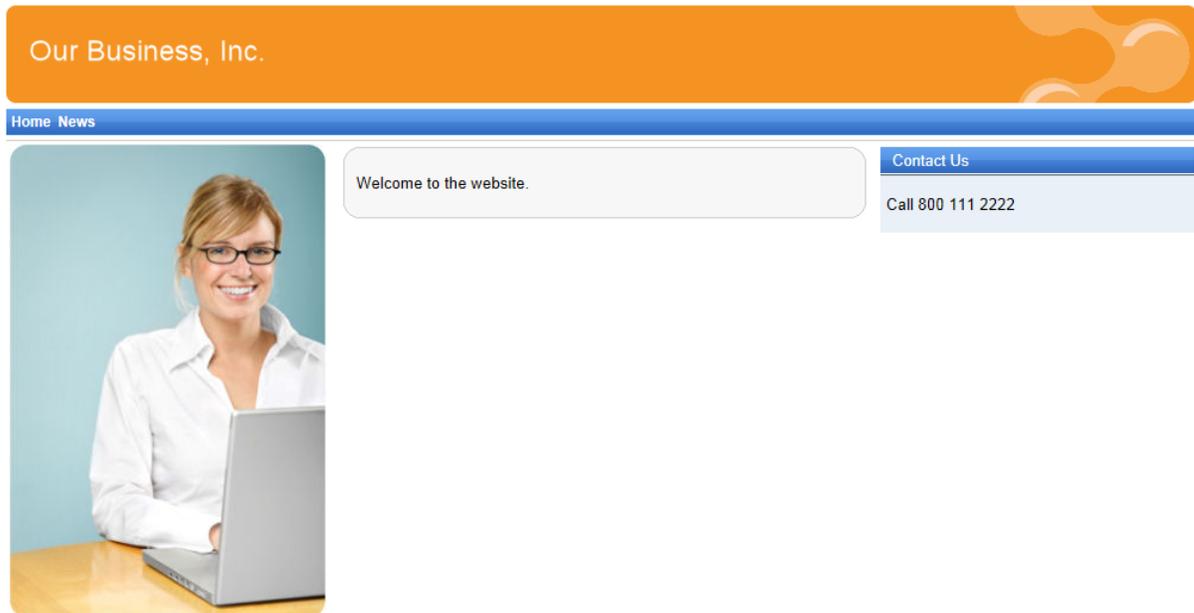
Now that the containers are prepared, you can assign them to specific instances of web parts.

1. Go back to **CMS Desk -> Content**, select the **Home** page and open the **Design** tab.
2. Configure () the **editabletext** web part in the **zoneCenter** zone.
3. Scroll down to the **Web part container** property and select *My website gray box*. Click **OK**.
4. Configure the **editabletext1** web part in the **zoneRight** zone and set the following properties in the **Web part container** section:

- **Web part container:** My website blue box
- **Container title:** Contact Us

5. Click **OK**.

If you now view the Home page in **Live site** mode, you can see the gray box around the welcome text and the blue box around the contact us text.



8.9 News page

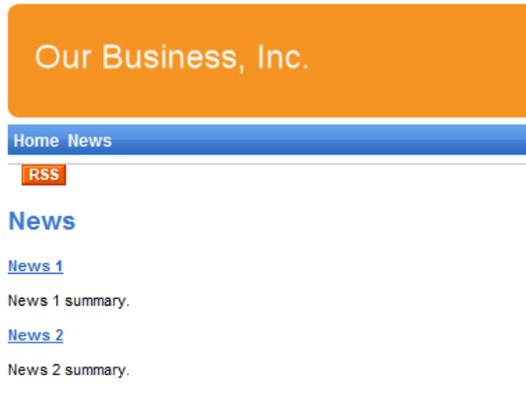
Now we will create the News section of the website.

1. Select the **/News** document in the CMS Desk content tree. The News page itself is already prepared, since it uses the predefined **News list** page template.
2. Click **New** in the main CMS Desk menu and choose the **News** document type.
3. Fill in the news document fields with the following values:
 - **News title:** News 1
 - **Release date:** click Now
 - **News summary:** News 1 summary.
 - **News text:** News 1 text.
 - **Publish from/to:** leave the fields blank
4. Click  **Save and create another** and enter the following values:
 - **News title:** News 2
 - **Release date:** click Now
 - **News summary:** News 2 summary.
 - **News text:** News 2 text.

- **Publish from/to:** leave the fields blank

5. Click  **Save**.

If you view the `/News` page in **Live site** mode, you can see a list of all news documents placed under the **News** section.



This is an example of how content is logically structured in Kentico CMS. If you select a specific news item, the page displays the detail view.



The breadcrumbs at the top of the page show the current path on the website: **News > News 1**. The position is also reflected in the default page URLs:

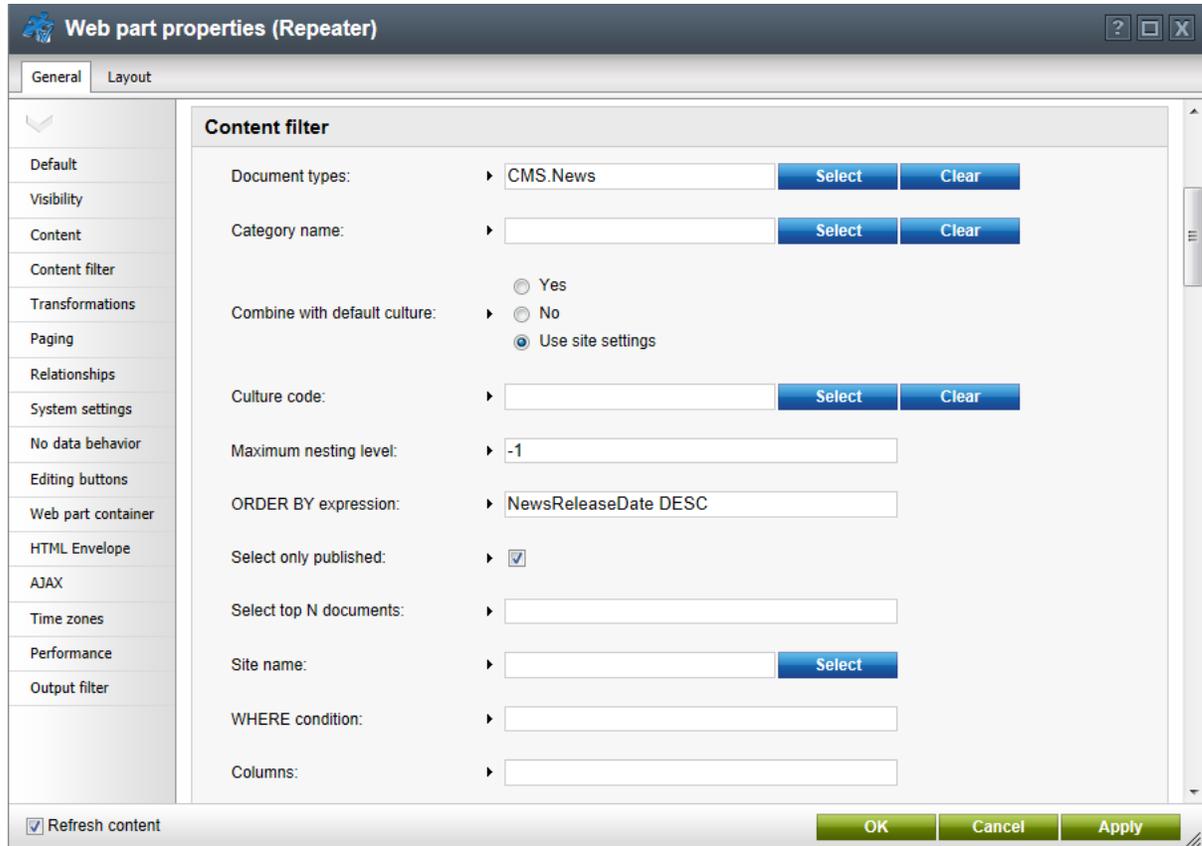
- The URL of the News page is `~/news.aspx`
- The URL of the News 1 page is `~/news/news-1.aspx`

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

How it works

You may be wondering how Kentico CMS generates the news list. The news page is a good example of using web parts to display structured document data on the website.

Select the **News** document and open the **Design** tab. The page contains several web parts, including the **NewsRepeater**. This web part provides various types of additional functionality, but its core is based on the standard ASP.NET Repeater control that you may already be familiar with. By configuring (🌐) the web part, you can view the settings of the properties.



The most important properties are the following:

Property	Value	Description
Path		Determines the path of the documents in the content tree from which the web part loads and displays data. If you leave the value empty, the web part searches for all documents under the current page.
Document types	cms.news	This property determines what types of documents the web part displays (news, products, blogs etc.).
Transformation	cms.news.pre view	Assigns the transformation that the web part uses to display the list of news items. Transformations work similar to the <i>ItemTemplate</i> element that you would use with a standard Repeater control.
Selected item transformation	cms.news.default	When a user selects a specific news item on the website, the repeater displays the details according to the specified transformation.

The following steps show how the page handles visitors:

1. A visitor arrives on the **/News** page.
2. The **Repeater** web part placed on the page template checks if a news document is currently selected (based on the value of the **Document types** property).
3. The web part finds out that the current document is a *page (menu item)*, so it looks for all underlying news documents and displays them as a list using the **cms.news.preview** transformation.
4. When the visitor selects a particular news item, such as **/News/News 1**, the repeater web part uses the **cms.news.default** transformation instead to display the details.



Path expressions

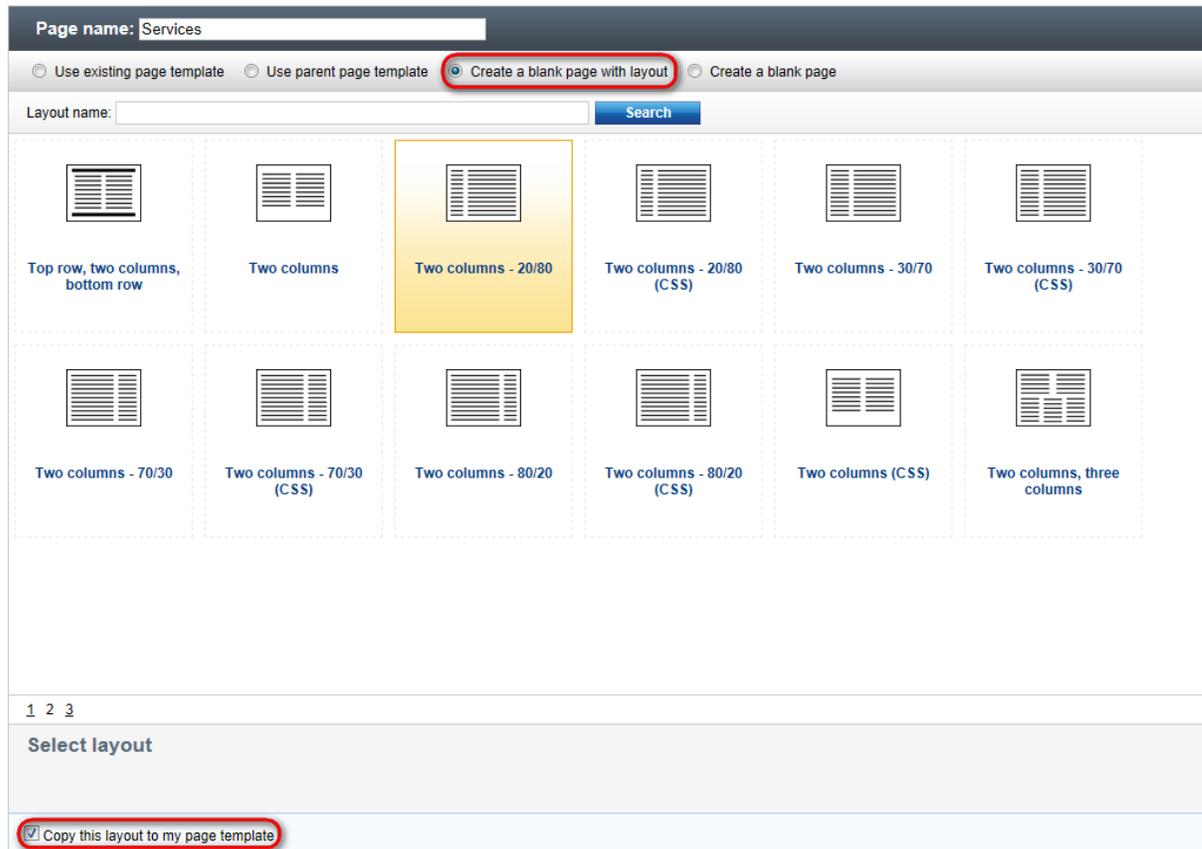
Listing web parts have the **Path** property that specifies which content the component loads and displays. This property supports the following expressions that allow you to select content dynamically:

/%	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 documents under the Images document, which is a child of the current document.
../contacts/%	All documents under the Contacts document on the same content level as the current document.
/{0}/%	<p>All documents under the document located on the first level of the current path.</p> <p><u>Example:</u></p> <p>If the currently selected document is:</p> <p><i>/news/news1</i></p> <p>the system evaluates the expression as:</p> <p><i>/news/%</i></p>

8.10 Services page

Now we will add a website section displaying information about services. The page template used for this section will contain a tree menu on the left and a single editable region.

1. Select the root document in the content tree (*My website*) and click **New** in the main CMS Desk menu.
2. Choose the **Page (menu item)** document type.
3. Type in *Services* as the **Page name** and choose the **Create a blank page with layout** option. Select the **Two columns - 20/80** layout with the **Copy this layout to my page template** box checked.



4. Click **Save** to create the page.
5. Open the **Design** tab and add the **Navigation -> Tree menu** web part into the **zoneB** zone (left zone on the middle row).
6. Set the following properties for the **Tree menu** web part:

Property	Value	Description
Content -> Path	/{0}/%	Configures the tree menu to display documents starting from the second level of the currently selected path.
Design -> Item image URL	~/app_themes/mysite/images/bullet.gif	Sets the path of the image that the menu displays next to items. The ~ character represents the root of the website. This relative path ensures that the web part displays the images correctly even if the website's virtual directory

		name changes.
Design -> Open item image URL	~/app_themes/mysite/images/bullet.gif	Specifies the image displayed next to items in the tree menu that belong on the path of the currently selected document.

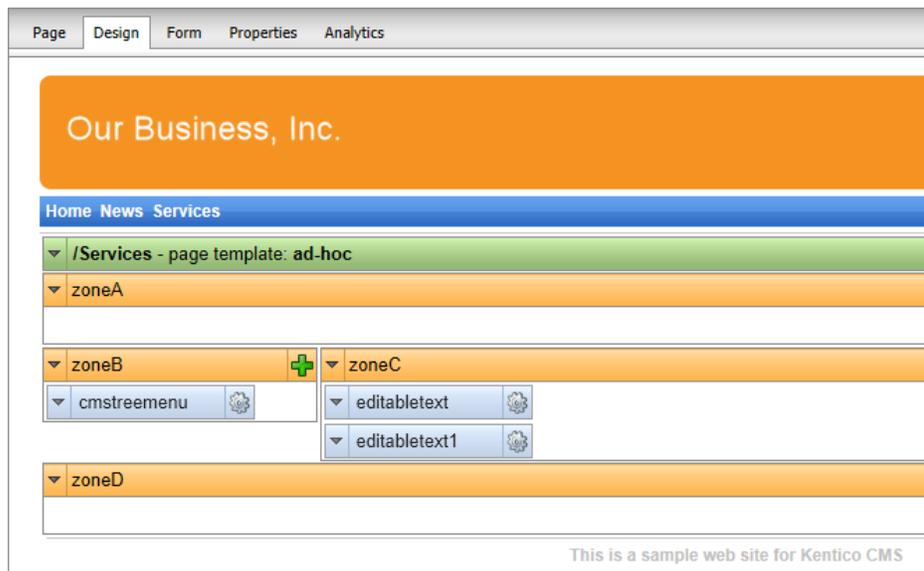
7. Add the **Text & Images -> Editable text** web part into the **zoneC** zone (right zone on the middle row) and set the following properties:

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

8. Add another **Editable text** web part below the first one and set the following properties:

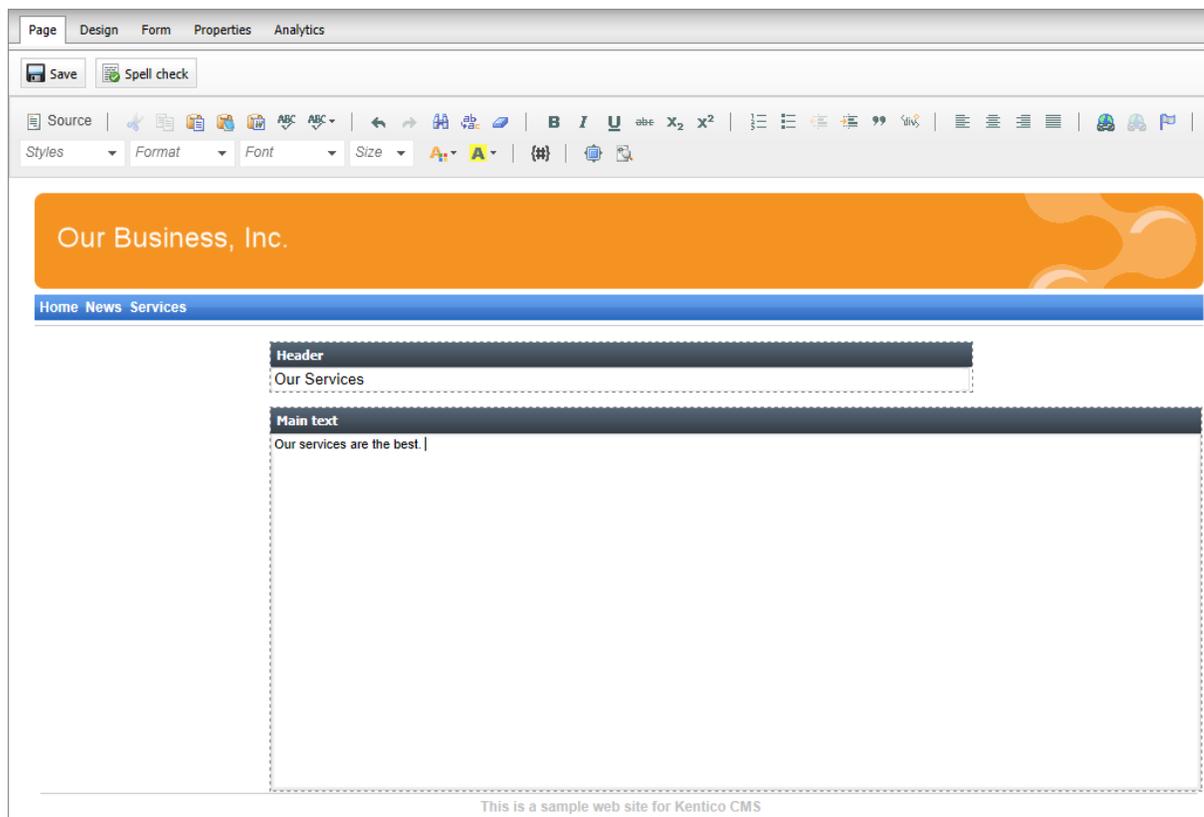
- **Editable region title:** Main text
- **Editable region type:** HTML Editor
- **Editable region height:** 300

The Design tab of the Services page should now look like this:



Entering text

1. Switch to the **Page** tab of the Services page and type some text into the editable regions.



2. Click  **Save**.

Creating a re-usable page template

Now we will save the current state of the page as a re-usable page template.

1. Go to the **Properties -> Template** tab of the *Services* document and click  **Save as new page template**.

2. Fill in the following values for the new template:

- **Template display name:** Services template
- **Template category:** General
- **Assign to the current document:** yes (checked)

3. Click **OK**.

You can now select the *Services template* directly when creating new pages.

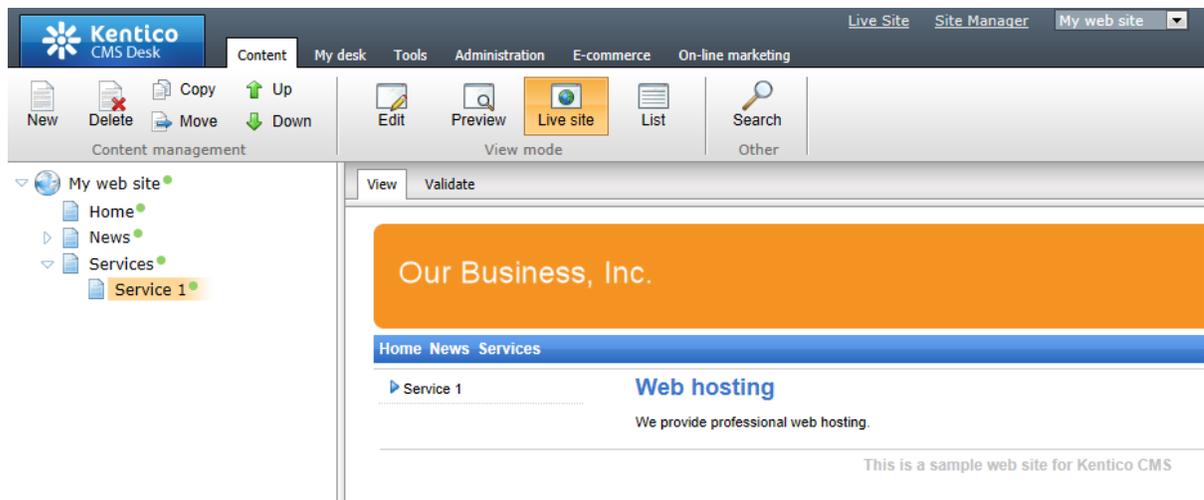
Adding sub-pages

Now use the *Services template* to add separate pages containing information about individual services under the **Services** page.

1. Select the **/Services** page in the content tree and click **New** on the main toolbar.

2. Choose the **Page (menu item)** document type.
3. Type in *Service 1* as the **Page name** and choose the **Use existing page template** option. Select the **General** category and the **Services template** page template.
4. Click  **Save**.
5. Type the following text into the editable regions on the **Page** tab:
 - **Header:** Web hosting
 - **Main text:** We provide professional web hosting.
6. Click  **Save**.

To see how the *Service 1* page appears to website visitors, switch to **Live site** mode.



The Services section demonstrates how to create a new page template from scratch and re-use it for any number of pages. The system stores the text of editable regions separately for every document, so you can enter unique content on each page.

8.11 Products page

8.11.1 Overview

Now we will add a products section displaying a list of computers and their technical specification. You will learn how to:

1. Create a [new document type](#) representing computer products
2. Write [transformations](#)
3. [Create a page](#) displaying a list of computer products

8.11.2 New document type

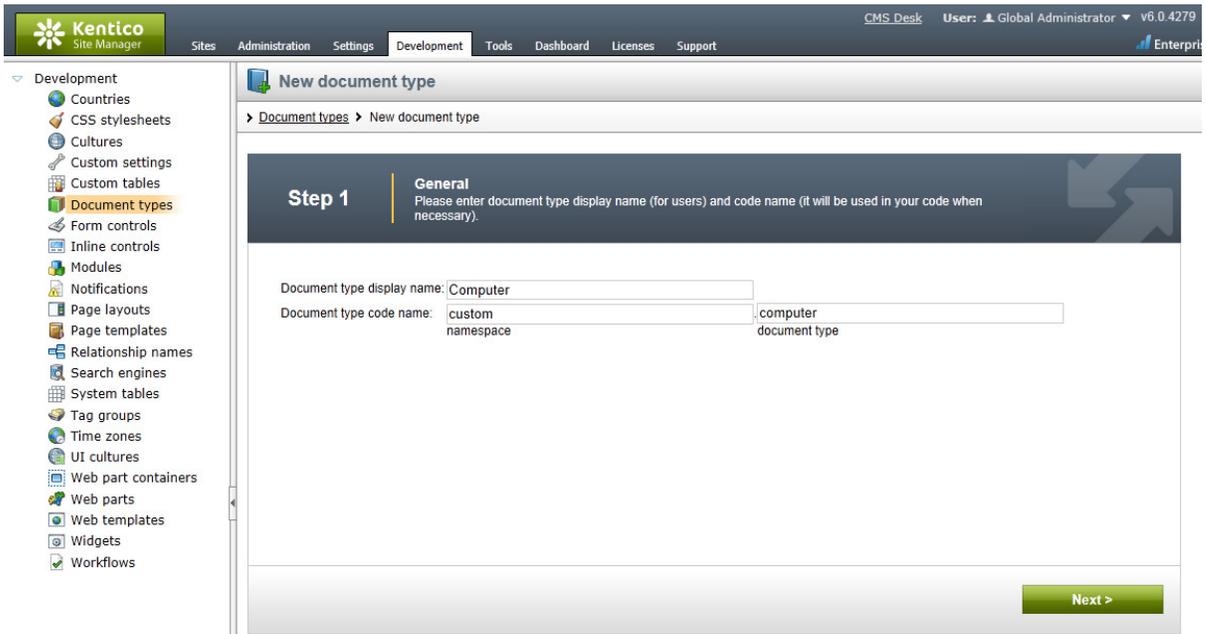
Each document in the Kentico CMS repository is of a certain type, such as news, product, article, etc. Every document type has its own data fields. Our document type describing computer products will have

fields storing the computer name, processor type, RAM size, disk size and product image.

1. Go to **Site Manager -> Development -> Document types** and click  **New document type**. This starts the **New document type** wizard.

2. Enter the following values in **Step 1**:

- **Document type display name:** *Computer* (the system displays this name to users in the administration interface)
- **Document type code name:** *custom.computer* (*custom* is a namespace to distinguish your document types from the default system types that use the *cms* namespace, *computer* is the identifier of the document type. You will use this value in the properties of controls later.)



The screenshot shows the Kentico Site Manager interface. The top navigation bar includes 'Sites', 'Administration', 'Settings', 'Development', 'Tools', 'Dashboard', 'Licenses', and 'Support'. The user is logged in as 'Global Administrator' on 'CMS Desk' version 'v6.0.4279'. The left sidebar lists various development options, with 'Document types' highlighted. The main content area is titled 'New document type' and shows 'Step 1: General'. The instructions for this step are: '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' containing 'Computer' and 'Document type code name' containing 'custom.namespace.computer.document.type'. A green 'Next >' button is located at the bottom right of the form.

Click **Next**.

3. In **Step 2**, you need to specify the name of the database table where the system will store the data of computer documents. You also need to enter the name of the table's primary key field. 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 creates a new database table for computer documents.

4. In **Step 3**, you need to define the fields of the document type (columns of the table). Click **New attribute** (+) to create the following fields. For each field, enter the values, click  **Save** 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:** Text box

- **Column name:** ComputerProcessorType
- **Attribute type:** Text
- **Attribute size:** 200
- **Field caption:** Processor type
- **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:** Text box

- **Column name:** ComputerHddSize

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

- **Column name:** ComputerImage
- **Attribute type:** File
- **Allow empty value:** yes (checked)
- **Field caption:** Image
- **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

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: Upload file
Field description:

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

Next >

Click **Next**.

5. In **Step 4**, choose the *ComputerName* field as the **Document name source**.

This means that when a user creates a new computer document, the system automatically fills in the document name based on the ComputerName value. The document name appears 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**.

6. In **Step 5**, select the document types that will be supported as parents for computer documents in the content tree. Click **Add document types**, select the **Page (menu item)** document type and click **OK**.

This means that users are only allowed to place computer documents under pages, not under articles, news items or other document types.

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**.

7. In **Step 6**, assign the document type to all websites where you wish to use it. Click **Add sites**, choose **My website** in the selection dialog and click **OK**.

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**.

8. **Step 7** of the wizard allows you to specify how the Smart search module indexes documents of this type and displays them in search results. Select the following values in the drop-downs:

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

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 >

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

9. Click **Finish** to complete the creation of the new document type.

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

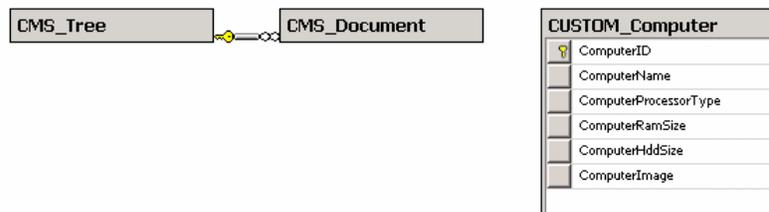
The wizard automatically creates not only the database table, but also the SQL queries for SELECT, INSERT, UPDATE, DELETE operations and a several default transformation.



How does the system store document content?

The CMS stores document content and all related data in three database tables:

- **CMS_TREE** (content tree structure)
- **CMS_Document** (general document properties, metadata and editable region content)
- The dedicated document type table - in this case **CUSTOM_Computer** (stores the values of the document type's specific fields)



The system automatically ensures all operations related to these tables. **You can easily write standard SQL SELECT queries to retrieve data** from the repository (i.e. the Microsoft SQL Server database).

You have learned how to create new document types. Continue in the [Transformations](#) topic, which

describes how to define the format used to display document data on the website.

8.11.3 Transformations

Now that you have created the new document type, you need to prepare the transformations that page components will use to display computer products on the website.

1. Go to **Site Manager -> Development -> Document types** and edit (✎) the **Computer (custom.computer)** document type.
2. Switch to the **Transformations** tab.

The screenshot shows the 'Document type properties' window for the 'Computer' document type. The 'Transformations' tab is selected, displaying a table of default transformations. The table has three columns: 'Actions', 'Transformation name', and 'Transformation type'. Each row includes edit and delete icons.

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

The New document type wizard has created several default transformations, which you can use as a base for your own transformations.

3. **Edit** (✎) the **Default** transformation, clear the original code and replace it with the following:

```
<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>

```

ASCX transformation code is similar to standard *ItemTemplate* elements that you may already be familiar with from using ASP.NET Repeater or DataList controls. It combines HTML with ASP.NET commands and data binding expressions (*Eval*). You may also use built-in methods that simplify various tasks, such as **GetImage**. For more information about the available transformation methods, click the [? Transformation examples](#) link above the code editor.

You will use the **Default** transformation for displaying the *details* of individual computer products.

4. Click  **Save**.

5. Return to the transformation list and edit the **Preview** transformation. Clear the default code and add the following code instead:

```

<div style="text-align:center;padding: 8px;margin: 4px;border: 1px solid #CCCCCC">
  <h2>
    <a href="<%# GetDocumentUrl() %>"><%# Eval("ComputerName") %></a>
  </h2>
  <%# GetImage("ComputerImage", 120) %>
</div>

```

Note the code used to create the link to specific documents. It consists of a standard HTML link tag and inserts the appropriate URL and link text dynamically:

```

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

```

You can generate an image tag containing the file uploaded into the given document's **ComputerImage** field using the **GetImage** method. The sample code calls the method with a parameter that ensures automatic server-side resizing of the image's longest side to 120 pixels:

```

<%# GetImage("ComputerImage", 120) %>

```

You will use the **Preview** transformation for displaying the *list* of computer documents on the main

products page.



Entering field names in transformations

When writing ASCX transformations, you often need to specify the names of data fields as parameters of the *Eval* data binding expression or other methods, such as *ComputerName* and *ComputerImage* in the examples above.

You can either type the names manually, or press the **CTRL + SPACE** key combination to access a list of available document fields and related objects.

6. Click  **Save**.

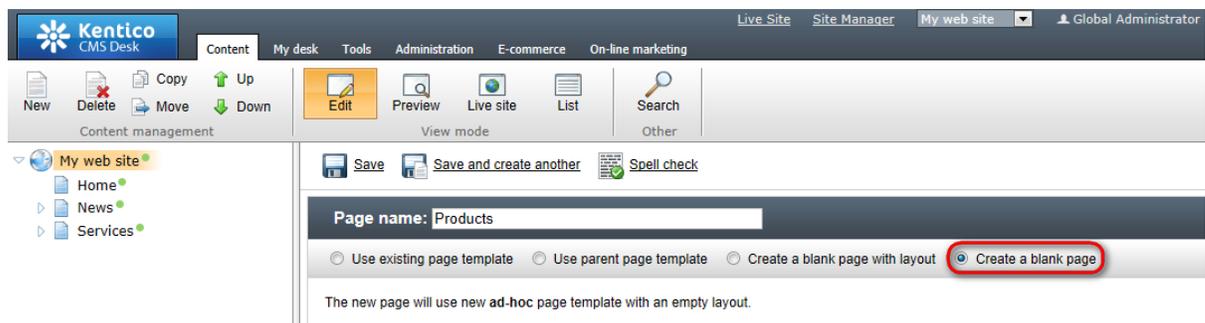
You have learned how to write transformations for displaying the content of structured documents. Continue in the [Page template](#) topic to create a website section that uses the *Preview* and *Default* transformations.

8.11.4 Page template

This topic describes how to add the product list page and publish computer specifications on the website.

Creating the product list page

1. Go to **CMS Desk -> Content**.
2. Select the root document (*My website*) and click **New**.
3. Choose the **Page (menu item)** document type.
4. Type *Products* as the **Page name** and choose the **Create a blank page** option.



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

6. Switch to the **Design** tab and add the **Navigation -> Breadcrumbs** web part into **zoneA**. Leave the default properties for the web part.

7. Add the **Listing and viewers** -> **Datalist** web part below the breadcrumbs. Set the following properties for the web part:

Property	Value	Description
Document types	custom.computer	Configures the datalist to display only documents of the <i>custom.computer</i> type (created in the Defining a new document type topic).
ORDER BY expression	ComputerName ASC	Sets the SQL ORDER BY clause that the web part uses when loading data. As a result, the datalist displays items in ascending alphabetical order based on the values of the <i>ComputerName</i> field.
Transformation	custom.computer.p review	Assigns the transformation that the datalist uses to display the list of computer products (as defined in Writing transformations).
Selected item transformation	custom.computer.d efault	When a user selects a specific computer document on the website, the web part displays the details according to the specified transformation.

The page is now ready to display underlying computer documents.

Adding computer documents

Now add documents representing individual computer products:

1. Select the **Products** page in the content tree, click **New** and choose the **Computer** document type.

2. Fill in the computer document fields with the following values:

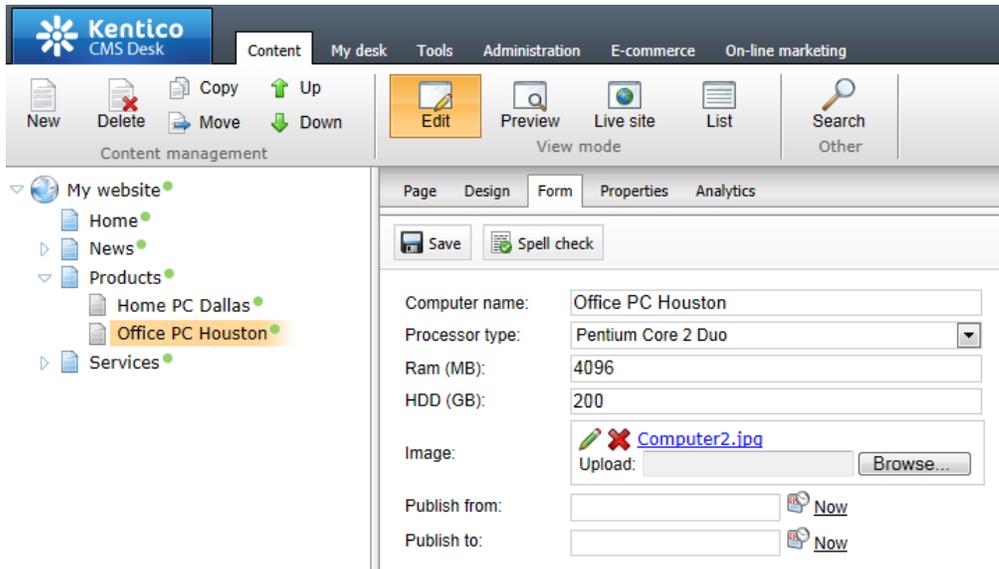
- **Computer name:** Home PC Dallas
- **Processor type:** Athlon
- **RAM (MB):** 2048
- **HDD (GB):** 160
- **Image:** upload an image (you can find sample images in the <Kentico CMS installation>\CodeSamples\SampleWebTemplate\Computer_Images folder)
- **Publish from/to:** leave the values blank

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

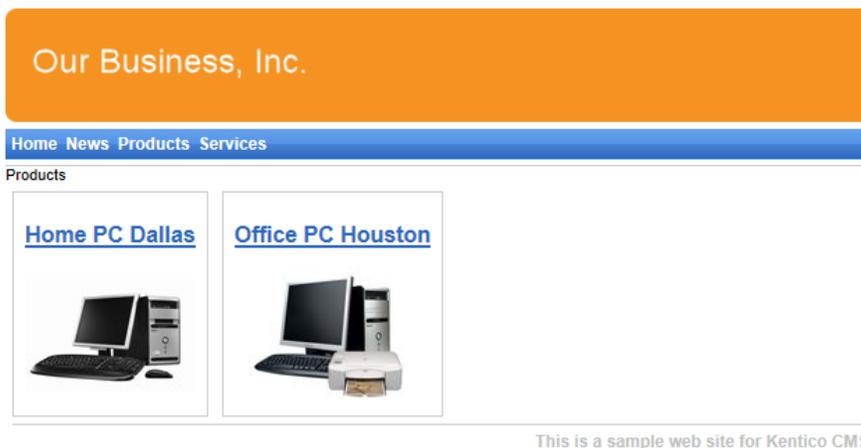
- **Computer name:** Office PC Houston
- **Processor type:** Pentium Core 2 Duo
- **RAM (MB):** 4096
- **HDD (GB):** 200
- **Image:** upload an image (you can find sample images in the <Kentico CMS installation>\CodeSamples\SampleWebTemplate\Computer_Images folder)
- **Publish from/to:** leave the values blank

4. Click  **Save**.

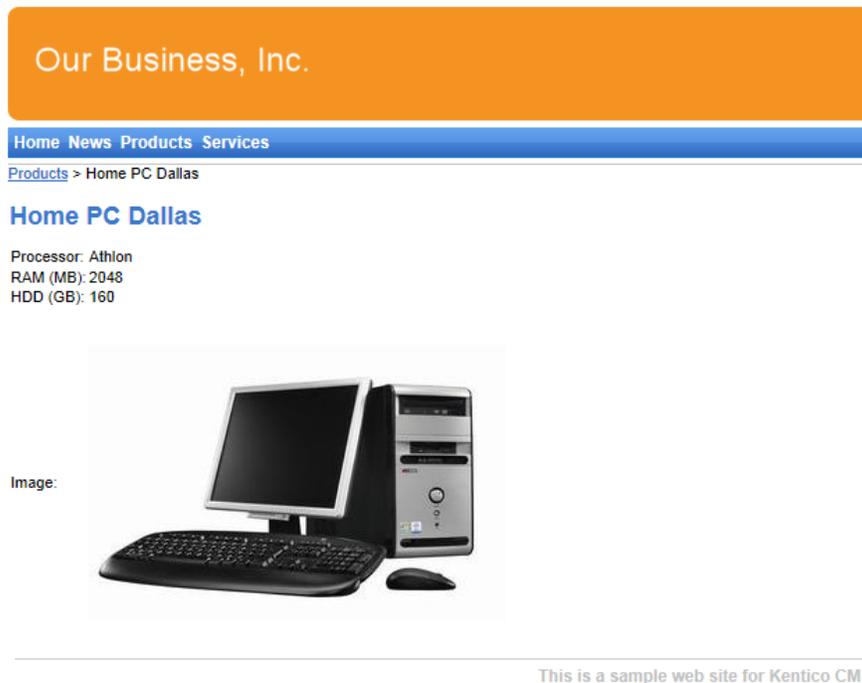
You can edit the field values of the computer documents at any time by switching to the **Form** tab.



If you view the `/Products` page in **Live site** mode, you can see a list of the two computer products (formatted according to the `custom.computer.preview` transformation).



When you click the title of a specific computer, the page displays the detail view (using the `custom.computer.default` transformation).



8.12 Search page

Kentico CMS allows users to perform index-based searches through all document content in the Kentico CMS repository, as well as other types of data. This topic describes how to add a basic search page to your website.

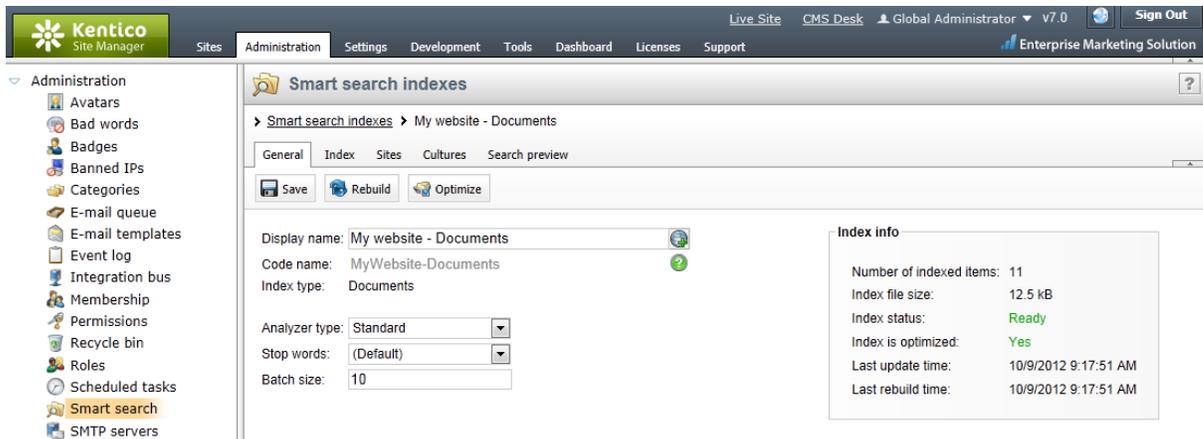
Creating a smart search index

Before you can use the search, you need to add a smart search index covering the website's documents.

1. Go to **Site Manager -> Administration -> Smart search**.
2. Click  **New index**.
3. Fill in the following details for the search index:
 - **Display name:** My website - Documents
 - **Index type:** Documents
 - **Analyzer type:** Standard
 - **Stop words:** Default
4. Click  **Save**. The index's editing interface opens.
5. Open the **Index** tab and click  **Add allowed content**.
6. Type `/%` into the **Path** field and click  **Save**. This ensures that the index includes all documents on the website.

7. Switch to the **Sites** tab and assign the index to **My website**.
8. Switch to the **Cultures** tab and choose the default culture of your site (typically *English - United States*).
9. Open the **General** tab and click  **Rebuild**.

Once the system rebuilds the index, you can start using it on the website. The **Index info** box on the right side of the tab displays the current status of the index and other relevant information.



The screenshot shows the Kentico Site Manager interface. The top navigation bar includes 'Live Site', 'CMS Desk', 'Global Administrator', 'v7.0', and 'Sign Out'. The main menu has 'Administration', 'Settings', 'Development', 'Tools', 'Dashboard', 'Licenses', and 'Support'. The left sidebar lists various administration tasks like 'Avatars', 'Bad words', 'Badges', etc. The main content area is titled 'Smart search indexes' and shows the configuration for 'My website - Documents'. The 'General' tab is active, displaying fields for 'Display name', 'Code name', 'Index type', 'Analyzer type', 'Stop words', and 'Batch size'. An 'Index info' box on the right shows: Number of indexed items: 11, Index file size: 12.5 kB, Index status: Ready, Index is optimized: Yes, Last update time: 10/9/2012 9:17:51 AM, Last rebuild time: 10/9/2012 9:17:51 AM.



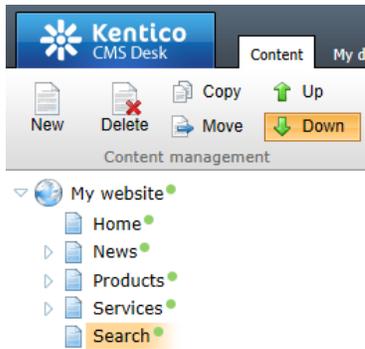
Searching through uploaded text files

You can also configure the CMS to search the text inside uploaded files, such as PDF, DOC or XLS documents. For additional details, see the following chapter in the Kentico CMS Developer's Guide: [Installation and deployment -> Additional configuration tasks -> Configuration of full-text search in files](#)

You do not need to configure this option now, since we will only use the basic document search in this tutorial.

Adding the search page

1. Return to **CMS Desk**. Select the root of the content tree (*My website*) and click **New**.
2. Choose the **Page (menu item)** document type.
3. Type *Search* as the **Page name** and choose the **Use existing page template** option. Select the **General** category and the **Search** page template.
4. Click  **Save** to create the page.
5. Click the **Down** () arrow on the main toolbar to move the **Search** page to the end of the document list.

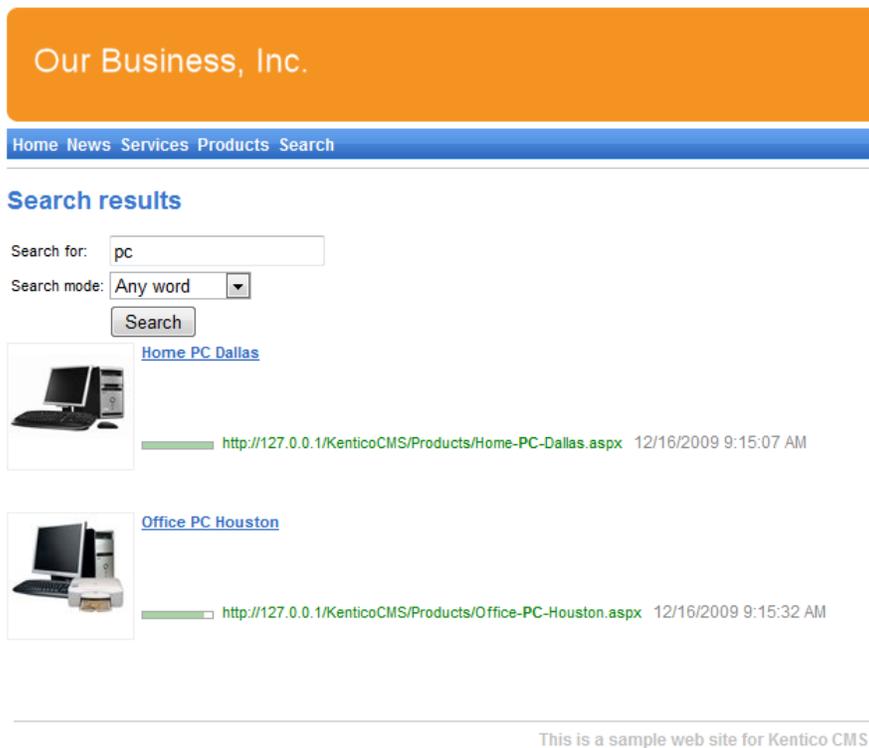


6. Switch to the **Design** tab and **Configure** (⚙️) the **SearchDialogWithResults** web part.

7. Click **Select** next to the **Indexes** property and choose the **My website - Documents** search index created in the previous section.

8. Click **OK**.

To try out the search functionality, view the `/Search` page in **Live site** mode. Type `PC` into the **Search for** box and click **Search**.



When you click a search result, the system redirects you to the corresponding page.

Modifying the format of the search results



If you prefer a different design of the search results, you can modify the format by editing the **SmartSearchResults** (or **SmartSearchResultsWithImages**) transformation in **Site Manager -> Development -> Document types -> Root -> Transformations**.

8.13 Secured section for partners

Kentico CMS provides a way to create secured site sections that can only be viewed by users who have a valid user name and password. This topic describes how to create a logon page for the purposes of user authentication and registration, as well as a secured page accessible only by logged in users.

Adding the secured partners page

Start by adding a new secured page that requires authentication:

1. Go to **CMS Desk -> Content**. Select the root of the content tree (*My website*) and click **New**.
2. Choose the **Page (menu item)** document type.
3. Type *Partners* as the **Page name** and choose the **Use existing page template** option. Select the **Templates with editable regions** category and the **Simple text** page template.
4. Click  **Save** to create the page.
5. Select the **Page** tab and type the following text into the editable region: *This is a secured page for partners*.
6. Click  **Save**.
7. Open the **Properties -> Security** tab of the *Partners* document.
8. Select **Yes** for the **Requires authentication** property in the **Access** section at the bottom of the dialog and click **OK**.

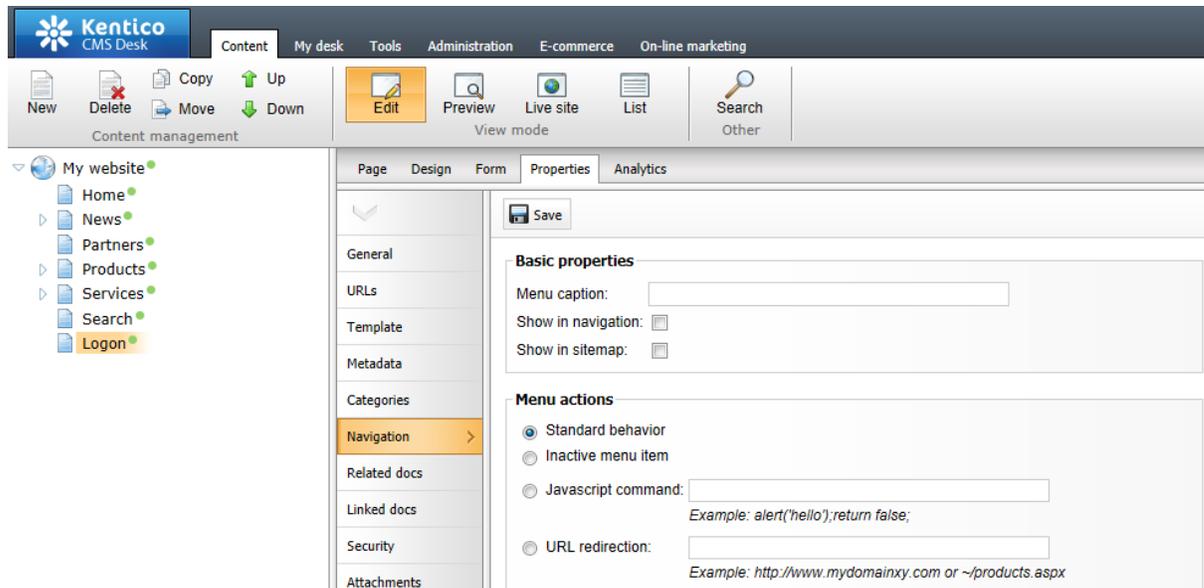
This ensures that only authenticated (logged in) users can access the page.

Creating the logon page

Now we will build a page where users can sign in to the website and anonymous visitors can register as new users. We will use a pre-defined page template that you can further customize as needed.

1. Select the root of the content tree (*My website*) and click **New**.
2. Choose the **Page (menu item)** document type.
3. Type *Logon* as the **Page name** and choose the **Use existing page template** option. Select the **Membership and security** category and the **Log-on page with registration form** page template.

4. Click  **Save** to create the page.
5. Click the **Down** () arrow on the main toolbar until the *Logon* page is at the end of the document list.
6. Open the **Properties** -> **Navigation** tab of the document.
7. Uncheck the **Show in navigation** and **Show in sitemap** boxes.
8. Click  **Save**.



Because of the document's navigation settings, the *Logon* page does not show up in the website's menu. You can this configuration for pages that have a specific purpose on the website, but are not part of the regular content.

Setting the website's logon page

When an anonymous visitor attempts to access a secured page that requires authentication (such as the *Partners* page on your sample website), the system redirects them to a logon page. By default, websites use the system page that appears when signing into CMS Desk. However, you can configure each website to use its own custom logon page.

1. Go to **Site Manager** -> **Settings** and click the **Security & Membership** category in the settings tree.
2. Select **My website** in the **Site** drop-down menu.
3. Uncheck the **Inherit from global settings** box next to the **Website logon page URL** setting and type in `~/Logon.aspx`. This is the relative URL of the logon page that you added to the website.
4. Click  **Save**.

The screenshot shows the Kentico Site Manager 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 left sidebar shows a tree view with 'Security & Membership' selected. The main content area displays the following settings:

- 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

At the bottom, there is an 'Export these settings' link.

The website's logon page is now ready.

Adding a sign out button to the website

The website now allows users to log in, so you should also provide a way to log out. You can do this by adding the appropriate web parts to the website's master page.

1. Go to **CMS Desk -> Content**. Select the root of the content tree (*My website*) and open the **Design** tab.

2. Add the following web parts to the **zoneMain** web part zone and set their properties:

Membership -> Logon & Registration -> Current user

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

Membership -> Logon & Registration -> Sign out button

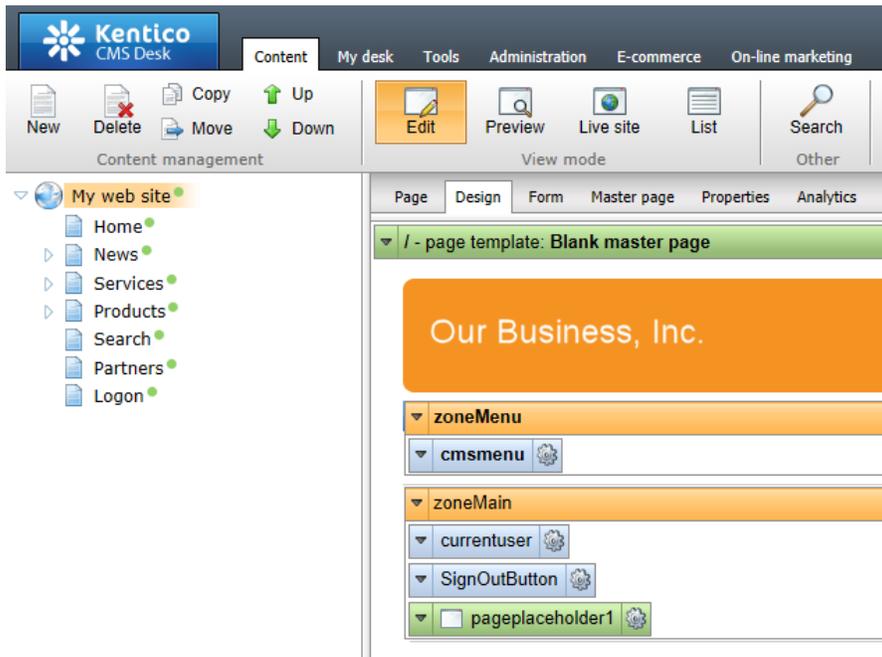
- **Content after:** `</div>`

The `<div>` tag encloses the *Current user* and *Sign out button* web parts and floats them to the right side of the page.

3. Set the following order for the web parts in **zoneMain** (you can drag web parts using your mouse):

1. Current user
2. Sign out button

3. Page placeholder



The **Sign out** button and the web part showing the current user's name are now visible for signed in users on all pages on the website.

Result - Logging in to the website

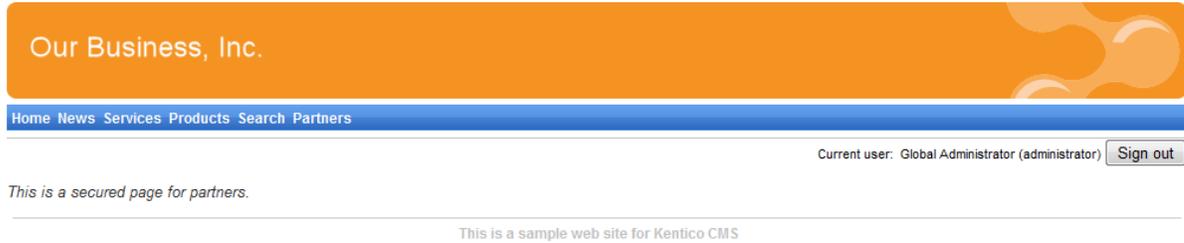
Now that you have added the logon page, secured section and sign out button to the website, you can test the new functionality from the perspective of a live site user.

1. **Sign out** of CMS Desk so you can view the website as a public visitor.
2. Click **Partners** in the main menu. The logon page appears.

 The screenshot shows the website's logon and registration page. At the top, there's a blue navigation bar with the text 'Home News Services Products Search Partners'. Below this is a large orange banner with the text 'Our Business, Inc.'. The main content area is divided into two sections. The left section is titled 'Log on' and contains a form with fields for 'User name:', 'Password:', and a 'Remember me' checkbox. There is a 'Log on' button and a link for 'Forgotten password'. The right section is titled 'Not a member yet? Sign up now!' and contains a form with fields for 'First name:', 'Last name:', 'E-mail:', 'Password:', and 'Confirm password:'. There is a 'Register' button. At the bottom of the page, there is a small text that says 'This is a sample web site for Kentico CMS'.

3. Log in as the administrator again or try registering a new account. After you sign in successfully, the

site automatically redirects you back to the **Partners** page.



Here you can see the content of the secured page, as well as the name of the current user and the **Sign Out** button.



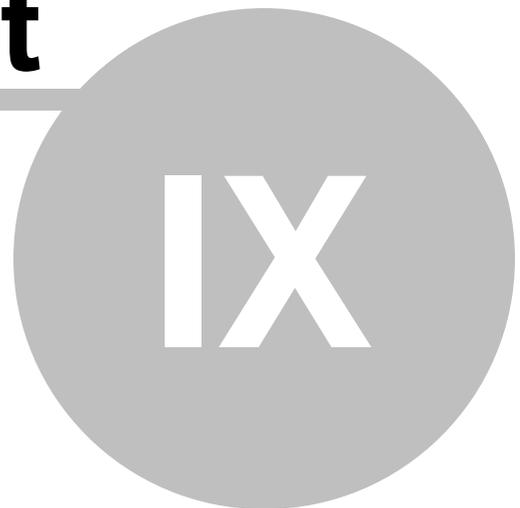
Displaying content based on user permissions

Kentico CMS also allows you to display content according to the read permissions of users. For example, you can grant the Read permission for a Gold partners section to members of the *Gold partners* role, so that only gold partners are able to see the corresponding menu item and page content.

You can find more information about permissions in the [Development -> Membership, permissions and security](#) chapter of the Developer's Guide.

This concludes the creation of 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 86

- C -

CSS styles 84
 custom code
 adding 65

- D -

Document type 120

- F -

Further steps 141

- H -

home page
 editing content 20
 modifying the structure 40
 widget customization 45

- I -

image
 inserting 25
 installation
 setup 8
 web application 9

- K -

Kentico CMS Overview 5

- L -

link 26
 logon page

configuring 135

- M -

menu design
 overview 87
 minimum requirements 7

- N -

new page 21
 new site
 CSS stylesheet 99
 home page 108
 main menu 106
 master page 102
 news page 113
 page placeholder 108
 products 120
 products - new document type 120
 products - page template 129
 products - transformations 127
 search page 132
 secured section 135
 services page 116
 web part containers 111
 wizard 91
 new site - Portal engine 91
 news item 28

- O -

Overview 5

- P -

page template
 creating 51
 modifying the layout 39
 overview 37
 re-using 54
 prerequisites 7

- S -

Setup 11

Sign out button
 configuring 137
Site Development Overview 32
Smart search 132
Support 5, 141
system requirements 7

- T -

Technical support 5, 141
Transformations 127

- U -

User interface overview 17

- V -

visual inheritance
 configuring 58
 overview 56

- W -

web installer 9
web part
 cloning 77
 container 63
 creating 71
 development overview 68
 inheritance 77
 layout 77
 modifications 77
web part zone
 adding 104
widgets
 creating 78
 customizing pages 45